Neural Networks

Mannem Charan AI21BTECH11019

Abstract

This report consists of my basic understanding of one of the popular Ml concept ”Neural Networks”.

1 Neural Networks

Neural Networks is a computational model used in machine learning for classification and regression problems.It has been modified over the years and theoritical it is said that it can fit any data.Neural Networks are introduced around 1940s and inspired from how the human brain process the information.It has been one of the successfull algorithmm over the years and it can take large amount of training data.This technique is used for solving problems in image processing,pattern recognization,NLP and any AI application you name.So first we will the architecture of neural network.

2 Architecture of Neural Network

To understand Neural Network, first we should know what is neuron. Neuron is biological term which is the basic unit for computation in human brain.It recieves input signal from other neurons/nodes and computes the output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs. The node applies a function to the weighted sum of its inputs.If this weighted sum crosses sum theshold value then you will get an output. The same is shown in fig 2.1, With that intro on neuron/node we will see how a simple neural network looks like as shown figure 2.2, A simple neural network consists of following things,

1) **Input Layer:** Input layer in the neural network comprises of all the information brought by the input neuron. And this information is passed to Hidden layer.
2) **Hidden layer:** The nodes in the hidden layers represents the activation functions and all the computations (using weights) are done in this layer and passes to the output layer. Note that in general neural network have more than one hidden layers so these computation are continued across these hidden layers using **weights** and **bias** until it hits the output layer.
3) **Output Layer:**Here we finally use an activation function that maps to the desired output format.
4) **Connections and weights :** The network consists of Connections b/w the layers which passes the output neuron/node i as the input of node j along with a weight

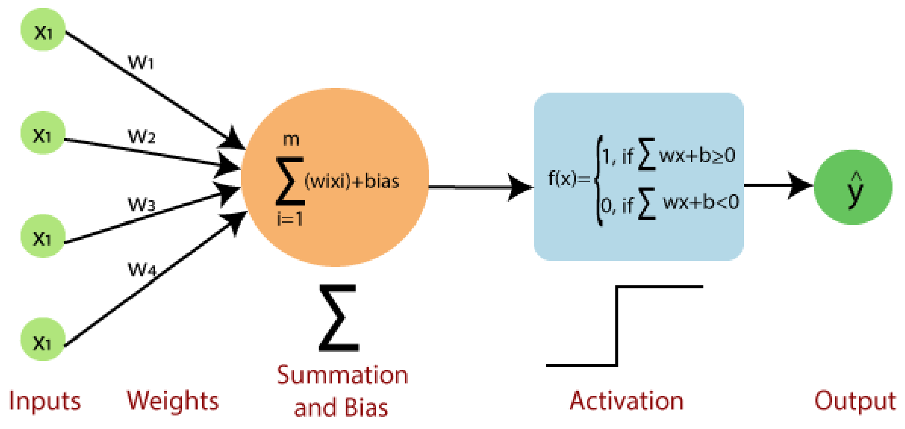

Fig. 2.1: Replication of how neurons compute the output signal

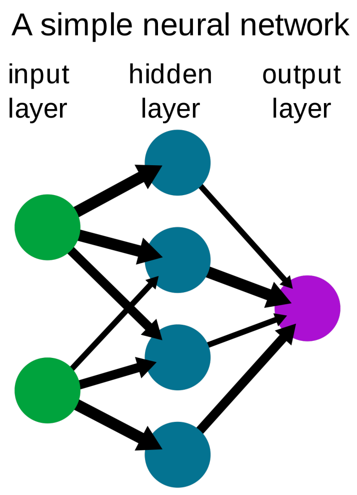

Fig. 2.2: Simple neural network

$W_{j|i}$.And just before surrendering the weighted input to the reciever node , the value will be adjusted with the threshold also known as bias.

5) **Activation Function :** The activation function defines the output of the node given the input at the node.
6) **Learning Rule:**The learning rule is process of modifying these weights and thresholds (*bias*) to fit the given input data.

So when we find the optimal weights and thresholds of the NN for the input data, we will get the predicted output of the input after travelling all these nodes.

3 Types Of Neural Networks

1) **ANN** ANN, artificial neural network is basic neural network and the information will flow only in one direction.It is feed-forward neural network.It can also contain hidden layers.It is used for Textual Data or Tabular Data. A widely used real-life application is Facial Recognition. It is comparatively less powerful than CNN and RNN.
2) **CNN** Convolutional Neural Network is one of leading flavour of NN at present.It is widely used in Computer vision.
3) **RNN** Recurrent Neural Network is better than ANN and less pwerful than CNN.In this type of model, the output from a processing node is fed back into nodes in the same or previous layers.

These neural networks are used in all three types of learning

1) Supervised learning
2) Unsupervised learning
3) Reinforcement learning

4 Backward Propagation

Atlast the optimised weights and bias for the training data are learned using a method known as ”Backward Propagation”.In this method we will learn the parameters from the output layer i.e., backwards. First we will start by assuming that we know all the weights and bias except the bias before output node and we will intialise this bias with 0 and we will calculate the SSR (Sum of squares of residuals) and minimise this error using gradient descent method w.r.t the bias.By doing this we will get optimal bias at the output node. From there we go to the intermediate weights and again we will assume that we know all the parameters except these weights and by minimising the SSR we will get optimal weights.It may sound weird but by thinking lot of load will be lift off, this kind of thinking helps in finding optimal weights and bias.