Machine learning y Deep learning con Python: Lectura 8

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

Aprendizaje profundo

El aprendizaje profundo o Deep Learning en inglés, es una categoría del Machine learning que trabaja con un tipo de algoritmos denominado redes neuronlaes.

Esta categoría utiliza una representación que simula lo que hace el cerebro humano y lo adapta para crear algoritmos de Machine learning.

Existen varios ejemplos de tareas que pueden ser realizadas por el cerebro humano, como reconocer objetos y personas, entender la música que escucha, comunicarse, etc.

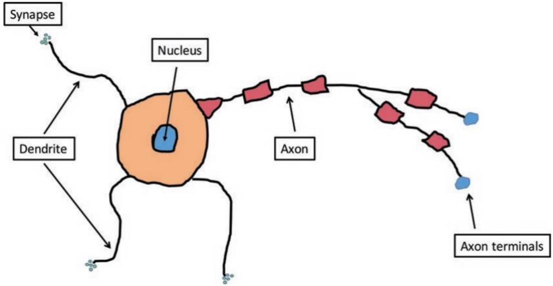
Actualmente es usado para detección de imágenes por foto o video, predecir textos, aplicaciones de mejora de imágen, sistemas de recomendaciones, etc.

Aprendizaje profundo

La historia se remonta a 1943, cuando Walter Pitts y Warren McCulloch crearon un modelo matemático para una neurona humana.

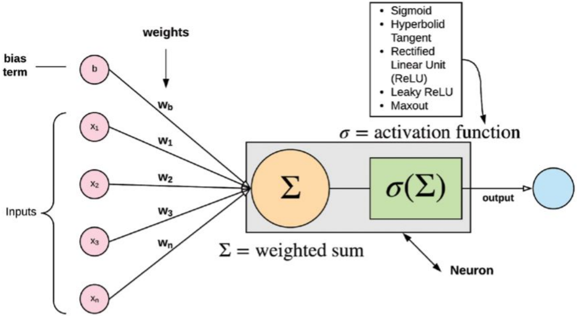
Los componentes principales para esta representación matemática son:

- Axón
- Dentrita
- Sinapsis



Perceptrón

Analizamos la función del perceptrón, cuando estudiamos la regresión logística. En una forma de representación gráfica, se ve de esta manera:



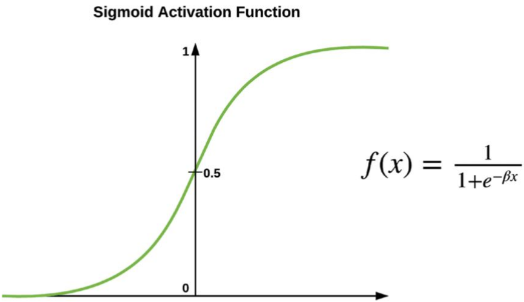
Donde el bias, es el intercepto, y $\sigma eslafunci\acute{o}ndeactivaci\acute{o}n$.

Sigmoide

Existen algunas funciones de activación como son: La Sigmoide, que es una función no lineal entre 0 y 1. Esto hace que se represente una brecha larga cuyo umbral es 0.5

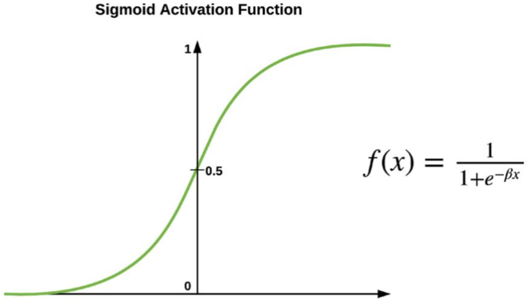
Sigmoide

Existen algunas funciones de activación como son: La Sigmoide, que es una función no lineal entre 0 y 1. Esto hace que se represente una brecha larga cuyo umbral es 0.5



Sigmoide

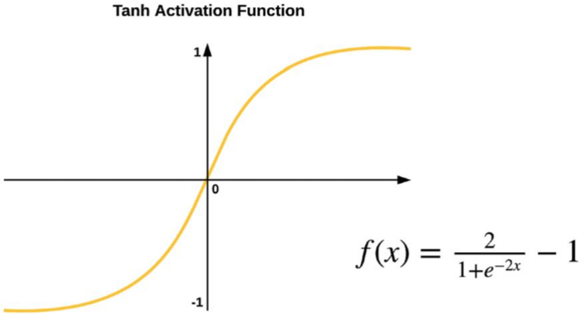
Existen algunas funciones de activación como son: La Sigmoide, que es una función no lineal entre 0 y 1. Esto hace que se represente una brecha larga cuyo umbral es 0.5



Sin embargo esta función de activación sufre con el efecto del desvanecimiento del gradiente, que es un problema que se sufre en la optimización del algoritmo. Otra desventaja de la función sigmoide es que no es centrada en 0. Esto tiene un efecto negativo al minimizar la función de costo.

Tangente hiperbólica

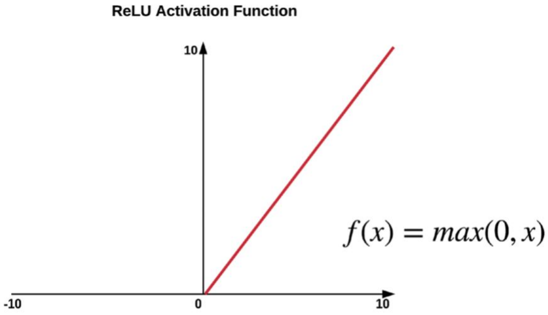
Para mejorar la función de activación sigmoide, se planteó la función de activación tangente hiperbólica. Esta función es centrada en 0 y el rango va de -1 a 1.



Sin embargo el problema del desvanecimiento del gradiente aún no se soluciona, ya que sigue siendo de caracter no lineal la función de activación.

Relu

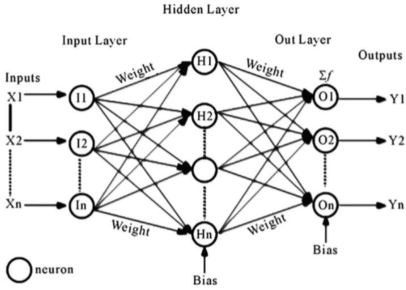
La función de activación relu, toma valores de 0 y una pendiente positiva de 1 para valores x, mayores a 0.



Redes neuronales

Cuando se coloca en un mismo algoritmo, una conexión de varios perceptrones, se realiza lo que es conocido como redes neuronales.

Una red neuronal, es un conjunto de capas conectadas mediante pesos de neuronas que al final logran obtener un resultado binario o de varias clases.
El método que utiliza las redes neuronales para lograr entrenarse se conoce como método de propagación inversa aka **backpropagation**.



Desvanecimiento del gradiente

El algoritmo de propagación inversa, optimiza la función de costo a través de las derivadas parciales de los pesos de las capas. Lo que hace una cadena de optimización, donde:

$$J = \frac{1}{n} \sum (y - \hat{y})^2$$

$$W_{n,k} = W_{n,k-1} - \alpha \frac{\partial(J)}{\partial(W)}$$

Por lo que cuando la derivada sea muy baja, las actualizaciones serán cortas, lo que no dará una aproximación correcta y el entrenamiento no será el óptimo

La función softmax

Para cuando existen varias clases, la función que se usa al final de la red, es la función softmax, que calcula la propabilidad que esa clase sea correcta, respecto a la data de entrada.

$$\sigma(k) = \frac{e^{\hat{y}(k_i)}}{\sum(e)^{\hat{y}(k_i))}}$$