

Introducción al reconocimiento de patrones mediante redes neuronales

Emiliano Aldabas-Rubira

UPC-Campus Terrassa-DEE-EUETIT Colom, 1 08222 Terrassa Barcelona

e-mail: aldabas@eel.upc.es

Resumen

En el marco de las jornadas de conferencias de ingeniería electrónica JCEE'02, se presenta un breve resumen de una de las aplicaciones típicas de las redes neuronales artificiales ANN (Artificial Neural Network). En ella, un perceptrón multinivel MLP (Multilayer Perceptron) se usa para el reconocimiento óptico de caracteres OCR (Optical Character Recognition). Por último, se simula una red neuronal en el entorno MATLAB, entrenándola mediante el conocido algoritmo back propagation BP.

1. Introducción

En la bibliografía se pueden encontrar diferentes definiciones sobre lo que es una red neuronal, quizás la más completa es la siguiente:

Una red neuronal artificial es un procesador distribuido en paralelo de forma masiva que tiene una tendencia natural para almacenar conocimiento de forma experimental y lo hace disponible para su uso.

Se parece al cerebro humano en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso de aprendizaje.
- Los pesos sinápticos o fuerza con que están interconectadas las neuronas se utilizan para almacenar la información.

Otras definiciones son:

- Una nueva forma de computación, inspirada en modelos biológicos.
- Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.
- Un sistema de computación hecho por un gran número de elementos simples, elementos de proceso interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.
- Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.

Aunque cada definición aporta información sobre lo que es una red neuronal, para saber como funcionan y en que consisten es necesario desarrollar una explicación extensa. Sin embargo, en este trabajo se describirá una aplicación típica de las redes neuronales multicapa, concretamente el reconocimiento de patrones.

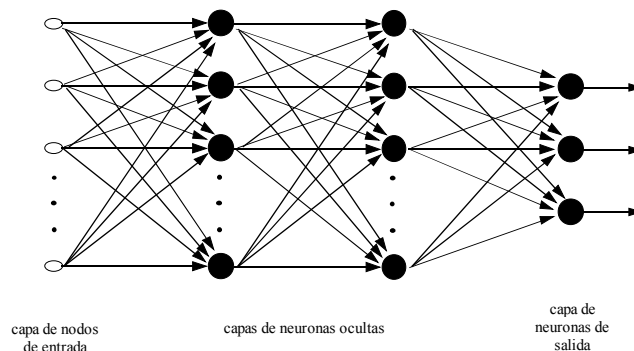


Fig. 1: Estructura de un perceptrón multinivel.

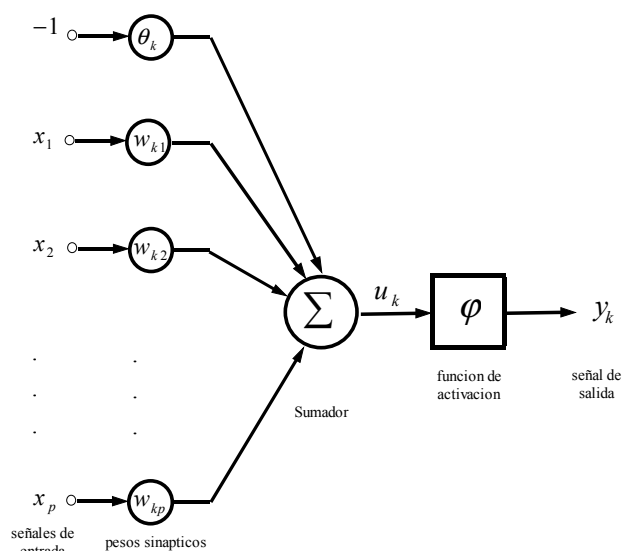


Fig. 2: Modelo de una neurona.

En este tipo de tarea hay un número fijo de categorías en las cuales las muestras de entrada deben clasificarse. Para ello primero se requiere una fase de entrenamiento en la que se presenta a la red los patrones que debe aprender y la categoría en cual clasificarlo. Entonces se le presenta a la red un patrón nuevo y desconocido pero que pertenece a alguna de las categorías aprendidas y esta debe decidir a que categoría se parece más.

La ventaja de usar redes neuronales está en el hecho que se pueden separar regiones no lineales de decisión tan complicadas como se desee dependiendo del número de neuronas y capas. Por lo tanto, las redes neuronales artificiales sirven para resolver problemas de clasificación de alta complejidad.

2. Perceptrón multinivel

Dentro de las redes neuronales, las que más utilizadas son las redes con múltiples capas que funcionan hacia delante. Esta red está compuesta por un conjunto de nodos de entrada que componen la capa de entrada, un conjunto de una o más capas ocultas de neuronas y una capa de neuronas de salida. La señal de entrada se propaga hacia adelante desde la capa de entrada por la oculta hasta la salida; este tipo de configuración se conoce como MLP o “MultiLayer Perceptrons” (figura 1) [1],[2],[3].

El hecho de que este tipo de red se aplique para resolver con éxito multitud de problemas se debe a la utilización del algoritmo de aprendizaje que actualmente está más extendido, el algoritmo o regla back propagation, el cual es una generalización de la regla LMS “Least Mean Square”, por lo tanto también se basa en la corrección del error.

Básicamente el proceso back propagation consiste en dos pasadas a través de las diferentes capas de la red, una pasada hacia adelante y una pasada hacia atrás. En la pasada hacia adelante, se aplica en la capa de entrada un patrón o vector de entrada, este propaga su efecto a través de las diferentes capas y como consecuencia produce un vector de salida. Durante este proceso, los pesos sinápticos de la red son fijos y no se modifican.

Durante la pasada hacia atrás en cambio, los pesos si se modifican de acuerdo con la regla de corrección del error. La señal de salida real se compara con la señal deseada y como resultado se obtiene una señal de error, que se propaga en dirección contraria a través de la red modificando los pesos, de forma que, al volver a pasar el vector de entrada hacia adelante, la respuesta obtenida se asemeje más a la salida deseada. Concretando, se puede decir que un perceptrón multicapa tiene tres características:

1. El modelo de cada neurona (figura 2) incluye una función no lineal. En este caso, a diferencia del perceptrón donde es la función escalón, y debido a la necesidad de que sea una función continua y derivable, es la función sigmoide, donde u_k es la suma total de la actividad interna en la neurona k (la señal de entrada) e y_k la salida que se produce en la neurona.

$$y_k = \frac{1}{1 + \exp(-u_k)} \quad (1)$$

2. La red contiene una o más capas ocultas de neuronas que no forman parte ni de la entrada ni de la salida. Estas neuronas ocultas capacitan a la red para aprender progresivamente cualquier correspondencia entre la entrada y la salida y almacenar internamente esta información.

3. La red posee un gran número de conexiones, estas vienen determinadas por los pesos de la red. Un cambio en la conexión entre las neuronas equivale a un cambio en los pesos.

La combinación de estas características, hace que la habilidad de esta red para aprender a partir del entrenamiento sea muy potente, por ejemplo es capaz de resolver el problema de la OR-exclusiva a diferencia del perceptrón.

De todas formas, este comportamiento hace que sea difícil conocer a priori la respuesta de la red. Esto se debe a dos motivos, el comportamiento no lineal de las neuronas, las cuales están muy interconectadas, (lo que hace difícil un análisis teórico de la red) y la existencia de neuronas ocultas, que impide poder “ver” como se produce el aprendizaje y determinar cuales son las características que mejorarían el aprendizaje.

El desarrollo del algoritmo back propagation proporciona un método eficiente para entrenar este tipo de redes. Aunque no es capaz de resolver todos los problemas, se ha demostrado como el mejor de todos. Su importancia está en su capacidad de autoadaptar los pesos de las neuronas intermedias para aprender la relación que existe entre el conjunto de vectores o patrones de entrada y su correspondiente salida, y poder aplicar esa relación después del entrenamiento a nuevos vectores de entrada imperfectos o con ruido. Esta capacidad se conoce como generalización. La red debe encontrar una representación interna que le permita generar las salidas deseadas durante la etapa de entrenamiento, y posteriormente durante el funcionamiento ser capaz de generar salidas para entradas que no le fueron mostradas durante el aprendizaje pero que se asemejan a alguna de las que si le fueron mostradas.

3. Ejemplo de aplicación

Para simular el funcionamiento de un perceptrón multinivel entrenado mediante el algoritmo back propagation, se plantea un sencillo problema de reconocimiento de óptico de caracteres. Su descripción es la siguiente:

Dado un panel de entrada compuesto por una matriz de 7x5 puntos, se consideran 12 clases diferentes donde se pretenden clasificar las muestras que se introducen. Los patrones que definen correctamente a cada una de las clases son los números del 0 al 9, el punto y el guión (figura 3). Cuando a la entrada se presente una muestra distinta de los patrones correctos, el sistema presentará a su salida la información decodificada de la clase a la que pertenece la muestra, o bien, de la clase a la cual se aproxima más.

En base a este planteamiento, la red neuronal dispone de 35 entradas que se corresponden con los puntos de la matriz numerados en la figura 4. El valor de cada entrada puede ser 0 si el punto es blanco y 1 si el punto es negro. Por otro lado, dispone de 12 salidas, una por cada clase. Cuando se introduzca una muestra a la entrada únicamente se activará la salida de la clase a la que pertenezca, permaneciendo las 11 restantes desactivadas con valores próximos a cero. Se considera que una salida está activada cuando su valor es próximo a la unidad.

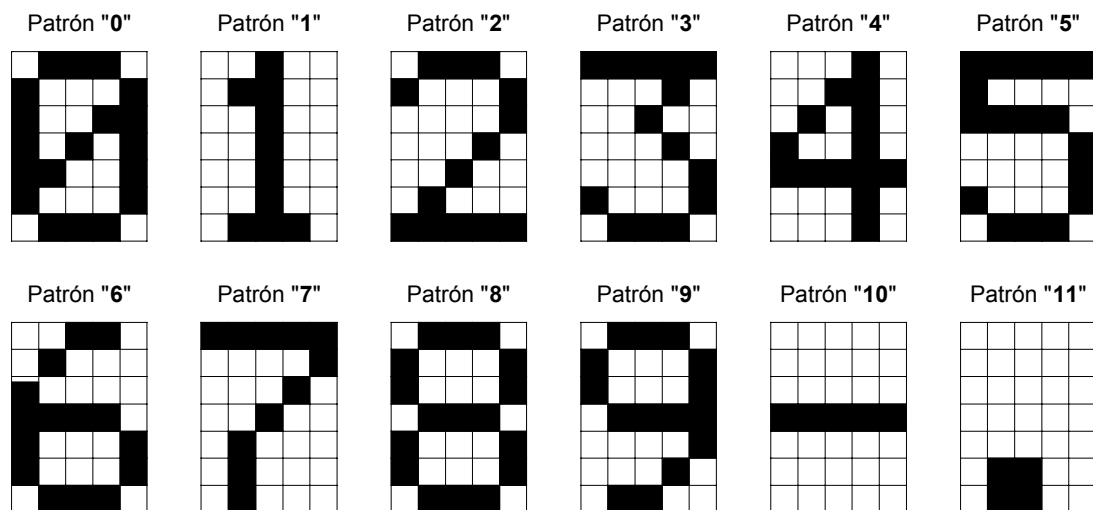


Fig. 3: Representación gráfica de los patrones de los caracteres numéricos en el formato de matriz de puntos 7x5.

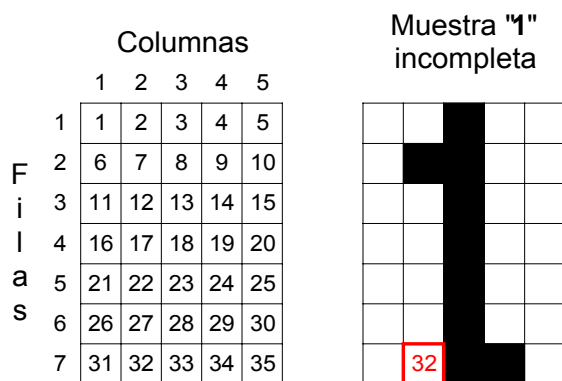


Fig. 4: Numeración de los puntos del display 7x5 y ejemplo de una muestra del patrón "1" con un punto erróneo.

Salida ideal.	Patrón "1" sin error.	Patrón "1" con error en el punto 32.
0	0.00000000038246	0.00000000022760
1	0.97094634494005	0.93598496351919
0	0.00483116118854	0.00425679820665
0	0.00000007267857	0.00000003653425
0	0.00001815880785	0.00001549628867
0	0.00000000010405	0.00000000001916
0	0.00000098764700	0.00000340726166
0	0.00038787588951	0.00012976360904
0	0.00016876462031	0.00008967151863
0	0.00127029941580	0.00051591379971
0	0.02144501839329	0.01084408021869
0	0.01863962414026	0.06933193518770

Tabla I: Resultados de simulación.

4. Resultados de simulación

Se ha desarrollado el programa *BP5.m* para el entorno MATLAB 5.0 Student Edition. En él se ha programado un perceptrón multinivel con 35 entradas y 12 salidas. También dispone de dos capas ocultas a las cuales se les puede modificar el número de sus neuronas. La red neuronal se ha entrenado con el algoritmo back propagation fijando el valor del momento en 0.8 y el factor de aprendizaje en 0.2. En este proceso únicamente se han usado doce muestras diferentes, es decir, los doce patrones sin ningún punto erróneo.

En la tabla I se muestran los resultados obtenidos para una red neuronal de tamaño 35-30-20-12. Se aprecia que tras el proceso de entrenamiento, el sistema responde de forma casi ideal cuando se introduce un patrón sin error.

De la misma manera, cuando se introduce una muestra con error en un punto (figura 4), la red clasifica perfectamente dicha muestra en la clase correcta. Este sencillo ejemplo sirve para confirmar que los perceptrones multinivel resuelven excelentemente el problema de clasificación de muestras, compitiendo con otros métodos como puedan ser los estadísticos.

Referencias

- [1] J. R. Hiler; V. J. Martínez; "Redes neuronales artificiales. Fundamentos, modelos y aplicaciones". Ed. Rama, 1995.
- [2] S. Haykin; "Neural networks. A comprehensive foundation". IEEE Press, 1994.
- [3] Joan Cabestany Moncusí, Sergi Bermejo Sánchez; "Xarxes Neuronals". <http://petrus.upc.es/~microele/neuronal/xn/>