

Práctica 5

Redes Neuronales en Keras



Asignatura: Inteligencia Artificial

Fecha: 5/01/2020

Autor: Diego Marco Beisty, 755232

Introducción

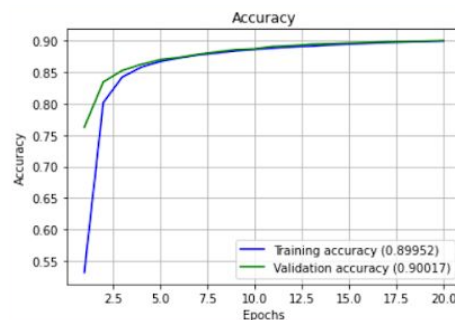
A lo largo de esta práctica se han probado y comentado distintas redes neuronales con distintos parámetros (función de activación, algoritmo de entrenamiento, número de neuronas de la/las capas ocultas).

Además en caso de producirse sobreajuste se han aplicado técnicas para reducirlo.

Perceptrón

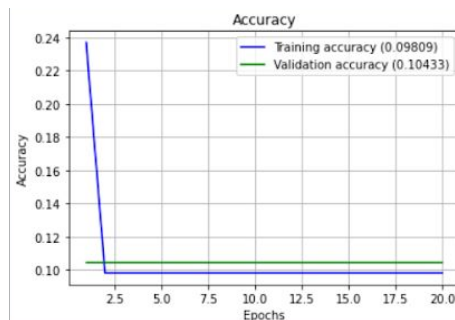
Un perceptrón se compone de una neurona capaz de resolver problemas separables linealmente. Es por ello que se ha configurado inicialmente una capa de 10 perceptrones para que cada uno sea capaz de identificar un dígito numérico distinto.

- Se ha probado el algoritmo “SGD” con la función de activación “sigmoideal”. Su tasa de acierto con los datos de entrenamiento es del 90% y con los datos de test es del 90.67%.



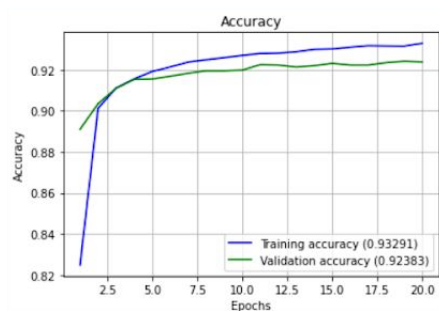
No se observa sobreajuste puesto que los resultados son prácticamente iguales para ambos grupos de datos.

- Se ha realizado otra prueba manteniendo el algoritmo “SGD” y cambiando a la función de activación “RELU”. Esta vez los resultados obtenidos han sido pésimos. Una tasa de acierto del 9.87% en datos de entrenamiento y 9.80% en datos de test.



Estos resultados se pueden deber a que esta función de activación está pensada para ser utilizada en capas ocultas para separar datos y no para ser utilizada en la capa de salida como ha sido el caso.

- Por último se ha realizado otra prueba, esta vez cambiando el algoritmo de entrenamiento por “Adam” y utilizando la función de activación “softmax”. Sorprendentemente se han obtenido mejores resultados que en las dos primeras pruebas. Concretamente, una tasa de acierto con datos de entrenamiento del 93.24% y con datos de test 92.53%. Sin embargo, si nos fijamos en la gráfica “Accuracy”, observamos que se produce sobreajuste puesto que los datos de test obtienen un resultado de éxito muy inferior al que se ha llegado a conseguir con los datos de entrenamiento.



Perceptrón multi-nivel con una capa oculta

Se ha añadido una capa oculta a la red neuronal y por lo tanto se espera obtener mejores resultados que con solo una capa de salida.

En las comprobaciones se ha utilizado el algoritmo de entrenamiento “SGD” con la función de activación “relu” para la capa oculta y la función de activación “softmax” para la capa de salida.

A continuación se presentan los resultados obtenidos al variar el número de neuronas en la capa oculta de la red:

Nº neuronas capa oculta	Tasa acierto entrenamiento	Tasa acierto test
10	91.24%	91.54%
15	91.78%	92.10%
20	92.25%	92.56%
50	92.86%	92.79%
100	93.52%	93.61%
300	93.85%	93.86%
500	93.92%	93.94%

Se observa que conforme aumenta el número de neuronas en la capa interna, mejora la tasa de acierto tanto de los datos de entrenamiento como la de los de test.

Cabe destacar que se ha probado este experimento con el algoritmo de aprendizaje “Adam” y se ha descubierto que genera mejores resultados que el algoritmo SGD. Sin embargo, el algoritmo “Adam” genera un sobreajuste muy notable. Por ejemplo, cuando se han configurado 50 neuronas en la capa interna, se ha obtenido una tasa de entrenamiento del 99.14% y una tasa de test de 97.51%. Para reducir el sobreajuste en este caso, se ha aplicado un dropout de 0,1.

Perceptrón multi-nivel con dos capas ocultas.

En este caso se ha añadido una nueva capa a la red neuronal.

Como en el caso anterior se ha probado el algoritmo SGD con la función de activación relu para las dos capas ocultas y la función de activación softmax para la capa de salida.

Se han observado los siguientes resultados:

Nº neuronas capa oculta 1	Nº neuronas capa oculta 2	Tasa acierto entrenamiento	Tasa acierto test
10	10	91.78%	91.98%
15	10	91.99%	91.87%
20	10	92.96%	92.66%
50	20	94.26%	94.36%
100	50	94.54%	94.37%
300	20	95.42%	95.10%
500	50	95.74%	95.41%

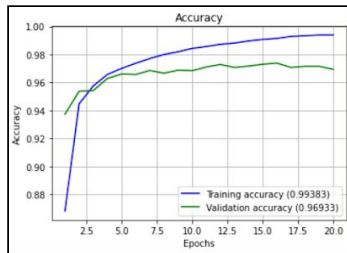
En ningún caso se ha observado sobreajuste. La mejor tasa de aciertos con los datos de test que se han obtenido con el algoritmo SGD es del 95.41%

Después, se ha utilizado el algoritmo Adam para probar distintas configuraciones de números de neuronas de las dos capas internas.

Se ha observado que en todos los casos probados, cuando el número de neuronas de una capa interna es superior a 10 se produce sobreajuste.

Por ejemplo, la siguiente configuración

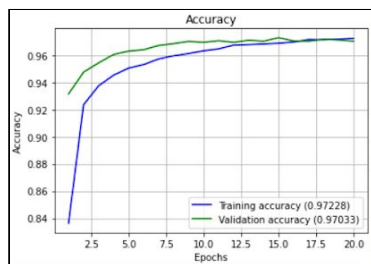
Nº neuronas capa oculta 1	Nº neuronas capa oculta 2	Tasa acierto entrenamiento	Tasa acierto test
50	50	99.26%	97.30%



Los datos de entrenamiento dan muy buenos resultados, pero los de test producen un 1.94% menos de aciertos.

Para reducir este sobreajuste se ha aplicado un dropout de 0.2.
Como consecuencia, se han obtenido los siguientes resultados:

Nº neuronas capa oculta 1	Nº neuronas capa oculta 2	Tasa acierto entrenamiento	Tasa acierto test
50	50	98.77%	97.34%



La tasa de éxito de los datos de entrenamiento se ha reducido casi un 1% y es más parecida a la tasa de éxito de los datos de test. Además en la gráfica se observa que ambos tipos de datos convergen hacia la misma tasa de éxito.
Por lo tanto se puede considerar que se ha eliminado el sobreajuste.

Red convolucional

Este tipo de redes se fundamentan en la aplicación de un conjunto de filtros sucesivos sobre la entrada.

En este apartado se ha ejecutado el fichero mnist_cnn.py que contiene un ejemplo de red convolucional. Se ha obtenido una precisión con los datos de test del 84.35%