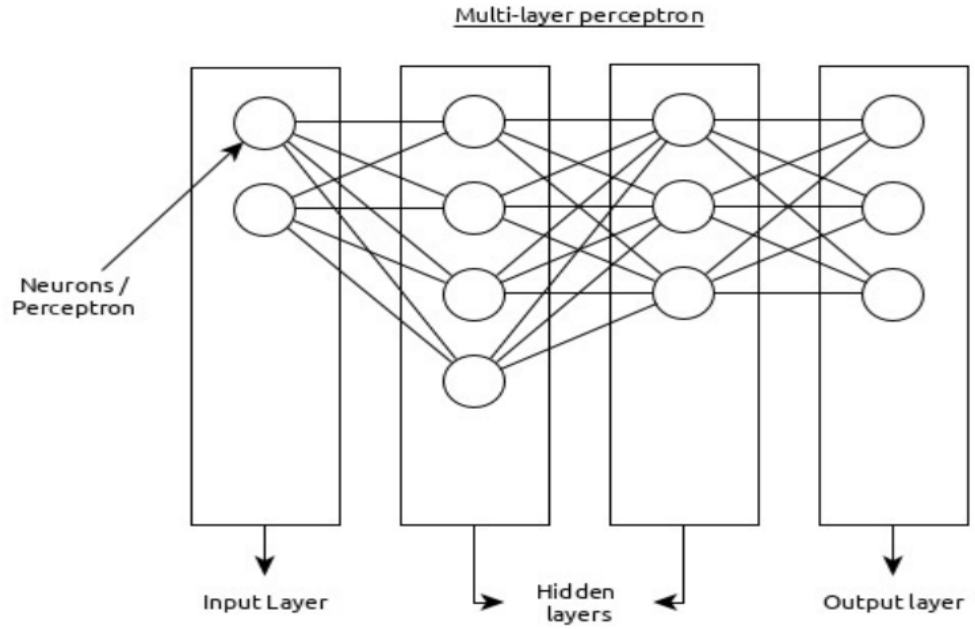
Types of Neural Networks

There are various types of Artificial Neural Networks(ANNs) but here we will discuss some of the well-known ANNs.

Multi-Layer perceptron

Multi-Layer perceptron is the simplest form of ANN. It consists of a single input layer, one or more hidden layers and finally an output layer. A layer consists of a collection of perceptrons. The input layer is basically one or more features of the input data. Every hidden layer consists of one or more neurons and processes a certain aspect of the feature and sends the processed information into the next hidden layer. The output layer process receives the data from the last hidden layer and finally outputs the result.

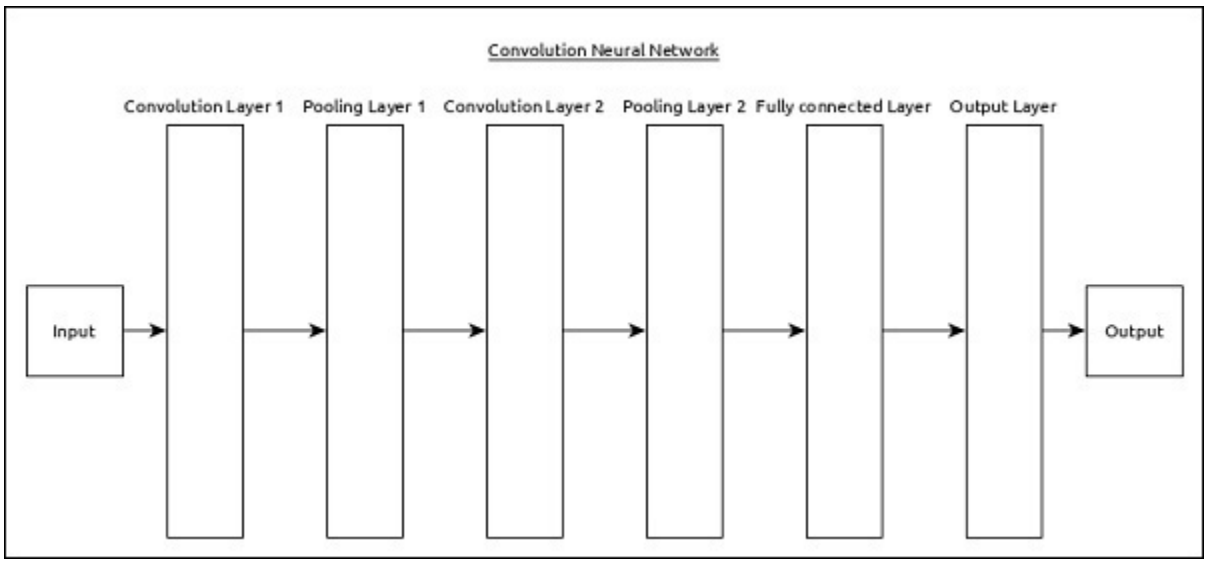


Convolution Neural Network (CNN)

Convolutional neural networks are one of the most popular ANN. It is widely used in the fields of image and video recognition. It is almost similar to a multi-layer perceptron except it contains a series of convolution layers and pooling layers before the fully connected hidden neuron layer. It has three important layers:

- **Convolution layer** – It is the primary building block and performs computational tasks based on convolution function.
- **Pooling layer** – It is arranged next to the convolution layer and is used to reduce the size of inputs by removing unnecessary information so computation can be performed faster.

- **Fully connected layer** – It is arranged next to a series of convolution and pooling layers and classifies input into various categories.



Here,

- 2 series of Convolution and pooling layers are used and it receives and processes the input (e.g. image).
- A single fully connected layer is used and it is used to output the data (e.g. classification of image).

Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNN) are useful to address the flaw in other ANN models. Well, Most of the ANN doesn't remember the steps from previous situations and learned to make decisions based on the context in training. Meanwhile, RNN stores the past information, and all its decisions are taken from what it has learned from the past.

This approach is mainly useful in image classification. Sometimes, we may need to look into the future to fix the past. In this case, bidirectional RNN is helpful to learn from the past and predict the future. For example, we have handwritten samples in multiple inputs. Suppose, we have confusion in one input then we need to check again other inputs to recognize the correct context which takes the decision from the past.