

REDES NEURONALES Y LÓGICA DIFUSA



Material Funciones de activación

DOCENTE: ALDO ALARCÓN SUCASACA
2021



Señor estudiante: Por favor, analizar el siguiente material.

Funciones

1. Soporte de sesión teórica

1.1.- Concepto de Capas

Las redes neuronales se encuentran conformadas por capas que a su vez cuentan con “neuronas” quienes se conectarán entre sí para enviarse mensajes y poder compartir información que será relevante para determinar el o los outputs correspondientes.

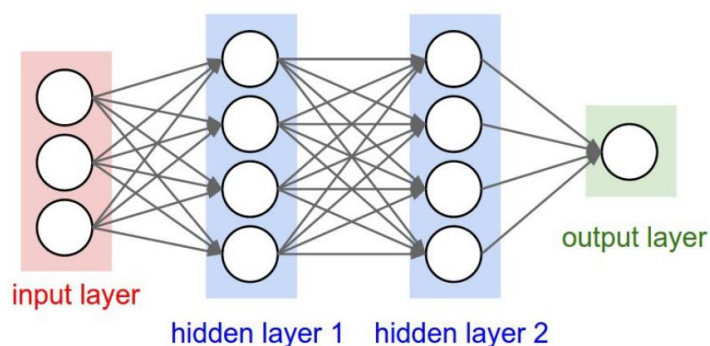
Las capas se encuentran divididas en grupos:

- Input,
- Hidden
- Output.

Input: estarán todas nuestras entradas o datos de entrenamiento estos contarán con pesos que permitirán expresar su importancia.

Hidden: puede estar conformada a su vez por una o varias capas el número de capas dependerá de qué tan sofisticado queremos nuestro modelo, sin embargo, es necesario recalcar que mientras más capas se tengan necesitaremos más recursos como tiempo y poder computacional.

Output: puede contar con una o varias neuronas dependerá del número de características que se desean llegar a encontrar.

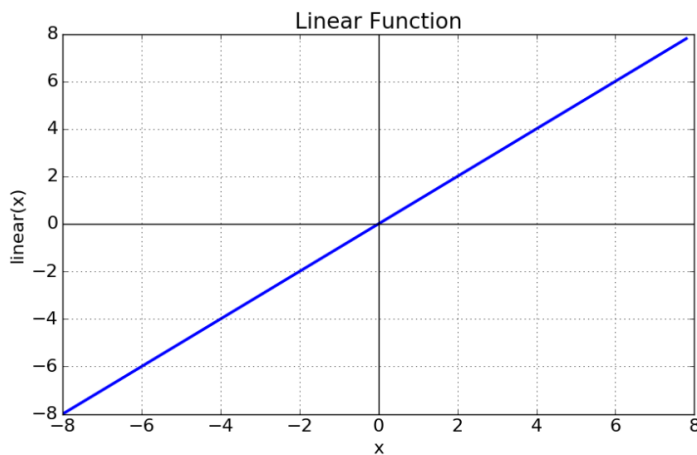


2.- Funciones de activación

La función de activación devuelve una salida que será generada por la neurona dada una entrada o conjunto de entradas. Cada una de las capas que conforman la red neuronal tienen una función de activación que permitirá reconstruir o predecir. Además, se debe considerar que en la red neuronal se usará una función no lineal debido a que le permite al modelo adaptarse para trabajar con la mayor cantidad de datos.

Las funciones de activación se dividen en dos tipos como: lineal y no lineal

2.1.- Función Lineal

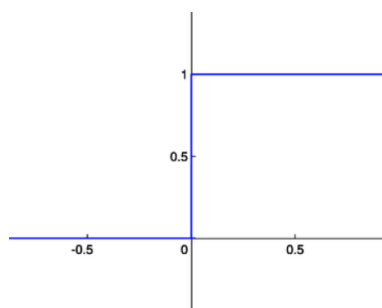


Esta función también conocida como identidad, permite que lo de la entrada sea igual a la salida por lo que si tengo una red neuronal de varias capas y aplicó función lineal se dice que es una regresión lineal. Por lo tanto, esta función de activación lineal se usa si a la salida se requiere una regresión lineal y de esta manera a la red neuronal que se le aplica la función va a generar un valor único. Por ejemplo, se usa cuando se solicita predecir el valor de un número de ventas.

2.2.- Funciones no lineales

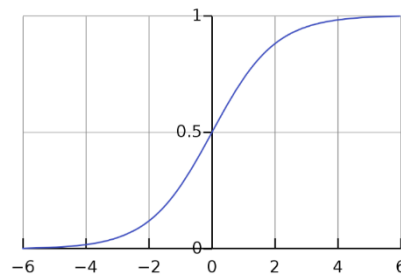
Función escalón

Esta función también conocida como escalón, indica que, si la x es menor que cero la neurona va a ser cero, pero cuando es mayor igual a cero dará como salida igual 1. Esta función se usa cuando se quiere clasificar o cuando se tiene salidas categóricas. Por ejemplo, se puede usar para predecir si compro algo o no.



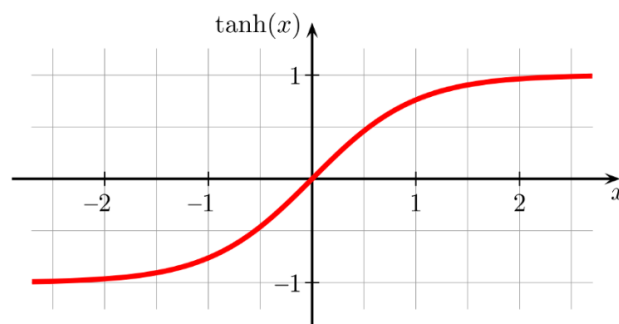
Función Sigmoide

Esta función también conocida como función logística, está en un rango de valores de salida entre cero y uno por lo que la salida es interpretada como una probabilidad. Si se evalúa la función con valores de entrada muy negativos, es decir $x < 0$ la función será igual a cero, si se evalúa en cero la función dará 0.5 y en valores altos su valor es aproximadamente a 1. Por lo que esta función se usa en la última capa y se usa para clasificar datos en dos categorías.



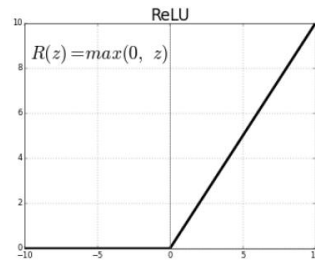
Función Tangente Hiperbólica

Esta función de activación llamada tangente hiperbólica tiene un rango de valores de salida entre -1 y 1. Se dice que esta función es un escalamiento de la función logística, por lo que a pesar que esta función está centrada tiene un problema similar a la sigmoide debido al problema de desaparición del gradiente, que se da cuando en el entrenamiento se genera un error con el algoritmo de propagación hacia atrás y debido a esto el error se va propagando entre las capas, por lo que en cada iteración toma un valor pequeño y la red no puede obtener un buen aprendizaje.



Función RELU

Esta función es la más utilizada debido a que permite el aprendizaje muy rápido en las redes neuronales. Si a esta función se le da valores de entrada muy negativos el resultado es cero, pero si se le da valores positivos queda igual y además el gradiente de esta función será cero en el segundo cuadrante y uno en el primer cuadrante.



Entrenamiento y pérdida

El objetivo de una red neuronal es aprender comportamientos que permitan destacar características de los objetos que se están tomando como input, todo este proceso es llamado Entrenamiento, sin embargo, es casi imposible obtener un cien por ciento de predicciones correctas, por lo que a la penalidad por una predicción incorrecta se la llama Pérdida. El que exista pérdida en nuestro modelo no es malo, ya que en caso de no tenerla es muy posible que se esté realizando overfitting, es decir que nuestro modelo es incapaz de generalizar. El objetivo del entrenamiento es medir la pérdida para luego ir iterando hasta que el algoritmo descubra los parámetros del modelo con la pérdida más baja posible una vez que esto ocurra el modelo converge.