SISTEMAS INTELIGENTES I

Bloque 2: Redes Neuronales

Guía de Prácticas

Javier J. Sánchez Medina
javier.sanchez@ulpgc.es

El propósito de esta guía de prácticas es dotar a los/as alumnos/as de Sistemas Inteligentes I, de los rudimentos prácticos en diseño y programación de Algoritmos Genéticos, Redes Neuronales y una introducción a la Minería de Datos

Esto complementa los contenidos teóricos que de esta asignatura serán impartidos.

La asignatura consta de 15 horas de prácticas de Aula y 15 horas de prácticas de Laboratorio. Se dividirán en los tres bloques arriba mencionados:

Bloque 1: Algoritmos Genéticos (6 semanas)
Bloque 2: Redes Neuronales Artificiales (6 semanas)
Bloque 3: Minería de Datos (2 semanas)

Las prácticas son individuales.

Práctica 4: Redes Neuronales. Perceptrón Multicapa. Backpropagation.

Introducción:

Las redes neuronales son ya una técnica clásica bio-inspirada con multitud de aplicaciones, generalmente centradas alrededor de la clasificación de patrones complejos de entrada. Existen generalmente dos aproximaciones fundamentalmente diferentes a las redes neuronales artificiales (ANN): las ANN supervisadas y no supervisadas (o auto-organizativas). La diferencia entre ambas radica fundamentalmente en si podemos aportar la respuesta "correcta" que esperamos de la red o no.

En esta práctica introductoria nos vamos a centrar en el primer tipo. Vamos a utilizar la quizás más utilizada red, un Perceptrón Multicapa con retropropagación.

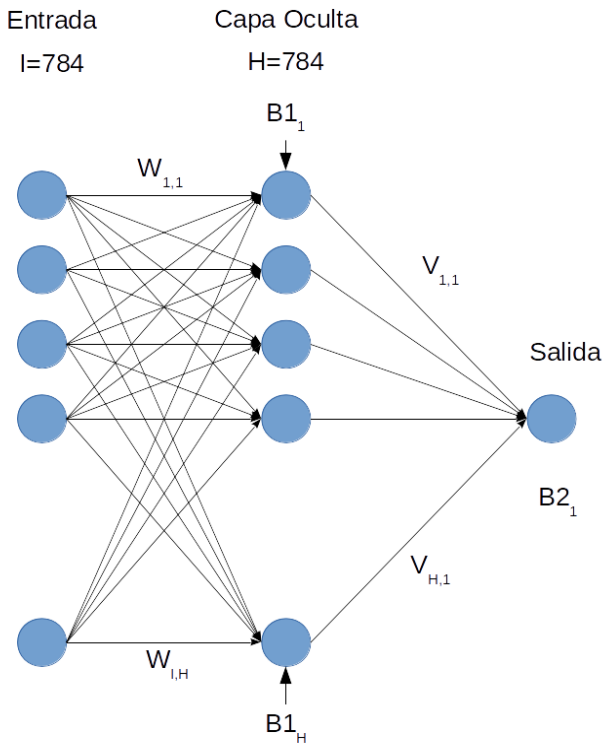
Problema a resolver:

Vamos a partir de la colección de imágenes de dígitos manuscritos del NIST (http://yann.lecun.com/exdb/mnist/). Esta colección consta de 60000 imágenes para entrenamiento y 20000 para test. Son imágenes en escala de grises (de 0 a 1), de 28x28 píxeles cada una. En el curso online se ponen las imágenes, y un script de Matlab para cargarlas en memoria.

La codificación de cada imágen es una ristra de 784 reales, colocando fila tras fila. Es decir, los primeros 28 números corresponden a la primera columna, los siguientes a las segunda y así hasta la columna 28 que acaba con el dígito número 784.

El objetivo de esta práctica es, partiendo de esas 784 entradas (fichero "images"), y de las salidas esperadas ("labels"), diseñar un perceptrón multicapa con retropropagación y entrenarlo con el conjunto de entrenamiento de la mejor forma posible para tener la tasa de acierto más alta al alimentar la red con los datos de test.

La estructura de la red es de libre elección para el alumno. Se recomienda empezar con una red sencilla como la siguiente:



Las ecuaciones que rigen el funcionamiento de la red son las siguientes:

Sea W la matriz(IxH) de pesos de las conexiones entre la capa de entrada y la capa oculta, sea X el vector de entradas y B1 el vector de sesgos de la capa oculta, la activación de la capa oculta será el vector A1 tal que así:

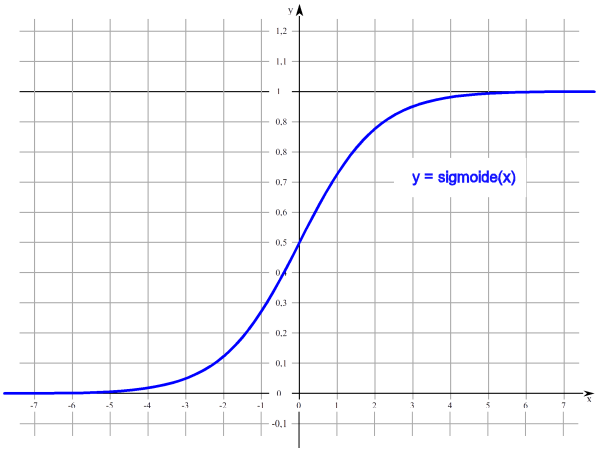
$$A1 = X \times W + B1$$

Nótese que estamos hablando del producto matricial.

Por lo general, en redes neuronales se suele utilizar una función de escalado de la señal de activación justo a continuación de cada neurona. La función más común es la sigmoide, cuya ecuación más genérica es la siguiente:

$$f(x) = \frac{1}{1 + e^{(-a(x-c))}}$$

El parámetro *c* está relacionado con el "centro" de la sigmoide y el parámetro *a* con su "pendiente". La sigmoidea para a=1 y c=0 sería la siguiente:



Se recomienda para la presente práctica estimar el valor medio de la activación de la capa oculta para centrar la sigmoidea y sacarle el máximo provecho al escalado que hace.

Una vez calculada la activación de la capa oculta y pasada por la sigmoide, se procede al cálculo de la señal en la capa de salida (en este caso que sólo son dos capas). Para ello, podemos utilizar la siguiente ecuación:

$$Out = A1_s \times V + B2$$

Siendo $A1_s$ la activación de la capa oculta una vez aplicada la sigmoide.

Por último es preciso escalar también la salida, y para ello podemos utilizar nuevamente la sigmoide.

Para el entrenamiento de la red neuronal, es decir, para el ajuste de los pesos de W, V, B1 y B2, de manera que clasifique correctamente las imágenes de entrada con la etiqueta que se corresponda, requerimos una rutina de actualización de los mismos, que en este caso se denomina retro-propagación por gradiente descendiente o la "Regla de Delta".

En pocas palabras, se compara la salida obtenida con la salida deseada y se calcula el error. Este error para calcular la “Delta” de la capa de salida, y luego, mediante propagación, para calcular la “Delta de la capa oculta.

La ecuación de la “Delta” de la capa de salida, para cada nodo *i* de salida es la siguiente:

$$\delta_{out}(i) = Out_s(i) \times (1 - Out_s(i)) \times (S(i) - Out_s(i))$$

- Out_s es el vector de salidas, una vez aplicada la sigmoide. En este caso, es un escalar, al haber un único nodo en la capa de salida.
- *S* es el vector de salidas esperadas (labels).

Para la capa oculta, para cada nodo *i* tenemos:

$$\delta_h(i) = A1_s(i) \times (1 - A1_s(i)) \times \sum_{j=1}^{O} \delta_o(j) * v_{i,j}$$

- *i* nodo de la capa oculta
- *j* nodo de la capa de salida.

Calculadas estas dos “deltas” la actualización de pesos es inmediata:

$$W' = W + \alpha \times X' \times \delta_h$$

$$B1' = B1 + \alpha \times \delta_h$$

$$V' = V + \alpha \times A1_s' \times \delta_{out}$$

$$B2' = B2 + \alpha \times \delta_{out}$$

- X'_s es la traspuesta de *X*, al igual que $A1_s'$ es la traspuesta de $A1_s$.

Para monitorizar el proceso de entrenamiento se suele utilizar el error mínimo cuadrático (MSE), cuya fórmula es la siguiente:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (S(i) - Out(i))^2$$

Siendo *S(i)* cada salida esperada y *Out(i)* la salida obtenida, ambas para el mísmo estímulo de entrada.

Este parámetro puede utilizarse para comparar la calidad de una arquitectura de ANN frente a otra, o para la condición de finalización del entrenamiento.

Objetivos de la práctica:

Los objetivos de esta práctica se pueden resumir como sigue:

1: Diseñar la red neuronal que permita la correcta clasificación de los dígitos manuscritos de entrada.
2: Entrenarla con el conjunto de datos provistos, lo mejor posible, reduciendo el MSE lo máximo posible.
3: Comprobar la calidad del entrenamiento calculando la tasa de acierto con el conjunto de muestras de test.

Entregables:

Memoria de menos de 10 páginas más un anexo con el código Matlab (u otros) con un anexo con el algoritmo utilizado para el entrenamiento y el testeo de la red neuronal.

La memoria debe incluir:

- Representación de la estructura de red neuronal utilizada y los parámetros que mejor resultado dieron (razón de aprendizaje, parámetros de la sigmoide, número de capas y nodos, etc.).
- Código fuente para tanto entrenamiento como test.
- Representación de la curva de aprendizaje (MSE)
- Representación de la tasa de acierto global y por dígito de la red entrenada