Deep Learning with keras

Regression

REGRESSION

- Regression analysis is a form of predictive modelling technique which investigates the relationship between a **dependent** (target) and **independent variable (s)** (predictor) and output the continuous variables or real values
- A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

“The goal is to produce a model that represents the ‘best fit’ to some observed data, according to an evaluation criterion.”

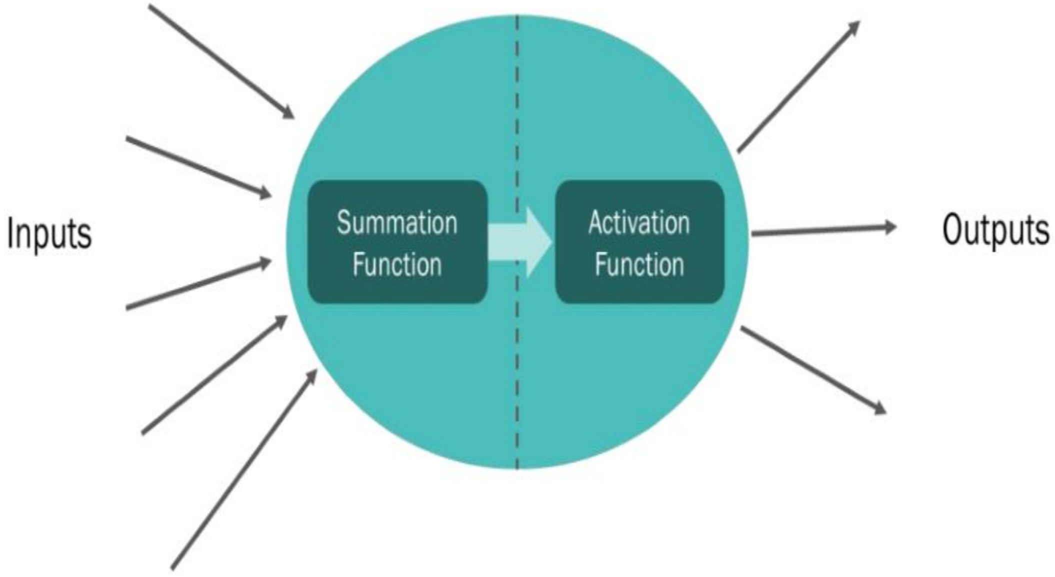
EXAMPLE OF REGRESSION

- Here the based on the data of salary and experience the model is trained .
- Now, for new input experience

the salary can be predicted which is not a category but a real value so this comes under regression .

ACTIVATION FUNCTION

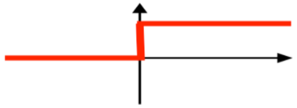
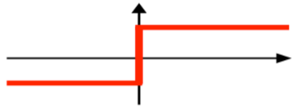
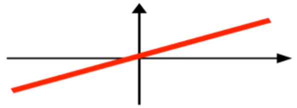
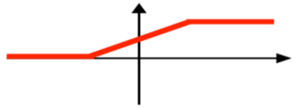
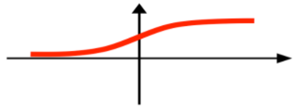
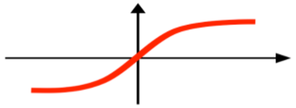
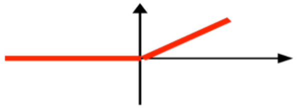
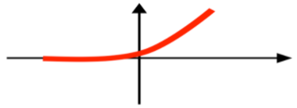
An activation function is a mapping of summed weighted input to the output of the neuron. It is called an activation/ transfer function because it governs the inception at which the neuron is activated and the strength of the output signal.

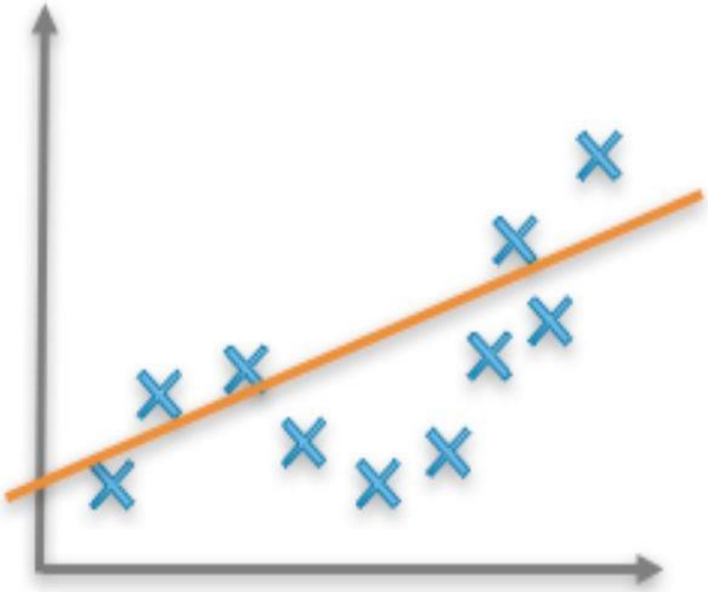


Summation and Activation Functions

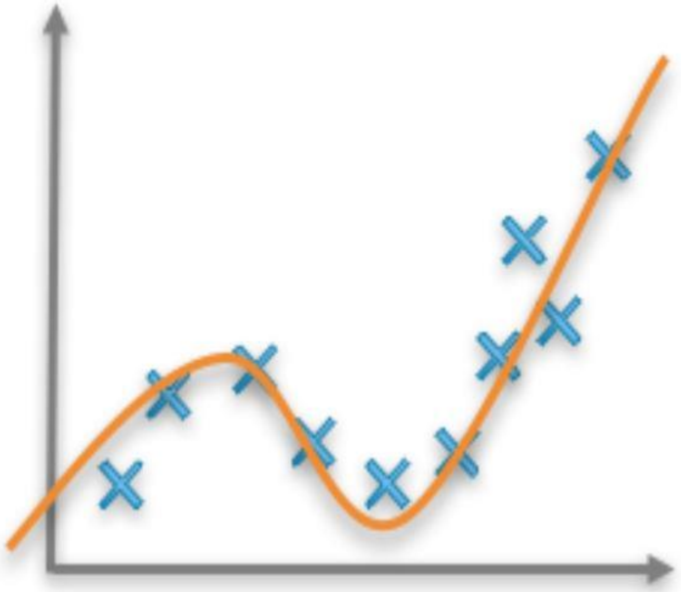
Mathematically,

$$Y = \Sigma(weight * input) + bias$$

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

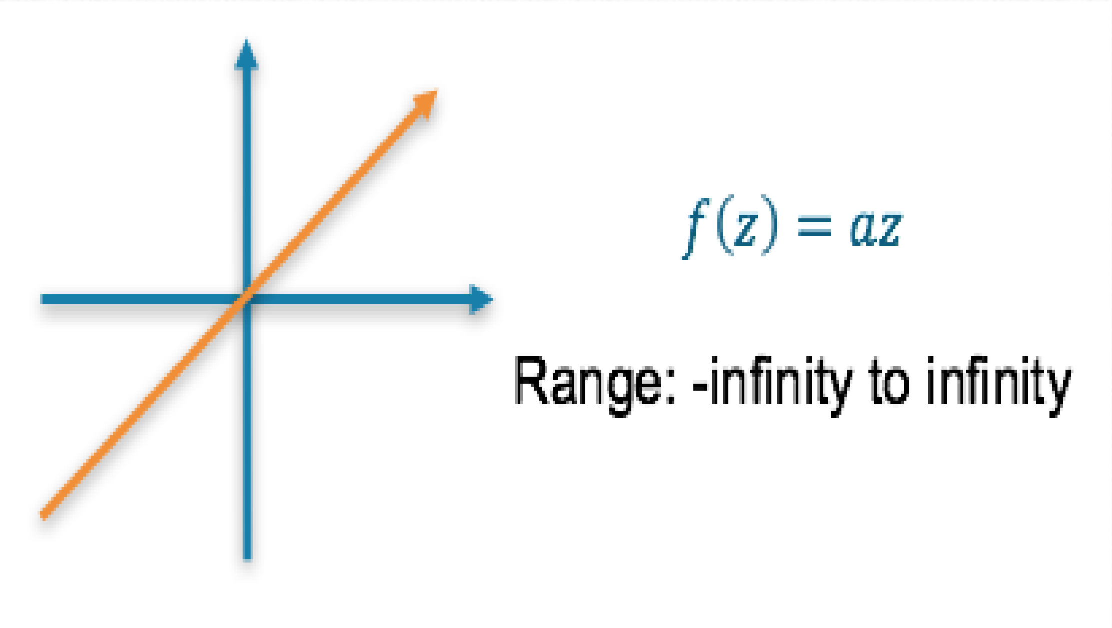


Linear function



Non-linear function

Linear — This results in a numerical value which we require



Back-propagation is not possible — The derivative of the function is a constant, and has no relation to the input, X. So it's not possible to go back and understand which weights in the input neurons can provide a better prediction.

Non-linear Activation Function

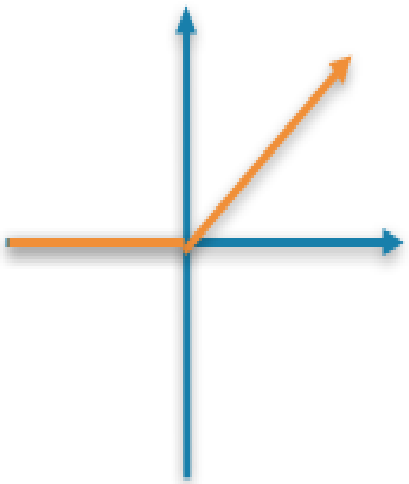
Modern neural network models use non-linear activation functions. They allow the model to create complex mappings between the network's inputs and outputs, which are essential for learning and modeling complex data, such as images, video, audio, and data sets which are non-linear or have high dimensionality.

1. They **allow back-propagation** because they have a derivative function which is related to the inputs.
2. They **allow "stacking" of multiple layers** of neurons to create a deep neural network. Multiple hidden layers of neurons are needed to learn complex data sets with high levels of accuracy.

ReLU (Rectified Linear Unit)

ReLU — This results in a numerical value greater than 0

1.Computationally efficient — allows the network to converge very quickly
2.Non-linear — although it looks like a linear function, ReLU for back-propagationn



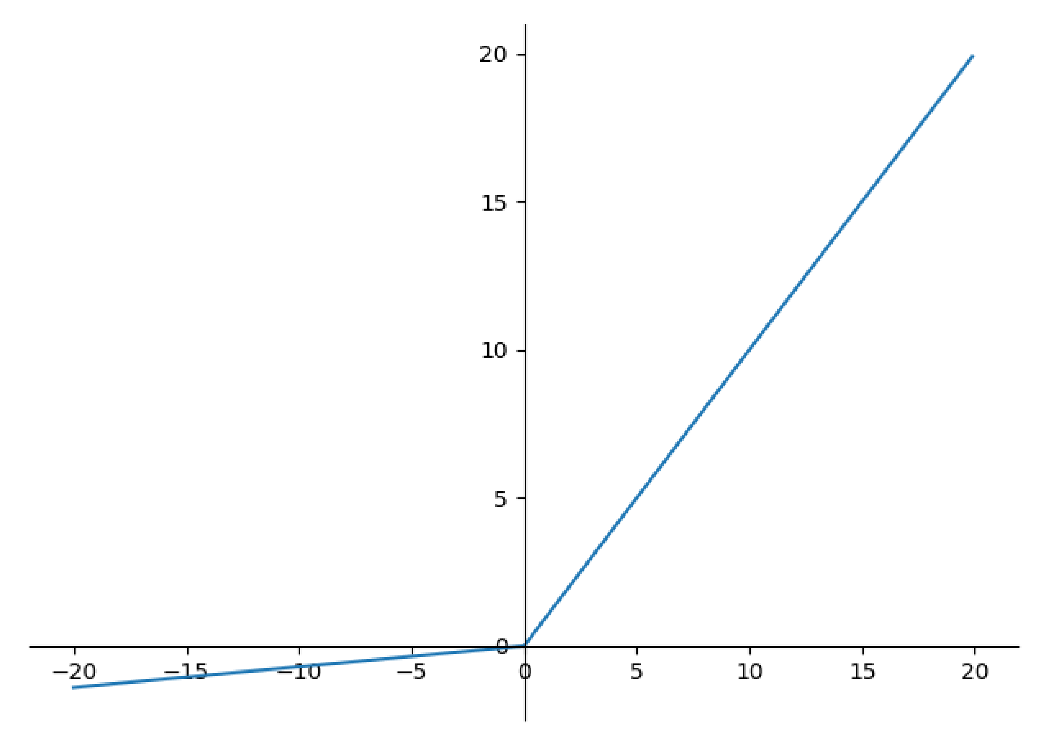
$$relu(z) = \max(0, z)$$

Range: 0 to infinity

$$R'(z) = \begin{Bmatrix} 1 & z > 0 \\ 0 & z < 0 \end{Bmatrix}$$

Leaky ReLU

Prevents dying ReLU problem — this variation of ReLU has a small positive slope in the negative area, so it does enable back-propagation, even for negative input values



- **Derivative:** f'(x) = {0.01 ; if z<0,
  1; otherwise}

PReLU

PReLU(Parametric ReLU, PReLU, ReLU with parameters) introduce a learnable parameter, **Different neurons can have different parameters**. **PReLU**, is an activation function that generalizes the traditional rectified unit with a slope for negative values

$$\mathrm{PReLU}_i(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma_i x & \text{if } x \leq 0 \end{cases}$$