Module 2

Q) Define activation function. Explain different types of activation functions.

- Activation Functions are extremely important feature of the Artificial Neural Network. They basically decide whether a neuron should be activated or not. It limits the output signal to a finite value.
- **Activation Function does the non-linear transformation** to the input making it capable to learn more complex relation between inputand output. It make the network capable of learning more complex pattern.
- Without an activation function, the neural network is just a linear regression model as it performs only summation of product of input and weights.

Eg. In the below image 2 requires a complex relation which is curve unlike a simple linear relation in image 1.

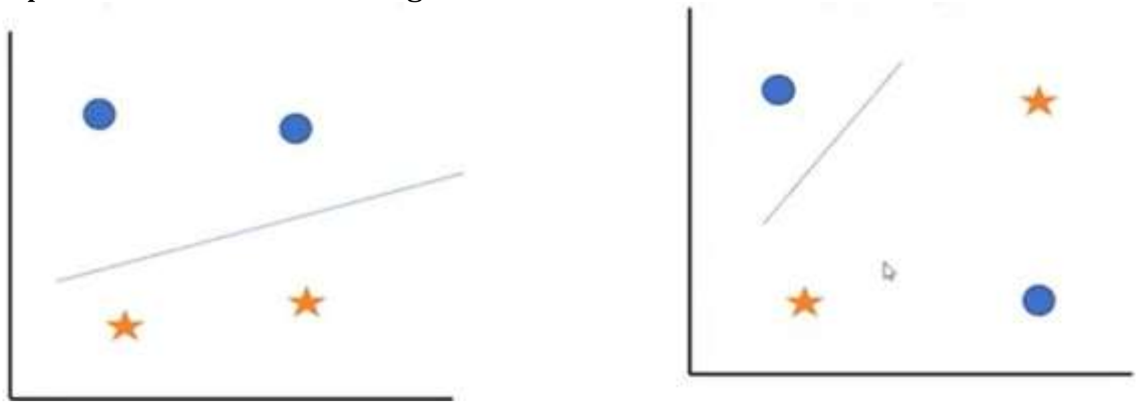


Fig. Illustrating the need of Activation Function for a complex problem.

Activation function must be efficient and it should **reduce the computation** time because the neural network sometimes trained on millions of data points.

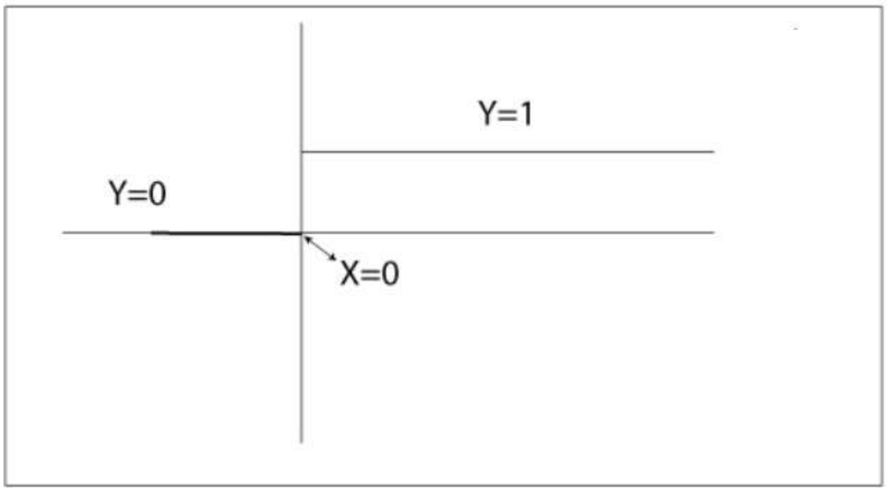
Types of AF:
The Activation Functions can be basically divided into 3 types-
1. Binary step Activation Function
2. Linear Activation Function
3. Non-linear Activation Functions

1. Binary Step Function

A binary step function is a threshold-based activation function. If the input value is above or below a certain threshold, the neuron is activated and

Module 2

sends exactly the same signal to the next layer.We decide some threshold value to decide output that neuron should be activated or deactivated.It is very simple and useful to classify binary problems or classifier.
Eg.f(x) = 1 if x > 0 else 0 if x <= 0



2. Linear or Identity Activation Function

As you can see the function is a line or linear. Therefore, the output of the functions will not be confined between any range.

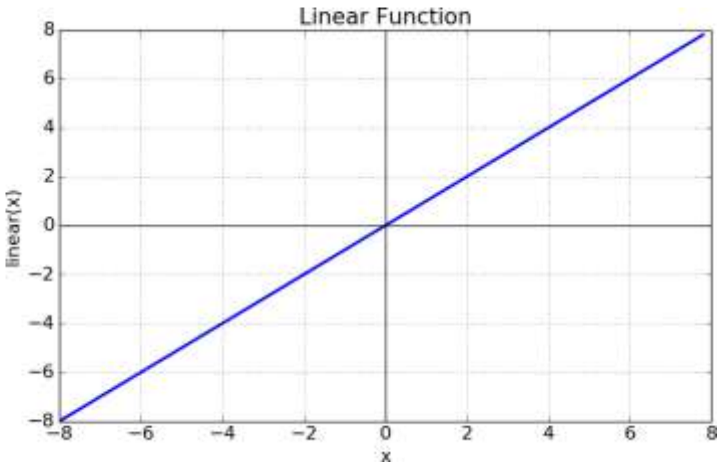


Fig: Linear Activation Function

Equation: f(x) = x

Range : (-infinity to infinity)

It doesn‟t help with the complexity or various parameters of usual data that is fed to the neural networks

3. Non-linear Activation Function

The Nonlinear Activation Functions are the most used activation functions. Nonlinearity helps to makes the graph look something like this.

Module 2

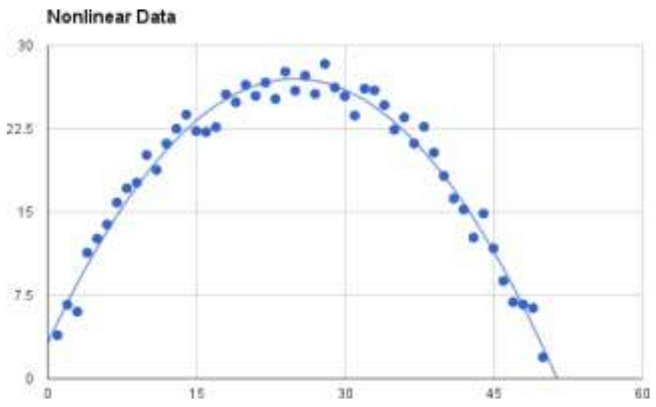


Fig: Non-linear Activation Function

The main terminologies needed to understand for nonlinear functions are:

Derivative or Differential: Change in y-axis w.r.t. change in x-axis.It is also known as slope.

Monotonic function: A function which is either entirely non-increasing ornon-decreasing.

The Nonlinear Activation Functions are mainly divided on the basis of their range or curves-

Advantage of Non-linear function over the Linear function :

Differential is possible in all the non -linear function.

Stacking of network is possible, which helps us in creating deep neural nets.

It makes it easy for the model to generalize

3.1 Sigmoid(Logistic AF)(σ):

The main reason why we use sigmoid function is it exists between **0 to 1.**

It is especially used for models where we have to predict the probability as output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

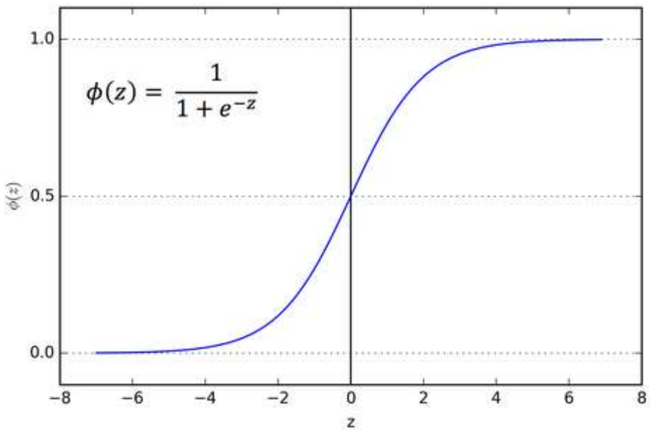


Fig: Sigmoid Function (S-shaped Curve)

The function is **differentiable and monotonic**. But function derivative isnot monotonic.

The logistic sigmoid function can cause a neural network to get stuck at the training time.

Module 2

Advantages

1. Easy to understand and apply
2. Easy to train on small dataset
3. Smooth gradient, preventing "jumps" in output values.
4. Output values bound between 0 and 1, normalizing the output of each neuron.

Disadvantages:

- Vanishing gradient—for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction.
- Outputs not zero centered.
- Computationally expensive

3.2 TanH(Hyperbolic Tangent AF):

TanH is also like logistic sigmoid but in better way. The range of the TanHfunction is from **-1 to +1.**

TanH is often preferred over the sigmoid neuron because it is zero centred. The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in tanh graph.

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

tanh(x) = 2 * sigmoid(2x) - 1

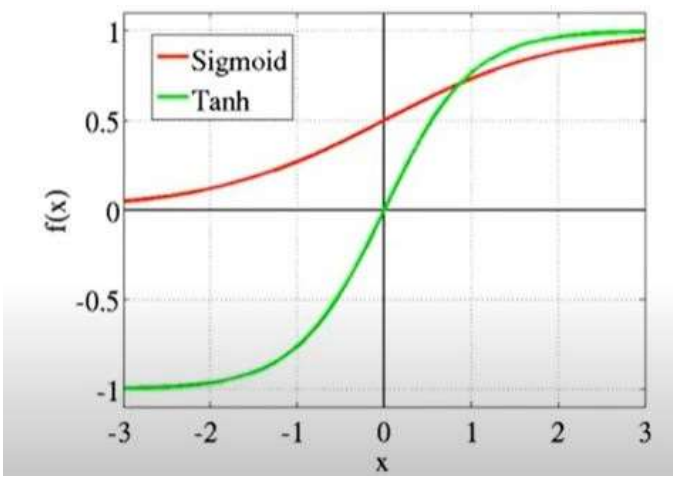


Fig. Sigmoid Vs Tanh

The function is **differentiable** and **monotonic**. But function derivative isnot monotonic.

Advantages

- **Zero centered**—making it easier to model inputs that have strongly negative, neutral, and strongly positive values.

Module 2

Disadvantages

- Like the Sigmoid function is also suffers from vanishing gradient problem
- hard to train on small datasets

3.3 ReLU(Rectified Linear Unit):

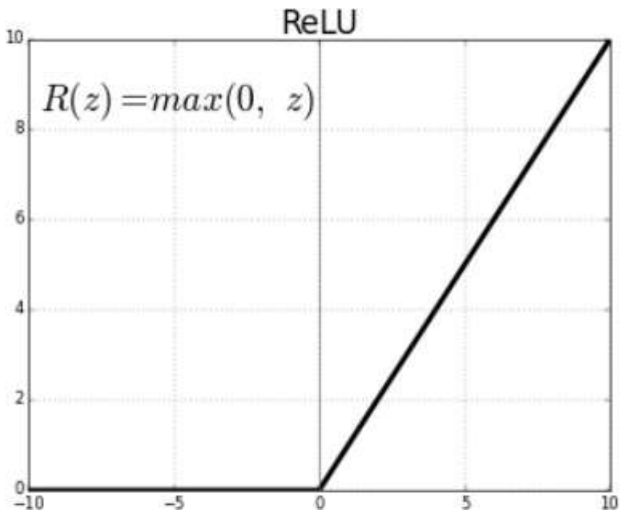
The ReLU is the most used activation function. It is used in almost all convolution neural networks in hidden layers only.
The ReLU is half rectified(from bottom). f(z) = 0, if z < 0
= z, otherwise
R(z) = max(0,z)
The range is **0 to inf.**

Advantages

- **Avoids vanishing gradient problem.**
- **Computationally efficient—**allows the network to converge very quickly
- **Non-linear—**although it looks like a linear function, ReLU has a derivative function and allows for backpropagation

Disadvantages

- Can only be used with a hidden layer
- hard to train on small datasets and need much data for learning non-linear behavior.
- **The Dying ReLU problem—**when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.



The function and its derivative **both are monotonic**.

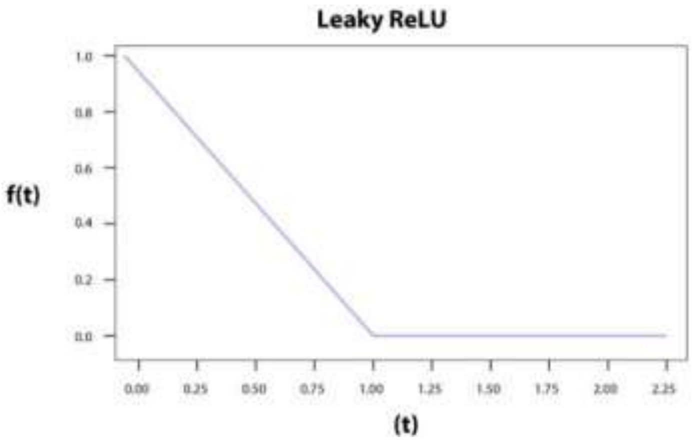
All the negative values are converted into zero, and this conversion rate is so fast that neither it can map nor fit into data properly which creates a problem.

Module 2

Leaky ReLU Activation Function

We needed the **Leaky ReLU activation** function to solve the „*Dying ReLU*‟ problem.

Leaky ReLU we do not make all negative inputs to zero but to a value nearto zero which solves the major issue of ReLU activation function.



R(z) = max(0.1*z,z)

Advantages

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

Disadvantages

- **Results not consistent**—leaky ReLU does not provide consistent predictions for negative input values.

3.4 Softmax:

- Sigmoid able to handle more than two cases(class label).
- Softmax can handle multiple cases. Softmax function squeeze the output for each class between 0 and 1 with sum of them is 1.
- It is ideally used in the final output layer of the classifier, where weare actually trying to attain the probabilities.
- Softmax produces multiple outputs for an input array. For this reason, we can build neural network models that can classify more than 2 classes instead of binary class solution.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

sigma = softmax
z_i = input vector
e^{z_i}} = standard exponential function for input vectorK = number of classes in the multi-class classifier

e^{z_j} = standard exponential function for output

vectore^{z_j} = standard exponential function for output vector

Advantages

Able to handle multiple classes only one class in other activation functions—normalizes the outputs for each class between 0 and 1with the sum of the probabilities been equal to 1, and divides by their sum, giving the probability of the input value being in a specific class.

Useful for output neurons—typically Softmax is used only for the output layer, for neural networks that need to classify inputs into multiple categories.