

**What is deep learning?**

At a very basic level, deep learning is a machine learning technique. It teaches a computer to filter inputs through layers to learn how to predict and classify information. Observations can be in the form of images, text, or sound.The inspiration for deep learning is the way that the human brain filters information. Its purpose is to mimic how the human brain works to create some real magic.

A **feedforward network** is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element computes based upon the weighted sum of its inputs. The new values become the new input values that feed the next layer (feed-forward). This continues through all the layers and determines the output. Feedforward networks are often used in, for example, data mining.

A **feedback network** (for example, a recurrent neural network) has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed. Since loops are present in this type of network, it becomes a non-linear dynamic system which changes continuously until it reaches a state of equilibrium. Feedback networks are often used in optimization problems where the network looks for the best arrangement of interconnected factors.

1.**Neuron**- Just like a neuron forms the basic element of our brain, a neuron forms the basic structure of a neural network. Just think of what we do when we get new information. When we get the information, we process it and then we generate an output. Similarly, in case of a neural network, a neuron receives an input, processes it and generates an output which is either sent to other neurons for further processing or it is the final output.

2) **Weights** – When input enters the neuron, it is multiplied by a weight. For example, if a neuron has two inputs, then each input will have has an associated weight assigned to it. We initialize the weights randomly and these weights are updated during the model training process. The neural network after training assigns a higher weight to the input it considers more important as compared to the ones which are considered less important. A weight of zero denotes that the particular feature is insignificant.Let’s assume the input to be a, and the weight associated to be W1. Then after passing through the node the input becomes a\*W1

3) **Bias** – In addition to the weights, another linear component is applied to the input, called as the bias. It is added to the result of weight multiplication to the input. The bias is basically added to change the range of the weight multiplied input. After adding the bias, the result would look like a\*W1+bias. This is the final linear component of the input transformation.

4) **Activation Function** – Once the linear component is applied to the input, a non-linear function is applied to it. This is done by applying the activation function to the linear combination.The activation function translates the input signals to output signals. The output after application of the activation function would look something like f(a\*W1+b) where f() is the activation function.

**What is a weighted sum?**

Inputs to a neuron can either be features from a training set or outputs from the neurons of a previous layer. Each connection between two neurons has a unique synapse with a unique weight attached. If you want to get from one neuron to the next, you have to travel along the synapse and pay the “toll” (weight). The neuron then applies an activation function to the sum of the weighted inputs from each incoming synapse. It passes the result on to all the neurons in the next layer. When we talk about updating weights in a network, we’re talking about adjusting the weights on these synapses.

**what is an activation function?**

In a nutshell, the activation function of a node defines the output of that node.The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1. It’s an abstraction that represents the rate of action potential firing in the cell. It’s a number that represents the likelihood that the cell will fire. At it’s simplest, the function is binary: **yes** (the neuron fires) or **no** (the neuron doesn’t fire). The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range. If you were using a function that maps a range between 0 and 1 to determine the likelihood that an image is a cat, for example, an output of 0.9 would show a 90% probability that your image is, in fact, a cat.

sigmoid(x) = 1/(1+e-x)

Relu f(x) = max(x,0).