Deep Learning with keras

CLASSIFICATION

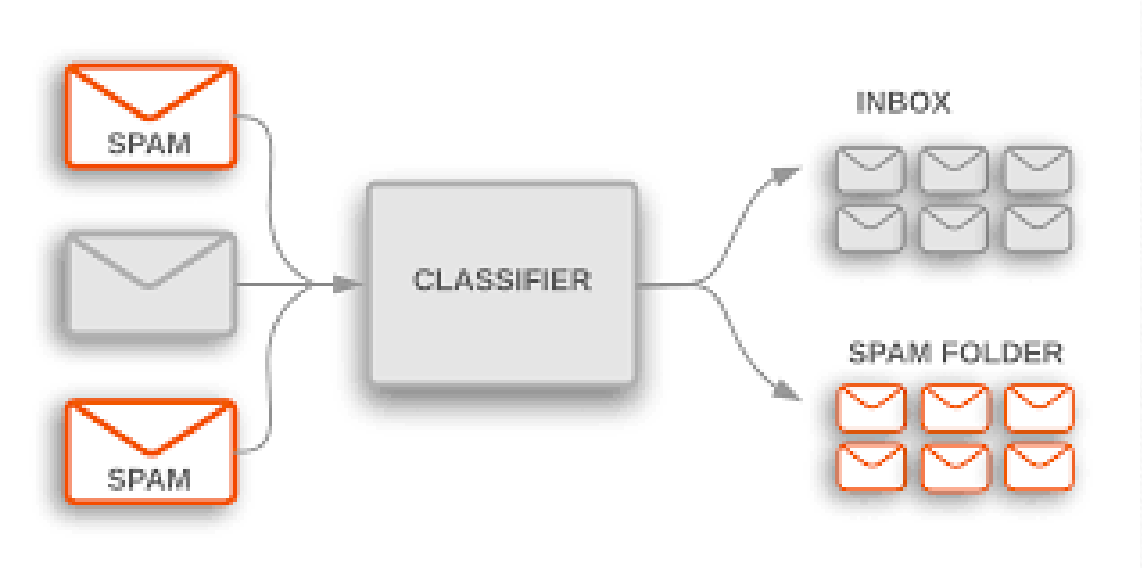
CLASSIFICATION

- **Classification** is the process of predicting the class of given data points. It comes under supervised learning as we are giving the data with labels along with output to train the model.

- A classification problem output variable will be category, such as "Red" or "blue" or "disease" and "no disease" or "spam" or not "spam" .

EXAMPLE OF CLASSIFICATION

- The dataset will be provided which has both spam and normal mails .
- Based on this data the model is trained and build after this the output can be predicted for any unknown input
- Finally classifies the email as spam or not

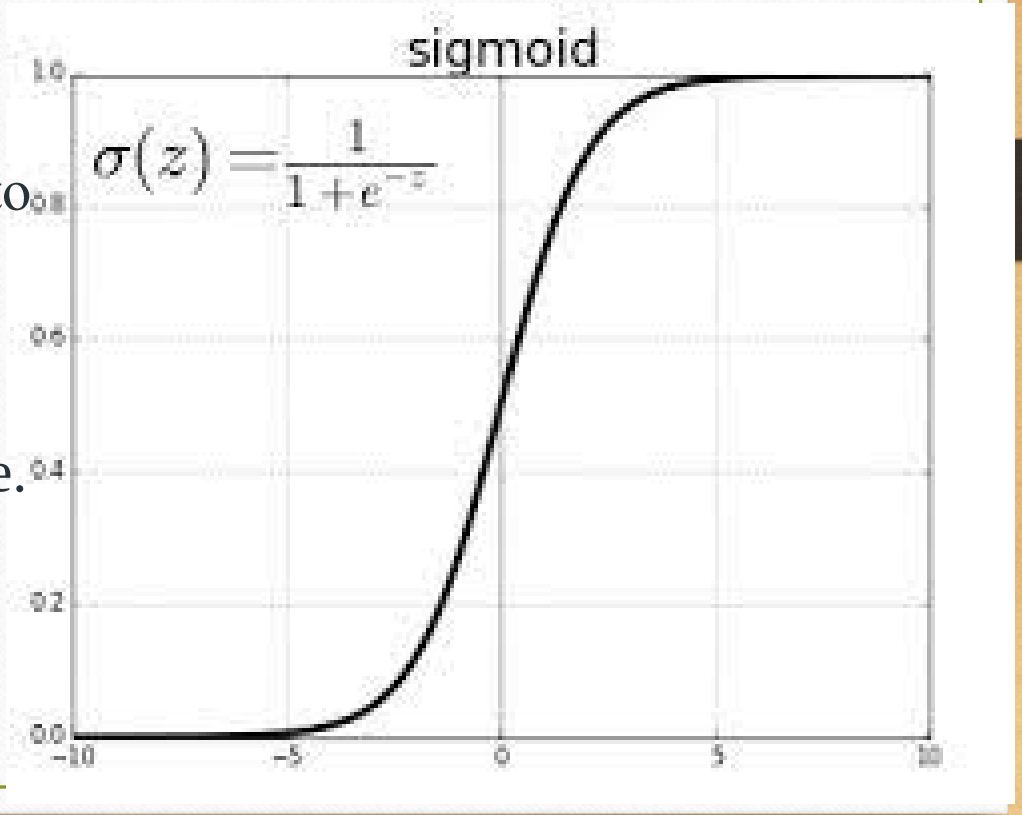


Sigmoid

It is also called as a *Binary classifier* or *Logistic Activation function* because function always pick value either 0(False) or 1 (True).

If the sigmoid neuron's output is larger than or equal to 0.5, it outputs 1; if the output is smaller than 0.5, it outputs 0.

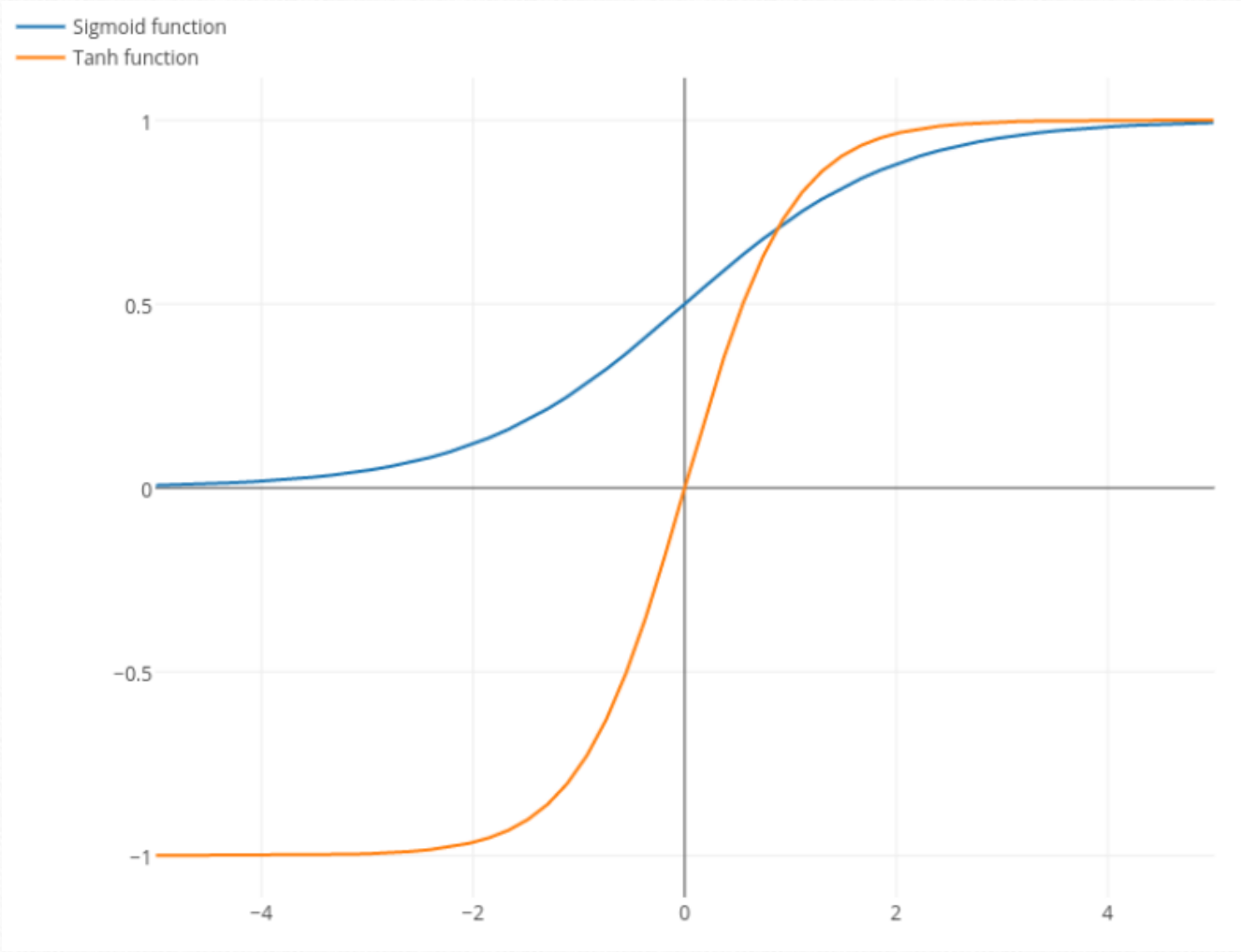
The sigmoid function does not have a jerk on its curve. It is smooth and it has a very nice and simple derivative, which is differentiable everywhere on the curve.

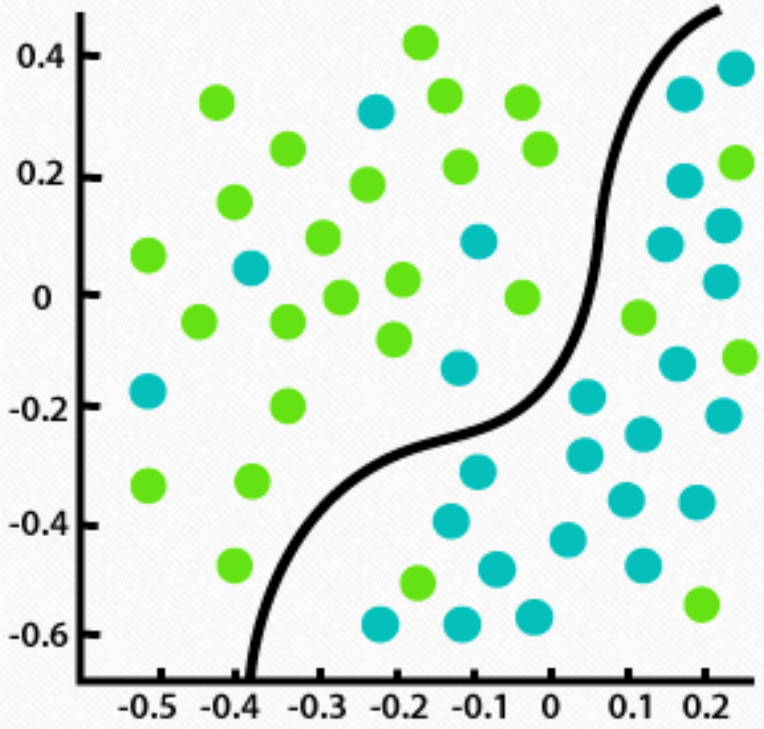


Tanh :

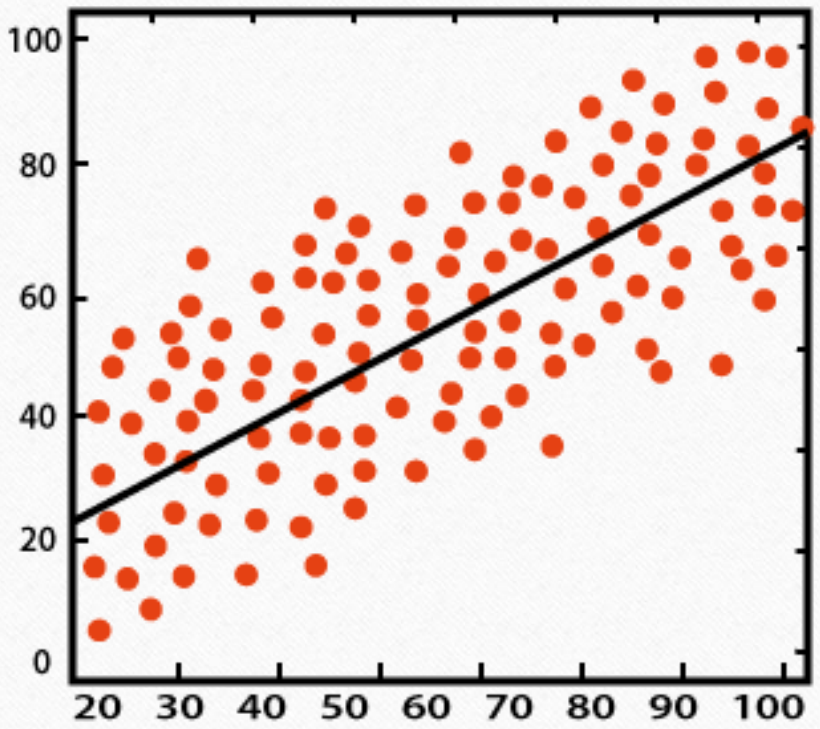
The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s — shaped). It is also called the hyperbolic tangent function.

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$





Classification



Regression

Softmax :

Softmax calculates a probability for every possible class. Softmax assumes that each example is a member of exactly one class

In this case it clear that the input belongs to class 1. So if the probability of any of these classes is changed, the probability value of the first class would also change.

Example :

$$2.33 \rightarrow P(\text{Class 1}) = \frac{\exp(2.33)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.83827314$$

$$-1.46 \rightarrow P(\text{Class 2}) = \frac{\exp(-1.46)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.01894129$$

$$0.56 \rightarrow P(\text{Class 3}) = \frac{\exp(0.56)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)} = 0.14278557$$

- from keras.models import Sequential
- from keras.layers import Dense

- from keras.layers import Dense, Dropout, Conv2D, MaxPool2D, Flatten

- model = Sequential()

- model.add(Dense(12, input_dim=8, activation='relu’))

- model.add(Dense(8, activation='relu’))

- model.add(Dense(1, activation='sigmoid’))

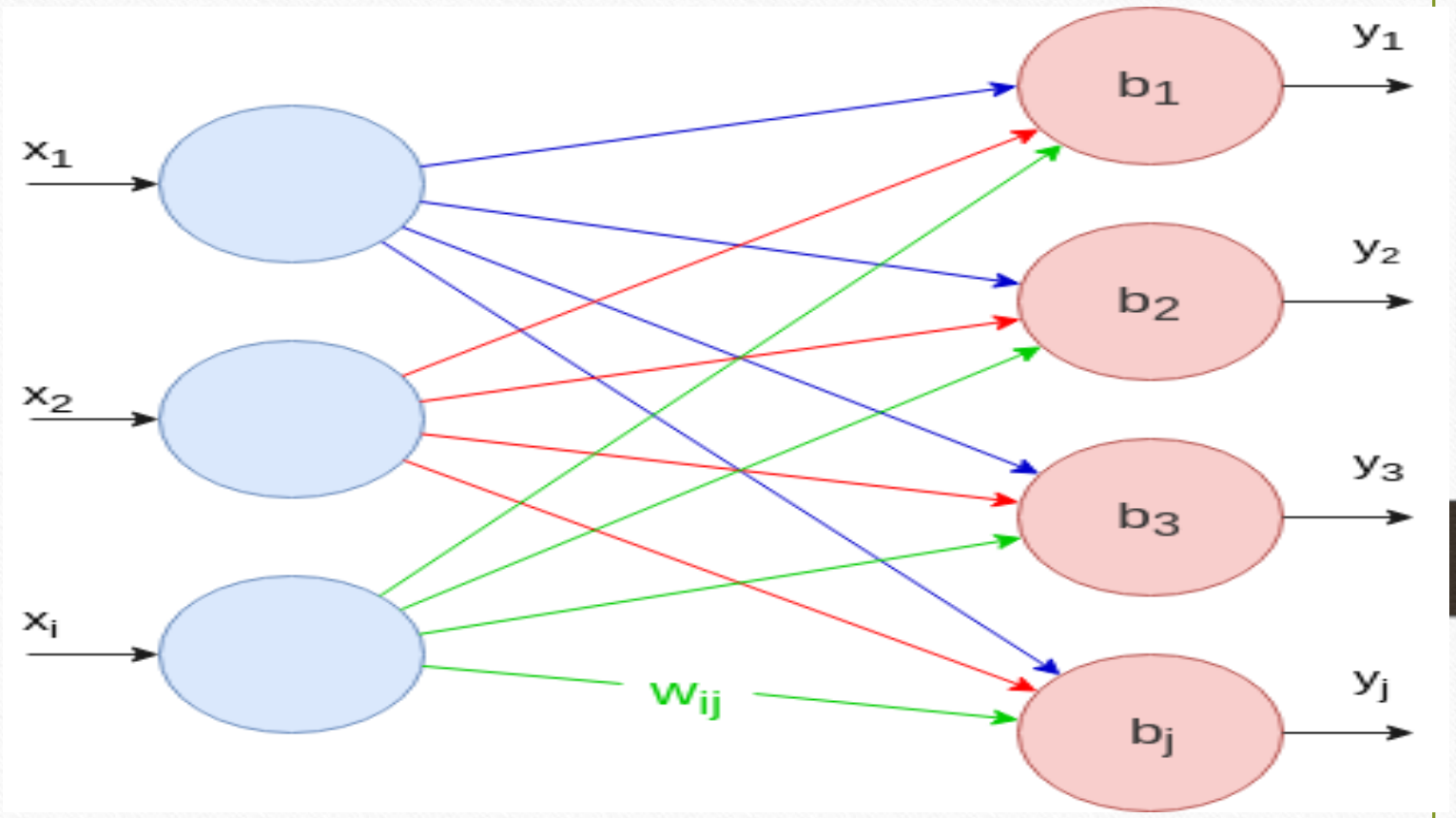
- model.add(Dense(128, activation=tf.nn.relu))

- model.add(dense(64))
  model.add(Activation('relu'))

Fully Connected Layer

FC layers are the most basic layers as every input neurons are connected to every output neurons.

With matrices, we can compute this formula for every output neuron in one shot using a **dot product** :



$$Y = XW + B$$

$$X = \begin{bmatrix} x_1 & \dots & x_i \end{bmatrix} \quad W = \begin{bmatrix} w_{11} \dots w_{1j} \\ \vdots \quad \ddots \quad \vdots \\ w_{i1} \dots w_{ij} \end{bmatrix} \quad B = \begin{bmatrix} b_1 & \dots & b_j \end{bmatrix}$$