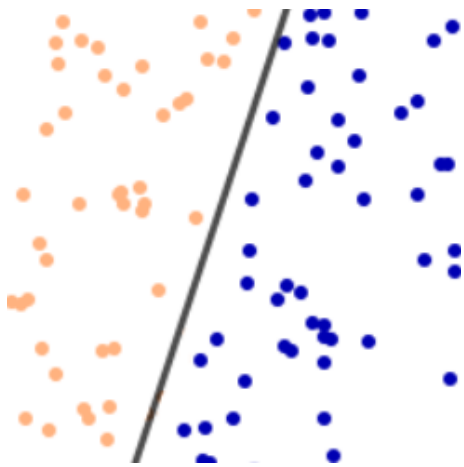
Perceptron Implementation Issues

The perceptron algorithm was one of the first Artificial Neural Networks formed and is the basic component of the multilayer perceptron (MLP). MLP consists of three layers of nodes with each node using a nonlinear activation function apart from the input layer nodes. This makes it different from a linear perceptron.

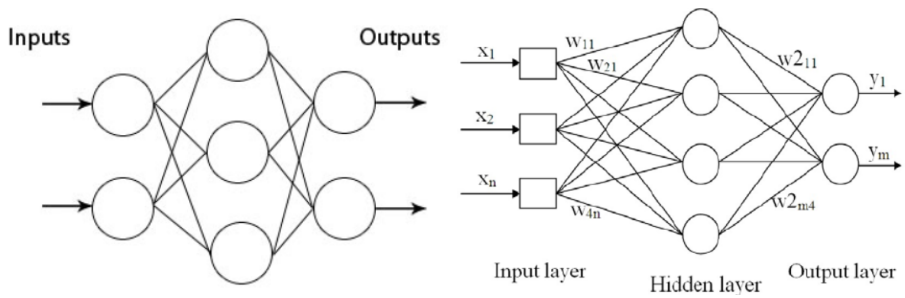
For instance, let us consider a plane with yellow and blue points separated by a straight line as seen in the figure below.



The black line divides the blue and yellow points into the left and right side of it, and this set is called linearly separable. Then, the perceptron algorithm learns where the line is during the training period. Once the learning is done it can predict where a given point would lie.

The following are the implementation issues associated with the perceptron algorithm:

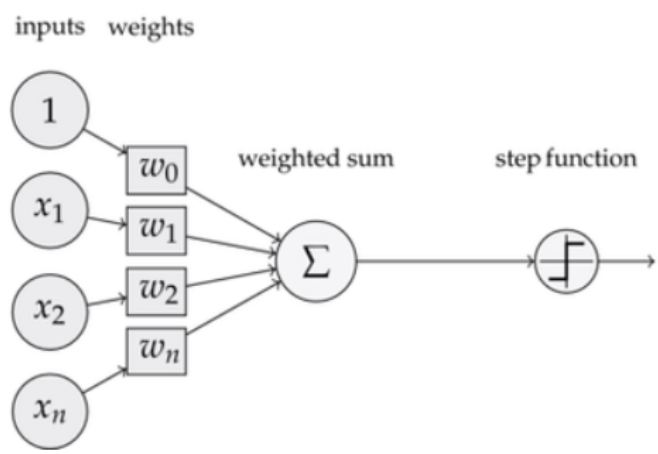
- The perceptron algorithm's output values can only take on two values, zero or one, since it is restricted by the hard-limit transfer function. As seen in the example above, the perceptron can also classify points as blue or yellow.
- This algorithm is only able to classify linearly separable sets of vectors i.e. if vectors are not linearly separable, then the learning process will never be able to classify the vectors properly. Consider the XOR logical operation. In this Boolean function, a line/ hyperplane cannot separate the positive and negative instances (Beale, Jackson).
- Perceptron always converges if the training data is linearly separable. If the data is separable by a hyperplane, then the perceptron will always converge.



The above figure depicts the components of a SLP and an MLP. As seen above, a SLP does not have a hidden layer

- The processing unit of a single-layer perceptron (SLP) network is able to categorize a set of patterns into two classes as the linear threshold function defines their linear separability.
- Due to the way the MLP is trained, it cannot guarantee the minima it stops at during the training process is the global minima. Therefore, it could get stuck in the local minima.

- While using the MLP algorithm, the user must set the number of Hidden Neurons. Setting this value too low may result in underfitting and setting this value too high may result in overfitting.
- Perceptron does not check to see if the chosen weight is optimal. Choosing additional point through input training improves the weights, but it does not check if this would be the most optimal.



The figure above depicts the major components of the perceptron

- With large training sets, error rate is low, but with new test it does not generalize, it just memorizes.
- The single-layer perceptron algorithm cannot make global generalizations based on examples learned locally.
- It is difficult/ almost impossible to design a learning algorithm for MLPs.
- The MLP entails a computationally expensive learning process. A large number of iterations are required for learning, which makes it unsuitable for real-time learning.

- The MLP has a scaling problem. It does not scale up well from small research systems to larger real systems. Both too many and too few units slow down learning.
- For the perceptron, the first layer gets its input from actual input in the network, while the perceptrons in the second layer get their inputs and outputs from the first layer. Sine learning corresponds to strengthening the connections between active inputs and units, but since the actual inputs are masked off, this is impossible to strengthen the correct parts of the network.

Beale, R. Jackson, T. Neural Computing – An Introduction. Retrieved from: https://books.google.co.in