

Ejercicios del algoritmo de retropropagación del error en redes neuronales

Ejercicio 1.

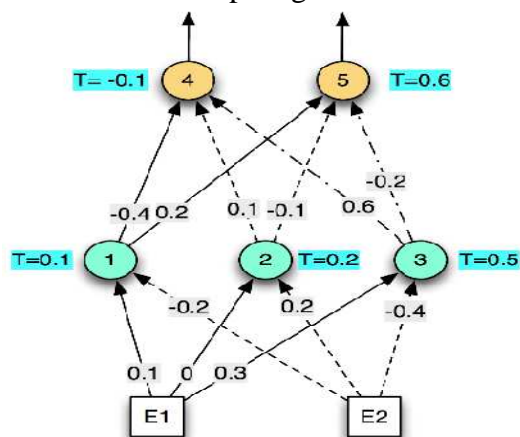
Disponemos de un perceptron simple con pesos $w_1=w_2=0,5$, $w_0=-0,5$ y un conjunto de entrenamiento $D = \{ \langle (1,0), 1 \rangle, \langle (0,1), 0,5 \rangle \}$, donde utilizamos la función sigmoide como activación. Obtener la variación que se produce en el error cuadrático medio cometido sobre el conjunto de entrenamiento D tras un paso del algoritmo de descenso por gradiente (factor de aprendizaje $\eta = 0,8$, versión offline).

Ejercicio 2.

Diseñe un perceptron simple con 3 valores de entrada y función de activación de salto que sirva para calcular la mayoría simple, es decir, que reciba 3 entradas binarias y devuelva un 1 si hay mas unos que ceros y un -1 en caso contrario (empate, o más ceros que unos).

Ejercicio 3.

Disponemos de una red neuronal como la que aparece en la figura, donde T es el umbral de cada neurona (es decir, el valor a partir del cual empieza a activarse) y todas las neuronas son de tipo sigmoide.



Alimentamos la red utilizando el patrón $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}=[E1=0,6, E2=0,1], \mathbf{y}=[S4=0, S5=1])$. Aplique la propagación hacia atrás del error, indicando como cambiarían los pesos, utilizando para ello una tasa de aprendizaje de 1.

Ejercicio 4.

Demostrar geoméricamente que un perceptron simple no puede calcular la función lógica XOR. Construir una red neuronal (usando la función de activación de salto) que sea capaz de obtenerla.

Ejercicio 5.

Imaginemos que queremos diseñar una red neuronal para reconocer dígitos manuscritos. Proponga un posible diseño para dicha red neuronal, incluyendo su arquitectura y funciones de activación y detallando como organizaría el proceso de recolección y preprocesamiento de los datos ¿Que algoritmo podríamos utilizar para entrenar la red?.

Ejercicio 6.

Aplique el algoritmo de retropropagación del error a un perceptron simple hasta obtener la primera actualización del peso w_2 con $\eta = 0,1$, usando activación sigmoide y el siguiente conjunto de entrenamiento, y pesos iniciales $w_0 = -0,1$ y $w_1=w_2=w_3=0,1$,

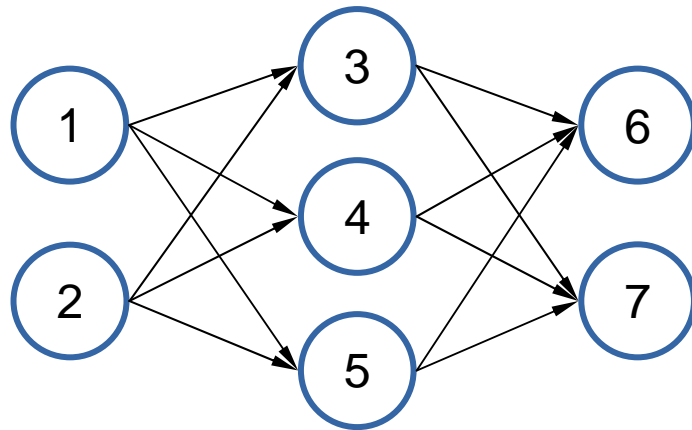
Patrón	x_1	x_2	x_3	y
p_1	0,7	0,2	0,1	0,3
p_2	0,3	0,5	0,2	0,8
p_3	0,1	0,1	0,8	0,6

Ejercicio 7

Diseñar un perceptron simple con 2 valores de entrada y función de activación de salto que sirva para calcular la mayoría simple, es decir, que reciba 2 entradas binarias y devuelva un 1 si hay mas unos que ceros y un -1 en caso contrario (empate, o más ceros que unos).

Ejercicio 8.

Supongamos la siguiente red neuronal:



Todas las neuronas tienen sesgo y función de activación sigmoide. El valor inicial de todos los pesos (incluidos los sesgos) es 1. Las neuronas 1 y 2 son las neuronas de entradas, mientras que las neuronas 6 y 7 son las neuronas de salida.

- Se pide aplicar el algoritmo de retropropagación del error para actualizar los tres pesos de la neurona 4, considerando los siguientes dos patrones: $\mathbf{p}_1 = \langle (1,0), (0,1) \rangle$ y $\mathbf{p}_2 = \langle (0,1), (1,0) \rangle$, donde $\langle (a,b), (c,d) \rangle$ quiere decir que el valor de entrada para la neurona 1 es a , el valor de entrada para la neurona 2 es b , el valor deseado para la neurona 6 es c y el valor deseado para la neurona 7 es d . Considerar un algoritmo *offline* y un factor de aprendizaje de 0,8.
- ¿Qué cálculos podría aprovechar si se pidiese lo mismo para la neurona 7?