Artificial Intelligence and Machine Learning

UNIT 5 – NEURAL NETWORKS

1, ReLU :

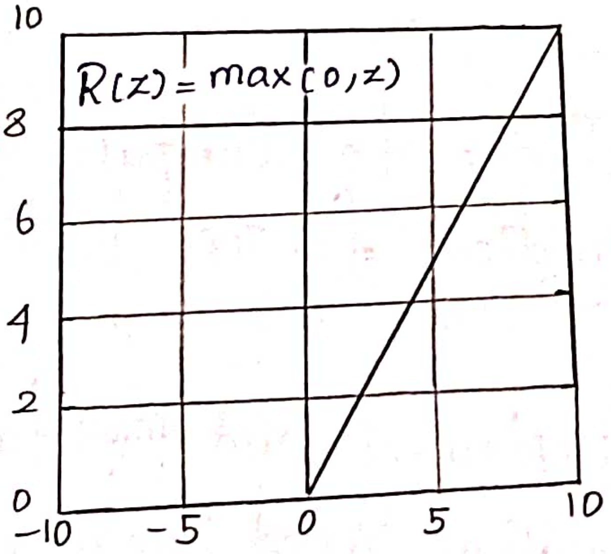
=> Rectified Linear Unit (ReLU) Solve the Vanishing Gradient Problem. ReLU is a non-linear function or Piecewise linear function that will output the input directly if it is Positive, Otherwise, it will Output Zero.

=> It is most Commonly used activation function in neural networks, especially in Convolutional Neural Networks (CNNs) and Multilayer Perceptrons.

=> Mathematically, it is expressed as

$$f(x) = \max(0, x)$$

Where x : input to neuron



ReLU function

=> The derivative of an activation function is required When updating the weights during the back Propagation of the error. The Slope of ReLU is 1 for Positive Values and 0 for negative Values. It becomes non-differentiable when the input x is Zero, but it can be Safely assumed to zero and Causes no Problem in Practice.

=> ReLU is used in the hidden layers instead of Sigmoid or tanh. The ReLU function Solves the Problem of Computational Complexity of the Logistic Sigmoid and Tanh functions

=> A ReLU activation unit is Known to be less likely to Create a Vanishing gradient Problem because its derivative is always 1 for Positive Values of the argument.

Advantages

a, ReLU is Simple to Compute and has a Predictable gradient for the back Propagation of the error.

b, Easy to implement and Very fast

c, The Calculation Speed is Very fast. The ReLU function has only a direct relationship.

d, It Can be used for deep network training.

Disadvantages:

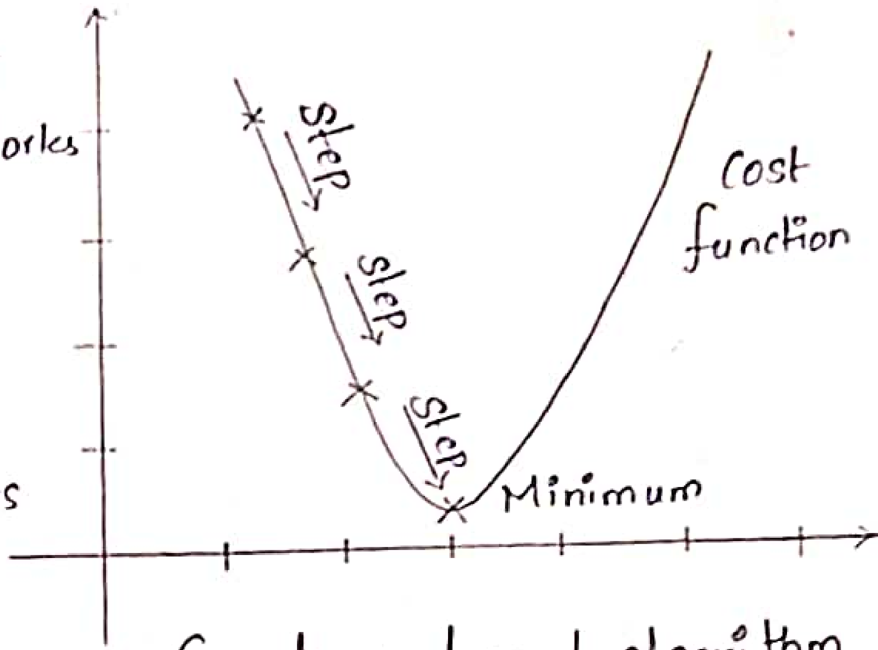
a, When the input is negative, ReLU is not fully functional which means when it Comes to the wrong number installed, ReLU will die. This Problem is also known as the dead Neurons Problem.

b, ReLU function Can Only be used within hidden layers of a Neural Network Model

2, Gradient Descent Optimization:

=> Gradient descent is an optimization algorithm in gadget mastering used to limit a feature with the aid of iteratively moving towards the minimal fee of the Charateristic

=> We essentially use this algorithm when we have to locate the least Possible Values which Could fulfill a given fee function. In gadget getting to know, greater regularly that not we try to limit loss features (like mean Squared error). By minimizing the loss Characteristic, we will improve Our model and Gradient descent is one of the most Popular algorithms used for this Cause.



Gradient descent algorithm

=> The graph Shows Exactly how Gradient Descent Set of rules Works

=> We first take a factor in the Value function and begin Shifting in steps in the direction of the minimum factor. The Size of that Step, Or how quickly we Ought to Converge to the minimum factor is defined by Learning Rate. We Can cowl more location with better learning fee but at the risk of Overshooting the minima. On the opposite hand, Small Steps/ Smaller gaining knowledge of charges will eat a number of time to attain the lowest point

=> Now, the direction wherein algorithm has to transpor (Closer to minimal) is also important. This is Calculated using derivatives. A Spinoff is largely Calculated because of the Slope. We get that Calculated because of the Slope. We get that tangent line to graph at the Point. The extra Steep the tangent, would Suggest more Steps would be needed to reach minimum Point, much less Steep might Suggest lesser Steps are required to reach the minimum factor.

3. Error Backpropagation :-

* Backpropagation is one of the important concepts of a neural network.

* Its algorithm calculates the gradient of the error function.

* And the set of methods used to efficiently train artificial neural networks following a gradient descent approach which exploits the chain rule.

* The main features of backpropagation are the iterative, recursive and efficient method through which it calculates the updated weight to improve the network

* Derivatives of the activation function to be known at network design time is required to backpropagation.

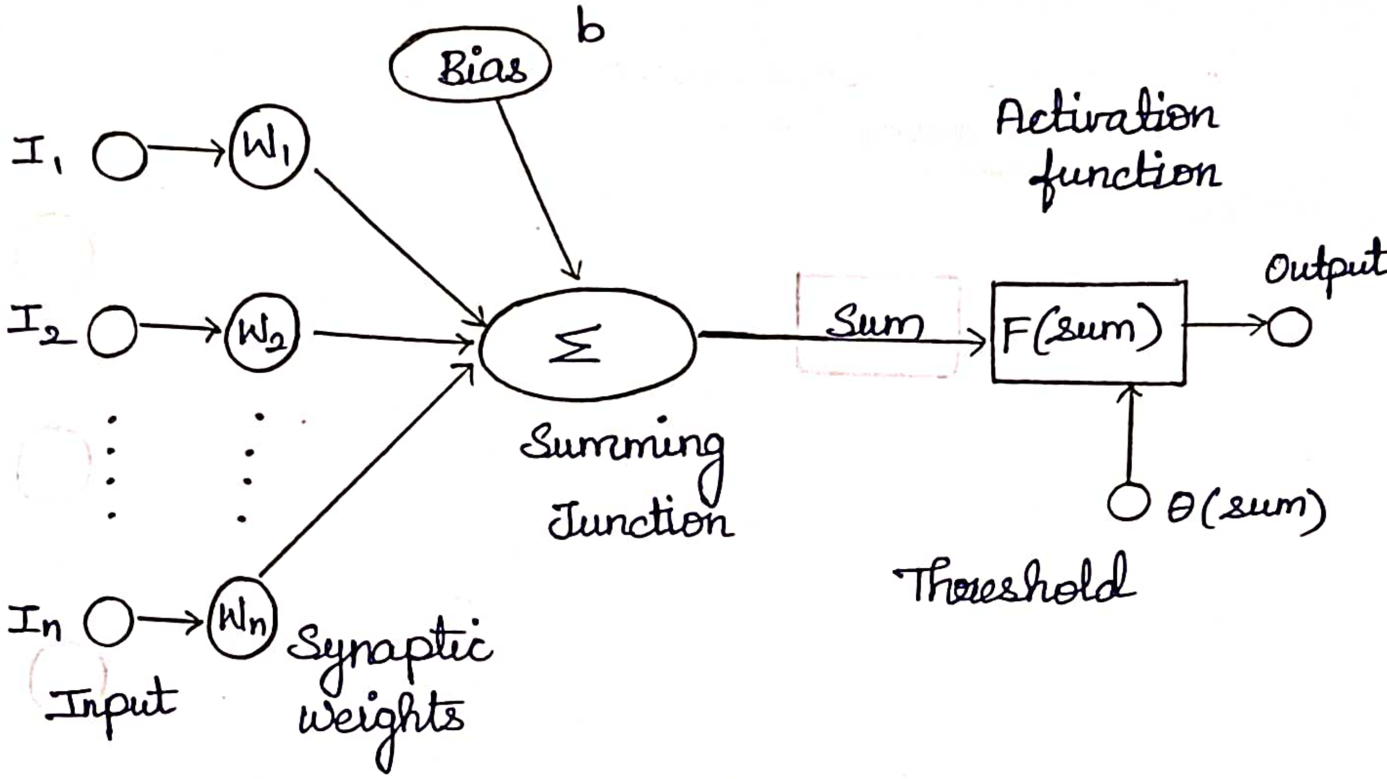
How backpropagation algorithm works?

* The algorithm in neural network computes the gradient of the loss function for a single weight by the chain rule.

* It efficiently computes one layer at a time, unlike a native direct computation.

* It computes the gradient, but it does not define how the gradient is used

* It generalizes the computation in the delta rule.



1. Inputs arrive through the preconnected path.

2. Input is modeled using real weights W. The weights are usually randomly selected.

3. calculate the output for every neuron from the input layer, to the hidden layers to output.

4. calculate the error in the outputs.

Errors = Actual output − desired output

5. Travel back from the output layer to hidden layer to adjust the weights such the error is decreased.

6. Repeating the process until the desired output is achieved.

Types of backpropagation networks :-

Two types of backpropagation networks are,

a. Static backpropagation,

b. Recurrent backpropagation.

Static backpropagation :-

* It is one kind of backpropagation network which produces a mapping of a static input for static output.

* It useful to solve static classification issues like optical character recognition.

Recurrent backpropagation :-

* Recurrent back propagation in data mining is fed forward until a fixed value is achieved.

* After that the error is computed and propagated backward.

Advantages :-

* It does not have any parameters to tune except for the number of inputs.

* It is a standard process that usually works well.

Disadvantages :-

* The performance of backpropagation relies very heavily on the training data.

It needs a very large amount of time for training.

4. Shallow Networks :-

* The term shallow refer to the number of layers in a neural network.

* It have small number of layers, usually regarded as having a single hidden layer and deep neural network refer to neural networks that have multiple hidden layers.

* Both types of networks perform certain tasks better than the other and selecting the right network depth is important for creating successful model.

* The values of the feature vector of the data to be classified are passed, to a hidden layer of nodes

* which generates a response according to some activation function. g, acting on the weighted sum of those values z.

* The responses of each unit in the hidden layer is then passed to a final, outpu layer, whose activation produces the classification prediction output.

Deep Network :-

* Deep learning is a new area of machine learning research, and multiple levels of representation and abstraction that help to make sense of data such as images, sound, text...

* It means using a neural network with several layers of nodes between input and output.

* deep multilayer neural networks with many hiddepn layers.

TensorFlow :-

* TensorFlow is one of the most popular frameworks used to build deep learning models.

* The framework is developed by Google Brain Team.

* Languages like C++, R and python are supported by the framework to create the models as well as the libraries.

* Eg: The translator used by Google is the best example of TensorFlow.

* some of the characteristics of Tensorflow is :-

i) Multiple GPU supported.

ii) one can visualise graphs and queues easily using Tensor Board.

iii) power documentation and larger support from community.

Keras :-

* Keras is built purely on python and can run on the top of TensorFlow.

* Due to this complexity and use of low level libraries, TensorFlow can be comparatively harder to adapt for new users as compared to keras.

* Keras has been developed keeping in mind the complexities in the deep learning models, and it can run quickly to get the results in minimum time.

* Convolutional as well as Recurrent Neural Networks are supported in Keras.

* The framework can run easily on cpu and GPU.

* The models in keras can be classified into 2 categories.

1. Sequential model :-

* The layers in the deep learning model are defined in a sequential manner.

* The implementation of the layers in this model will also be done sequentially.

2. Keras functional API :-

Deep learning models that has multiple outputs or has shared layers i.e more complex models can be implemented in Keras function API