Sesión 9-Tema 7: Redes Neuronales

Tema 7: Redes Neuronales – Resumen

Las redes neuronales artificiales (RNA), están basadas en redes biológicas, compuestas por muchas capas de neuronas que se comunican entre sí. Cada neurona de una capa está conectada con todas las neuronas de la siguiente capa, las neuronas aprenden de forma parecida a cómo lo hace Adaboost, construyendo clasificadores fuertes a partir de clasificadores débiles.

Se basan en algo que llamamos memoria asociativa, es decir, asignar a una imagen una etiqueta, de modo que, una vez que la RNA a clasificado esa imagen, podamos comparar el resultado con la etiqueta dada y comprobar si se ha clasificado correctamente.

Cómo están basadas en redes biológicas, su función de activación trabaja analógicamente, igual que una neurona biológica, la cual se activa según el cambio del nivel de PH que tenga la neurona.

Existen 2 tipos de neuronas, el perceptrón, con una función de activación escalonada, es decir, 0 o 1, y la Neurona sigmoidea, la cual tiene decimales (tiene pendiente la función de activación).

Perceptrón:

Tiene una serie de valores de entrada y produce una salida, estos valores de entrada tienen un peso asignado, de modo que ponderamos esos valores (x1 * w1 + x2 * w2 …+ xn * wn), donde W es el groso de conexión entre 2 neuronas.

Figura 1: Perceptrón simple

Primero debemos entrenar la red, obtener los valores W para cada entrada y el umbral de la red, la salida será 0 o 1 en función de si el sumatorio X1 * w1 + -> + xn * wn es estrictamente mayor que el umbral (salida = 1) o lo contrario (salida = 0).

Si representamos las entradas y pesos como tuplas, podemos representar el sumatorio como el producto escalar de x * w, además con el cálculo del *bias(b)* (facilidad de dispararse el perceptrón) b = -umbral, podemos obtener 1 si w * x + b > 0 o obtener 0 en el caso contrario.

El perceptrón artificial define una recta la cual crea un hiperplano que se divide en 2 semiplanos, el lado positivo (1) y el negativo (0), en este estarán todos los puntos del plano, vector normal al plano de cada punto y lo svalores de entrada. El único problema de esto es que no se pueden separar los datos, para ello debemos combinar varios perceptrones.

Perceptrón multicapa (MLP):

Aumentamos el número de capas del perceptrón, dividiendo la arquitectura en 3 partes, Capa de entrada, Capas ocultas (intermedias) y Capa de salida, la salida de una capa será la entrada de la siguiente y así hasta la salida. El aumento del nº de capas permite tomar decisiones más complejas. Existen 2 tipos de redes:

- Shallow networks: 1 capa intermedia

- Deep Networks: 2 o más capas (suelen ser de 5 o 10 capas). Se entrenan mediante 'búsqueda por gradiente' (buscar el mínimo), o 'back propagation' (ajustar pesos para obtener el valor que me interesa). Si nos pasamos con el nº de capas puede ser una red inestable.

Figura 2: Perceptrón multicapa

El entrenamiento de la red se basa en observar ejemplos (un gran nº de ellos), para ello tenemos 2 conjuntos. Uno de entrenamiento, con el que calculamos los pesos y umbrales, y otro de pruebas o test, donde observaremos el error que tiene frente a ejemplos nuevos.

Si da error, debemos cambiar los pesos y biases de las capas intermedias, pero al tener una función activación escalonada, el cambio tarda mucho en hacer efecto, ahí entra la neurona sigmoidea, donde las pequeñas modificaciones cambian la pendiente de esta.

El cálculo de la activación de basa en la siguiente función: a' = sigma(Wa + b), donde a' es la salida de una capa y la entrada a la siguiente. Recordad que Wa es el producto escalar de los valores de entrada con los pesos de la entrada.

Podemos definir una función de coste que evalúe el error cuadrático medio:

Figura 3: Función de coste

Donde y(x) es el valor esperado, y a es la salida, nuestro objetivo es minimizar

el error que esto produce, ya que la función es derivable, para ello utilizamos el descenso por gradiente actualizando los pesos y biases.

Figura 4: Minimización de coste

Calcular la activación de una neurona en función de las activaciones de la capa anterior:

Figura 5: Activación de una neurona

Podemos expresar la función de coste a optimizar a partir de la activación de la última capa *L:*

Figura 6: Función de coste a partir de la última capa

Podemos expresar el coste como la suma del coste individual para cada entrada *x*

Figura 7: Coste individual para cada entrada

Perceptrón multicapa

Índice WIki individual