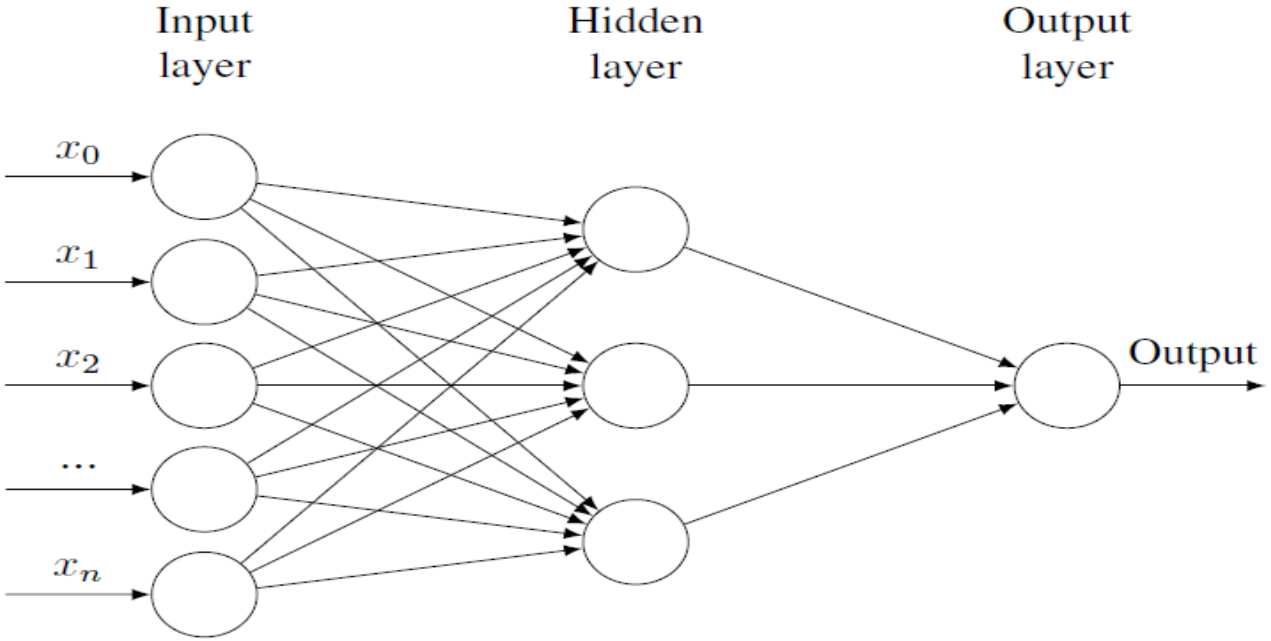
Multi Layer Perceptron (MLP)

Feedforward Network
Multi Layer Perceptron (MLP)

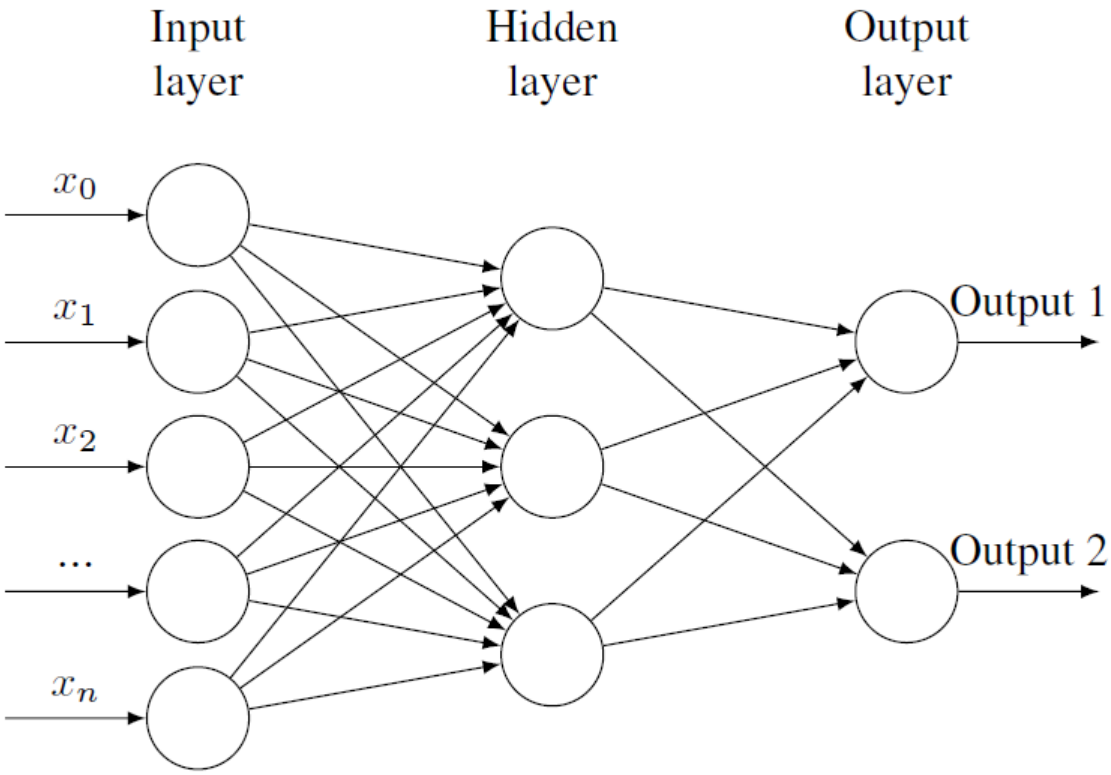
- Networks in which the input signal is fed continuously in one direction from connection to connection until it reaches the output layer
- Feedforward networks have been extensively applied to real-world problems.



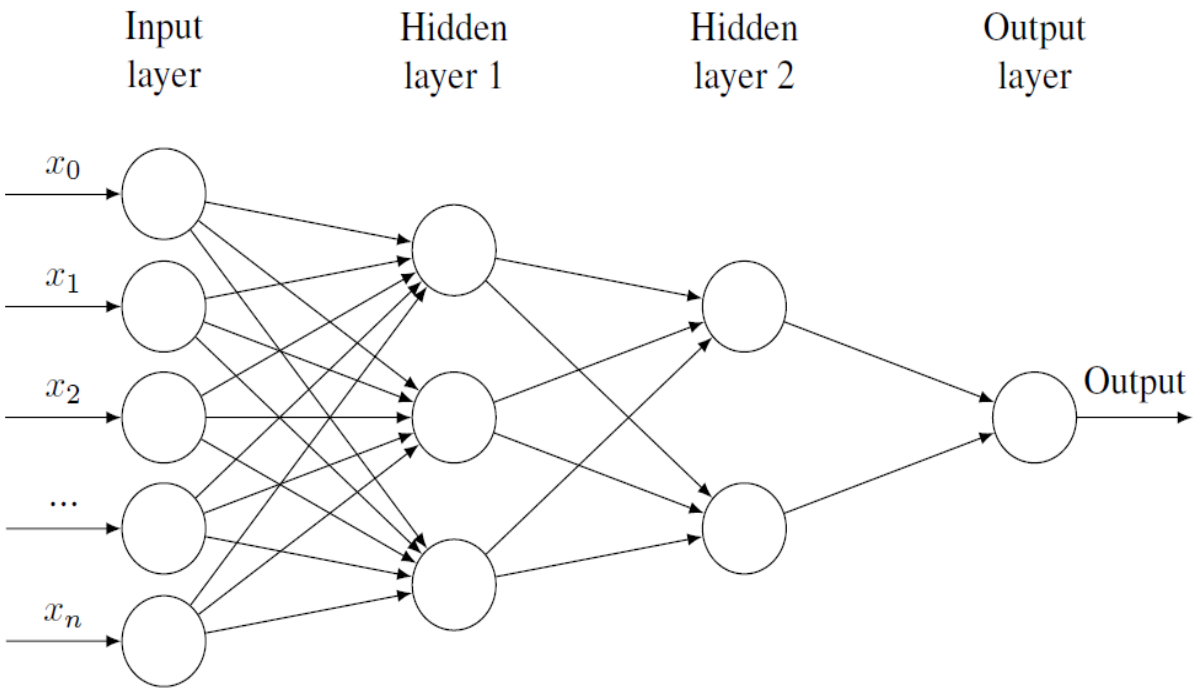
- Networks which allows signals to travel in both directions using loops are called recurrent networks (or, feedback networks).

Number of Nodes

- The number of input nodes is predetermined by the number of features in the input data
- The number of output nodes is predetermined by the number of outcomes to be modeled or the number of class levels in the outcome.
- The number of hidden nodes is decided by the user to decide prior to training the model.
- The appropriate number depends on the number of input nodes, the amount of training data, the amount of noisy data, and the complexity of the learning task, among many other factors.



(a) Network with one hidden layer and two output nodes



(b) Network with two hidden layers

Training algorithm

- Two algorithms are used for learning a single perceptron
  - Perceptron rule
    - used when the training data set is linearly separable
  - Delta rule.
    - Used when the training data set is not linearly separable
- The algorithm which is now commonly used to train an ANN is known as backpropagation.

- cost function
  - a function that measures how well the algorithm maps the target function that it is trying to guess or a function that determines how well the algorithm performs in an optimization problem.
  - loss function, the objective function, the scoring function, or the error function.

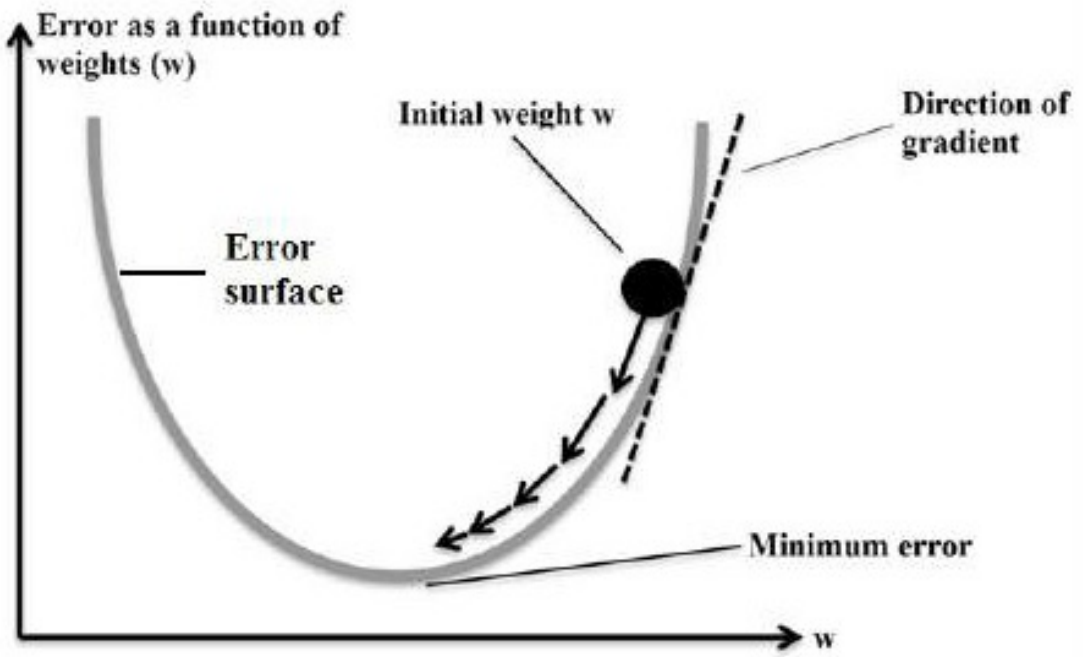
Backpropagation

- Initially the weights are assigned at random.
- Then the algorithm iterates through many cycles of two processes until a stopping criterion is reached. Each cycle is known as an epoch.
- Each epoch includes:
  - A forward phase in which the neurons are activated in sequence from the input layer to the output layer, applying each neuron's weights and activation function along the way.
  - Upon reaching the final layer, an output signal is produced.

- A backward phase in which the network's output signal resulting from the forward phase is compared to the true target value in the training data.
- The difference between the network's output signal and the true value results in an error that is propagated backwards in the network to modify the connection weights between neurons and reduce future errors.

Gradient Descent Method

- The technique used to determine how much a weight should be changed
- At every stage of the computation, the error is a function of the weights



- The sum of squares of the differences between the predicted and actual values of y can be taken as a cost function for the algorithm.

$$\mathrm{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

- The mean of the sum of squares of the differences between the predicted and actual values of y, denoted by MSE

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

- The aim is to minimize MSE

Minimize $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

Gradient Descent

- Cost function can be plotted as a function of parameter estimates i.e. parameter range of hypothesis function and the cost resulting from selecting a particular set of parameters.
- Move downward towards pits in the graph, to find the minimum value.
- Gradient Descent step-downs the cost function in the direction of the steepest descent.
- The size of each step is determined by parameter **α** known as **Learning Rate**.

Advantages

- Multilayer Perceptrons are widely used to solve problems requiring supervised learning
- It can be used to solve complex nonlinear problems.
- It handles large amounts of input data well.
- Makes quick predictions after training.
- The same accuracy ratio can be achieved even with smaller samples.