Funciones de activación, retropropagación y descenso del gradiente

Ahora vamos a ver en detalle el funcionamiento del perceptrón introduciendo conceptos con los que se encontrará cuando utilice las redes neuronales para realizar sus proyectos de clasificación.

1. Función de activación

Como hemos visto durante la presentación de la neurona biológica, su activación se produce cuando se sobrepasa un umbral de tensión eléctrica. En el caso de nuestra neurona artificial, su activación también se desencadenará en función de un umbral.

La primera etapa en la determinación de la activación de una neurona artificial es hacer la suma de los productos de los pesos sinápticos con los valores de entrada. Esta etapa se puede considerar como una preactivación de la neurona. Una vez hecha la suma, vamos a utilizar una función de activación que nos permitirá determinar si debe activarse la neurona. Esta activación será de hecho la predicción de nuestra neurona. Para calcular esta predicción, podemos utilizar varias funciones.

**a. Función de umbral binario**

La primera función que puede sernos útil es la función de umbral binario. Esta función devuelve un valor igual a 0 o 1. Tomamos como hipótesis que la suma ponderada de los distintos valores de cada entrada es 0,8 y que hemos definido un umbral a 0,5. Como 0,8 es mayor que 0,5, la función reenvía el valor 1 (la neurona se activa). Si el valor de esta suma ponderada fuera inferior a 0,5, la función de activación reenviaría 0.

Aquí podemos ver el ejemplo con números:

| **Valor en entrada** | **Peso vinculado a la entrada** | **Valor de la entrada \* valor del peso** |
| --- | --- | --- |
| 2 | 0,2 | 0,4 |
| 1 | 0,1 | 0,1 |
| 3 | 0,1 | 0,3 |

La suma ponderada es 0,8 (0,4 + 0,1 + 0,3), lo que provoca la activación de la neurona.

Gráfico

Descripción generada automáticamente

*Función de umbral binario*

**b. Función sigmoide**

Como hemos visto en los distintos ejemplos de los capítulos anteriores, la predicción que obtenemos rara vez es igual a 0 o a 1. Solemos obtener valores numéricos comprendidos entre 0 y 1 (0,50, 0,99…) que expresan un porcentaje de probabilidad. Por lo tanto, la función de umbral binario no puede responder a nuestras necesidades. Por eso vamos a utilizar otra función, llamada sigmoide, con cambios de valores entre 0 y 1 más progresivos.

He aquí la fórmula matemática de esta función que da lugar a la curva en S ilustrada por la figura que aparece a continuación.



Gráfico, Histograma

Descripción generada automáticamente

*Función sigmoide o curva en S*

El inconveniente de esta función es que valores negativos pueden dar lugar a valores positivos.

Si por ejemplo x = -0,2, entonces el cálculo de la función sigmoide de x tendrá el valor 0,45.



La función sigmoide se utilizada sobre todo en los casos de predicciones de probabilidades.

**c. Función tangente hiperbólica (tanH)**

La función de tangente hiperbólica se parece un poco a la función sigmoide, pero tiene la ventaja de que para todo valor negativo permanece o se convierte en muy negativa.

Diagrama

Descripción generada automáticamente

La función tangente hiperbólica se utiliza mucho en casos como este, de clasificación entre dos clases.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Función tangente hiperbólica*

**d. Función ReLU (Rectified Linear Unit, unidad de rectificación lineal)**

Aunque las funciones sigmoide y tangente hiperbólica se utilizan como función de activación, tienen un problema relacionado con lo que se denomina la saturación. En efecto, los valores elevados permanecen en el intervalo [0, 1] y los más pequeños en el intervalo [-1 y 0]. Una vez alcanzada la saturación, el aprendizaje de la red se vuelve complejo.

Como veremos a continuación el aprendizaje se hace ajustando los distintos pesos de la red neuronal en función del error que ha cometido durante su clasificación. En el caso de una saturación, la actualización de estos pesos se vuelve difícil y repercute en el aprendizaje.

La función ReLU palía este problema mediante un funcionamiento bastante sencillo. Si el valor procedente de la suma ponderada es inferior a 0, la función toma el valor 0. En caso contrario, toma el valor resultante de la suma.

Una de las ventajas es su bajo coste en términos de cálculo porque se trata de tomar el máximo entre el valor 0 y el valor de la suma ponderada. Su segunda ventaja es que no padece la saturación, porque no tiene valor límite en la zona positiva. Sin embargo, cuando esta función se enfrenta a un valor negativo, se le atribuye automáticamente el valor 0, lo que además implica un no aprendizaje de la red.

La función Leakly ReLU aporta la solución para paliar este problema, multiplicando el valor negativo por 0,01 para permitir desencadenar un aprendizaje de la red.

A pesar de su inconveniente, la función ReLU es la más utilizada en la actualidad como función de activación en una red neuronal.

**e. Función softMax**

El resultado de la función sigmoide (también llamada regresión logística) es una probabilidad de pertenencia a un grupo dado. Si tenemos que clasificar una observación según dos grupos distintos, A y B, y obtenemos el valor 0,7 para el grupo A mediante la función sigmoide, eso significa que la observación pertenece al grupo A con una probabilidad del 70 %, y al grupo B, con una probabilidad del 30 %.

En este ejemplo hemos definido dos grupos A y B (también llamados clases) cuya suma de probabilidades es igual a 1 (0,7 + 0,3), pero es posible que tengamos que clasificar la observación entre muchas clases. Entonces interviene el algoritmo SoftMax, que atribuye una probabilidad a cada una de las distintas clases mientras garantiza que la suma de estas probabilidades sea igual a 1.

| **Clase** | **Probabilidad** |
| --- | --- |
| Animal | 0,01 |
| Fruta | 0,95 |
| Vehículo | 0,04 |

Este tipo de función de activación se utiliza generalmente en una red neuronal multicapa y en el caso de clasificaciones con clases múltiples.