Feed Forward Neural Network and Back propagation Networks:

Neural networks have become a foundational concept. Two key processes associated with neural networks are Feed-Forward and Back-Propagation. Understanding the difference between these two is crucial for anyone diving into the field of deep learning. This article explores the intricacies of both processes, highlighting their roles, differences, and significance in training neural networks.

What is a Feed-Forward Neural Network?

Overview

A Feed-Forward Neural Network (FFNN) is the simplest form of artificial neural networks. It consists of layers of neurons where information moves in only one direction—forward—from the input layer, through hidden layers, to the output layer. There are no cycles or loops, hence the term "feed-forward."

Key Characteristics

- Unidirectional Data Flow: In a feed-forward network, data flows from the input layer to the output layer without looping back.
- Layer Structure: Typically comprises an input layer, one or more hidden layers, and an output layer.
- Activation Functions: Neurons in the hidden layers apply activation functions (like ReLU, Sigmoid, or Tanh) to introduce non-linearity.
- Output Generation: The final output layer produces a prediction or classification based on the weighted sum of inputs.

Applications

Feed-Forward Neural Networks are commonly used in:

- Classification tasks (e.g., image recognition)
- Regression analysis
- Pattern recognition
- Time series prediction

What is a feed forward neural network?

Feed forward neural networks are artificial neural networks in which nodes do not form loops. This type of neural network is also known as a multi-layer neural network as all information is only passed forward.

During data flow, input nodes receive data, which travel through hidden layers, and exit output nodes. No links exist in the network that could get used to by sending information back from the output node.

A feed forward neural network approximates functions in the following way:

- An algorithm calculates classifiers by using the formula y = f* (x).
- Input x is therefore assigned to category y.
- According to the feed forward model, y = f (x; θ). This value determines the closest approximation of the function.

Feed forward neural networks serve as the basis for object detection in photos, as shown in the Google Photos app.

Cost function

In a feed forward neural network, the cost function plays an important role. The categorized data points are little affected by minor adjustments to weights and biases.

Thus, a smooth cost function can get used to determine a method of adjusting weights and biases to improve performance.

Following is a definition of the mean square error cost function:

$$C(w, b) \equiv \frac{1}{2n} \sum_{x} \|y(x) - a\|^2.$$

Where,

w = the weights gathered in the network

b = biases

n = number of inputs for training

a = output vectors

x = input

‖v‖ = vector v's normal length

Loss function

The loss function of a neural network gets used to determine if an adjustment needs to be made in the learning process.

Neurons in the output layer are equal to the number of classes. Showing the differences between predicted and actual probability distributions. Following is the cross-entropy loss for binary classification.

Cross Entropy Loss:

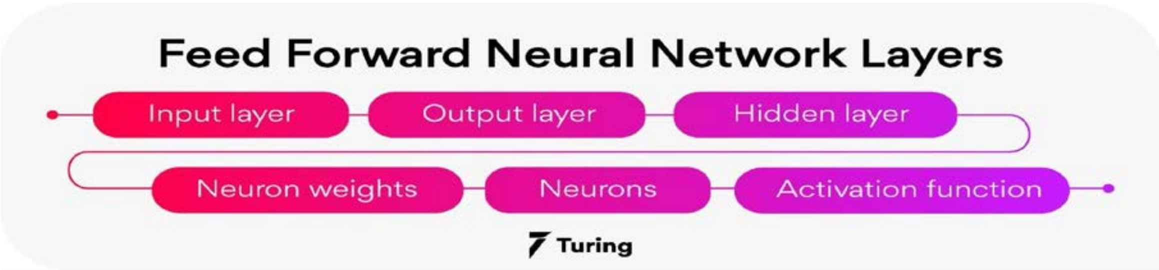
$$L(\Theta) = \begin{cases} -log(\hat{y}) & \text{if } y = 1 \\ -log(1-\hat{y}) & \text{if } y = 0 \end{cases}$$

As a result of multiclass categorization, a cross-entropy loss occurs:

Cross Entropy Loss:

$$L(\Theta) = -\sum_{i=1}^{k} y_i \log(\hat{y}_i)$$

Layers of feed forward neural network



- Input layer:

The neurons of this layer receive input and pass it on to the other layers of the network. Feature or attribute numbers in the dataset must match the number of neurons in the input layer.

- Output layer:

According to the type of model getting built, this layer represents the forecasted feature.

- Hidden layer:

Input and output layers get separated by hidden layers. Depending on the type of model, there may be several hidden layers.

There are several neurons in hidden layers that transform the input before actually transferring it to the next layer. This network gets constantly updated with weights in order to make it easier to predict.

- Neuron weights:

Neurons get connected by a weight, which measures their strength or magnitude. Similar to linear regression coefficients, input weights can also get compared.

Weight is normally between 0 and 1, with a value between 0 and 1.

- Neurons:

Artificial neurons get used in feed forward networks, which later get adapted from biological neurons. A neural network consists of artificial neurons.

Neurons function in two ways: first, they create weighted input sums, and second, they activate the sums to make them normal.

Activation functions can either be linear or nonlinear. Neurons have weights based on their inputs. During the learning phase, the network studies these weights.

- Activation Function:

Neurons are responsible for making decisions in this area.

According to the activation function, the neurons determine whether to make a linear or nonlinear decision. Since it passes through so many layers, it prevents the cascading effect from increasing neuron outputs.

An activation function can be classified into three major categories: sigmoid, Tanh, and Rectified Linear Unit (ReLu).

- Sigmoid:

Input values between 0 and 1 get mapped to the output values.

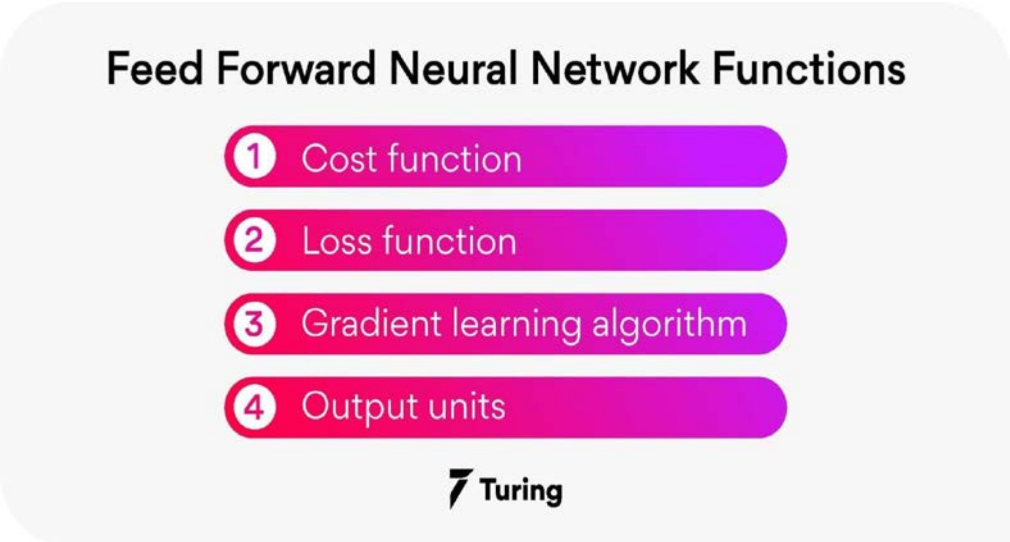
- Tanh:

A value between -1 and 1 gets mapped to the input values.

- Rectified linear Unit:

Only positive values are allowed to flow through this function. Negative values get mapped to 0.

Function in feed forward neural network



What is the working principle of a feed forward neural network?