La arquitectura de la red puede ser dividida en tres partes:

* La capa de entrada en donde los datos son introducidos.
* La cada de salida que hace una predicción
* Las capas ocultas, que se encuentran entre la capa de salida y la capa de entrada. Las capas ocultas son quienes hacen las operaciones matemáticas.

\_\_  
Dentro de la arquitectura de la red neuronal ocurren muchas operaciones de producto punto entre las entradas de cada perceptron con sus respectivos pesos.

Estas operaciones son lineales.

Las funciones de activación son la solución al colapso de las linealidades de las capas de la red neuronal.

https://www.wolframalpha.com/

Escriba la function de activacion y le dira como debe ser, por ejemplo escribe **sigmoid(x)**

FUNCIONES DE ACTIVACION MAS USADAS

|  |  |
| --- | --- |
| Escalonada, 0, 1 | Signo / signum  Igual a escalonada solo que de -1 , 1 , para abrir a datos negativos  #si x es menor a 0 devuelva 0  #si x es mayor a 0 devuelve 1  def step(x)  reurn np.piecewise(x,[x<0.0, x>0.0], [0,1]) |
| Sigmoidan    De 0 a 1 con valore continuos, buena para probabilidad | import numpy as np  import matplotlib.pyplot as plt  #sigmoid  def sigmoid(x)  return 1/(1+np.exp(-x)  x= np.linspace(-10,10,100)  plit.plot(x, sigmoid(x)) |
| Similar pero de -1, 1 la tangente hiberbolica o tanh  Continua, -1, 1 para probabilidades | # para tanh use la funcion tanh que trae numpy :')  np.tanh(x) |
| Funcion Lineal ReLu | La mas usada para las capas ocultas, si el valor es menor a 0, le pone cero, y si no el valor que viene, hasta el infinito. Enriquece el valor.  # relu  def relu(x):  return np.piecewise(x, [x < 0, x > 0], [0, lambda x: x]) |

SOFTMAX PARA PROBABILIDAD, para cuando la salida es categorias, no es binaria, no es 0,1 o -1,1

