Koniocortex-Like Network Unsupervised Learning Surpasses Supervised Results on WBCD Breast Cancer Database

J. Fombellida[1](✉), F.J. Ropero-Peláez[2], and D. Andina[1]

[1] Group for Automation in Signals and Communications,
Universidad Politécnica de Madrid, 28040 Madrid, Spain
jfv@alumnos.upm.es, d.andina@upm.es

[2] Center of Mathematics, Computation and Cognition,
Universidade Federal do ABC, Santo André, Brazil
francisco.pelaez@ufabc.edu.br

Abstract. Koniocortex-Like Network 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 Koniocortex-Like Network competition and pattern classification emerges naturally due to the interplay of inhibitory interneurons, metaplasticity and intrinsic plasticity. 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 a relevant standard real application that is the objective of state-of-the-art research: the diagnosis of breast cancer data from the Wisconsin Breast Cancer Database.

Keywords: Metaplasticity · Koniocortex · Plasticity · KLN · WBCD · Feature extraction · Competition

1 Introduction

The koniocortex is a common denomination for all regions of the cerebral cortex containing a granular layer (layer IV). The granular (grainy) texture of this layer is due the abundance of spiny stellate 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 Koniocortex-like networks (KLN) are neural models that possess at least two layers: the first layer containing neurons that are similar to the thalamo-cortical neurons of the thalamus, and the second layer whose neurons resemble the spiny and the inhibitory interneurons of the fourth layer of the koniocortex. As demonstrated with living brain tissues in which only the fourth layer of the cortex was active (remaining layers were reset through freezing [8]), these two layers constitute a network in which competition and auto-organization are found. For example, the biological

J.M. Ferrández Vicente et al. (Eds.): IWINAC 2017, Part II, LNCS 10338, pp. 32–41, 2017.
DOI: 10.1007/978-3-319-59773-7_4

koniocortex network exhibits competition [13] because only a very small number of spiny stellate neurons are active in the presence of sensory stimuli. This behavior resembles the Winner-Take-All (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 neuro through calculation, in the case of the KLNs, the winning neuron emerges naturally from the interaction between the neurons. At the same time, non-winning neurons become silent due to the dynamics of the neurons in the network, not because they are algorithmically reset. Previous seminal 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 [9–11], the value of the firing threshold [10,18] and the steepness of the activation function [18].

Regarding the KLN, the main properties involved in competitive learning are synaptic metaplasticity and intrinsic plasticity. Intrinsic plasticity adjusts the global excitability of the neuron so that highly excited neurons will be less excitable in the future, and vice versa. In this paper, we continue the research started in previous works [3,4,14–17]. Here we use KLN networks to classify the patterns in the Wisconsin Breast Cancer Database (WBCD) [19]. For assessing the classification accuracy of this algorithm, we used the most common performance measures: specifity, sensitivity and accuracy. The results obtained were validated using the 10-fold cross-validation method. The paper is organized as follows. Section 2 presents a detailed description of the database and the algorithms. In Sect. 3 the experimental results obtained are shown. A brief discussion of these results is showed in Sect. 4 and, finally, Sect. 5 summarizes the main conclusions.

2 Materials and Methods

2.1 WBCD Dataset

Breast cancer is a malignant tumor that develops from breast cells. Although research has identified some of the risk factors that increase a woman's chance of developing breast cancer, the inherent cause of most breast cancers remains unknown.

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%. 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 (WBCD). This data base has been used several times in the literature and many high impact studies has used these inputs for classification including systems based in Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and Neuro-Fuzzy techniques, among others. This situation makes this database very useful in order to compare the performances of the results obtained with the state of the art.

2.2 Data Preparation

The WBCD contains 699 patterns, each of this pattern is composed by 9 numerical attributes that corresponds to different physical characteristics that can be considered as markers of the possible presence of cancer in the sample. 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 malicious nature of the sample. The database contains a field that indicates the final diagnosis of the nature of the sample.

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

It has empirically been proved that the classifiers based on neural networks produce better results if the training sets are equilibrated 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 malicious patterns will be repeated instead of eliminating some benign patterns to get these equilibrated sets. It has been considered better to duplicate a small number of malicious elements as inputs for the networks instead of losing the potential information present in some of the benign elements.

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 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 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. In each one of the training sets that will be used as inputs to the networks for training the system and evaluating the evolution of the error will consist in 9 of the previous 10 groups. 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 sets created from the initial ones. 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.

2.3 Koniocortex-Like Network 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 $\overrightarrow{i} = \overrightarrow{I} /|| \overrightarrow{I} ||$ (lower case notation meaning vector normalization) yields the net-input of neuron j. Normalization is performed with the l_1-norm in which:

$$|| \overrightarrow{I} || = \sum_{i=1}^{n} |I_i| \tag{1}$$

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

During the training the weights are modified using the incremental version of the presynaptic rule:

$$\triangle \omega = \xi I(O - \omega) \tag{2}$$

where O and I are the postsynaptic and presynaptic action potential probabilities, respectively, and ξ, a learning factor.

The presynaptic rule is based in the empirical plasticity curve [5] that shows a relation between postsynaptic voltage and the modification of the synaptic weight. This rule is also influenced by metaplasticity [1,2], a homeostatic property which elongates the plasticity curves rightwards for higher initial synaptic weights.

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

$$O_j = \frac{1}{1 + e^{-k(net_j + 0.5 - 2s^j)}} \tag{3}$$

where k is a curve-compressing factor and s^j the horizontal shift of the activation function ranging from zero to one, $0 < s^j < 1$. In our experiment 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 [6,7] as shown in Fig. 1, the homeostatic property that makes very active neurons to be moderated and inactive neurons to increment its firing rate. According to this property [7], the activation function gradually shifts leftwards or rightwards regulating the activation of scarcely or highly activated neurons, respectively.

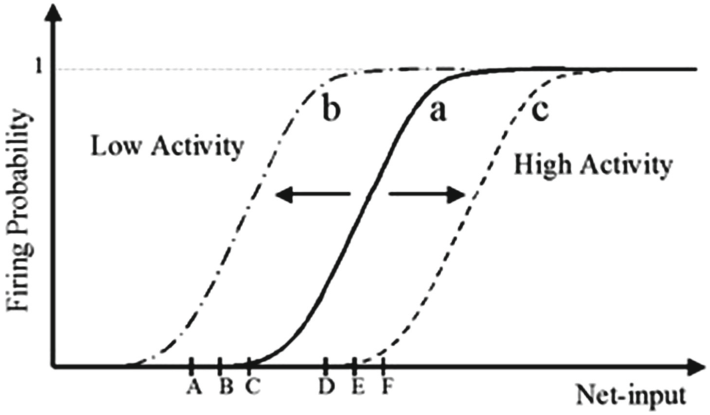

Fig. 1. Intrinsic plasticity allows the neurons' activation function to shift horizontally so that the activation function "follows" the average net-input of the neuron. (a) Initial position of the sigmoidal activation function. (b) In the case of a low regime of net-input values (as in A, B and C), 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.

In the experiment, 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:

$$O^j = f(|| \overrightarrow{T^j_{\vec{I}}} ||, s^j) \tag{4}$$

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 $t-1$.

$$s_t^j = \frac{\upsilon.O_{t-1} + s_{t-1}^j}{\upsilon + 1} \tag{5}$$

where υ 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 of the shift at time $t-1$.

Figure 2 is the complete version of the KLN model used in the experiment. In the KLN, "B" labeled neurons are inhibitory neurons endowed with intrinsic plasticity. "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, patterns become more uncorrelated and easier to classify.

Figure 2 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.

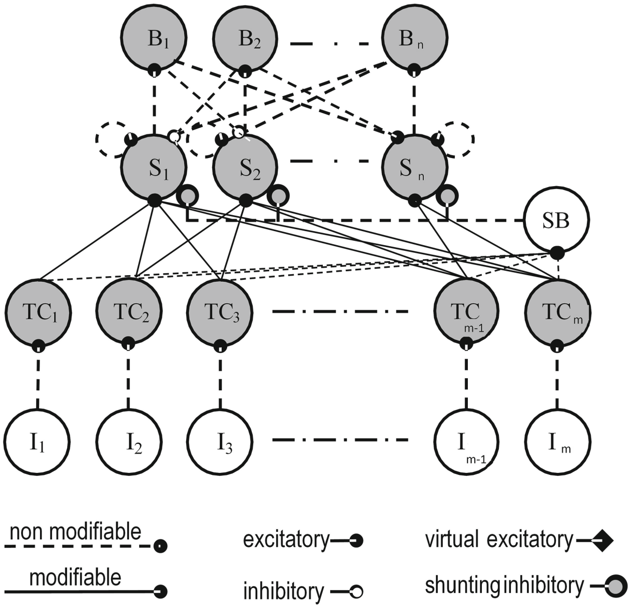

Fig. 2. Architecture of KLN applied to the classification of WBCD.

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 Sneurons) by this quantity.

3 Results

3.1 Network Characteristics

In this section we present the results obtained in this research. All the models used in this study were trained and tested with the same data and validated using 10-fold cross-validation.

This KLN has 9 neurons in its input layer corresponding to the number of elements that form each input pattern, 9 neurons in its TC layer (similar number as input layer to be coherent with the KLN structure), 2 neurons in the S layer as two classes are considered in the experiment, and 2 neurons in the upper B layer. 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 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 obtained for the parameters have been obtained 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 υ set to 0.025 and ξ to 0.001, the initial sigmoid shift was 0.5, initial weights from TC to S neurons were negligible and random, and non-modifiable weights were set to $W_{S_S} = 0.85$, $W_{S_B} = 0.98$, $W_{I_{TC}} = 1.0$ and $W_{B_S} = 0.5$.

3.2 Evaluation Method

In each one of the experiments 50 networks have been 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 in the creation of the networks. From the results obtained for the same network with each one of the folders the mean confusion matrix is obtained for each network. Once we have these 50 mean values an additional calculation is made and the final mean value is obtained as the final result of the experiment.

The following hypothesis are defined and used to define the confusion matrix:

- H (1/1): The pattern is malicious and has been classified as malicious.
- H (1/0): The pattern is benign and has been classified as malicious.
- H (0/1): The pattern is malicious and has been classified as benign.
- H (0/0): The pattern is benign and has been classified as benign (Table 1).

Table 1. Confussion matrix model

True positive H (1/1)	False positive H (1/0)
False negative H (0/1)	True negative H (0/0)

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

3.3 Classification Results

We have performed different experiments 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 (benign or malign sample) is selected by the network. Table 2 presents the results obtained:

Table 2. Sensitivity and accuracy evolution depending on the number of epochs

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%

The best results obtained correspond to 1000 epochs where almost all the malign patterns are correctly recognized. Confusion matrix for this experiment is shown in Table 3.

Table 3. Confusion matrix 1000 epochs

99.94%	6.12%
0.06%	93.88%

4 Discussion

- 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.
- The best sensitivity is obtained with the higher number of epochs in the training. Almost all the patters corresponding 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 [12], so KLN performs a Deeper Learning of the information in the Data Set, without the need of a Deep Network.

5 Conclusions

In this paper we have applied the theoretical basis of the Koniocortex-Like Network to a real complex Data Set as is the classification of real breast cancer input patterns. The simulations show that the unsupervised learning that emerges from individual neurons properties surpasses results even of several advanced state-of-the-art supervised learning algorithms. Nevertheless KLN is still a very novel model and the results presented are just from a non optimized prototype, this bio-inspired model seems to be able to compete in deeper learning and better performance than many state-of-the-art Artificial Neural Network models.

References

1. Abraham, W.C., Bear, M.F.: Metaplasticity: the plasticity of synaptic plasticity. Trends Neurosci. **19**, 126–130 (1996)
2. Abraham, W.C., Tate, W.P.: Metaplasticity: a new vista across the field of synaptic plasticity. Prog. Neurobiol. **52**, 303–323 (1997)
3. Andina, D., Alvarez-Vellisco, A., Jevtic, A., Fombellida, J.: Artificial metaplasticity can improve artificial neural network learning. Intell. Autom. Soft Comput. Spec. Issue Sig. Process. Soft Comput. **15**(4), 681–694 (2009)

4. Andina, D., Ropero-Pelaez, J.: On the biological plausibility of artificial metaplasticity learning algorithm. Neurocomputing (2012). http://dx.doi.org/10.1016/j.neucom.2012.09.028
5. Artola, A., Brocher, S., Singer, W.: Different voltage-dependent threshold for inducing long-term depression and long-term potentiation in slices of rat visual córtex. Nature **347**, 69–72 (1990)
6. Desai, N.S.: Homeostatic plasticity in the CNS: synaptic and intrinsic forms. J. Physiol. **97**(4–6), 391–402 (2003)
7. Desai, N.S., Rutherford, L.C., Turrigiano, G.G.: Plasticity in the intrinsic excitability of cortical pyramidal neurons. Nat. Neurosci. **2**, 515–520 (1999)
8. Ferster, D., Chung, S., Wheat, H.: Orientation selectivity of thalamic input to simple cells of cat visual cortex. Nature **380**(6571), 249–252 (1996)
9. Fukai, T., Tanaka, S.: 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)
10. Kaski, S., Kohonen, T.: Winner-take-all networks for physiological models of competitive learning. Neural Netw. **7**(6/7), 973–984 (1994)
11. Mao, Z.H., Massaquoi, S.G.: Dynamics of Winner-Take-All competition in recurrent neural networks with lateral inhibition. IEEE Trans. Neural Netw. **18**, 55–69 (2007)
12. Marcano-Cedeño, A., Quintanilla-Dominguez, J., Andina, D.: Breast cancer classification applying artificial metaplasticity algorithm. Neurocomputing **74**(8), 1243–1250 (2011)
13. Miller, K.D.: Synaptic economics: competition and cooperation in synaptic plasticity. Neuron **17**, 371–374 (1996)
14. Quintanilla-Dominguez, J., Cortina-Januchs, M.G., Ojeda-Magaa, B., Jevtic, A., Vega-Corona, A., Andina, D.: Microcalcification detection applying artificial neural networks and mathematical morphology in digital mammograms. In: World Automation Congress (WAC) (2010)
15. Ropero-Peláez, F.J., Andina, D.: Do biological synapses perform probabilistic computations? Neurocomputing (2012). http://dx.doi.org/10.1016/j.neucom.2012.08.042
16. Ropero-Peláez, F.J., Andina, D.: The Koniocortex-like network: a new biologically plausible unsupervised neural network. In: Ferrández Vicente, J.M., Álvarez-Sánchez, J.R., de la Paz López, F., Toledo-Moreo, F.J., Adeli, H. (eds.) IWINAC 2015. LNCS, vol. 9107, pp. 163–174. Springer, Cham (2015). doi:10.1007/978-3-319-18914-7_17
17. Ropero-Peláez, F.J., Aguiar-Furucho, M.A., Andina, D.: Intrinsic plasticity for natural competition in Koniocortex-like neural networks. Int. J. Neural Syst. **26**(5), 1650040 (2016). http://www.worldscientific.com/doi/abs/10.1142/S0129065716500404
18. Yang, J.F., Chen, C.M.: Winner-Take-All neural network using the highest threshold. IEEE Trans. Neural Netw. **11**, 194–199 (2000)
19. http://archive.ics.uci.edu/ml/datasets.html