UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

CONTRIBUTION OF ARTIFICIAL METAPLASTICITY TO
PATTERN RECOGNITION

DOCTORAL THESIS

Juan Fombellida Vetas
Ingeniero de Telecomunicación

2018

DEPARTAMENTO DE SEÑALES, SISTEMAS Y RADIOCOMUNICACIONES
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE MADRID

CONTRIBUTION OF ARTIFICIAL METAPLASTICITY TO PATTERN RECOGNITION

DOCTORAL THESIS

Author:

Juan Fombellida Vetas

Ingeniero de Telecomunicación

Director:

Diego Andina de la Fuente

Doctor Ingeniero de Telecomunicación

Co-Director:

José Manuel Ferrández Vicente

Doctor Ingeniero en Informática

2018

DOCTORAL THESIS
CONTRIBUTION OF ARTIFICIAL METAPLASTICITY TO PATTERN
RECOGNITION

AUTHOR: Juan Fombellida Vetas
DIRECTOR: Diego Andina de la Fuente
CO-DIRECTOR: José Manuel Ferrández Vicente

Tribunal assigned by His Excellency the Chancellor of the Universidad Politécnica de Madrid, on the day ____ of _____ of 2018.

PRESIDENT: _____

SECRETARY: _____

VOCAL: _____

VOCAL: _____

VOCAL: _____

ALTERNATE: _____

ALTERNATE: _____

Lecture and defense of the thesis performed on the day ____ of _____ on 2018.

at E.T.S. de Ingenieros de Telecomunicación.

Grade: _____

THE PRESIDENT

THE VOCALS

THE SECRETARY

To my loving family

Acknowledgements

And I would like to acknowledge the following institutions that support the development of the work included in this thesis:

Grupo de Automatización en Señales y Comunicaciones (GASC).

Universidad Politécnica de Madrid (UPM).

Abstract

Artificial Neural Networks design and training algorithms are based many times on the optimization of an objective error function used to provide an evaluation of the performances of the network. The value of the error depends basically on the weight values of the different connections between the neurons of the network. The learning methods modify and update the different weight values following a strategy that tends to minimize the final error in the network performance.

The neural network theory identifies the weight values as a representation of the synaptic weights in the biological neural networks, and their ability to change their values can be interpreted as a kind of artificial plasticity inspired by the demonstrated biological counterpart process. The biological metaplasticity is related to the processes of memory and learning as an inherent property of the biological neuron connections, and consists in the capacity of modifying the learning mechanism using the information present in the network itself. In such a way, Artificial MetaPlasticity (AMP), is interpreted as the ability to change the efficiency of artificial plasticity depending on certain elements used in the training.

A very efficient AMP model (as a function of learning time and performance) is the approach that connects metaplasticity and Shannon's information theory, which establishes that less frequent patterns carry more information than frequent patterns. This model defines AMP as a learning procedure that produces greater modifications in the synaptic weights when less frequent patterns are presented to the network than when frequent patterns are used, as a way of extracting more information from the former than from the latter. In this doctoral thesis the AMP theory is implemented using different Artificial Neural Network (ANN), models and different learning paradigms. The networks are used as classifiers or predictors of synthetic and real data sets in order to be able to compare and evaluate the results obtained with several state of the art methods. The AMP theory is implemented over two general learning methods:

- Supervised training: The BackPropagation Algorithm (BPA), is one of the best known and most used algorithms to training the neural networks. This algorithm compares the ideal results with the real results obtained at the networks output and calculates an error value. This value is used to modify the weight values in order to get a final trained network that minimizes the differences between the ideal and the real results. The BPA has been successfully applied to several pattern classification problems in areas such as: medicine, bioinformatic, banking, climatological predictions, etc. However the classic algorithm has shown some limitations that prevent this method to reach an optimal efficiency level (convergence, speed problems and classification accuracy).

Artificial Metaplasticity modification to the classic BPA, is in this case implemented in a Multilayer Perceptron (MLP), neural network. The Artificial Metaplasticity on Multilayer Perceptron (AMMLP) model was applied in the ANNs training phase. During the training phase the AMMLP algorithm updates the weights assigning higher values to the less frequent activations than to the more frequent ones. AMMLP achieves a more efficient training and improves MLP performance.

The suggested AMMLP algorithm was applied to different problems related to pattern classification or prediction in different areas and considering different methods for obtaining the information from the data set. Modeling this interpretation in the training phase, the hypothesis of an improved training shows a much more efficient training maintaining the ANN performance. This algorithm has achieved deeper learning on several multidisciplinary data sets without the need of a deep network.

- Unsupervised training: Koniocortex-Like Network (KLN) is a novel category of bio-inspired neural networks whose architecture and properties are inspired in the biological koniocortex, the first layer of the cortex that receives information from the thalamus. In the KLN competition and pattern classification emerges naturally due to the interplay of inhibitory inter-neurons, metaplasticity and intrinsic plasticity. This behavior resembles a Winner Take All (WTA) mode of operation, where the most active neuron “wins”, i.e. fires, while neighboring ones remain silent. Although a winning neuron is identified by calculation in many artificial neural networks models, in biological neural networks the winning neuron emerges from a natural dynamic

process.

Recently proposed, it has shown a big potential for complex tasks with unsupervised learning. Now for the first time, its competitive results are proved in several relevant real applications. The simulations show that the unsupervised learning that emerges from individual neurons properties is comparable and even surpasses results obtained with several advanced state-of-the-art supervised and unsupervised learning algorithms.

Table of contents

Acknowledgement	ix
Abstract	xi
List of figures	xix
List of tables	xxiii
Nomenclature	xxv
1 Introduction	1
1.1 Introduction	1
1.2 State of the Art Study	3
1.2.1 BPA Supervised Training	3
1.2.2 Unsupervised Training	9
1.3 Motivation	19
1.4 Justification	21
1.5 Objectives	23
1.5.1 General Objective	23
1.5.2 Specific Objectives	24
1.6 Impact	25
1.7 Thesis Structure	26
2 Neural Networks	29
2.1 Introduction and Motivation of the Neural Networks	29
2.2 The Biological Neuron	30
2.3 Artificial Neural Networks	34
2.3.1 Brief History of ANNs	34
2.3.2 Main Characteristics of Neural Networks	36

2.3.3	The Artificial Neuron	38
2.3.4	Basis of the Learning Methods for ANNs	42
2.3.5	Architectural Models for ANNs	44
3	Supervised Learning: The Perceptron	51
3.1	Introduction	51
3.2	Single Layer Perceptron	52
3.2.1	Learning Method for the Single Layer Perceptron	55
3.2.2	Functional Problems of the Single Layer Perceptron	58
3.3	MultiLayer Perceptron	60
3.4	The BackPropagation Algorithm	63
3.4.1	The Gradient Method	63
4	Unsupervised Learning Networks	71
4.1	Introduction	71
4.2	Hebbian Learning Rule	72
4.3	Hopfield Network	72
4.4	Competitive Learning Rules	74
4.5	Self-Organizing Map	76
4.5.1	Vector Quantization	78
4.5.2	SOM Training Method	79
4.5.3	Discussion on SOM ANN	81
4.6	Learning Vector Quantization	82
4.6.1	Type One LVQ	83
4.6.2	Type Two LVQ	84
4.6.3	Type Three LVQ	85
4.7	Adaptive Resonance Theory Models	86
4.7.1	Stability-Plasticity Problem in Competitive Learning	86
4.7.2	ART1 Model	87
4.7.3	ART2 Model	90
5	Metaplasticity Concepts	93
5.1	Introduction	93
5.2	Plasticity in Synapses	94
5.3	Intrinsic Plasticity	97
5.4	Biological Metaplasticity	98

5.5	Artificial Metaplasticity	100
5.5.1	Application of Shannon’s Information Theory to Artificial Metaplasticity	101
5.5.2	Probabilistic Computation Derived from Synaptic Behavior	104
6	AMP Applied to Supervised Learning	111
6.1	Introduction	111
6.2	AMP Implementation in MLP Training	111
6.3	Modification of the BPA	113
7	AMP Applied to Unsupervised Learning	117
7.1	Introduction	117
7.2	KLN Biological Basis	117
7.3	Sequential Implementation of AMP in the KLN Model	118
7.4	Complete KLN Model	122
7.4.1	Details of the KLN Modeling	126
8	Experiment 1: Radar Signal Detection	129
8.1	Introduction	129
8.2	Radar System	129
8.2.1	Radar System Scheme	131
8.3	Methodology: Binary Detection	132
8.3.1	Marcum Theoretical Model	135
8.4	Data Preparation	136
8.5	Network Characteristics	137
8.6	Results	139
8.7	Discussion of the Results	147
8.8	State of the Art Study	148
9	Experiment 2: Breast Cancer Data Classification	153
9.1	Introduction	153
9.2	WBCD Dataset	154
9.2.1	Data Preparation	155
9.3	AMMLP Applied to WBCD Classification	157
9.3.1	Evaluation Method	158
9.3.2	Results	160
9.3.3	Discussion of the Results	165

9.4	KLN Applied to WBCD Classification	167
9.4.1	Evaluation Method	168
9.4.2	Results	169
9.4.3	Discussion of the Results	170
9.5	State of the Art Study	170
10	Experiment 3: Credit Scoring Data Classification	175
10.1	Introduction	175
10.2	ACAD Dataset	176
10.2.1	Data Preparation	177
10.3	AMMLP Applied to ACAD Classification	179
10.3.1	Evaluation Method	180
10.3.2	Results	181
10.3.3	Discussion of the Results	186
10.4	KLN Applied to ACAD Classification	188
10.4.1	Evaluation Method	189
10.4.2	Results	189
10.4.3	Discussion of the Results	191
10.5	State of the Art Study	191
11	Experiment 4: Pollutant Concentration Prediction	197
11.1	Introduction	197
11.2	Salamanca Pollutant Concentration Dataset	198
11.2.1	Data Preparation	199
11.3	AMMLP Applied to Pollutant Prediction	201
11.3.1	Evaluation Method	202
11.3.2	Results	202
11.3.3	Discussion of the Results	207
11.4	State of the Art Study	209
12	Conclusions	215
13	Contributions, Future Research Lines and Publications	219
13.1	Contributions	219
13.2	Future Research Lines	222
13.3	Publications	222
13.3.1	Journal Publications	222

13.3.2 Congress Publications	223
--	-----

References	227
-------------------	------------

List of figures

2.1	Simple scheme of the synapses union between two neurons	31
2.2	Neurotransmitters liberation scheme	31
2.3	Complete synapses process	32
2.4	Hebbian pre-post associative mechanism	33
2.5	Representation of a McCulloch-Pitts artificial neuron	39
2.6	Example of the architecture of a feed-forward network	45
2.7	Example of the architecture of a recurrent network	46
2.8	Architecture of the MultiLayer Perceptron	47
2.9	Architecture of the Self-Organizing Map	48
2.10	Architecture of the Radial Basis Function Network	48
2.11	Basic taxonomy of the ANNs	49
3.1	Single Layer Perceptron scheme	52
3.2	AND logic function	54
3.3	OR logic function	54
3.4	Single Layer Perceptron with N neurons scheme	55
3.5	XOR logic function	59
3.6	MultiLayer Perceptron structure	60
3.7	Possible separable regions generated by a MultiLayer Perceptron	62
4.1	Hopfield network structure	74
4.2	SOM network basic structure	78
4.3	LVQ network basic structure	83
4.4	ART1 network basic structure	88
4.5	ART2 network basic structure	91
5.1	Long Term Potentiation process	95
5.2	Long Term Depression process	96

5.3	Synaptic weight modification of a biological synapse according to its post-synaptic activation	97
5.4	LTP modification of a biological synapse according to its post-synaptic activation	99
5.5	LTP and LTD modification of a biological synapse according to the change of the synaptic weight	100
5.6	Normal sampling for two Gaussian distribution corresponding to two different classes	102
5.7	Importance sampling for two Gaussian distribution corresponding to two different classes	103
5.8	Weighted training cycle. Note that the expected output will not be used for unsupervised training. Weighting function performs the artificial metaplasticity through statistical information extracted from input patterns.	105
7.1	Structure of the KLN network version 1: Bayesian decision framework . . .	119
7.2	Structure of the KLN network version 2: Artificial competition added to the model	120
7.3	Structure of the KLN network version 3: Lateral inhibition added to the model	121
7.4	Simplified structure of the KLN network final version	122
7.5	Intrinsic plasticity allows the neurons' activation function to shift horizontally so that the activation function "follows" the average net-input of the neuron. (a) Initial position of the sigmoidal activation function. (b) In the case of a low regime of net-input values (as in A, B and C), intrinsic plasticity shifts the sigmoid leftwards. (c) In the case of a high regime of net-input values (as in D, E and F), intrinsic plasticity shifts the sigmoid rightwards increasing the sensitivity of the neuron	124
7.6	Complete structure of the KLN model	125
8.1	Functional principle of a radar system	130
8.2	Simplified scheme of a radar system	132
8.3	Theoretical detection curves of the Marcum model	136
8.4	Sigmoid logarithmic activation function	137
8.5	Evolution of the classification error - BPA and AMMLP - Radar input patterns	140
8.6	Evolution of the classification error - Gaussian AMMLP - Radar input patterns	140
8.7	Probability of detection $P_{fa} = 10^{-2}$ - Gaussian AMMLP - Radar input patterns	141
8.8	Probability of detection $P_{fa} = 10^{-3}$ - Gaussian AMMLP - Radar input patterns	142

8.9	Probability of detection $P_{fa} = 10^{-4}$ - Gaussian AMMLP - Radar input patterns	142
8.10	Evolution of the classification error - Network output AMMLP - Radar input patterns	143
8.11	Probability of detection $P_{fa} = 10^{-2}$ - Network output AMMLP - Radar input patterns	144
8.12	Probability of detection $P_{fa} = 10^{-3}$ - Network output AMMLP - Radar input patterns	144
8.13	Probability of detection $P_{fa} = 10^{-4}$ - Network output AMMLP - Radar input patterns	145
8.14	Evolution of the classification error - Best network - Radar input patterns .	145
8.15	Probability of detection $P_{fa} = 10^{-2}$ - Best network - Radar input patterns .	146
8.16	Probability of detection $P_{fa} = 10^{-3}$ - Best network - Radar input patterns .	146
8.17	Probability of detection $P_{fa} = 10^{-4}$ - Best network - Radar input patterns .	147
9.1	Evolution of the classification error $\eta = 1$ - Nominal BPA - WBCD	161
9.2	Detail of the first iterations of the evolution of the classification error $\eta = 1$ - Nominal BPA - WBCD	161
9.3	ROC $\eta = 1$ - Nominal BPA - WBCD	162
9.4	Evolution of the classification error $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD	163
9.5	Detail of the first iterations of the evolution of the classification error $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD	164
9.6	ROC $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD	164
9.7	Evolution of the classification error $\eta = 23$ - Network output AMMLP - WBCD	165
9.8	Detail of the first iterations of the evolution of the classification error $\eta = 23$ - Network output AMMLP - WBCD	166
9.9	ROC $\eta = 23$ - Network output AMMLP - WBCD	166
10.1	Evolution of the classification error $\eta = 1$ - Nominal BPA - ACAD	182
10.2	Detail of the first iterations of the evolution of the classification error $\eta = 1$ - Nominal BPA - ACAD	182
10.3	ROC $\eta = 1$ - Nominal BPA - ACAD	183
10.4	Evolution of the classification error $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD	184

10.5	Detail of the first iterations of the evolution of the classification error $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD	184
10.6	ROC $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD	185
10.7	Evolution of the classification error $\eta = 23$ - Network output AMMLP - ACAD	186
10.8	Detail of the first iterations of the evolution of the classification error $\eta = 23$ - Network output AMMLP - ACAD	187
10.9	ROC $\eta = 23$ - Network output AMMLP - ACAD	187
11.1	Evolution of the MSE error $\eta = 1$ - Nominal BPA - K-Means pollutant prediction	203
11.2	Detail of the first iterations of the evolution of the MSE error $\eta = 1$ - Nominal BPA - K-Means pollutant prediction	203
11.3	Evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - K-Means pollutant prediction	204
11.4	Detail of the first iterations of the evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - K-Means pollutant prediction	205
11.5	Evolution of the MSE error $\eta = 1$ - Nominal BPA - Fuzzy C-Means pollutant prediction	206
11.6	Detail of the first iterations of the evolution of the MSE error $\eta = 1$ - Nominal BPA - Fuzzy C-Means pollutant prediction	206
11.7	Evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - Fuzzy C-Means pollutant prediction	207
11.8	Detail of the first iterations of the evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - Fuzzy C-Means pollutant prediction	208

List of tables

2.1	Differences between a traditional computer and a biological neural system .	37
2.2	Activation functions	41
9.1	Description of the attributes of the WBCD	155
9.2	Confusion matrix model	158
9.3	Confusion matrix $\eta = 1$ - Nominal BPA - WBCD	160
9.4	Confusion matrix $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD	162
9.5	Confusion matrix $\eta = 23$ - Network output AMMLP - WBCD	165
9.6	Sensitivity and accuracy evolution depending on the number of epochs - KLN - WBCD	169
9.7	Confusion matrix 1000 epochs - KLN - WBCD	169
9.8	State of the art study for WBCD classification	174
10.1	Description of the continuous type attributes of the ACAD	178
10.2	Confusion matrix $\eta = 1$ - Nominal BPA - ACAD	181
10.3	Confusion matrix $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD	183
10.4	Confusion matrix $\eta = 23$ - Network output AMMLP - ACAD	185
10.5	Sensitivity and accuracy evolution depending on the number of epochs - KLN - ACAD	190
10.6	Confusion matrix 14 epochs - KLN - ACAD	191
10.7	State of the art study for ACAD classification	196

Nomenclature

Acronyms / Abbreviations

ACAD	Australian Credit Approval Dataset
AEMN	Automatic Environmental Monitoring Network
AI	Artificial Intelligence
AIS	Artificial Immune System
AMI	Average Mutual Information
AMMLP	Artificial Metaplasticity on MultiLayer Perceptron
AMP	Artificial MetaPlasticity
ANMBP	Algorithm Neighborhood Modified BackPropagation
ANN	Artificial Neural Network
AP	Atmospheric Pressure
ARC	Average Random Choosing
ART	Adaptive Resonance Theory
AUC	Area Under the Curve
BCM	Bienenstock-Cooper-Munro
BD	Business Data
BI	Business Intelligence
BL	Boltzman Learning

BPA	BackPropagation Algorithm
BPDC	BackPropagation-DeCorrelation
BPVS	BackPropagation with Variable Stepsize
BPWE	BackPropagation by Weight Extrapolation
CF	Collaborative Filtering
CG	Conjugate Gradient
CI	Computational Intelligence
CLC	Clustering-Launched Classification
CNN	Combined Neural Network
CNN	Convolutional Neural Networks
CSBP	Cuckoo Search Back-Propagation
DBN	Deep Belief Networks
DCGAN	Deep Convolutional Generative Adversarial Networks
DDB	Dynamic of Decision Boundaries
DSA	Dynamic Self-Adaptation
EBP	Emotional BackPropagation
ECL	Error-Correction Learning
ELEANNE	Efficient LEarning Algorithms for Neural NEtworks
ELM	Extreme Learning Machine
ES	Evolution Strategies
ES	Expert System
ESP	Error Saturation Prevention
F-PM	First-Principle Model

FA	Firefly Algorithm
FBPP	Filtered BackProPagation
FCM	Fuzzy Cognitive Map
FFNN	Feed-Forward Neuronal Networks
FGBP	Fuzzy General BackPropagation
FN	False Negative
FNN	False Nearest Neighbors
FP	False Positive
FSR	Forward Scattering Radar
GA	Genetic Algorithms
GASC	Grupo de Automatización en Señales y Comunicaciones
GDAM	Gradient Descent Method with Adaptive Momentum
GDBPA	Gradient Descent Back Propagation Algorithm
GONN	Genetically Optimized Neural Network
GRBF	Generalized Radial Basis Function
HL	Hebbian Learning
HL	Hidden Layer
HMM	Hidden Markov Model
HRR	High Resolution Range
IBPLN	Incremental BackPropagation Learning Network
ICT	Information and Communication Technology
IL	Input Layer
KLN	Koniocortex-Like Network

KTPM	Kohonen Topology-Preserving Mapping
LBG	Linde-Buzo-Gray
LCFNN	Local Coupled Feedforward Neural Network
LMSE	Least Mean Square Error Reconstruction
LPEBP	Learning Phase Evaluation BackPropagation
LS-PEN	Least Squares and PENalty
LS-SVM	Least Square Support Vector Machine
LS	Least Squares
LTANN	Laplace Transform Artificial Neural Networks
LTD	Long Term Depression
LTM	Long Term Memory
LTP	Long Term Potentiation
LVQ	Learning Vector Quantization
MAE	Mean Absolute Error
MAR	Multivariate Adaptive Regression
MCQP	Multicriteria Convex Quadric Programming
MLEANN	Meta-Learning Evolutionary Artificial Neural Network
MLP	MultiLayer Perceptron
MSE	Mean Square Error
NIDS	Network Intrusion Detection Systems
OL	Output Layer
OSD	Optimum Steepest Descent
PBIL	Population-Based Incremental Learning

PCA	Principal Component Analysis
PCNN	Pulse Coupled Neural Networks
PDF	Probability Density Function
PD	Probability of Detection
PDP	Parallel Distributed Processing
PFA	Probability of False Alarm
PNN	Probabilistic Neural Network
PNN	Pruned Neural Network
P	Precipitation
RBFN	Radial Basis Function Networks
RBPA	Robust BackPropagation Algorithm
RF	Radio Frequency
RH	Relative Humidity
RIAC	Rule Induction through Approximate Classification
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RS	Rough Set
SAR	Synthetic Aperture Radar
SCBP	Split-Complex BackPropagation
SDP	Semi-Defined Programming
SD	Steepest Descent
SLP	Single Layer Perceptron
SNR	Signal to Noise Ratio

SOFM	Self-Organizing Feature Maps
SOINN	Self-Organizing Incremental Neural Network
SOM	Self-Organizing Map
SR	Solar Radiation
STDP	Spike Timing Dependent Plasticity
STM	Short Term Memory
TD-FALCON	Time Differences Fusion Architecture for Learning, COgnition, and Navigation
TN	True Negative
TP	True Positive
T	Temperature
UPM	Universidad Politécnica de Madrid
VQ	Vector Quantization
WBCD	Wisconsin Breast Cancer Dataset
WD	Wind Direction
WS	Wind Speed
WTA	Winner Take All

Chapter 1

Introduction

1.1 Introduction

Artificial Neural Networks (ANNs) are inspired in the biological neural networks present in the human brain. These ANNs are integrated by different elements that can be identified with the ones present in the biological networks, in order to implement the same functions performed in the real brain structure. There are multiple possible structures for organizing these elements that try to replicate different biological schemes.

The ANNs try to replicate not only the structure of the human brain but also some of its more important characteristics and properties. Among the more relevant ones of these characteristics, the ANNs models are able to learn from experience, obtaining information from examples, being able to project this information to new situations, and getting abstract relationships from data sets.

ANN design and training algorithms are many times based on the optimization of an objective error function used to provide an evaluation of the performances of the network. The knowledge paradigm of the network is included not only in the structure and the elements but also in the relation between these elements. These relations are often represented as the weights of the connections between the artificial neurons. The learning methods modify and update the different weight values following a strategy that tends to minimize the final error in the network performance.

In order to minimize this error several learning algorithms have been proposed by different authors. Those algorithms in general can be divided between supervised and unsupervised

depending on if the ideal result for the network output is used in the training or not. In both cases the algorithms present limitations related to convergence, learning speed and generalization capabilities of the final network obtained.

For the supervised learning algorithm class, one of the most popular methods is the BackPropagation Algorithm (BPA). This algorithm have been applied to different structures and data sets with relative success. Several authors have identified problems inherent to this algorithm related to a long convergence time or the possibility of getting stuck in a error local minimum without reaching the error function global minimum [1],[2],[3]. For the unsupervised model the most common structure is the self competitive model where the system tries to extract information from the data set and provide a response exciting one neuron over the others in the output layer. This Winner Takes All (WTA) approach does not use any information of the expected ideal output from the network and the real output is determined by resemblance to the neighbors. However, there are some issues associated with unsupervised learning scheme: one problem is that depending on the structure it is difficult for the network to determine the stopping point, the other main problem of unsupervised learning networks is that information extracted from such networks is less accurate than using other supervised learning schemes [4],[5].

Considering these limitations several proposals have been developed by the scientist community presenting modifications or variations of the basic algorithms. Some of those proposal are focused on solving the convergence speed problems meanwhile others try to obtain better results for bigger data sets, being able to apply the methods for more general applications. In general it can be said that no one of those modifications is able to provide a satisfactory solutions for all the problems in terms of results and performance. The majority of the modifications proposed imply more calculation, selecting some parameters that limit the application area, the necessity of previous knowledge of the data set that it is not always available. So getting a global solution for the identified problems is still a state-of-the-art problem for the researchers community.

This doctoral thesis includes the proposal and development of several neural models based in the biological property called Metaplasticity in order to solve or at least to improve the results related to the known problems of the previously exposed neural network learning algorithms. Metaplasticity is a biological concept widely studied in different areas like biology, physiology, medical science, neurology, neuroscience and psychologies among others,

and it is a matter of continuous research [6], [7], [8], [9], [10]. One of the most important advantages of the Artificial Metaplasticity (AMP) method is that it can be implemented in different ANN models. In this thesis we have included this theory in two different models:

- Multilayer Perceptron (MLP) network: For the supervised learning method we have developed a model called Artificial Metaplasticity on Multilayer Perceptron (AMMLP).
- Self-organized WTA network: For the unsupervised learning method we have developed a model called Koniocortex-Like Network (KLN) due to its resemblance with the koniocortex layer of the human brain.

AMP modeling is introduced in the training phase of the learning methods applied to the ANNs. Among the different models that uses AMP presented in the literature it is considered that the a very efficient one from the learning and performances points of view is the one that connects metaplasticity with Shannon information theory [11]. During the training this application gives more importance to the less frequent input patterns and subtracts influence to the most frequent ones in order to obtain more information from the data set. In this way we can get to a more efficient method while maintaining the quality of the results. Different applications of this AMP theory to real data sets can be found on [12], [13], [14] and [15].

1.2 State of the Art Study

In this section a review of the state-of-the-art research is presented considering the two different types of learning paradigm used in this doctoral thesis. The study covers from the year 1990 to 2017.

1.2.1 BPA Supervised Training

The BPA algorithm has been applied to several real problems with successful results although some difficulties are still present. The following paragraphs present a selection of the modifications proposed by different authors in order to solve the problems inherent to the learning algorithm.

- In 1990 Leonard & Kramer [16] developed a method based on the Conjugate Gradient (CG), that consists in an application to some example data sets of a linear searching method using the descendent gradient in the conjugate direction. The authors showed that this CG method can be considered a BPA modification that uses a dynamic adjustment of the learning rate and the momentum value.

- In 1991 Lee & Weidman [17] stated that in order to improve the results of the ANN training it was needed to use Expert Systems (ES). It was declared that using a ES for supervising the training was more efficient than using the classic training method. Kim & Ra [18] formulated an algorithm called Dynamic of Decision Boundaries (DDB) to select the initial values for the weights, in this way stability and process speed is improved.
- In 1992 Scalero & Tepedelenlioglu [19] developed an algorithm to improve the results of the classic BPA based on the minimization of the Mean Square Error (MSE) calculated between the ideal and the real outputs considering the sum of the values instead of calculating the MSE for each weight value. Karayiannis & Venetsanopoulos [20] proposed a general criteria for the training of the single/multi layer Feed-Forward Neuronal Networks (FFNN) based in the delta rule. Those algorithms were called Efficient LEarning Algorithms for Neural Networks (ELEANNE).
- In 1993 Anand *et al* [21] analyzed the convergence speed problem inherent to BPA used as classifier for two data classes. The author proposed an alternative Descent Vector (DV) approach that calculates a vector that points to the down direction for both classes so the error is minimized due to the weight movement across the vector direction.
- In 1994 Chen & Jain [22] proposed an algorithm called Robust BackPropagation Algorithm (RBPA) that is able to resist the noise effects and to avoid the greater part of the errors during the approach phase. This algorithm presents several advantages over the classic BPA as using correlation of data instead of interpolation of training patterns, robustness against errors and an improved convergence rate.
- In 1995 Alpsan *et al* [23] made a comparative study for the methods proposed for improving the application of the classic BPA to real medical problems based on optimization processes. The author concludes that the basic method can be considered quick enough or providing good generalized results depending on the concrete problem type.
- In 1996 Solomon & Van-Hemmen [24] proposed a new genetic algorithm based on Dynamic Self-Adaptation (DSA) in order to improve and speed up the BPA learning phase. The algorithm takes the learning rate from the previous step and generates small variations in order to minimize a cost function. Fu *et al* [25] presented a new incremental method to be used for pattern recognition called Incremental BackPropagation

Learning Network (IBPLN) that uses a limited modification of the weights and an structural adaptation of the learning rules by applying a initial knowledge of the data set to limit the variation possibilities.

- In 1997 Magoulas [26] *et al* developed the BackPropagation with Variable Stepsize (BPVS) method based on a modification of the Steepest Descent (SD) that allows using a variable size for the step. Yam *et al* [27] formulate a new view of the Least Squares (LS) to calculate the initial optimum values for the ANNs weights. Doing so the initial error is minor and the number of iterations for the global training decreases dramatically.
- In 1998 Sexton *et al* [28] used Genetic Algorithms (GA) to improve the generalization results of the classic BPA. In this way the authors demonstrate that a reconstruction of the architecture is not needed if the initial one is complex enough and it is adapted to a global searching algorithm.
- In 1999 Kamarthi & Pittne [29] proposed an improvement of the BPA applied to feed forward networks. This method is called BackPropagation by Weight Extrapolation (BPWE) and it is based on the concept that if the weights are extrapolated it is possible to reduce the number of iterations of the process. Cho & Chow [30] formulated an algorithm of global hybrid learning based on LS and the searching with PENalty method (LS-PEN). The LS part is used to determine the weights between the input and hidden layer meanwhile the penalty optimization is used for the weights between the hidden and output layer. Ampazisa *et al* [31] proposed a dynamic system that is able to speed up the learning phase by reducing the time expended searching for the minimum in the vicinity calculations.
- In 2000 Yam & Chow [32] developed an algorithm to determine the initial optimal values for the weights included in the feed forward networks based on Cauchy inequality and a linear algebraic method. This algorithm guarantees that the neurons results are included in the active region so the convergence rate is increased. Chaudhuri & Bhattacharya [3] proposed a method to speed up the convergence rate of the BPA based on an intelligent selection of the training patterns. This method did not imply any modification of the classic BPA but it showed good results in the cases where the classes are not easily separable.
- In 2001 Lee *et al* [33] proposed an alternative to the descendent gradient called Error Saturation Prevention (ESP) to avoid that the error in the output reaches a saturated

value not useful for the network learning. The authors applied this method also to the neurons located in the internal layers to adjust the learning terms.

- In 2002 Mandische [34] provided a method for the learning evolution using Evolution Strategies (ES) as an alternative to the methods based in the gradient. The main advantage is that this method can be used in networks with derivable activation function. Hoo *et al* [35] proposed to use the information present in the First-Principle Model (F-PM) to give some directional meaning to the estimation provided by the ANN. This is achieved by modifying the objective function including an additional term related to the differences between the estimated results and the model outputs during the first steps of the training.
- In 2003 Eom *et al* [36] proposed the Fuzzy General BackPropagation (FGBP) method to improve the BPA results with a fuzzy logic system that automatically adjust the gain parameter of the activation function. Zweiri *et al* [37] added a new proportional factor to the learning rate and momentum parameter values, generating a three-term BPA that was more robust to a wrong election of the initial weights considering the concrete values selected for the three training parameters.
- In 2004 Abraham [38] used the evolutionary algorithm theory to adapt the BPA creating the Meta-Learning Evolutionary Artificial Neural Network (MLEANN) for a adaptive optimization of the parameters of the network considering the concrete problem. Wang *et al* [39] proposed a variation of the classic algorithm that uses different activation functions in the neurons from the hidden layer avoiding the network to get stuck in a local minimum.
- In 2005 Pernía-Espinoza *et al* [40] proposed an improvement of the BPA training that estimated the scale to be used, considering it a variable that depended on a Huber function of the errors obtained in each iteration.
- In 2006 Steil [41] proposed an algorithm called BackPropagation-DeCorrelation BPDC to test and supervise the stability for big networks that only use adaptation for the output layer. The method combines the backpropagation of errors, the use of a temporal dynamic memory adapted to the decorrelation of the activations and the use of some non-adaptive internal neurons to reduce complexity. Behera *et al* [42] formulated two variations of the BPA based on the update of the weights using the Lyapunov function. The variation substitutes the fixed learning rate for an adaptive one.

- In 2007 Wang *et al* [43] proposed an interactive model to improve the performance of the classic BPA. The model successfully combined an adaptive learning rate with variations on the frequency used to update the weight values.
- In 2008 Khasman [44] presented a modification of the BPA called Emotional Back-Propagation (EBP) that it is based in two emotion that the author considers that affect to learning: anxiety and confidence. When a new task is learned the anxiety is high at the beginning meanwhile the confidence level is low. With the practice this situation changes due to positive feedback. Yang *et al* [45] used a Split-Complex BackPropagation (SCBP) to increase the initial values for the neuron weights compared to the adjustment quantities. Soliman & Mohamed [46] proposed a modification of the BPA based on the matrix multiplication for parallel processing. The authors used a scalar instruction set architecture and a similar vector set.
- In 2009 Cheng & Park [47] developed and algorithm to improve the performance of the BPA called Learning Phase Evaluation BackPropagation (LPEBP). The algorithm divides the learning stage in several phases and evaluates the results at the end of each phase. Kathirvalavakumar & Jeyaseeli [48] presented a training algorithm called Algorithm Neighborhood Modified BackPropagation (ANMBP) for ANNs with hidden layers based in the vicinity of the network structure, and a substitution of the fixed learning parameters with adaptive ones. This method is more efficient in terms of training error, memory and training time. Bai *et al* [49] formulated the BP algorithm with a variable slope of the activation function based on the usage of different learning rates applied to the slope of the function. This simple change showed that the classic BPA can obtain good performance results with the commented adjustment.
- In 2010 Sun [50] formulated an algorithm called Local Coupled Feedforward Neural Network (LCFNN) that assign to each hidden node a direction in the input space so each input patten only activates the nearby nodes. The dimension on the search in the input space and the computational load are not affected by increasing the size of the network.
- In 2011 Örkücü & Bal [51] compared the results obtained with the classic BPA with the results obtained using GA. The experimental comparison are examined by 10 real world data sets and a large scale simulation data. A comparative analysis on the real data sets and simulation data shows that the GA may offer efficient alternative to traditional training methods for the classification problem. Rehman *et al* [52] used

a modified backpropagation neural network in real data concerning noise-induced hearing loss problem. This study focuses on proposing a new framework on using Gradient Descent Back Propagation Algorithm (GDBPA) model with an improvement on the momentum value to identify the important factors that directly affect the hearing ability of industrial workers. Srikant *et al* [53] used an improvement in the classic network using simulated annealing, which can automatically and effectively optimize the network architecture, as opposed to the conventional trial and error BPA method.

- In 2012 Rheman & Nawi [54] proposed an algorithm for improving the current working performance of BPA by adaptively changing the momentum value and at the same time keeping the gain parameter fixed for all nodes in the neural network. The efficiency of the proposed method is demonstrated by simulations on three classification problems and results show that Gradient Descent Method with Adaptive Momentum (GDAM) is better for the classification problems with the classic BPA. Li *et al* [55] presented the analysis of the characteristics and mathematical theory of BPA neural network and also points out the shortcomings of BPA as well as several methods for improvement.
- In 2013 Solanki & Jethva [56] proposed a modified BPA based on minimization of the sum of the squares of errors. The algorithm was implemented on benchmark XOR problem with weights randomly obtained at every run inside a concrete range in order to check robustness. The test obtained better performance than the classic one concerning number of iterations and speed of convergence. Modh *et al* [57] used nature inspired meta-heuristic algorithms that provide a derivative-free solution to optimize complex problems. The author proposed a new meta-heuristic search algorithm, called cuckoo search, based on cuckoo bird's behavior to train BP in achieving fast convergence rate and to avoid local minimum problem. The performance of the proposed Cuckoo Search Back-Propagation (CSBP) is compared with BPA algorithm with OR and XOR data sets and the simulation results show that the computational efficiency of BPA training process is highly enhanced when coupled with the proposed hybrid method.
- In 2014 Kavousi-Fart *et al* [58] proposed an hybrid method based on Firefly Algorithm (FA) and ANNs to reach a reliable and accurate forecasting model. The proposed method makes use of both the learning ability of ANN and the search ability of FA to create a nonlinear mapping between the input and output pattern data. The work preserves a good balance between ANN traditional training techniques such as BPA method and evolutionary random search ability of FA in a hybrid framework.

Salari *et al* [59] combined k-Nearest Neighbor algorithms, GA, and ANNs in the implementation of a novel hybrid feature selection-classification model.

- In 2015 Huang *et al* [60] investigated on an iteration optimization approach integrating BPA with GA. The main idea of the approach is that a BPA model is first developed and trained using fewer learning samples, then the trained BPA model is solved using GA in the feasible region to search the model optimum. The result of the verification conducted based on this optimum configuration is added as a new sample into the training pattern set for the training of the BPA model.
- In 2016 Kostencka & Kozacki [61] proposed a Filtered BackProPagation (FBPP) as a reconstruction technique that is used in diffractive holographic tomography. The major advantage of the algorithm is the space-domain implementation, which enables avoiding the error-prone interpolation in the spectral domain. Kumar *et al* [62] investigated the suitability of using backpropagation neural networks for the task of hand written character recognition. The author utilized new MSE with a regularization function for implementing BPA neural network while employing hand written characters. Witesty [63] proposed to use BPA for detecting polycystic ovary syndrome using 2D and 3D images. The algorithm is able to extract information from the images and insert it the BPA algorithm.
- In 2017 Luo *et al* [64] considered a bidirectional BPA. The normal BPA propagates errors from the output layer to the hidden layers in an exact manner using the transpose of the feedforward weights. The author proposed a biologically plausible paradigm of neural architecture with two bidirectional learning algorithms with trainable feedforward and feedback weights. The feedforward weights are used to relay activations from the inputs to target outputs. The feedback weights pass the error signals from the output layer to the hidden layers.

1.2.2 Unsupervised Training

Different kinds of unsupervised methods for training neural networks are reviewed in the following section in order to have a global view of the different areas covered. As can be seen many of the process proposed are related to modeling of the different areas of the human brain.

- In 1990 Krishnamurthy *et al* [65] proposed the use of neural networks for Vector Quantization (VQ). The authors showed how a collection of neural units can be used efficiently for VQ encoding, with the units performing the bulk of the computation in parallel, and described two unsupervised neural network learning algorithms for training the VQ. The VQ codewords were determined in an adaptive manner, compared to the popular Linde-Buzo-Gray (LBG) training algorithm, which requires that all the training data is processed in a batch mode. The neural network approach allowed the adaptation to the changing statistics of the input data. Huntsberger & Ajjimarangsee [66] presented the problem of poor separability of input vectors for neural network training. This paper introduced four new algorithms based on the Kohonen Self-Organizing Feature Maps (SOFM) which were capable of generating a continuous output considering the different inputs presented as patterns to the networks.
- In 1991 Lin & Lee [67] presented a general neural network connectionism model for fuzzy logic control and decision systems. This connectionism model, in the form of feedforward multilayer net, combines the idea of fuzzy logic controller and neural network structure and learning abilities into an integrated neural network-based fuzzy logic control and decision system. A fuzzy logic control decision network is constructed automatically by learning the training examples itself. By combining both unsupervised (self-organized) and supervised learning schemes, the learning speed converges much faster than the original BPA. Rose *et al* [68] introduced an unsupervised neural network method, Kohonen Topology-Preserving Mapping (KTPM), applied to a wide matrix of physicochemical property data. Kohonen mapping compared favorably with non-linear unsupervised statistical pattern recognition methods for 2D representation of compound similarity and for classification. Burke & Rangwala [69] discussed the application of neural network-based pattern recognition techniques for monitoring the metal-cutting process. The application of unsupervised neural network learning method Adaptive Resonance Theory (ART) to pattern recognition of sensor signal features provided good classification accuracy considering the input data set.
- In 1992 Intrator [70] proposed a method that used a unsupervised neural network for dimensionality reduction that seeks direction emphasizing multi-modality. This leads to a statistical insight into the synaptic modification equations governing learning in Bienenstock-Cooper-Munro (BCM) neurons. Samad & Harp [71] showed how the Kohonen SOFM model can be extended so that partial training data can be utilized. Given input stimuli in which values for some elements or features are absent, the match

computation and the weight updates are performed in the input subspace defined by the available values. Some examples in which data is inherently incomplete are presented to demonstrate the effectiveness of the extension.

- In 1993 Xu [72] proposed a new self-organizing net based on the principle of Least Mean Square Error Reconstruction (LMSER) of an input pattern. The author shows that the LMSER rule let the network's weights to converge to rotations of the data first principal components. These converged points are stable and corresponding to the global minimum in the MSE landscape, which has many saddles but no local minimum.
- In 1994 Balakrishana *et al* [73] made a empirical comparison between neural networks using Kohonen learning with a traditional clustering method (K-means) in an experimental design using simulated data with known cluster solutions. Two types of neural networks were examined, both of which used unsupervised learning to perform the clustering. Generally, the K-means procedure had fewer points misclassified while the classification accuracy of neural networks worsened as the number of clusters increase. Rigoll [74] proposed an approach for a hybrid connectionism-Hidden Markov Model (HMM) speech recognition system based on the use of a neural network. The neural network is trained with a new learning algorithm that it is an unsupervised learning algorithm for perceptron-like neural networks that are usually trained in the supervised mode. The neural network is not trained using the standard BPA but using instead a newly developed self-organizing learning approach.
- In 1995 Carpenter & Ross [75] proposed a new neural network architecture for the recognition of patten classes after unsupervised learning. Applications include spatial-temporal image understanding and prediction and 3D object recognition from a series of ambiguous 2D views. The architecture achieves a synthesis of ART and spatial and temporal evidence integration for dynamic predictive mapping. Malakooti & Yang [76] developed a unsupervised learning clustering neural network method for solving machine-part group formation problems. The authors modified the competitive learning algorithm by using the generalized Euclidean distance, and a momentum term in the weight vector updating equations. The cluster structure can be adjusted by changing the coefficients in the generalized Euclidean distance. Thomopoulos *et al* [77] created a self-organizing artificial neural network that exhibits deterministically reliable behavior to noise interference, considering that the noise does not exceed

a pre-specified level of tolerance. The complexity of the proposed ANN, in terms of neuron requirements versus stored patterns, increases linearly with the number of stored patterns and their dimensionality. The self-organization is based on the idea of competitive generation and elimination of attraction in the pattern space.

- In 1996 Becker & Plumbey [78] reviewed the unsupervised neural network learning procedures used to pre-process raw data to extract features information for classification. The learning algorithms reviewed here are grouped into three sections: information-preserving methods, density estimation methods, and feature extraction methods. Cichocki & Unbehauen [79] developed two unsupervised, self-normalizing, adaptive learning algorithms for robust blind identification and separation of independent source signals. One of these algorithms is developed for on-line learning of a single-layer feed-forward neural network model and a second one for a feedback (fully recurrent) neural network model. The authors indicated that the algorithms ensure the separation of extremely weak or badly scaled stationary signals, as well as a successful separation even if the mixture matrix is very ill-conditioned.
- In 1997 Zheng *et al* [80] presented a modified self-organizing map with nonlinear weight adjustments applied to reduce the number of breast biopsies necessary for breast cancer diagnosis. Tissue features information representing a hyperspace of data points is used as inputs to the self-organizing map that objectively segments population distributions of lesions and accurately establishes benign and malignant regions. The experimental results also suggest that the modified self-organizing map provided more accurate population distribution maps than conventional Kohonen maps.
- In 1998 Roussinov & Chen [81] presented a research in which the authors developed a scalable textual classification and categorization system based on the Kohonen's Self-Organizing Map (SOM) algorithm. The proposed data structure and algorithm took advantage of the sparsity of coordinates in the input vectors and reduced the computational complexity by several order of magnitude. Srinivasan *et al* [82] proposed a practical implementation of a hybrid short-term electrical load forecasting model for a power system control center. This hybrid architecture incorporates a Kohonen self-organizing feature map with unsupervised learning for classification of daily load patterns, a supervised backpropagation neural network for mapping the temperature/load relationship, and a fuzzy ES for post-processing of neural network outputs.

- In 1999 Kim [83] presented a fuzzy neural network which utilizes a similarity measure of the relative distance and a fuzzy learning rule. The fuzzy learning rule consists of a fuzzy membership value, an intra-cluster membership value, and a function of the number of iterations. The proposed fuzzy neural network updated weights of all committed output neurons regardless of winning or losing without any knowledge of the ideal classification result. Christodoulou & Pattichis [84] used an unsupervised schema for the recognition of patterns related to electromagnetic clinical signals. The system is based in an ANN technique with unsupervised learning, using a modified version of the SOFM. Törönen *et al* [85] applied a SOM for the analysis and organization of published data of yeast gene expression and showed the possibilities of the SOM for the analysis and visualization of gene expression profiles.
- In 2000 Chon *et al* [86] presented an analysis of patterns of temporal variation in community dynamics conducted by combining two unsupervised ANNs, the ART and the Kohonen network. The sampled data was initially trained by ART, the weights of which preserved conformational characteristics during the process of the training. Subsequently these weights were rearranged sequentially and were provided as input to the Kohonen network to reveal temporal variations. Weber *et al* [87] obtained a method to learn object class models from unlabeled and unsegmented cluttered scenes for the purpose of visual object recognition. The variability within a class is represented by a joint probability density function (PDF) on the shape of the constellation and the output of part detectors.
- In 2001 Cecchi *et al* [88] elaborated a neural network model based in adult neurogenesis process. The authors presented a scheme in which incorporation of new neurons proceeds at a constant rate, while their survival is activity-dependent and thus contingent on new neurons establishing suitable connections. The use of a simple mathematical model following these rules organizes its activity so as to maximize the difference between its responses, and to adapt to changing environmental conditions in unsupervised way in agreement with the neurophysiological data from the biological model.
- In 2002 Spratling & Johnson [89] worked on the idea of how the integration of lateral inhibition methods affects the results of unsupervised learning. An alternative neural network architecture was presented in which nodes compete for the right to receive inputs rather than for the right to generate outputs. This form of competition,

implemented through pre-integration lateral inhibition, provided appropriate coding properties that can be used to learn such representations efficiently. Bohte *et al* [90] demonstrated that spiking neural networks encoding information in the timing of single spikes are capable of computing and learning clusters from realistic data. The authors presented how a spiking neural network based on spike-time coding and Hebbian learning can successfully perform unsupervised clustering on real-world data and evaluated how temporal synchrony in a multilayer network can induce hierarchical clustering.

- In 2003 Papageorgiou *et al* [91] modified a technique for modeling systems: the Fuzzy Cognitive Map (FCM). This is a soft computing technique that combines the theories of neural networks and fuzzy logic. The methodology of developing FCMs is easily adaptable but relies on human experience and knowledge, and thus FCMs exhibit weaknesses and dependence on human experts. In order to overcome these deficiencies a possible solution presented was the utilization of the unsupervised Hebbian algorithm to nonlinear units for training FCMs. Anastasio & Patton [92] worked in modeling a corticotectal like system. Connection weights from primary and modulated inputs are trained in stages one (Hebb) and two (Hebb-anti-Hebb), respectively, of an unsupervised two-stage algorithm. Two-stage training caused the units to extract information concerning simulated targets from their inputs. The correspondence between model and data suggested that the training captured important features of self-organization in the real corticotectal system. Enfadil & Isa [93] presented an approach for automated knowledge acquisition system using Kohonen SOMs and k-means clustering. The verification of the produced knowledge was done by comparison with conventional ES.
- In 2004 Lücke & von der Malsburg [94] studied a model of the cortical macrocolumn consisting of a collection of inhibitory coupled minicolumns. The proposed system overcame several severe deficits of systems based on single neurons as cerebral functional units. Minicolumns showed to be able to organize their collective inputs without supervision by Hebbian plasticity into selective receptive field shapes, thereby becoming classifiers for input patterns. Arleo *et al* [95] presented a state space representation constructed by unsupervised Hebbian learning during exploration. As a result of learning, a representation of the continuous 2D manifold in the high-dimensional input space was found. The visual scene was modeled using the responses of modified Gabor filters placed at the nodes of a sparse Log-polar graph.

- In 2005 Pugh *et al* [96] used particle swarm optimization on problems with noisy performance evaluation, focusing on unsupervised robotic learning. The authors adapted a technique of overcoming noise used in GA for use with particle swarm optimization, and evaluated the performance of both the original algorithm and the noise-resistant method for several numerical problems with added noise, as well as unsupervised learning of obstacle avoidance.
- In 2006 Kuo *et al* [97] made an study dedicated to a novel two-stage method, which first uses a SOM neural network to determine the number of clusters as the starting point, and then uses genetic K-means algorithm to find the final solution. Papageorgiou *et al* [98] identified the FCM model problems and proposed to restructure the system through adjustment the weights of FCM interconnections using specific learning algorithms for FCMs. Two unsupervised learning algorithms are presented and compared for training FCMs and how they define, select or fine-tuning weights of the causal interconnections among concepts. The simulations results of training the process system verified the effectiveness, validity and advantageous characteristics of those learning techniques. Zhang & Zulkernine [99] worked in an anomaly detection for Network Intrusion Detection Systems (NIDS). Most anomaly based NIDSs employ supervised algorithms, whose performances highly depend on attack-free training data. However, this kind of training data is difficult to obtain in real world network environment that leads to high false positive rate of supervised NIDSs. The authors applied one data mining algorithms called random forests algorithm in anomaly based NIDSs. Without attack-free training data, random forests algorithm can detect outliers in data sets of network traffic.
- In 2007 Masquelier & Thorpe [100] developed a rule called Spike Timing Dependent Plasticity (STDP) that modifies synaptic strength as a function of the relative timing of pre and post-synaptic spikes. When a neuron is repeatedly presented with similar inputs, STDP is known to have the effect of concentrating high synaptic weights on afferents that systematically fire early, while post-synaptic spike latencies decrease. Memisevic & Hinton [101] described a probabilistic model for learning distributed representations of image transformations. The model was defined as a gated conditional random field that is trained to predict transformations of its inputs.
- In 2008 Lee *et al* [102] presented an unsupervised learning model that mimics certain properties of visual area V2 of the biological brain. The model included a sparse variant

of the Deep Belief Networks (DBN) with two layers of nodes: the first layer, similar to prior work on sparse coding, results in localized, oriented, edge filters, and the second layer encodes correlations of the first layer responses in the data. Tan *et al* [103] presented a neural architecture for learning category nodes encoding mappings across multimodal patterns involving sensory inputs, actions, and rewards. By integrating ART models and temporal difference methods, the proposed neural model, called TD Fusion Architecture for Learning, COgnition, and Navigation (TD-FALCON), enabled an autonomous agent to adapt and function in a dynamic environment with immediate as well as delayed evaluative feedback (reinforcement) signals.

- In 2009 Tsui [104] applied a unsupervised learning technique to the problem of hardware variance degradation of the WiFi based localization systems. Although manual adjustment was able to reduce positional error, it was proposed to use an unsupervised learning method to automatically solve the hardware variance problem.
- In 2010 Shen & Hasegawa [105] presented a Self-Organizing Incremental Neural Network (SOINN). SOINN was able to represent the topology structure of input data, incrementally learn new knowledge without destroy of learned knowledge, and process online non-stationary data. It was free of prior conditions such as a suitable network structure or network size, and it was also robust to noise. SOINN has been adapted for unsupervised learning, supervised learning, semi-supervised learning, and active learning tasks. Chang *et al* [106] studied the phenomenon of evaporation and the effects on the distribution of water in the hydrological cycle. The authors proposed a SOM to assess the variability of daily evaporation based on meteorological variables. The daily meteorological data sets from a climate gauge were collected as inputs to the SOM and then were classified into a topology map based on their similarities to investigate their relationships to assess their effort in the evaporation. To accurately estimate the daily evaporation based on the input pattern, the weights that connect the clustered centers in a hidden layer with the output were trained by using the least square regression method.
- In 2011 Lee *et al* [107] studied the use of unsupervised learning of hierarchical generative models such as DBN; however, scaling such models to full-sized, high-dimensional images remained as a difficult problem. To address this problem, the authors presented the convolutional DBN, a hierarchical generative model that scales to realistic image sizes. Socher *et al* [108] introduced a novel machine unsupervised

learning framework based on recursive auto-encoders for sentence-level prediction of sentiment label distributions. The method learns vector space representations for multi-word phrases. Fernández-Navarro *et al* [109] proposed a methodology for training a new model of ANN called the Generalized Radial Basis Function (GRBF) neural network. This model was based on generalized Gaussian distribution, which modified the classic Gaussian distribution by adding a new parameter. The model parameters were optimized through a modified version of the Extreme Learning Machine (ELM) algorithm. The model obtained better results in accuracy than the corresponding sigmoidal, hard-limit, triangular basis and radial basis functions for almost all data sets, producing the highest mean accuracy rank when compared with these other basis functions.

- In 2012 Hsu [110] proposed an unsupervised recognition system for single-trial classification of motor imagery and electroencephalogram data. Competitive Hopfield Neural Network (CHNN) clustering was used for the discrimination of left and right data posterior to selecting active segment and extracting fractal features in multi-scale. CHNN clustering was adopted to recognize extracted features. Boniecki *et al* [111] used the classification properties of Kohonen-type networks for a neural model for the quality based identification of vegetables. The resulting empirical data were subsequently used to draw up a topological SOFM which features cluster centers of comparable cases.
- In 2013 Sermanet *et al* [112] worked in the pedestrian detection problem. Adding to the list of successful applications of deep learning methods to vision, the author reported state-of-the-art and competitive results on all major pedestrian data sets with a convolutional network model. The model used a few new twists, such as multi-stage features, connections that skip layers to integrate global shape information with local distinctive motif information, and an unsupervised method based on convolutional sparse coding to pre-train the filters at each stage.
- In 2014 Cui *et al* [113] recognized and predicted temporal sequences of sensory inputs coming from natural environments. Based on many known properties of cortical neurons: hierarchical temporal memory sequence memory was recently proposed as a theoretical framework for sequence learning in the cortex. In this paper, the authors analyzed properties of sequence memory and applied it to sequence learning and prediction problems with streaming data. The model was able to continuously learn a large number of variable-order temporal sequences using an unsupervised

Hebbian-like learning rule. Peng *et al* [114] focused on the unsupervised learning problem within membrane computing, and proposed an innovative solution inspired by membrane computing techniques, the fuzzy membrane clustering algorithm. An evolution–communication system with nested membrane structure was the core component of the algorithm. The feasible cluster centers were represented by means of objects, and three types of membranes were considered: evolution, local store, and global store. Based on the designed membrane structure and the inherent communication mechanism, a modified differential evolution mechanism was developed to evolve the objects in the system.

- In 2015 Radford *et al* [115] presented unsupervised learning with Convolutional Networks (CNNs) applied to computer vision applications. In this work it was tried to reduce the gap between the success of CNNs for supervised learning and unsupervised learning. The authors introduced a class of CNNs called deep Convolutional Generative Adversarial Networks (DCGAN), that have certain architectural constraints specifically adapted to provide reliable and robust results in unsupervised learning.
- In 2016 Kheradpisheha *et al* [116] presented a computational model to mimic the hierarchical processing of the ventral visual pathway of the human brain. The model is an asynchronous feedforward spiking neural network. When the network was presented with natural images, the neurons in the entry layers detect edges, and the most activated ones fire first, while neurons in higher layers are equipped with spike timing-dependent plasticity. These neurons progressively become selective to intermediate complexity visual features appropriate for object categorization. Misra *et al* [117] presented an approach for learning a visual representation from the raw spatial-temporal signals in videos. The representation was learned without supervision from semantic labels. The method determined whether a sequence of frames from a video is in the correct temporal order.
- In 2017 Roy & Basu [118] proposed a WTA architecture employing neurons with nonlinear dendrites and an online unsupervised structural plasticity rule for training it. To aid hardware implementations, the network employed only binary synapses. The proposed learning rule was inspired by spike-timing-dependent plasticity but differed for each dendrite based on its activation level. It trained the WTA network through formation and elimination of connections between inputs and synapses. To demonstrate the performance of the proposed network and learning rule, the method

used was solving solve two-class, four-class, and six-class classification of random Poisson spike time inputs. The results indicated that by proper tuning of the inhibitory time constant of the WTA, a tradeoff between specificity and sensitivity of the network can be achieved.

1.3 Motivation

The works performed in the past as part of the complete global investigation trajectory are devoted to the application of artificial intelligence techniques to different real world problems. As can be observed in section 13.3 several areas have been covered as studies related to defense, automatic image processing, public health, financial decision making and environmental care.

This thesis main motivation is to apply the Metaplasticity principles and aspects to different neural networks structures and check the results and performance of this new technique with different real problems. In order to provide a wide study of the application field of this idea it has been considered convenient to work on different paradigms of the neural networks structure and in each case try to improve the results and solve the main problems associated to each one of them. It will be necessary to study the Metaplasticity characteristics and the way of modifying the classic neural network models to include these characteristics. In addition the application needs to be tested against the classic approach by using a set of procedures and methodologies selected for each specific experiment included in the thesis.

Taking into account the supervised learning method, in concrete to the BPA learning algorithm the idea is to solve the slow convergence problems inherent to the algorithm as well as to improve the classification accuracy results. Using as basis the investigations presented in the state of the art analysis it is considered that two main way of working: in one side heuristics techniques based on the study of the algorithm itself and the results of making variations to the basic elements on method (learning rate, structure, momentum value...); and in the other side some of the method proposed variations on the nature of the algorithm considering improvements that affects the mathematical basis of the basic learning method.

Considering the unsupervised learning methods it can be seen that there is a tendency to try to replicate small pieces of the biological human brain. Each one of those parts are usually devoted to one specific area of the knowledge (image processing, patterns classifiers,

adaptation to new elements in a known data set...). In some cases these unsupervised learning method applied to self-organization maps or auto-adaptive networks are combined with other techniques including mainly fuzzy logic or reinforcement of the results using supervised algorithms.

This thesis applies the metaplasticity theory to both types of networks in order to improve the performance and solve the most typical issues that appear in the execution of these training system. For the AMMLP the idea is to reduce the computation time and the number of iterations required during the training in order to improve or at least maintain the performance results with a high accuracy value in the classification. For the WTA training we will propose a scheme based on the koniocortex layer of the human brain using the properties of the biological brain in order to help the system obtain the relevant linear independent information of the data set to reduce the number of iteration of the network with higher accuracy than the obtained whit other unsupervised methods.

The second motivation of the thesis is the application of the proposed algorithm variants to real world problems. In this cases several classification or prediction problems are considered for each type of network depending on the concrete characteristics of the data set. Doing so we will use different standard data sets in order to compare the performance results, the convergence speed and the accuracy of the proposed algorithm:

- Radar pattern classification: In this case the multilayer perceptron structure is used as a classifier of radar input pattern trying to distinguish between the ones corresponding to a signal from the ones that only contain noise. We have selected this one as the first experiment using a AMMLP structure because the high amount of data available can be use to work in the integration of the Metaplasticity theory inside the neural networks structures making possible to tune parameters and get very representative statistical results [13] [119].
- Breast cancer pattern classification: The correct pattern classification is tested using the Wisconsin Breast Cancer Dataset (WBCD) [120] as the input data for two types of neural networks in order to distinguish between benign and malignant patterns: in one case we use a AMMLP model [15], [121] and also it has been used for studying the possibilities and biological plausibility of the KLN model [122].
- Credit scoring classification: Inside the Business Intelligence (BI) area the credit scoring techniques is used as the application area for the Metaplasticity theory. The

input data used as patterns to test the classifying skills of our model is the Australian Credit Approval Dataset (ACAD) [123]. We applied a AMMLP structure to distinguish between the situations in which credit should be granted or denied [124]. This data set has also been used as an input to a KLN to check the possible application of this structure to other real data sets.

- Atmospheric contamination prediction: This experiment consists in the application of an AMMLP structure to the pollutant concentrations prediction in cities. The city selected is Salamanca, Mexico, that has been ranked as one of the most polluted cities in the world. In this case the system tries to predict the values of the concentration of pollutant over 24 hours using a AMMLP network model [125].

1.4 Justification

The main justification of the thesis is the research in one of the most important and interesting areas of science: the human brain and its biological properties. Neuroscience is a relatively new field, but the more research that is done, the more it appears that much of human behavior and mental processes are intimately intertwined with activity in the brain. Understanding the brain is important because its effects permeate all human behavior.

The brain is the most complex object in the known universe. Some 100 billion neurons release hundreds of neurotransmitters and peptides in a dynamic spanning timescales from the microsecond to the lifetime. In addition to behavior, the study of brain data provide information about the mind that cannot be gleaned from even the most careful studies of psychologist. In short, brain data provide a physical grounding that constrains the myriad otherwise-plausible models of cognition. And one of the most powerful ways to study the brain behavior and properties is using simulated structures where theoretical paradigms can be applied in a controlled environment: the ANNs.

As further justification we can consider the application of these methods and techniques to areas of special relevance where the information obtained can be used as a real improvement in the performance of decision making for different areas as important as defense, health, banking or environmental care.

- Radar pattern classification: Radar is surveillance system used in all the countries of the world in different that have been employed in different areas as on ground, in air,

on sea, and in space for detecting and locating objects. Nowadays the radar system have been diversified in order to include this technology as part of tasks related to air and terrestrial traffic control, radar astronomy, air-defense systems, antimissile systems, marine radars to locate landmarks and other ships, aircraft anticollision systems, ocean surveillance systems, outer space surveillance and rendezvous systems, meteorological precipitation monitoring, altimetry and flight control systems, guided missile target locating systems, ground-penetrating radar for geological observations, and range-controlled radar for public health surveillance.

The modern radar systems are closely related to digital signal processing and machine learning. In this way processing the radar signals provides useful information even from very high noise levels situation, that can be used to improve the analysis results to be used in decision making automated systems.

- **Breast cancer pattern classification:** Breast cancer is the leading type of cancer in women, accounting for 25% of all cases. The consequences of developing a breast cancer vary depending on the cancer type, extent of disease, and person's age. Survival rates in the developed world are increasing and are highly depending on the stage of the development when the cancer is detected.

Breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, or a red scaly patch of skin. In those with distant spread of the disease, there may be bone pain, swollen lymph nodes, shortness of breath, or yellow skin. Breast cancer is sometimes found after symptoms appear, but many women with breast cancer have no symptoms. This is why regular breast cancer screening is so important. Different tests can be used to look for and diagnose breast cancer as for example mammograms. In this way an automatic detection aiding system can increase significantly the possibilities of an early detection.

- **Credit scoring classification:** A credit score is a numerical method based on a level analysis of a person's credit files, to represent the credit worthiness of an individual or institution. Normally the information used as inputs for a credit score process is primarily coming from credit report information sourced from credit bureaus.

The financial system agents, such as banks and credit card companies, use credit scores results to evaluate the potential risk posed by lending money to consumers and to mitigate losses due to bad debt. In this way they are able to qualify the potential

receivers of a loan, at what interest rate, and what credit limits. The use of credit or identity scoring prior to authorizing access or granting credit is an implementation of a trusted system. This system is not limited to banks and nowadays have been adapted to other type of business as mobile phone companies, insurance companies, landlords, and government departments. Machine learning can be used in parallel to data mining techniques in order to provide additional confidence to this credit approval process.

- Atmospheric contamination prediction: The impact of the atmospheric pollution is vital for climate and environment. The atmosphere is essential for life and the associated alterations have great repercussion in all the living beings and the complete planet. Changes in the atmosphere may change the climate, destroy the ozone layer or provoke acid rain.

Big industries led to a major economic development and rapid population growth mainly in the second half of the twentieth century. The impact in the air quality is important and significant efforts have been made to measure the concentrations of pollutants. The main pollution sources are locally based plants in the chemical and power generation sectors. Prediction of pollution levels is performed with the aim of alerting the population about possible health consequences and to advice the industry to take the adequate measures.

1.5 Objectives

1.5.1 General Objective

As derived from the thesis motivation and justification a set of objectives are defined to be covered during the complete study. The main objective of this doctoral thesis is the application of the Metaplasticity theory and the study of the influence of its related parameters to two different neural network learning paradigms: the BPA supervised one (in the form of a MultiLayer Perceptron Network) and the WTA unsupervised one (in the form of a Koniocortex Like Network). Doing so we will try to improve the results obtained with the unmodified structures applied to different real data sets. The comparison of the results will allow us to study if the modified structures are able to solve some of the presented problems inherent to the learning methods obtaining high accuracy levels in the classification/prediction results.

1.5.2 Specific Objectives

The following lists contains the specific objectives that were formulated at the beginning of the research and that are considered as the basis of the works presented here:

- Define the biological metaplasticity concept and its influence in the different neural network processes.
- Perform a study about the application of new bioinspired concepts derived from machine learning (Artificial Metaplasticity and Intrinsic Plasticity) to different neural network models for classification/prediction problems.
- Define and implement the neural model based on Artificial Metaplasticity applied to supervised learning method in a MLP model (AMMLP) as a previously needed step for the real data sets processing study.
- Define and implement the neural model based on Artificial Metaplasticity applied to unsupervised learning method in a WTA model (KLN) as a previously needed step for the real data sets processing study.
- Obtain a training method for neural network based on Shannon's information theory independent from the inputs patterns statistical distribution to be checked in different experimental situations.
- Apply the information present in the network output considered as a probability estimation to the modified BPA in different experimental situations.
- Perform an empirical study of the influence of the parameters' values used in the different learning methods in order to improve the efficiency and results of the proposed models.
- Apply the models to highly impacted problems that allows a reliable comparison of the results with other state-of-the-art methods and with the classic version of the algorithms.
- Dissemination of the results in highly prestigious scientist forums.
- Dissemination of the results through free software platforms.

1.6 Impact

- The results obtained on the thesis will provide information about the way of working of some learning mechanisms of the biological brain as well as the connection between the Shannon's information theory and the way the human being is capable to separate the essential information from the superfluous one.
- The application of AMP to classic learning algorithms can help solving the problems inherent to the methodology as low speed, problems for convergence or necessity of large amount of data to have accurate results.
- The application of the modified neural network to the radar pattern recognition will provide a method to reduce the false alarm probability with respect to the unmodified structures applicable in surveillance and recognition systems that will rise the efficiency of defense, meteorological, traffic or science investigation programs.
- The application of the modified neural networks to the breast cancer detection problem will improve the results obtained with other method to ease the detection of the illness in an early stage increasing dramatically the possibility of survival of the affected patient.
- The application of the modified neural networks to credit scoring is a powerful tool to be used in the financial system to reduce the cost of credit analysis, enable faster credit decisions, provide closer monitoring of existing accounts and reduce possible risks.
- The application of the modified neural networks to pollutant concentration prediction improves the reliability of the system compared with the classic neural networks and allows the authorities to take appropriate decisions concerning the activation of environmental contingencies to minimize the risks for the population health.

1.7 Thesis Structure

In this section a summary of the content of the different chapters can be found. The first 7 chapters have an informative and theoretical focus, meanwhile the rest of the chapters present the practical application of the theory exposed:

- Chapter 1: Contains a short introduction about the subject of the thesis, including the state of the art study, motivation, justification, objectives and impact of the study performed

- Chapter 2: Includes an introduction to the biological and artificial neural networks. Structures and elements are presented and an explanation of the classification of the different networks types, learning methods and architecture paradigms is included.
- Chapter 3: In this chapter there is a detailed description of one of the neural network models used in the experiments: the perceptron. The chapter includes the model characteristics and limitations and also the details of the classic BPA algorithm used for this kind of networks.
- Chapter 4: The chapter is devoted to the description of the self competitive network topology used for unsupervised learning paradigm and the basis of the most usual training methods applied to this kind of networks.
- Chapter 5: Presents the biological property called metaplasticity including the mechanism, functions and concepts associated to it. As additional information some other biological properties that are related to metaplasticity are explained, as plasticity and intrinsic plasticity. In this way we will have the theoretical basis to understand the application of these properties to the artificial neural networks models. In addition AMP concept is presented jointly with its relation with the Shannon's information theory
- Chapter 6: Presents the application of the Artificial Metaplasticity concept to the perceptron topology.
- Chapter 7: Presents the application of the Artificial Metaplasticity concept to the unsupervised learning paradigm in the form of the Koniocortex-Like Network topology.
- Chapters 8 - 11: The experiments related to the application of Artificial Metaplasticity to different study areas are presented, including material, methodology and a short discussion of the results obtained for each specific case.
- Chapter 12: Global conclusions of the study are presented and discussed.
- Chapter 13: The original contributions of the thesis are presented as well as a detailed list with possible future investigation lines derived from the results obtained in this work. The chapter also includes the detailed list of the publications derived from the work performed in this thesis.

Chapter 2

Neural Networks

2.1 Introduction and Motivation of the Neural Networks

The human brain is an information processing unit with a set of very notorious characteristics: it is capable of processing at great speed big amount of data coming from the senses, combining this information, comparing the new data with the one that it is stored in the memory and finally providing adequate response to even new situations. Among all the brain capabilities the most impressive one is the capacity to learn how to represent and process the information needed for developing the rest of the human activities without specific instructions for doing it so.

There are still lots of uncertainties about the way the brain learns how to process the information obtained, but the researchers have developed several models that try to mimic those skills. These models are called ANNs or connectionism computational models. The elaboration of these models include different areas to be covered: first it is needed to deduce the essential characteristics of the neurons and its connections, and after that to solve the problem of the implementation of the model in a computer based system in order to be able to simulate the behavior of the real brain.

It is interesting to describe the connectionism models using an analogy with the nervous system. Based on this and considering the history of the ANNs we can study the computational implementation of the models, the technical applications derived and the technologies developed in order to get productive use of their capabilities.

The brain structure is not a single monolithic processor, but it is conformed by billions of simple processors called neurons. Each one of these neurons perform a kind of very simple calculation in an imprecise and slow way. The science tries to emulate this structure and connections using the so called Artificial Intelligence (AI). This AI science is fed by the neuroscience that is progressing in a way that allows us to better understand the physical and logical structure of the human brain. This fact together with the technology advances allow representing more complex structures and speed up parallel calculations. This situation is leading to increase the investigation effort in the field of computational bioinspired systems.

2.2 The Biological Neuron

The human brain is continuously receiving information from a lot of sources simultaneously, and it is capable to process all these stimulus in real time and provide an adequate response. The ANNs try to emulate the behavior of the human brain. The known systems are not able to reach such complexity, but the functions included are inspired in the biological model considering the processing of information coming from the environment in real time, making comparisons and combinations to get adequate responses to even new situations. The blocks for building both kind of structures (the biological and the artificial ones) are extremely simple elements highly interconnected. These units present a great adaptation capability and are able to manage fuzzy information and parallel processing.

The human brain is composed by a great number of neurons (approximately 10^{11}) highly interconnected creating what it is called a neural network. Each one of the basic nervous system cells including its prolongations is called neuron. All these cells are formed by the same basic parts independently of its size and shape, these parts are the body (also called soma), the axon and the dendrites. These cells are excitable units, specialized in the reception of stimulus and conduction of the nervous impulse. The contact method between a neuron and other (or an auto connection with the same neuron itself in some cases) is called synapses and it is determined by the complexity of a chemical process that stabilizes the function of the neural network. A simple scheme of the connection between two neurons can be found on figure 2.1

The electrical impulse is called action potential. This potential travels along the axons and it is the result of an electrical discharge generated due to the flow of sodium and potassium ions (with positive charge) through the cellular membrane, from the extracellular fluid to the

Fig. 2.1 Simple scheme of the synapses union between two neurons

internal cytoplasm. The electrical transmission goes from the center body of the cell, along the axon to the different synaptic unions.

These action potentials are low frequency, slow conducted that cannot jump between neurons, so the real communication between cells is made through the liberation of chemical neurotransmitters in the synapses process. An example of this chemical communication between neurons can be found on figure 2.2

Fig. 2.2 Neurotransmitters liberation scheme

The excitatory inputs that arrive to the neuron through the dendrites reduce the potential difference between both sides of the cellular membrane. In addition the depolarization process in the axon modifies the permeability of the membrane, due to this there is a strong flux of positive sodium ions that enters in the cell body, contributing even more to the depolarization. At a concrete moment the potential reach a critical value called threshold that produces a positive feedback that forces the cellular membrane to invert its polarity. The permeability mechanism is in a refractory state during some milliseconds after each exposition to this ionic activity.

When the membrane is near to the threshold that provokes the action potential generation due to the charge displacement there is a excitement effect, and when the membrane is stabilized there is an inhibitory effect. Each synapses produces only a small effect and to determine the real intensity of the impulse response each neuron has to integrate continuously over 1000 synaptic signals that are combined in the cellular body. The complete process can be found on figure 2.3.

Fig. 2.3 Complete synapses process

As a summary of the process a neuron receives through the axon or dendrites several little electrical impulses and the body integrates all the incoming signals. The result of the integration can be an excitation that it is transmitted to other neurons or an inhibition that

mitigates the strength of the global impulse or even eliminates it. All the neural processes are determined by this basic mechanism. When the human body receives an external stimulus through the external neurons, these ones, depending on the intensity and location of the signal, start a long chain of connections that finally reach the brain to provoke an adequate response.

Some of the neural structures are determined at the birth moment, but other are developed thanks to learning processes in which new neural connections are established and other are completely eliminated. The neural structures are changing the whole life, these changes consist in the reinforcement or weaken of the different synaptic unions. For example it is believed that new memories can be created by a modification of the intensity values between synaptic connections. Hebb [126] suggested that the associative learning may be produced by a simple cellular mechanism and proposed that the associations can be generated by a coincident neural activity: when an axon of a neuron A excites neuron B and participates in the global activation, then there is any kind of development process of metabolic change in the neurons so the efficiency of A as exciting partner of B is intensified. According to this Hebbian learning rule, the coincidence of the activity of the pre-synaptic neurons (that provide the input impulse) with the post-synaptic ones (that receive the impulse) is really important to reinforce the connection between this two neurons. This is called the pre-post associative process that is showed in figure 2.4.

Fig. 2.4 Hebbian pre-post associative mechanism

2.3 Artificial Neural Networks

2.3.1 Brief History of ANNs

The origin of the connectionism models can be situated in the formal definition of the artificial neuron provided by McCulloch & Pitts in 1943 [127] as a binary device with several inputs and outputs. In 1949 Hebb [126] introduced two fundamental ideas that have had a decisive influence in the neural network development: the idea of that perception is represented in the brain as a set of neurons that are active at the same time, and the idea of that memory is localized in the connections between the neurons (synapses).

The hypotheses provided by Hebb are based on physiological investigations, and present in an intuitive way the method used by neurons to memorize information. This method is synthetically represented as the Hebb learning rule. The main characteristic of this rule is that the connection between two neurons is reinforced if both neurons are activated at the same time.

In 1956 in Dartmouth it took place the first international conference about AI. There were discussions about the potential use of computers to simulate all the aspects of learning and any other characteristic of the human intelligence. A first simulation of a neural network model was also presented.

In 1958 Roseblatt published the results of an ambitious investigation project: the development of the perceptron. It was an identifier of binary optical pattern with a binary output [128]. The capabilities of the perceptron lead to the development of the delta learning rule that allows using inputs and outputs inside a continuous range.

In 1960 Widrow published a theory about the neural adaptation and presented two models inspired in this theory: the Adaline (Adaptive Linear Neuron) and the Madaline (Multiple Adaline) [129]. Those models were adapted to several applications and allowed using a neural network in a real world problem for first time: the application of adaptive filters to eliminate the echoes in the telephone lines.

In 1966 Minsky & Papert published a critic of the perceptron model [130] revealing serious limitations of the structure basis, for example the impossibility to represent a XOR

function due to its linear nature limitation. This work provoked that the investigations in AI dramatically decreased. Nevertheless in the 70's new ideas appear:

- Grossberg used the brain physiological principles to develop an important mathematical theory that lead to the ART model [131].
- Anderson studied and developed several associative memory models. One of the most important is the linear auto-association model called brain-state-in-a-box [132].
- Kohonen *et al* continued the Anderson work and developed competitive models based on lateral inhibition principle [133]. The main contribution was a procedure that allowed that physically adjacent units learn to represent similar input patterns. These kind of structures are known as Kohonen networks.

In the following decades the interest in AI raised again due to the work of the Parallel Distributed Processing (PDP) group and the contributions made by Hopfield. On one hand Rumelhart *et al* created the PDP group that provide the most influential AI manuals since the Minsky & Papert critic. The most important of the text provided is the one dedicated to the BPA method that proposed a solution for the problems identified in the mentioned critic [1].

In other line Hopfield elaborated a network model consistent in processing units interconnected that tried to reach minimum energy. This work is based in the theory presented by Grossberg. The Hopfield model is very illustrative considering the brain processes for storing and recovering the memory information [134].

Nowadays the brain models are developed from a multidisciplinary formulation with different points of view:

- Computational: The idea is to develop efficient computational models trying to maximize simplicity and processing power. Inside this area we can highlight the inductive learning and the shape recognition methods.
- Cognitive: It is focused in the cognitive capabilities of these models without considering the need of a biological plausibility. These models are powerful and simple considering the knowledge representation.

- **Biocognitive:** Similar to the previous one but considering that the models should be biological representative. The models try to represent the general characteristics of those biological mechanisms to apply them to typical cognitive tasks getting advantage of the inherent qualities of the brain as an information processor.
- **Physiologic:** The basis is to understand the natural mechanism implied in the real cognitive processes: perception, reasoning, memory...

2.3.2 Main Characteristics of Neural Networks

Darpa defines a neural network as a system composed of several simple processing elements that operate in parallel for a function which is determined by the network structure and the weights of the connections. Processing of information is performed in each one of the nodes or computational elements [135].

For Haykin a neural network is a massive distributed parallel processor that has a natural ease for storing the knowledge obtained from experience to make it useful in future situations [2]. There are two main aspects of resemblance between the ANNs and the human brain:

1. Knowledge is obtained by the network through a learning process.
2. Inter neuronal connections, known as synaptic weights, are used to store this knowledge.

Kohonen define the ANNs as a network composed by simple elements interconnected in parallel with a hierarchical organization. The elements try to have an interaction with the objects of the real world the same way as the biological nervous system does [136].

Summarizing an artificial neural network can be considered as an information processing system that presents some behavior characteristics in common with the biological neural networks. These basic aspects are:

- Information processing is performed by many simple elements called neurons.
- Signals are distributed between the neurons through connection links.
- Each connection link has a weight associated, in a typical neuron this weight multiplies the intensity of the received signal.

Table 2.1 Differences between a traditional computer and a biological neural system

Characteristic	Sequential Computer	Biological Neural System
Processing Unit	Complex, fast and unique	Simple, slow and many
Memory	Separated from the processor, localized	Integrated in the processor, distributed, accessible by content
Data Processing Method	Centralized, sequential, stored programs (not designed to modify its own program)	Distributed, parallel, self-learning (designed for self modifications)
Reliability	Vulnerable	Robust
Specialization	Symbolic and numeric manipulation	Perception problems
Operational Environment	Well defined and restricted	Fuzzy defined without clear restrictions

- Each neuron applies an activation function (normally not linear) to the inputs received to determine the output signal.

Other important issue to be studied is the difference between the way of working of a conventional computer program and an artificial neural networks. The conventional computer programs (known as expert system inside the AI) are similar to the ANN in the way that both try to represent knowledge, but there are radical differences in the way they try to do it. Meanwhile the expert system are closer to deductive reasoning (obtain behavior rules) the ANNs are closer to inductive reasoning (learning using practical examples).

In order to make a brief summary of the wrapped up conditions of the neural networks, the table 2.1 includes the fundamental differences between a traditional computer and a biological neural system.

So it can be considered that the ANNs are a good solution to solve those problems that people are able to deal with but computer don't. These kind of problems include pattern recognition and prediction problems (that require the understanding of a certain tendency on the input data).

With all this considerations in mind a very first division of the neural networks can be presented:

- **Pattern couplers or hetero-associative memories:** This kind of network is formed by two or more layers and has the objective of creating an association, generally using a supervised learning method, between pattern pairs called input pattern and output pattern. The idea is that when presenting at the input a pattern of a concrete class the network will recover at the output the pattern that were associated during the learning process. Due to the generalization capability of the neural network the system will be able to use the knowledge acquired during the training to provide the correctly associated output pattern to a input pattern that has never been presented to the network before.
- **Competitive networks or auto-organizing maps:** This kind of networks can have one or more layers with the common characteristic of generating a competition between the single units so at the end of the process only one of the neurons is activated and the others remain inactive. The usual training method for these kind of networks is an unsupervised learning method, presenting input patterns to the network and selecting the unit which associated weights are more similar to the input. In this way the winning connections are reinforced meanwhile the losing ones are weakened.

Natural competition between units can be obtained simulating a neuro-physiologic characteristic of the brain cortex called lateral inhibition. This is done considering the existence of inhibitory connections inside a layer and exciting connections between layers. Doing so the presentation of an input pattern will activate a neuron and at the same time it will inhibit the others in the same layer. Finally each neuron only reacts with a concrete type of input.

- **Demand satisfaction or probabilistic adaptation models:** This kind of network are asynchronous models, what means that each unit is updated or not during a processing cycle depending on a probability value.

The objective of these networks is to obtain almost optimal solutions to problems that need to take into account a great number of simultaneous demands. The network starts in a dynamic and unstable state and tries to get to a stable final state through a relaxation process in which the majority of demands are satisfied at the same time.

2.3.3 The Artificial Neuron

The first step is to try to simulate the behavior of the simple basic information processing unit. These units are called artificial neurons. Among the different types of artificial neurons that

have been defined the most common one is the McCulloch-Pitts neuron that can be observed in figure 2.5.

Fig. 2.5 Representation of a McCulloch-Pitts artificial neuron

An artificial neuron is an elemental processor in the sense that is capable of processing an input vector $\bar{x}(x_1, x_2, \dots, x_N)$, ($\bar{x} \in R^n$) generating an unique output or response. The principal elements of this artificial neuron are:

- The inputs provided by other neurons. Here the electric signals are converted into numerical values.
- The synaptic weights $w_{ij} \in R$. In a similar way than a biological neuron, there are synaptic connections between the dendrite of one cell and the axon of other cell. In the

artificial neurons for each input coming from other neuron there is an assignation of an importance factor in the form of a weight value. This weight is a number that is modified during the network training, and so this value stores the information in order to use the network for its specific objective.

- A propagation rule. With the inputs and the weight values the neuron performs some operation in order to obtain the post-synaptic potential value. One of the most common operation is adding the inputs considering the importance of each one (the weight associated to each input). It is called weighted sum:

$$h_i = \sum_j w_{ij}x_j \quad (2.1)$$

where x_j is the j th input, w_{ij} the weight assigned to the neurons connection between j and i , and h_i is the output of neuron i . Other propagation rule commonly used is the euclidean distance.

- An activation function. The value obtained from the propagation rule is filtered using a function called activation function that provides the final output of the neuron. Depending on the objective of the network it is possible to chose between different functions [2], [137]. Table 2.2 includes some of the most typical activation functions.

Many times it is needed to use an activation function different than the identity one in order to have a limited value output, as the real biological neurons have an output which is limited in amplitude. In addition each neuron has a number associated called bias that can be seen as the threshold value from which the post-synaptic potential of the neuron provokes a significant output. This term $\theta \in R$ is included in the weighted sum that is inserted in the activation function as if it was an additional input to the neuron.

$$y = f\left(\sum_{i=0}^{N-1} w_{ij}x_i - \theta\right) = f\left(\sum_{i=0}^N w_{ij}x_i\right) \quad (2.2)$$

Where y is the output of the network, f is the activation function, N is the number of components of the input vector, w_{ij} is the weight corresponding to the i th input to neuron j , x_i is the value of the i th input and θ in the value of the bias.

Table 2.2 Activation functions

	Function	Range	Graphic
Identity	$y = x$	$[-\infty, +\infty]$
Step	$y = H(x)$	$[-1, +1]$
Piecewise Linear	$y = \begin{cases} -1 & \text{if } x < -l \\ x & \text{if } -l \leq x \leq l \\ +1 & \text{if } x > l \end{cases}$	$[-1, +1]$
Sigmoid	$y = \frac{1}{1+e^{-x}}$	$[-1, +1]$
Gaussian	$y = Ae^{-Bx^2}$	$[0, +1]$
Sinusoidal	$y = A \sin(\omega x + \varphi)$	$[-1, +1]$

2.3.4 Basis of the Learning Methods for ANNs

In AI the ANNs are considered as a paradigm of learning and automated processing inspired in the biological nervous system of the animals. The most important property of the nervous system is the capability of extracting information of the environment and improve its performance. This can be defined as "a simulation process in which the free parameters of a neural network have been adapted to the environment injected in the network" [138]. With the help of a learning procedure an ANN is capable to extract and store the information included in the input data. This information is stored in the connection weights and can be recovered for a future use.

In any neural network it is possible to distinguish three phases: the training, and optional test phase, and the application phase. For the training a learning rule shall be adopted. During the training the inputs will be presented to the networks and the weights will be updated, in order to try to minimize an error or energy function. A learning rule defines exactly how the network weights shall be updated (adjusted) in the different cycles (epochs) of the training process considering the environment and the inputs presented to the network.

The learning rules can be divided between supervised and unsupervised. For the supervised training method the system presents a set of example inputs to the network adding the information of the ideal target values that should be generated at the output of the network. Considering the target output and the real one generated by the network it is possible to compute an error signal. The objective of the training procedure is to modify the parameters of the network in a way so the real output of the network should be the closer possible to the ideal one, minimizing the error signal in that case. The concrete modification of the parameters is calculated based in the input signal and in the error signal [2], [139].

For the unsupervised method the input pattern doesn't contain any information about the desired output and there is no external supervision of the training process. In this cases the network is let free to follow some kind of self-organization rule. There are two main models for this kind of rules:

- Self-organizing learning: The parameters of the network are modified according to the learning rules without any kind of supervision, so it is the network itself the one that detects any possible tendency in the input data.

- Reinforcement learning: It can be considered a special case of unsupervised learning because of the use of a "critic" to control the training. The critic doesn't give the network the desired output, it is only limited to indicate if the real output is correct or not [140].

The literature has established five rules or basic norms for the ANN learning process. These methods are: Error-Correction Learning (ECL), Memory-based learning, Hebbian Learning (HL), Competitive learning and Boltzman Learning (BL) [2], [141]. A short explanation of these five paradigms is provided:

- Error-Correction Learning: the method tries to correct an estimation error. For a particular training pattern, the error is considered as the distance between the real output of the network and the ideal one. For example if we consider the supervised learning algorithm, the generalized delta rule modifies the weights of the network performing an update of these values for each iteration of the training. This update is proportional to the variation rate of the error considering the weights, in negative sense.

$$\Delta w_{ij} = \eta y_i (\hat{y}_j - y_j) = w_{ij}(t+1) - w_{ij}(t) \quad t = 0, 1, \dots, n \quad (2.3)$$

- Memory-based learning: the functions store all the past experiences or training patterns (x_i, d_i) , in an explicit way in a big size memory. This memory contains the input-output patterns (x_i, d_i) correctly classified. The classification of a new pattern is made through the recovery and analysis of an stored old pattern.
- Hebbian Learning: The Hebb learning rule was introduced to explain the learning process in the biological neurons, and indicates that a particular connection will be reinforced if the neurons of both extremes are active at the same time in a persistent way [126]. In mathematical terms this hypothesis can be written as:

$$dw_{ij}(n) = \eta y_i(n) x_i(n) \quad (2.4)$$

Where $dw_{ij}(n)$ is the change in the neuron connection strength between i and j , $y_i(n)$ is the output of neuron j and $x_i(n)$ is an input to the neuron. $\eta \in R$ is a learning factor and $n \in R$ specifies a concrete epoch of the learning process.

- **Competitive learning:** the neurons of the output layer perform a competition between them to get active, the WTA approach is an example of this learning type. Normally these methods use statistical properties of the input data [2], [140], [142].
- **Boltzman Learning:** It is a stochastic process based on statistical mechanics. It is a recurrent network where the neurons work as binary nodes that can be in two states: active or inactive. An energy function is applied to the machine to measure the energy contained in the network. A neuron is randomly selected and its state is changed during the learning process and then the energy is measured again. This process continues until an equilibrium state is obtained.

In general when programming the learning method for a neural network it is needed to find an equilibrium between two parameters: the complexity as the number of patterns that are presented at the input of the network to obtain a certain level of generalization, and the computational load required for this amount of work.

2.3.5 Architectural Models for ANNs

The architecture or topology of a neural network is referred to the physical disposition of the nodes inside the network. The nodes create layers that share a common input and provide an output to other set of common nodes [2], [137]. The nexus between nodes are simulated through the synaptic connections.

Information is propagated in one sense, from a pre-synaptic neuron (origin) to a post-synaptic one (destiny). It is possible to have a situation where the output of a neuron is considered as an input to its own node in a sustained feedback process. Each synaptic connection is characterized by a weight value (so each connection has a real number associated to it that shows the importance of the connection compared to the rest), that is called connection strength. If this weight is positive then the connection is an exciting one, and if it is negative is an inhibitory one. These connections are the ways the information is transmitted inside the network [2], [3], [137].

In general the neurons are grouped in layers, so a complete ANN is formed by several neuron layers. Depending on its specific function each layer receives an specific designation:

- **Input Layer (IL):** A set of non-processing nodes that receive the input data and feed the network.

- Hidden Layer (HL): These layers introduces degrees of freedom in the network function. The number of layers depends on the network type considered.
- Output Layer (OL): This layer provides the final output of the network processing.

Considering the flow of the information inside the network two types of networks can be considered:

- Feed-forward networks: The information is progressing in only one sense, moving forward from the inputs nodes to the output nodes through the hidden nodes (if existing). The inclusion of these hidden nodes allows to increase the complexity of the calculations performed by the network. There are no loops inside the network. These networks are usually used as classifiers. An example can be seen on figure 2.6.
- Recurrent networks: The information can be transmitted in both senses, allowing connections between nodes of the higher layers with nodes of the lower layers. These network are also called feedback networks. An example can be seen on figure 2.7.

Fig. 2.6 Example of the architecture of a feed-forward network

Globally there can be established three main architecture types as the most usual ones: the MLP, The SOM and the Radial Basis Function Networks (RBFN) [2], [137]. A short description can be found on the following paragraphs (the MLP and the SOM models will be covered in further detail in chapters 3 and 4 of this thesis).

Fig. 2.7 Example of the architecture of a recurrent network

- **MultiLayer Perceptron:** It is one of the most common architectures. It is based in a simple model called Single Layer Perceptron (SLP), including one or more hidden layers in the structure. It is a feed-forward network which hidden layer neurons use as propagation rule the weighted sum of all the inputs considering the synaptic weights w_{ij} . Over this sum a sigmoidal activation function is applied so the output levels are limited.

The most usual learning method applied to this kind of network is the one based on the backpropagation of the error value. The idea is to calculate an error value at the output of the network and propagate it from the output layer to the input one, passing through the hidden nodes. During this process the weights of the connections are adjusted to reduce the error value.

The structure of the network can be found on figure 2.8.

- **Self-Organizing Maps:** In this kind of networks the supervised learning methods are normally not applied. The network doesn't have information of the ideal output patterns to check its response. The idea is to be able to get information about tendencies in the input data set and modify its weights to be able to recognize and classify this tendency regular information or classes.

Fig. 2.8 Architecture of the MultiLayer Perceptron

The structure is formed by an input layer to get the information inside the network, hidden layers to be able to process this information and an output layer that is a matrix of neurons. Depending on the class of the input data one of this inputs is usually activated meanwhile the others are inhibited. The idea is to classify the input patterns inside groups with similar characteristics, so an input belonging to this group will always activate the same output. The unit activated by each group is not known *a priori*, it is observed after completing the network training. The networks performs a mapping of an input space to an output space with a certain topological order. Sometimes the input data activates more than one neuron at the output, in this case a demographic code is being used.

An example can be found on figure 2.9.

- **Radial Basis Function Network:** This kind of network is characterized by using an hybrid learning method. The architecture contains three layers: one input layer, on hidden layer and one output layer. The neurons of the hidden layer instead of calculating the result of the sigmoidal activation function over the weighted sum of the inputs, calculate the euclidean distance between the synaptic weight vectors and the input, and apply a Gaussian radial function over this distance value. For the hidden layer learning paradigm several methods can be applied, being one of the most usual ones the use of k-means for unsupervised clustering.

Fig. 2.9 Architecture of the Self-Organizing Map

An example can be found on figure 2.10.

Fig. 2.10 Architecture of the Radial Basis Function Network

As a summary a taxonomical classification of the artificial neural network can be found in [137]. This classification is shown on figure 2.11.

Fig. 2.11 Basic taxonomy of the ANNs

Chapter 3

Supervised Learning: The Perceptron

3.1 Introduction

The first neural network introduced was developed in 1943 by McCulloch & Pitts [127]. This article presented the idea that the neuron activities and the existing relations between them can be studied using the propositional logic. The authors believed that the neurons have a bi-stable behavior (generating an all-nothing output) similar to the behavior of the electrical commutation units. In addition they presented the idea of creating computing machines with an architecture similar to the nervous system networks. These ideas anticipated connectionism concepts considering that the laws that govern the brain functionality are more related to the information than to the material characteristics: the learning process is considered more important than inborn characteristics. They considered that the initial state of the brain is a set of random networks, and that the stimulus received are the ones that configure the connections and consequently determine the behavior of the system. The global idea was to consider the brain as a computational organism.

The authors called artificial neurons to the elements that form the network, and provided an explanation of how those neurons could make logical calculations inside the network. The initial units were binary commutation units that get the input signals, and make the sum of the values multiplied by the corresponding weights. This result was compared with a determined threshold and if it was higher then the output of the neuron was set to one (1) and the neuron is considered active, and if the result was lower the output of the neuron was set to zero (0) and the neuron is considered inactive.

This model was evolved considering different propagation rules and activation functions more complex than a threshold. The election of this configuration was dependent on the environment of the network: range of input and outputs, type of learning paradigm, error calculation function and the concrete functionality desired for the network.

3.2 Single Layer Perceptron

In 1958 Rosenblatt [128] provided a critic to the work performed by McCulloch & Pitts. Instead of basing the analysis in symbolic logic applied to quite idealized structural networks, he defended the use of probabilistic methods considering that the connectivity existing in the neural networks was greatly influenced by random processes. Rosenblatt thought that the best tool for analyzing the neural networks was the probability theory and this lead to the proposition of a statistical separability theory used to study the properties of this interconnection networks that include a small random component. Finally Rosenblatt proposed a McCulloch-Pitts network capable of modifying its connection weights if the response was not correct. This kind of network was called perceptron and the author demonstrated that it was possible to train the system to classify some kind of patterns, so being able to recognize simple forms.

The perceptron is constituted by a set of sensors that receive the input patterns to be classified and an output layer that provides the classification result: either a value 1 (active) or 0 (inactive). The scheme of this perceptron is included in figure 3.1.

Fig. 3.1 Single Layer Perceptron scheme

Considering the way of working of the SLP, let's suppose a function f from R^n to $\{-1, 1\}$ that applies an input pattern $x = (x_1, x_2, \dots, x_n)^T$ in the desired output space $z \in \{-1, 1\}$, so $f(x) = z$. The information that we have available about this function comes from p training pairs in the form:

$$\{x_1, z_1\}, \{x_2, z_2\}, \dots, \{x_p, z_p\} \quad (3.1)$$

Where $x_i \in R^n$ and $f(x_i) = z_i \in \{-1, 1\}$, $i = 1, 2, \dots, p$. This function performs a partition of the input pattern space R^n , considering two possible output: 1 and -1 . So the function is in fact a classifier of the inputs in two different classes. If we try to produce a simple device capable of performing this function using as the starting point a well known set of relations between patterns, we can use a bipolar processing unit defined by the following mathematical expression:

$$f(x_1, x_2, \dots, x_n) = \begin{cases} 1 \rightarrow w_1x_1 + \dots + w_nx_n \geq \theta \\ -1 \rightarrow w_1x_1 + \dots + w_nx_n < \theta \end{cases} \quad (3.2)$$

Where the parameters $w_1, w_2, \dots, w_n$ are called synaptic weights and are applied to the input values $x_1, x_2, \dots, x_n$ that are the function arguments. The weighted sum is called synaptic potential and the parameter θ is the threshold.

Defining a binary processing unit as a mathematical function with domain in the n -dimensional set $\{0, 1\}^n$ and range in the set defined by $\{0, 1\}$ as:

$$y = f(x_1, x_2, \dots, x_n) = \begin{cases} 1 \rightarrow w_1x_1 + \dots + w_nx_n \geq \theta \\ 0 \rightarrow w_1x_1 + \dots + w_nx_n < \theta \end{cases} \quad (3.3)$$

When the output of this processing unit is equal to 1 it is said that the unit is active and presents the 1 state, meanwhile if the output is equal to 0 the unit is inactive and presents the 0 state. So the SLP is equivalent to a linear discrimination function, where the weights determine the slope of the straight line and the threshold determines the position respect to the coordinates origin.

For example it can be seen that the SLP is able to provide a solution to problems like the AND function (figure 3.2) and the OR function (figure 3.3).

In general if one layer contains N artificial neurons with M common inputs, the output will consist in a vector with N components, each one corresponding to one of the neurons of

Fig. 3.2 AND logic function

Fig. 3.3 OR logic function

the network. In this way it will be possible to classify the input into $2N$ regions. These regions will be separated in a multidimensional space by a hyperplane [139]. The figure 3.4 shows an example of this architecture.

Fig. 3.4 Single Layer Perceptron with N neurons scheme

3.2.1 Learning Method for the Single Layer Perceptron

In order to determine the values for the synaptic weights and for the threshold it is needed to follow a process that starts with initial random values and continues with the iterative adaptation of these values when the unit output is different from the desired one. The rule used to modify those values is known as the learning rule for the SLP and consists in the presentation of a training pattern to the network and the calculation of the new values for the weights and thresholds for each node, adding a correction term. The mathematical expression for the values modification is the following one:

$$\begin{cases} w_i(t+1) = w_i(t) + \Delta w_i(t) \\ \theta(t+1) = \theta(t) + \Delta \theta(t) \end{cases} \quad (3.4)$$

In the initial configuration the connection weights get random values, so when an stimulus is presented to the network the output generated is random too, and normally those outputs don't match the desired ones. It is considered that the networks has been trained when the weights have been modified in a way that the obtained output matches the desired one.

The procedure for the training is simple: an input is presented to the network and it is transmitted through the network until reaching the output layer. In this process some units may be activated, if the units activated are the correct ones then no adjustment of the weights take place. If the unit should be active and it is not, then the connection weights are increased. If the unit should be inactive and it is not, then the connection weights are decreased. This procedure is repeated for all the available patterns.

In a numerical view this procedure can be summarized as the convergence of a weight matrix $W(t)$, that is modified in each iteration, to a final value W that for an input vector X generates a correct output vector Y . From [143] the following elements of the algorithm are obtained:

- N : Number of artificial neurons of the SLP.
- M : Number of input data.
- i : Variable used for the input numbering.
- j : Variable used for the neuron numbering.
- T : Time iteration when the weight matrix has reached the desired value.
- t : Variable used for time numbering, $t = 0, 1, 2, \dots, T$.
- $X(t)$: Perceptron input vector for the iteration t .

$$X(t) = (x_0(t), x_1(t), \dots, x_i(t), \dots, x_{M-1}(t))^t \quad (3.5)$$

- $Y(t)$: Desired output vector corresponding to $X(t)$.

$$Y(t) = (y_0(t), y_1(t), \dots, y_i(t), \dots, y_{N-1}(t))^t \quad (3.6)$$

- $W(t)$: Weight matrix corresponding to iteration t . The dimensions of the matrix are $M \times N$ and the components values are the ones modified through the training process until reaching the definitive matrix.

$$W(t) = \begin{pmatrix} w_{00}(t) & \dots & \dots & w_{0N-1}(t) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ w_{M-10}(t) & \dots & \dots & w_{M-1N-1}(t) \end{pmatrix} \quad (3.7)$$

- θ_j : Threshold corresponding to neuron j .
- $\hat{Y}(t)$: Obtained output vector corresponding to $X(t)$. The objective of the training is that this vector gets the most similar possible to the ideal one.

$$\hat{Y}(t) = (\hat{y}_0(t), \hat{y}_1(t), \dots, \hat{y}_i(t), \dots, \hat{y}_{N-1}(t))^t \quad (3.8)$$

- η : Network gain or learning rate coefficient. Is a positive number smaller than 1 and it is the driver of the network learning speed and the algorithm convergence capability.

Once that the elements have been defined lets study the different phases of the algorithm:

1. Random initialization of the values corresponding the the weights and the thresholds ($w_{ij}(0)$ and $\theta_j(0)$).
2. A pair of training patterns ($X(t)$ and $Y(t)$) are presented to the input of the perceptron. In this way we obtain the real output of the network $\hat{Y}(t)$ in the following way:

$$\begin{aligned} \hat{Y}(t) &= (\hat{y}_0(t), \hat{y}_1(t), \dots, \hat{y}_i(t), \dots, \hat{y}_{N-1}(t))^t \\ \hat{y}_j(t) &= f\left(\sum_{i=0}^{M-1} w_{ij}(t) x_i(t) - \theta_j\right) \\ f(x) &= \text{sign}(x) \end{aligned} \quad (3.9)$$

3. Using the obtained output we can calculate the weight matrix for the following iteration. The way to update the weights and the threshold is described in the following equation:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \eta (y_j(t) - \hat{y}_j(t)) x_i(t) \\ \theta_{ij}(t+1) &= \theta_{ij}(t) + \eta (y_j(t) - \hat{y}_j(t)) x_i(t) \end{aligned} \quad (3.10)$$

4. In the case that the real output is similar to the ideal one then there is no change in the weights and thresholds. In this case we can consider that the weight matrix has finally reach the final value W in iteration T .
5. In case the convergence is not reach yet, then we can make $t = t + 1$ and go back to step 2 with a new training pattern.

Convergence speed for the weights and the thresholds is determined by the learning rate η . If this value is near to 0 the changes of the values between iterations will be small and the training will be slow, but if the value is near to the 1 (the higher limit) then the changes will be higher and the process faster but it is possible that the training will be unstable and won't converge to the final values. The adequate value will depend on the concrete application and on the concrete training pairs presented to the network. The objective is to obtain a balanced value between stability and convergence speed.

3.2.2 Functional Problems of the Single Layer Perceptron

In 1969 Minsky & Papert [130] analyzed the capabilities and limitations of the SLP and demonstrated that there are several problems that this network (or any simple network similar to the SLP) is not able to solve.

The problem was related to the fact that the SLP is only able to distinguish between linear separable patterns. Taking into account that many problems don't respond to this quality then the authors conclude that the SLP couldn't be considered as appropriate classifiers. A data set is considered as linear separable when the space determine by all the inputs can be divided in two regions by an straight hyperline, so all the input of one category are situated at one side of the hyperplane and all the inputs of the other category are situated at the other side of the hyperplane.

The easiest way to understand the critic of Minsky & Papert is the XOR problem. This disjunctive separation can't be solved with a SLP because if we have a look to the figure 3.5 representing the XOR function it is not possible to separate both regions with one straight line.

SLP can generate only one separation line, but more lines can be added to the solution if we include hidden layers between the input and output layers. This lead to the generation of

Fig. 3.5 XOR logic function

the MultiLayer Perceptron to be studied in the following section.

3.3 MultiLayer Perceptron

Considering the limitations of the SLP the idea of including hidden layers to solve those problems was obtained as a combination of the works performed by Werbos [144], Parker [145] and Rumelhart [1]. The result was called MultiLayer Perceptron and its main characteristic was that this structure was able to solve non linear separable problems.

A multi layer network implies the existence of neurons included in intermediate levels between the input and the output layers, considering the internal connections between the nodes of the different layers. Inside this kind of networks we have previously identified a type of network called feedforward networks in which all the connections are made in the direction from the input to the output of the network.

The MLP is a feedforward network composed by one or more intermediate layers that allows the system to establish decision regions much more complex than the ones obtained with a SLP [2], [137], [146]. Depending on the used criteria we can consider the existence of a first neuron layer with a function consisting exclusively on the delivery of the input signals to the neurons in the next layer, without any kind of additional processing. In other way we can consider that the components of the input vector enter directly on the first layers that process the signals. The structure of a MLP can be found on figure 3.6

Fig. 3.6 MultiLayer Perceptron structure

In general it can be stated that a MLP with one hidden layer and a linear activation function in the output layer can perform an approximation arbitrarily close to any continuous

function if the number of hidden neurons is high enough.

In order to better understand the possibilities of the MLP structures to be used for non linear separable problems we will use matrix notation. To simplify the calculation we can suppose that we are working with artificial neuron with a threshold value equal to 0 and with linear activation functions. The output of the i th layer of the network can be expressed as:

$$Y^{(i)} = W^{(i)t} Y^{(i-1)} \quad (3.11)$$

In particular for the first layer we have:

$$Y^{(1)} = W^{(1)t} X \quad (3.12)$$

In general if we express the output (for example for two layers) in function of the inputs we obtain:

$$Y^{(2)} = W^{(2)t} Y^{(1)} = W^{(2)t} (W^{(1)t} X) = (W^{(2)t} W^{(1)t}) X = W_{Tot}^t X \quad (3.13)$$

It can be seen that the same function can be performed by a SLP with an weight matrix W_{Tot}^t . So a MLP with linear activation functions is equivalent to a SLP with a linear activation function and a different weight matrix. In this way the classification problem for non linear separable regions is not solved with this kind of MLP.

The solution to this problem is to introduce non linearity characteristics in the activation function of the neurons of the MLP, having finally a differentiable and non linear activation function (as for example a sigmoidal function). As the basic perceptron can only establish two regions separated by a linear boundary if we introduce an additional layer it is possible to create a convex region in this space. This convex region is created as the intersection of the neurons of the second layer, each one of these replicating the behavior of a SLP [146]. To create this intersection the AND function is activated when all the neurons of the hidden layer are active and the output neuron uses an appropriate activation function. The result will be a region with a number of sides similar to the number of neurons in the hidden layer. Adding more layers we can make these regions as complex as needed by the dimension of the problem [139], this is shown in figure 3.7.

For the number of layers it is considered that 3 layers are enough to simulate any region with arbitrary precision with a finite number of discontinuities. Taking this analysis

Structure	Types of Decision Regions	Exclusive-OR Problem	Classes with Meshed regions	Most General Region Shapes
Single-Layer	Half Plane Bounded By Hyperplane
Two-Layer	Convex Open Or Closed Regions
Three-Layer	Arbitrary (Complexity Limited by No. of Nodes)

Fig. 3.7 Possible separable regions generated by a MultiLayer Perceptron

into account in the design of the network, it has to be considered how many neurons are needed for the hidden layer of a MLP structure. In general this number shall be high enough to be able to create a region complex enough to solve the concrete problem under study, but if the number is too big then the weight estimation could not be reliable for the input pattern available. Normally the number or nodes of the first layer shall be high enough to provide 3 or more sides for each convex area generated by the nodes in the second layer. In general we can consider at least 3 nodes in the first layer per node in the second layer [139].

Using perceptrons with non linearity properties different than a hard limit, the behavior of the network is more complex, with decision regions limited with lines softer than a straight line. Although this characteristic all the MLP networks can be trained using the BPA. Nevertheless for the MLP it hasn't been possible to demonstrate the convergence of this training algorithm, but it has been proved that a single hidden layer with sigmoidal activation function is enough to establish any kind of classification [147].

3.4 The BackPropagation Algorithm

The BPA is one of the more used algorithms for training the MLP networks. The base of the algorithm is the delta rule algorithm [148], that was extended by Rumbelhart, Hinton & Williams in 1986 [1] with the contribution of the study performed by Werbos [149]. The formal method allows the network to learn the association existing between the input patterns and the corresponding classes adapted to several level of neurons.

The BPA consist in the learning of a predefined set of input-output pairs given as examples. First an input pattern is applied as stimulus to the first layer of the network, and the information is propagated trough the complete network until a final output is generated. This output is compared with the ideal one that is available as the second part of the training pair and an error value is calculated. Following this calculation the error value is transmitted backwards from the neurons in the output layer to the hidden layer neurons that contribute to the obtained output value. This process is repeated layer by layer until all the neurons of the complete network have received an error value corresponding to its contribution to the general network error. Considering this relative error value the connection weight is adjusted for each neuron and connection of the network.

The importance of the BPA consists in its capability to adapt the weights corresponding to the intermediate neurons in order to learn the existent relation between the input pattern and its corresponding output value. It is important to remark the generalization capability of this algorithm, consisting in the capability to produce a satisfactory output to inputs that have not been used during the training phase. The network has to find an internal representation of the information that allows to generate the desired output when the training patterns are present, and in addition to be applicable to new input patterns to classify them correctly. This generalization capability is determined by the number and variation of the input patterns used during the training phase.

The BPA uses the gradient method to find the way to minimize the error.

3.4.1 The Gradient Method

Using the BPA the idea is get a successfully trained final network. The training will consist in obtaining the final values of the weight matrix and of the thresholds for each one of the artificial neurons of the network. We can consider that the training is finished when the w_{ij}

and θ_j values obtained correspond to W final matrix, so for each $X(t)$ input we obtain the ideal $Y(t)$ output of the training pair.

There is no analytical way to perform this calculation so a successive approximations method is used. During the sequence of iterations, the method tries to reduce the network error until reaching a predetermined objective. Due to the fact that we are using a supervised training method that include for each training pattern the input data and the ideal output associated we can conclude that the error of the network performance is entirely associated to the weights and thresholds values. This error function will be noted as $\varepsilon(W)$ and will be considered as the objective function to be minimized.

There are several method to minimize this objective error function inside the MLP training possibilities. For this thesis we will use the gradient method, that consists in initializing the network values with very small random values and updating its values at each iteration according to:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \eta \nabla_{ij}^{(l)}(t) \quad (3.14)$$

The elements of the equation are the following ones:

- l : It is referred to the index of the layer where the artificial neuron is situated, $l = 0, \dots, L$, being $l = 1$ the first hidden layer and $l = L$ the output layer.
- t : It corresponds with the iteration counter of the training process.
- η : Network learning rate, between 0 and 1.
- ∇ : Gradient of the objective function to be minimized $\varepsilon(W)$. It is defined as:

$$\nabla_{ij}^{(l)} = \frac{\partial \varepsilon(W)}{\partial w_{ij}^{(l)}} \quad (3.15)$$

The expression of equation 3.15 can be modified using the chain rule as:

$$\begin{aligned}
\nabla_{ij}^{(l)} &= \frac{\partial \varepsilon(W)}{\partial \hat{y}_j^{(l)}} \cdot \frac{\partial \hat{y}_j^{(l)}}{\partial w_{ij}^{(l)}} \\
\frac{\partial \hat{y}_j^{(l)}}{\partial w_{ij}^{(l)}} &= \frac{\partial f_j^{(l)}(w_j^{(l)} \hat{Y}^{(l-1)})}{\partial (w_j^{(l)} \hat{Y}^{(l-1)})} \cdot \frac{\partial (w_j^{(l)} \hat{Y}^{(l-1)})}{\partial w_{ij}^{(l)}} \\
\frac{\partial (w_j^{(l)} \hat{Y}^{(l-1)})}{\partial w_{ij}^{(l)}} &= \hat{y}_i^{(l-1)} \\
\frac{\partial f_j^{(l)}(w_j^{(l)} \hat{Y}^{(l-1)})}{\partial (w_j^{(l)} \hat{Y}^{(l-1)})} &= f_j^{(l)'} \\
\nabla_{ij}^{(l)} &= \frac{\partial \varepsilon(W)}{\partial \hat{y}_j^{(l)}} \cdot \hat{y}_i^{(l-1)} \cdot f_j^{(l)'}
\end{aligned} \tag{3.16}$$

Where:

- $w_j^{(l)}$: It is the row vector corresponding the weights of the connections of neuron j of layer l .
- $\hat{Y}^{(l-1)}$: It is the column vector corresponding to the inputs of layer l .
- $f_j^{(l)}$: It is the activation function of node j of layer l .

So we can define the error for each node as a delta term expressed as:

$$\delta_j^{(l)} = \frac{\partial \varepsilon(w)}{\partial \hat{y}_j^{(l)}} \cdot f_j^{(l)'} = \frac{\partial \varepsilon(w)}{\partial \hat{y}_j^{(l)}} \cdot \frac{1}{\hat{y}_i^{(l-1)}} \tag{3.17}$$

And the gradient of the objective function can be rewrite as:

$$\nabla_{ij}^{(l)} = -\delta_j^{(l)} \cdot \hat{y}_i^{(l-1)} \tag{3.18}$$

In order to be able to propagate the update of the weights based on the error correction we can express the gradient of the function considering the error terms of the previous layers:

$$\begin{aligned}
\frac{\partial \varepsilon(w)}{\partial \hat{y}_j^{(l)}} &= \sum_{n=0}^{N_{l+1}-1} \frac{\partial \varepsilon(w)}{\partial \hat{y}_n^{(l+1)}} \cdot \frac{\partial \hat{y}_n^{(l+1)}}{\partial \hat{y}_j^{(l)}} \\
\frac{\partial \hat{y}_n^{(l+1)}}{\partial \hat{y}_j^{(l)}} &= \frac{\partial f_n^{(l+1)}(w_n^{(l+1)} \hat{Y}^{(l)})}{\partial (w_n^{(l+1)} \hat{Y}^{(l)})} \cdot \frac{\partial (w_n^{(l+1)} \hat{Y}^{(l)})}{\partial \hat{y}_j^{(l)}} \\
\frac{\partial (w_n^{(l+1)} \hat{Y}^{(l)})}{\partial \hat{y}_j^{(l)}} &= w_{jn}^{(l+1)} \\
\frac{\partial f_n^{(l+1)}(w_n^{(l+1)} \hat{Y}^{(l)})}{\partial (w_n^{(l+1)} \hat{Y}^{(l)})} &= f_n^{(l+1)'} \\
\frac{\partial \varepsilon(w)}{\partial \hat{y}_j^{(l)}} &= \sum_{n=0}^{N_{l+1}-1} \frac{\partial \varepsilon(w)}{\partial \hat{y}_n^{(l+1)}} \cdot w_{jn}^{(l+1)} \cdot f_n^{(l+1)'}
\end{aligned} \tag{3.19}$$

Finally we obtain an expression with the relationship between the gradient, the error terms and the weights of the previous layer:

$$\frac{\partial \mathcal{E}(w)}{\partial \hat{y}_j^{(l)}} = \sum_{n=0}^{N_{l+1}-1} \delta_n^{(l+1)} \cdot w_{jn}^{(l+1)} \quad (3.20)$$

Now we can express the time variation of the connection weights as (with an equivalent expression for the threshold values):

$$w_{ij}^{(l)} = \begin{cases} w_{ij}^{(L)}(t) + \eta \hat{y}_i^{(L-1)}(t) \delta_j^{(L)}(t) \rightarrow \\ l = L; 0 \leq i \leq N_{L-1}; 0 \leq j \leq N_L - 1 \\ w_{ij}^{(l)}(t) + \eta \hat{y}_i^{(l-1)}(t) \delta_j^{(l)}(t) \rightarrow \\ l = L-1, \dots, 1; 0 \leq i \leq N_{l-1}; 0 \leq j \leq N_l - 1 \end{cases} \quad (3.21)$$

The error term of this expression is calculated as:

$$\delta_j^{(l)}(t) = f_j^{(l)'} \sum_{n=0}^{N_{l+1}-1} \delta_n^{(l+1)}(t) \cdot w_{jn}^{(l+1)}(t) \quad (3.22)$$

So to calculate the error term (and the the weights and thresholds) corresponding to a layer l it is needed to calculate before the values corresponding to the following layer $l+1$. This is not applicable to the last layer where the error term is calculated as the difference between the real and the ideal output of the network. The conclusion is that for calculating new values it shall be done from the output to the input of the network.

We can now define a training algorithm for the MLP with L layers and $N_1, \dots, N_L$ artificial neurons in each layer. We want to be able to solve problems with non linear separable regions so it is needed that the activation function of the nodes is not linear. For example a very useful function that will be used in this thesis is the sigmoidal function.

$$\begin{aligned} f(x) &= \frac{1}{1+e^{-x}} \\ f'(x) &= f(x)(1-f(x)) \end{aligned} \quad (3.23)$$

The steps to be executed to update the weights and thresholds to obtain a real output similar to the ideal one are the following ones:

1. Initialization of the network weights and thresholds to small random values.
2. Present to the input network an input pattern $X(t)$ corresponding to a training pair. At the output network we want to obtain an output $Y(t)$ of the same pair.

3. Calculate the output of the network according to the following expressions. For any layer l of the network:

$$\hat{y}_j^{(l)} = f \left(\sum_{i=0}^{N_{l-1}-1} w_{ij}^{(l)}(t) \hat{y}_i^{(l-1)}(t) - \theta_j^{(l)} \right); 0 \leq j \leq N_l - 1; 0 \leq l \leq L \quad (3.24)$$

Where

$$\begin{aligned} \hat{y}_j^{(0)} &= x_i \\ \hat{y}_i^{(L)} &= \hat{y}_i \theta_j^{(l)} \end{aligned} \quad (3.25)$$

4. Calculate the error term at the output as the basis for updating the weights of the previous layers:

$$\delta_j^{(L)} = \frac{\partial \mathcal{E}(W)}{\partial \hat{y}_j^{(L)}} f_j^{(L)'} \quad (3.26)$$

And the new weights for the output layer are:

$$\begin{aligned} w_j^{(L)}(t+1) &= w_{ij}^{(L)}(t) + \eta \delta_j^{(L)}(t) \hat{y}_i^{(L-1)}(t) \\ 0 \leq i \leq N_{L-1} - 1; 0 \leq j \leq N_L - 1 \end{aligned} \quad (3.27)$$

The error terms for the rest of the layers are calculated as:

$$\delta_j^{(l)}(t) = f_j^{(l)'} \sum_{n=0}^{N_{l+1}-1} \delta_n^{(l+1)}(t) \cdot w_{jn}^{(l+1)}(t) \quad (3.28)$$

And the weight modification for the rest of the layers is:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) + \eta \delta_j^{(l)}(t) \hat{y}_i^{(l-1)}(t) \quad (3.29)$$

5. In case the training objective is reached this is the final point of the algorithm. If not we have to go to the following iteration $t = t + 1$ using a new pattern in step 2.

We have to consider here the value of the learning rate for the algorithm. If this value η is near to 0 then the training will be slow but we can assure the convergence of the algorithm.

If the value is near 1 the training will be faster but the algorithm can be unstable. In order to assure the algorithm convergence the learning rate shall be between two values:

$$0 < \eta < \frac{1}{E[|X|^2]} = \frac{1}{\frac{1}{P} \sum_p \sum_i x_{ij}^2} \quad (3.30)$$

Being $p = 0, 1, \dots, P$ the counter of training patterns and x_{ij} the input value i of layer j coming from neuron j of layer $l - 1$.

In order to accelerate the algorithm convergence a good technique is to smooth the slope of the weight changes. To do so when a weight is updated we can consider the change experimented by the same weight in the previous iteration. This is done using a term called momentum α [150]. Applying this modification equation 3.29 can be rewritten as:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) + \eta \delta_j^{(l)}(t) \hat{y}_j^{(l-1)}(t) + \alpha [w_{ij}^{(l)}(t) - w_{ij}^{(l)}(t-1)] \quad (3.31)$$

$$0 < \alpha < 1$$

The BPA method is the most used for training MLP networks. In some concrete cases it is possible to find situations where the algorithm is less efficient or even unstable. This is normally due to an inadequate initialization of the weights and thresholds or an incorrect election of the learning rate value. It is considered a good practice to train the network with different parameter values in order to check and select the best performing configuration. Some of the problems related to the BPA are:

- In some cases the algorithm requires a lot of computation time to obtain simple solutions. It has to be considered that the calculation volume is proportional to the number of connections and normally when the size of the network increases the system needs more examples to complete the training.
- The possibility of "catastrophic interference" or performance worsened, as consequence of adding new training patterns.
- The maximum gradient path is not always the fastest path to the optimal minimum solution of the error function.
- If the slope of the gradient evolution is very small the training can take a lot of time.
- If the slope of the gradient evolution gets null then the training is stopped.

- The neurons can be paralyzed when the weights are adjusted to very high values, then the unit output can be saturated because the unit is working out of the variable part of the activation function. In order to avoid this it is recommendable to randomize the connection weights with small values and use a small learning rate, even if the training is slower.
- Time instability can happen if we use a high value for the learning rate, then there will be big increments in the weights values so it is easy to overpass the correct value for the weight and have to compensate in the following step. This could lead to continuous oscillations.
- The algorithm can be stuck in a local minimum of the error function. Using the descendent gradient method the algorithm follows the error function surface that can finish in a local minimum different from the global optimum minimum.

Chapter 4

Unsupervised Learning Networks

4.1 Introduction

Unlike traditional computing, Computational Intelligence (CI) is tolerant to imprecise information, partial truth and uncertainty on the final desired results of the learning process. Some kind of neural networks use a learning process known as unsupervised learning mode that presents a training process that does not include any knowledge of the ideal output of the network. During the training only the input patterns are presented to the network, and the system is intended to learn automatically how to modify its connection weights in order to cluster the data in different groups with similar features.

In contrast with the supervised learning the unsupervised methods don't require a correct answer associated to each input pattern in the training data set. The methods explore the underlying structure in the data, or correlations between patterns, and organize the inputs into categories considering these correlations.

There exist an extensive group of learning rules, algorithms and structures related to the unsupervised methods, including as the main ones Hebbian principal components analysis, competitive learning rules, Hopfield associative memory structures, SOFM, Learning Vector Quantization (LVQ) or ART algorithms. In this chapter we will review these variants in order to better understand the way of working of the unsupervised learning systems.

4.2 Hebbian Learning Rule

The equation that determines the operation of a classic artificial neuron is characterized by the weights applied to the neuron inputs and the threshold values for each concrete application. The learning method shall determine the way of choosing those values in order to be able to solve the problem considered. From the historical point of view, the Hebbian learning rule is the oldest of the possible learning method used to solve this problem. It was presented by Hebb [126] considering that the value of a synaptic union would be increased whenever the input and the outputs of the connected neurons were simultaneously activated. So as much as the connections are more frequently used they are reinforced emulating the biological phenomenon of habit and learning by means of repetition.

Mathematically, the Hebbian rule can be described as:

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_j(t) x_i(t) \quad (4.1)$$

Where x_i and y_j are the output values of neurons i and j respectively, which are connected by the synapse determine by weight w_{ij} with η as the learning rate. Note that x_i is the input to the synapse.

A remarkable property of the Hebbian rule is that the learning process is local to each connection. The change in every synaptic weight only depends on the activities of the two neurons connected by it. This allows to simplify greatly the complexity of the algorithms based on this rule.

A complete neural networks uses the Hebbian learning rule when the values of the weights are increased accordingly with the excitation level of the input and output neurons of the connections over the global structure. The complete learning method consists in successive iterations using only the network's input and real output but never using the desired ideal output, so it can be considered as an unsupervised learning method.

4.3 Hopfield Network

The Hopfield Network is a neural network structure based on the Hebbian learning rule [134]. The network is based on the concept of an energy function, the evolution of this

cost function is the guide to design recurrent networks and to understand its dynamic behavior.

The mathematical expressions linked to this structure indicate the principle of storing information in a network as a dynamically stable attractor. A Hopfield Network with n units has two versions considering the type of values used: binary and continuously valued. Let v_i be the state or output of the i th unit. For binary networks v_i is either +1 or -1 but for continuous networks v_i can be any value between 0 and 1.

Considering w_{ij} as the synaptic weight on the connection from units i to units j and the Hopfield network as a symmetric network ($w_{ij} = w_{ji} \forall i, j$) without self-feedback connections ($w_{ii} = 0 \forall i$), then the dynamics for the binary Hopfield networks are:

$$v_i = \text{sgn} \left(\sum_j w_{ij} v_j - \theta_j \right) \quad (4.2)$$

The dynamic update of the network state can be carried out in a synchronous or asynchronous way. In the first one all the neurons of the network are updated at every iteration meanwhile in the second one only one unit is updated at each step (this unit is selected randomly).

Once the status of the network is defined we can compute the energy stored in it, for a concrete state $v = (v_1, v_2, \dots, v_n)$ it is given by:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} v_i v_j \quad (4.3)$$

The most important property of the energy function is that as network state evolves according to the network dynamics, the network energy always decreases and eventually reaches a local minimum point (attractor) where the network stays with a constant energy. This implies that if a combinatorial optimization problem can be formulated as an energy function to be minimized, the Hopfield network can be used to find the optimal (or suboptimal) solution by letting the network evolve freely.

When a set of input data is used to train the network, the structure can be considered as an associative memory. Any pattern present in the basin of an stored pattern can be used as a kind of index to retrieve it from the network. An associative memory has two phases:

- Storage phase: The weights in the network are determined so that the attractors of the network memorize a set of patterns to be stored.
- Retrieval phase: The input pattern is used as the initial state of the network, and the network evolves according to its dynamics. A pattern is produced (or retrieved) when the network reaches equilibrium.

An example of a Hopfield network structure can be found on figure 4.1.

Fig. 4.1 Hopfield network structure

4.4 Competitive Learning Rules

This learning paradigm presents one main difference with the Hebbian learning rule. Meanwhile with the Hebbian learning several output units can be activated simultaneously, with the competitive learning rules the output units compete among themselves for activation. The result is that only one neuron can be active at any given time. This WTA approach can be found in different biological structures [2].

The competitive learning model is normally used in classification tasks where the patterns are grouped automatically based on data correlations. For each group created in the network

a single output unit is associated and activated every time a pattern belonging to the group is presented at the network's input.

Assuming a sequence of statistical samples of a vectorial observable $x = x(t) \in R^n$, where t is the time coordinate and a set of variable reference vectors $\{m_i(t) : m_i \in R^n, i = 1, 2, \dots, k\}$. If $m_i(0)$ is initialized in a random way for example, then $x(t)$ can be compared with all the existing $m_i(t)$ at each instant of time and $m_i(t)$ is to be updated to match even more closely the current $x(t)$. Considering the weight values w_{ij} for the connections of a single output unit to all the input ones, each output unit is also connected to the rest of the output units through inhibitory connections, and to itself with a self-feedback excitatory weight. The result of all this connections is that only one unit that presents the highest excitation is activated and the rest of the output units are forced to be inactive:

$$\|w_i^* - x\| \leq \|w_i - x\| \quad (4.4)$$

The learning rule expression can be stated as:

$$\Delta w_{ij} = \begin{cases} \eta (x_j^u - w_{i^*j}) & i = i^* \\ 0 & i \neq i^* \end{cases} \quad (4.5)$$

This means that only the weights corresponding to the winner neuron are updated, so the pattern information is stored only in the weights of the activated unit. In this way it can be considered that the output networks "moves" closer to the input cluster area. The network will not stop learning until the learning rate η is 0 so the output mapping can evolve and the same input pattern can fire different outputs in different stages of the training phase producing a problem with the stability of the system. The system is said to be stable if no pattern in the training data changes its category after a finite number of learning iterations. One way to achieve stability is to force the learning rate to decrease gradually towards 0 as the learning process takes place. However, this artificial freezing of learning causes another problem related to the plasticity, which is the ability to adapt to new data. This is known as Grossberg's stability-plasticity dilemma in competitive learning [151].

4.5 Self-Organizing Map

The SOM is a network structure that uses a learning system based in the competitive learning rules that was initially presented by Kohonen [136], [152]. It has the property of creating a spatially organized representation of the features included in the input patterns and their abstractions. In this respect this kind of structures are very similar to the topographically organized maps that can be found in the cortex region of most biological brains, where the recognition task usually take place.

A main principle in biological brain organization is that the neurons are grouped so those that are physically close collaborate in the processing of the same kind of stimulus [139]. This has been demonstrated according to the medical studies that show that different regions of the brain are dedicated to different specific tasks. According to [153] it is possible to induce both excitatory and inhibitory stimulus to modify and disturb the local brain functions. If this stimulus are located in a concrete brain region it is possible to study the associated cognitive task that is associated to this specific brain map.

The detailed organizational view of the brain has evolved, specially in complex animals with various cortices in the cell mass that seems to behave as maps [154]. A particular location of the neural response in the map directly correspond to a specific modality and quality of sensory signal. The internal representation of information in the brain is generally organized spatially. So artificial SOMs and brain maps have many features in common.

The SOM structure is based in neighboring cells in a neural network that compete in their activities by means of lateral inhibition connections. The neurons develops adaptively to be able to classify different input patterns into clusters. The SOM is a sheet-like ANN, whose cells are tuned to various input patterns that belong to different classes through an unsupervised learning process. As a WTA model, in the basic map only one cell at a time responds to the input pattern. Considered as a map of features, the location of the different responses tend to get ordered in a meaningful coordinate system for the different kind of inputs, in which similar models are closer to each other in the map than the more different ones. The SOM implements an orderly mapping of a high-dimensional distribution onto a regular lower-dimensional scheme, and the spatial location of each output unit is considered as the spatial domain of all the input belonging to that class. Summarizing, the dimension of

the input subspace is reduced, physically grouping the data.

The system is able to covert high complex information included in non-linear statistical relationship between patterns in a simple geometric interpretation in a reduced dimension map. The information can be considered as it was compressed maintaining the topological metrical behavior in a kind of map of abstractions. The presence or absence of an active response at a particular location provides an interpretation of the input information, and not so much the exact input-output signal transformation or magnitude of the response. The segmentation performed by the SOM is normally high efficient compared with typical neural network operations. Kohonen's SOM can be used for projection of multi-variable data, density approximation and clustering system.

The SOM usually consists in a two dimensional grid of neurons. The dimension of the input vector establishes an equivalent number of nodes of the input layer. The output layer dimension is reduced respect to the input one and establishes the final mapping possibilities. The input neurons drive the input vector to the output layer considering the connection weights that are the components of the reference vector associated with the output neuron. It is very important to clear define the neighborhood and distance measurement in this topology. An example of this structure can be found on figure 4.2. This structure has been used as basis for evolving models based in this philosophy [155], [156], [157].

The node of the output layer that is activated corresponds to the classification result obtained for the concrete input pattern. The basic idea is that inputs with similar characteristics fire the same output neuron while non-similar inputs fire different outputs. Only the neuron that presents the minimum difference between the input and the output weights will be activated, provoking the inhibition of the surrounding ones. At the beginning of the training the adjustment of the weights is done in a wide area around the winning node, but when the training progresses the affected area is reduced. Doing so the network self-organizes. We can consider that the algorithm behaves as a vectorial quantifier when the number of desired classes is previously defined and a sufficient amount of data is available to complete the training. It has to be taken into account that the result of the mapping may differ depending on the order of presentation of the input data, specially if the amount of input data available is small.

Fig. 4.2 SOM network basic structure

After a successful training the output neurons that are physically close present similar reference vectors. Then a labeling process is needed to define the classification so input patterns of known classes are presented to the network and depending on the output activated the class labels are finally assigned to that neurons.

4.5.1 Vector Quantization

As a previous step for the SOM learning algorithm explanation we will present the vector quantization method. This is a classification method that produces an approximation to a continuous pdf $p(x)$ of an input variable x using a finite number of coding vector $m_i, i = 1, 2, \dots, k$, [158], [159]. Using as origin a set of predefined codes the approximation of x consist in the calculation of the closed vector m_c from the available ones. Considering E as the power of the reconstruction error we have:

$$E = \int \|x - m_c\| p(x) dx \quad (4.6)$$

Considering dx as the volume differential in x space and index c as the best matching code:

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \quad (4.7)$$

So 4.6 defines a placement of the code vectors in the signal space as an approximation to the density function $p(x)$. For the delta rule that can be used to define these values, let $m_c = m_c(t)$ the closest code to the input $x = x(t)$ considering an euclidean metric, and the gradient descent-step optimization of E as:

$$\begin{aligned} m_c(t+1) &= m_c(t) + \alpha(t) [x(t) m_c(t)] \\ m_i(t+1) &= m_i(t) \text{ for } i \neq c \end{aligned} \quad (4.8)$$

With $\alpha(t)$ monotonically decreasing between 0 and 1. The $m_i(t)$ of the above process actually is converted into a set of input feature detectors. In the above process and its biophysical counterparts, all the cells act independently. Therefore the order in which they are assigned to the different domains of input signals is more or less random, most strongly depending on the initial values of the $m(0)$ [160].

4.5.2 SOM Training Method

In order to provide an explanation of the self-organizing function we will focus in the simplest and most effective form of the algorithm, even in authentic biological networks this tendency may be disguised by additional brain functions. Two essential effects that lead to the organization of the information in SOM structures are:

- Spatial concentration of the network activity around the neuron that it is best tuned to the concrete input.
- Further sensitization or tuning of the best-matching cell and its topological neighbors to the concrete input.

There is a bi-dimensional grid formed by the different output neurons, that are connected in parallel to the input neurons processing the input pattern note as $x = [x_1, x_2, \dots, x_n]^T \in R^n$. The weight vector corresponding to a neuron i is denoted by $m_i = [m_{i1}, m_{i2}, \dots, m_{in}]^T \in R^n$, and it is initialized randomly. From here it is possible to calculate a measure of the match between the input and the weights using the inner product $x^T m_i$, but considering the mapping nature of the algorithm it is better to use the euclidean distance. The minimum distance defines the winner output neuron.

For the formation of the maps it is fundamental in the learning process that the neurons are interconnected between them, this is derived from the competitive learning paradigm. Thus working with topological related subsets it is needed to impose an interconnected

correction mechanism. Through the learning process the data subsets will activate different cells, so the nets corrections at each cell will tend to be smoothed in the long term. The weight vectors tend to reach values that are ordered along the axes of the vectors.

In this neural network model it is needed to model the spatial neighboring cells connections using lateral feedback connections and other kind of interactions like the inhibitory ones. To enforce lateral interaction directly in a general form, for arbitrary network structures we can define different sets of cells N_c around a concrete cell c . At each iteration of the learning process the cells inside N_c are updated, meanwhile the cells out of the sets are left unaltered. The neighborhood is centered in that cell which presents the best match for input x :

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \quad (4.9)$$

The size of the group of neurons associated can be time-variable. At the beginning the set is bigger and will be smaller trough the execution of the learning process. The acquired order may not be destroyed later on, an finally considering the WTA approach $N_c = \{c\}$ the learning is reduced to simple competitive learning. The final result is the topological order of the map structure.

The updating process can be expressed in mathematical notation as:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t) [x(t) m_i(t)] & \text{if } i \in N_c(t) \\ m_i(t) & \text{if } i \notin N_c(t) \end{cases} \quad (4.10)$$

Where $\alpha(t)$ is a scalar adaptation gain between 0 and 1 that should decrease with the iterations for this kind of methods. Kohonen presented this approach with an alternative notation introducing a kernel function $h_{ci} = h_{ci}(t)$

$$m_i(t+1) = m_i(t) + h_{ci}(t) [x(t) m_i(t)] \quad (4.11)$$

The simplest form of this kernel function is being equal to $\alpha(t)$ inside the neighborhood set, and equal to 0 outside this group. But a more biological like function can be selected, for example a bell curve [152], considering always that the value should decrease with time.

In order to successfully apply this algorithm some elements have to be considered:

- Since learning is a stochastic process, the final statistical accuracy of the mapping depends on the number of steps, which must be reasonably large. The number of steps must be at least 500 times the number of network units. The number of components in x has no effect on the number of iteration steps,
- At the beginning of the process $\alpha(t)$ shall be close to 1 and after that it shall present a monotonically decremental evolution. The decreasing rule may present different forms depending on the concrete application. The ordering of m_i occurs during the initial period, meanwhile the rest of the training is used for fine adjustment.
- The choice of the neighborhood shall carefully be taken into account, if the initial value is too small then the map will not be ordered globally and various small parcels of the map will be ordered separately.
- Sometimes it has been suggested that x should be normalized before it is used in the algorithm. Normalization is not necessary in principle, but it may improve numerical accuracy because the resulting reference vectors tend to have the same dynamic range in this situation.
- Random initialization was originally used to show that there exists a strong self-organizing tendency in the SOM, so that the order can even emerge when starting from a completely unordered state. But if the initial values for the model vectors are selected as a regular array of vectorial values that lie on the subspace spanned by the eigenvectors corresponding to the two largest principal components of input data, computation of the SOM can be made faster by orders of magnitude. The SOM is then already approximately organized in the beginning and it is possible to start with a narrower neighborhood function and smaller learning rate factor.

4.5.3 Discussion on SOM ANN

Considering the SOM structure there are several aspects that may be considered that will lead to the evolution of this system in more complex ones. Among this elements to be considered inside the SOM paradigm the following ones can be highlighted:

- Using primary signal elements (as temporal samples of a speech waveform or pixels of an image) may not be useful for the SOM structure. This is specially true for fine-structure input patterns, for example lines in a drawing. It is not possible to find the

differences between the global elements if the information is not transformed before the network processing.

- The maximum capacity of a map has to be considered. For example if it is possible to use a SOM for storage a large amount of knowledge. Normally the brain maps are not so extensive, they are used to process small amount of data to be used to improve the results of the associative memory.
- One beneficial property of the maps is that their parameters usually stabilize out into a narrow dynamic range, and the accuracy requirements are then modest. In this case even integer arithmetic operations provide sufficient accuracy
- A basic one level map is capable of creating an structured representation of the input data. It may be possible to hierarchically link several structures of this type in a kind of deep network [161], [162].
- Normally the use of SOM is dedicated to visualization of data spaces. This kind of structures has been used with success applied to state diagrams for processes and machines or data mining applications like statistical tables or document collections.

4.6 Learning Vector Quantization

If a system based on the SOM structure is going to be used as a classifier of input pattern into discrete subsets then it is possible to introduce some differences in the method by adding a supervised learning modification to improve the results. The SOM as the regular vector quantization methods try to approximate the input signal values or their associated pdfs by vectors in order to minimize a quantization error function. If the input data set is going to be classified into a finite number of predefined classes that may include more than one vector in each one of them, it is not important to distinguish the results between the concrete vector. Only the decisions made at class border are important.

It is possible to calculate effective values for the vectors such they define near-optimal decision border between the classes, even in the sense of classical Bayesian decision theory. These methods are called Learning Vector Quantization [163], [164], [165]. An example of the LVQ structure can be found on 4.3. The network present three layers: an input layer, a hidden layer and the output layer. The network is fully connected between the input and hidden layer, but only partially connected between the hidden and the output layer, where

each output neuron is linked only to a cluster of hidden neurons. Hidden and output neurons only present binary outputs, so when an input pattern is supplied to the network the hidden neuron which vector is closest to the pattern wins the competition and activates the output neuron linked to the cluster that contains the active hidden neuron.

Fig. 4.3 LVQ network basic structure

4.6.1 Type One LVQ

If each class contains more than one vector m_i , the class regions are defined by simplest nearest-neighbor comparison of the input x and the set of vectors. The class of x is defined by the set that contains the corresponding m_i .

To define the optimal placement of m_i in a learning process the first step is to use the SOM algorithm to determine the overall statistical density function of the input space. The following step is to determine the label of the vectors using a number of inputs with a known classification result and to assign the neurons to the different classes according to the frequency of each m_i being the closest to the calibration vector of a particular class. This part of the method is considered as supervised learning because the system uses the available knowledge of the classes corresponding to each model vector.

It is possible to improve the classification results if the vectors are pulled away from the decision borders in order to define the separation between classes more accurately. If m_c is

the vector closest to an input x considering the euclidean metrics, this vector also defines the final classification of x . To update the vectors the following cases apply:

$$\begin{aligned}
 m_c(t+1) &= m_c(t) + \alpha(t) [x(t) - m_c(t)] \\
 &\quad \text{if } x \text{ is classified correctly} \\
 m_c(t+1) &= m_c(t) - \alpha(t) [x(t) - m_c(t)] \\
 &\quad \text{if } x \text{ is classified incorrectly} \\
 m_i(t+1) &= m_i(t) \\
 &\quad \text{for } i \neq c
 \end{aligned} \tag{4.12}$$

Considering $\alpha(t)$ a scalar gain between 0 and 1 that decreases in time.

This algorithm tends to reduce the point density of m_i around the decision surfaces. The minus sign in the second equation of 4.12 may be interpreted as a correction in the same direction for the class to which m_c belongs but with the probability function of the neighboring class subtracted from that of m_c . The difference between the density functions of the neighboring classes by definition drops to zero at the Bayes border, so previous expression induces more accurate classification results.

4.6.2 Type Two LVQ

The previous algorithm can be modified to comply better with the Bayesian decision theory [164]. Considering to vectors m_i and m_j that correspond to different classes and are closest neighbors in the vector space, if these vectors are initially in the wrong position the incorrect discrimination border will be the mid plane between the two vectors. It is possible to define a symmetric window of non-zero width around the mid plane and set that the correction of the vectors shall only be performed if the input x is inside the window on the wrong side of the mid plane.

The corrections of both vectors in average shall have a direction so the mid plane moves towards the crossing surface of the class distributions and asymptotically coincide with the Bayes decision border. The mathematical expression of this is the following one:

$$\begin{aligned}
m_i(t+1) &= m_i(t) - \alpha(t) [x(t) - m_i(t)] \\
m_j(t+1) &= m_j(t) + \alpha(t) [x(t) - m_j(t)] \\
&\text{if } C_i \text{ is the nearest class, but } x \text{ belongs to } C_j \neq C_i \\
&\text{where } C_j \text{ is the next class, and } x \text{ falls inside the window} \\
m_k(t+1) &= m_k(t) \\
&\text{for the rest of the cases}
\end{aligned} \tag{4.13}$$

The optimal width of the window must be determined experimentally, and it depends on the number of available samples. The classification accuracy of the LVQ2 is improved when the decision surface is shifted towards the Bayes limit but after that, m_i continues moving away. Therefore this algorithm shall be applied for a relatively short time only.

4.6.3 Type Three LVQ

In the LVQ2 algorithm there is no limit to the modification of m_i that can derive in a long run of the process. There are two negative effects related to the long application of the algorithm. The first problem is that corrections are proportional to the differences between x and m_i , and x and m_j , so the correction of m_j as the correct class is bigger than the corresponding to the incorrect class m_i . This leads to monotonically decreasing distances between the vectors $\|m_i - m_j\|$. It is possible to compensate this accepting all the training vectors from the window if one belongs to the correct class and the other to the incorrect class.

The other problem comes from the continuous application of 4.13 that may lead to another asymptotic equilibrium of m_j different from the optimal one. The way to solve this is to include the following modifications in the mathematical approach:

$$\begin{aligned}
m_i(t+1) &= m_i(t) - \alpha(t) [x(t) - m_i(t)] \\
m_j(t+1) &= m_j(t) + \alpha(t) [x(t) - m_j(t)] \\
&\text{if } m_i \text{ and } m_j \text{ are the nearest vectors to } x, \\
&\text{and } x \text{ and } m_j \text{ belong to the same class} \\
&\text{while } x \text{ and } m_i \text{ belong to different classes} \\
&\text{and } x \text{ falls inside the window} \\
m_k(t+1) &= m_k(t) + \epsilon \alpha(t) [x(t) - m_i(t)] \\
&\text{for } k \in \{i, j\} \text{ if } x, m_j \text{ and } m_i \text{ belong to the same class}
\end{aligned} \tag{4.14}$$

Notice that in LVQ1 only one of the m_i values was changed at each step, meanwhile LVQ2 and LVQ3 change two vectors simultaneously.

4.7 Adaptive Resonance Theory Models

Considering the stability-plasticity problem presented in section 4.4 relative to the competitive learning paradigm, Carpenter & Grossberg presented some learning models that solve this issue [166], [167]. The problem can be summarized as how is possible to learn new patterns (plasticity) and prevent the existing knowledge to be erased or corrupted (stability).

Inside the ART one basic idea is that top-down learning expectations focus attention upon bottom-up information in a way that protects the previously learned memories to be erased by new patterns. The system still enables new knowledge to be incorporated in the system considering having sufficient output units in the network, without using them until it is really needed. The resulting ART structure is a neural network that self-organize stable recognition codes in response to arbitrary sequences of input patterns, as a special case of the more general cognitive process.

4.7.1 Stability-Plasticity Problem in Competitive Learning

ART models are an evolution of the analysis of the basic competitive learning structures. In this basic model a set of input patterns is presented to a network F_1 that can modify its adaptive weights or Long Term Memory (LTM) traces used in the connections between F_1 to a coding level F_2 . Level F_2 is designed as a competitive network capable of choosing the neuron that receives the adequate input in a WTA model. The winning population then triggers associative pattern learning within the vector of LTM traces which sent its inputs through the adaptive filter.

Consider X as the input of the i th node v_i of $F_1, i = 1, \dots, M$, y_i the activity or Short Term Memory (STM) of v_i and v_j the j th node of $F_2, i = M + 1, \dots, N$. Let w_{ij} be the adaptive weight (related to LTM) of the connection between v_i and v_j , then the normalized activity of v_i in response to input pattern X is:

$$y_i = \frac{x_i}{\sum_{k=1}^M X_k} \quad (4.15)$$

We can consider for simplicity that the output of v_i is still X so the total signal received at v_j is:

$$T_j = \sum_{i=1}^M x_i w_{ij} \quad (4.16)$$

and the output of this node is:

$$y_j = \begin{cases} 1 & \text{if } T_j > \max(T_k : k \neq j) \\ 0 & \text{if } T_j < \max(T_k : k \neq j) \end{cases} \quad (4.17)$$

This summarizes that the node in F_2 that receives the largest signal is chosen for STM storage. For the LTM storage terms only the winning node is changed and the rest remains unaltered. Vector W_j learns by reducing the error between itself and the normalized vector X in the direction of the steepest descent.

$$\Delta w_{ij} = \epsilon y_j (-w_{ij} + y_i) \quad (4.18)$$

Depending on the concrete data set it has been proved [168] that if not many input patterns are presented to F_1 , or if the input pattern forms not too many cluster relative to the number of coding nodes in F_2 , the learning of the recognition code eventually stabilizes and the learning process selects the best distribution of the weight values. But it is possible that a competitive model not always learns an stable code in response to an input. Moreover, the F_2 response to a given input pattern might never settle down as learning progresses. Such unstable learning in response to a previous input is due to the learning that occurs in response to the new presented inputs patterns. So the network's adaptability, or plasticity, enables prior learning to be washed away by more recent learning in response to a wide variety of input environments. Learning can also become unstable due to changes in the input environment as for example probability distributions or a deterministic sequence of inputs

ART learning was introduce to integrate the competitive learning model into a self regulating control structure with a stable knowledge and efficient response to an arbitrary sequence of input patterns.

4.7.2 ART1 Model

The ART1 architecture is shown in figure 4.4. It is capable of performing a stable learning of an input set of binary sequences until it uses the full memory capability. The adaptive weights of the connections oscillate during the learning in response to the inputs in order to avoid getting stuck in a local minimum of the memory state. After the learning is stable then

each input activates the more representative F_2 neuron.

Fig. 4.4 ART1 network basic structure

The ART1 structure encodes binary input patterns in the weight values of a bottom-up adaptive filter w_i . A second top-down adaptive filter w_j contained in the paths from F_2 to F_1 performs the classification stabilization tasks, it plays the role of learned expectation in the ART system.

The algorithm behaves in the following way:

1. An input pattern X activates F_1 that fires the node v_{j1} at F_2 which receives the largest signal from the previous layer.
2. F_2 reads its top-down expectation to F_1 , so the bottom-up input pattern and the top-down expectation are matched in F_1 .
3. If there is a mismatch then a reset is activated from F_1 to F_2 . This reset shut off the node v_{j1} for the rest of the coding cycle and deactivates the top-down expectation controlled by v_{j1} .
4. In this case F_1 reactivates the input signal and F_2 reinterprets it so other neuron v_{j2} is selected.
5. The complete cycle is repeated until one of the following cases takes place:

- A node v_{jm} is correctly chosen. The extent of similarity is controlled by a vigilance parameter $\rho, 0 < \rho < 1$, which in fact determines the number of final categories.
- A previously uncommitted node from F_2 is selected.
- The full capacity of the network is reached and it is not possible to include the new input.

Until one of the possibilities occurs there is no learning in the network because all the STM variations happen so fast that the slower LTM cannot respond at the same rate. Significant learning only happens at the real end of each learning iteration for each input pattern.

Mathematically, given a input vector X of length n the output of the auxiliary unit that checks the resonance is given as:

$$A = \text{sgn} \left(\sum_j x_j - n \sum_i y_{2j} - 0.5 \right) \quad (4.19)$$

Where y_{2j} depends on if a neuron is active. The output of an F_1 unit is:

$$y_{1j} = \text{sgn} \left(x_j + \sum_i w_{ij} y_{2i} + A - 1.5 \right) = \begin{cases} x_j & \text{if no output } O_j \text{ is on} \\ x_j \cap \sum_i w_{ji} y_{2i} & \end{cases} \quad (4.20)$$

Reset signal is only active is the similarity is less than the vigilance level controlled by ρ .

When the testing cycle finishes with a match of an existing node, the input pattern and the expectation modify the activity response in the first layer F_1 that now computes a fusion between both. In this case the input pattern and the expectation mutually reinforce each other by the feedback of the resonance of the STM activation. The real learning is inside the LTM weights of the connections, that include the new information the fusion inside F_1 . The system allows alteration of one of its prior learned codes only if an input pattern is sufficiently similar to what it already knows, to allow a further refinement of its knowledge.

If the cycle ends with the selection of a new node then bottom-up and top-down filters are linked in the first layer of the network but there is no alteration of the the activity response in F_1 . If the capacity of the network is reached and no adequate match happens then the learning is automatically inhibited.

The ART method refines the already learning codes using new information, and also new patterns can be included in the system without the risk of erasing the previous acquired knowledge. The method ensures that if the same input data is repeatedly presented to the network, its LTM and STM are unchanged (the network is stable). Also, if there are sufficient output neurons to represent all the different classes new patterns can be learned and assigned to an uncommitted node (the network is plastic).

4.7.3 ART2 Model

One limitation of the ART1 is the exclusive application to digital binary inputs. In order to be able to apply the ART paradigm to analog input the ART2 model was developed [169]. Considering the architecture of the system a sequence of arbitrary input patterns can be modified using after a pre-processing modification, and then the outcome of this change is used as the input of the network. The typical pre-process for the ART2 system enables to deform, shift, dilate and rotate the original data to be recognized as examples of the same category.

ART 2 architectures can autonomously classify arbitrary sequences of analog input patterns into categories of arbitrary coarseness while suppressing arbitrary levels of noise. The structure accomplishes this by modifying the ART1 architecture to incorporate solutions of several additional design problems into their circuitry. In particular, level F_1 , is split into separate sub-levels for receiving bottom-up input patterns, for receiving top-down expectations, and for matching the bottom-up and top-down data.

A particular structure responding to ART2 model can be found in figure 4.5. The feature presentation field F_1 consist in 4 loops. An input pattern will be processed first in the two lower loops to eliminate the inherent noise and to accentuate the main features of the signal. This enhanced input will go to the upper levels of F_1 to excite the classification neurons in F_2 . The rest of the mathematical processing is similar to the ART1 model.

Fig. 4.5 ART2 network basic structure

Chapter 5

Metaplasticity Concepts

5.1 Introduction

Metaplasticity is a concept derived from the biological science that has been proved to be applicable to other areas of knowledge as neuro-science, computer science, medical science, psychologist and physiologist among others [6], [7], [8], [9], [10]. Some authors consider this process of an importance such as being the key element in the memory and learning process in human brain [170].

As an introduction to the metaplasticity concept it is necessary to present some studies from the biologist literature. Ramón y Cajal [171] considered that the neural nervous system was formed by individual elements called neurons that can be connected between them in order to form a bigger and more complex general system. The communication between those separate elements is called synapses. These synapses are neural structures that modulate the activity of one of the neurons, converting this activity into a higher or lower activation (voltage) in the other neuron involved in the connection. The magnitude that relates the activity to the activation is called synapses efficiency or weight. The author indicated that the occurrence of continuous activity, in form of excitations or inhibitions, in those connections was related with the memory and knowledge mechanism through a process called synaptic plasticity. Konorski [172] indicated that the simultaneous activation of two different neurons may create new physical connections between the cells that didn't exist before. This situation is related to persistent modifications of the biological memory. Finally Hebb [126] indicated that during training the synaptic connections are enforced due to the correlated activity between pre and post synaptic neurons, as reflected in the Hebbian learning rule.

5.2 Plasticity in Synapses

In order to be able to represent the models related to the intelligence behavior as memory or learning, it is needed to be able to incorporate information to the neurons. The synaptic connections between neurons shall be modifiable, so the efficiency of those connections is variable, changing its characteristics depending on the experience. The ability of the synaptic connections to modify its efficiency is known as synaptic plasticity and it is related to several cellular situations that are able to modify the behavior of the global neural system.

Synaptic plasticity refers to the efficient modulation of information transmission between neurons, being related to the regulation of the number of ionic channels in synapses. Synaptic plasticity mechanisms involve both molecular and structural modifications that affect synaptic performance, either enhancing or depressing neuronal transmission.

The synapses may modify its structure and characteristics to adapt the subject behavior to the changes in the environment. These alterations may last only a short period of time or have a higher duration, even become permanent, depending on the repetition rate of the stimulus that causes the synaptic alteration [173].

As the synapses takes place between two neurons, a wide variety of changes may affect to any one of the two cells. These changes can be applied to the pre-synaptic neuron modifying its properties related to the liberation of neurotransmitters, or to the post-synaptic neuron affecting its response capability to those neurotransmitters. Normally there is a relationship between the changes affecting both neurons, and the continuous occurrence may cause the connection to have a higher or lower efficiency. The first situation is called Long Term Potentiation (LTP) and the second one is called Long Term Depression (LTD). These long term changes are mainly affecting characteristics as memory or learning, meanwhile short term changes allow the neural system to be able to process and integrate the information with more or less efficiency in a short period of time [174], [175].

LTP was presented by Bliss & Lomo in [176]. The authors demonstrated that high frequency stimulation of the neurons of a concrete part of the hypo campus provokes an increment of its activation potential levels that is maintained even for a period of days. When the neurons are not active the reception channel for the neurotransmitter is blocked by magnesium ions, but if the post-synaptic neuron receives an intense stimulus maintained in time

then the reception channel is opened and connected to the transmitter. This causes a change in the polarization of the post-synaptic neuron. This depolarization provokes the expulsion of the magnesium ions and allows the entrance of the calcium ions inside the post-synaptic neuron. The calcium activates some specific proteins that causes the post-synaptic neuron to be more sensitive to future stimulus, so the LTP is facilitated. This process can be observed graphically on figure 5.1.

Fig. 5.1 Long Term Potentiation process

LTD is a perdurable reduction in the synapses efficiency provoked by a reduction of the sensibility of the post-synaptic neuron [177], [178]. LTD is induced by low frequency stimulus that generates a reduction in the post-synaptic neuron calcium levels. If the synapses have experimented a LTP before, the threshold for the neuro-transmitter is higher, what may increase the probability of the reduction of the calcium flow causing a LTD. This negative feedback maintains the synaptic plasticity of the connection. This process can be observed

graphically on figure 5.2.

Fig. 5.2 Long Term Depression process

The calcium levels determines if a LTP or LTD takes place in the connection between neurons. A small increase in the calcium levels causes future depression, meanwhile a small reduction causes a future potentiation. This behavior allows the system to maintain an equilibrium as an homeostatic process.

To model brain learning we can use the synaptic weight modification, considering that the weight changes depend on the value of post-synaptic activation. The curve presented in figure 5.3 is obtained by modifying the response of the post-synaptic neuron to a pre-synaptic stimulus. If the value of this post synaptic depolarization is above the LTP threshold (white dot), a positive variation of synaptic weight, that is synaptic potentiation, occurs. If the value of post-synaptic activation is between the LTP threshold and the LTD threshold (black dot), a

negative variation of synaptic weight, that is, synaptic depression, takes place. Below the LTD threshold, neither potentiation nor depression occurs.

Fig. 5.3 Synaptic weight modification of a biological synapse according to its post-synaptic activation

5.3 Intrinsic Plasticity

The intrinsic plasticity concept is related to the internal electrical characteristics of a single neuron in opposition to synaptic plasticity which is related to the connection between two neurons itself. This plasticity is considered as the persistent modification of the neuron inherent characteristics due to the external synaptic activity. It is regulated by the physiological characteristics of the ionic channels of the neuron membrane and it has a considerable influence in other neural processes as the synaptic integration, the electric signal propagation and the biological metaplasticity.

Intrinsic plasticity is considered as an homeostatic process that leads the neuron to a balance behavior. The system adjust the excitation and inhibition synaptic capabilities so each neuron operates in a optimal range related to its own characteristics. This process

balances each neuron operation, leading the complete system to a global balanced behavior, trying to avoid extreme situation like total inhibition or saturation in the neuron responses.

5.4 Biological Metaplasticity

Synaptic metaplasticity is defined as the plasticity property of synaptic plasticity. Metaplasticity concept was introduced by Abraham [6] related to the neuron's capability to regenerate synaptic connections or the creation of complete new synaptic connections, and now is applied to other fields of biology, neuroscience, physiology and neurology. It was defined as the induction of synaptic changes based on the previous synaptic activity [179]. So metaplasticity is highly dependent on the synaptic activation history.

In neurological sciences the metaplasticity concept is used to refer to a higher plasticity level, considered as a change in the way synaptic plasticity takes place. Metaplasticity process is closely related to the variations in the depolarization levels in the post-synaptic neuron that affect the synaptic plasticity. As presented before those variations are able to facilitate the synaptic potentiations and to inhibit the synaptic depressions or the other way around, depending on the concrete change of the ionic levels of the post-synaptic neuron. LTP takes place after high levels of post-synaptic depolarization, and LTD takes place after low levels of post-synaptic depolarization.

Metaplasticity is revealed in the neurons characteristics as changes in the LTP and LTD thresholds in the synaptic connection, considering the initial synaptic weight value and the activity levels that have affected the connection [180]. Considering these changes it is possible to create a link between the metaplasticity concept and other behaviors with high level of abstraction as learning or memory [170].

Considering the curve included in figure 5.3 and the works presented by Ropero-Peláez & Andina in [14] metaplasticity can be represented as variations in the the curve elongation with respect to the level of synaptic activity or a concomitant change in the synaptic weight. It implies a rightward shift of the LTP threshold T_M , according to the averaged level of post-synaptic firing $\bar{a}_M, M = 0, 1, 2, \dots$, as shown in figure 5.4. If post-synaptic activity is high, the metaplasticity property will cause the curve to be shifted reinforcing the LTP. The curves corresponding to the higher levels of post-synaptic firing $\bar{a}_M$ have higher values for the LTP threshold. According to other studies [181] it is proved that LTD thresholds are also

modified in the same circumstances. So if synapses are positively primed (i.e. there is an increment in weight), the interval between thresholds broadens, thereby favoring subsequent synaptic depression. If, instead of potentiated, synapses are negatively primed, the depression interval between both limits becomes narrower, thereby favoring future synaptic potentiation.

Fig. 5.4 LTP modification of a biological synapse according to its post-synaptic activation

The value of the activation levels determines the changes in the synaptic weights and the learning capability associated. Not only biological synapses strength changes with its participation in neurons activity, but also the efficiency of the change is different depending on the stimulus. According to [13] high post-synaptic activity is related to high frequency excitations, and in the same way the curves representing low previous synaptic activity correspond to low frequent excitations.

Figure 5.5 represents the effects of biological metaplasticity considering the variations of the synaptic weight Δw . If the regime of synaptic activations is changed, so that w reaches a higher value w_1 , metaplasticity is manifested by the modification of the shape of the curve. In this new situation, if a concrete post-synaptic voltage is applied (see asterisk in the curve), instead of obtaining a null variation of weight, the new curve yields a negative variation (a decrement) of weight (see vertical arrow). So LTP metaplasticity is conventionally referred to as the rightward shift of the LTP threshold for higher synaptic weights.

Though metaplasticity represents a compensatory mechanism that slows down the rate of synaptic efficiency growth, it does not prevent the connection to either saturate or become zero. In case the synaptic weight increments its value, the activation levels are increased

Fig. 5.5 LTP and LTD modification of a biological synapse according to the change of the synaptic weight

too. This causes the weight to be increased again entering in a positive feedback loop that eventually leads the connection to saturation. Similar reasoning can be made with a weight decrement. Biological neurons avoid these situations by making use of homeostatic compensatory mechanisms as intrinsic plasticity. As explained before according to intrinsic plasticity, highly activated neurons shift their activation thresholds towards higher levels of post-synaptic activity so they become less excitable. Conversely, less activated neurons shift their activation threshold towards lower levels of post-synaptic activity, becoming more excitable in the future. The purpose of intrinsic plasticity is to shift the position of the activation threshold so that the sensitive region always corresponds with the average input level [182]. This situation shows the close relation between the different plasticity concepts involved in learning.

5.5 Artificial Metaplasticity

The application of the metaplasticity concept to the ANN field is based in the Hebb theory [126] that establishes that the learning property of the biological neural system is related to the synaptic connections. This weight value of these connections is heavily modified by the

correlation between the activation activities of the pre and post-synaptic neurons.

Synaptic plasticity property applied to the ANN is modeled usually as changes in the weight values associated to the connections between artificial neurons [137]. These weight values are the artificial equivalency to the biological synaptic efficiency. These values are the most important parameters related to the learning and performance of the ANNs. The different characteristics presented for the biological networks as the intrinsic plasticity or the metaplasticity can be applied to the weights associated to the connections in the different ANN structures. Those biological capabilities may improve the results obtained with the ANN [183], [184], [185], [186].

5.5.1 Application of Shannon's Information Theory to Artificial Metaplasticity

Considering the different approaches for the application of the biological concepts presented preciously to the ANN, we will focus in this section in the one that applies the Shannon's information theory [11] to the AMP. The basic concept relies in the idea that less frequent patterns of a data set contain more information than the more frequent ones [13].

The relation between AMP and information theory is focused in the objective of improving the results in learning convergence and performance by capturing information associated with significant rare events. It is based on the idea of modifying the ANN learning procedure such that less frequent patterns which can contribute heavily to the performance, are considered with greater relevance during learning without changing the convergence of the error minimization algorithm. It has been based on the hypothesis that biological metaplasticity property may have a significant effect on the adaptation of natural neural processes to extract more information from less frequent patterns (low synaptic activity) that, according to Shannon's theory, implicitly carry more information. So considering the resemblance with the biological model it can be interpreted that the less frequent patterns induce more adaptation of the neural system. Instead of repeating the same type of patterns that can cause a bias in the network towards one particular class, placing greater emphasis on the diversity of patterns may cause an improvement in the performance of the results.

If we consider the input patterns that are presented to a network, the data set will normally respond to a concrete probability density function, so we will have a higher amount of patterns where the pdf is bigger. In consequence the network will be able to predict with more precision the region of the input space that corresponds with a higher pdf. For example if the network works as a classifier capable to distinguish between two different classes, each one with its own pdf, the final input of the network will be a set of data centered in the regions where each pdf have the higher values. For example considering two Gaussian distributions centered in 1 and -1, most of the inputs will be located around this two values, as can be seen in figure 5.6.

Fig. 5.6 Normal sampling for two Gaussian distribution corresponding to two different classes

On the contrary if we are working on a classifier the most interesting region is the one between the two classes where the inputs may likely belong to any one of the classes with more or less similar probability. It will be more difficult for the network to correctly classify the inputs in this area. So it will be very practical to use the probabilistic information available of the input data set to give more importance to this kind of inputs in order to obtain more information for the classification task. In this way it is possible to reduce the number of training iterations or to obtain better precision with the same number of inputs. This importance sampling modification is can be observed in figure 5.7.

This concept was introduced by Kahn [187] considering the importance sampling as a technique based on the modification of the underlying statistical values of the system making

Fig. 5.7 Importance sampling for two Gaussian distribution corresponding to two different classes

the outputs that contribute in a more important way, to be generated with higher frequency. The author demonstrated that if the simulation statistics are modified in the correct way it is possible to reduce the variance of the results for the same amount of calculations. The biggest difficulty to apply this concept is to get the probabilistic information of the input data set, because it is not always possible to get the analytical pdf of the input patterns, so a suboptimal approximation can be used.

The relation with the metaplasticity is based on the hypothesis that biological synaptic metaplasticity could be directly influenced by the information carried by the input stimulus of the neurons, or training patterns in its artificial counterpart. As stated by the information theory, the data patterns that are less frequent carry more information than the frequent ones. So, it seems reasonable to design ANN learning in order to extract more information from less frequent patterns by being more sensible to them during the training phase.

If the strategy of an ANN learning procedure is to minimize an expected error E_M defined by the following expression:

$$E_M = \varepsilon \{E(x)\} \quad (5.1)$$

Being x a random variable of the training input vector $x = (x_1, x_2, \dots, x_n), (x \in R^n)$, with R^n the n -dimensional space and $E(x)$ the expression of a given error criterion as a function of the inputs applied in ANN training, used to update the weights in each training iteration step, then it is possible to make the following manipulation:

$$E_M = \int_{R^n} E(x) f_X(x) dx = \int_{R^n} e(x) dx = \int_{R^n} \frac{e(x)}{f_X^*(x)} f_X^*(x) dx = \epsilon^* \left\{ \frac{e(x)}{f_X^*(x)} \right\} \quad (5.2)$$

Al calculate E_M through the following estimator:

$$\hat{E}_M = \frac{1}{M} \sum_{k=1}^M \frac{e(x_k^*)}{f_X^*(x_k^*)} \quad (5.3)$$

Where $x_k^*, k = 1, \dots, M$ are independent sample vector with a pdf $f_X^*(x)$ that can be chosen by the designer if $f_X^*(x) \neq 0$ wherever $e(x) \neq 0, \forall x \in R^n$. From equation 5.2 the optimal value for the pdf is:

$$f_X^*(x)_{opt} = \frac{1}{E_M} e(x) \quad (5.4)$$

In practice is impossible to know $f_X^*(x)_{opt}$, but a sub-optimal function can be chosen in order to improve the training convergence. The more closer to the optimum the better results should be achieved. The idea is that an error objective function can be weighted by a proper function without affecting the final error objective for each class. In figure 5.8 we present a block diagram for this weighted training.

The proposed hypothesis is giving more priority for weights updating to the less frequent activations over the more frequent ones, then metaplasticity is being artificially modeled. This hypothesis connects AMP with information theory entropy concept: less frequent events carry more information than frequent ones. Proposed learning applies AMP to gain efficiency, reducing the number of patterns necessary to complete the training [14].

5.5.2 Probabilistic Computation Derived from Synaptic Behavior

This subsection is devoted to the study of the incremental and probabilistic versions of the synaptic rule [131] considering the mentioned synaptic properties [188]. According to

Fig. 5.8 Weighted training cycle. Note that the expected output will not be used for unsupervised training. Weighting function performs the artificial metaplasticity through statistical information extracted from input patterns.

the incremental version of the pre-synaptic rule the variation of the synaptic weight w is calculated as:

$$\Delta w = \xi I (O - w) \quad (5.5)$$

Where I and O are the pre-synaptic and post-synaptic activation potential probabilities and ξ a small positive constant. According to the work presented in [131] and [189] equation 5.5 is asymptotically equivalent to a probabilistic computation for the weights:

$$w = P(o/i) \quad (5.6)$$

Being $P(o/i)$ the conditional probability of a post-synaptic potential o given a pre-synaptic potential i . A very simple ANN approach with its first step considering a single rate code neuron with an output that can be interpreted as a firing probability. The second step is based in a spiking neuron model with a binary output depending on the calculated firing probability. For the rate code neuron:

$$e_j = w_j I_j \quad (5.7)$$

Where I is the pre-synaptic potential, w is the synaptic weight and e is the excitatory post-synaptic potential for a concrete synapses j . Considering this, the activation a is calculated as the sum of the post-synaptic potential of all the synapses involved:

$$a = \sum_{j=1}^n e_j = \sum_{j=1}^n w_j I_j \quad (5.8)$$

The probability O calculation consist in the application of a non-linear function to the activation level (for example a sigmoid):

$$O = f(a) \quad (5.9)$$

To simplify the mathematical expression it is better to only consider the linear part of the function (it is biologically coherent considering plasticity properties that allow the activation functions of biological neurons to be shifted so that their linear part is placed around the neuron's average operation point). So O can be expressed as:

$$O = ka + b \quad (5.10)$$

Being k and b numerical factors. This expression is used to calculate the synaptic weights variations according to equation 5.5.

For the spiking neuron the synaptic weights are calculated based the correlation between pre-synaptic and post-synaptic potentials considered as binary values (if the potential is triggered it is considered as 1, if not it is considered as 0). If i is the output of the pre-synaptic neuron and o is the output of the post-synaptic neuron then the synaptic weights are calculated according to the probabilistic expression of the pre-synaptic rule as:

$$w = P(o/i) = \frac{n(o \cap i)}{n(i)} \quad (5.11)$$

Where n in the numerator counts the number of concurrent pre-synaptic and post-synaptic unitary outputs. The calculation of e and a , being w a conditional probability, responds to:

$$e_j = P(o/i_j) I_j \quad (5.12)$$

In the spiking neuron o is generated according to O which is obtained from a as:

$$O = \frac{1}{1 + e^{-(a-s)/T}} \quad (5.13)$$

Where parameter T is usually called temperature and regulates the inclination of the curve and parameter s is the shift of the sigmoid.

Once that the mathematical basic is set, it is possible to propose a relation with the synaptic properties referred previously [188]:

- For the incremental version of the pre-synaptic rule from equation 5.5, it is possible to check the consistency with the synaptic direction, because if $I = 0$ then there is neither potentiation, nor depression, so term I can be considered as an allowance term.

Considering the intrinsic plasticity property, O can be made proportional to a in the linear part of the function. From $O = ka + b$, simplifying $k = 1, b = 0$, then the probability of potential in the pre-synaptic neuron I can be replaced by a/w and the calculation of the synaptic weights expressed with an equation that represents the metaplasticity of the LTP threshold as:

$$\Delta w = \xi \left(\frac{a}{w} \right) (a - w) = \xi \left[\left(\frac{a}{w} \right)^2 - a \right] \quad (5.14)$$

For the contribution of various synapses the post-synaptic activation a is the sum of the contribution of the synapses involved wI and the potential added by nearby synapses a_n so:

$$a = wI + a_n \quad (5.15)$$

That applied together with equation 5.5 leads to:

$$\Delta w = \xi \frac{[a - a_n]_+}{w} (a - w) \quad (5.16)$$

Where $[a - a_n]_+$ is the positive component of $(a - a_n)$ or 0 if $(a - a_n)$ is negative. For low post-synaptic activations $a < a_n$ neither potentiation nor depression takes place, and for $a = a_n$ the LTD threshold is set. If the same mathematical approach is applied to equation 5.14 then:

$$\Delta w = \xi I (wI + a_n - w) \quad (5.17)$$

That represents the capability of a LTP threshold to be shifted leftward for greater values of a_n . This equation shows the plasticity properties presented in [190] as synaptic potentiation with negligible values of I in the case of very large influence of a_n .

Finally to consider the metaplasticity of LTD threshold due to the leakage of ions that takes place in the biological post-synaptic activation, it is possible to model this situation with a subtractive term proportional to the synaptic weight:

$$a = IW - kw + a_n \quad (5.18)$$

That applied together with equation 5.5 leads to an equation that represents that LTD threshold metaplasticity:

$$\Delta w = \xi \left[\frac{a - a_n}{w} + k \right]_+ (a - w) \quad (5.19)$$

Graphical representation of these equations can be found on [188].

- For the probabilistic version of the pre-synaptic rule from equation 5.6 it was assumed the linearity in the neuron activation function. If the output given by the sigmoid function based on the activation potential is:

$$O = f(a) \quad (5.20)$$

Now the neuron output o is binary and the probability of a unitary output is given by a non linear logistic activation function that yields a binary value 1 or 0 according to the probability of O :

$$o = f'(a) \quad (5.21)$$

According to this the variation of the synaptic weights $w - w_0$ can be calculated for each activation a and initial weight w_0 using equation 5.6, and binary output o calculated with equation 5.21 using binary input i from probability I . Even in the case of not considering the synapses of nearby connections, simplifying $I = a/w$ it is possible to observe the LTP threshold metaplasticity, but not the LTD threshold metaplasticity

property.

Adding the influence of nearby synapses and the the leakage of ions corresponding to post-synaptic activation by:

$$I = \left[\frac{a - a_n}{w} + k \right]_+ \quad (5.22)$$

It is possible to represent both LTP and LTD metaplasticity in the mathematical expression.

Finally if we consider the shift value s from equation 5.13, the intrinsic plasticity property of the synapses can be represented in a simple way. It is possible to add this property in the calculation of the shift s at iteration t based on the value of the shift and the output probability of the neuron O in previous iteration $t - 1$:

$$S_t = \frac{v \cdot O_{t-1} + S_{t-1}}{v + 1} \quad (5.23)$$

Where v is the shifting velocity that adjust the shifting rate of the activation function.

Graphical representation of these equations can be found on [188].

Chapter 6

AMP Applied to Supervised Learning

6.1 Introduction

The synaptic plasticity concepts presented in the previous chapter can be applied to the ANNs structures and algorithm in different ways. This chapter presents a possible implementation of the artificial metaplasticity and its relationship with the Shannon's information theory applied to supervised learning methods, concretely to the training algorithm of a MLP neural network. The idea is to be able to improve the ANN training in terms of a more efficient learning and final performance, in a way that would be also coherent with the biological hypotheses.

6.2 AMP Implementation in MLP Training

Considering the mathematical theory introduced in section 5.5.1 it is possible to improve the results obtained in the network training process taking advantage of the Shannon's information theory. This theory establishes that less frequent patterns provide more information than more frequent ones, so considering the pdf distribution of the data set under study it is possible to integrate this information in the training model to improve the overall results of the process and the final performance.

To integrate this idea in the training algorithm of a MLP network the implementation of the BPA has to be reformulated. This implementation is called AMMLP model, and consists in that during the training phase, the weight matrix W , that models the synaptic strength of its artificial neurons, is updated according to the probability of the input patterns and therefore of the corresponding synaptic activations. During the training phase, the AMMLP could

be considered a new probabilistic version of the pre-synaptic rule, as during this phase the algorithm assigns higher values for updating the weights to the less probable activations than to the ones with higher probability.

To analytically introduce AMP in arbitrary MLP training and considering the function of equation 5.4 related to the distribution of the input vectors, $x \in R^n$, with R^n the n-dimensional space $x = (x_1, x_2, \dots, x_n)$, $x_i \in R^1$, $i = 1, 2, \dots, n$, it is applied to the error function to be minimized in the BPA:

$$\widehat{E}[W(t)] = \frac{E[W(t)]}{f_X^*(x)} \quad (6.1)$$

Where $W(t)$ is the weight matrix that contains the information of the synaptic connections for each iteration t and $E[W(t)]$ is the error function considered in the classic BPA. By introducing the pdf information of the data set in function $f_X^*(x)$ it is possible to model the AMP inside the BPA.

The modified error function $\widehat{E}[W(t)]$ is minimized at each iteration for each weight update. Considering $l, j, i \in N$ as the indexes for MLP layer, node and counter respectively, for each component $w_{ij}^{(l)}(t) \in R$ part of $W(t)$ it is possible to provide an expression for the weight reinforcement:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \eta \frac{\partial \widehat{E}[W(t)]}{\partial w_{ij}^{(l)}} = w_{ij}^{(l)}(t) - \eta \frac{1}{f_X^*} \frac{\partial E[W(t)]}{\partial w_{ij}^{(l)}} \quad (6.2)$$

Being $\eta \in R^+$ the learning rate value. As the weighting function proposed depends only on the pattern distribution instead than on the network own parameters, the AMMLP algorithm can be summarized as a weighting operation for updating each connection weight in each iteration of the learning process:

$$\Delta^* w = w^*(x) \Delta w \quad (6.3)$$

With $\Delta w = w(t+1) - w(t)$ as the weight update value for the classic BPA and w^* the realization of the described weighting function for each input pattern. In this way the probabilistic information included in the input patterns is introduced in the algorithm as a way of modeling the metaplasticity concepts, concretely it can be considered as a new probabilistic version of the pre-synaptic rule [191], [188].

6.3 Modification of the BPA

Previous studies have demonstrated that a classifier based on MLP trained using the BPA is itself and estimator of the *a posteriori* probability of the class of the input pattern [13], [192]. This idea allows a direct implementation of the AMP in the BPA algorithm for a MLP.

If the system is designed to classify L classes $H_l, l = 0, 1, \dots, L - 1$ considering the Bayes theorem and the use of the output of the network as an estimator of the probability of the input data set:

$$\hat{y}_l = P(H_l/x) = \frac{f_X(x/H_l)P(H_l)}{f_X(x)} \quad (6.4)$$

For equation 6.1 we can assume that $f_X^*(x) = f_X(x)$ and then for each class l :

$$\hat{E}_{M_l} = \frac{1}{M_l} \sum_{k=1}^{M_l} \frac{E(x_k) f_X(x_k/H_l)}{f_X(x_k)} \cong \frac{1}{M_l} E(x_k) \frac{\hat{y}}{P(H_l)} \quad (6.5)$$

That leads to the inclusion of the probability related function f_X^* in the error calculation equation of the BPA as:

$$\begin{aligned} \frac{\partial \varepsilon(W)}{\partial w_i^{(l)}} &= \frac{\partial}{\partial w_i^{(l)}} \left(\frac{1}{2} \frac{(y - \hat{y}^{(l)})^2}{f_X^*(x)} \right) = \frac{1}{f_X^*(x)} \frac{\partial \varepsilon(W)}{\partial w_i^{(l)}} \\ \delta_j^{(l)} &= (y - \hat{y}^{(l)}) \frac{f_j'^{(j)}(x)}{f_X^*(x)} \end{aligned} \quad (6.6)$$

This expression includes the propagation of the error to previous layers as in usual BPA. The inclusion of the probability function in the mathematical expression of the general algorithm affects the weight values in each iteration as:

$$w^*(x) = \frac{1}{f_X^*(x)} = \frac{\hat{y}_l}{P(H_l)} = \frac{f_X(x/H_l)}{f_X(x)} \quad (6.7)$$

The idea is to use the existing information of the distribution of the occurrence probability of the inputs patterns to improve the classification results and the speed of the learning curve (being able to use a smaller number of patterns as inputs to the network to obtain the same results, that can be very useful to avoid overspecialization if the data base contains very few available patterns). At this point two possibilities are considered for the algorithm:

- Considering a estimation of *a posteriori* probability: Previous mathematical model allows a direct implementation of the metaplasticity by assuming that once than a network has started to learn, the output of the network is an estimation of the input

class probability, so it is used to quantify a pattern's frequency. For this implementation if the output is far from the expected ideal result the learning ratio is reinforced in the next iteration, and if the output is close to the expected one the the learning ratio is slowed down. Note that if the output is not related to the input occurrence frequency, as it happens in first steps of BPA training algorithm with a randomly initialized network, the training may not converge. In this first steps, the outputs of the MLP does not provide yet any valid estimation of the probabilities, but rather random values corresponding to initial estimation of the MLP weights. It is then better in these first steps of training, either to apply ordinary BPA training or to use another valid weighting function till the training process starts to minimize the error objective.

Considering the final output of the network we can use the following function for applying the metaplasticity theory. If the desired output activation correspond to input vectors $x \in H_1$ so $y = 1$, then the AMP is implemented by:

$$f_X^*(x) = \hat{y}_L \quad (6.8)$$

And for the complementary class patterns $x \in H_0$ so $y = 0$:

$$f_X^*(x) = (1 - \hat{y}_L) \quad (6.9)$$

This method allows using AMMLP without any kind of previous knowledge of the pdf of the input data set.

- Considering a suboptimal solution as *a priori* estimation of the input data set pdf: One typical plausible example for $f_X^*(x)$ is

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^N} \cdot e^{B \sum_{i=1}^N x_i^2}} = \frac{1}{w_X^*(x)} \quad (6.10)$$

where $w_X^*(x)$ is defined as $1/f_X^*(x)$, N is the number of neurons in the MLP input, and parameters A and $B \in R^+$ are algorithm parametric values which depend on the specific application of the AMMLP algorithm. Equation 6.10 is a Gaussian distribution, so it has been assumed that X pdf is Gaussian. Then, $w_X^*(x)$ has high values for infrequent x values and close to 1 for the frequent ones and can therefore be straightforwardly

applied in weights updating procedure to model the biological metaplasticity during learning.

The approximation provided by equations 6.8 and 6.9 is more accurate when the training evolves, and using equation 6.10 or nominal BPA is recommended for first training iterations where the *a posteriori* approximation is not valid and can lead to convergence problems.

The AMMLP is modeled in the algorithm by the application of any of the proposed weighting functions in the weight updating process during the learning phase:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) + \eta \delta_j^{(l)} \hat{y}_i^{(l-1)} \quad (6.11)$$

Where $w_{ij}^{(l)}$ are the synaptic weights of a neuron j in the layer l in a training iteration t , $\hat{y}_i^{(l-1)}$ is the output of neuron i of the previous layer (the input x_i for the first layer) and $\delta_j^{(l)}$ the backpropagated error term. To calculate the error of the last layer L being y the ideal output and considering a sigmoid activation function:

$$\delta^{(L)} = (y - \hat{y}^{(L)}) \cdot \hat{y}^{(L)} \cdot (1 - \hat{y}^{(L)}) / f_X^*(x) \quad (6.12)$$

As a summary this method allows a faster learning than classic ones and the resulting systems achieve performance that competes with the most powerful methods of the state-of-the-art, for any multidisciplinary real data sets where it has been tested [13], [15], [124], [125]. This is possible by obtaining information from the data sets, not only from typical patterns in them, but also from atypical ones. Even the method is implemented for an MLP it is applicable to several ANN structures as several learning paradigms rely on the minimization of an error function.

AMMPL also offers an alternative paradigm of deep learning as the improvement of overall learning does not rely in the increasing complexity of the artificial intelligence system by adding neuron layers, but in the more efficient model of the learning in each of its components. This fact produces an emergent ability on the intelligent system for deeper information extraction and information classification improvement on the data set. Due to the general nature of the AMP concept, that is as wide as plasticity itself, this method has the potential to be generalized in machine learning.

Chapter 7

AMP Applied to Unsupervised Learning

7.1 Introduction

The metaplasticity theory exposed previously can also be applied to unsupervised neural network structures. In this case a novel category of bio-inspired neural network known as the Koniocortex-Like Network is studied, presenting its resemblance with the human brain structures and the mathematical analysis that integrates the AMP behavior. In the KLN model competition and pattern classification emerges naturally due to the interplay of inhibitory inter neurons, metaplasticity and intrinsic plasticity.

7.2 KLN Biological Basis

The KLN model was introduced by Ropero-Peláez & Andina in [193], [194]. It is based in a region of the human brain named koniocortex and it tries to reproduce the brain behavior related to learning tasks by taking advantage of the synapses plasticity possibilities already studied.

The koniocortex is a common denomination for all regions of the cerebral cortex containing a granular layer (layer IV). The granular texture of this layer is due the abundance of spiny stellated neurons that directly receive neural projections from the thalamus. The thalamus, at the center of the brain, is the main relay station from the senses to the cortex. The KLNs are neural models that possess at least two layers: the first layer containing neurons that are similar to the thalamus-cortical neurons of the thalamus, and the second layer whose neurons resemble the spiny and the inhibitory inter neurons of the fourth layer of the koniocortex. This system behave as a topographic map that changes it boundaries and receptive fields

according to sensory experience. In other words, map reorganization and feature extraction can be performed in a simple two layered neural network.

As demonstrated with living brain tissues in which only the thalamus and the fourth layer of the cortex were active (remaining layers were reset through freezing [195]), these two layers constitute a network in which competition and auto-organization are found. For example, the biological koniocortex network exhibits competition [196] because only a very small number of spiny stellated neurons are active in the presence of sensory stimuli. This behavior resembles the WTA process of competitive artificial networks. In WTA, the most active neuron remains active while the other neurons are set to zero.

The difference between conventional competitive networks and the biologically inspired KLN is that, while conventional competitive neurons find the most active neuron through calculation, in the case of the KLN, the winning neuron emerges naturally from the interaction between the neurons. At the same time non-winning nodes become silent due to the dynamics of the neurons in the network, not because they are algorithmically reset but with lateral inhibition as an important driving force. Previous works also studied the neural dynamics leading to emergent competition in terms of the different properties potentially involved in the process, like the strength and range of lateral inhibition [197], [198], [199], the value of the firing threshold [197], [200] and the steepness of the activation function [200]. Additionally findings related to homeostatic neural mechanisms such as metaplasticity and intrinsic plasticity contribute to WTA dynamics and make competition possible [201], [202], [182], [203].

7.3 Sequential Implementation of AMP in the KLN Model

The relation between AMP and this unsupervised neural network model is based in the properties presented in 5.5.2. The main properties involved in competitive learning are synaptic metaplasticity and intrinsic plasticity. As it was presented intrinsic plasticity adjusts the global excitability of the neuron so that highly excited neurons will be less excitable in the future, and vice versa. The idea is to implement a probabilistic computation at the neuronal level consistent with psycho-physical experiments showing a Bayesian type computation [204], [205], [206], [207].

The evolution from a complete artificial neural framework to this biologically inspired neural network is made in different steps replacing non-biological properties by biological ones. In the KLN a WTA kind of operation takes place without the need of applying an external function for forcing the maximally activated neuron. Different versions of this network are presented in [193] where the evolution of the system can be followed in detail:

1. Version 1 of the network is a Bayesian decision framework used as basic structure to place the biological properties: In this model the input patterns are normalized, these inputs are presented to the network and all the patterns belonging to a specific category produce the firing of a specific neuron, while remaining neurons are forced to be silent. Along the presentation of a certain category of patterns, the associated neuron set of weights converge through a process of Hebbian modification to a prototype T_j .

In the testing phase all prototypes are projected over the testing input pattern. The neuron that fires with the highest output indicates the category to which the testing pattern belongs to. This is similar to the Bayes decision rule. The structure of this network can be found on figure 7.1.

Fig. 7.1 Structure of the KLN network version 1: Bayesian decision framework

2. Version 2 of the network added competition to the network, not biological but externally driven: This model is able to perform feature extraction and classification tasks. Neuron's output in this network results from its inherent dynamics that is highly influenced by intrinsic and synaptic plasticity. Over this process, there is an external modification that checks the output of the network and the highest output is forced to be 1 and the rest of the outputs are set to 0 in a WTA approach. This is similar to apply

an activation function to the Bayesian decision rule. After applying the activation function to the projections generated by the input data over the prototypes of each neuron output, the system obtain the winning neuron. This output is set to 1 and the rest of the neurons are set to 0. Competition is not a result of the network dynamics but due to an external algorithm.

For the following input patterns it has to be noted that the pre-synaptic rule makes the associated weights to be modified. The weights from active inputs to active neurons are reinforced, and the weights from active inputs to non active neurons are reduced. Weights from null inputs to non winning neurons remain unchanged.

Additionally intrinsic plasticity modify the activation functions: the activation curve of the active neuron is shifted rightwards, and the activation curves of non winning neurons are shifted leftwards. Then this intrinsic plasticity compensates the weight changes associated to the pre-synaptic rule by making the winning neuron less sensitive due to the shift of the activation function, and leading to have a different winning neuron depending on the input. Along this process each prototype T_j evolves from representing each category of the input data. The structure of this network can be found on figure 7.2.

Fig. 7.2 Structure of the KLN network version 2: Artificial competition added to the model

3. Version 3 includes a type of lateral inhibition that together with the previous plasticity properties allows competition in a biologically plausible way. This lateral inhibition is included in the inherent dynamics of the neural network behavior jointly with intrinsic plasticity and the pre-synaptic rule:

- Intrinsic plasticity performs an homeostatic control of the neuron's firing rates.
- The activation curves present steep slopes corresponding to the sigmoidal functions.
- Pre-synaptic rule modifies the weights according to the activation of each neuron.
- Lateral inhibition allows each neuron to inhibit its neighbors in a competitive frame.

The network is created with very small weights that initially don't allow the inputs to generate significant outputs. This make the sigmoidal activation functions to be shifted leftwards until one of the neurons is fired becoming the winning one. The firing of this neuron precludes remaining neurons to fire due to lateral inhibition. This neuron will stop winning in the future because of intrinsic plasticity shifting its activation function rightwards. In this model inhibitory connections weights are fixed instead as being modifiable as the rest of the weights associated to the neurons. The structure of this network can be found on figure 7.3.

Fig. 7.3 Structure of the KLN network version 3: Lateral inhibition added to the model

4. Version 4 is the final KLN network: This structure performs competition and pattern classification similarly to previous version. The difference is that lateral inhibitory connections are substituted by inhibitory inter neurons with non modifiable negative connections. Biological neurons involved in a competitive process are never inhibitory as they are in version 3, the idea is to keep the neuron as excitatory but including in

the structure ancillary inhibitory inter-neurons. In this model competition and learning emerge from individual neuron properties without the need of any external supervisor. A simplified structure of this network can be found on figure 7.4.

Fig. 7.4 Simplified structure of the KLN network final version

7.4 Complete KLN Model

The KLN network is based in a mathematical model formed by rate code neurons whose outputs O_j are limited between the values 0 and 1. These values represent the probability of occurrence of an action potential. Considering in one side the normalized input pattern $\vec{i} = \vec{I} / \|\vec{I}\|$ (lower case notation meaning vector normalization) as the net-input of neuron j . Normalization is performed with the l_1 -norm in which:

$$\|\vec{I}\| = \sum_{i=1}^n |I_i| \quad (7.1)$$

And in the other side the neuron's j weights as the components of a vector prototype $\vec{T}^j$, so that $\vec{T}^j = \vec{W}^j = [W_{j1}, W_{j2}, \dots, W_{jn}]$. The inner product of weights and the pattern to be classified, the net-input of neuron j is calculated as the modulus of the projection of prototype $\vec{T}^j$ over input pattern $\vec{I}$:

$$net_j = \|\vec{W}^j \cdot \vec{i}\| = \|\vec{T}^j \cdot \vec{i}\| = \|\vec{T}^j_I\| \quad (7.2)$$

During the training the weights are modified using the incremental version of the pre-synaptic rule:

$$\Delta\omega = \xi I(O - \omega) \quad (7.3)$$

Where O and I are the post-synaptic and pre-synaptic action potential probabilities, respectively, and ξ , a learning factor. The application of metaplasticity to this pre-synaptic rule is explained in detail in section 5.5.2.

For relating the net-input of neuron O^j to its firing probability, O_j a conventional sigmoidal activation function was used:

$$O_j = \frac{1}{1 + e^{-k(net_j + 0.5 - 2s^j)}} \quad (7.4)$$

Where k is a curve-compressing factor and s^j the horizontal shift of the activation function ranging from 0 to 1, $0 < s^j < 1$. In this implementation the adjustment of the shifting is completely shifted leftwards with $s^j = 0$ and when it is completely shifted rightwards with $s^j = 1$.

Real neuron exhibits intrinsic plasticity [182],[203]. The homeostatic property that makes very active neurons to be moderated and inactive neurons to increment its firing rate as shown in figure 7.5. According to this property [203], the activation function gradually shifts leftwards or rightwards regulating the activation of scarcely or highly activated neurons, respectively.

In the model, parameter s^j is mathematically incorporated to the simulations in the neuron's activation function f relating the net-input of the neuron to its spiking probability O_j :

$$O_j = f\left(\left\|\vec{T_I^j}\right\|, s^j\right) \quad (7.5)$$

The following equation calculates the shift of the activation function, s at time t in terms of the shift and output probability of the neuron at time $t - 1$.

$$s_t^j = \frac{vO_{t-1} + s_{t-1}^j}{v + 1} \quad (7.6)$$

Fig. 7.5 Intrinsic plasticity allows the neurons' activation function to shift horizontally so that the activation function "follows" the average net-input of the neuron. (a) Initial position of the sigmoidal activation function. (b) In the case of a low regime of net-input values (as in A, B and C), intrinsic plasticity shifts the sigmoid leftwards. (c) In the case of a high regime of net-input values (as in D, E and F), intrinsic plasticity shifts the sigmoid rightwards increasing the sensitivity of the neuron

Where v is the shifting velocity parameter. It is a small arbitrary factor for adjusting the shifting rate of the activation function. Notice that when both the shift and the output at time $t - 1$ are equal, the shift at time t continues having the same value than at time $t - 1$.

Figure 7.6 is the complete version of the KLN model used in the experiment. In the KLN structure, B labeled neurons are inhibitory neurons including intrinsic plasticity capabilities. S labeled neurons are the main neurons engaged in competition and also present intrinsic plasticity. Since each S contacts a single B , intrinsic plasticity is concomitantly regulated in both types of neurons. So if S is highly activated it is the same for associated B . This implies that S reduces its excitability and B the inhibitory field surrounding S , affecting the final activated neuron in future classification performances. TC neurons can use intrinsic plasticity to remove the mean of a series of input values. When removing the average value, patterns become more uncorrelated and easier to classify.

Figure 7.6 shows that each S neuron has a recurrent connection on itself that was initially intended for allowing a sustained activation over time in simple rate-code neurons. Recurrent connections are extremely rare in real neurons. Despite of this, this kind of recurrent connection was indeed present in the koniocortex.

Fig. 7.6 Complete structure of the KLN model

Finally *SB* neuron is incorporated to the model to be used in pattern normalization. Similarly to real shunting/dividing inter-neurons, *SB* neurons perform the arithmetical summation of its inputs (*TC* outputs), dividing the activation of its target neurons (the *S* neurons) by this quantity.

Testing of this model with synthetic inputs can be found on [193] including its adaptation capability to modify the results of the network if some of the synthetic input pattern models is substituted by other one in the middle of the training. In [194] a more complex associative neural network is developed based on the KLN, capable of dealing with incomplete synthetic patterns and suited to operate similarly to a LVQ network.

7.4.1 Details of the KLN Modeling

The notation used in the model for naming the different neurons of the KLN model is taken from the real koniocortex brain structure. So *TC* means thalamus-cortical, *S* means spiny, *B* means basket and *SB* means shunting basket neurons. In principle the architecture included in figure 7.6 is almost identical to the real biological one. Only one exception is considered due to uncertainties in the biological literature that does not clarify completely the spiny neuron connections: in order to establish a fixed model, each spiny neuron was connected to a single basket neuron and each basket neuron to a spiny neuron not being the one from which it receives the original stimulus.

The main resemblance between the KLN and the real koniocortex is that inhibition is predominant in the network dynamics. This allows that only highly tuned neurons are activated and the WTA process to take place without external inference. In addition other similarities between both structures can be found:

- The biological spiny neurons connections undergo synaptic potentiation and also depression, and the KLN *S* neurons connections are plastic including those potentiation and depression changes by mean of the pre-synaptic rule.
- The real koniocortex neurons use an activation function with a significant steep slope. In the same way the KLN model presents a sigmoid function modified by the compression factor k that allow to control the concrete characteristic of the slope.
- The spiny neurons are capable to adapt its firing to the spike frequency, so within an action potential train the frequency gradually decreases and each sudden output

increment is followed by a gradual decrement. The KLN model includes a recurrent connection that allows the output slowly fade away. This kind of auto-connection is rare in real biological neurons but it does appear in the koniocortex spiny neurons [208].

- The real shunting basket neurons perform precise summation of the thalamus-cortical inputs, and the modeled *SB* neurons are used to perform the l_1 normalization by summing their inputs and using the result to normalize the *S* neuron inputs.

On the contrary the main difference between koniocortex and the KLN is that koniocortex preserves the topography of sensory patterns but the KLN does not.

Intrinsic plasticity is not usually included in the different models of the brain structure [209], normally because the shift variations are negligible when the network is stabilized at a permanent regime of activity. But without this mechanism the shift variation is eliminated from the dynamics and this may lead to competition collapse and memory stored in synaptic weights to be lost.

The KLN uses the linear independent components of the input data to classify the data set as the real koniocortex deals with orthogonal components arriving from the thalamus. The intrinsic plasticity implementation in the *TC* layer contributes to subtract the average neuron's activity level from current activity, so an average pattern is eliminated from each incoming pattern contributing to highlight the differences between input patterns.

Chapter 8

Experiment 1: Radar Signal Detection

8.1 Introduction

The first experiment included in this thesis is devoted to checking the theoretical hypothesis presented in section 5.5.1 about the relation between AMP and Shannon's information theory, considering the probability information of the input data set and its applicability to the AMMLP model to improve the training speed and performance of the model.

In order to clearly demonstrate the applicability of the hypotheses, synthetic inputs have been used instead of real data. Doing so we are able to control the concrete pdf characteristics of the input data comparing it with the theoretical limits of the detector. In this case synthetic radar signals have been used for this purpose in the different variations of the experiment.

The results of these experiments have been published in [13] and [119].

8.2 Radar System

The radar is a radio-determination system based in the comparison between radio-electric signals that are reflected or retransmitted from a position to be determined. The radar mainly consists in an electronic system used to detect objects and to determine the distance to those objects' position. The system send Radio Frequency (RF) waves to these objects, the waves are reflected in the object surface and received back in the radar system that is able to calculate the distance to the object based in the time used by the signal to complete the path. This functional principle can be observed in figure 8.1.

Fig. 8.1 Functional principle of a radar system

Usually radar is a term used to classify the systems that operate in the microwave area of the specter. In this zone it is possible to obtain high performance in the transmission of the electromagnetic waves in the atmosphere independently of the solar illumination, the rain and the cloud conditions.

In the beginning it was the defense area the one that initiated the radar development and evolution. Through the time this technology has been transferred to the civil and commercial areas in the telecommunication field. During the last years the scientific investigation has pushed strongly in the development of new applications to the radar systems working in the Earth observation using detection techniques in radar system embedded in airplanes or satellites. Nowadays the radar systems are used in different fields like navigation, air traffic control and weather phenomena detection.

8.2.1 Radar System Scheme

A radar system is mainly composed by an electromagnetic wave emitter connected to a very directive antenna, and a receiver working in the same frequency margin connected to an additional directive antenna situated as closer as possible to the emitter's one and oriented in the same direction. The emitter transmits the electromagnetic wave using its antenna in a concrete direction, then part of the RF energy is reflected when the waves hit an object and this reflected energy is received at the receiver's antenna as an echo signal. In some models the emitter and receiver use the same antenna using a diplexer. Additionally between the emitter and the receiver a synchronizing mechanism is needed to adjust the working cycles of both elements. Finally at the receiver's output an indicator is connected to present the objects detected by the radar. A simplified scheme of a radar system is presented in figure 8.2.

From the point of view of this thesis the receiver is the most important part of the radar system. It shall present the following characteristics:

- Having a gain value high enough to amplify the echoes coming from further objects.
- The receiver own noise shall be minimized to avoid masking the weaker echo signals (this is specially important in the first stages of the detection chain according to the quadrupole theory).

Fig. 8.2 Simplified scheme of a radar system

- The system shall minimize the distortion of the echoes at the receiver output respect the ones received at the antenna. This allows improving the accuracy in the determination of the position and size of the detected object.

The superheterodyne receiver is the most usual model in the radar systems. It is normally composed by one or more RF amplifying stages, a mixer that takes the intermediate frequency signal, some intermediate frequency amplifying stages, and some components that filter this signal in order to ease the interpretation of the data. The signal at this point is the one that it is sent to the binary detector. In the binary detector the information is interpreted to check if the signal corresponds to a target or exclusively to noise.

8.3 Methodology: Binary Detection

In a radar system the detector shall establish a threshold value that maximizes the detection of the objects being able to distinguish clearly the presence of purely noisy signals. All the electronic systems present noise perturbation, that in the case of a radar system will be overlapped to the target echo signal normally in the form of Gaussian noise. In this case there will be two possibilities or classes for the signals: one corresponding to the echo of the

electromagnetic wave against an object contaminated by the noise, and one corresponding exclusively to the electronic noise. Depending on the type of signal received the detector shall identify one of the situations.

The criteria used is quite simple, if after the processing of the detector (in this case the output of a neural network) the output is higher than the established threshold then it is considered that the input corresponds to a target, and if the value is lower then it is considered that it corresponds to a noise signal. It is important to study the variations in the results related to the position of the threshold to maximize detection and minimize the false probability alarm in the detection. To find a balance point for the threshold, in practice the experiments fix a value of one of the probabilities, and try to maximize the other one.

Noise is considered as a random process with changing values for its amplitude through the time. Due to this it is not possible to process the noise signal in a deterministic way, and an statistical approach shall be used. When a random signal (noise) is overlapped to a deterministic one (echo) the joint result is converted to a random signal so the statistical approach shall be used too in the situation of having a target.

Mathematically this is expressed as:

- Input corresponding to a echo signal with overlapped noise:

$$r(t) = s(t) + n(t) \quad (8.1)$$

That corresponds to hypothesis H_1 for the detector ($x \in X_1$).

- Input corresponding to only noise:

$$r(t) = n(t) \quad (8.2)$$

That corresponds to hypothesis H_0 for the detector ($x \in X_0$).

To distinguish between the two situation the detector will use the optimal discriminant Bayes function that calculates the *a posteriori* probabilities $P(H_i/x)$ of the input belonging to one of the classes, once that the input is presented and result of the network observed. *A priori* probabilities are also used in the form of pdf functions, so considering one of the

hypotheses as true the pdf function allows the calculation of the probability of the input vector variable taking values in a concrete interval $P(x/H_i)$.

Taking this into account 4 situations are possible:

- Signal detection probability $P(x \in X_1/H_1)$: The system detects a real target.
- Noise detection probability $P(x \in X_0/H_0)$: The system detects noise correctly.
- False alarm probability $P(x \in X_1/H_0)$: The system detects a false target.
- Loss probability $P(x \in X_0/H_1)$: The system is not able to detect a real target.

Once than the network has processed the input and provided a final value the system shall decide which hypothesis is the correct one. For this task the detector selects the highest *a posteriori* probability value and the optimal discriminant function $g_0(x)$ is defined:

$$g_0(x) = \begin{cases} P(H_1/x) - P(H_0/x) > 0 \rightarrow H_1 \\ P(H_1/x) - P(H_0/x) < 0 \rightarrow H_0 \end{cases} \quad (8.3)$$

Considering the mathematical expression of the Bayes theorem:

$$P(A_i/B) = \frac{P(B/A_i)P(A_i)}{P(B)} = \frac{P(B/A_i)P(A_i)}{\sum_i P(B/A_i)P(A_i)} \quad (8.4)$$

That leads to a rewrite of equation 8.3 in the form of a likelihood ratio:

$$\Lambda(x) = \frac{P(x/H_1)}{P(x/H_0)} \quad (8.5)$$

And comparing it with a constant value threshold it is possible to determine the decision regions:

$$\begin{aligned} \Lambda(x) &> \frac{P(H_0)}{P(H_1)} \rightarrow H_1 \\ \Lambda(x) &< \frac{P(H_0)}{P(H_1)} \rightarrow H_0 \end{aligned} \quad (8.6)$$

According to the theory presented by Neyman & Pearson in [210] the system will fix the Probability of False Alarm (PFA) and try to maximize the Probability of Detection (PD). Considering a threshold between the decision regions the statistical theory indicates that for a fixed PFA it is possible to define a value λ that maximizes the PD so:

$$\begin{aligned} \frac{P(x/H_1)}{P(x/H_0)} &> \lambda \rightarrow H_1 \\ \frac{P(x/H_1)}{P(x/H_0)} &< \lambda \rightarrow H_0 \end{aligned} \quad (8.7)$$

Being possible to calculate the concrete threshold value from the limit imposed to the PFA.

8.3.1 Marcum Theoretical Model

Marcum presented the fundamentals of the radar detection in [211], it is based on a simplified model of the pulsed radar that allowed the author to establish a set of values corresponding to the theoretical limits of the detection curves (these curves represent, for a fixed PFA, which is the maximum value for PD depending on the Signal to Noise Ratio (SNR) of the detector input). Then it is possible to check the performance of the classification result by comparison of the theoretical limits with the results of the detector using as inputs the Marcum patterns.

In this experiment the detection threshold is determined by fixing a value for the PFA using a Montecarlo approach. Several synthetic noise patterns are presented to the network and the threshold is decided by selecting the one that satisfies a concrete value for the PFA (calculating this PFA as the number of times that the Montecarlo approach has detected an echo erroneously divided between the total number of signals). Once that those values are fixed then the different echo input data sets, with various SNR values, are presented to the network calculating the PD according to the network results. So for the same detector there are different PD curves depending on the concrete PFA and SNR values used in the calculation.

Marcum theoretical limits are presented in figure 8.3. it can be seen that as the PFA goes smaller (and the threshold goes higher) the PD values are smaller too for the same SNR value.

Fig. 8.3 Theoretical detection curves of the Marcum model

8.4 Data Preparation

The inputs of the network are synthetic patterns generated with a computer program that allows establishing the SNR value of the synthetic data. The system is able to separate the phase and quadrature components of the intermediate frequency signal and using filters it is possible to obtain the real and imaginary parts of the complex envelope of the signal. This envelope is sampled and presented to the network input using N consecutive samples of both components ($2N$ values in total). The output of the network is a scalar value between 0 and 1 that is compared by the threshold determined for each PFA.

During the training phase two balanced data sets (with the same number of echo and noise signal patterns) are needed, one training set and one test set. The first one is composed by the input patterns used in the training algorithm, and the second one is used to evaluate the evolution of the classification error during the training (these two are composed of patterns of $SNR = 13.2dB$ for class H_1). Finally a validation set is used to obtain the detection probability curves (corresponding to different values of SNR). As the computer program is able to generate as much training, test and validation data as needed, then in order to avoid loss of generalization capability in the trained networks no pattern will be repeated between

any of the sets.

8.5 Network Characteristics

To define the training method using the MLP we have to consider that the activation function used in all the neurons of the system is sigmoid logarithmic that responds to the following expression:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (8.8)$$

The graphical representation of this function can be found on figure 8.4. In order to make the training method as much efficient as possible the main idea is to maintain the activation of the different neurons in the linear part of the function and not using the parts where the function is saturated. As can be observed in figure 8.4 it can be considered that the function is outside the saturation regions if the inputs are in the $[-5, +5]$ interval approximately.

Fig. 8.4 Sigmoid logarithmic activation function

The global input to the activation function is the summation of the inputs to the neuron multiplied by the different weights applied to each one of these inputs, so the final expression of the input is $\sigma \sum \omega_i x_i$. The final value of this joint contribution multiplied by a σ factor should remain in the lineal part of the inputs interval, in this way we can consider that the

neuron is not initially saturated and gives useful information to the training (if the neurons are saturated in any of the two possible upper or lower ranges, the output of the neuron does not provide information).

For all the experiments in this thesis we have considered that the input values of the patterns x_i have to be normalized. The different values of each parameter will be divided by the maximum module of all the existing values for this parameter. In this way we can assume that we are working in a hypersphere of radius 1.

For the radar classification experiments the initialization of the weights of the neurons is random but included in an interval $[-0.3, +0.3]$, parameter value σ in the sigmoidal activation function is constant and equal to 1. Doing this so the range of inputs to the activation function $\sigma \sum \omega_i x_i$ will be limited to the interval $[-4.8, +4.8]$. Then the initial part of the training is compliant with the premise of not saturating the output of the neurons.

So finally for this concrete experiment we have:

1. Structure of the network:

- Number of inputs: equal to the number of attributes of the pattern (16).
- Number of hidden layers: 1
- Number of neurons included in the hidden layer: 8 neurons are considered ideal for a tradeoff between the flexibility and the complexity of the system [119].
- Number of neurons in the output layer: 1 to classify in two classes.
- Activation function: Sigmoidal with output included in the interval $(0, 1)$. It is the same for all the artificial neurons.

2. Conditions considered to finalize the network training (depending on the concrete experiment):

- Reach a defined number of inputs presented to the network (3000) to have enough iterations to reach an stable output without overspecializing the network (denoted as $3is_m$).

The concrete metaplasticity values of the different experiments have been determined empirically in order to minimize the classification error.

8.6 Results

Experiments have been carried out in order to evaluate the AMMLP algorithm in comparison with the classic BPA, in a neural detector for the radar signal classification task [13], [119]. Different sets of experiments have been carried out:

- Comparison between the AMMLP and the classic BPA for a large number of input patterns: In this case the objective of the experiment is to check the capabilities of the AMMLP considering the speed of convergence in the classification error. 100 networks with random initial values have been trained using the same data sets and mean values have been calculated. Doing so we eliminate statistically the possible influence of the concrete values for the initial weights and bias.

In this case we will use the modification of the AMMLP that uses *a priori* probability knowledge of the pdf of the signals as a Gaussian distribution, considering the type of noise added to the synthetic signals. The expression can be found on equation 6.10 with the empirically determined values of $A = 35$ and $B = 0.5$. Figure 8.5 shows the error evolution comparison of the network trained with AMMLP and classical BPA training. Error is calculated as the rate of misclassified patterns of the test set out of the total number of patterns. It can be noticed that the AMMLP training algorithm requires much less iterations than the classical BPA (a maximum of 7500 iterations have been set in order to allow the BPA to obtain good classification results), which shortens the total training time.

- Use of *a priori* knowledge of the pdf: In this case equation 6.10 is used to train the networks in order to check the results against the Marcum theoretical limits. In this experiments a total amount of 3000 training iterations have been used with the empirically determined values of $A = 40$ and $B = 0.5$. When Gaussian function is applied as weighting function, the training was conducted as usual BPA training, that is, maintaining the expression of the error function to be minimized from the beginning to the end of the training. Figure 8.6 shows the evolution of the mean classification error for the 100 networks trained.

In order to check the performance of the obtained model it is possible to compare the PD value with the Marcum limits for different values of the PFA. It has been tested

Fig. 8.5 Evolution of the classification error - BPA and AMMLP - Radar input patterns

Fig. 8.6 Evolution of the classification error - Gaussian AMMLP - Radar input patterns

[13] that the results obtained are better if a fixed number of patterns are used instead of stopping the training when a concrete error value is obtained. The results can be found on figures 8.7, 8.8 and 8.9.

Fig. 8.7 Probability of detection $P_{fa} = 10^{-2}$ - Gaussian AMMLP - Radar input patterns

- Use of *a posteriori* knowledge of the pdf: In this case equations 6.8 and 6.9 are used to train the networks in order to check the results against the Marcum theoretical limits. In this experiments a total amount of 3000 training iterations have been used. But in this case, the weighting function is not valid until the output of the network is a sufficiently good approximation of the *a posteriori* probabilities of the inputs. In the first iterations $\hat{y}$ can converge to values very close to zero, and then the training curve evolution freezes around error classification value of 0.1 or 0.2 and the MLP does not learn. So, in this training method, function from equation 6.10 was applied until the error probability achieved a sufficient low value (0.2) and then switched to function based on the output of the network until the end of the training. It is possible to use the classic BPA for the first iterations instead of the Gaussian approach. Figure 8.10 shows the evolution of the mean classification error for the 100 networks trained.

Fig. 8.8 Probability of detection $P_{fa} = 10^{-3}$ - Gaussian AMMLP - Radar input patterns

Fig. 8.9 Probability of detection $P_{fa} = 10^{-4}$ - Gaussian AMMLP - Radar input patterns

Fig. 8.10 Evolution of the classification error - Network output AMMLP - Radar input patterns

In order to check the performance of the obtained model it is possible to compare the PD value with the Marcum limits for different values of the PFA. As in the previous experiment the results obtained are better if a fixed number of patterns are used instead of stopping the training when a concrete error value is obtained. The results can be found on figures 8.11, 8.12 and 8.13.

For this implementation of the AMMLP it is possible to check the evolution of the best performing network. The error probability evolution of the best network obtained is shown in figure 8.14. Only 355 iterations were needed to reach the zero classification error. The network has a rapid error evolution to the zero value, with a low number of iterations. This allows the system to save time and computing resources.

Figures 8.15, 8.16 and 8.17 show the characteristics of trained network for PFA of 10^{-2} , 10^{-3} and 10^{-4} respectively. The distance between two curves is less than 1 dB, although the number of iterations used in the training was very small.

Fig. 8.11 Probability of detection $P_{fa} = 10^{-2}$ - Network output AMMLP - Radar input patterns

Fig. 8.12 Probability of detection $P_{fa} = 10^{-3}$ - Network output AMMLP - Radar input patterns

Fig. 8.13 Probability of detection $P_{fa} = 10^{-4}$ - Network output AMMLP - Radar input patterns

Fig. 8.14 Evolution of the classification error - Best network - Radar input patterns

Fig. 8.15 Probability of detection $P_{fa} = 10^{-2}$ - Best network - Radar input patterns

Fig. 8.16 Probability of detection $P_{fa} = 10^{-3}$ - Best network - Radar input patterns

Fig. 8.17 Probability of detection $P_{fa} = 10^{-4}$ - Best network - Radar input patterns

8.7 Discussion of the Results

The performed experiments take advantage of the MLP implicit estimation of the distribution of classes. It has been detected that, even if the distribution estimations are rough, the proposed artificial metaplasticity strategy can be successfully applied, as expected from the deductions in section 5.5, where many sub-optimum functions can be applied for the purpose of accelerating the training (Gaussian assumption and real output have been successfully tested). An standard Gaussian distribution assumption for the input distributions provides improvements in learning, although it is more interesting to apply this assumption in the first learning iterations, until reaching an empiric threshold after which the implicit estimation carried out by the MLP provides a better weighting function.

The results show that apart from being applied as a learning algorithm, AMMLP weighting can as well be applied as a learning set pre-processing strategy, where instead of taking random samples to train the network, it is possible to study its distribution and construct a training set where less frequent patterns occur more frequently and the occurrence of the more frequent ones is reduced. In the proposed case of AMMLP, when a proper weighting

function is found, the number of patterns needed to train the network reduces drastically.

As illustrated by the best obtained network, it is possible to obtain zero classification error with 355 patterns, while even for the best results with usual BPA it required five times more. The results suggest that the proposed learning algorithm can be of critical significance and usefulness when few training patterns are available, a case very common in many ANNs problems as some of the data sets used in the following experiments.

8.8 State of the Art Study

This section contains a brief state-of-the-art study of the application of artificial intelligence methods to the processing of radar signal.

- In 2000 Al-Nuaimy *et al* [212] addressed the task of locating buried utilities using ground penetrating radar using a neural network classifier, a pattern recognition stage, and additional pre-processing for feature-extraction and image processing stages. A backpropagation neural network was employed to identify portions of the radar image corresponding to target reflections from various types of buried objects. The system was tested on data containing pipes, cables and anti-personnel landmines with results in near real time. Granger *et al* [213] used a fuzzy ART map neural network to classify radar data that is incomplete including a limited number of training cases, missing components and missing class labels. Patnaik & Mishra [214] presented a general study about the possibilities of neural network application on the different fields of microwave engineering including signal processing and recognition of patterns.
- In 2001 Shi & Zhang [215] presented a Gabor atom neural network that used a multilayer feedforward neural network structure. Its input layer constitutes the feature extraction part, whereas the hidden layer and the output layer constitute the signal classification part. From the physics point of view, it was shown that the time-shifted, frequency-modulated, and scaled Gaussian function is adequate for a basic model for the signal of high-resolution radar.
- In 2002 Du *et al* [216] presented a review of the use of neural network methods in various signal-processing scenarios, specially in antenna array signal processing. Shieh & Lin [217] proposed a three-layer vector neural network with a supervised learning

algorithm applied in emitter identification. In the training phase, the values of the three features were presented to the input nodes of the network. A new vector-type backpropagation learning algorithm was derived from an error function defined by the network's actual output and the desired output indicating the correct emitter type of the corresponding feature intervals. The simulated results showed that the proposed algorithm was capable of accelerating the convergence and avoid local minimum.

- In 2003 Lee *et al* [218] presented a natural frequency-based neural network for radar target recognition that took advantage of an aspect angle independence of a natural frequency. Comparison was made with time domain response-based neural networks. McConaghy *et al* [219] used a RBFN to classify real-life audio radar signals that were collected by a ground surveillance radar mounted on a tank. Different signal classification methods were considered like a linear auto-regressive model and nonlinear predictors to model the audio data and then classify the signals according to the prediction errors.
- In 2004 Bruzzone *et al* [220] addressed a system for the classification of multi temporal Synthetic Aperture Radar (SAR) images, developed by integrating an analysis of the multi temporal SAR signal physics with a pattern recognition approach. Karvonen [221] showed a method for segmentation and classification of Baltic Sea ice SAR images, based on Pulse Coupled Neural Networks (PCNN). Also, automated training, based on decomposing the total pixel value distribution into a mixture of class distributions was discussed.
- In 2005 Avci *et al* [222] proposed an intelligent target recognition system for target echo signal of High Resolution Range (HRR) radars. This models dealt with combination of the feature extraction and classification from measured real target echo signal waveforms using X-band pulse radar using a combined approach of two layers: wavelet and MLP. Shaw *et al* [223] focused its study in the use of a neural network approach to automate and facilitate the post-processing of ground penetrating radar results. The radar data was reduced to a simplified data set by using an edge detection routine displaying a hyperbolic image formatted using a MLP network with a single hidden layer used to recognize a simplified hyperbolic shape.
- In 2006 Radoi *et al* [224] addressed the problem of the automatic classification of super-resolution SAR images from an anechoic chamber experiment involving aircraft models. The radar images of these targets were reconstructed using multiple signal

combination and the classification was performed by a self-organizing neural network supervised ART which was compared with an MLP and a fuzzy neural network.

- In 2007 Ting *et al* [225] presented a recognition model which combined four steps such as recognition information's pretreatment, sample data's reduction by rough set, neural network's learning and training and network's recognition to information which is not identified. The model mixed rough set's strong rule extraction ability and the classification ability of neural network through processing initial information and reducing data for the training of the network.
- In 2008 Ren *et al* [226] proposed a method based on mutual information and fuzzy support vector machines for recognizing radar emitter signals. The wavelet ridges and higher-order statistics were used to extract signal features. Mutual information measures was used to reduce the redundant components from the feature vectors set. Rico & Cluckie [227] designed a system to classify weather radar clutter echoes obtained from ground-based dual-polarization weather radar systems. The clutter signals were obtained from ground clutter, sea clutter, and anomalous propagation echoes, which represented sources of error in quantitative radar rainfall estimation.
- In 2009 Ibrahim *et al* [228] presented a method to classify data coming from a Forward Scattering Radar (FSR). The features from the raw radar signal were manually extracted, then the data was classified by and MLP using three variations of the BPA. The proposed architecture was compared with a K nearest neighbor classifier. Sbartaï *et al* [229] addressed a model based on the combination of radar technology and ANN for the non-destructive evaluation of the water and chloride contents of concrete. Input data to the statistical models were extracted from time-domain signals of direct and reflected radar waves.
- In 2010 Park *et al* [230] proposed a target classification system using a 24 GHz microwave radar sensor for the active safety system. The basic idea of this paper was that the pedestrians and the vehicles have different reflection characteristics for a microwave. A MLP was employed to classify the targets and the probabilistic fusion was conducted over time to improve the classification accuracy.
- In 2011 Molchanov *et al* [231] presented an approach to ground moving targets classification by using information features contained in micro-Doppler radar signal. The model was based on using discrete cosine transform coefficients extracted from radar signature as a classification feature and a MLP as a classifier.

- In 2012 Islam *et al* [232] used different artificial intelligence techniques for clutter signals identification in the context of radar based precipitation estimation. The clutter signals considered were extracted from of ground clutter, sea clutter and anomalous propagation whereas the explored techniques included the SVM, the ANN, the decision tree and the nearest neighbor systems.
- In 2013 Montazer *et al* [233] used a technique called fuzzy Optimum Steepest Descent (OSD) in applications, which need real time capabilities for retraining neural networks. The proposed method used fuzzy clustering in order to improve the functionality of the OSD learning algorithm. This improvement was due to initialization of RBF units more precisely using fuzzy C-Means clustering algorithm that resulted in producing a different network response in different retraining attempts. Petrov *et al* [234] evaluated several network topologies, training parameters, input and output coding and data transformation. A large data set of intercepted generic radar signal samples was used for investigating and evaluating the possibilities.
- In 2014 Lin *et al* [235] proposed a classifier which used on-line learning and is attractive for real time applications, such as electronic support measures. A self-organizing interval type-2 fuzzy neural network was proposed for radar emitter identification. Wagner [236] presented a combination of CNNs and SVMs for the automatic recognition of ground radar targets. From the CNN the feature extraction part was used, but instead of the fully connected MLP in the decision stage a SVM was applied. With this combination the idea is to increase the generalization capability of the classifier, while the computation time was kept low.
- In 2015 Ressel *et al* [237] examined the performance of an automated sea ice classification algorithm based on TerraSAR-X ScanSAR data. In the first step of the process chain, gray-level co-occurrence matrix-based texture features were extracted from the image. In the second step, these data were fed into an artificial neural network to classify each pixel. Szymczyk & Szymczyk [238] presented a Laplace Transform Artificial Neural Networks (LTANN), that proposed to use the Laplace transform instead of ordinary weights and a linear activation function for the artificial neurons.
- In 2016 Ding *et al* [239] proposed to improve the performance of SAR target recognition using an ANN combined with the data of invariance under target translation, the invariance under speckle variation in different observations, and the tolerance of pose missing in usual training data. Gong *et al* [240] presented a change detection

approach for SAR images based on deep learning. The learning algorithm for deep architectures included unsupervised feature learning and supervised fine-tuning to complete classification.

Chapter 9

Experiment 2: Breast Cancer Data Classification

9.1 Introduction

The second experiment included in this thesis applies the theoretical hypotheses presented in sections 5.5.1 and 5.5.2 considering the application of AMP in supervised and unsupervised paradigms to a practical example with real data. One of the biggest problem of nowadays society, public administration and medical community is the early detection of cancer disease. One of the most important variant of these problems is the breast cancer that affects millions of people around the world and that can be treated with successful results with an early detection. In order to facilitate this task ANNs can be applied to classify the data of the medical samples into malign or benign categories.

In these experiments two different approaches have been applied: a AMMLP structure and a KLN structure. This two networks have been used to classify the patterns available in the Wisconsin Breast Cancer Database (WBCD). Results of these two implementations of the AMP will be compared with classic BPA and other state-of-the-art algorithms.

For the AMMLP the first step is to study the influence of the parameters' values used in the nominal BPA for the MLP to be sure that the results obtained with the modifications of the AMMLP method are compared with the best performing network using a classic approach. The following experiments use *a priori* and *a posteriori* probability estimations of the input distribution inside the MLP and AMMLP theory to implement the metaplasticity

learning algorithm in order to improve the results of the basic training method.

Considering the KLN, this structure model has shown big potential for synthetic inputs [193], [194] and the idea is to test if its results can be competitive using real data in comparison with other supervised and unsupervised techniques.

The results of these experiments have been published in [15], [121] and [122].

9.2 WBCD Dataset

Breast cancer is a malignant tumor that develops from breast cells. Although research has identified some of the risk factors (i.e., aging, genetic risk factors, family history, menstrual periods, not having children and obesity) that increase a woman's chance of developing breast cancer, the inherent cause of most breast cancers remains unknown [241], [242]. Further than this, even the question of how many of these risk factors cause cells to become cancerous remains without response, even important discoveries are being made about this topic.

The correct pattern classification of breast cancer is an important worldwide medical problem. Cancer is one of the major causes of mortality around the world and research into cancer diagnosis and treatment has become an important issue for the scientific community. If the cancerous cells are detected before they spread to other organs, the survival rate is greater than 97% [243]. A major class of problems in medical science involves disease diagnosis based on various tests performed on patients. For this reason, the use of classifier systems in medical diagnosis is increasing. Artificial intelligence classification techniques can enhance current research.

This study analyzed the Wisconsin Breast Cancer Database [244]. This data base has been used several times in the literature and many high impact studies have used these inputs for classification including systems based in ANN, SVM and neuro-fuzzy techniques among others, so is very useful in order to compare the results with the state-of-the-art. This data base contains 699 patterns, each of these patterns is composed by 9 numerical attributes that correspond to different physical characteristics that can be considered as markers of the possible presence of cancer in the sample.

Table 9.1 Description of the attributes of the WBCD

Attribute Number	Attribute Description	Attribute Values	Mean Value	Variance Value
1	Clump thickness	1-10	4.44	2.82
2	Uniformity of cell size	1-10	3.15	3.07
3	Uniformity of cell shape	1-10	3.22	2.99
4	Marginal adhesion	1-10	2.83	2.86
5	Single epithelial cell size	1-10	2.23	2.22
6	Bare nuclei	1-10	3.54	3.64
7	Bland chromatin	1-10	3.45	2.45
8	Normal nucleoli	1-10	2.87	3.05
9	Mitoses	1-10	1.60	1.73

9.2.1 Data Preparation

Numerically the attributes have been evaluated manually by an expert with values between 1 and 10, being value 1 the closest to an indicator of a benign nature of the sample and value 10 the closest to an indicator of a malign nature of the sample. Beside the attributes in the database there is an additional field that indicates the final diagnosis of the real nature of the sample. This value will be used as the ideal output of the network to be considered during the training or in the performance evaluation.

The original data base contains 16 samples whose attributes are not completely filled. In order to work with a homogeneous set of patterns with all the numerical attributes filled in, these incomplete elements have been eliminated from the experiment. Finally we will use 683 patterns that are divided in 444 benign samples (65%) and 239 malign samples (35%). The table 9.1 includes the description of the attributes of the WBCD.

Normally the classifiers based on neural networks produce better results if the training sets are balanced, presenting the same number of patterns belonging to each one of the possible classes. In order to achieve this situation in the creation of the sets used to train and to evaluate the system some malign patterns will be repeated instead of eliminating some benign patterns to get these balanced sets. It has been considered better to duplicate a small number of malign elements as inputs for the networks instead of losing the potential information present in some of the benign elements. The repetition of these elements is not considered excessive so there will be no overspecialization in the statistical values that can affect the generic pattern classification.

Depending on the concrete inputs used for training and for performance evaluation it is possible to have a numerical influence on the results. To obtain results statistically independent of the distribution of the patterns a *10 fold cross validation* evaluation method has been considered. Using this method the possible dependence of the results with the distribution of the samples in the training or performance evaluation sets is eliminated: all the samples are used to train the networks and all the samples are used to evaluate the performance of the results. This is applied to different executions of the experiment for the same initial neural networks, and finally mean values are calculated to establish the final performance results.

For this experiment we have created ten data sets from the WBCD with the following distribution of patterns:

- G1: 90 total patterns: 45 benign and 45 malign
- G2: 90 total patterns: 45 benign and 45 malign
- G3: 90 total patterns: 45 benign and 45 malign
- G4: 88 total patterns: 44 benign and 44 malign
- G5: 88 total patterns: 44 benign and 44 malign
- G6: 88 total patterns: 44 benign and 44 malign
- G7: 88 total patterns: 44 benign and 44 malign
- G8: 88 total patterns: 44 benign and 44 malign
- G9: 88 total patterns: 44 benign and 44 malign
- G10: 90 total patterns: 45 benign and 45 malign

Using these 10 initial sets we will create 10 different data groups. The training data groups that will be used as inputs to the networks for training the system and evaluating the evolution of the error will consist in the union of 9 of the previous 10 sets. The final evaluation that calculates the performance of the performance of the network will use the other initial set. The 10 folders will be created with the variation of the initial set that is used for evaluation and not for training.

The networks are trained from the same initial aleatory weights presenting the data corresponding to each of the 10 final folders created from the initial sets. Finally the mean values of the results will be calculated to eliminate the possible statistical influence in the results due to the concrete fixed selection of some patterns to train the system and the fixed selection of other patterns to evaluate the results.

9.3 AMMLP Applied to WBCD Classification

Based on the theory exposed in previous chapters an AMMLP model is applied for the classification of the WBCD data set. For these experiments a MLP neural network have been used with a input composed by 9 attributes contained in each single pattern, a hidden layer composed by 8 neurons (considering that experiments presented in previous chapter proved that 8 neurons is enough to get the flexibility needed), and an output layer with just 1 neuron (result of the classification for 2 classes).

The BPA presents some limitations and problems during the MLP training as seen in section 3.4. The AMMLP aims to improve BPA by including a variable learning rate $\eta(x) = \frac{\eta}{f_x^*}$ in the training phase affecting the weights in each iteration step based on an estimation of the inherent probability distribution of training patterns.

For the WBCD classification experiment the activation function used in all the neurons of the system is sigmoidal, input patterns set normalized and the initialization of the weights of the neurons is random but included in an interval $[-0.5, +0.5]$, parameter value σ in the sigmoidal activation function is constant and equal to 1. Doing this so the range of inputs to the activation function $\sigma \sum \omega_i x_i$ will be limited to the interval $[-4.5, +4.5]$. Then the initial part of the training is compliant with the premise of not saturating the output of the neurons.

As a summary of the experiment conditions:

1. Structure of the network:

- Number of inputs: equal to the number of attributes of the pattern (9).
- Number of hidden layers: 1.

Table 9.2 Confusion matrix model

True Positive $H(1/1)$	False Positive $H(1/0)$
False Negative $H(0/1)$	True Negative $H(0/0)$

- Number of neurons included in the hidden layer: Based on previous experience 8 neurons are considered adequate for a tradeoff between the flexibility in the definition of the decision regions and the complexity of the system.
- Number of neurons in the output layer: 1 to classify in two classes.
- Activation function: Sigmoidal with output included in the interval $(0, 1)$.

2. Conditions considered to finalize the network training:

- Reach a defined number of input patterns (2000) presented to the network to have enough iterations to reach a stable output without overspecializing the network.

9.3.1 Evaluation Method

In each one of the experiments 50 networks have been considered and trained. Using the 10 fold cross validation method the results are not dependent of the concrete patterns used for training and for performance evaluation. Using 50 different initial networks and calculating mean values we assure that the results are independent of the initial random values for the weights and bias established in the creation of the networks.

The following hypothesis are defined:

- True Positive (TP) $H(1/1)$: The pattern is malign and has been classified as malign.
- False Positive (FP) $H(1/0)$: The pattern is benign and has been classified as malign.
- False Negative (FN) $H(0/1)$: The pattern is malign and has been classified as benign.
- True Negative (TN) $H(0/0)$: The pattern is benign and has been classified as benign.

Using these hypotheses it is possible to build the confusion matrix in table 9.2.

To evaluate the performance of the classifiers the following measures are used and defined as follows:

- Sensitivity: Defined as the capability of correctly detecting malign patterns:

$$Sensitivity = \frac{TP}{TP + FN}(\%) \quad (9.1)$$

- Accuracy: Defined as the capability of correctly classifying both kind of patterns:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}(\%) \quad (9.2)$$

- Receiver Operating Characteristic (ROC): It is a two dimension measure that is widely used to check the results of diagnostic test [245], [246], [247]. It is a graphic of sensitivity versus the complementary value of the specificity that represents the fraction of true positives classifications versus the fraction of false positive classifications based on a calculation made using a threshold for the system's output. In a ROC curve FP is represented in x axis and TP in y axis. For the generation of the curve the classification probabilities obtained in the different tests are used jointly with the real label of each input patter [247].

Associated to the ROC we can measure the Area Under the Curve (AUC) as a way to calculate the exactness of the classifying tests. If the ROC curve elevates quickly to the right upper corner of the graphic then the AUC will be close to 1 and it can be considered that the test is reliable, but if the value is near to 0.5 then the system is not performing well [246]. The best way to calculate the AUC is using an approximation based on the sum of areas of rectangles with differential dimensions.

50 initial networks have been trained and evaluated with the 10 folder cross validation algorithm. From the results obtained for the same network with each one of the folders the mean confusion matrix is obtained for each network. Once these 50 mean values are calculated an additional calculation is made and the final mean value is obtained as the definitive result of the experiment.

The most important figure in these experiments in the sensitivity (considered as TP percentage). These is due to the intrinsic nature of the experiment (it is much more important to detect all the malign patterns than classifying as malign a benign input).

Table 9.3 Confusion matrix $\eta = 1$ - Nominal BPA - WBCD

95.5316 %	2.5720 %
4.4684 %	97.4280 %

9.3.2 Results

Three sets of cases are included in the global WBCD classification experiment:

- Nominal BPA: In this first part of the training the behavior of the nominal BPA is studied, maintaining the learning rate η inside the theoretical limits $(0, 1]$. The objective is to be able to compare the results of the AMMLP with the best possible implementation of the BPA. The tests performed indicate that the best results obtained for accuracy and sensitivity corresponds to value $\eta = 1$:

- Accuracy = 96.4797%
- Sensitivity = 95.5315%

The confusion matrix for this learning rate is included in table 9.3.

The evolution of the classification error during the training phase is shown in figure 9.1. The evolution of the error in the first iterations of the training is included in more detail in figure 9.2.

The ROC of this experiment can be found on figure 9.3 and the AUC value calculated is 0.9946.

- Use of *a priori* knowledge of the pdf: In this case a Gaussian approach for the AMMLP is used to train the networks in order to check the results against the classic BPA and other AMMLP approximation using the output of the network. This experiment will be divided in two parts, the initial part of the training will use the classic BPA until classification error reach 0.3, after that the second and principal part of the training will use equation 6.10 from error 0.3 to the finalization of the training. With this methodology we can assure that the results obtained can be compared with the ones presented in the following section of the experiment (the following experiment needs the network to have started learning in order to assure the theoretical conditions for the

Fig. 9.1 Evolution of the classification error $\eta = 1$ - Nominal BPA - WBCD

Fig. 9.2 Detail of the first iterations of the evolution of the classification error $\eta = 1$ - Nominal BPA - WBCD

Fig. 9.3 ROC $\eta = 1$ - Nominal BPA - WBCDTable 9.4 Confusion matrix $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD

99.2162 %	4.0540 %
0.7838 %	95.9460 %

method).

The Gaussian function has two parameters to be studied, and the experimental results indicate that the best performance is obtained for both accuracy and sensitivity with values $A = 10$ and $B = 0.45$:

- Accuracy = 97.5811%
- Sensitivity = 99.2162%

The confusion matrix for this AMMLP approach is included in table 9.4.

The evolution of the classification error during the training phase can be observed in figure 9.4. A closer view of the evolution of the classification error during the first

steps of the training is shown in figure 9.5.

Fig. 9.4 Evolution of the classification error $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD

The ROC of this experiment can be found on figure 9.6 and AUC calculated is 0.9949.

- Use of *a posteriori* knowledge of the pdf: In this case the output of the network is used to implement the AMMLP. This experiment is also be divided in two parts in order to have comparable conditions with the previous one. The first part of the training will use the classic BPA until classification error reach 0.3, after that the second and principal part of the training will use equations 6.8 and 6.9 from error 0.3 to the finalization of the training. Doing so it is possible to use the output of the network as a pdf estimation considering that the network is not using completely random weights and bias values.

The empirical study of the results leads to a value of $\eta = 23$. The associated classification results are:

- Accuracy = 97.8716%
- Sensitivity = 98.8784%

Fig. 9.5 Detail of the first iterations of the evolution of the classification error $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD

Fig. 9.6 ROC $A = 10$ $B = 0.45$ - Gaussian AMMLP - WBCD

Table 9.5 Confusion matrix $\eta = 23$ - Network output AMMLP - WBCD

98.8784 %	3.1352 %
1.1216 %	96.8648 %

The confusion matrix for this AMMLP approach is included in table 9.5.

The evolution of the classification error during the training phase is shown in figure 9.7 and the detailed evolution of the error curve in the first iterations of the training can be found in figure 9.8.

Fig. 9.7 Evolution of the classification error $\eta = 23$ - Network output AMMLP - WBCD

The ROC of this experiment can be found on figure 9.9 and AUC calculated is 0.9949.

9.3.3 Discussion of the Results

The two AMMLP experiments have produced better results both for accuracy and sensitivity. Considering the use of 2000 patterns the best sensitivity result is obtained in the Gaussian experiment (99,2 %) and the best accuracy result is obtained in the output as probability

Fig. 9.8 Detail of the first iterations of the evolution of the classification error $\eta = 23$ - Network output AMMLP - WBCD

Fig. 9.9 ROC $\eta = 23$ - Network output AMMLP - WBCD

estimation experiment (97,8 %).

If we compare the results obtained in the nominal BPA $\eta = 1$ experiment with the one consisting in the use of the output as probability estimation we can observe a considerable improvement in the quality of the results (for both accuracy and sensitivity). Then we can confirm the theory that states that once the network has started to learn, the output can be considered as a reliable estimation of the probability of the input patterns distribution.

Considering the evolution of the classification error in the different experiments it is possible to see a considerable quicker learning (measured in iterations to achieve a concrete value for the error) for the AMMLP based on the output of the network compared with classic BPA. In addition the classic BPA and the Gaussian AMMLP present a lot of peaks in the error evolution meanwhile the AMMLP based on the network's output is more smooth and natural. In order to obtain a determined value for this error using a limited amount of input patterns the best option is to use the output of the network as the modifier metaplasticity function. This possibility can be very useful in applications where the number of available patterns to be classified is very small and we want to avoid overspecialization in the training process.

9.4 KLN Applied to WBCD Classification

The KLN structure is applied to check the capabilities of this model in the classification of data from a relevant standard real application. In this experiment 50 different initial networks have been tested. All the models used in this study were trained and tested with the same data and validated using the 10-folder cross-validation.

Considering the concrete experiment characteristics:

1. Structure of the network:

- Number of inputs: This KLN has 9 neurons in its input layer corresponding to the number of elements that form each input pattern.
- Number of *TC* neurons: There are 9 neurons in the *TC* layer (similar number as input layer to be coherent with the KLN structure).

- Number of S neurons: There are 2 neurons in the S layer as two classes are considered in the experiment.
- Number of B neurons: There 2 neurons in the upper B layer associated to the S output neurons.
- Activation function: Sigmoidal according to equations 7.4 and 7.6.

2. Conditions considered to finalize the network training:

- Reach a defined number of epochs in the network training, the number of epochs will vary during the experiment.

Once the input is fed to the network, its activation is propagated until all layers are activated. At the end of the process one of the neurons presents a higher output than the other one, so it is possible to classify the patterns taking into account which output is activated and which one is inhibited. The WTA process occurs naturally as an emergent consequence of the individual computation of each neuron without the need of externally monitoring the network.

It is very important to remark that this first prototype of the network applied to the WBCD classification has demonstrated to be extremely sensitive to the concrete values of the training mathematical parameters. Even minimal deviations from the values used in this experiment can cause a dramatic worsening of the results or even a non convergence of the learning algorithm, so we cannot consider this prototype as a robust implementation of the mathematical theory. In this case the values established for the parameters have been set using a Montecarlo approach with many simulations until adequate results have been obtained demonstrating that for very concrete values the network is able to learn without external supervision. The final values used in this simulation are $v = 0.025$, $\xi = 0.001$, compression factor for the curve $k = 25$, the initial sigmoid shift $s = 0.5$, initial weights from TC to S neurons are negligible and random, and non-modifiable weights are set to $W_{S_S} = 0.85$, $W_{S_B} = 0.98$, $W_{I_{TC}} = 1.0$ and $W_{B_S} = 0.5$.

9.4.1 Evaluation Method

The concepts of confusion matrix, sensitivity and accuracy presented in section 9.3.1 are applicable to this KLN experiment. As in previous experiment in each one of the variations of the experiment 50 networks have been trained using the 10 folder cross validation method to obtain results independent of the concrete patterns used for training and for performance

Table 9.6 Sensitivity and accuracy evolution depending on the number of epochs - KLN - WBCD

Epochs	Sensitivity	Accuracy
1	61.92%	58.33%
5	73.76%	61.36%
10	76.04%	64.77%
50	85.79%	77.97%
100	89.97%	83.69%
250	96.70%	92.51%
500	98.72%	95.57%
1000	99.94%	96.91%

Table 9.7 Confusion matrix 1000 epochs - KLN - WBCD

99.94%	6.12%
0.06%	93.88%

evaluation. After that the mean values are calculated to assure that the results are independent of the initial random values in the creation of the networks. This is done for all the performance figures to obtain the final results of the experiment.

9.4.2 Results

Several variations of the experiment have been performed with different number of epochs in each experiment (considering one epoch like presenting the full group of input patterns once to the network). The output of the network is integrated by two neurons, depending on which one presents the higher level at the output we have considered that one of the classes (benign or malign sample) is selected by the network. Table 9.6 presents the results obtained.

The best results obtained correspond to 1000 epochs where almost all the malign patterns are correctly recognized. The associated classification results are:

- Accuracy = 96.91%
- Sensitivity = 99.94%

Confusion matrix for this variation of the experiment is shown in table 9.7.

9.4.3 Discussion of the Results

The first point to be considered is the concrete application of the network as a cancer pattern classifier. In this concrete application the most important figure is sensitivity because we want to detect all the malign patterns above all. The objective of the network as a classifier is to reduce to the minimal the false negative results although this can cause a worsening in the figure corresponding to false positive classifications. This real data is presented to the KLN model in order to check if the good results obtained with synthetic data [193], [194] are similar in this case. It can be seen that the network is able to classify correctly this data set.

The best sensitivity is obtained with the higher number of epochs in the training. Almost all the patterns corresponding to the malign inputs are correctly classified. On the contrary several benign patterns are misclassified, in this particular application is not considered a problem as the sensitivity and not accuracy is the driver figure. The evolution of the sensitivity shows that even one epoch is sufficient to start learning, the results improve highly when repeating the inputs.

Comparing the KLN unsupervised results, they improve the results of advanced supervised methods as those presented in section 9.3, so KLN performs a deeper learning of the information in the data set, without the need of a deep network.

9.5 State of the Art Study

This section contains information about different studies performed about the application of artificial intelligence methods in the classification of the WBCD patterns used in this experiment.

- In 1996 Quinlan [248] modified the C4.5 algorithm to get a better adaptation to continuous attributes by modifying the formation and evaluation of the data sets. The trials showed that this modification led to smaller decision trees with higher predictive accuracies. Hamilton [249] presented the Rule Induction through Approximate Classification (RIAC) method for inducing rules from examples based on the theory of rough sets. With this method boundary and negative regions of the targets were defined using statistical information, meanwhile superfluous attributes were eliminated and general rules generated. Ster & Dobnikar [250] evaluated the performance of

different methods in the same conditions in order to obtain a reliable comparison of the performance of the algorithms over some standard data sets.

- In 1999 Nauck & Kruse [251] applied a neuro-fuzzy classification approach to the WBCD. The authors presented interactive strategies for pruning rules and variables in order to be able to interpret the system linguistically so if the classifier is used to derive a diagnosis for some patient, both the physician and the patient obtain the information why the diagnosis was selected and how clear the result was. Peña-Reyes & Sipper [252] presented a combination of fuzzy system and evolutionary algorithms to be applied to WBCD and other diagnostic systems. The results involved a few simple set of rules making the system human interpretable.
- In 2000 Setiono [253] proposed a variation on the classification method for the WBCD based in a pre-processing of the data. This task involved selecting some relevant attributes among all the available ones, and using this reduced set to extract the rules used in the classification.
- In 2002 Albrecht *et al* [254] combined logarithmic simulated annealing with a MLP to process the WBCD data set. The algorithm used an inhomogeneous Markov chain, whereas in most applications of simulated annealing homogeneous Markov chains are used as the underlying model. For inhomogeneous Markov chains the convergence to optimum solutions was provided under some conditions on the convergence space, thus, this implementations emulated a true approximation process to optimum solutions.
- In 2003 Abonyi & Szeifert [255] proposed a fuzzy model structure where each rule can represent more than one classes with different probabilities. The obtained classifier could be considered as an extension of the quadratic Bayes classifier that utilized mixture of models for estimating the class conditional densities.
- In 2007 Ubeyli [256] checked the classification accuracies of different classifiers, considering a MLP, a Combined Neural Network (CNN), a Probabilistic Neural Network (PNN), a Recurrent Neural Network (RNN) and a SVM. In this research the SVM achieved diagnostic accuracies higher than that of the other automated diagnostic systems. Polat & Günes [257] used a Least Square Support Vector Machine (LS-SVM) for the diagnosis of the WBCD data set. The robustness of the LS-SVM was examined using classification accuracy, analysis of sensitivity and specificity, k-fold cross-validation method and confusion matrix. Guijarro-Berdias *et al* [258]

applied a learning algorithm based on linear-least-squares. Results presented showed that the algorithm upgrades the learning speed of several backpropagation algorithms, while preserving good optimization accuracy. Due to its performance results and computational cost it was considered for large networks and training sets.

- In 2009 Akay [259] presented a model based in SVM with a grid search to optimize the model parameters. The input characteristics selected from the data set were chosen applying F-score statistic calculation. Karabatak & Ince [260] proposed an automatic diagnosis system based on association rules and ANN. Association rules were used for reducing the dimension of breast cancer database and ANN was used for intelligent classification, the dimension of input feature space was reduced from nine to four by using this method.
- In 2010 Peng *et al* [261] presented a technique to reduce the dimensionality of the data set. This technique was based in filter integration and envelope methods inside a sequential search to improve the performance of the classification. Conforti & Guido [262] generated a optimum kernel function using Semi-Defined Programming (SDP) over a SVM structure. This SDP presented high computational load and time when the data sets increase their size.
- In 2011 Paulin & Santhakumaran [263] checked several algorithms over the WBCD. The training algorithms used are gradient descent with momentum and adaptive learning, resilient back propagation, Quasi-Newton and Levenberg-Marquardt. The performances of these four algorithms were compared with the standard steepest descent BPA. Chen *et al* [264] presented a Rough Set (RS) based SVM classifier applied to breast cancer diagnosis. In the proposed method, RS reduction algorithm was employed as a feature selection tool to remove the redundant features and further improve the diagnostic accuracy by SVM. Fallahi & Jafari [265] presented an automatic system for detection of breast cancer using data pre-processing and Bayesian network. In this study, pre-processing algorithm was used for reducing the dimension of breast cancer database then a pre-processing was done on the data and ultimately Bayesian network classifier is used for classification.
- In 2012 Raad *et al* [266] presented a comparison of the results obtained for the classification of WBCD data set between the application of MLP and RBF with momentum and a variable learning rate. The RBF offered better classification results than the MLP.

- In 2013 Inan *et al* [267] proposed a hybrid feature selection method, using association rules (*A priori* algorithm) and Principal Component Analysis (PCA) together with ANN classifier. This hybrid system, decreased the size of data and improved the training speed compared with classic methods. Shajahaan *et al* [268] analyzed the applicability of decision trees to predict the presence of breast cancer and compared the result with other supervised learning algorithms.
- In 2014 Utomo *et al* [269] implemented ANN with extreme learning techniques for diagnosing the WBCD data set. Results showed that ELM ANN was a better generalization classifier model than BPA. Meanwhile BPA needed to tuning several parameters, ELM ANN was a simple tuning free algorithm where the only one to be defined was the number of hidden nodes. Additionally BPA worked only for differentiable activation functions in hidden and output nodes while ELM ANN could use both differentiable and undifferentiable activation functions.
- In 2015 Bhardwaj & Tiwari [270] proposed a Genetically Optimized Neural Network (GONN) algorithm, for solving classification problems. The algorithm evolved a neural network genetically to optimize its architecture (structure and weight) for classification but introducing new crossover and mutation operators which differ from standard crossover and mutation operators to reduce the destructive nature of these operators. Onan [271] proposed a classification model consisting of three phases: instance selection, feature selection and classification. In instance selection, the fuzzy-rough instance selection method based on weak gamma evaluator was utilized to remove useless or erroneous instances. In feature selection, the consistency-based feature selection method was used in conjunction with a re-ranking algorithm. In the classification phase of the model, the fuzzy-rough nearest neighbor algorithm was utilized.
- In 2016 Abdel-Zaher & Eldeib [272] presented an scheme for classification of the WBCD data set developed using a DBN unsupervised path followed by classic BPA supervised method. The construction was a BPA based ANN with Liebenberg-Marquardt learning function while weights were initialized from DBN path.

Table 9.8 includes the name of the cited researchers, the algorithm or method used in the study and the values for sensitivity (if available) and accuracy obtained.

Table 9.8 State of the art study for WBCD classification

Author (Year)	Method	Sensitivity (%)	Accuracy (%)
Quinlan (1996) [248]	C4.5	-	94.74%
Hamilton (1996) [249]	RIAC	-	94.99%
Ster (1996) [250]	Fisher	-	96.80%
Nauck (1999) [251]	NEFCLASS	-	95.06%
Peña-Reyes (1999) [252]	Fuzzy-GA	-	97.07%
Setiono (2000) [253]	NeuroRule-rule 3	-	98.24%
Albrecht (2002) [254]	LSA	98.80%	98.80%
Abonyi (2003) [255]	Supervised fuzzy clustering	-	95.57%
Ubeyli (2007) [256]	SVM	99.37%	99.54%
Polat (2007) [257]	LS-SVM	-	98.53%
Polat (2007) [257]	LS-SVM	-	98.53%
Guijarro-Berdias (2007) [258]	LS-MLP	-	96.00%
Akay (2009) [259]	FS-SVM	-	99.51%
Karabatak (2009) [260]	AR+NN	-	95.60%
Peng (2010) [261]	CFW	-	0.997 (AUC)
Conforti (2010) [262]	SVM-SDP	-	96.79%
Paulin (2011) [263]	LM	-	99.28%
Chen (2011) [264]	RS-SVM	100.00%	96.87%
Fallahi (2011) [265]	Bayesian	-	98.10%
Raad (2012) [266]	RBF	-	97.00%
Inan (2013) [267]	AP+NN	-	98.29%
Shajahaan (2013) [268]	Random Tree	100.00%	100.00%
Utomo (2014) [269]	ELM ANN	94.80%	96.40%
Bhardwaj (2015) [270]	GONN	-	99.26%
Onan (2015) [271]	Fuzzy-R NN	-	99.71%
Abdel-Zaher (2016) [272]	DBN	100.00%	99.68%
This study	Gaussian AMMLP	99.21 %	97.58 %
This study	Network Output AMMLP	98.87 %	97.87 %
This study	KLN	99.94 %	96.91 %

Chapter 10

Experiment 3: Credit Scoring Data Classification

10.1 Introduction

The third experiment included in this thesis continues with the objectives presented in the previous experiment. In this case it is considered applying the theory exposed in this thesis to other practical example with real data. To tackle the complex problem of providing Business Intelligence (BI) solutions based on Business Data (BD), bioinspired machine learning is considered. As a case study both AMMLP and KLN networks are applied to the automation of credit approval decision based on collected client data analyzed, showing its potential and improvements over the state-of-the art techniques.

In this experiments two different approaches have been applied: a AMMLP structure and a KLN structure. This two networks have been used to classify the patterns available in the Australian Credit Approval Database (ACAD). Results of these two implementations of the AMP will be compared with classic and state-of-the-art classification methods applied to the same data set.

The steps followed in this experiment are similar than from the classification of breast cancer data set. For the AMMLP the first step is to study the influence of the parameters' values used in the nominal BPA applied to a MLP to be sure that the results obtained with the AMMLP modification of the method are compared with the best performing unaltered approach of backpropagation. After that *a priori* and *a posteriori* probability estimations of the input distribution are applied by the AMMLP theory to implement the metaplasticity

learning paradigm.

The KLN structure is tested to check if the prototype model can be adapted to several problems (different than the WBCD classification) in a competitive way in comparison with other classification methods.

The results of these experiments have been published in [124].

10.2 ACAD Dataset

BD solutions for business services and utilities is one of the key issues in the development of nowadays Industry 4.0 revolution. System networks are in the need of new powerful intelligent systems to successfully process all the available information from the different users to study their behavior and improve service and associated business. New interactive Information and Communication Technology (ICT) infrastructures are being developed to support the optimal exploitation of the business results in different sectors. BD processing empower the associated business and creates opportunities up and down the industry value chain. These new utilities require management of services and technologies that span ICT systems.

Technology solutions introduce new technical and organizational concepts. Companies are switching into computer driven efficient management of complex information [273], [274], that interprets the information of the different actors, improving the efficiency, reliability and economy. Business processes will need to be established to accommodate these technology and market evolutions. These applications require distribution of data and a computing platform [274]. A business model aligns the utility goals by creating rate-based opportunities and reducing expenses and risks [275].

BI [276] is a way to collect, correlate, and analyze events from multiple sources, leveraging existing tools while gaining end-to-end management of the entire network. BI is the practice of interpreting and visualizing data to make useful business-oriented decisions. BI decision support applications facilitate many such multi-dimensional analysis like online analytical processing, click-stream analysis, balance score-card, preparation, visualization, querying, reporting, and charting, data mining for text content and voice, forecasting, geo-spatial analysis, enterprise portal implementation, knowledge management, digital dashboard

access and other cross functional activities. BI systems are well recognized as contributing to decision making. These system are most commonly identified as technological solutions holding quality information in well-designed data stores, enabling them to make the right decisions or take the right actions.

BD and BI systems complexity make their evaluation difficult. Analyzing them as a whole [277], because of its complex and multidisciplinary nature, it is very difficult to become to classical engineering solutions that assess the overall success. Here is where bioinspired ANNs that implement adaptive, reactive, complex and distributed processing may provide technological solutions. They have the potential to provide a more efficient and effective use of explicit and implicit knowledge in BD.

The financial crisis has put the focus of investigation in banking issues, specially in the ones related to the approval of credits. Until recently, the decisions related to credit loans were based on individual perceptions and the human capacity to assess the risk. The growing demand for credit has led to the use of a statistical method, known as credit scoring, to decide whether to or not to grant credit. The method of punctuation credit [278] is widely used for consumer loans, and it is getting more used to commercial loans. The credit score is a binary classification task of basic finance. A punctuation advantage of the credit score method is the reduction of the costs of credit analysis: faster credit decisions, greater control and reduce potential risks.

The ACAD [279] data set has been selected for this business intelligence application in order to compare the results of automatic decision method with the manual one. The database contains 690 cases with 15 attributes whose evaluation lead to the approval or rejection of a credit petition.

10.2.1 Data Preparation

The database contains 690 cases, divided into two classes, 307 applicants "accepted" (45%) and 383 applicants "rejected" (55%). Each applicant has 15 features, including 6 nominal, 8 numeric attributes and one label of each class (accepted or rejected). This final value will be used as the ideal output of the networks that will be checked during the training or in the performance evaluation. This set of data is interesting because there is a good mix of attributes: continuous and nominal, nominal ratings with small and large values. Another

Table 10.1 Description of the continuous type attributes of the ACAD

Attribute Number	Attribute Description	Attribute Values	Mean Value
2	Age	13.75-80.25	31.57
3	Debt	0-28	2.750
8	Years employed	0-28.5	2.225
11	Credit score	0-67	2.402
15	Income	0-100000	1019

important characteristic is that few values are missing. The table 10.1 includes the description of the continuous type attributes of the ACAD.

As in the previous experiment it is considered that the classifier will provide better results if the training sets are balanced, presenting the same number of patterns belonging to each one of the possible classes. Considering this an adaptation of the data set is needed so some accepted patterns have been repeated instead of eliminating rejected patterns to avoid a potential loss of the information included in the eliminated patterns. The repetition of these elements is small due to the difference of number of input from each class so there will be no overspecialization in the statistical values for generic pattern classification.

To avoid having results influenced by the concrete selection of patterns for training and performance evaluation the *10 fold cross validation* evaluation method has been considered as in WBCD experiment. All the samples are used to train the networks and all the samples are used to evaluate the performance of the results in different executions of the experiment. The same initial neural networks are used and mean values of the results are calculated to determine the final performance of the system.

For this experiment we have created ten data sets from the ACAD with the following distribution of patterns:

- G1: 78 total patterns: 39 rejected and 39 accepted
- G2: 78 total patterns: 39 rejected and 39 accepted
- G3: 78 total patterns: 39 rejected and 39 accepted
- G4: 76 total patterns: 38 rejected and 38 accepted

- G5: 76 total patterns: 38 rejected and 38 accepted
- G6: 76 total patterns: 38 rejected and 38 accepted
- G7: 76 total patterns: 38 rejected and 38 accepted
- G8: 76 total patterns: 38 rejected and 38 accepted
- G9: 76 total patterns: 38 rejected and 38 accepted
- G10: 76 total patterns: 38 rejected and 38 accepted

The method to create the 10 folders is explained in section 9.2.1. The networks are trained from the same initial aleatory weights and bias values presenting the data corresponding to each of the 10 folders. Finally the mean values of the results will be calculated to eliminate the possible statistical influence in the results due to a fixed selection of the patterns to train the system and the fixed selection of the patterns to evaluate the performance.

10.3 AMMLP Applied to ACAD Classification

An AMMLP model approach is applied for the classification of the ACAD data set. For these experiments a MLP neural network have been used with a input composed by 14 attributes as the input patterns, a hidden layer composed by 8 neurons (considering that experiments presented in previous chapter proved that 8 neurons is enough to get the flexibility needed without an excessive computational load), and an output layer with just 1 neuron (providing the final estimation of the class for the input pattern).

As for the WBCD classification experiment, for the ACAD the activation function used in all the neurons of the system is sigmoidal. The input patterns set is normalized and the initialization of the weights of the neurons is random but included in an interval $[-0.35, +0.35]$, parameter value σ in the sigmoidal activation function is constant and equal to 1. Doing so the inputs to the activation function $\sigma \sum \omega_i x_i$ are limited to the interval $[-4.9, +4.9]$ that is inside the linear part of the sigmoidal function, avoiding a initial saturation of the neurons.

Thus the final experiment characteristics are:

1. Structure of the network:

- Number of inputs: equal to the number of attributes of the pattern (14).
- Number of hidden layers: 1.
- Number of neurons included in the hidden layer: Empirically it has been proved that 8 neurons is ideal for compensating the flexibility in the definition of the decision regions and the computational load required by the system.
- Number of neurons in the output layer: 1 to classify in two classes.
- Activation function: Sigmoidal with output included in the interval $(0, 1)$.

2. Conditions considered to finalize the network training:

- Reach a defined number of input patterns (2000) presented to the network to have enough iterations to reach a stable output without overspecializing the network.

10.3.1 Evaluation Method

The same 50 initial networks have been trained and tested in each one of the variations of the experiment, doing so and calculating mean values of the results it is possible to obtain results independent from the initial random values given to the parameters in the creation of the networks. This fact and the use the 10 fold cross validation method allows eliminating the dependability with the concrete patterns used for training and for performance evaluation, so the final results have a high degree of reliability.

The following hypothesis are defined:

- True Positive (TP) $H(1/1)$: The pattern is accepted and has been classified as accepted.
- False Positive (FP) $H(1/0)$: The pattern is rejected and has been classified as accepted.
- False Negative (FN) $H(0/1)$: The pattern is accepted and has been classified as rejected.
- True Negative (TN) $H(0/0)$: The pattern is rejected and has been classified as rejected.

In the performance evaluation the concepts of confusion matrix, sensitivity, accuracy, ROC and AUC presented in section 9.3.1 are applied. Sensitivity has been considered as the driver figure in these experiments.

Table 10.2 Confusion matrix $\eta = 1$ - Nominal BPA - ACAD

91.8434 %	21.1018 %
8.9368 %	78.8982 %

10.3.2 Results

Three sets of cases are included in the global ACAD classification experiment inside the AMMLP method:

- Nominal BPA: The behavior of the nominal BPA is studied, maintaining the learning rate η inside the limits $(0, 1]$. The objective is to be able to compare the results of the AMMLP with the best possible implementation of the BPA experiment results. The test shows that the best results obtained for accuracy and sensitivity corresponds to value $\eta = 1$:
 - Accuracy = 83.3708%
 - Sensitivity = 91.8433%

The confusion matrix for this learning rate is included in table 10.2.

The evolution of the classification error during the training phase is shown in figure 10.1. The evolution of the error in the first iterations of the training is included in more detail in figure 10.2.

The ROC of this experiment can be found on figure 10.3, the AUC value calculated is 0.9108.

- Use of *a priori* knowledge of the pdf: The Gaussian function approach for the AMMLP is applied for the ACAD data set. As for WCDB this experiment will be divided in two parts, the initial part of the training will use the classic BPA until classification error reach 0.3, after that the training will use equation 6.10 from error 0.3 to the finalization.

Best performing values for the Gaussian function parameters according to the empirical results are $A = 33$ and $B = 0.5$ for the highest values both for accuracy and sensitivity :

- Accuracy = 85.7624%

Fig. 10.1 Evolution of the classification error $\eta = 1$ - Nominal BPA - ACAD

Fig. 10.2 Detail of the first iterations of the evolution of the classification error $\eta = 1$ - Nominal BPA - ACAD

Fig. 10.3 ROC $\eta = 1$ - Nominal BPA - ACADTable 10.3 Confusion matrix $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD

92.0000 %	20.4752 %
8.0000 %	79.5248 %

– Sensitivity = 92.0000%

The confusion matrix for this AMMLP approach is included in table 10.3.

The evolution of the classification error during the training phase is shown in figure 10.4. For a detailed view of the classification error evolution in the first part of the training, the information is included in figure 10.5.

The ROC of this experiment is shown in figure 10.6 and the AUC value obtained is 0.9115.

- Use of *a posteriori* knowledge of the pdf: In this case the output of the MLP is considered as a pdf estimation in the AMMLP implementation. This experiment is also be divided in two parts in order to have comparable conditions with the one based on a

Fig. 10.4 Evolution of the classification error $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD

Fig. 10.5 Detail of the first iterations of the evolution of the classification error $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACAD

Fig. 10.6 ROC $A = 33$ $B = 0.5$ - Gaussian AMMLP - ACADTable 10.4 Confusion matrix $\eta = 23$ - Network output AMMLP - ACAD

91.9060 %	20.1044 %
8.0940 %	79.8956 %

Gaussian pdf. The first part of the training will use the classic BPA until classification error reaches 0.3, after that the second part of the training will use equations 6.8 and 6.9 from error 0.3 to the finalization of the training. This is established so that the AMMLP training starts from a situation when the network has started to learn and the output can be really considered as an estimation of the pdf of the input data set.

The experiments show that the best performance is reached with a value of $\eta = 23$. The associated classification results are:

- Accuracy = 85.9008%
- Sensitivity = 91.9060%

The confusion matrix for this AMMLP approach is included in table 10.4.

The evolution of the classification error during the training phase is shown in figure 10.7. In addition the evolution of the classification error in the first part of the training is included in more detail in figure 10.8.

Fig. 10.7 Evolution of the classification error $\eta = 23$ - Network output AMMLP - ACAD

The ROC of this experiment can be found on figure 10.9 and AUC calculated is 0.9096.

10.3.3 Discussion of the Results

Both AMMLP experiments have produced better results for accuracy and sensitivity. Considering the use of 2000 patterns the best sensitivity result is obtained in the Gaussian experiment (92,0 %) and the best accuracy result is obtained in the output as probability estimation experiment (85,9 %). In general the results obtained from the two AMMLP implementations are very similar between them once the values for the different parameters have been selected. Differences are considered negligible in both accuracy and sensitivity.

If we compare the results obtained in the nominal BPA $\eta = 1$ experiment with the one consisting in the use of the output as probability estimation we can observe a good improvement in the values of the figures under evaluation. It is probed again that once the network

Fig. 10.8 Detail of the first iterations of the evolution of the classification error $\eta = 23$ - Network output AMMLP - ACAD

Fig. 10.9 ROC $\eta = 23$ - Network output AMMLP - ACAD

has started to learn the MLP output can be considered as a estimation of the probability of the input patterns distribution as it was exposed in [14] [188].

As observed in the WBCD classification experiment it is possible to check that the learning (as an evolution of the classification error) is quicker for the AMMLP based on the output of the network compared with classic BPA (from the point where the learning paradigm is changed in the training). In addition the classic BPA and the Gaussian AMMLP present an unstable classification error evolution meanwhile the AMMLP based on the network's output is more smooth and constant.

10.4 KLN Applied to ACAD Classification

The KLN structure is applied to check the classification results of this model with an additional set of real data. In this experiment 50 different initial networks have been trained, all the models used in this study were trained and tested with the same data and validated using the 10-folder cross-validation.

Considering the concrete experiment characteristics:

1. Structure of the network:

- Number of inputs: This KLN has 14 neurons in its input layer corresponding to the number of elements that form each input pattern.
- Number of TC neurons: There are 14 neurons in the TC layer (similar number as input layer to be coherent with the KLN structure).
- Number of S neurons: There are 2 neurons in the S layer as two classes are considered in the experiment.
- Number of B neurons: There 2 neurons in the upper B layer associated to the S output neurons.
- Activation function: Sigmoidal according to equations 7.4 and 7.6.

2. Conditions considered to finalize the network training:

- Reach a defined number of epochs in the network training, the number of epochs will vary during the experiment.

When an input is presented to the network and the processing is complete, one of the output neurons will present a higher activation value than the other, which output will be inhibited by the winning node. The classification of the pattern will be made considering the comparison between the output values, and the performance of the network will be evaluated in comparison with the ideal output of the network.

As in WBCD experiment this KLN prototype has demonstrated to be extremely sensitive to the concrete values of the training mathematical parameters, so very small changes from the used values cause big differences in the results, even in some cases lead to a non convergence of the learning algorithm. In this case the values obtained for the parameters have been obtained using a Montecarlo approach with many simulations until adequate results have been obtained. The final values used in this simulation are $\nu = 0.025$, $\xi = 0.001$, compression factor for the curve $k = 20.13$, the initial sigmoid shift $s = 0.5$, initial weights from TC to S neurons are negligible and random, and non-modifiable weights are set to $W_{S_S} = 0.85$, $W_{S_B} = 0.98$, $W_{I_{TC}} = 1.0$ and $W_{B_S} = 0.5$.

10.4.1 Evaluation Method

The evaluation method is similar to section 10.4.1. In each one of the variations of the experiment 50 networks have been trained using the 10 folder cross validation method to obtain results independent of the concrete patterns used for training and for performance evaluation. After that the mean values of the results obtained by all the networks are calculated to assure that the results are independent of the initial random values in the creation of the networks.

10.4.2 Results

Several variations of the experiment have been performed with different number of epochs in each experiment (considering one epoch like presenting the full set of input patterns once to the network). The output of the network is integrated by two neurons, depending on which one presents the higher level at the output we have considered that one of the classes (accepted or rejected sample) is selected by the network. Table 10.5 presents the results obtained.

It can be seen that the results are quite similar for the different number of epochs. The best results obtained correspond to 14 epochs where a high percentage of accepted patterns

Table 10.5 Sensitivity and accuracy evolution depending on the number of epochs - KLN - ACAD

Epochs	Sensitivity	Accuracy
1	66.00%	63.72%
5	84.68%	81.07%
8	87.13%	82.19%
10	87.15%	82.68%
12	87.14%	82.63%
13	87.56%	82.84%
14	87.66%	83.10%
15	87.65%	82.87%
16	87.33%	83.06%
17	87.32%	83.04%
18	86.98%	83.02%
20	86.93%	83.47%
25	86.89%	83.35%
50	86.25%	83.32%
75	85.81%	82.85%
100	86.06%	82.15%
250	84.49%	81.49%
500	84.41%	82.51%
1000	84.34%	82.60%
1500	84.21%	82.68%
2000	83.85%	82.70%

Table 10.6 Confusion matrix 14 epochs - KLN - ACAD

87.66%	21.46%
12.34%	78.54%

are correctly recognized with a good value for the general accuracy figure. The associated classification results are:

- Accuracy = 83.10%
- Sensitivity = 87.66%

Confusion matrix for this variation of the experiment is shown in table 10.6.

10.4.3 Discussion of the Results

This second experiment related to the use of KLN with real data shows that even the system is very sensitive to the concrete values of the network parameters it is able to obtain good classifications results from the data set. Even with a small amount of epochs the system is able to start learning and the results don't present big variations neither from the sensitivity nor for the accuracy values once the network has been able to obtain the linear independent information from the data set. Once that this situation is reached it is not possible to improve the results with a higher number of epochs in the training.

Comparing the KLN unsupervised results, they are worse than the results of advanced supervised methods as those presented in section 10.3, but with differences under 5% in the worst situation so KLN is considered to perform successful classification task without the need of previous information about the ideal result of the each pattern of the data set.

10.5 State of the Art Study

This section contains information about different studies performed about the application of artificial intelligence methods in the classification of the ACAD patterns used in this experiment.

- In 2000 West [280] investigated the credit scoring accuracy of five ANNs models: MLP, mixture-of-experts, RBFN, LVQ and fuzzy ART. Results were bench marked against

more traditional methods under consideration for commercial applications and showed that the MLP was not the most accurate model, and that both the mixture-of-experts and RBFN models presented better results for credit scoring applications.

- In 2005 Ong *et al* [281] presented a genetic algorithm applied to the ACAD data set and compared the results against other methods as classic ANN, decision trees, rough sets, and logistic regression models. The authors presented the advantages of the algorithm over the other methods considering that it was a non parametric tool adaptable to all data sets, that could determine the adequate discriminant function automatically rather than assigning the transfer function by decision makers, and in addition that it could also select the important variable automatically. Huang *et al* [282] proposed a two-stage genetic programming variation incorporating the advantages of the if-then rules and the discriminant function maintaining the inherent characteristics of the classic genetic algorithms.
- In 2007 Martens *et al* [283] focused in the extraction of human comprehensible rules from classic SVM approach keeping as much the accuracy of the SVM as possible. The experiments presented showed that the SVM rule extraction techniques lost only a small percentage in performance compared to SVMs and therefore ranked at the top of comprehensible classification techniques. Hoffmann *et al* [284] investigated the usefulness of fuzzy classification rules for credit data mining purposes. The authors proposed two evolutionary fuzzy rule learners: an evolution strategy that generated approximate fuzzy rules, whereby each rule had its own specific definition of membership functions, and a GA that extracted descriptive fuzzy rules, where all fuzzy rules share a common, linguistically interpretable definition of membership functions in disjunctive normal form. Huang *et al* [285] combined GAs with SVM classifier. The proposed hybrid strategy could simultaneously perform feature selection task and model parameters optimization.
- In 2008 Peng *et al* [286] proposed a mathematical programming model Multicriteria Convex Quadric Programming (MCQP) model that was efficient and scalable to massive problems because it only needed to solve linear equations to find the global optimal solution. Kernel functions were introduced to the model to solve nonlinear problems. Tsai & Wu [287] compared the performance of a single classifier based on MLP with the results of multiple classifiers (or classifier ensembles) applied to the ACAD.

- In 2009 Nanni & Lumini [288] presented the comparison of different ANN classification methods implemented and applied to the ACAD database. The performance of MLP was very similar to that obtained by the ensembles of MLP while the performance of the other classifiers improved using the ensembles of classifiers (mainly using random subspaces for the input data set). Xu *et al* [289] proposed three link analysis algorithms based on the pre-process of SVM. The proposed algorithms had two major phases which were called input weighted adjustor and class by SVM based models. The link relation by analysis and the integrated relation of applicants through their information is considered into the input vector of the next phase. In the other phase, an algorithm was proposed based on general SVM model. Luo *et al* [290] presented a classifier, Clustering-Launched Classification (CLC). The advantages of using CLC were that it could classify data efficiently and only needed one parameter to be selected. Tsai [291] compared different feature extraction methods in combination with a MLP as classifier, which were t-test, correlation matrix, stepwise regression, principle component analysis and factor analysis. Ping [292] constructed a hybrid SVM-based credit scoring models to evaluate the applicant's credit score from the applicant's input features using different variations. The modifications considered were using neighborhood rough set to select input features, using grid search to optimize RBF kernel parameters, and using the hybrid optimal input features and model parameters to solve the credit scoring problem with 10-fold cross validation. Chen & Li [293] proposed four approaches combining with the SVM classifier for features selection that retains sufficient information for classification purpose. SVM classifier combined with conventional statistical linear discriminant analysis, decision tree, rough sets and F-score approaches as the feature pre-processing steps to optimize input space by removing both irrelevant and redundant data.
- In 2011 Wang *et al* [294] conducted comparative assessment of the performance of three ensemble methods, i.e., bagging, boosting, and stacking, based on four base learners: logistic regression analysis, decision tree, ANN and SVM. Experimental results indicated that the three ensemble methods can substantially improve individual base learners.
- In 2012 Wang *et al* [295] proposed two modifications to decision trees. RS-Bagging DT and Bagging-RS DT, which were based on two ensemble strategies: bagging and random subspace, to reduce the influences of the noise data and the redundant attributes of data and to get the relatively higher classification accuracy. Vukovic *et al*

[296] proposed a case-based reasoning model that used preference theory functions for similarity measurements between cases. A GA was used to select the right preference function for every feature and set the appropriate parameters, or more precisely, for setting the parameters of each preference function, as to set attribute weights

- In 2014 Tsai [297] developed a novel hybrid financial distress model based on combining the clustering technique and classifier ensembles. In particular, two clustering techniques, SOMs and k-means and three classification techniques, logistic regression, MLP, and decision trees, were used to develop four different types of bankruptcy prediction models. As a result, 21 different models were compared in terms of average prediction accuracy. Semenkina & Semenkin [298] proposed a fuzzy classifier automated design based on a hybrid self-configuring evolutionary algorithm. The self-configuring genetic programming algorithm was suggested for the choice of effective fuzzy rule bases and for the tuning of linguistic variables. An additional feature of the proposed approach allowed using genetic programming for the selection of the most informative combination of problem inputs.
- In 2015 Zhao *et al* [299] presented a higher accuracy credit scoring model based on MLP neural networks that have been trained with the BPA. The model focused on enhancing credit scoring models in three aspects: to optimize the data distribution in data sets using a method called Average Random Choosing (ARC), to compare effects of training–validation–test instance numbers and finally to find the most suitable number of hidden units. Liang *et al* [300] conducted an study to examine the effect of performing filter and wrapper based feature selection methods on financial distress prediction. The effect of feature selection on the prediction models obtained using various classification techniques was also investigated. Depending on the chosen techniques, performing feature selection did not always improve the prediction performance.
- In 2016 Xiao *et al* [301] proposed an approach where supervised clustering was employed to partition the data samples of each class into a number of clusters. Clusters from different classes were then pairwise combined to form a number of training subsets. In each training subset, a specific base classifier was constructed. For a sample whose class label needed to be predicted, the outputs of these base classifiers were combined by weighted voting. The weight associated with a base classifier was determined by its classification performance in the neighborhood of the sample. Chen *et al* [302] used Artificial Immune System (AIS) paradigm combined with Population-Based Incremental Learning (PBIL) and Collaborative Filtering (CF) for network

intrusion detection. AIS was a tool used for extirpating antigens inspired by the principles and processes of the natural immune system. PBIL used past experiences to evolve into new species through learning and adopting the idea of CF for classification. This research combined the three previous approaches to a classifier capable to detect intrusion in the data set under study.

Table 10.7 includes the name of the cited researchers, the algorithm or method used in the study and the values for sensitivity (if available) and accuracy obtained.

Table 10.7 State of the art study for ACAD classification

Author (Year)	Method	Sensitivity (%)	Accuracy (%)
West (2000) [280]	MOE	86.70%	86.68%
Ong (2005) [281]	GP	-	88.27%
Huang (2005) [282]	2SGP	-	89.17%
Martens (2007) [283]	SVM	-	85.70%
Hoffmann (2007) [284]	Bayes	-	86.70%
Huang (2007) [285]	GA-SVM	-	86.90%
Peng (2008) [286]	MCQP	87.00%	86.38%
Tsai (2008) [287]	Multiple Classifiers	-	87.25%
Nanni (2009) [288]	LMNC	-	87.44%
Xu (2009) [289]	HARA	-	89.28%
Luo (2009) [290]	CLC	-	86.52%
Tsai (2009) [291]	PCA	-	89.93%
Ping (2009) [292]	RBF-SVM	-	87.52%
Chen (2009) [293]	LDA+SVN	-	86.52%
Wang (2011) [294]	Boosting	-	86.56%
Wang (2012) [295]	RS-Bagging DT	-	88.17%
Vukovic (2012) [296]	Pure CBR-PF-GA	-	88.12%
Tsai (2014) [297]	SOM + MLP	-	91.61%
Semenkina (2014) [298]	SelfCEA	-	90.22%
Zhao (2015) [299]	MLP Scoring	-	87.00%
Liang (2015) [300]	Naive Bayes	-	86.09%
Xiao (2016) [301]	ECSC	-	87.62%
Chen (2016) [302]	PBIL-AIS	-	86.23%
This study	Gaussian AMMLP	92.00%	85.76%
This study	Network Output AMMLP	91.90%	85.90%
This study	KLN	87.66%	83.10%

Chapter 11

Experiment 4: Pollutant Concentration Prediction

11.1 Introduction

The fourth experiment of this thesis proposes a variation in the use of the AMP inside a supervised learning environment. In this case instead of using the AMMLP as a classifier, the system is used for prediction of values of a continuous variable. The model is applied to a real data problem as it is the pollutant concentration of a concrete area in Mexico, the city of Salamanca. Salamanca has been ranked as one of the most polluted cities in Mexico, where the industry in the area led to a major economic development and rapid population growth in the last few years. The concerning registered pollutants are particles in the order of 10 micrometers or less (PM_{10}). The prediction of concentrations of those pollutants can be a powerful tool in order to take preventive public measures such as the reduction of emissions and alerting the affected population.

For the steps followed in this experiment for the AMMLP, the first part is to study the influence of the parameters' values used in the classic nominal BPA for the MLP to assure that the results obtained with the AMMLP modification are compared with the best performing networks using the usual method. After that *a priori* probability estimation of the input distribution is used inside the MLP to implement the AMMLP theory in order to improve the results of the basic training method.

The results of these experiments have been published in [125].

11.2 Salamanca Pollutant Concentration Dataset

Atmospheric pollution is currently one of the most important environmental problems at global scale [303]. It affects all societies, no matter the degree of socioeconomic development they have, and has a significant impact on human health. According to a recent study, carried out by the European Environmental Agency, atmospheric pollution is the environmental factor with the greatest impact on human health in Europe and is responsible for the greatest number of environmentally related illnesses. The estimates of this study show that 20 million european citizens suffer from respiratory problems every day related to air pollution [304].

The air quality in cities is highly impacted depending on the degree of industrialization, population density, traffic density, topographical characteristics and meteorological variables [305], [306], [307] and many other social and environmental characteristics. Among the different aspects that have influence on this problem it has been demonstrated that the global and regional variations in the climate together with the topographical conditions of the studied area highly affect the transport and dispersion of pollutants [308], [309]. Parallel studies are focused on the appearance of severe episodes of pollution due not only to sudden increases in the pollutant concentration but also to certain meteorological conditions that reduce the ability of the atmosphere to disperse the concentrations [310], [311].

Among the meteorological parameters that have influence on the pollutant concentrations wind patterns, clouds, rain and temperature can affect how quickly pollutants move away from an area. As a consequence of the alterations in the pollutant concentrations the changes in the chemical composition of the atmosphere may produce changes in climate, bring acid rain, destroy the ozone layer and affect the Earth's biodiversity. So it is proved that air quality is decisive for human health and the environment. These problems have attracted the interest of many researchers, which have developed different forecasting strategies.

Salamanca city is located in the state of Guanajuato, Mexico, and it has an approximate population of 273 271 inhabitants. The city is 340 km northwest from Mexico City, with coordinates 20°34'09" North latitude, and 101°11'39" West longitude. The Automatic Environmental Monitoring Network (AEMN) was installed in Salamanca in 2007. The system has the necessary instrumentation to measure the concentration of several pollutants as well as the meteorological variables. The measured meteorological variables are: Wind Direction (WD), Wind Speed (WS), Temperature (T), Relative Humidity (RH), Atmospheric Pressure

(AP), Precipitation (P) and Solar Radiation (SR).

The proposed model predicts the average concentration of PM_{10} for the next 24 hours based on the information provided by the AEMN. It is an evolution of the model presented in [312], adding the AMP to the system. The model combines ANNs with clustering algorithms that have been previously applied in order to find relationships among pollutant and meteorological variables. These relationships that are expressed as groups where the inputs are pre-classified. This cluster label allows getting additional information in order to obtain a more accurate prediction model. The clustering algorithm is not part of the study of this article, only the results are used as an input to the network, details of the complete clustering process can be found on [312].

11.2.1 Data Preparation

Due to the large amount of signaling information included in the database as well as to the random erroneous measurements, it is necessary to review and refine the gathered information. Interpolation method was used to replace the missing data. For the pre-processing of the data based on different clustering algorithm (K-Means and FCM) PM_{10} pollutant concentrations, wind direction, wind speed, temperature and relative humidity were used to create patterns. These patterns were created as follows:

$$P = [C_{PM_{10}}, WS, WDI, T, HR] \quad (11.1)$$

where, $C_{PM_{10}}$ is the previous day value for the PM_{10} concentration, WS is wind speed, WDI is the Wind Direction Index, T is temperature and HR is the relative humidity. The WDI is calculated according to:

$$WDI = 1 + \sin(WD + \pi/4) \quad (11.2)$$

The final network input patterns will contain a label corresponding to the result of the clustering algorithm. In order to compare the results obtained with the ones presented by [312] the same input patterns original groups have been used:

- K-Means: Prediction based on the previous day using 8 clusters.
- FCM: Prediction based on the previous day using 7 clusters.

In both cases the total amount of patterns of the network is 962.

AS in previous experiments the *10 fold cross validation* evaluation method has been considered. All the samples are used to train the networks and all the samples are used to evaluate the performance of the results in different executions of the experiment for the same initial neural networks, mean values are calculated to establish the final performance results.

For this experiment we have created ten data sets from the pollutant data set with the following distribution of patterns:

- G1: 97 patterns
- G2: 97 patterns
- G3: 96 patterns
- G4: 96 patterns
- G5: 96 patterns
- G6: 96 patterns
- G7: 96 patterns
- G8: 96 patterns
- G9: 96 patterns
- G10: 96 patterns

The method to create the 10 folders is explained in section 9.2.1. The networks are trained from the same initial aleatory weights and bias values presenting the data corresponding to each of the 10 folders. Finally the mean values of the results will be calculated to eliminate the possible statistical influence in the results due to the concrete fixed selection of some patterns to train the system and the fixed selection of other patterns to evaluate the results.

11.3 AMMLP Applied to Pollutant Prediction

An AMMLP model approach is applied for the classification of the Salamanca pollutant concentration data set. For these experiments a MLP neural network have been used with a input composed by 6 attributes contained in each single pattern, a hidden layer composed by 3 neurons (considering that experiments presented in [312] proved that 3 neurons is enough to get the flexibility needed), and an output layer with just 1 neuron (that provides the final value for the prediction).

As for the previous experiments, the activation function used in all the neurons of the system is sigmoidal. The input patterns set is normalized and the initialization of the weights of the neurons is random but included in an interval $[-0.8, +0.8]$, parameter value σ in the sigmoidal activation function is constant and equal to 1. Considering these values the range of inputs to the activation function $\sigma \sum \omega_i x_i$ will be included in the interval $[-4.8, +4.8]$ that corresponds with the linear part of the activation function, so the initial response of the neurons will not be saturated.

Thus the final experiment characteristics are:

1. Structure of the network:

- Number of inputs: equal to the number of attributes of the pattern (6).
- Number of hidden layers: 1.
- Number of neurons included in the hidden layer: Based on previous experience 3 neurons are considered enough to provide good prediction flexibility without increasing complexity of the system too much.
- Number of neurons in the output layer: 1 to provide a final numerical value.
- Activation function: Sigmoidal with output included in the interval $(0, 1)$.

2. Conditions considered to finalize the network training:

- Reach a defined number of input patterns (200) presented to the network to have enough iterations to reach a stable output without overspecializing the network. In each iteration the complete training set is presented to the network.

11.3.1 Evaluation Method

The same 50 initial networks have been trained in each one of the executions of the experiment, doing so and calculating mean values of the results it is possible to obtain performance values independent of the initial random value in the creation of the networks. This method together with the use the 10 fold cross validation method to eliminate the dependability with the concrete patterns used for training and for performance evaluation makes the final results having a high degree of reliability.

The AMMLP model performance was evaluated through the following parameters: Mean Absolute Error (MAE) and Mean Square Error (MSE):

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (11.3)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i|^2 \quad (11.4)$$

Being Y_i the expected result and $\hat{Y}_i$ the obtained one.

11.3.2 Results

Two sets of cases are included in the pollutant prediction experiment inside the AMMLP method, inside each case two sub-cases are considered:

1. Experiments using K-Means clustered inputs

- Nominal BPA: In this first part of the training the behavior of the nominal BPA is studied, maintaining the learning rate η inside the limits $(0, 1]$. The objective is to be obtain the best possible BPA results for the comparison with the AMMLP. The tests indicate that the best results obtained for MSE and MAE corresponds to value $\eta = 1$:

- $MSE = 0.0012$

- $MAE = 0.0270$

The evolution of the MSE error during the training phase is shown in figure 11.1. The evolution of the error in the first iterations of the training is included in more detail in figure 11.2.

Fig. 11.1 Evolution of the MSE error $\eta = 1$ - Nominal BPA - K-Means pollutant prediction

Fig. 11.2 Detail of the first iterations of the evolution of the MSE error $\eta = 1$ - Nominal BPA - K-Means pollutant prediction

- Use of *a priori* knowledge of the pdf: The Gaussian function approach for the AMMLP is applied for the pollutant data set. This experiment will use equation 6.10 during the complete training.

Best performing values for the Gaussian function parameters according to the empirical results are $A = 10$ and $B = 0.55$ for the best performance both for MSE and MAE:

- $\text{MSE} = 0.0010$
- $\text{MAE} = 0.0245$

The evolution of the MSE error during the training phase is shown in figure 11.3. To check the detailed evolution of the MSE error during the first part of the training the information is included in figure 11.4.

Fig. 11.3 Evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - K-Means pollutant prediction

2. Experiments using Fuzzy C-Means clustered inputs

Fig. 11.4 Detail of the first iterations of the evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - K-Means pollutant prediction

- Nominal BPA: This execution of the experiment studies the result of the nominal BPA considering the learning rate η within the nominal limits $(0, 1]$. Doing so it is possible to be obtain the best possible BPA results and compare then with the AMMLP performance. The test indicate that the best results obtained for MSE and MAE corresponds to value $\eta = 1$:
 - MSE = 0.0012
 - MAE = 0.0267

The evolution of the MSE error during the training phase is shown in figure 11.5. The evolution of the MSE error in the first part of the training process is included in more detail in figure 11.6.

- Use of *a priori* knowledge of the pdf: The Gaussian function approach for the AMMLP is applied for the pollutant data set. This experiment will use equation 6.10 during the complete training.

Fig. 11.5 Evolution of the MSE error $\eta = 1$ - Nominal BPA - Fuzzy C-Means pollutant prediction

Fig. 11.6 Detail of the first iterations of the evolution of the MSE error $\eta = 1$ - Nominal BPA - Fuzzy C-Means pollutant prediction

Best performing values for the Gaussian function parameters according to the empirical results are $A = 10$ and $B = 0.55$ for the best performance both for MSE and MAE with similar results than in the previous clustering:

- MSE = 0.0010
- MAE = 0.0245

The evolution of the MSE error during the training phase is shown in figure 11.7. The evolution of the error in the first iterations of the training is included in more detail in figure 11.8.

Fig. 11.7 Evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - Fuzzy C-Means pollutant prediction

11.3.3 Discussion of the Results

The results obtained from the Gaussian estimation applied to the metaplasticity improve those recently obtained in [312] for both MSE and MAE. We can confirm then that a deeper learning is obtained from the same data set just by applying the bio-inspired AMP. The hypothesis that the patterns with less occurrence frequency (atypical data) contain more information than the patterns with more frequency is so reinforced, and this information can

Fig. 11.8 Detail of the first iterations of the evolution of the MSE error $A = 10$ $B = 0.55$ - Gaussian AMMLP - Fuzzy C-Means pollutant prediction

be applied in ANN learning to improve the prediction results.

The results obtained from the variants K-Means and Fuzzy C-Means are very similar between them once the values for the different parameters have been selected. Differences are considered negligible in both MSE and MAE. The amount of information provided by both types of clustering can be considered similar.

In the detail of the evolution of the error it can be observed that the evolution is much faster using metaplasticity and also the asymptotic final performance. It can be extended to a very high improvement method in case of general Big Data, where atypical patterns have usually irrelevant impact in the overall learning process while being responsible of relevant information, as it happens with the influence of these atypical patterns in the PM_{10} concentration prediction.

11.4 State of the Art Study

This section contains a brief state-of-the-art study of the application of different methods to pollutant and contamination data processing.

- In 2000 Perez *et al* [308] investigated the prediction of the concentration of $PM_{2,5}$ particles using different models as MLP, linear regression and persistence. The MLP network presented the best results with an error value of 30% for the prediction in the first hours of the day and 60% for the last hours of the day, using statistical values of the meteorological variables. Ando *et al* [313] provided an estimation of the CO and SO_2 levels for the city of Catania in Italy. The authors used two different methods: an MLP and a optimization method using MSE. The model used variables as the number of vehicles, the wind speed, the turbulence intensity, atmospheric pressure and temperature.
- In 2001 Schlink *et al* [314] implemented a Box-Jenkins model for the prediction of O_3 concentrations with a time horizon of 1 and 2 days. The statistical data used were mean concentrations of ozone for 1-8 hours, NO concentrations and the temperature. The result concluded that the use of the temperature data improves the accuracy of the results.
- In 2003 Silva *et al* [315] compared four methods for the prediction of $PM_{2,5}$ and PM_{10} particles for the city of Santiago de Chile using as input variables the temperature, relative humidity, wind speed, wind direction, month, week day, hour and concentration level of particles. The analysis was made using non parametric discriminants, a BPA neural network mode, a multiple linear regression model and Multivariate Adaptive Regression (MAR) models. The author concluded that the MAR model was faster and presented more accurate results compared with the other methods. Kukonen *et al* [316] implemented an MLP for the prediction of urban NO_2 and PM_{10} concentrations, using information of traffic flow and meteorological variables as input data. The results showed that the NO_2 model performance was better than the PM_{10} one.
- In 2004 Nagendra & Khare [305] implemented an MLP to predict the mean value of the CO after 8 hours in Delhi, India. The pollutant concentration is closely linked to the vehicle emissions so the authors analyzed three scenarios using as inputs meteorological variables, traffic characteristics and particle concentrations. Jiang *et al* [317] designed an MLP to analyze the air quality considering PM_{10} , SO_2 and NO_x in Shanghai. The results obtained were evaluated using the correlation coefficient with the real values.

- In 2005 Corani [318] predicted the concentrations for O_3 and PM_{10} for Milan, Italy. The authors compared a feedforward ANN, a Pruned Neural Network (PNN) and a lazy learning method. The author concluded that there were not important differences between the different methods. Ordieres *et al* [319] predicted the value of $PM_{2.5}$ for the frontier between USA and Mexico. The prediction used three ANN: a classic MLP, a RBFN and a square MLP. The results were compared with classic methods as persistence model and linear regression. The best results were obtained using the RBN.
- In 2006 Agirre-Basurkoa *et al* [320] compared two MLP and a linear regression model to predict the O_3 and NO_2 concentrations in real time for 8 hours in advance in Bilbao, Spain. The models used the traffic value variables, meteorological variables and previous pollutant concentration levels. The results of the MLP outperformed the linear regression model. Salini & Pérez [321] designed an ANN for the prediction of $PM_{2.5}$ particles for Santiago de Chile using the Average Mutual Information (AMI) and the False Nearest Neighbors (FNN) methods. This methods were used to obtain the average of the input data in a concrete period of time and the appropriate correlation using the correct amount of data to obtain a better modeling. Grivas & Chaloulakou [322] predicted the concentration per hour of PM_{10} in Athens, Greece, using different meteorological variables. A MLP was used first considering all the possible input data, second using only a selection of the possible data and third using only the previous concentration values, without important differences in the results obtained. Schlink *et al* [323] compared different statistical techniques for the prediction of O_3 . The lost information of the data set was substituted using a SOM network or linear interpolation. The results obtained with the SOM were better than using interpolation.
- In 2007 Brunelli *et al* [324] predicted the maximum concentration values for the pollutants SO_2 , O_3 , PM_{10} , NO_2 and CO in Palermo, Italy. The proposed model was a recurrent ANN that used meteorological variables as wind direction, wind speed, atmospheric pressure and ambient temperature. Dutot *et al* [325] proposed a model that combines a MLP and neural classifier to predict the concentration of O_3 in central France. The inputs for the network used were the maximum and minimum values for several meteorological variables and the previous values of the pollutant concentration. The results obtained reached 72% in the prediction accuracy. Karatzas & Kaltsatos [326] predicted the value for O_3 one day in advance in Thessaloniki, Greece. The

authors combined the meteorological values of two different monitoring stations and reached accuracy values of 83%.

- In 2008 Díaz-Robles *et al* [327] implemented a hybrid model for the prediction of PM_{10} in Temuco, Chile. An MLP and a Box-Jenkins models were combined to improve the accuracy of the results. These results were compared with classic prediction methods providing similar performance. Coman *et al* [328] predicted the O_3 concentration for the monitoring network of Paris, France, by the comparison between a dynamic ANN model and an statistical one. The experiments based on the ANN considered the meteorological variables in one set of experiment that provided better results than in the case of dismissing this information. The authors also considered the amount of iterations and data presented to the network to avoid over specialization in the results. Kurt *et al* [329] proposed an ANN to predict SO_2 , PM_{10} and CO pollutant concentration 1, 2 and 3 days in advance. The ANN model was feed forward and included a pre-processing stage to eliminate the gaps in the data set. Salazar-Ruíz *et al* [330] proposed several methods for predicting the O_3 concentration based in a data set from several zones from Mexico and California. The methods considered were persistence, linear regression, Ridge regression, MLP, SVM and a recurrent ANN. The best results were obtained by the MLP and the recurrent ANN.
- In 2009 Sharma *et al* [331] implemented a Box-Jenkins model to predict the total number of suspended particles in the air. The model studied the quantity of data needed for having reliable results and at least 50 days of successive observations were needed. Barai *et al* [332] proposed a model for prediction of SO_2 , PM_{10} and NO_2 for Maidstone, UK, using a MLP with sliding windows for the data set. Each window included meteorological and concentration data for a time range of 24 hours. Hrust *et al* [333] predicted the hourly concentration for different pollutants using a persistence, linear and MLP models. The system used an uni-variant regression model to determine the interval to be used for the input variables.
- In 2010 Kurt & Oktay [334] proposed a geographical model for predicting SO_2 , PM_{10} and CO using the information available for the near districts of the zone under study. The model used 3 different districts and a triangulation of the data, and for every variable an ANN was trained. Mesin *et al* [335] provided the prevision of the PM_{10} concentration values for Sweden. The system used partial mutual information criteria to eliminate redundant and non relevant data, and included it as input to

a MLP considering minimum, maximum and mean values for meteorological and pollutant concentration data from three days. The result was a correlation coefficient of 0.91. Sfetsos & Vlachogiannis [336] predicted the daily average value for the PM_{10} concentration in Athens, Greece. The authors tried to use shorter periods of time to better model the data daily series. This information was applied to a linear regression model, a nearest neighbor model and a feed forward ANN.

- In 2011 Mitra & Barman [337] used an feed forward ANN to predict the concentration of CO_2 where input variables were temperature, time, and pressure. This concentration value was associated to the presence of coumarin, a chemical with several applications in industry. Turan *et al* [338] presented a work where ANNs were applied for the prediction of percentage adsorption efficiency for the removal of Cu_2 ions from industrial filtering residuals by pumice. The effect of operational parameters such as initial pH, adsorbent dosage, temperature, and contact time was studied to optimize the conditions for maximum removal of ions. Shivakumar *et al* [339] presented an emission performance model for engines based in ANN. To train the network, compression ratio, injection timing, blend percentage, percentage load, were used as the input parameters where as engine performance parameters like brake thermal efficiency, brake specific energy consumption, exhaust gas temperature were used as the output parameters for the performance model and engine exhaust emissions values were used as the output parameters for the emission model.
- In 2012 Frontistis *et al* [340] proposed an ANN comprising five input variables (reaction time, TiO_2 and EE_2 concentration, organic content and conductivity of the water matrix), and an output variable (EE_2 conversion) to optimize the prediction of EE_2 degradation. The network, based on tangent sigmoid and linear transfer functions for the hidden, input and output layers, respectively, and the Levenberg–Marquardt back propagation training algorithm, provided good prediction values.
- In 2013 Khajeh *et al* [341] proposed a method for the determination of the amount of blue methylene in water samples, developed from silver nano particles based solid-phase extraction method and UV–Vis spectrophotometry. Response surface methodology and hybrid of ANN particle swarm optimization were used to develop predictive models for simulation and optimization. Zhang *et al* [342] studied the concentration estimations of indoor contaminants for air quality monitoring in dwellings using chaos based optimization ANN network. The authors presented a chaotic sequence modification of the BPA to train the system.

- In 2014 Hattab & Motelica-Heino [343] developed and tested an inverse ANN model for the prediction of the optimal soil treatment to reduce copper toxicity assessed by a given target concentration of Cu in dwarf bean leaves from selected soil inputs. In this study the inputs were the soil pH, electrical conductivity, dissolved organic carbon and a given target toxicity value of Cu, whereas the output was the best treatment to reduce the given toxicity level.
- In 2015 Varley *et al* [344] proposed a method to characterize Ra contamination by the interrogation of characteristic Montecarlo calibration spectra using a combination of principal component analysis and ANN. The technique demonstrated that it could reliably distinguish spectra that contained contributions from point sources from those of background or dissociated contamination (homogeneously distributed).
- In 2016 Adams & Kanaroglou [345] proposed an approach for the measurement of $PM_{2.5}$ and NO_2 by applying ANN within a framework that was inspired by land use regression. The authors modeled the air pollution data with a number of predictor variables that included information on the surrounding land use characteristics, the meteorological conditions, air pollution concentrations from fixed location monitors, and traffic information during the time of data collection. The model performances were measured with a coefficient of determination at 0.78 and 0.34 for $PM_{2.5}$ and NO_2 , respectively. Giwa et al [346] presented a new configuration of an electrically-enhanced membrane bio-reactor to treat medium strength waste water at Masdar City, Abu Dhabi, United Arab Emirates. The integrated setup enhanced the reduction of waste water contaminant concentrations. ANNs based ensemble model was used to model the experimental findings of pollutants concentration removal given the initial mixed compositions.

Chapter 12

Conclusions

This thesis proposes and develops two different neural models based in the application of the Artificial Metaplasticity and Intrinsic Plasticity to supervised and unsupervised ANN learning methods in order to improve the training results considering performance, reliability and convergence speed. Metaplasticity and Intrinsic Plasticity are biological properties of the neural system that are related mainly to memory and learning. In addition the Shannon's information theory is also applied to the ANN training methods, based on the premise that less frequent input patterns of the data set contains more information than the more frequent patterns, by giving more relevance to these less frequent patterns in the training execution. Two learning methods have been developed and applied to synthetic and real data: AMMLP for supervised learning and KLN for unsupervised learning.

- The synthetic radar signal classification experiment is the first approach to model the AMMLP considering its relationship with the Shannon's information theory, by giving more relevance to less frequent patterns as a way to model the neuron's biological metaplasticity property. The idea is that the modification can improve the training evolution and performance results. In this case two different possibilities are implemented and tested, being able to apply the *a priori* knowledge of the pdf of the data set or using the MLP output itself as *a posteriori* estimation of the input pattern probability. The results showed that by modeling this interpretation of AMP in the training phase it is possible to obtain a much more efficient training, and at the same time maintaining at least the same performance provided by the classic BPA. It is possible to check that in the best performing network the distance between the results obtained and the Marcum theoretical limit for the network performance is less than 1 dB and the number of training iterations needed to reach this solution is significantly

smaller than using the classic BPA. These facts make the proposed training strategy and the algorithm very interesting in the cases where a low number of input patterns are available to train the ANN. In this way it can be concluded that both implementations possibilities of the AMMLP can help solving the problems identified for the nominal BPA. Furthermore it is also concluded that the theory that considers the MLP output as a probability estimator can be considered correct in this case, eliminating the need of having previous statistical knowledge about the input data set.

- The classification of the WBCD is used in order to test the capabilities of the AMP using a real data set. In this cases two different approaches are considered, in one hand the use of a supervised training method in the form of the AMMLP and in the other hand the use of an unsupervised training method in the form of a KLN. Considering the AMMLP two possibilities are applied for the input data set probability estimation (the Gaussian approach and the use of the MLP output as an estimator). The results are better in accuracy and sensibility than the ones obtained by classic BPA, and for the concrete case of the use of the output of the network the classification error evolution is more natural and smooth than in the other cases. Due to the general nature of the metaplasticity concept, that it is as wide as plasticity itself, this results are coherent with the hypotheses mentioned in the thesis. Considering the unsupervised KLN artificial intrinsic plasticity implementation it is proved that this model can be applied to a real data set. The simulations show that the unsupervised learning that emerges from individual neurons properties surpasses results even of the supervised learning methods. Nevertheless the results presented are from a non optimized prototype that present problems of robustness and convergence if the parameter values are not specially tuned for the concrete application, this bio-inspired model seems to be able to compete in deeper learning and performance. A state-of-the-art comparison for the classification of this data set is presented and the results show that both AMMLP and KLN presents comparable and even better results than the other algorithms studied, so it is considered that both AMP implementations can be considered as an additional source of information for the medical diagnostic in this area.
- The classification of the ACAD is used in the same way than the WBCD to check the capabilities of the AMP implementation as a classifier of a real data set. The same set of tests of the previous experiments are performed and results compared. Considering the AMMLP approach the sensibility and accuracy are better in both AMP implementations (the Gaussian approach and the use of the MLP output as an

estimator) than using the best obtained version of the BPA. When the classification error evolution is observed the use of the output as a pdf estimator provides a curve that seems to be more similar to natural learning than in the other cases, without big peaks and with a regular evolution. As the system is able to converge with a reduced number of input data patterns it is possible to reach the final state of the network without the risk of an over specialization for the training patterns. For the KLN, in this case it is observed that the results are very similar to the ones obtained in the supervised training. The figures obtained for accuracy and sensitivity are almost the same with independence of the number of iterations used in the unsupervised training, so it seems that the linear independent information present in the data is identified and processed with a very small number of iterations which allows to reduce the computational load of the method. A state-of-the-art comparison for the classification of this data set is presented and the results show that both AMMLP and KLN presents results similar than the other algorithms considered. The use of these learning and evaluation methods should be considered as a tool to support the decisions of the financial risk analysis process and in a more general way a contribution to the development of Industry 4.0. and Business Intelligence.

- The pollutant concentration data set is used to check the performances of the AMMLP model as a predictor in addition of classification capabilities tested in previous experiments. The goal of this research was to improve the accuracy of the prediction obtained in previous recent published results while reducing the training time. The results show that the use of of the AMMLP based on a Gaussian estimation of the data set pdf is an option for accurate pollutant concentration prediction and could be used as a computer aided system for taking decisions protecting public health. It also can be concluded that atypical patterns in the continuously growing data set of measures can include information of physical phenomena responsible of significant changes in PM_{10} concentration, and most similar prediction methods are not considering properly their value for the overall performance of the prediction.
- All the results obtained in the experiments consider different values for the parameters involved in all the training methods under evaluation, selecting the best performing ones, including the classic BPA used for comparison, the AMMLP and the KLN. The number of iterations considered in the experiments and the concrete structure used in the different models of ANN have been empirically adapted to procure the best results for performance and convergence speed. These values and structures can be used as a

first approximation for future research, specially in the KLN case where the prototype behavior is very sensitive for the parameters' concrete values.

- The AMMLP application in all the experiments performed is able to improve the results in terms of convergence speed and performance in comparison with the classic BPA. The reduction in the number of input patterns used to reach an stable training result is specially useful for experiments where the amount of data available is very low.
- The results obtained in all the experiments that uses the AMMLP as a classifier applying the output of the network as the weight modifier demonstrate the theory that states that the output of the MLP is a reliable *a posteriori* estimator of the pdf of the input data set once that the network has started to learn. The evolution of the classification error in those cases is specially smooth and more similar to the natural learning curve than in other implementations of the AMMLP or with the classic BPA.
- The results obtained by the KLN as a classifier indicate that the implementation of intrinsic plasticity and artificial metaplasticity in a unsupervised training can be applied with very good results to real data sets in addition to the test performed only with synthetic data up to now in other studies. The system has demonstrated that once the parameters are tuned to the concrete application, it is possible to provide results comparable to other state-of-the-art systems (ACAD) or even better (WBCD).

Chapter 13

Contributions, Future Research Lines and Publications

13.1 Contributions

This section presents the main contributions included in this thesis in relation with the objectives presented in section 1.5. They can be summarized as applying the implementation of the AMP and intrinsic plasticity properties to supervised and unsupervised neural network models in real world problems to improve the results obtained with the classic methods. The following lists includes the contributions in detail:

- Artificial Metaplasticity is implemented in a classifier based on MLP taking into account its relationship with the Shannon's information theory and the use of the output of the network as *a posteriori* estimator of the pdf of the input data set and applied to real and synthetic data sets.
- The implementation of Artificial Metaplasticity in a predictor based on MLP taking into account its relationship with the Shannon's information theory and the use of *a priori* estimation of the pdf as a Gaussian data set is applied successfully to a real data set.
- Intrinsic Plasticity and Artificial Metaplasticity are implemented in a classifier based on KLN unsupervised neural network and applied to real data sets.
- Synthetic radar signal classification problem is studied using the AMMLP based on the output of the network and results are compared to the Gaussian implementation of the AMMLP and the best performing BPA. The results obtained by the new implementation

of the AMMLP show that the best network performance is very close to the Marcum theoretical limits for this kind of detectors using a extremely small number of input patterns during the training process. The results prove that the output of the MLP can be considered as a reliable estimator of the pdf of the input data set.

- The relevant and critical problem of breast cancer pattern classification is tackled using two different approaches based on the AMP and intrinsic plasticity obtaining high accuracy and sensitivity in the classification results. The results obtained using the AMMLP based on the output of the network are better than the ones obtained by the best performing BPA, and at least comparable and in some cases better than the results obtained by the state-of-the-art algorithms reviewed in this thesis. The results obtained by the unsupervised KLN outperformed the results obtained using supervised learning methods. Based on the global results obtained in the experiments performed in this thesis it can be considered that the use of AMP and intrinsic plasticity can be used as a second opinion for the breast cancer diagnosis task.
- To check the generalization possibilities of the neural network models proposed the problem of credit approval pattern classification is studied with the same approach than the previous experiment obtaining high accuracy and sensitivity results. The classification values provided by the AMMLP based on the output of the network are better than the results obtained by the best performing BPA, and can be considered comparable and even better than some of the results obtained by the state-of-the-art algorithms reviewed for the same classification problem. The results obtained using the KLN model for this credit approval data set classification task show that the model is able to integrate the linear independent information present in the data set with a very low number of iterations with a performance comparable to the other methods reviewed. The application of AMP and intrinsic plasticity can be considered as a powerful tool for supporting the financial credit evaluation.
- The application of the AMMLP as a predictor is tested in this thesis considering the problem of the PM_{10} pollutant concentration that represents a real risk for public health. The input data set under consideration includes information about the meteorological variables and the relationship between patterns obtained by different clustering algorithms. The AMMLP based on a Gaussian pdf approximation of this input data set provided better results than the best performing BPA for both MSE and MAE error estimators. The model proposed in this thesis provides additional information that

can be transmitted to the population and public authorities to avoid exposition to high concentrations of the considered pollutant and the derived health problems, especially for high sensitive population as the children or the elderly.

- All the experiment results include the an empirical study of the parameters related to the AMP and intrinsic plasticity for each one of the neural model under evaluation. The number of iterations considered in the experiments and the concrete structure used in the different models of ANN have been also empirically studied to obtain the best results considering performance and convergence speed. These values and structures can be used as a first approximation for future research, specially in the KLN case where the prototype behavior is very sensitive for the parameters concrete values.
- The AMMLP application in all the experiments performed in comparison with the classic BPA is able to improve the results in terms of convergence speed, accuracy and sensitivity. The reduction in the number of input patterns used to reach an stable training is specially useful for experiments where the amount of data available is very low. The system globally can be considered as a high efficient method for neural network supervised learning.
- The use of the AMMLP as a classifier applying the output of the network as the weight modifier confirms the theory that considers the output of the MLP as a reliable *a posteriori* estimator of the pdf of the input data set, fulfilling the important condition that the network has started to learn before (for example using the classic BPA or other AMMLP implementation). The evolution of the classification error in those cases is specially smooth and more similar to the natural learning curve than in comparison with other experiments.
- The results obtained by the KLN as a classifier indicate that this unsupervised intrinsic plasticity model can be applied with very good results to real data sets in addition to synthetic data. The system has demonstrated that, once the parameters are tuned to the concrete application, is able to provide results comparable or even better than the rest of the state-of-the-art algorithms considered.

13.2 Future Research Lines

Based on the experience acquired during the development of this thesis it is possible to propose several research lines that can be based on the works developed and described in previous chapters:

- Implementation of the AMP and intrinsic plasticity in other supervised or unsupervised neural network models and comparison of results with the experiments presented in this thesis.
- Analytic optimization of the parameters included in the neural network models, in particular the ones related to the KLN in order to be able to obtain a more robust structure less sensitive to the concrete application characteristic values.
- Application of an optimization pre-processing phase to the classification process in the same way that the clustering algorithms are applied before the prediction performed by the AMMLP. The idea is to improve the performance and speed results of the current versions of the methods.
- Combination of the additional complementary biological characteristics in an unique neural network model (supervised or unsupervised) to be able study the synergies between all properties in a artificial model closer to the real brain structure and behavior.
- Application of the neural networks models proposed to additional relevant classification and prediction problems.

13.3 Publications

13.3.1 Journal Publications

1. **Journal:** Intelligent Automation and Soft Computing.

JCR: 0,349.

Quartile: Q4.

Title: Artificial metaplasticity can improve artificial neural networks learning.

Authors: D. Andina, A. Álvarez-Vellisco, J. Aleksandar and J. Fombellida.

Volume: 15(4).

Pages: 683–696.

Year: 2009.

2. **Journal:** Neural Computing and Applications.
JCR: 0,435.
Quartile: Q2.
Title: Tackling Business Intelligence with bioinspired deep learning.
Authors: J. Fombellida, I. Martín-Rubio, S. Torres-Alegre and D. Andina.
Volume: Accepted.
Pages: Accepted.
Year: Accepted.
3. **Journal:** Frontiers in Neuroinformatics.
JCR: 3,870.
Quartile: Q1.
Title: On the role of neuroinformatics in Industry 4.0 and Business Intelligence.
Authors: I. Martín-Rubio, J. Fombellida, A. L. Romera-Zarza and D. Andina.
Volume: Pending on final acceptance.
Pages: Pending on final acceptance.
Year: Pending on final acceptance.
4. **Journal:** Neural Computing and Applications.
JCR: 0,435.
Quartile: Q2.
Title: AMSOM: Artificial metaplasticity in SOM neural networks. Application to MIT-BIH arrhythmias data base.
Authors: S. Torres-Alegre, J. Fombellida, J. A. Piñuela-Izquierdo and D. Andina.
Volume: Pending on final acceptance.
Pages: Pending on final acceptance.
Year: Pending on final acceptance.

13.3.2 Congress Publications

1. **Congress:** CITSA 2007 - Int. Conference on Cybernetics and Information Technologies, Systems and Applications and CCCT 2007 - Int. Conference on Computing, Communications and Control Technologies.
Title: Metaplasticity artificial neural networks model application to radar detection.

Authors: D. Andina and J. Fombellida.

Publication: Journal of Systemics, Cybernetics and Informatics.

Volume: 5(6).

Pages: 44–48.

Place: Orlando, Florida, USA.

Year: 2007.

2. **Congress:** WAC 2008 - World Automation Congress.

Title: Edge detection algorithms in intensity SAR images: Empirical comparison of performances and proposed improvements.

Authors: I. Melgar, J. Fombellida, J., Gomez and J. Seijas.

Publication: World Automation Congress Proceedings.

Place: Waikoloa, HI, USA.

Year: 2008.

3. **Congress:** WAC 2008 - World Automation Congress.

Title: SAR image segmentation algorithms: Performance comparison and improvement proposals.

Authors: J. Fombellida, I. Melgar, J., Gomez and J. Seijas.

Publication: World Automation Congress Proceedings.

Place: Waikoloa, HI, USA.

Year: 2008.

4. **Congress:** INDIN 2009 - IEEE International Conference on Industrial Informatics.

Title: Swarm architectures for ground-based air defense systems of systems.

Authors: I. Melgar, J. Fombellida, A. Jevti and J. Seijas.

Publication: 2009 7th IEEE International Conference on Industrial Informatics.

Place: Cardiff, Wales.

Year: 2009.

5. **Congress:** INDIN 2009 - IEEE International Conference on Industrial Informatics.

Title: Combining edge detection algorithms in intensity SAR images.

Authors: J. Fombellida, I. Melgar and J. Seijas.

Publication: 2009 7th IEEE International Conference on Industrial Informatics.

Place: Cardiff, Wales.

Year: 2009.

6. **Congress:** IECON 2009 - 35th Annual Conference of IEEE Industrial Electronics.
Title: Cooperation and competition based on free market in swarm system architectures for air defense .
Authors: I. Melgar, J. Fombellida, J. Seijas and F. Quintana.
Publication: IECON Proceedings.
Place: Porto, Portugal.
Year: 2009.
7. **Congress:** IWINAC 15 - International Work-conference on the Interplay between Natural and Artificial Computation.
Title: Artificial metaplasticity for deep learning: Application to WBCD breast cancer database classification.
Authors: J. Fombellida, S. Torres-Alegre, J. A. Piñuela-Izquierdo and D. Andina.
Publication: Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, Proceedings, Part II.
Pages: 399–408.
Place: Elche, Spain.
Year: 2015.
8. **Congress:** IWINAC 15 - International Work-conference on the Interplay between Natural and Artificial Computation.
Title: Artificial metaplasticity: Application to MIT-BIH arrhythmias database.
Authors: S. Torres-Alegre, J. Fombellida, J. A. Piñuela-Izquierdo and D. Andina.
Publication: Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, Proceedings, Part I.
Pages: 133–142.
Place: Elche, Spain.
Year: 2015.
9. **Congress:** WAC 2016 - World Automation Congress.
Title: Application of artificial metaplasticity fundamentals to WBCD breast cancer database classification method.
Authors: J. Fombellida, I. Martin-Rubio and D. Andina
Publication: World Automation Congress Proceedings.

Place: Rio Grande, Puerto Rico.

Year: 2016.

10. **Congress:** IWINAC 2017 - International Work-conference on the Interplay between Natural and Artificial Computation.

Title: Koniocortex-like network unsupervised learning surpasses supervised results on WBCD breast cancer database.

Authors: J. Fombellida and F. J. Ropero-Peláez and D. Andina.

Publication: Biomedical Applications Based on Natural and Artificial Computing: International Work-Conference on the Interplay Between Natural and Artificial Computation, Proceedings, Part II.

Pages: 32–41.

Place: Coruña, Spain.

Year: 2017.

11. **Congress:** IWINAC 2017 - International Work-conference on the Interplay between Natural and Artificial Computation.

Title: Supervised metaplasticity for big data: Application to pollutant concentrations forecast.

Authors: J. Fombellida and M. J. Alarcon and S. Torres-Alegre and D. Andina.

Publication: Biomedical Applications Based on Natural and Artificial Computing: International Work-Conference on the Interplay Between Natural and Artificial Computation, Proceedings, Part II.

Pages: 374–383.

Place: Coruña, Spain.

Year: 2017.

References

- [1] D. E. Rumelhart, J. J. McClelland, and the PDP Research Group. Explorations in the microstructure of cognition: Foundations. *Parallel distributed processing in the microstructure in cognition*, 1:216–271, 1986.
- [2] S. Haykin. Neural networks: a comprehensive foundation. *Prentice-Hall*, 1994.
- [3] B. B. Chaudhuri and U. Bhattacharya. Efficient training and improved performance of multilayer perceptron in pattern classification. *Neurocomputing*, 34(4):11–27, 2000.
- [4] Z. Dokur and T. Ölmez. Heart sound classification using wavelet transform and incremental self-organizing map. *Digital Signal Processing*, 18:951–959, 2008.
- [5] K. Lee, D. Booth, and P. Alam. A comparison of supervised and unsupervised neural networks in predicting bankruptcy of korean firms. *Expert Systems with Applications*, 2:1–16, 2005.
- [6] W. C. Abraham. Activity-dependent regulation of synaptic plasticity (metaplasticity) in the hippocampus. *The Hippocampus: Functions and Clinical Relevance*, pages 15–26, 1996.
- [7] D. Parker and S. Grillner. Activity-dependent metaplasticity of inhibitory and excitatory synaptic transmission in the lamprey spinal cord locomotor. *The Journal of Neuroscience*, 19(5):1647–1656, 1999.
- [8] T. C. Thiagarajana, M. Lindsdoga, A. Malgarolib, and R. W. Tsiena. LTP and adaptation to inactivity: Overlapping mechanisms and implications for metaplasticity. *Neuropharmacology*, 52(1):156–175, 2007.
- [9] J. Young, C. Isiegas, T. Abel, and P. V. Nguyen. Metaplasticity of the latephase of long-term potentiation a critical role for protein kinase a in synaptic tagging. *European Journal of Neuroscience*, 23:1784–1794, 2006.
- [10] X. Yu, H. Z. Shouval, and J. J. Knierim. A biophysical model of synaptic plasticity and metaplasticity can account for the dynamics of the backward shift of hippocampal place fields. *Journal of Neurophysiology*, 100:983–992, 2008.
- [11] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.
- [12] J. Ropero-Pelaez and J. R. Piqueira. Biological clues for up to date artificial neurons. *Computational Intelligence for Engineering and Manufacturing*, 2007.

- [13] D. Andina, A. Álvarez Vellisco, A. Jevtíc, and J. Fombellida. Artificial metaplasticity can improve artificial neural network learning. *Intelligent Automation and Soft Computing, Special Issue in Signal Processing and Soft Computing*, 15(4):681–649, 2009.
- [14] J. Ropero-Pelaez and D. Andina. On the biological plausibility of artificial metaplasticity learning algorithm. *Neurocomputing*, 114:32–35, 2012.
- [15] J. Fombellida, I. Martín-Rubio, and D. Andina. Application of artificial metaplasticity fundamentals to WBCD breast cancer database classification method. *World Automation Congress Proceedings*, 2016.
- [16] J. Leonard and M. A. Kramer. Improvement of the backpropagation algorithm for training neural networks. *Computers and Chemical Engineering*, 14(3):337–341, 1990.
- [17] M. Lee and S. T. Weidman. Expert systems for guiding backpropagation training of layered perceptrons. *Expert Systems with Applications*, 2(1):73–81, 1991.
- [18] J. B. Ra and Y. K. Kim. Weight value initialization for improving training speed in the backpropagation network. *Proc. IEEE International Joint Conference on Neural Networks, IJCNN'91*, 3:2396–2401, 1991.
- [19] N. Tepedelenlioglu and R. S. Scalero. A fast new algorithm for training feedforward neural networks. *IEEE Transactions on Signal Processing*, 40(1):202–210, 1992.
- [20] N. B. Karayiannis and A. N. Venetsanopoulos. Fast learning algorithms for neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(7):453–474, 1992.
- [21] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.
- [22] D. S. Chen and R. C. Jain. A robust backpropagation learning algorithm for function approximation. *IEEE Transactions on Neural Networks*, 5(3):467–479, 1994.
- [23] D. Alpsan, M. Towsey, Z. Ozdamar, A. C. Tsoi, and D. N. Ghista. Efficacy of modified backpropagation and optimisation methods on a real-world medical problem. *Neural Networks*, 8(6):945–962, 1995.
- [24] R. Solomon and J. L. Van-Hemmen. Accelerating backpropagation through dynamic self-adaptation. *Neural Networks*, 9(4):589–601, 1996.
- [25] L. Fu, H. H. Hsu, and J. C. Principe. Incremental backpropagation learning networks. *IEEE Transactions on Neural Networks*, 7(3):757–761, 1996.
- [26] G. D. Magoulas and M. N. Vrahatis. Effective backpropagation training with variable stepsize neural networks. *Neural Networks*, 10(1):69–82, 1997.
- [27] Y. F. Yam, T. W. Chow, and C. T. Leung. A new method in determining initial weights of feedforward neural networks for training enhancement. *Neurocomputing*, 16(1):23–32, 1997.

- [28] R. S. Sexton, R. E. Dorsey, and J. D. Johnson. Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation. *Decision Support Systems*, 22(2):171–185, 1998.
- [29] S. V. Kamarthi and S. Pittne. Accelerating neural network training using weight extrapolations. *Neural Networks*, 12(9):1285–1299, 1999.
- [30] S. Y. Cho and T. W. Chow. Training multilayer neural networks using fast global learning algorithm - least-squares and penalized optimization methods. *Neurocomputing*, 25(1):115–131, 1999.
- [31] K. Ampazisa, S. J. Perantonisa, and J. G. Taylor. Dynamics of multilayer networks in the vicinity of temporary minima. *Neural Networks*, 12(1):43–58, 1999.
- [32] Y. F. Yam and T. W. Chow. A weight initialization method for improving training speed in feedforward neural network. *Neurocomputing*, 30(1):219–232, 2000.
- [33] H. M. Lee, C. M. Chen, and T. C. Huang. Learning efficiency improvement of back-propagation algorithm by error saturation prevention method. *Neurocomputing*, 41:125–143, 2001.
- [34] M. Mandische. A comparison of evolution strategies and backpropagation for neural network training. *Neurocomputing*, 42:87–117, 2002.
- [35] K. A. Hoo, E. D. Sinzinger, and M. J. Piovoso. Improvements in the predictive capability of neural networks. *Journal of Process Control*, 12(1):193–202, 2002.
- [36] K. Eom, K. Jung, and H. Sirisena. Performance improvement of backpropagation algorithm by automatic activation function gain tuning using fuzzy logic. *Neurocomputing*, 50:439–460, 2003.
- [37] Y. H. Zweiri, J. F. Whidborne, and L. D. Seneviratne. A three-term backpropagation algorithm. *Neurocomputing*, 50:305–318, 2003.
- [38] A. Abraham. Meta learning evolutionary artificial neural networks. *Neurocomputing*, 56:1–38, 2004.
- [39] X. G. Wang, Z. Tang, H. Tamura, M. Ishii, and W. D. Sun. An improved backpropagation algorithm to avoid the local minima problem. *Neurocomputing*, 56:455–460, 2004.
- [40] A. V. Pernía-Espinoza, J. B. Ordieres-Meré, F. J. Martínez de Pisón, and A. González-Marcos. TAO-robust backpropagation learning algorithm. *Neural Networks*, 18(2):191–204, 2005.
- [41] J. J. Steil. Online stability of backpropagation-decorrelation recurrent learning. *Neurocomputing*, 69(7-9):191–204, 2006.
- [42] L. Behera, S. Kumar, and A. Patnaik. On adaptive learning rate that guarantees convergence in feedforward networks. *IEEE Transactions on Neural Networks*, 17(5):1116–1125, 2006.

- [43] C. H. Wang, C. H. Kao, and W. H. Lee. A new interactive model for improving the learning performance of back propagation neural network. *Automation in Construction*, 16(6):745–758, 2007.
- [44] A. Khashman. A modified backpropagation learning algorithm with added emotional coefficients. *IEEE Transactions on Neural Networks*, 19(11):1896–1909, 2008.
- [45] S. S. Yang, S. Siu, and C. L. Ho. Analysis of the initial values in split-complex backpropagation algorithm. *IEEE Transactions on Neural Networks*, 19(9):1564–1573, 2008.
- [46] M. I. Soliman and S. A. Mohamed. A highly efficient implementation of a backpropagation learning algorithm using matrix ISA. *Journal of Parallel and Distributed Computing*, 68(7):949–961, 2008.
- [47] H. L. Cheng and S. C. Park. An efficient document classification model using an improved back propagation neural network and singular value decomposition. *Expert Systems with Applications*, 36(2):3208–3215, 2009.
- [48] T. Kathirvalavakumar and S. Jeyaseeli-Subavathi. Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks. *Neurocomputing*, 72:3915–3921, 2009.
- [49] Y. Bai, H. Zhang, and Y. Hao. The performance of the backpropagation algorithm with varying slope of the activation function. *Chaos, Solitons and Fractals*, 40(1):69–77, 2009.
- [50] Y. Bai, H. Zhang, and Y. Hao. Local coupled feedforward neural network. *Neural Networks*, 23(1):108–113, 2010.
- [51] H. H. Örkücü and H. Bal. Comparing performances of backpropagation and genetic algorithms in the data classification. *Expert Systems with Applications*, 38(4):3703–3709, 2011.
- [52] M. Z. Rehman, N. M. Nawi, and M. I. Ghazali. Noise-Induced Hearing Loss (NIHL) prediction in humans using a modified back propagation neural network. *International Journal on Advanced Science, Engineering and Information Technology*, 1(2):185–189, 2011.
- [53] R. R. Srikant, P. V. Krishna, and N.D. Rao. Online tool wear prediction in wet machining using modified back propagation neural network. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(7):1009–1018, 2011.
- [54] M. Z. Rehman and N. M. Nawi. The effect of adaptive momentum in improving the accuracy of gradient descent back propagation algorithm on classification problems. *International Conference on Software Engineering and Computer Systems ICSECS 2011: Software Engineering and Computer Systems*, pages 380–390, 2012.

- [55] J. Li, J. Cheng, J. Shi, and F. Huang. Advances in computer science and information engineering. *International Conference on Software Engineering and Computer Systems ICSECS 2011: Software Engineering and Computer Systems*, 2:553–558, 2012.
- [56] S. Sholanki and H. B. Jethva. Modified back propagation algorithm of feed forward networks. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(6):2278–3075, 2013.
- [57] N. N. Mohd, A. Khan, and M. Z. Rehman. A new back-propagation neural network optimized with cuckoo search algorithm. *Computational Science and Its Applications – ICCSA 2013: 13th International Conference*, pages 413–426, 2013.
- [58] A. Kavousi-Fart, T. Nikman, and M. Golmaryami. Short term load forecasting of distribution systems by a new hybrid modified FA-backpropagation method. *Journal of Intelligent and Fuzzy Systems*, 26(1):517–522, 2014.
- [59] N. Salari, S. Shohaimi, M. Nallappan, and I. Karishnarajah. A novel hybrid classification model of genetic algorithms, modified k-nearest neighbor and developed backpropagation neural network. *PLOS One*, 9(11), 2014.
- [60] H. Huang, J. Li, and C. Xiao. A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm. *Expert Systems with Applications*, 42(1):146–155, 2015.
- [61] J. Kostencka and T. Kozacki. Space-domain, filtered backpropagation algorithm for tomographic configuration with scanning of illumination. *Proc. SPIE 9890, Optical Micro- and Nanometrology*, 6, 2016.
- [62] S. Kumar, R. N. Verma, and A. Kumar. Modified mean square error with regularization algorithm for efficient classification of patterns in back-propagation neural network. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, 5(5):25–29, 2016.
- [63] U. N. Wisesty, N. Jondri, and Adiwijaya. Modified backpropagation algorithm for polycystic ovary syndrome detection based on ultrasound images. *Recent Advances on Soft Computing and Data Mining: The Second International Conference on Soft Computing and Data Mining (SCDM-2016)*, pages 141–151, 2016.
- [64] H. Luo, J. Fu, and J. Glass. Bidirectional backpropagation: Towards biologically plausible error signal transmission in neural networks. *Neural and Evolutionary Computing*, 2017.
- [65] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen. Neural networks for vector quantization of speech and images. *IEEE Journal on Selected Areas in Communications*, 8(8):1449–1457, 1990.
- [66] T. L. Huntsberger and P. Ajjimarangsee. Parallel self-organizing feature maps for unsupervised pattern recognition. *International Journal of General Systems*, 16(4):357–372, 1990.

- [67] C. T. Lin and C. S. G. Lee. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12):1320–1336, 1991.
- [68] V. S. Rose, I. F. Croall, and H. J. H. Macfie. An application of unsupervised neural network methodology kohonen topology-preserving mapping to QSAR analysis. *Molecular Informatics*, 10(1):6–15, 1991.
- [69] L. Burke and S. Rangwala. Tool condition monitoring in metal cutting: A neural network approach. *Journal of Intelligent Manufacturing*, 2(5):269–280, 1991.
- [70] N. Intrator. Feature extraction using an unsupervised neural network. *Neural Computation*, 4(1):98–107, 1992.
- [71] T. Samad and S. A. Harp. Self-organization with partial data. *Network: Computation in Neural Systems*, 3(2):205–212, 1992.
- [72] L. Xu. Least mean square error reconstruction principle for self-organizing neural-nets. *Neural Networks*, 6(5):627–648, 1993.
- [73] P. V. Balakrishnan, M. C. Cooper, S. J. Varghese, and P. A. Lewis. A study of the classification capabilities of neural networks using unsupervised learning: A comparison with k-means clustering. *Psychometrika*, 59(4):509–525, 1994.
- [74] G. Rigoll. Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems. *IEEE Transactions on Speech and Audio Processing*, 2(1):175–184, 1994.
- [75] G. A. Carpenter and W. D. Ross. Art-emap: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, 6(4):805–818, 1995.
- [76] B. Malakooti and Z. Yang. A variable-parameter unsupervised learning clustering neural network approach with application to machine-part group formation. *International Journal of Production Research*, 33(9):2395–2413, 1995.
- [77] S. C. A. Thomopoulos, D. K. Bougoulas, and C Wann. Dignet: an unsupervised-learning clustering algorithm for clustering and data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 31(1):21–38, 1995.
- [78] S. Becker and M. Plumbley. Unsupervised neural network learning procedures for feature extraction and classification. *Applied Intelligence*, 6(3):185–203, 1996.
- [79] A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 43(11):894–906, 1996.
- [80] Y. Zheng, J. F. Greenleaf, and J. J. Gisvold. Reduction of breast biopsies with a modified self-organizing map. *IEEE Transactions on Neural Networks*, 8(6):1386–1396, 1997.
- [81] D. G. Roussinov and H. Chen. A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. *Communication and Cognition in Artificial Intelligence Journal*, 15(1-2):81–111, 1998.

- [82] D. Srinivasan, S. S. Tan, C. S. Chang, and E. K. Chan. Practical implementation of a hybrid fuzzy neural network for one-day-ahead load forecasting. *IEEE Proceedings - Generation, Transmission and Distribution*, 145(6):687–692, 1998.
- [83] Y. S. Kim. An unsupervised neural network using a fuzzy learning rule. *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, 1999.
- [84] C. I. Christodoulou and C. S. Pattichis. Unsupervised pattern recognition for the classification of emg signals. *IEEE Transactions on Biomedical Engineering*, 46(2):169–178, 1999.
- [85] P. Törönen, M. Kolehmainen, G. Wong, and E. Castrén. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451(2):142–146, 1999.
- [86] T. Chon, Y. Park, and J. H. Park. Determining temporal pattern of community dynamics by using unsupervised learning algorithms. *Ecological Modelling*, 132(1-2):151–166, 2000.
- [87] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *Computer Vision - ECCV 2000: 6th European Conference on Computer Vision Dublin*, pages 18–32, 2000.
- [88] G. A. Cecchi, L. T. Petreanu, A. Alvarez-Buylla, and M. O. Magnasco. Unsupervised learning and adaptation in a model of adult neurogenesis. *Journal of Computational Neuroscience*, 11(2):175–182, 2001.
- [89] M. W. Spratling and M. H. Johnson. Preintegration lateral inhibition enhances unsupervised learning. *Neural Computation*, 14(9):2157–2179, 2002.
- [90] S. M. Bohte, H. La Poutre, and J. N. Kok. Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks. *IEEE Transactions on Neural Networks*, 13(2):426–435, 2002.
- [91] E. Papageorgiou, C. Stylios, and P. Groumpos. Fuzzy cognitive map learning based on nonlinear Hebbian rule. *Lecture Notes in Computer Science*, 2903:256–268, 2003.
- [92] T. Anastasio and P. E. Patton. A two-stage unsupervised learning algorithm reproduces multisensory enhancement in a neural network model of the corticotectal system. *Journal of Neuroscience*, 23(17):6713–6727, 2003.
- [93] N. Enfadil and D. Isa. Automated knowledge acquisition based on unsupervised neural network and expert system paradigms. *Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003, Oxford, UK, September 2003. Proceedings, Part I*, 2773:134–140, 2003.
- [94] J. Lücke and C. von der Malsburg. Rapid processing and unsupervised learning in a model of the cortical macrocolumn. *Neural Computation*, 16(3):501–533, 2004.
- [95] A. Arleo, F. Smeraldi, and W. Gerstner. Cognitive navigation based on non uniform gabor space sampling, unsupervised growing networks, and reinforcement learning. *IEEE Transactions on Neural Networks*, 15(3):639–652, 2004.

- [96] J. Lücke and C. von der Malsburg. Particle swarm optimization for unsupervised robotic learning. *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005.
- [97] R. J. Kuo, Y. L. An, H. S. Wang, and W. J. Chung. Particle swarm optimization for unsupervised robotic learning. *Expert Systems with Applications*, 30(2):313–324, 2006.
- [98] E. I. Papageorgioua, C. Styliosb, and P. P. Groumposa. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *International Journal of Human-Computer Studies*, 64(8):727–743, 2006.
- [99] J. Zhang and M. Zulkernine. Anomaly based network intrusion detection with unsupervised outlier detection. *IEEE International Conference on Communications, 2006. ICC '06*, 2006.
- [100] T. Masquelier and S. J. Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput Biol*, 3(2), 2007.
- [101] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 2007.
- [102] H. Lee, E. Chaitanya, and Y. Andrew. Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems*, 20:873–880, 2008.
- [103] A. Tan, N. Lu, and D. Xiao. Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. *IEEE Transactions on Neural Networks*, 19(2):230–244, 2008.
- [104] A. W. Tsui, Y. Chuang, and H. Chu. Unsupervised learning for solving RSS hardware variance problem in wifi localization. *Mobile Networks and Applications*, 14(5):677–691, 2009.
- [105] F. Shen and O. Hasegawa. Self-organizing incremental neural network and its application. *Artificial Neural Networks – ICANN 2010: 20th International Conference*, pages 535–540, 2010.
- [106] F. Chang, L. Chang, H. Kao, and a G. Wu. Assessing the effort of meteorological variables for evaporation estimation by self-organizing map neural network. *Journal of Hydrology*, 384(1-2):118–129, 2010.
- [107] F. Chang, L. Chang, H. Kao, and a G. Wu. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.
- [108] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161, 2011.

- [109] F. Fernández-Navarro, C. Hervás-Martínez, J. Sanchez-Monedero, and P. A. Gutiérrez. MELM-GRBF: A modified version of the extreme learning machine for generalized radial basis function neural networks. *Neurocomputing*, 74(16):2502–2510, 2011.
- [110] W. Hsu. Application of competitive Hopfield neural network to brain-computer interface systems. *International Journal of Neural Systems*, 22(51):51–62, 2012.
- [111] P. Boniecki, K. Nowakowski, R. Tomczak, S. Kujawa, and H. Piekarska-Boniecka. The application of the Kohonen neural network in the nonparametric-quality-based classification of tomatoes. *Proc. SPIE 8334, Fourth International Conference on Digital Image Processing (ICDIP 2012)*, 2012.
- [112] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun. Pedestrian detection with unsupervised multi-stage feature learning. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3626–3633, 2013.
- [113] Y. Cui, S. Ahmada, and J. Hawkins. Continuous online sequence learning with an unsupervised neural network model. *Neural and Evolutionary Computing (cs.NE); Neurons and Cognition (q-bio.NC)*, 2014.
- [114] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez. Continuous online sequence learning with an unsupervised neural network model. *Information Sciences*, 304, 2014.
- [115] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2015.
- [116] S. R. Kheradpisheha, M. Ganjtabesh, and T. Masquelier. Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition. *Neurocomputing*, 205:382–392, 2016.
- [117] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*, pages 527–544, 2016.
- [118] S. Roy and A. Basu. An online unsupervised structural plasticity algorithm for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):900–910, 2017.
- [119] D. Andina and J. Fombellida. Metaplasticity artificial neural networks model application to radar detection. *Journal of Systemics, Cybernetics and Informatics*, 5(6):44–48, 2007.
- [120] WBCD. [https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+\(diagnostic\)](https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)). *Wisconsin Breast Cancer DataBase*, 2013.
- [121] J. Fombellida, S. Torres-Alegre, J. A. Piñuela-Izquierdo, and D. Andina. Artificial metaplasticity for deep learning: Application to WBCD breast cancer database classification. *Bioinspired Computation in Artificial Systems: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part II*, pages 399–408, 2015.

- [122] J. Fombellida, F. J. Ropero-Peláez, and D. Andina. Koniocortex-like network unsupervised learning surpasses supervised results on WBCD breast cancer database. *International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2017, Corunna, Spain, June 19-23, 2017, Proceedings, Part II*, pages 32–41, 2017.
- [123] ACAD. [https://archive.ics.uci.edu/ml/datasets/statlog+\(australian+credit+approval\)](https://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)). *Australian Credit Approval Database*, 2013.
- [124] J. Fombellida, I. Martín-Rubio, S. Torres-Alegre, and D. Andina. Tackling Business Intelligence with bioinspired deep learning. *Neural Computing and Applications*, Accepted, 2018.
- [125] J. Fombellida, M. J. Alarcón, S. Torres-Alegre, and D. Andina. Supervised metaplasticity for big data: Application to pollutant concentrations forecast. *International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2017, Corunna, Spain, June 19-23, 2017, Proceedings, Part II*, pages 374–383, 2017.
- [126] D. O. Hebb. The organization of behavior. *Laurence Erlbaum Associates*, 2002 (reedition of the 1949 original), 1949.
- [127] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical Biophysics*, 5:115–133, 1943.
- [128] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [129] B. Widrow. An adaptive ‘adaline’ neuron using chemical ‘memistors’. *Stanford Electronics Laboratories Technical Report*, 1553(2), 1960.
- [130] M. Minsky and S. Papert. Perceptrons: An introduction to computational geometry. *MIT Press, Cambridge, MA*, 1969.
- [131] S. Grossberg. Adaptive pattern classification and universal recoding: II. feedback, expectation, olfaction, illusions. *Biological Cybernetics*, 23(4):187–202, 1976.
- [132] M. Minsky and S. Papert. Brain modeling and memory. *Proceedings of the First International Conference on Mathematical Modeling*, 1977.
- [133] T. Kohonen, P. Lehtiö, J. Rovamo, J. Hyvärinen, K. Bry, and L. Vainio. A principle of neural associative memory. *Neuroscience*, 2(6):1065–1076, 1977.
- [134] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- [135] DARPA. Darpa neural network study. *AFCEA International Press*, page 60, 1988.
- [136] T. Kohonen. The self-organizing map. *Neurocomputing*, 21:1–6, 1998.
- [137] D. Andina and D. T. Pham. Computational intelligence for engineering and manufacturing. *Springer-Verlag, The Netherlands*, 2007.

- [138] J. M. Mendel and R. W. McLaren. Reinforcement-learning control and pattern recognition systems. *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, pages 287–318, 1970.
- [139] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 3(4):4–22, 1987.
- [140] J. M. Zurada. Introduction to artificial neural systems. *West Publishing Company USA*, 1999.
- [141] M. H. Hassoun. Fundamentals of artificial neural networks. *MIT Press, Cambridge, MA*, 1995.
- [142] A. K. Jain, J. Mao, and K. M. Mohiuddin. Artificial neural networks: a tutorial. *Comput. IEEE March*, pages 31–44, 1996.
- [143] D. Andina. Optimización de detectores neuronales. aplicación a radar y sonar. *Tesis Doctoral, ETSIT, Universidad Politécnica de Madrid*, 1995.
- [144] P. J. Werbos. Backpropagation through time: What it does and how to do it. *IEEE Proceedings*, 78(10):1550–1560, 1990.
- [145] D. Parker. A comparison of algorithms for neuron-like cells. *AIP Proc.: Neural Networks for Computing*, 151:327–332, 1986.
- [146] J. R. Hilera and V. J. Martínez. Redes neuronales artificiales: fundamentos, modelos y aplicaciones. *RA-MA. Madrid.*, 1995.
- [147] G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
- [148] B. Widrow and M. E. Hoff. Associative storage and retrieval of digital information in network of adaptive "neurons". *Biological Prototypes and Sinthetic Systems*, 1, 1989.
- [149] P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Harvard University Unpublished doctoral dissertation*, 1974.
- [150] V. V. Phansalkar and P. S. Sastry. Analysis of the back-propagation algorithm with momentum. *IEEE Trans Neural Network*, 5(3):505–506, 1994.
- [151] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63, 1987.
- [152] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [153] G. A. Ojemann. Brain organization for language from the perspective of electrical stimulation mapping. *Behav. Brain Sci*, 2:189–230, 1983.
- [154] E. I. Knudsen, S. du Lac, and S. D. Esterly. Computational maps in the brain. *Ann. Rev. Neurosci*, 10:41–65, 1983.

- [155] M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks 2000*, 2000.
- [156] J. Blackmore and R. Miikkulainen. Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map. *IEEE International Conference on Neural Networks 1993*, 1993.
- [157] B. Fritzke. Growing grid: a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2(5):9–13, 1995.
- [158] R. M. Gray. Vector quantization. *IEEE ASSP Mag*, 1:4–29, 1984.
- [159] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization. *IEEE Trans. Communication*, 28:84–95, 1984.
- [160] Ch. v. d. Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100, 1973.
- [161] J. Kangas, T. Kohonen, and J. Laaksonen. Variants of self-organizing maps. *IEEE Trans. Neural Networks*, 1:93–99, 1990.
- [162] J. Kangas, T. Kohonen, and J. Laaksonen. Combining self-organizing maps. *Proc. Int./Joint Conf. on Neural Networks, IJCNN 89*, 2:499–502, 1989.
- [163] T. Kohonen. Learning vector quantization. *Helsinki University of Technology, Laboratory of Computer and Information Science, Report TKK-F-A-60*, 1986.
- [164] T. Kohonen. An introduction to neural computing. *Neural Networks*, 1(1):3–16, 1988.
- [165] T. Kohonen, C. Barna, and R. Chrisley. Statistical pattern recognition with neural networks: Benchmarking studies. *Proc. IEEE Int. Conf. on Neural Networks, ICNN-88*, 1:61–68, 1988.
- [166] G. A. Carpenter and S. Grossberg. Pattern recognition by self-organizing neural networks. *MIT Press, Cambridge, Mass*, 1991.
- [167] G. A. Carpenter and S. Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88, 1988.
- [168] S. Grossberg. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121–134, 1976.
- [169] G. A. Carpenter and S. Grossberg. ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, pages 4919–4930, 1987.
- [170] W. C. Abraham. Metaplasticity: Key element in memory and learning? *News in Physiological Sciences*, 14(2):85, 1999.
- [171] S. R. Cajal. La fine structure des centres nerveux. *Proceedings of Royal Society*, 55:444–468, 1894.

- [172] J. Konorski. Conditioned reflexes and neuron organization. *Cambridge University Press, Cambridge*, 1948.
- [173] R. Malenka and M. Bear. LTP and LTD : An embarrassment of riches. *Neuron*, 44(1):5–21, 2004.
- [174] P. Jedlicka. Synaptic plasticity, metaplasticity and BCM theory. *Institute of Pathophysiology, Medical Faculty, Comenius University*, 103(4):137–143, 2002.
- [175] K. Todd, A. Serrano, J. Lacaille, and R. Robitaille. Glial cells in synaptic plasticity. *Journal of Physiology*, 99:75–83, 2007.
- [176] T. V. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232(2):331–356, 1973.
- [177] G. Barrionuevo, F. Schotteler, and G. Lynch. The effects of repetitive low frequency stimulation on control and potentiated synaptic responses in the hippocampus. *Life Sci*, 27:2385–2391, 1980.
- [178] P. V. Massey and Z. B. F. Bashir. Long-term depression: multiple forms and implications for brain function. *Trends in Neurosciences*, 30(4):176–184, 2007.
- [179] M. Chiappalone, A. Vato, L. Berdondini, M. Koudelka, and S. Martinoia. Network dynamics and synchronous activity in cultured cortical neurons. *Int. J. Neural Syst.*, 17(2):87–103, 2007.
- [180] W. C. Abraham. Metaplasticity: tuning synapses and networks for plasticity. *Nature Reviews Neuroscience*, 9:387–399, 2008.
- [181] B. Mockett, C. Coussens, and W. C. Abraham. NMDA receptor-mediated metaplasticity during the induction of long-term depression by low-frequency stimulation. *Eur. J. Neurosci.*, 15:1819–1826, 2008.
- [182] N. S. Desai. Homeostatic plasticity in the CNS: synaptic and intrinsic forms. *J. Physiol.*, 97:391–402, 2003.
- [183] S. J. Martin, P. D. Grimwood, and R. G. Morris. Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual Review of Neuroscience*, 23:649–711, 2000.
- [184] G. Neves, S. F. Cooke, and T. V. Bliss. Synaptic plasticity, memory and the hippocampus: a neural network approach to causality. *Nature Rev. Neurosci.*, 9:65–75, 2008.
- [185] E. Kinto, E. Del-Moral-Hernández, A. Marcano-Cerdeño, and J. Ropero-Peláez. A preliminary neural model for movement direction recognition based on biologically plausible plasticity rules. *Proc. IWINAC 2007, Lecture Notes in Computer Science*, 4528:628–636, 2008.
- [186] J. L. Monteiro, M. Lobo-Netto, D. Andina, and J. Ropero-Peláez. Using neural networks to simulate the alzheimer’s disease. *Proc. WAC 2008, World Automation Congress*, pages 1–6, 2008.

- [187] H. Kahn. Use of different monte carlo sampling techniques. *Santa Monica, CA: RAND Corporation*, 1955.
- [188] J. R. Peláez and D. Andina. Do biological synapses perform probabilistic computations? *Neurocomputing*, 114:24–31, 2012.
- [189] A. A. Minai. Covariance learning of correlated patterns in competitive networks. *Neural Comput.*, 9:667–681, 1997.
- [190] A. Artola and W. Singer. Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. *Trends Neurosci.*, 16:480–487, 1993.
- [191] F. J. Ropero-Peláez and D. Andina. Probabilistic versus incremental presynaptic learning in biologically plausible synapses. *Found. Nat. Artif. Comput. LNCS 6686*, pages 80–89, 2011.
- [192] D. W. Rucky, S. K. Rogers, M. Kabrisk, M. E. Oxley, and B. W. Suter. The multi-layer perceptron as an approximation to a bayes optimal discrimination function. *IEEE Trans. Neural Network*, 1(4):296–298, 1990.
- [193] F. J. Ropero-Peláez and D. Andina. The koniocortex-like network: A new biologically plausible unsupervised neural network. *Artificial Computation in Biology and Medicine: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, June 1-5, 2015, Proceedings, Part I*, pages 163–174, 2015.
- [194] D. W. Rucky, S. K. Rogers, M. Kabrisk, M. E. Oxley, and B. W. Suter. Intrinsic plasticity for natural competition in koniocortex-like neural networks. *International Journal of Neural Systems*, 26(5), 2016.
- [195] D. Fester, S. Chung, and H. Wheat. Orientation selectivity of thalamic input to simple cells of cat visual cortex. *Nature*, 380(6571):249–252, 1996.
- [196] K. D. Miller. Synaptic economics: Competition and cooperation in synaptic plasticity. *Neuron*, 17:371–374, 1996.
- [197] S. Kaski and T. Kohonen. Winner-take-all networks for physiological models of competitive learning. *Neural networks*, 7(6/7):973–984, 1994.
- [198] T. Fukai and S. Tanaka. A simple neural network exhibiting selective activation of neuronal ensembles: From winner-take-all to winners-share-all. *Neural Comput.*, 9(1):77–97, 1997.
- [199] Z. H. Mao and S. G. Massaquoi. Dynamics of winner-take-all competition in recurrent neural networks with lateral inhibition. *IEEE Trans. Neural Netw.*, 18:55–69, 2007.
- [200] J. F. Yang and C. M. Chen. Winner-take-all neural network using the highest threshold. *IEEE Trans. Neural Netw.*, 11:194–199, 2000.
- [201] W. C. Abraham and M. F. Bear. Metaplasticity: the plasticity of synaptic plasticity. *Trends Neurosci.*, 19:126–130, 1996.

- [202] W. C. Abraham and W. P. Tate. Metaplasticity: a new vista across the field of synaptic plasticity. *Prog Neurobiol.*, 52:303–323, 1997.
- [203] N. S. Desai, L. C. Rutherford, and G. G. Turrigiano. Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nat Neurosci.*, 2:515–520, 1999.
- [204] R. P. N. Rao. An optimal estimation approach to visual perception and learning. *Vision Res.*, 39:1963–1989, 1999.
- [205] F. Hurliman, D. C. Kiper, and M. Carandini. Testing the bayesian model of perceived speed. *Vision Res.*, 42:2253–2257, 1999.
- [206] K. Kording and D. M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427:244–246, 2004.
- [207] A. Soltani and X. Wang. Synaptic computation underlying probabilistic inference. *Nat. Neurosci.*, 13:112–119, 2010.
- [208] V. E. Okhotin. Cytophysiology of spiny stellate cells in the striate cortex and their role in the excitatory mechanism of intracortical synaptic circulation. *Neurosci. Behav. Physiol.*, 26(8):825–836, 2006.
- [209] B. Strack, K. Jacobs, and K. J. Cios. Simulating vertical and horizontal inhibition with short term dynamics in a multi-column multi-layer model of neocortex. *Int J Neural Syst.*, 24(5):825–836, 2014.
- [210] J. Neyman and E. S. Pearson. Joint statistical papers. *University of California Press*, 1966.
- [211] J. I. Marcum. A statistical theory of target detection by pulsed radar. *Mathematical Appendix IRE Trans.*, IT-6:59–267, 1960.
- [212] W. Al-Nuaimy, Y. Huang, M. Nakhkash, M.T.C. Fang, V.T. Nguyen, A. Eriksen, and J. I. Marcum. Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition. *Journal of Applied Geophysics*, 43:157–165, 2000.
- [213] E. Granger, M. A. Rubin, S. Grossbert, and P. Lavoie. Classification of incomplete data using the fuzzy ARTMAP neural network. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000*, pages 35–40, 2000.
- [214] A. Patnaik and R. K. Mishra. ANN techniques in microwave engineering. *IEEE Microwave Magazine*, 1(1):55–60, 2000.
- [215] Y. Shi and X. Zhang. A gabor atom network for signal classification with application in radar target recognition. *IEEE Transactions on Signal Processing*, 49(12):2994–3004, 2001.
- [216] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy. Neural methods for antenna array signal processing: a review. *Signal Processing*, 82:547–561, 2002.
- [217] C. S. Shieh and C. T. Lin. A vector neural network for emitter identification. *IEEE Transactions on Antennas and Propagation*, 50(8):1120–1127, 2002.

- [218] J. H. Lee, I. S. Choi, and H. T. Kim. Natural frequency-based neural network approach to radar target recognition. *IEEE Transactions on Signal Processing*, 51(12):3191–3197, 2003.
- [219] T. McConaghy, H. Leung, E. Bossé, and V. Varadan. Classification of audio radar signals using radial basis function neural networks. *IEEE Transactions on Instrumentation and Measurement*, 52(6):1771–1779, 2003.
- [220] L. Bruzzone, M. Marconcini, U. Wegmüller, and A. Wiesmann. An advanced system for the automatic classification of multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 42(6):1321–1333, 2004.
- [221] J. Karvonen. Baltic sea ice SAR segmentation and classification using modified pulse-coupled neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1566–1574, 2004.
- [222] E. Avci, I. Turkoglua, and M. Poyraz. Intelligent target recognition based on wavelet packet neural network. *Expert Systems with Applications*, 29:175–182, 2005.
- [223] M. R. Shaw, S. G. Millard, T. C. K. Molyneaux, M. J. Taylor, and J. H. Bungey. Location of steel reinforcement in concrete using ground penetrating radar and neural networks. *NDTE International*, 38:203–212, 2005.
- [224] E. Radoi, A. Quinquis, and F. Totir. Supervised self-organizing classification of superresolution ISAR images: An anechoic chamber experiment. *EURASIP Journal on Applied Signal Processing*, pages 1–14, 2006.
- [225] C. Ting, L. Jingqing, and S. Bing. Research on rough set-neural network and its application in radar signal recognition. *Eighth International Conference on Electronic Measurement and Instruments*, 3:764–768, 2007.
- [226] M. Ren, J. Cai, Y. Zhu, and M. He. Radar emitter signal classification based on mutual information and fuzzy support vector machines. *ICSP2008 Proceedings*, pages 1641–1646, 2008.
- [227] M. A. Rico-Ramirez and D. Cluckie. Classification of ground clutter and anomalous propagation using dual-polarization weather radar. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):1892–1904, 2008.
- [228] N. K. Ibrahim, R. S. A. Raja-Abdullah, and M. I. Saripan. Artificial neural network approach in radar target classification. *Journal of Computer Science*, 5(1):23–32, 2009.
- [229] Z. M. Sbartai, S. Laurens, K. Viriyametanont, J. P. Balayssac, and G. Arliguie. Non-destructive evaluation of concrete physical condition using radar and artificial neural networks. *Construction and Building Materials*, 23:837–845, 2009.
- [230] S. Park, J. P. Hwang, E. Kim, H. Lee, and H. G. Jung. A neural network approach to target classification for active safety system using microwave radar. *Expert Systems with Applications*, 37:2340–2346, 2010.

- [231] P. Molchanov, J. Astola, K. Egiazarian, and A. Totsky. Ground moving target classification by using DCT coefficients extracted from micro-doppler radar signatures and artificial neuron network. *Microwaves, Radar and Remote Sensing Symposium*, pages 173–176, 2011.
- [232] T. Islam, M. A. Rico-Ramirez, D. Han, and P. K. Srivastava. Artificial intelligence techniques for clutter identification with polarimetric radar signatures. *Atmospheric Research*, 109–110:95–113, 2012.
- [233] G. Montazer, H. Khoshniat, and V. Fathi. Improvement of RBF neural networks using fuzzy-OSD algorithm in an online radar pulse classification system. *Applied Soft Computing*, 13:3831–3838, 2013.
- [234] N. Petrov, I. Jordanov, and J. Roe. Radar emitter signals recognition and classification with feedforward networks. *Procedia Computer Science*, 22:1192–1200, 2013.
- [235] C. M. Lin, Y. M. Chen, and C. S. Hsueh. A self-organizing interval type-2 fuzzy neural network for radar emitter identification. *International Journal of Fuzzy Systems*, 16(1):20–30, 2014.
- [236] S. Wagner. Combination of convolutional feature extraction and support vector machines for radar ATR. *17th International Conference on Information Fusion*, 2014.
- [237] R. Ressel, A. Frost, and S. Lehner. A neural network-based classification for sea ice types on X-band SAR images. *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing*, 8(7):3672–3680, 2015.
- [238] P. Szymczyk and M. Szymczyk. Classification of geological structure using ground penetrating radar and laplace transform artificial neural networks. *Neurocomputing*, 148:354–362, 2015.
- [239] J. Ding, B. Chen, H. Liu, and M. Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and Remote Sensing Letters*, 13(3):354–362, 2016.
- [240] M. Gong, J. Zhao, J. Liu, and Q. Miao. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(1):125–138, 2016.
- [241] I. Christoyianni, E. Dermatas, and G. Kokkinakis. Fast detection of masses in computer-aided mammography. *IEEE Signal Processing Magazine*, 17(1):54–64, 2000.
- [242] P. S. Rodrigues, G. A. Giraldi, R. F. Chang, and J. S. Suri. Non-extensive entropy for cad systems of breast cancer images. *19th Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–128, 2006.
- [243] <http://www.cancer.org>. *American Cancer Society Homepage*, 2008.
- [244] <http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+>

- [245] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
- [246] L. Hopley and J. V. Schalkwyk. The magnificent ROC. 2001.
- [247] X. H. Zhou and J. Harezlak. Comparison of bandwidth selection methods for kernel smoothing of ROC curves. *Statistics in Medicine*, 21(14):2045–2055, 2002.
- [248] J. R. Quinlan. Improved use of continuous attributes in C4,5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
- [249] H. J. Hamiton, N. Shan, and N. Cercone. Riacc: A rule induction algorithm based on approximate classification technical report. *University of Regina*, 1996.
- [250] B. Ster and A. Dobnikar. Neural networks in medical diagnosis: Comparison with other methods. *Proc. EANN'96, International Conference on Engineering Applications of Neural Networks*, pages 427–430, 1996.
- [251] D. Nauck and R. Kruse. Obtaining interpretable fuzzy classification rules from medical data. *Artificial Intelligence in Medicine*, 16:149–169, 1999.
- [252] C. A. Peña-Reyes and M. D. Sipper. A fuzzy-genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine*, 17:131–155, 1999.
- [253] R. Setiono. Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 18(3):205–217, 2000.
- [254] A. A. Albrecht, G. Lappas, S. A. Vinterbo, C. K. Wong, and L. Ohno-Machado. Two applications of the LSA machine. *Proc. ICONIP '02, 9th International Conference on Neural Information Processing*, pages 184–189, 2002.
- [255] J. Abonyi and F. Szeifert. Supervised fuzzy clustering for the identification of fuzzy classifiers. *Pattern Recognition Letters*, 14(24):2195–2207, 2003.
- [256] E. D. Ubeyli. Implementing automated diagnostic systems for breast cancer detection. *Expert Systems with Applications*, 33(4):1054–1062, 2007.
- [257] K. Polat K and S. Günes. Breast cancer diagnosis using least square support vector machine. *Digital Signal Processing*, 17(4):694–701, 2007.
- [258] B. Guijarro-Berdias, O. Fontenla-Romero, B. Perez-Sanchez, and P. Fraguera. A linear learning method for multilayer perceptrons using least squares. *Proc. IDEAL 2007, Lecture Notes in Computer Science*, 4881:365–374, 2007.
- [259] M. F. Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2):3240–3247, 2009.
- [260] M. Karabatak and M. Cevdet-Ince. An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications*, 36:3465–3469, 2009.
- [261] L. Peng, B. Yang, and J. Jiang. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43(1):15–23, 2010.

- [262] D. Conforti and R. L. Guido. Kernel based support vector machine via semidefinite programming: Application to medical diagnosis. *Computers and Operations Research*, 37:1389–1394, 2010.
- [263] F. Paulin and A. Santhakumaran. Classification of breast cancer by comparing back propagation training algorithms. *International Journal on Computer Science and Engineering*, 3:327–332, 2011.
- [264] H. Chen, B. Yang, J. Liu, and D. Liu. A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 38(7):9014–9022, 2011.
- [265] A. Fallahi and S. Jafari. An expert system for detection of breast cancer using data preprocessing and bayesian network. *International Journal of Advanced Science and Technology*, 34:65–70, 2011.
- [266] A. Raad, A. Kalakech, and M. Ayache. Breast cancer classification using neural network approach: Mlp and rbf. *International Arab Conference on On Information Technology*, pages 15–19, 2012.
- [267] O. Inan, M. S. Uzer, and N. Yilmaz. A new hybrid feature selection method based on association rules and pca for detection. *International Journal of Innovative Computing, Information and Control*, 9(2):727–739, 2013.
- [268] S. S. Shajahaan, S. Shanthi, and V. Manochitra. Application of data mining techniques to model breast cancer data. *International Journal of Emerging Technology and Advanced Engineering*, 3(11):362–369, 2013.
- [269] C. P. Utomo, A. Kardiana, and R. Yuliwulandari. Breast cancer diagnosis using artificial neural networks with extreme learning techniques. *International Journal of Advanced Research in Artificial Intelligence*, 3(7):10–14, 2014.
- [270] A. Bhardwaj and A. Tiwari. Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, 42(10):4611–4620, 2015.
- [271] A. Onan. A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer. *Expert Systems with Applications*, 42(20):6844–6852, 2015.
- [272] A. M. Abdel-Zaher and A. Eldeib. Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46:139–144, 2016.
- [273] A. Groznik. Towards multi utility management in Europe. *ICEE 2014: International Conference on Energy and Environment*, 2(8):195, 2014.
- [274] J. Cardenas, L. Gemoets, J. Rosas, and R. Sarfi. A literature survey on smart grid distribution: an analytical approach. *Journal of Cleaner Production*, 65:202–216, 2014.
- [275] F. Starace. The utility industry in 2020. *Handbook Utility Management*, pages 202–216, 2009.

- [276] J. Noble. The core of it. *CIO Insight*, pages 15–17, 2006.
- [277] M. Kezunovic, V. Vittal, S. Meliopoulos, and T. Mount. The big picture: Smart research for large-scale integrated smart grid solutions. *IEEE Power and Energy Magazine*, 10(4):22–34, 2014.
- [278] L. C. Thomas. A survey of credit and behavioral scoring: forecasting financial. *International Journal of Forecasting*, 16(2):149–172, 2000.
- [279] [https://archive.ics.uci.edu/ml/datasets/statlog+\(australian+credit+approval\)](https://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)).
- [280] D. West. Neural network credit scoring models. *Computers and Operations Research*, 27:1131–1152, 2000.
- [281] C. S. Ong, J. J. Huang, and G. H. Tzeng. Building credit scoring models using genetic programming. *Expert Systems with Applications*, 29:41–47, 2005.
- [282] J. J. Huang, G. H. Tzeng, and C. S. Ong. Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation*, 174(2):1039–1053, 2005.
- [283] D. Martens, B. Baesens, T. Van Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1466–1476, 2007.
- [284] F. Hoffmann, B. Baesens, C. Mues, T. Van Gestel, and J. Vanthienen. Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms. *European Journal of Operational Research*, 177:540–556, 2007.
- [285] C. L. Huang, M. C. Chen, and C. J. Wang. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33(4):847–856, 2007.
- [286] Y. Peng, G. Kou, Y. Shi, and Z. Chen. A multi-criteria convex quadratic programming model for credit data analysis. *Decision Support Systems*, 44(4):1016–1030, 2008.
- [287] C. F. Tsai and J. W. Z. Wu. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, 44(4):2639–2649, 2008.
- [288] L. Nanni and A. Lumini. An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, 36:3028–3033, 2009.
- [289] X. Xu, C. Zhou, and Z. Wang. Credit scoring algorithm based on link analysis ranking with support vector machine. *Expert Systems with Applications*, 36(2):2625–2632, 2009.
- [290] S. T. Luo, B. W. Cheng, and C. H. Hsieh. Prediction model building with clustering-launched classification and support vector machines in credit scoring. *Expert Systems with Applications*, 36(4):7562–7566, 2009.
- [291] C. F. Tsai. Feature selection in bankruptcy prediction. *Knowledge-Based Systems*, 22(22):120–127, 2009.

- [292] Y. Ping. Hybrid classifier using neighborhood rough set and SVM for credit scoring. *Proc. BIFE '09, International Conference on Business Intelligence and Financial Engineering*, pages 138–142, 2009.
- [293] F. L. Chen and F. C. Li. Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37(7):4902–4909, 2009.
- [294] G. Wang, J. Hao, J. Ma, and H. Kiang. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223–230, 2011.
- [295] G. Wang, J. Ma, L. Huang, and K. Xu. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, 26:61–68, 2012.
- [296] S. Vukovic, B. Delibasic, A. Uzelac, and M. Suknovic. A case-based reasoning model that uses preference theory functions for credit scoring. *Expert Systems with Applications*, 39(9):8389–8395, 2012.
- [297] C. F. Tsai. Combining cluster analysis with classifier ensembles to predict financial distress. *Information Fusion*, 16:46–58, 2014.
- [298] M. Semenkina and E. Semenkin. Hybrid self-configuring evolutionary algorithm for automated design of fuzzy classifier. *International Conference in Swarm Intelligence ICSI 2014*, pages 310–317, 2014.
- [299] Z. Zhao, S. Xu, B. H. Kang, M. M. J. Kabir, Y. Liu, and R. Wasinger. Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, 42:3508–3516, 2015.
- [300] D. Liang, D. F. Tsai, and H. T. Wu. The effect of feature selection on financial distress prediction. *Knowledge-Based Systems*, 73:289–297, 2015.
- [301] H. Xiao, Z. Xiao, and Y. Wang. Ensemble classification based on supervised clustering for credit scoring. *Applied Soft Computing*, 43:73–86, 2016.
- [302] M. H. Chen, P. C. Chang, and J. L. Wu. A population-based incremental learning approach with artificial immune system for network intrusion detection. *Engineering Applications of Artificial Intelligence*, 51:171–181, 2016.
- [303] U.S.EPA (U.S. Environmental Protection Agency). www.epa.gov/air/airpollutants.html. 2012.
- [304] SESA (Spanish acronym for Spanish Society of Environmental Health). <http://www.sanidadambiental.com/2008/08/19/environment-and-health/>. 2008.
- [305] S. M. S. Nagendra and M. Khare. Artificial neural network based line source models for vehicular exhaust emission predictions of an urban roadway. *Transportation Research Part D—Transport and Environment*, 9:199–208, 2004.
- [306] M. Celik and I. Kadi. The relation between meteorological factors and pollutants concentration in Karabuk City. *Journal of Science*, 20:89–95, 2007.

- [307] G. D'Amato, L. Cecchi, M. D'Amato, and G. Liccardi. Urban air pollution and climate change as environmental risk factors of respiratory allergy: An update. *Journal of Investigational Allergology and Clinical Immunology*, 20:95–102, 2010.
- [308] P. Perez, A. Trier, and J. Reyes. Prediction of $pm_{2.5}$ concentrations several hours in advance using neural networks in Santiago, Chile. *Atmospheric Environment*, 34:1189–1196, 2000.
- [309] S. H. Lee and Y. H. Sung and H. W. Lee. Impact of regional trans-boundary ozone associated with complex terrain on urban air quality. *Atmospheric Environment*, 42:7384–7396, 2008.
- [310] H. K. Elminir. Dependence of urban air pollutants on meteorology. *Science of the Total Environment*, 350:225–237, 2005.
- [311] J. L. Pearce, J. Beringer, N. Nicholls, R. J. Hyndman, and N. J. Tapper. Quantifying the influence of local meteorology on air quality using generalized additive models. *Atmospheric Environment*, 45:1328–1336, 2011.
- [312] M. G. Cortina-Januchs, J. Quintanilla-Dominguez, A. Vega-Corona, and D. Andina. Development of a model for forecasting of PM_{10} concentrations in Salamanca, Mexico. *Atmospheric Pollution Research*, 6:626–634, 2015.
- [313] B. Ando, S. Baglio, S. Graziani, and N. Pitrone. Models for air quality management and assessment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 30(3):358–363, 2000.
- [314] U. Schlink, S. John, and O. Herbarth. Transfer-function models predicting ozone in urban air. *Contribution to the SATURN Project, Annual Report*, 2001.
- [315] C. Silva, S. Alvarado, and R. Montaña. Modelamiento de la contaminación atmosférica por partículas: comparación de cuatro procedimientos predictivos en Santiago, Chile. *Biomatemática*, 13:113–128, 2003.
- [316] J. Kukonen, L. Partanen, A. Karppinen, J. Ruuskanen, H. Junninen, M. Kolehmainen, H. Niska, S. Dorling, T. Chatterton, R. Foxall, and G. Cawley. Extensive evaluation of neural network models for the prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements in central helsinki. *Atmospheric Environment*, 37:4539–4550, 2003.
- [317] D. Jiang, Y. Zhang, X. Hu, Y. Zeng, J. Tan, and D. Shao. Progress in developing an ANN model for air pollution index forecast. *8th International Conference on Atmospheric Sciences and Applications to Air Quality (ASAAQ)*, 38(40):7055–7064, 2004.
- [318] G. Corani. Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. *Ecological Modelling*, 185(2–4):513–529, 2005.
- [319] J. B. Ordieres, E. P. Vergara, R. S. Capuz, and R. E. Salazar. Neural network prediction model for fine particulate matter $PM_{2.5}$ on the US-Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua). *Environmental Modelling and Software*, 20(5):547–559, 2005.

- [320] E. Agirre-Basurkoa, G. Ibarra-Berastegib, and I. Madariaga. Regression and multilayer perceptron-based models to forecast hourly O_3 and NO_2 levels in the Bilbao area. *Environmental Modelling and Software*, 21(4):430–446, 2006.
- [321] G. Salini and P. Pérez. Estudio de series temporales de contaminación ambiental mediante técnicas de redes neuronales artificiales. *Revista chilena de ingeniería*, 14(3):284–290, 2006.
- [322] G. Grivas and A. Chaloulakou. Artificial neural network models for prediction of PM_{10} hourly concentrations, in the Greater Area of Athens, Greece. *Atmospheric Environment*, 40(7):1216–1229, 2006.
- [323] U. Schlink, O. Herbarth, M. Richter, S. Dorling, G. Nunnari, G. Cawley, and E. Pelikan. Statistical models to assess the health effects and to forecast ground-level ozone. *Environmental Modelling and Software*, 21(4):547–558, 2006.
- [324] U. Brunelli, V. Piazza, L. Pignato, F. Sorbello, and S. Vitabile. Two-days ahead prediction of daily maximum concentrations of SO_2 , O_3 , PM_{10} , NO_2 , CO in the urban area of Palermo, Italy. *Atmospheric Environment*, 41(14):2967–2995, 2007.
- [325] A. L. Dutot, J. Rynkiewicz, F. E. Steiner, and J. Rude. A 24-h forecast of ozone peaks and exceedance levels using neural classifiers and weather predictions. *Environmental Modelling and Software*, 22(9):1261–1269, 2007.
- [326] K. D. Karatzas and S. Kaltsatos. Air pollution modelling with the aid of computational intelligence methods in Thessaloniki, Greece. *Simulation Modelling Practice and Theory*, 15(10):1310–1319, 2007.
- [327] L. A. Díaz-Robles, J. C. Ortega, J. S. Fu, G. D. Reed, J. C. Chow, J. G. Watson, and J. A. Moncada-Herrera. A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of temuco, chile. *Atmospheric Environment*, 42(35):833–840, 2008.
- [328] A. Coman, A. Ionescu, and Y. Candau. Hourly ozone prediction for a 24-h horizon using neural networks. *Environmental Modelling and Software*, 23(12):1407–1421, 2008.
- [329] A. Kurt, B. Gulbagci, F. Karaca, and O. Alagha. An online air pollution forecasting system using neural networks. *Environment International*, 12(5):592–598, 2008.
- [330] E. Salazar-Ruíz, J. B. Ordieres, E. P. Vergara, and S. F. Capuz-Rizo. Development and comparative analysis of tropospheric ozone prediction models using linear and artificial intelligence-based models in Mexicali, Baja California (Mexico) and Calexico, California (US). *Environmental Modelling and Software*, 23(8):1056–1069, 2008.
- [331] P. Sharma, A. Chandra, and S. Kaushik. Forecasts using box jenkins models for the ambient air quality data of Delhi City. *Environmental Modelling and Software*, 157(1):105–112, 2009.
- [332] S. V. Barai, A. K. Gupta, and J. Kodali. Air quality forecaster: Moving window based neuro models. *Advances in Intelligent and Soft Computing: Applications of Soft Computing*, pages 137–146, 2009.

- [333] L. Hrust, K. Z. Bencetic, J. Krizan, O. Antonio, and P. Hercog. Neural network forecasting of air pollutants hourly concentrations using optimised temporal averages of meteorological variables and pollutant concentrations. *Atmospheric Environment*, 43(35):5588–5596, 2009.
- [334] A. Kurt and A. B. Oktay. Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. *Expert Systems with Applications*, 37(12):7986–7992, 2010.
- [335] L. Mesin, F. Orione, R. Taormina, and E. Pasero. A feature selection method for air quality forecasting. *Artificial Neural Networks ICANN 2010*, 6354:489–494, 2010.
- [336] A. Sfetsos and D. Vlachogiannis. A new methodology development for the regulatory forecasting of PM_{10} . application in the Greater Athens Area, Greece. *Atmospheric Environment*, 44(26):3159–3172, 2010.
- [337] P. Mitra and P. C. Barman. Coumarin extraction from cuscuta reflexa using supercritical fluid carbon dioxide and development of an artificial neural network model to predict the coumarin yield. *Food Bioprocess Technol*, 4:737–744, 2011.
- [338] N. G. Turan, B. Mesci, and O. Ozgonenel. The use of artificial neural networks (ANN) for modeling of adsorption of Cu(II) from industrial leachate by pumice. *Chemical Engineering Journal*, 171:1091–1097, 2011.
- [339] P. Shivakumar, P. P. Srinivasa, and B. R. R. Srinivasa. Artificial neural network based prediction of performance and emission characteristics of a variable compression ratio CI engine using WCO as a biodiesel at different injection timings. *Applied Energy*, 88:2344–2354, 2011.
- [340] Z. Frontistis, V. M. Daskalaki, E. Hapeshi, C. Drosou, D. Fatta-Kassinou, N. P. Xekoukoulotakis, and D. Mantzavinos. Photocatalytic (UV-A/ TiO_2) degradation of 17 α -ethynylestradiol in environmental matrices: Experimental studies and artificial neural network modeling. *Journal of Photochemistry and Photobiology A: Chemistry*, 240:33–41, 2012.
- [341] M. Khajeh, M. Kaykhaii, and A. Sharafi. Application of PSO-artificial neural network and response surface methodology for removal of methylene blue using silver nanoparticles from water samples. *Journal of Industrial and Engineering Chemistry*, 19:1624–1630, 2013.
- [342] L. Zhang, F. Tian, S. Liu, J. Guo, B. Hu, Q. Ye, L. Dang, X. Peng, C. Kadri, and J. Feng. Chaos based neural network optimization for concentration estimation of indoor air contaminants by an electronic nose. *Sensors and Actuators A: Physical*, 189:161–167, 2013.
- [343] N. Hattab and M. Motelica-Heino. Application of an inverse neural network model for the identification of optimal amendment to reduce copper toxicity in phytoremediated contaminated soils. *Journal of Geochemical Exploration*, 136:14–23, 2014.
- [344] A. Varley, A. Tyler, L. Smith, and P. Dale. Development of a neural network approach to characterise 226Ra contamination at legacy sites using gamma-ray spectra taken from boreholes. *Journal of Environmental Radioactivity*, 140:130–140, 2015.

-
- [345] M. D. Adams and P. S. Kanaroglou. Mapping real-time air pollution health risk for environmental management: Combining mobile and stationary air pollution monitoring with neural network models. *Journal of Environmental Management*, 168:133–141, 2016.
- [346] A. Giwa, S. Daer, I. Ahmed, P. R. Marpu, and S. W. Hasan. Experimental investigation and artificial neural networks ANNs modeling of electrically-enhanced membrane bioreactor for waste water treatment. *Journal of Water Process Engineering*, 11:88–97, 2016.