Graduado en Matemáticas e Informática

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Extracción de atributos faciales mediante redes convolucionales

Alumno - Miguel Maricalva
Tutor - Luis Baumela

Madrid, Julio 2017

Abstract

This project focuses on the extraction of facial attributes –namely, gender, age and some other minor features– using convolutional neural networks. We build on previous work [11] that has developed an efficient, accurate architecture for the task of face recognition and aim to use this architecture as a feature extractor on which to develop our classification networks. On this basis, our gender classification network achieves state of the art results on the CelebA test set [37].

The next facet of our project aims to check if a multi-label model could outperform independently trained networks with the same data by taking advantage of correlations between these attributes on the training data; for example taking advantage of the gender-specific age characteristics and age-specific gender characteristics inherent to facial images. The results, although not managing to overcome the individually trained models, fall really close behind, proving at least that this architecture is able to learn to classify multiple attributes without a mayor loss of accuracy.

Resumen

Este trabajo se centra en la extracción de atributos faciales –género y edad entre otros– mediante redes neuronales convolucionales. Nos basamos en un trabajo previo [11] que desarrolla una red de gran eficiencia y precisión en la tares de reconocimiento facial y lo usamos como extractor de características para construir sobre este nuestros sistemas clasificadores. De esta forma obtenemos una red de clasificación de género que alcanza el estado del arte en el conjunto de evaluación de CelebA [37].

La siguiente parte de nuestro proyecto trata de comprobar si una red multi-etiqueta puede obtener mejores resultados que redes entrenadas independientemente con los mismos datos aprovechando la correlación entre estos atributos en los datos de entrenamiento; por ejemplo la clasificación de género debe tener en cuenta características que son dependientes de la edad, y viceversa. Aunque los resultados no han probado esta hipótesis, si se han quedado lo suficientemente cerca como para afirmar que esta red tiene capacidad como para clasificar múltiples atributos simultáneamente sin sufrir grandes pérdidas de precisión.

Índice

1 Introducción

En la última década el número de imagenes subidas a la red a crecido exponencialmente. Toda esta cantidad de nuevos datos ha permitido a los investigadores en Vision Artificial desarrollar nuevas técnicas que antes eran irrelevantes o impracticables. Así, han aparecido sistemas de reconocimiento de imagenes de gran precisión y eficiencia basados en redes convolucionales que han revolucionado el campo en los últimos años, si bien es cierto que ya se usaron este tipo de redes a principio de los 80 [12].

La aplicación de estos métodos va desde las sugerencias para etiquetar a amigos en fotos de Facebook a detección de viandantes en sistemas de piloto automático que desarrollan varias compañias automovilísticas actualmente. Ahora sí, el siguiente gran paso a dar es no sólo ser capaz de decir cuántas caras hay en una imagen y dónde están, sino también qué rasgos tienen estas caras. El objetivo de este proyecto es precisamente intentar clasificar la edad, género y otros atributos faciales a partir de imágenes de rostros.

Las aplicaciones de esta tecnología tienen un gran rango de alcance. Redes sociales como Facebook podrían usar este tipo de información para inferir el contexto de una imagen. Por ejemplo, si en una foto se ve a varias personas de unos 20 años alrededor de una mesa con papeles podría subtitular la foto como "sesión de estudio", mientras que si los sujetos en la foto son de edad más avanzada, la podría titular como "reunión de trabajo". Además, teniendo una idea de la edad, género de una persona la tarea de reconocer a esa persona se hace inmensamente más fácil. Esto podría usarse en dispositivos de ayuda asistida para quien padece pérdida parcial o total de visión.

La clasificación por rasgos como género o edad es un problema intrínsecamente complicado, más aún que otros retos en visión artificial. La principal razón para explicar esta diferencia está en el tipo de datos que se necesitan para entrenar a estos sistemas. Mientras que la clasificación de objetos comunes suele tener disponible millones de imagenes para entrenar, las bases de datos etiquetadas con estos rasgos faciales son de un tamaño mucho menor, normalmente del orden de los miles o decenas de miles como mucho, datos como la fecha de nacimiento de los individuos en las imagenes muy raramente están disponibles.

Enfrentándose a este problema en [7] por ejemplo decidieron adecuar su arquitectura a estas limitaciones con decisiones como la de hacer redes poco profundas para la clasificación de estos rasgos. Nosotros en cambio

vamos a tomar un enfoque distinto, vamos a utilizar como soporte el sistema de reconocimiento facial desarrollado en [11], que aunque no está diseñado para clasificar por rasgos faciales, está entrenado con millones de imágenes, por lo que nos servirá para construir sobre él ese extractor de rasgos faciales que buscamos.

2 Trabajos Previos

La clasificación por rasgos faciales ha sido investigada durante décadas. Se han probado muchas técnicas a lo largo de los años, llegando a diferentes grados de éxito. En [2] podemos encontrar un estudio de las diferentes técnicas state-of-the-art en la clasificación por edad que se han desarrollado a lo largo de los años. Los primeros intentos [3] se centraban en la identificación de características faciales introducidas manualmete y las diferencias en tamaño y proporción de estas como rasgos de diferencia de edad, características como el tamaño de los ojos orejas, boca y la distancia entre estos. Algunos consiguieron buenos resultados en condiciones muy controladas (casi idéntica iluminación, orientación y visibilidad entre las imágenes), pero pocos [4] intentaron enfrentarse a las dificultades de las variables del mundo real como la diferente calidad o claridad en las imágenes.

Con respecto al género, podemos encontrar en [5] un estudio holístico de los métodos aplicados a este tipo de clasificación. Ya en 1991 consideraban redes neuronales para clasificar caras por género en [6], pero tradicionalmente la forma de enfrentarse a este problema se ha basado en características escogidas a mano, ya sean globales [13],[14],[15],[16] o locales [18],[19],[20],[22],[21]. Perez et al. [24] usaron una combinación de ambos tipos de características que compartían información. O'Toole et al. [23] demostraron que la información relativa a la profundidad también servía de ayuda. La mayoría de estos métodos fueron puestos a prueba con el benchmark FERET [25] que está muy controlado y para el cual se han llegado a alcanzar resultados casi perfectos [21][24]. Shan [27] puso en práctica patrones binarios locales (LBP) con buenos resultados en imágenes frontales de caras pertenecientes al conjunto "Labeleld Faces on the Wild" (LFW).

En cuanto a la clasificación, las máquinas de soporte vectorial (SVMs) han sido predominantes, ya sea solas [14] o con algoritmos de boosting como Adaboost en [27], llegando a altos niveles de precisión en bases de datos sencillas. Cuando las imágenes se complican más dado a obstrucciones o cambios en el punto de vista, Towels y Arbel [22] demostraron la superioridad de los clasificadores Bayesianos respecto a las SVM usando un sistema basado en múltiples características locales invariantes respecto de la escala.

En [28] se pueden encontrar más trabajos clásicos. El principal problema de estas técnicas basadas en características escogidas a mano es que no generalizan bien además de requerir un conocimiento muy grande del dominio, del que no siempre se dispone, lo cual no permite a estos sistemas lograr resultados tan buenos en aplicaciones del mundo real, donde las imágenes pueden cambiar su orientación, iluminación, enfoque y un largo etcétera.

Como ya mencionábamos previamente, el uso de redes neuronales artificiales en tareas de clasificación ha existido durante décadas. En los 90 empezaron a ser utilizadas en clasificación de género [29], [30], [13]. Sin embargo sus arquitecturas poco profundas, dada la falta de capacidad computacional y de volumen de datos de entrenamiento disponible, restringían su rendimiento y aplicación.

No fue hasta 2012 cuando Krizhevsky et al. [31] ganaron el ImageNet Recognition Challenge con una red convolucional y este tipo de sistemas recobraron atención. Desde entonces en estos años multitud de redes neuronales profundas han tenido enorme éxito en tareas de reconocimiento visual, entre las que se encuentra la clasificación por género. Verma et al. [32] señaló que las características que las redes neuronales convolucionales (CNN) sintetizan corresponden a las que los neurólogos identifican que usa el cerebro humano para reconocer el género. Inspirados en la técnica de "dropout" usada en CNNs, Eidinger et al. [33] entrenaron una SVM con un dropout aleatorio de ciertas características logrando resultados prometedores en la relativamente pequeña base de datos Adience, en la que Levi y Hassner [34] más tarde entrenaron y testearon una CNN no muy profunda.

En vez de entrenar con imágenes enteras, Mansanet et al. [35] entrenaron redes relativamente poco profundas usando recortes locales de las imágenes, logrando así mejores resultados que redes de igual profundidad basadas en imágenes completas. Para mejorar la precisión se han buscado redes cada vez más profundas. Pero cuanta más profundidad tenemos, más parametros necesitaremos entrenar y por tanto más imágenes son necesarias. Por este motivo recientemente se han creado o adaptado bases de datos de caras de gran escala etiquetadas por género entre otros atributos. Liu et al. [37] proporcionaron etiquetados de 40 atributos (género incluido) para los populares LFW [36] y CelebFaces [39], que han sido de enorme ayuda para el desarrollo de este trabajo.

Prácticamente todos estos artículos tratan de resolver la clasificación por rasgos de uno en uno independientemente, siendo la edad y el género los más comunes. Pero en 2015 [7] cambia esta tendencia tratando los problemas de edad y género al mismo tiempo. Teniendo en cuenta

además imágenes con distintas orientaciones, iluminación y condiciones del mundo real y ampliando el muestreo de estas al considerar distintas regiones de cada una en la clasificación.

Su sistema se basa en redes neuronales convolucionales profundas (CNNs), siguiendo el patrón que ha explotado estos últimos años en la comunidad de la visión, ya que éstas han demostrado una incomparable precisión y eficiencia en otros tipos de clasificación de imágenes. La primera aplicación práctica de CNNs fue LeNet-5 [8] ya que las redes profundas que se intentaron desarrollar a principios de los 90 resultaron impracticables debido a la potencia y el coste del hardware necesario en esa época. En los últimos años, gracias la gran y barata capacidad computacional y al aumento también exponencial de imagenes diponibles para analizar, ImageNet [10] revivió el interés en las CNNs demostrando que ahora no sólo son factibles si no terriblemente eficientes, y GoogLeNet [9] ha seguido incrementando la profundidad de estas redes para obtener aún mejores resultados.

Así, los autores de [7] utilizaron estas ventajas para construir una red con rendimiento state-of-the-art. Esto a pesar de que su diseño no es muy profundo, para no sobre alimentar la red, ya que la base de datos es relativamente pequeña. Las redes más profundas como GoogLeNet [9] funcionan mucho mejor con millones de datos, pero tienden a meter mucho ruido cuando la cantidad de datos es menor.

Finalmente, para este trabajo nos apoyaremos fundamentalmente en la CNN creada por Parkhi et al. [11], donde construyen una nueva colección de imágenes de rostros etiquetadas con nombre mayor que ninguna de dominio público disponible hasta la fecha, asemejándose al orden de colecciones industriales privados como los de Facebook o Google. Además de esto, que ya es un gran paso adelante en la tarea de clasificación de rostros, también desarrollan una nueva arquitectura para su CNN que aun manteniendo una simplicidad en su estructura que no se encuentra en otras muchas arquitecturas, sí alcanza una eficiencia y unos resultados cercanos al state-of-the-art de CNNs que usan los conuntos de datos mucho mayores antes mencionados. Debemos tener en cuenta que esta CNN no clasifica las caras por rasgos extraídos como pretendemos hacer nosotros, si no que lo hace directamente por sus identidades, es un sistema de reconocimiento facial. Aun así este sistema extrae automáticamente características de alto nivel en sus capas finales, las cuales nos serán útiles para determinar rasgos como la edad, género, etc.

3 Background teórico

3.1 Red Neuronal

Las redes neuronales son un modelo de procesado de información inspirado en los sistemas neuronales biológicos. Una red neuronal está compuesta de neuronas artificiales, organizadas en capas. En la figura 1 podemos ver un esquema de como sería una red neuronal básica.

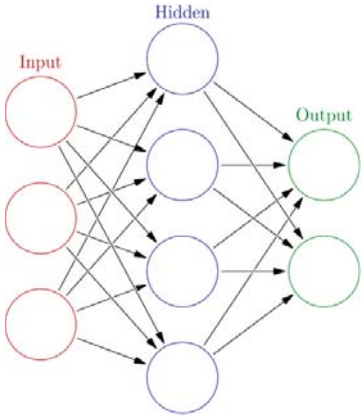

Figura 1: Imagen de una red neuronal de 3 capas con 3 nodos de entrada, 4 en la capa oculta y 2 de salida. Fuente: [40]

De forma similar a una neurona biológica, una neurona artificial recoje como entrada la señal de de muchas otras neuronas (de igual manera que una biológica recibe señales por sus conexiones sinápticas) y genera una única salida que enviará de nuevo a muchas otras otras(a través del axón en el caso de las biológicas). Las entradas son evaluadas con distintos costes, de forma que cada una afecta a la salida con una fuerza diferente. En la figura 2 tenemos la representación gráfica de una neurona artificial.

Cada capa en la red está compuesta de neuronas que comparten normalmente la misma función de activación. La elección de la función de activación es clave en el desempeño que tendrá la red. El tipo de función de activación más común fue la sigmoide durante mucho tiempo – una función cuya pendiente es muy grande en el entorno de 0 pero luego tiende a 0 según la función se acerca a -infinito o +infinito. En la figura 3 podemos ver algunos ejemplos de este tipo de funciones de activación, aunque actualmente las funciones ReLU han ganado mucha relevancia, en el siguiente apartado las explicaremos.

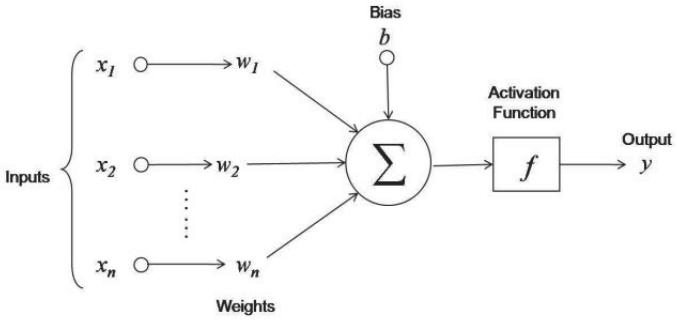

Figura 2: Una neurona artificial. Las entradas $A_1..A_n$ se multiplican por sus respectivos pesos $W_1..W_n$, estos productos son sumados y se les añade un valor de sesgo. El resultado pasa después por la función de activación, que genera la salida de la neurona. Fuente: [41]

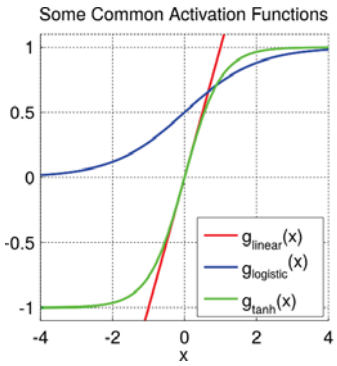

Figura 3: Funciones de activación comunes. Se aprecia como las funciones logística y tanh, 2 tipos de sigmoides cambian rápido en 0 y se aplanen en los extremos. Fuente: [42]

Las neuronas y sus conexiones forman un grafo dirigido. Una red feed-forward es una ANN cuyos nodos están conectados de forma no cíclica, mientras que una red neuronal recurrente tiene ciclos. Todas las redes que pondremos en práctica en este trabajo son de tipo feed-forward.

3.1.1 Redes neuronales convolucionales

Una red neuronal convolucional es un tipo de red feed-forward inspirada en la estrucura del cortex visual de los gatos, estudiado por Hubel and Wiesel en los 60 [17] Como hemos mencionado anteriormente las redes convolucionales se empezaron a utilizar en lo años 90 con trabajos como el deLecun et al. [8], y desde entonces los aspectos arquitectónicos básicos de estas redes no han cambiado mucho:

- Campos receptivos locales. Cada neurona recibe sus entradas de una

área pequeña de la capa anterior. Esto permite que las primeras capas de la red puedan reconocer detalles más pequeños de una imagen como contornos o esquinas, antes de sintetizarlos en características más abstractas en las capas posteriores.

- Pesos compartidos. Una capa convolucional se compone de una serie de filtros espacialmente pequeños que se desplazan (convolucionan) por todo el volumen de entrada calculando el producto escalar con cada posición de la entrada creando así un nuevo plano de datos por cada filtro. Cada uno de estos planos de activación comparte los pesos utilizados, todos sus datos han sido obtenidos mediante la misma operación sobre la capa anterior. Esto se ejemplifica en la figura 5, donde todas las flechas del mismo color representan el mismo filtro. Así, el nuevo volumen de datos habrá reducido su ancho y altura, ya que los filtros reducen la espacialidad, pero habrá aumentado su profundidad en tantas veces como filtros hayamos utilizado.

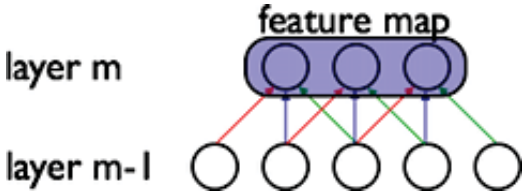

Figura 4: Un ejemplo de un filtro convolucional. Las flechas del mismo color representan pesos iguales. Fuente: [43]

- Max-pooling o Average-pooling Estas capas fueron creadas por los problemas en el cálculo de volúmenes tan grandes de datos. Para solucionarlo las capas de pooling reciben su entrada de regiones pequeñas (normalmente 2x2) no superpuestas de la capa anterior y calculan el valor máximo o medio. Esto reduce la dimensión de la capa manteniendo la invariancia respecto a la posición de las características obtenidas. [26]

- Rectified Linear Unit (ReLU) Como comentábamos antes tradicionalmente se han usado funciones sigmoides como activación de las neuronas. Pero en los últimos años se ha hecho popular la Unidad Linear Rectificada (ReLU) f(x)=max(0,x) en la que simplemente se truncan a 0 todos los valores negativos. Esto tiene ventajas como una gran aceleración de la convergencia (e.g. con un factor 6 en el descenso por gradiente (SGD) de Krizhevsky et al. ImageNet Classification with Deep Convolutional) o la obvia velocidad de computo en comparación a una sigmoide.

También tiene desventajas como que puede ser muy frágil al entrenar, pudiendo 'matar' neuronas dejando el gradiente que pase por ellas a 0 irremediablemente durante todo el entrenamiento. Nos hemos encontrado con este problema durante el desarrollo de este trabajo, pero es resoluble con un fino ajuste de la tasa de aprendizaje.

Otra solución a este problema es la modificación "Leaky ReLU" en la que en vez de dejar a 0 los valores negativos, se usa una pequeña pendiente negativa (de 0.01 o así), permitiendo a la neurona aprender aunque tenga valores negativos. Esta pendiente se puede incluso convertir en otro parámetro a aprender para cada neurona, como hicieron en 2015 Kaiming He et al. [44].

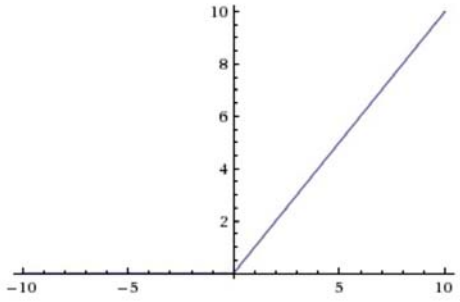

Figura 5: Gráfica de la función de activación ReLU. Fuente:[45]

- Fully Connected (FC) Las capas Fully Connected son básicamente convoluciones de "un sólo filtro" y campos receptivos globales, es decir, todas las neuronas están emparejadas 1 a 1 con todas las activaciones de la capa anterior. Esto les permite realizar una abstracción del más alto nivel al tener cada neurona una visión global de los datos a analizar, siendo así especialmente útiles en la clasificación final de la imágen.

  Usaremos en estas capas una técnica de normalización conocida como Dropout. Introducida por Srivastava et al. [46], esta técnica es muy simple y enormemente útil para prevenir el "overfitting" o sobreentrenamiento. Consiste en que según la probabilidad dada por un parametro (normalmente 50 %) las neuronas de estas capas se desactivarán. Esto obliga a la red a ser precisa incluso con la ausencia de una parte de la información, impidiendo que dependa demasiado de grupos pequeños de neuronas.

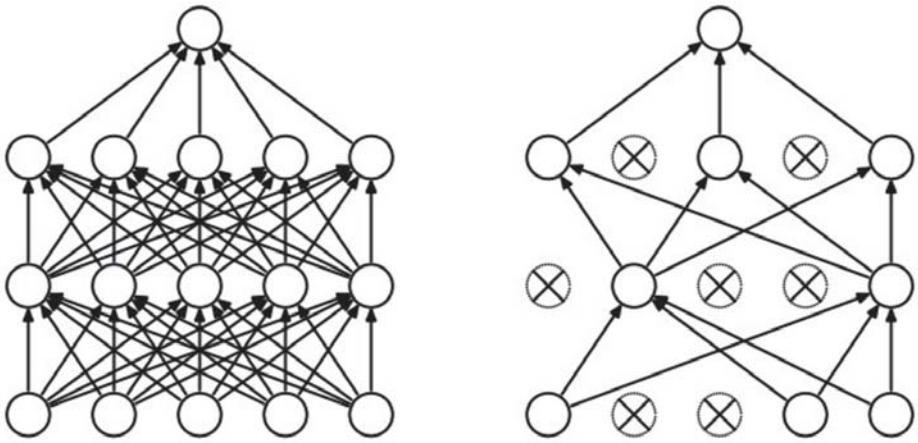

Figura 6: Red con 2 capas fully connected y la misma red tras aplicar Dropout. Fuente: [46]

3.2 Aprendizaje

El aprendizaje es la fase más importante y lo que da sentido a las redes neuronales. Se trata de ajustar todos los pesos de las neuronas de cada una de las capas de las que hemos estado hablando, a partir de grandes conjuntos de imágenes previamente etiquetadas. Esto es posible gracias al proceso conocido como 'backpropagation' por el cual tras haber introducido una imagen en la red y haber obtenido un resultado, se computa el error de este resultado y con él se actualizan los parámetros de la red según la influencia que hayan tenido en el resultado cada neurona.

Un factor vital para el desempeño de nuestra red es la función de actualización que usemos. Ésta debe alcanzar la mayor velocidad de convergencia posible sin renunciar a la eficiencia, debido al inmenso volumen de computación que va a soportar. Existen multitud de diferentes funciones de actualización y la lista sigue creciendo ya que es un área muy activa de investigación, pero aquí sólo explicaremos 2 de los más importantes:

- Stochastic Gradient Descent (SGD) with momentum 1 La forma más simple es actualizar los pesos siguiendo el descenso del gradiente (ya que normalmente queremos minimizar la función de pérdida). Siendo x el vector de pérdida y dx el gradiente, la función de actualización más simple sería $x \ += \ -lr * dx$ Donde lr es el learning rate, una constante –un hiperparámetro– y cuando es suficientemente bajo nos asegurará una mejora en la función de pérdida.

  Pero esta aproximación es demasiado lenta, por eso en la gran mayoría de los casos se usa junto a un 'momento', lo que desde una perspectiva física sería una aceleración gravitacional con su correspondiente

inercia interpretando la función de pérdida como la altura de cada punto en un valle. De esta forma el gradiente en vez de integrar directamente la posición, pasaría a modificar sólamente la velocidad, que después tendrá su efecto en la posición:

$v = mu * v - lr * dx$
$x \mathrel{+}= v$

Aquí vemos las nuevas variables velocidad v, que sería inicializada a 0 y el momento mu otro hiperparámetro que equivaldría al coeficiente de fricción y suele tener un valor de $0,9$. Este momento se encarga de mantener controlada la velocidad y es necesario para que en algún momento la función se pare en un valor estable óptimo.

Este es el tipo de estrategia que usamos al comienzo de éste trabajo, ya que era la más simple de entender. Pero depende demasiado del hiperparámetro fijado inicialmente, si éste es muy elevado la red tendrá problemas de divergencia mientras que si es muy pequeño la convergencia será demasiado lenta. Por esto cuando llegó el momento de optimizar los resultados, buscamos nuevos métodos como ADAM.

- Adaptative Moment Estimation (ADAM)

  Adam [47] es un méteodo adaptativo por-parámetro, es decir, no modifica el *learning_rate* globalmente y de la misma forma para todos los parámetros si no que lo hace individualmente para cada uno. Por ejemplo uno de los factores en los que se basa es la frecuencia con la que un parámetro recibe actualizaciones; cuando ésta es menor, las actualizaciones serán más grandes gracias a un learning_rate más alto. Esto soluciona los problemas que tenía el SGD al fijar los hiperparámetros; es necesario seguir eligiendo hiperparámetros, pero aquí son mucho menos determinantes a la hora de conseguir buenos resultados. Además de esto ADAM también usa trucos como una forma más refinada de momento que el SGD y un mecanismo de correción de sesgo. Todas estas mejoras en conjunto hacen que sea uno de los métodos más empleados en este tipo de redes actualmente, y con el que hemos realizado la mayor parte de este trabajo.

3.2.1 Softmax

Después de extraer las características de alto nivel debemos utilizar un clasificador para finalmente asignar cada imagen a la categoría que le corresponda. Para esto se pueden utilizar diversos métodos, como se ha comentado antes tradicionalmente se han usado SVMs, pero para éste trabajo hemos decidido utilizar la función Softmax que es la más común actualmente en Redes Neuronales y tiene la ventaja de que devuelve probablidades

para cada clase en vez de las puntuaciónes no calibradas y más difíciles de interpretar de las SVMs.

Esta función es básicamente una generalización de la regresión logística pero para para varias clases mutuamente excluyentes. Así nos permite mapear el vector de características final de dimensión k en un vector de probabilidades de 0 a 1 en el que la suma de todas ellas es 1.

La función softmax se define como: $f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$

Donde z_j es el valor de cada una de las neuronas (para $j = 1, ..., k$) y el denominador actua como normalizador.

4 Herramientas, Tecnologías y Bases de datos

4.1 caffe, DIGITS

Caffe es uno de los frameworks de deep learning más utilizados actualmente. Desarrollado por el Berkeley Vision and Learning Center (BVLC) de la universidad de Berkley [48], está desarrollado en C++ y tiene interfaz en Matlab y Python. Caffe está centrado principalmente en el campo de la visión artificial, lo que le permite ser el más rápido en este campo.

También tiene algunos inconvenientes, como que la documentación es pobre, teniendo muchas veces que mirar directamente el código para saber lo que realmente hace. También es complicada la instalación y puesta a punto, que puede crear algún que otro quebradero de cabeza, debido a las numerosas dependencias a resolver.

Pero la mayor ventaja que tiene, a parte de ser rápido y robusto, es lo muy extendido que está su uso, al ser la herramienta líder para sistemas de visión artificial. Esto hace que podamos encontrar multitud de modelos ya entrenados disponibles en el "caffe model zoo" un sistema para compartir modelos con diferentes arquitecturas de forma estandarizada desarrollados en Caffe.

Por todo esto es por lo que hemos elegido utilizar Caffe para este trabajo, controlándolo y modificándolo a través de su interfaz en Python. En Python también se han desarrollado todos los scripts para la preparación de las bases de datos y se ha realizado la evaluación de los modelos. El entrenamiento lo empezamos haciendo en Python pero más tarde se pasó

a DIGITS.

El NVIDIA Deep Learning GPU Training System (DIGITS) es una interfaz gráfica (web) para entrenar redes neuronales en Caffe orientada obviamente a tarjetas gráficas de NVIDIA. Por un lado ofrece interesantes herramientas como gráficas a tiempo real del rendimiento del entrenamiento y visualizaciones de las activaciones neuronales para cada capa. Esto último es especialmente útil para saber con precisión qué es lo que está pasando en el interior de la red, y nos ha servido a lo largo de este trabajo para localizar algunos problemas.

Por otro lado DIGITS también simplifica muchas tareas relativas a la sistemización de los experimentos. A lo largo de los meses de trabajo hemos llegado a realizar más de 60 experimentos distintos (redes entrenadas) y las posibilidades que ofrece DIGITS a la hora de organizarlos en grupos, compararlos fácilmente, cambiar rápidamente la configuración, etc., se agradecen mucho.

4.2 Red VGG-Face

Todas las redes puestas en marcha en este trabajo son modificaciones de VGG-Face [11]. Elegimos esta red para basar nuestro trabajo por tratarse de una arquitectura de primer nivel –quedó finalista en el ImageNet Challenge 2014 sólo superada por GoogLeNet– cuyos autores han puesto a disposición pública. Esta red está basada en VGG-Very-Deep-16 descrita en [49] y al igual que ella está compuesta de 16 capas de concolución, que junto a las de pooling, ReLU y evaluación (softmax) hacen un total de 38. Elegimos usar esta arquitectura por se una de primer nivelPodemos observar la arquitectura de esta red en 7.

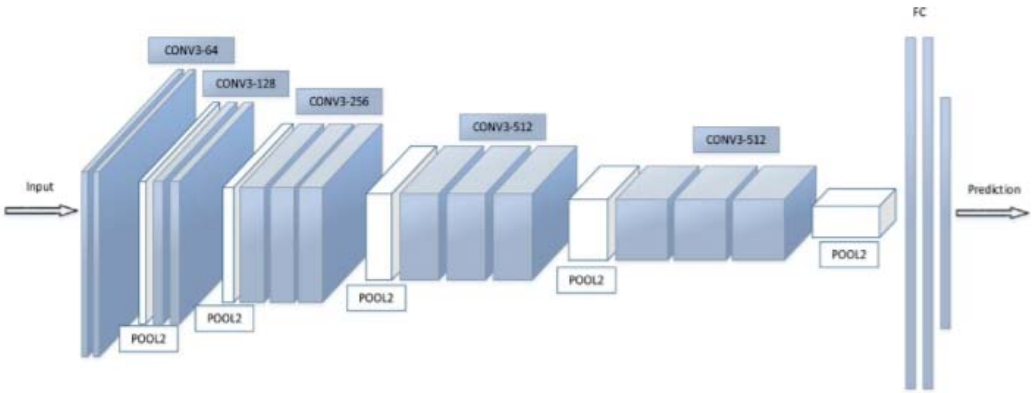

Figura 7: Arquitectura de la red VGG-Face. Fuente:[50]

Zisserman y compañía propusieron este tipo de arquitectura buscando aumentar la profundidad sin que el número de parámetros de la red se disparara, reduciendo por tanto el tamaño de los filtros para ganar eficiencia. Así este modelo se compone básicamente de filtros de convolución de 3x3 y filtros de max-pooling de 2x2. Para simular convoluciones mayores de 5x5 o 7x7 se apilan 2 o 3 capas de convolución seguidas de una de pooling, así conseguimos aumentar la profundidad de la red sin que el número de parámetros crezca en exceso. La siguiente tabla proporciona detalles adicionales, donde podemos observar cómo aumenta el volumen y por tanto el número de parámetros según vamos utilizando más filtros en cada capa.

Layer (1-11)	Volume	Parameters	Layer (12-22)	Volume	Parameters
INPUT	224 × 224 × 3	0	CONV3-512	28 × 28 × 512	(3 * 3 * 256) * 512 = 1,179,648
CONV3-64	224 × 224 × 64	(3 * 3 * 3) * 64 = 1,728	CONV3-512	28 × 28 × 512	(3 * 3 * 512) * 512 = 2,359,296
CONV3-64	224 × 224 × 64	(3 * 3 * 64) * 64 = 36,864	CONV3-512	28 × 28 × 512	(3 * 3 * 512) * 512 = 2,359,296
POOL2	112 × 112 × 64	0	POOL2	14 × 14 × 512	0
CONV3-128	112 × 112 × 128	(3 * 3 * 64) * 128 = 73,728	CONV3-512	14 × 14 × 512	(3 * 3 * 512) * 512 = 2,359,296
CONV3-128	112 × 112 × 128	(3 * 3 * 128) * 128 = 147,456	CONV3-512	14 × 14 × 512	(3 * 3 * 512) * 512 = 2,359,296
POOL2	56 × 56 × 128	0	CONV3-512	14 × 14 × 512	(3 * 3 * 512) * 512 = 2,359,296
CONV3-256	56 × 56 × 256	(3 * 3 * 128) * 256 = 294,912	POOL2	7 × 7 × 512	0
CONV3-256	56 × 56 × 256	(3 * 3 * 256) * 256 = 589,824	FC-1	1 × 1 × 4096	7 * 7 * 512 * 4096 = 102,760,448
CONV3-256	56 × 56 × 256	(3 * 3 * 256) * 256 = 589,824	FC-2	1 × 1 × 4096	4096 * 4096 = 16,777,216
POOL2	28 × 28 × 256	0	FC-3	1 × 1 × 2622	4096 * 2622 = 10,739,712

Figura 8: Capas detalladas de la red VGG-Face. Fuente:[50]

VGG-Face está originalmente entrenada para reconocer a 2622 personas diferentes y así sus 2622 salidas en la última capa corresponden a estas clases. También está configurada para recibir como entrada imágenes de 224x224 pixeles con color codificado en RGB. La base de datos utilizada para su entrenamiento, que contiene 982.803 imágenes, es casi 3 órdenes de magnitud más grande que cualquier otra colección de imágenes de caras puesta a disposición pública hasta entonces. Sólo las bases de datos privadas de compañías como Facebook o Google son mayores, siendo la de Facebook por ejemplo también del orden de los millones. Los investigadores lograron construir una base de datos de este tamaño siguiendo un proceso que combina métodos manuales y automáticos y que se resumen de la siguiente manera:

1. Obtuvieron una lista de gente famosa de the Internet Movie Database (IMDb)

2. Recolectaron imágenes de estos famosos usando La búsqueda de imágenes de Google

3. Eliminaron las fotos de gente que ya estaban presentes en LFW y Youtube Faces

4. Filtraron las imágenes en 2 fases, primero de forma automática y luego manual para mejorar la pureza de los clusters y eliminar duplicados.

4.3 Bases de Datos

Aquí describiremos las bases de datos que hemos utilizado a lo largo del proyecto.

4.3.1 Labeled Faces on the Wild (LFW)

Creada por Huang et al. en 2007, Labeled Faces On the Wild (LFW) es una base de datos de imágenes faciales de uso muy común para la evaluación de algoritmos de reconocimiento facial. Contiene 13233 imágenes RGB de 5749 individuos diferentes (una media de 2.3 imágenes por persona). Las imágenes tienen una dimensión de 250x250. También podemos encontrar definidos pares de imágenes divididos en conjuntos para entrenar y testear los sistemas de reconocimiento facial [36].

Para replicar los resultados de Deep Face [11], seguimos como ellos el protocolo de evaluación estándar definido como "unrestricted setting" que permite usar datos externos a LFW para el entrenamiento. Además para la evaluación utilizamos el "equal error rate" (EER) definido como la tasa de error en la curva ROC donde la tasa de falsos positivos es igual a la de falsos negativos. La ventaja frente a la medida de precisión estándar es que es independiente al umbral τ que escojamos.

4.3.2 Annotated Facial Landmarks in the Wild (AFLW)

Annotated Facial Landmarks in the Wild (AFLW), desarrollada por los investigadores M Koestinger et al. de la Technical University of Graz [51] nos proporciona una colección de imágenes faciales recolectadas en la web, exhibiendo una gran variedad en apariencia (orientación, expresión, etnia, edad, sexo) y condiciones medioambientales. En total alrededor de 25k caras son anotadas con hasta 21 puntos de referencia por imagen.

A parte de los 21 puntos de referencia, este conjunto incluye información sobre la orientación de los rostyros, los ángulos en los que la cabeza está inclinada, y es muy utilizada en sistemas de estimación de orientación como en los que están trabajando compañeros del laboratorio de visión computacional. Aunque esto no sea el ámbito de estudio de éste trabajo,

esa variedad en la orientación de las imágenes le da una "dificultad" extra a este conjunto que nos resultó útil para evaluar nuestro sistema cuando los resultados que obteníamos en el reconocimiento de género resultaban difícilmente mejorables.

4.3.3 Large-scale CelebFaces Attributes (CelebA)

CelebFaces Attributes Dataset (CelebA) [37] es la principal base de datos que hemos utilizado a lo largo de este trabajo, con la que se han entrenado y testeado la mayoría de redes puestas en funcionamiento. Fue desarrollada como una ampliación de 40 atributos faciales binarios a la ya popular base de Datos CelebFaces [39].

Este conjunto de gran escala contiene 202.599 imágenes de 10.177 identidades. A pesar de su tamaño relativamente grande, la mayoría de las imágenes son fotos de retrato con fondos simples tomadas por fotógrafos profesionales y por lo tanto son consideradas menos "difíciles" que las de otros conjuntos.

Para nuestros experimentos hemos usado la misma división en entrenamiento/validación/evaluación (train/val/test) que sugieren en [37], ya que separa los conjuntos por identidades, añadiendo un nivel más de objetividad para prevenir el overfitting.

5 Modelos

Hagamos un resúmen de las redes en las que hemos trabajado durante el desarrollo de este proyecto:

- *Genre_CelebA y Genre_AFLW*

  Esta es la red a la que más tiempo hemos dedicado, básicamente toda la primera mitad del proyecto. Se trataba de demostrar que podíamos utilizar una Red Neuronal de grandes dimensiones entrenada previamente como un extractor de características para una tarea tangencialmente relaccionada con la tarea original. En este caso optamos por empezar creando el mejor clasificador por género posible, y como veremos lo hicimos reentrenando únicamente las 3 capas fully connected de la red con el conjunto de imágenes CelebA.

  También montamos una versión alternativa de esta red utilizando la base de datos AFLW en vez de CelebA cuando alcanzamos el estado del arte con ésta última. Las capas reentrenadas son las mismas y se

trataba de comprobar si la red rendía igual de bien en en un conjunto más complicado, en el que las caras fotografiadas no están todas de frente si no que tienen múltiples niveles de inclinación en los 3 ejes.

- *Multi*4

  Tras los buenos resultados del clasificador por género el siguiente reto era crear una red que clasificara múltiples atributos al mismo tiempo. Decidimos intentarlo con 4 y además de género y edad (al ser los atributos binarios se divide en 'jóven' y 'viejo'), escogimos 'llevar bigote' y 'labios pintados' de entre los 40 diferentes atributos que nos ofrece CelebA.

  El método con el que hemos puesto en práctica esta clasificación múltiple consiste en realizar un producto cartesiano de nuestros 4 conjuntos binarios, obteniendo así $2^4 = 16$ nuevas clases que engloban todas las combinaciones posibles de nuestros atributos. De estas 16 clases la mitad representan conjuntos despreciables, es decir, estas 8 clases (las que incluyen las intersecciones "mujeres" con "bigote" y "hombres" con "labios pintados") sólo suman alrededor del 0,3 % del total de imágenes de entrenamiento, por lo que las ignoraremos en nuestro sistema clasificador.

  La arquitectura de esta red permanece entonces idéntica a la anterior con la diferencia de que la última capa clasificadora en vez de 4 tendrá 8 salidas, correspondientes a nuestras 8 clases. Por tanto, quedará en nuestras manos la organización y definición de estas 8 clases en el etiquetado de los datos de entrenamiento y la evaluación particular de la precisión de la red para cada uno de los atributos, ya que la precisión estándar que obtendremos de esta red con respecto a nuestras 8 "meta-categorías" no nos será útil en si misma.

- *GenAge*

  Al ver que la red Multi4 no daba los resultados esperados decidimos bajar un escalón y simplificar nuestra red de clasificación multietiqueta. Esta red clasificará únicamente 2 atributos, género y edad. La arquitectura de la red y el método de partición son los mismos, realizamos el producto cartesiano de nuestros 2 atributos para quedarnos con 4 clases en las que clasificaremos nuestras imágenes.

- Redes de clasificación unitarias para cada atributo

  Tras optimizar todo lo posible los resultados de las redes multietiqueta, sólo quedaba evaluar su eficiencia en comparación con redes

que clasifican individualmente cada uno de los atributos, y esas son las últimas redes que hemos puesto en marcha para este trabajo. Estas redes son idénticas al clasificador de género inicial, replicando todos los detalles de su configuración, ya que estaban optimizados al máximo para la clasificación de un sólo atributo, sea este *género*, *edad*, *bigote o labios pintados*

6 Metodología

Nuestro flujo de trabajo consta de 3 fases. En primer lugar, preparamos las imágenes de entrenamiento y evaluación adaptándolas a las características necesarias. Después estas imágenes son empleadas para entrenar las redes construidas, y finalmente estas redes son evaluadas con otros el correspondiente conjunto de imágenes de test.

6.1 Preprocesado

En este bloque extraemos toda la información que necesitamos de nuestras bases de datos y la sometemos a la preparaciones necesarias para su uso por las redes neuronales correspondientes. El primer problema al que nos enfrentamos para esta tarea es que en conjuntos como CelebA sólo el 30 % de cada imagen representa información útil, el rostro. Esto significa que debemos recortar las imágenes para que el aprendizaje de la red sea óptimo, y hemos encontrado distintos métodos para sacar este recorte.

En su versión original, la que teníamos a nuestra disposición en el laboratorio, CelebA sólo nos daba 5 puntos de referencia (landmarks) para los 2 ojos, nariz, y las 2 comisuras de la boca, y a partir de ellas, en proporción a las distancias entre estos puntos estimamos las dimensiones de las caras para recortar las imágenes. Esta solución no era muy óptima debido a la gran variedad de proporciones en los rostros que hacía imposible encontrar una fórmula que se adaptara a todos ellos.

Afortunadamente un tiempo después nos encontramos con la existencia de una ampliación de CelebA que incluye los marcos delimitadores (bounding boxes) calculados por un algoritmo de detección de rostros. Esta otra solución también trajo problemas, como que cuando las caras estaban en posición horizontal o incluso inclinadas hacia abajo el detector fallaba por lo que tuvimos que retocar bastante el script para calcular unos marcos aceptables. Pero en general estos nuevos marcos funcionaban mejor que nuestra precaria solución y nos quedamos con ellos.

Después procedemos a reescalar las imágenes a 240x240p, que es lo que acepta nuestra red como entrada. Por otro lado en este tipo de sistemas es típico el uso de técnicas de Data Augmentation para aumentar el muestreo de datos aplicando a las imágenes originales rotaciones, traslaciones, recortes, etc. Nosotros hemos prescindido de éstas técnicas, debido por un lado a que no tenemos problema de escasez de datos, nuestras bases de datos son bastante grandes, y que además el hardware que utilizamos no admite más volumen de computación, cuando ya cada experimento tarda en entrenarse en torno a las 20 horas.

En cuanto a la organización y el formato de estas nuevas bases de datos, ha dependido de lo que requiere DIGITS, la interfaz que usamos para entrenar nuestras redes. En cuanto a la división de las imágenes en conjuntos de entrenamiento, validación y evaluación, DIGITS requiere que vayan divididos en carpetas separadas, mientras que para la división por categorías DIGITS acepta 2 distintos formatos, entre los cuales hemos transitado:

- Durante la primera fase de este trabajo en la que nuestra red sólo trataba de clasificar el género de los individuos, optamos por la forma más simple de organización, separar las imágenes en 2 subcarpetas (en cada uno de los conjuntos /test, /train, /val) por categoría, es decir hombre y mujer. En las sucesivas fases donde empezamos a entrenar redes más complejas, con hasta 8 categorías diferentes nos dimos cuenta de que el sistema por carpetas no era nada eficiente en el sentido de que para cada nueva red tendríamos que copiar la base de datos entera según la nueva clasificación por categorías.

- De esta forma nos pasamos a un nuevo sistema de una única base de datos donde las imágenes quedan todas en la misma carpeta ya sea /test, /val o /train y no necesitan volver a tocarse. Por otro lado cada red tiene un archivo de texto donde empareja cada imágen con la clase a la que pertenece. De esta manera para cada nueva red sólo necesitamos crear un archivo que defina la nueva distribución por categorías y todas acceden a la misma base de datos, facilitando y agilizando la tarea.

Como hemos explicado en CelebA utilizamos la división sugerida por los investigadores que separa sus 202,599 imágenes en 162,770 (80 %) para el entrenamiento, 19,867 (10 %) para validación y 19,962 (10 %) para evaluación , manteniendo las identidades de cada conjunto independientes. Sin embargo el conjunto AFLW no provee de una división similar, por lo que hicimos una propia dividiendo las imágenes aleatoriamente en 3 conjuntos con el 85 %, 10 % y 5 % respectivamente. En este paso es importante hacer

la separación una sóla vez y mantener el conjunto de evaluación sin alterar para que todos los experimentos se evalúen en igualdad de condiciones.

6.2 Entrenamiento

En este bloque las redes aprenderán a reconocer rasgos faciales a través de repetidas actualizaciones de sus parámetros gracias a las imágenes de entrenamiento etiquetadas que le proporcionamos. Como ya hemos explicado, todas nuestras redes son modificaciones de $VGG - Face$ en las que sólo reentrenamos las últimas capas. De esta forma reencaminando los descriptores de alto nivel que $VGG - Face$ utiliza para identificar personas hacia las tareas de clasificación en distintos atributos que requerimos en nuestro caso.

Inicialmente reentrenábamos únicamente las 3 capas "fully connected", partiendo en las 2 primeras de los valores que tienen en $VGG - Face$ como valor inicial y reentrenando la tercera y última de 0, ya que su salida varía en función del número de categorías con el que estemos trabajando. En este primer caso eran 2, ya que sólo clasificábamos por género.

Esto nos dio buen resultado en esa primera red, *Genre* pero cuando pasamos a probar redes que clasificaban múltiples atributos al mismo tiempo, los resultados no fueron nada satisfactorios. Esto lo atribuimos a que estas redes necesitan adquirir características de alguna manera más complejas, así que empezamos a probar a reentrenar las últimas capas de convolución además de las fully connected.

La búsqueda de cuántas de estas capas de convolución entrenar y con qué "intensidad" (learning rate) fue ciertamente complicada. La red original está entrenada con una cantidad inmensa de imágenes, alrededor del millón, y por tanto los valores de sus neuronas están enormemente refinados y calibrados. Esto combinado con la fragilidad propia de las neuronas de activación tipo ReLU, que al recibir un cambio súbito de gradiente pueden actualizar sus pesos de forma que no se volverá a activar para ninguna imagen de nuevo, quedando efectivamente "muerta", hace que al intentar reentrenar estas capas la red rompiera muy fácilmente. Al morir muchas neuronas de una capa hacen mucho más probable que mueran las de la capa siguiente, creando un efecto cascada que llega al punto en el que todas las neuronas de una capa están muertas, no dejando pasar ninguna información y creando una red inútil que devuelve el mismo resultado a cualquier imágen de entrada. Curiosamente, incluso este resultado único se optimiza en las capas de la red posteriores al "corte", siendo la estimación de la clase de cada una de las imágenes de evaluación el mismo vector de probabili-

dades que coinciden precisamente con las proporciones de cada clase en el conjunto de imágenes de entrenamiento.

Tras mucho ensayo y error la solución más óptima a la que llegamos es entrenar las últimas 6 capas de convolución, que corresponden a los 2 últimos grupos grupos de los 5 que hay en total (como ya explicamos la arquitectura VGG agrupa series de convoluciones pequeñas de 3 en 3 para simular convoluciones mayores). Pero la clave está en que entrenamos estas capas con un learning rate enormemente reducido. Si ya el learning rate inicial con el que entrenamos las capas fully connected suele ser de 10^{-5} en estas capas lo reducimos en 2 ordenes de magnitud, dejándolo en 10^{-7}.

6.3 Evaluación

Mientras que con el entrenamiento DIGITS nos ha sido inmensamente útil y ha sido prácticamente la única herramienta que hemos usado para ello, no nos ha servido igual de bien para la evaluación.

DIGITS dispone de herramientas de evaluación para una o un conjunto de varias imágenes. El modo de evaluación de imágenes únicas es el único que hemos llegado a usar porque muestra las visualizaciones de activación neuronal de cada una de las capas. Esto nos resultó especialmente útil a la hora de seleccionar las capas que íbamos a reentrenar, ya que nos permite ver qué tipo de información está detectando cada filtro y en el caso de las neuronas que mueren nos permite identificar exactamente dónde empiezan a morir.

Sin embargo a la hora de evaluar nuestros conjuntos de imágenes para conocer el rendimiento exacto de nuestras redes, la herramienta de DIGITS resulta inútil, ya que es tremendamente lenta para evaluar estas imágenes.

Por tanto recurrimos a utilizar caffe desde la interfaz estándar de Python, creando un script para cada red a evaluar que en el caso de las redes que clasifican múltiples atributos además de darnos la precisión global nos da la precisión particular para cada atributo. Otro añadido a este script que utilizamos sobretodo al principio es el guardado de las imágenes que el sistema ha clasificado erróneamente, ya que queríamos conocer en qué tipo de imágenes estaba fallando nuestro sistema ya hacer mejores aproximaciones a partir de ahí.

7 Resultados

A continuación presentaremos los resultados que han obtenido las distintas redes que hemos puesto en marcha en este trabajo. Como hemos comentado, el método de evaluación ha sido a través de un script de Python que simplemente introduce las imágenes de nuestro conjunto de test en la red clasificadora y nos decuelve un porcentaje de acierto de las predicciones realizadas. En el caso de las redes de clasificación multi-etiqueta, el script también se encarga de descomponer cada una de las "meta-clases" para obtener los resultados individualizados de el porcentaje de acierto para cada uno de los atributos analizados.

- *Genre_CelebA y Genre_AFLW*

  Tras múltiples iteraciones modificando los hiperparámetros de la red, ésta es la configuración óptima a la que llegamos:

  - Lerning rate inicial: 10^{-3}
  - Política de learning rate: Step Down (step size 33 %)
  - momentum: 0.9
  - weight decay: 10^{-5}

  Con esta configuración la precisión de la red *Genre_CelebA* es del 98,53 %.

  Este resultado iguala o incluso supera estado del arte en clasificación de género con el conjunto CelebA que encontramos en [38] situado en el 98 %.

  Con respecto al conjunto AFLW, que como ya explicamos es más complicado debido a que los rostros se encuentran en una gran variedad de inclinaciones, la precisión de la red *Genre_AFLW* es de 87,74 %.

- *GenAge*

  En esta red conseguimos superar el problema de encontrar el punto de equilibrio entre entrenar un mayor número de neuronas y hacerlo de la forma menos agresiva posible para que las neuronas no murieran por un cambio brusco del gradiente rompiendo la red.

  La configuración de esta red sigue los parámetros de la anterior con algunas diferencias. Por un lado el learning rate inicial pasa a ser 10^{-5} Además, al reentrenar una sección mayor de la red, las 6 últimas capas de convolución encontradas en los sectores 4 y 5 se entrenan multiplicando el learning rate por un factor de disminución extra de 10^{-2},

dejando su learning rate inicial en 10^{-7}.

Con esta configuración, los resultados de la red GenAge son:

- Género - 97,83 %
- Edad - 86,38 %
- Ambos atributos acertados: 84,61

Como podemos observar en la clasificación de género nos quedamos un punto por debajo del resultado respecto a la red que lo clasifica individualmente. También la edad queda ligeramente por debajo de su estado del arte. Esto demustra por tanto que al menos en nuestro caso con estas condiciones no se cumple la hipótesis de que una red multi-etiqueta con atributos de alguna forma correlacionados mejoraría los resultados respecto a las redes individuales.

- *Multi*4

  Esta red empezamos a desarrollarla antes que GenAge, pero no conseguíamos configurarla correctamente y no llegamos a obtener resultados significativos, por ello decidimos dejarla atrás y pasar a la versión simplificada de sólo 2 atributos.

  Sin embargo, tras solventar el problema de la muerte de neuronas en GenAge, volvimos con esa configuración a volver a intentarlo en Multi4 retocando únicamente el learning rate reduciéndolo hasta 10^{-6}(10^{-8} en las capas de convolución) y de esta manera sí conseguimos finalemnte unos buenos resultados.

  Las precisiones obtenidas por la red Multi4 son:

  - Género - 97,25 %
  - Edad - 85,64 %
  - Bigote - 96,38 %
  - Labios pintados - 92,29 %
  - Acierto de los 4 atributos simultáneamente - 84,61

  Efectivamente, los resultados vuelven a empeorar ligeramente respecto respecto a los de las redes que clasifican menos atributos. Los atributos de bigote y labios pintados también desempeñan peor que las redes que los analizan individualmente. Esto dibuja una tendencia que si bien refuta nuestra hipótesis, tiene todo el sentido del mundo y esto es que una red una red tiene una "capacidad de aprendizaje",

y cuantos más atributos queramos aprender simultáneamente con esa red, peor aprenderá cada uno de ellos.

8 Conclusiones y Trabajo Futuro

Durante el desarrollo de este proyecto se han podido evaluar los diferentes bloques que forman un sistema de reconocimiento de atributos faciales. Hemos podido observar el uso de redes convolucionales es una alternativa muy potente como extractores de características en problemas de visión computacional en general y de reconocimiento facial en particular.

Hemos comprobado también, después de estudiar distintas variantes y configuraciones, que desarrollar un sistema de extracción de atributos faciales basado en en una red convolucional de gran escala entrenada previamente como pueda ser VGG-Face nos permite apoyarnos en hombros de gigantes para alcanzar resultados del más alto nivel en problemas particulares sin necesariamente disponer de los recursos técnicos computacionales que requeriría entrenar una red de 0 por nuestra cuenta.

Otra de las lecciones que hemos aprendido es la enorme importancia de escoger y calibrar con mucho tacto los learning rates con los que va a trabajar la red, incluso utilizando funciones de actualización de parámetros adaptativas como Adam. En una pequeña variación de este learning rate puede estar la diferencia entre que una red pueda tener buenos resultados o que se rompa por completo debido a la muerte masiva de neuronas.

La hipótesis que queríamos comprobar no se ha cumplido, esto es, que entrenar una red para extraer diferentes atributos al mismo tiempo traería ventajas respecto a los sistemas de extracción con redes individuales para cada atributo. Esto sería debido a que por ejemplo las características que permiten reconocer a una niña en vez de a un niño no son exactamente las mismas que permiten reconocer a una anciana en vez de a un anciano, por lo tanto la agrupación de los datos que provoca analizar género y edad simultáneamente permitiría que estos dos atributos se "ayudasen" mutuamente. Quiere decir, que al reconocer la red que se trata de una persona anciana, podría utilizar un set diferente de neuronas para reconocer el género que el que usa cuando se activan las neuronas que señalan que tratamos con una persona jóven. Esta hipótesis se extrapolaría entonces a cualquier conjunto de atributos que compartiesen algún tipo de correlación en los datos de los que provienen.

Mientras que nuestros experimentos han refutado esta hipótesis al

menos para las condiciones en las que trabajamos nosotros, sí que es cierto que las diferencias en los resultados han sido mínimas, la precisión en la clasificación de los diferentes atributos en las redes múltiples estaba a menos de un punto porcentual por debajo de las redes individuales. Probando así al menos que una red de esta envergadura tiene capacidad de sobra para extraer múltiples atributos faciales de forma simultánea. Si bien también hemos observado una tendencia lógica a que cuantos más atributos analiza una red, con menor precisión los clasifica.

Con respecto a nuevas líneas de investigación futuras, hay varios caminos que consideramos seguir durante el desarrollo del proyecto y finalmente no hicimos, ya sea por falta de tiempo, de recursos computacionales o simplemete de capacidades técnica como la de modificar el código base de Caffe introduciendo nuestras propias capas.

Consideramos por ejemplo entrenar la red VGG-Face desde 0 con nuestras bases de datos esto es, CelebA y AFLW, para comprobar si podíamos alcanzar unos resultados comparables a los obtenidos. Esta idea quedó descartada al tener en cuenta la profundidad de la red en cuestión y el hecho de que nosotros trabajamos con una tarjeta gráfica modesta de 4GB que ya tarda alrededor de 20 horas en entrenar únicamente las últimas capas en nuestros experimentos.

Otra de estas vías es la posibilidad de otorgar pesos según la proporción de imágenes pertenecientes al una clase respecto al total a las imágenes de dicha clase, compensando de esta manera los desequilibrios que encontramos cuando por ejemplo en la red Multi4 algunas de las categorías tenían 10 veces más instancias que otras.

Finalmente existen también otros métodos para realizar la clasificación multi-etiqueta en redes neuronales a parte de el producto cartesiano de las clases. Uno de ellos sería dividir la salida de la red en distintas capas clasificadoras paralelas, que trabajarían de forma independiente y para las que habría que establecer una política de ponderación de sus errores y obtener un error común con el que entrenar la red.

9 Referencias

[1] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops, June 2015.

[2] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(11):1955–1976, Nov 2010.

[3] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pages 762–767, Jun 1994.

[4] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security, 9(12):2170–2179, Dec 2014.

[5] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(3):541–547, March 2008.

[6] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3, NIPS-3, pages 572–577, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[7] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops, June 2015.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1106–1114. 2012.

[11] O. M. Parkhi, A. Vedaldi, A. Zisserman Deep Face Recognition British Machine Vision Conference, 2015

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E., Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

[13] S. Gutta and H. Wechsler. Gender and ethnic classification of human faces using hybrid classifiers. In Neural Networks, 1999. IJCNN'99. International Joint Conference on, volume 6, pages 4084–4089. IEEE, 1999

[14] B. Moghaddam and M.-H. Yang. Learning gender with support faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):707–711, 2002.

[15] A. Jain, J. Huang, and S. Fang. Gender identification using frontal facial images. In 2005 IEEE International Conference on Multimedia and Expo, pages 4–pp. IEEE, 2005.

[16] S. Baluja and H. A. Rowley. Boosting sex identification performance. International Journal of computer vision, 71(1):111–119, 2007.

[17] Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 1962;160: 106–154. doi:10.1113/jphysiol.1962.sp00683

[18] G. Shakhnarovich, P. A. Viola, and B. Moghaddam. A uni- fied learning framework for real time face detection and classification. In Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 14–21. IEEE, 2002.

[19] C. Ben Abdelkader and P. Griffin. A local region-based approach to gender classification from face images. computer vision and pattern recognition-workshops, 2005. cvpr workshops. In IEEE Computer Society Conference on. IEEE, 2005.

[20] Z. Yang, M. Li, and H. Ai. An experimental study on automatic face gender classification. In 18th International Conference on Pattern Recognition (ICPR'06), volume 3, pages 1099–1102. IEEE, 2006.

[21] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza. Gender recognition from face images with local wld descriptor. In 19th International Conference on Systems, Signals and Image Processing (IWSSIP), pages 417–420. IEEE, 2012.

[22] M. Toews and T. Arbel. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(9):1567–1581, 2009.

[23] A. J. O'toole, T. Vetter, N. F. Troje, and H. H. Bulthoff. Sex classification is better with three-dimensional head structure than with image intensity information. Perception, 26(1):75– 84, 1997.

[24] C. Perez, J. Tapia, P. Estevez, and C. Held. Gender classifica- tion from face images using mutual information and feature fusion. International Journal of Optomechatronics, 6(1):92– 119, 2012.

[25] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. Image and vision computing, 16(5):295–306, 1998.

[26] Lee H, Grosse R, Ranganath R, Ng AY. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: ACM; 2009. pp. 609–616. doi:10.1145/1553374.1553453

[27] C. Shan. Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters, 33(4):431–437, 2012.

[28] D. Reid, S. Samangooei, C. Chen, M. Nixon, and A. Ross. Soft biometrics for surveillance: an overview. Machine learning: theory and applications. Elsevier, pages 327–352, 2013.

[29] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In NIPS, volume 1, page 2, 1990.

[30] B. Poggio, R. Brunelli, and T. Poggio. Hyberbf networks for gender classification. 1992.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[32] A. Verma and L. Vig. Using convolutional neural networks to discover cogntively validated features for gender classification. In Soft Computing and Machine Intelligence (ISCMI), 2014 International Conference on, pages 33–37. IEEE, 2014.

[33] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security, 9(12):2170–2179, 2014.

[34] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 34–42, 2015.

[35] J. Mansanet, A. Albiol, and R. Paredes. Local deep neural networks for gender recognition. Pattern Recognition Letters, 70:80–86, 2016.

[36] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[37] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

[38] MOON: A Mixed Objective Optimization Network for the Recognition of Facial Attributes Rudd, Ethan; Günther, Manuel; Boult, Terrance eprint arXiv:1603.07027 03/2016

[39] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In Advances in Neural Information Processing Systems, pages 1988–1996, 2014.

[40] https://commons.wikimedia.org/wiki/File:Colored$_neural_network.svg$2,3,3$Artificial Neuro$
$http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node$

[41] [42] dustinstansbury. Derivation: Derivatives for Common Neural Network Activation Functions. In: The Clever Machine [Internet]. 9 Sep 2014 [citado el 12 de Mayo 2017]. Disponible: https://theclevermachine.wordpress.com/2014/09/08/derivation-derivatives-for-common-neural-network-activation-functions/

[43] Convolutional Neural Networks (LeNet) — DeepLearning 0.1 documentation [Internet]. [citado el 17 de Mayo 2017]. Disponible: http://deeplearning.net/tutorial/lenet.html

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015. CoRR abs/1502.01852.

[45] Stanford University CS231n: Convolutional Neural Networks for Visual Recognition. [citado el 27 de Mayo 2017]. Disponible: http://cs231n.github.io/neural-networks-1/

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15 (2014) 1929-1958.

[47] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization CoRR 2014 abs/1412.6980

[48] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor Caffe: Convolutional Architecture for Fast Feature Embedding 2014, arXiv preprint arXiv:1408.5093

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014

[50] El Khiyari, H. and Wechsler, H. (2016) Face Recognition across Time Lapse Using Convolutional Neural Networks. Journal of Information Security, 7, 141-151. doi: 10.4236/jis.2016.73010.

[51] Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization Martin Koestinger, Paul Wohlhart, Peter M. Roth and Horst Bischof In Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, 2011

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Thu Jul 06 18:59:40 CEST 2017
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)