Classification Algorithms

Why Classify?

To Explain (Profile)

Explaining in the classification world is called Profiling

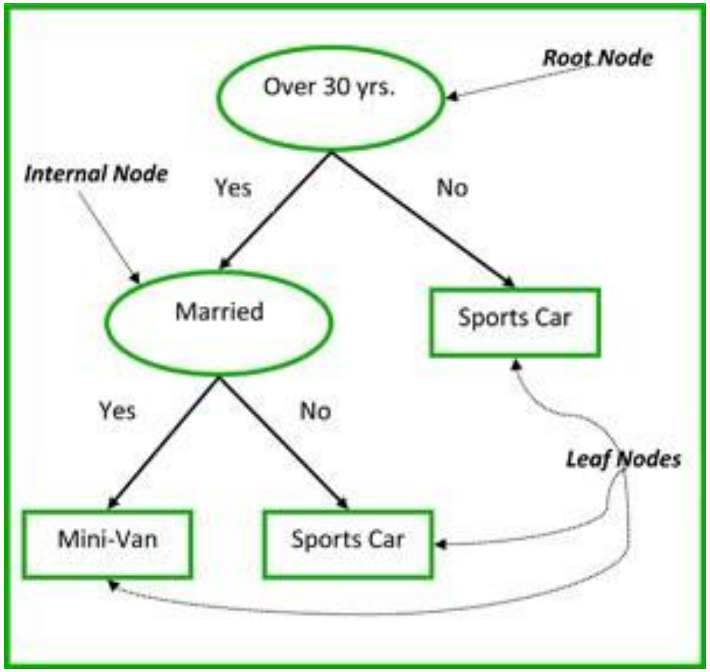
Or

To Predict (Classify)

Predicting the class of new records is called Classifying

Decision Tree

- Supervised Learning Algorithm
- A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails)
- each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes)
- The paths from root to leaf represent classification rules.



CART

- Classification and Regression Decision Trees, a recursive partitioning method

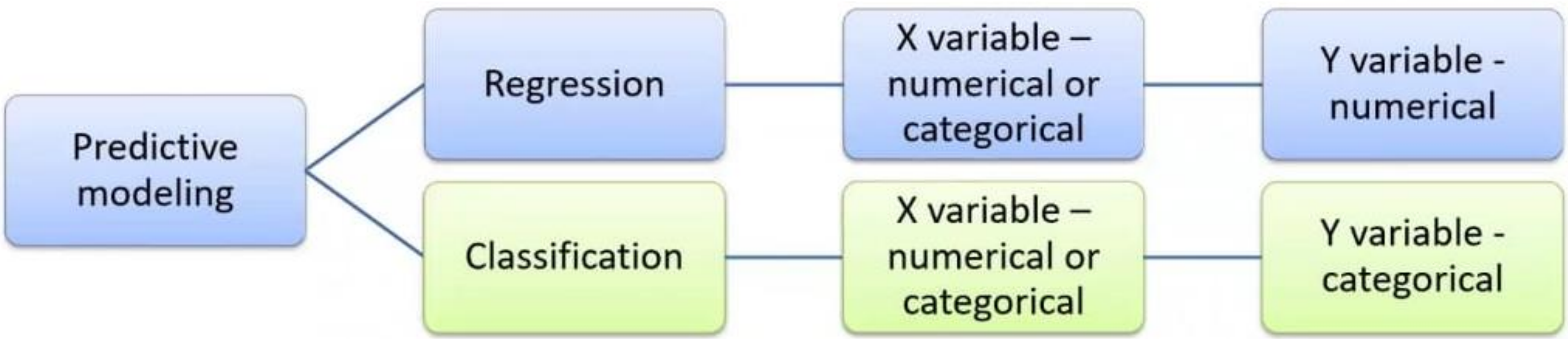
- Root node represents a single input variable (x) and a split point is made on that variable.

- The leaf nodes of the tree contain an output variable (y) which is used to make a classification/prediction

- ***Classification Trees:*** the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall into.

- ***Regression Trees:*** the target variable is continuous and tree is used to predict it's value.

Classification vs Regression



CART | Splitting Criteria

- CART uses the Gini Index as measure of impurity
- Chooses best variable for splitting

- Gini of a Node

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: p(j | t) is the relative frequency of class j at node t).

- Gini of Split Node is computed as Weighted Avg Gini of each Node at Split Node level

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

ni = number of records at child i
n = Total number of records in parent node

- Gini Gain = Gini(t) – Gini(split)

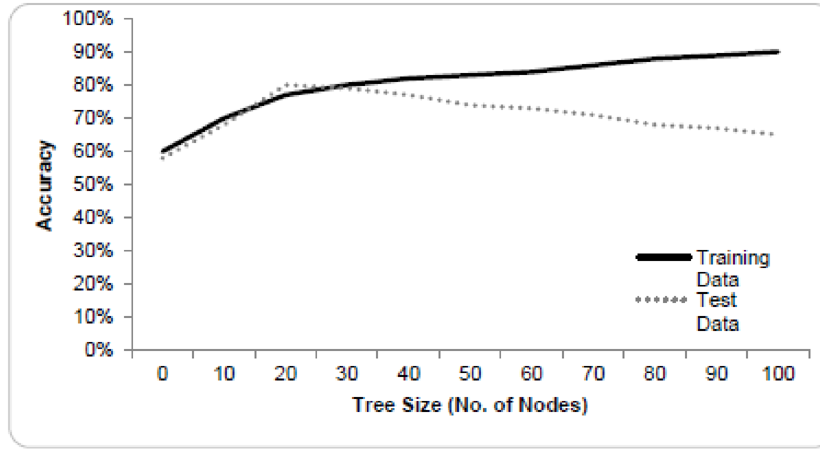
Variable Importance

- Variables ranked from most important down to least important.

- Variable importance is calculated by the sum of the decrease in error when split by a variable.

Concepts | Over-fitting

- If you grow the tree too long you will run the risk of over-fitting
- Classification model may not work well on unseen data



How do we avoid Over-fitting?

Stopping Rule: don't expand a node if the impurity reduction of the best split is below some threshold

Pruning: grow a very large tree and merge back nodes

Pruning

- Pruning is a process of removing the parts of the tree which adds very little to the classification power of the tree.

- removes leaves and branches to improve the performance of the decision tree when moving from the Training Set (where the classification is known) to real-world applications (where the classification is unknown).

- Pruning usually results in
  - reducing size of tree, avoids unnecessary complexity,
  - to avoid overfitting of the data sets when classifying new data.

Random Forest

Drawbacks in Decision Trees

- High Variance - The model gets unstable with a very small variation in data.

- High probability of Overfitting

- Accuracy in a single tree might be less

Ensemble Models

- Principle: Group of weak learners are combined to a strong learner, increasing accuracy of the model

- multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results.

- Techniques: Bagging and Boosting

Bagging

- Bootstrap Aggregating
- Bootstrapping - random sampling with replacement
- reduces overfitting (variance of a Decision Tree)

Bagging Algorithm

- Create several subsets of data from training sample chosen randomly with replacement.
- By sampling with replacement, some observations may be repeated in each subset.
- Now, each collection of subset data is used to train their decision trees.
- As a result, we end up with an ensemble of different trees (models).
- each model runs independent and parallel, and all outputs are aggregated
- Average of all the predictions from different trees are used which is more robust than a single decision tree. (Mode for classification)

Random Forest

- Ensemble Technique

- Random Forest is an extension over bagging.

- In addition to taking the random subset of data, it also takes the random selection of features rather than using all features to grow trees.

- When you have many random trees, it's called Random Forest

- Help reduce over-fitting (separate pruning not required)
  – Note: there is possibility of high over-fitting at individual tree level but averaging removes the overfitting.

- Higher the number of trees in the forest, high the accuracy results.

RF Algorithm

1. Suppose there are N observations and M features in training data set. First, a sample from training data set is taken randomly with replacement.

2. A subset (m) of M features are selected randomly and whichever feature gives the best split is used to split the node iteratively. For the next split, again random m features are chosen and split is made. At each split in a decision tree, m features are chosen.

3. Repeat Step 2 and the tree is grown to the largest.

4. Build forest by repeating steps 1 to 3 for “n” samples to create “n” number of trees.

Neural Network

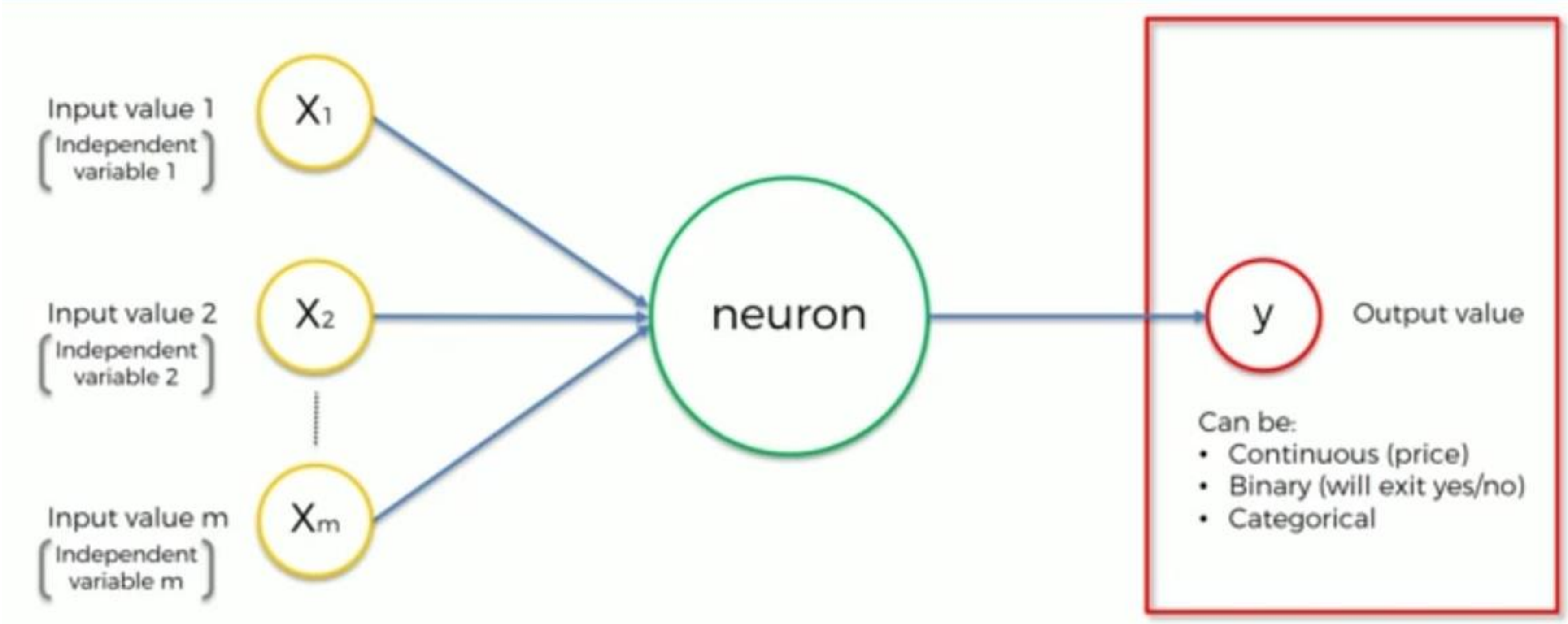
Neural Network Architecture

- Made of layers with many interconnected nodes(neurons)

- There are three main layers,
  - Input Layer
  - Hidden Layer
  - Output Layer

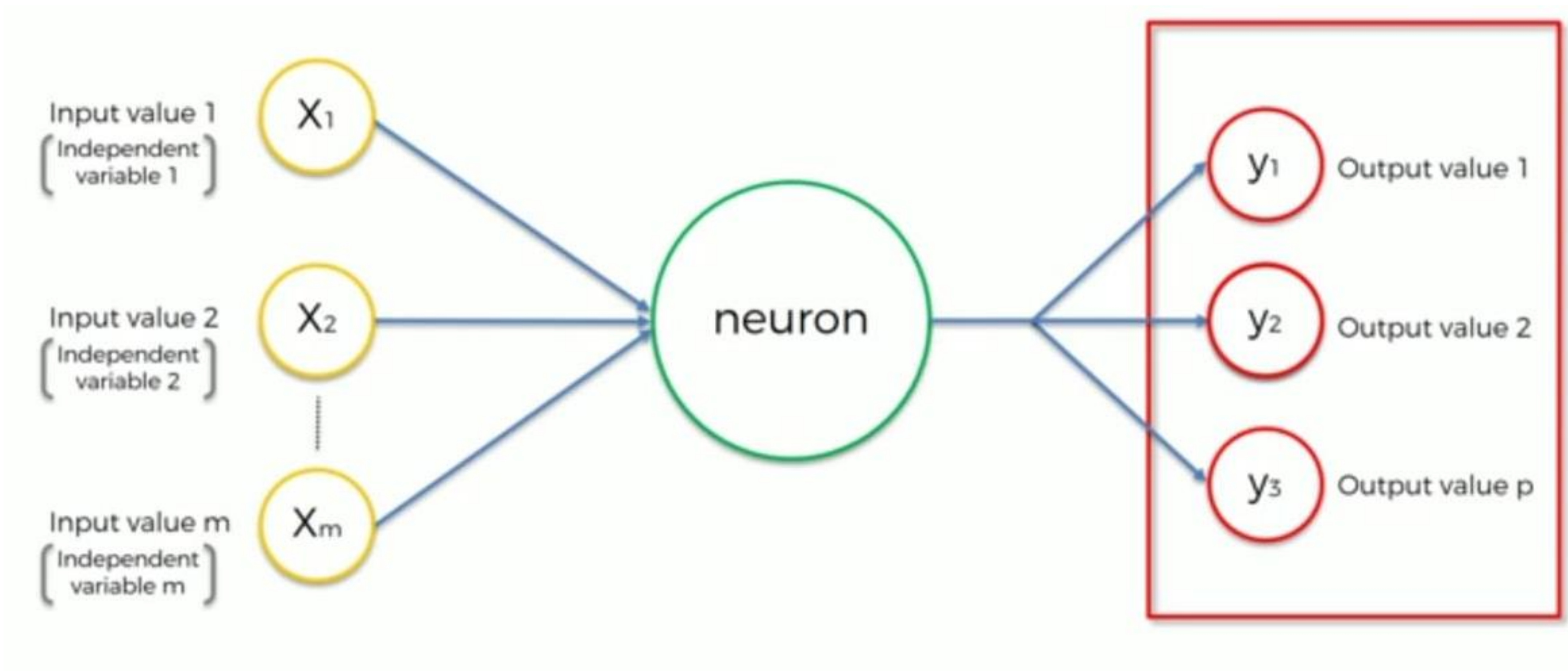
- Hidden Layer can be one or more

Basic structure of ANN

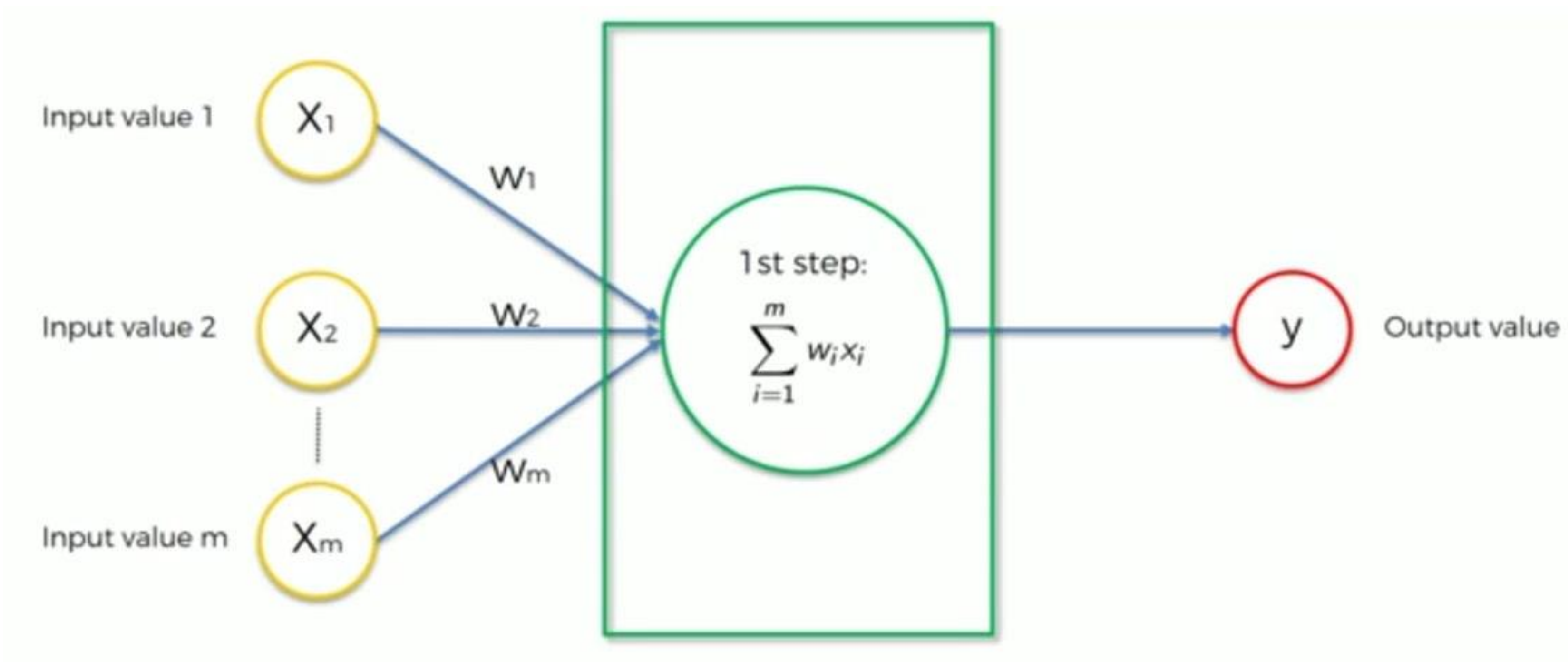


Basic structure of ANN

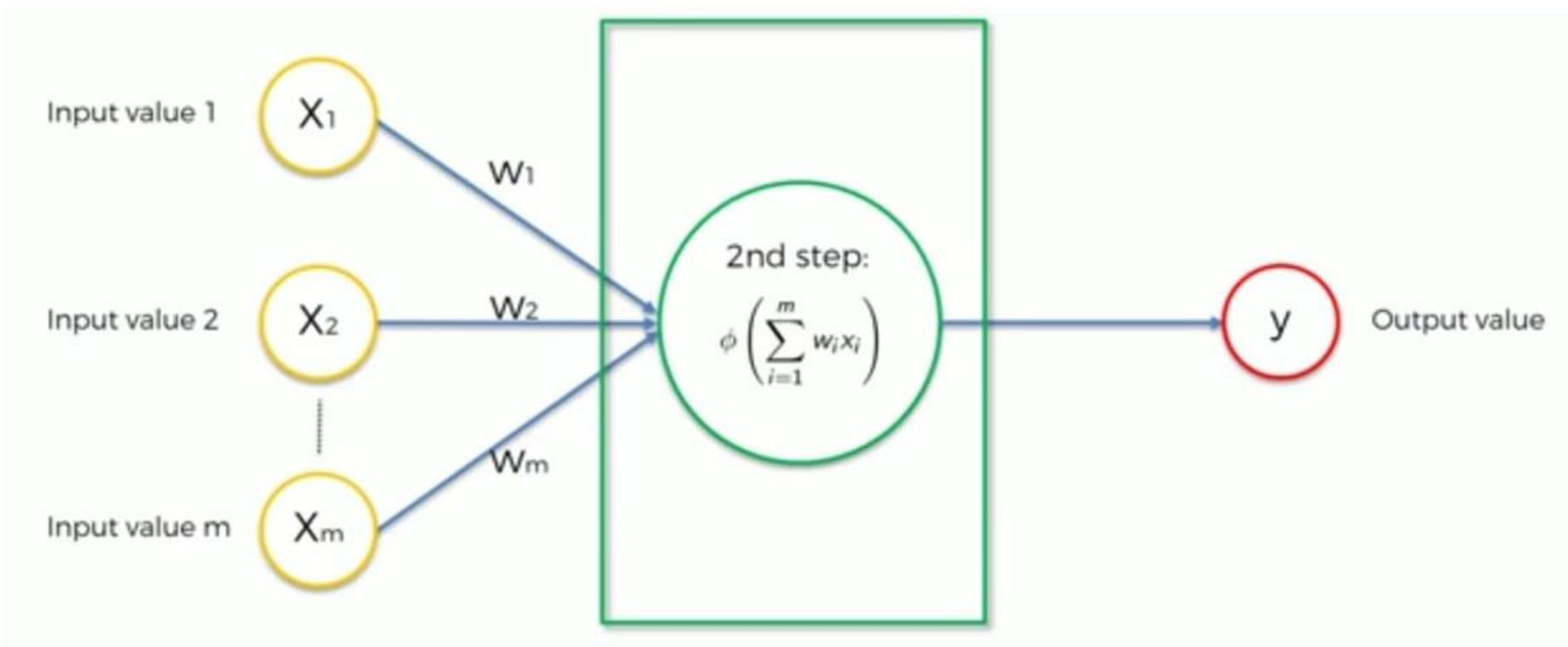
When Y is Categorical,



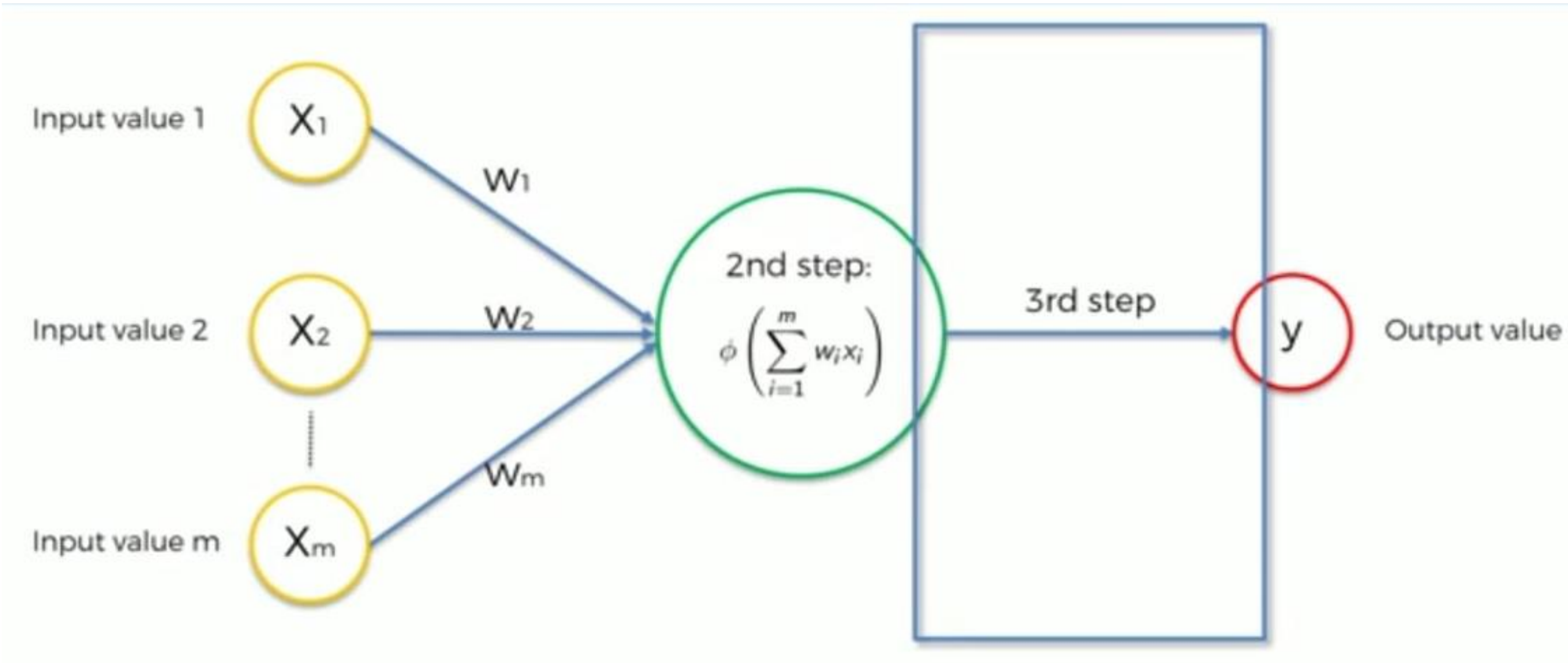
Forward Propagation



Forward Propagation

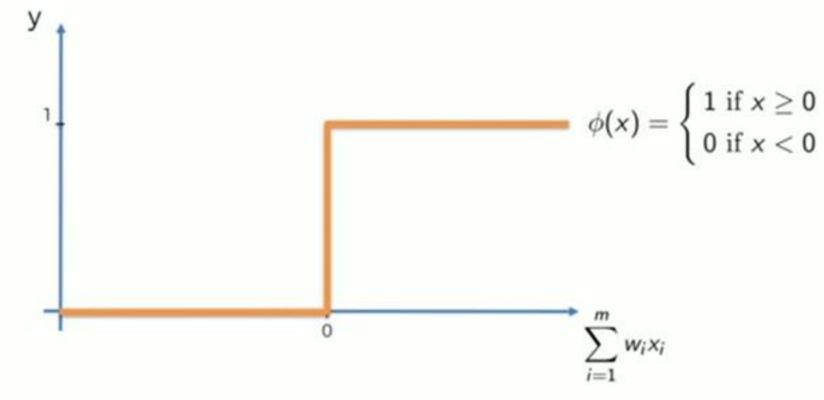


Forward Propagation

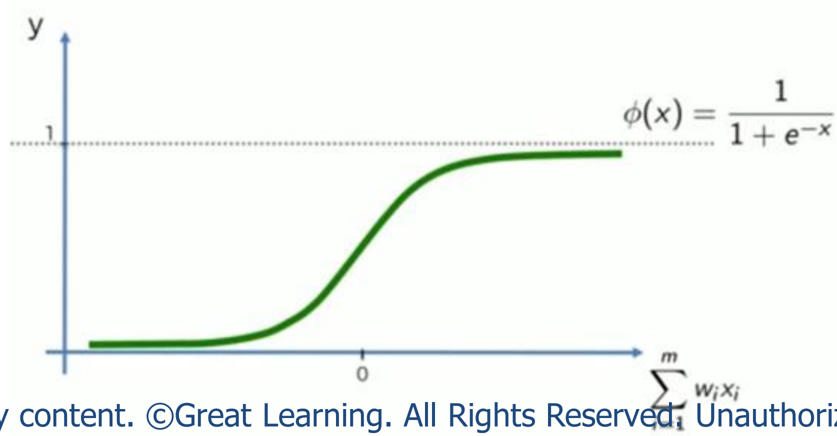


Types of Activation Functions

- Threshold

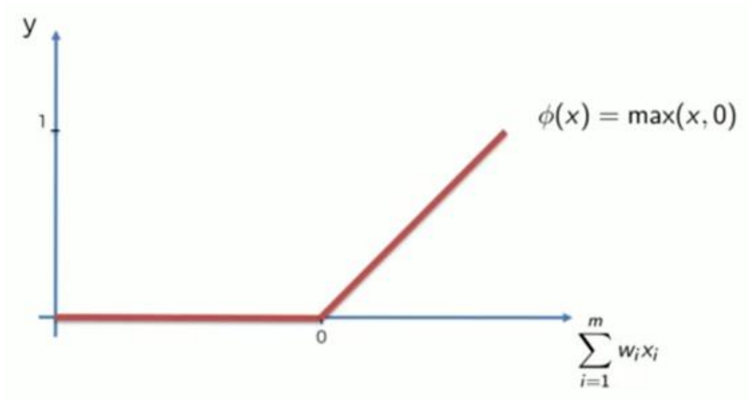


- Sigmoid

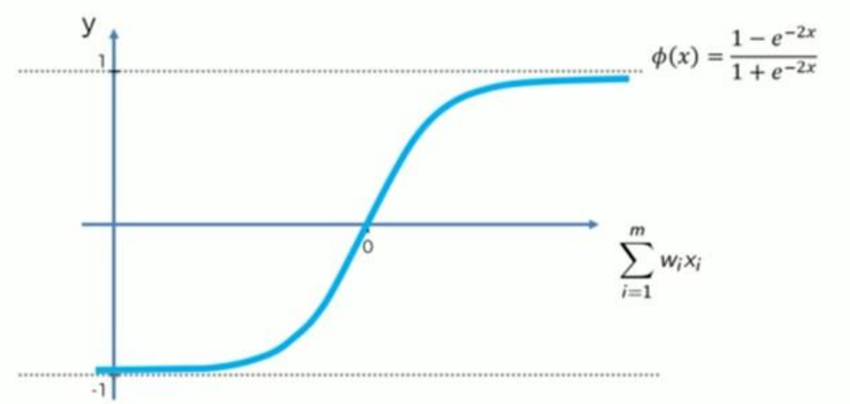


Activation Functions

- Rectifier

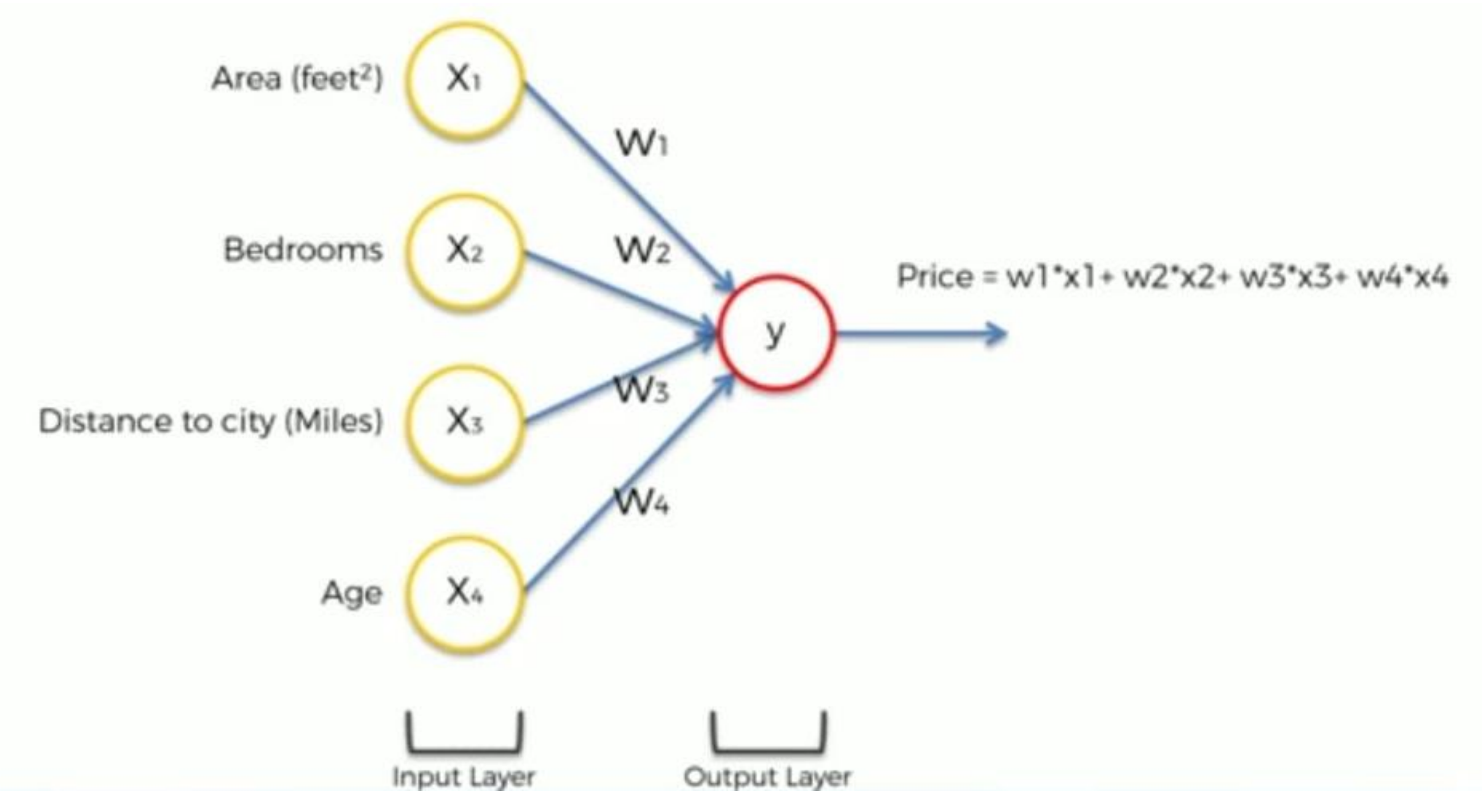


- Hyperbolic Tangent



How do NN work ?

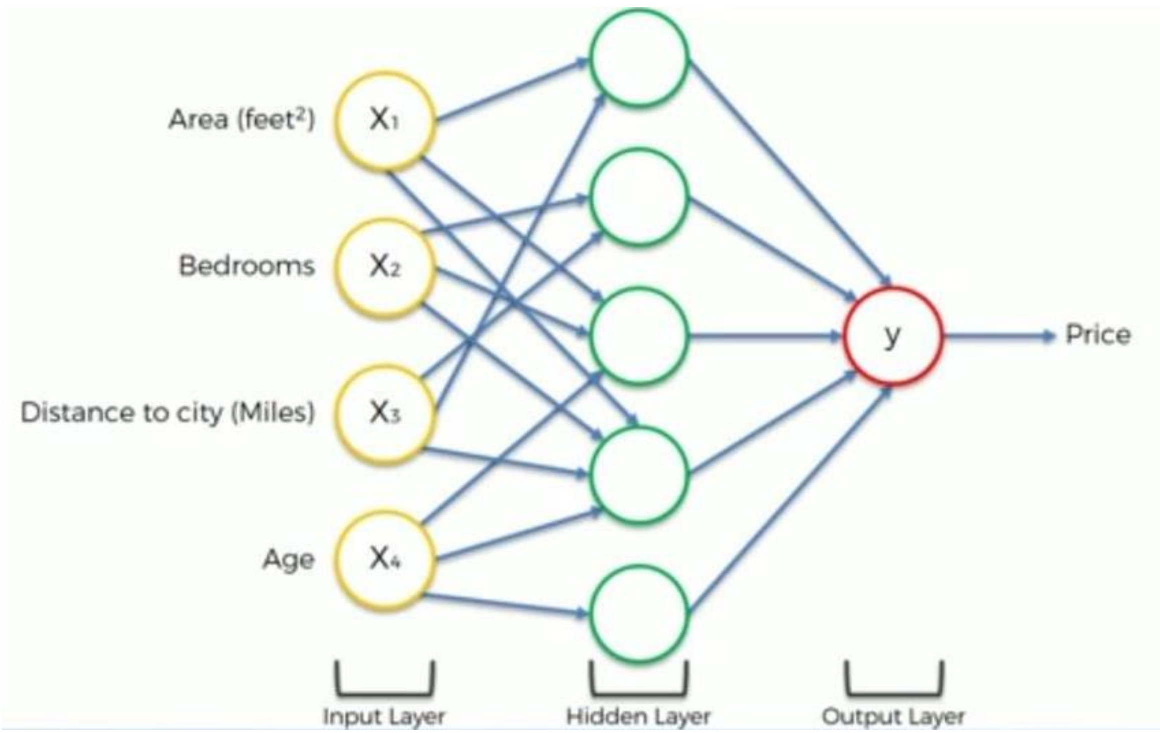
Single Layer Network,



Input nodes process the incoming data exactly asreceived

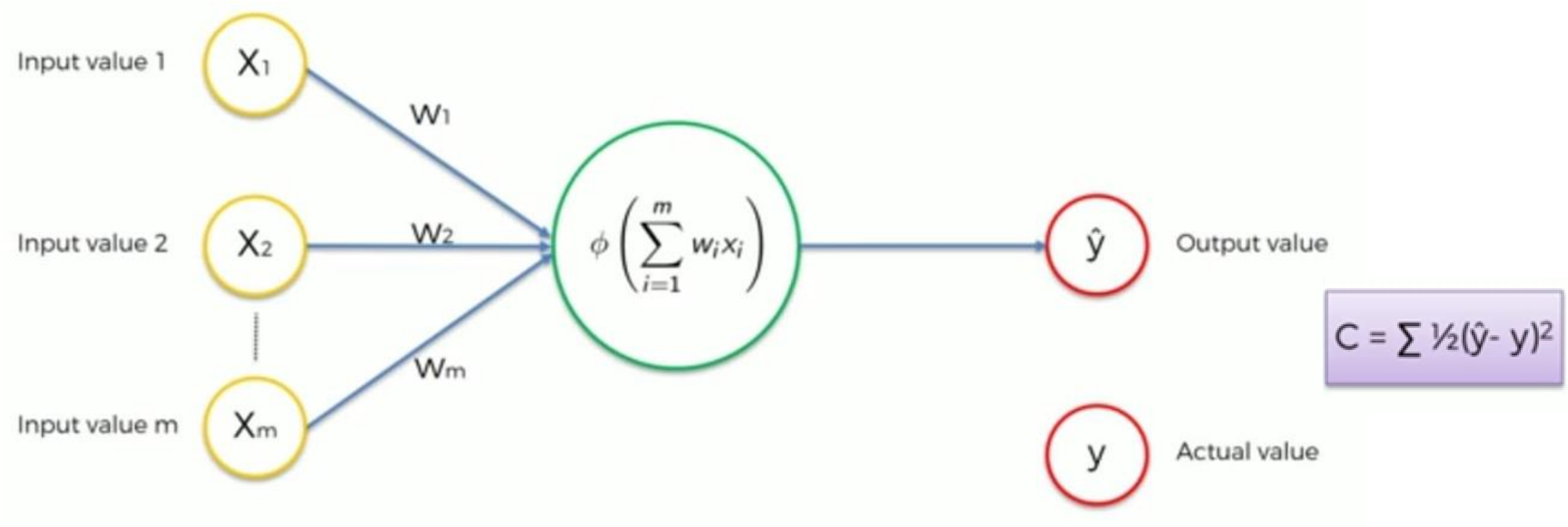
How do NN work ?

Multi Layer Network,



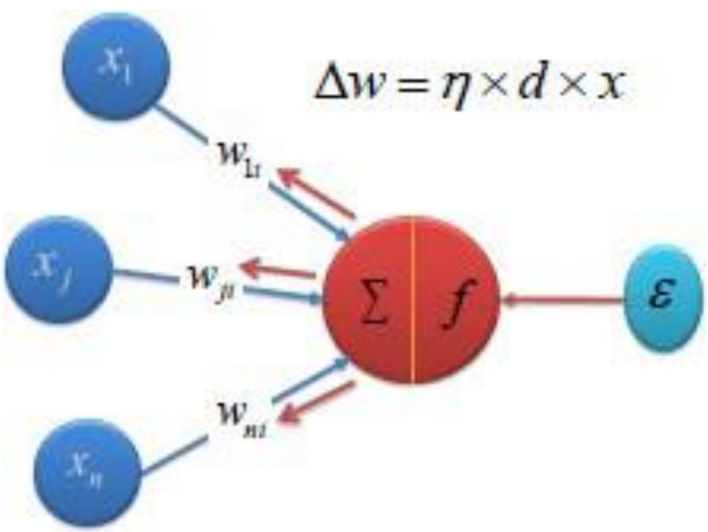
Adds one or more hidden layers that process the signals from the input nodes prior to reaching the output node

How do NN learn ?



Backpropagation

- Backward error propagation or backpropagation



•The output node gives a predicted value

•The difference between predicted value and actual value is the error

•Error propagated backward by apportioning them to each node's weights

•In proportion to the amount of this error the node is responsible for

Fully Connected NN

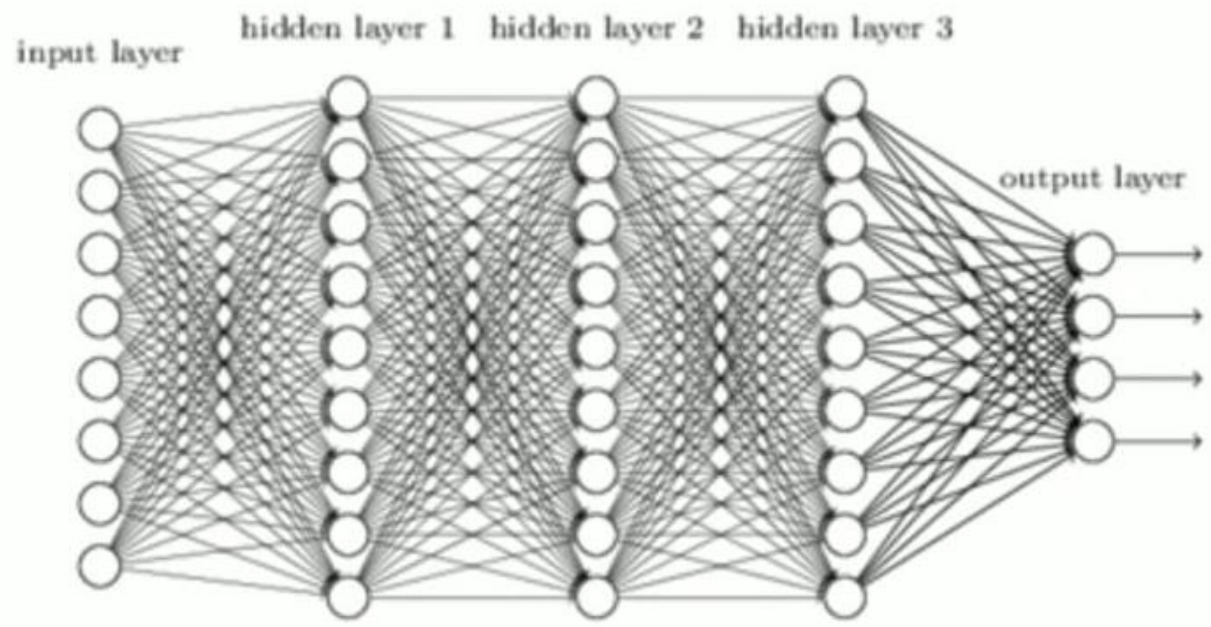
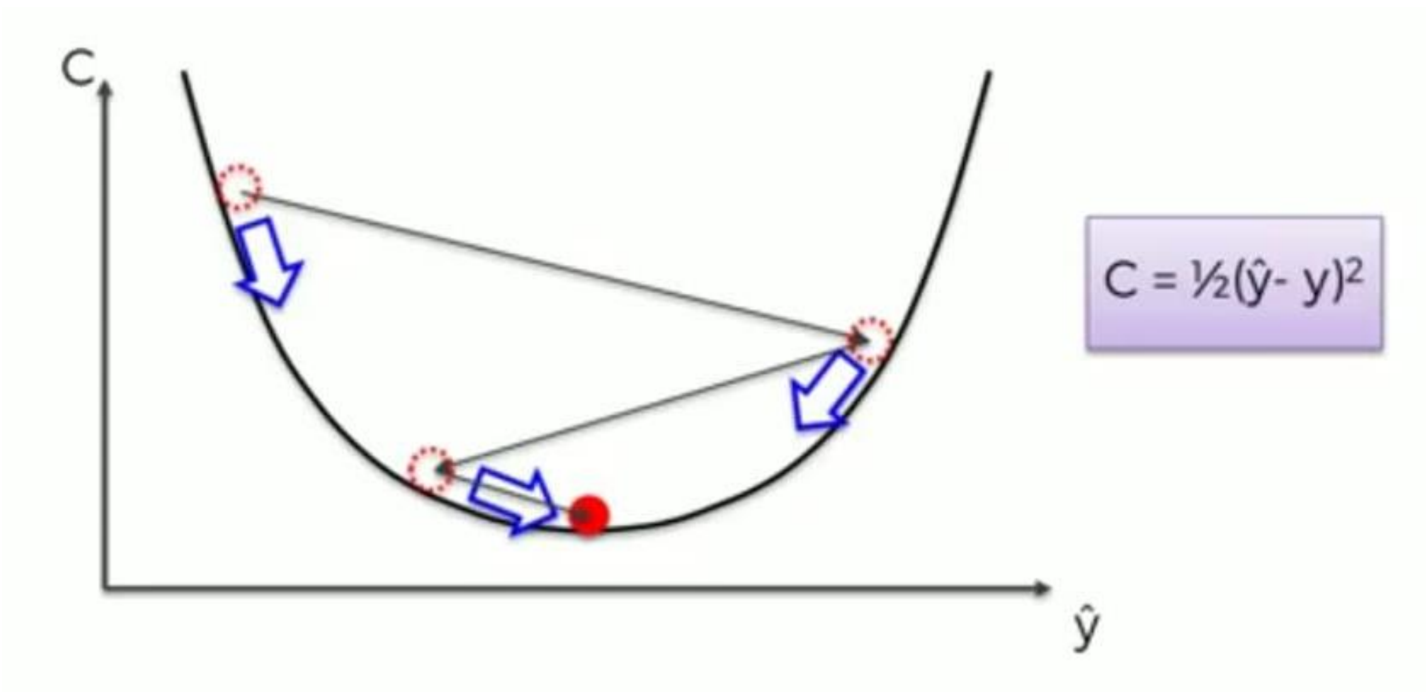


Image Source: neuralnetworksanddeeplearning.com

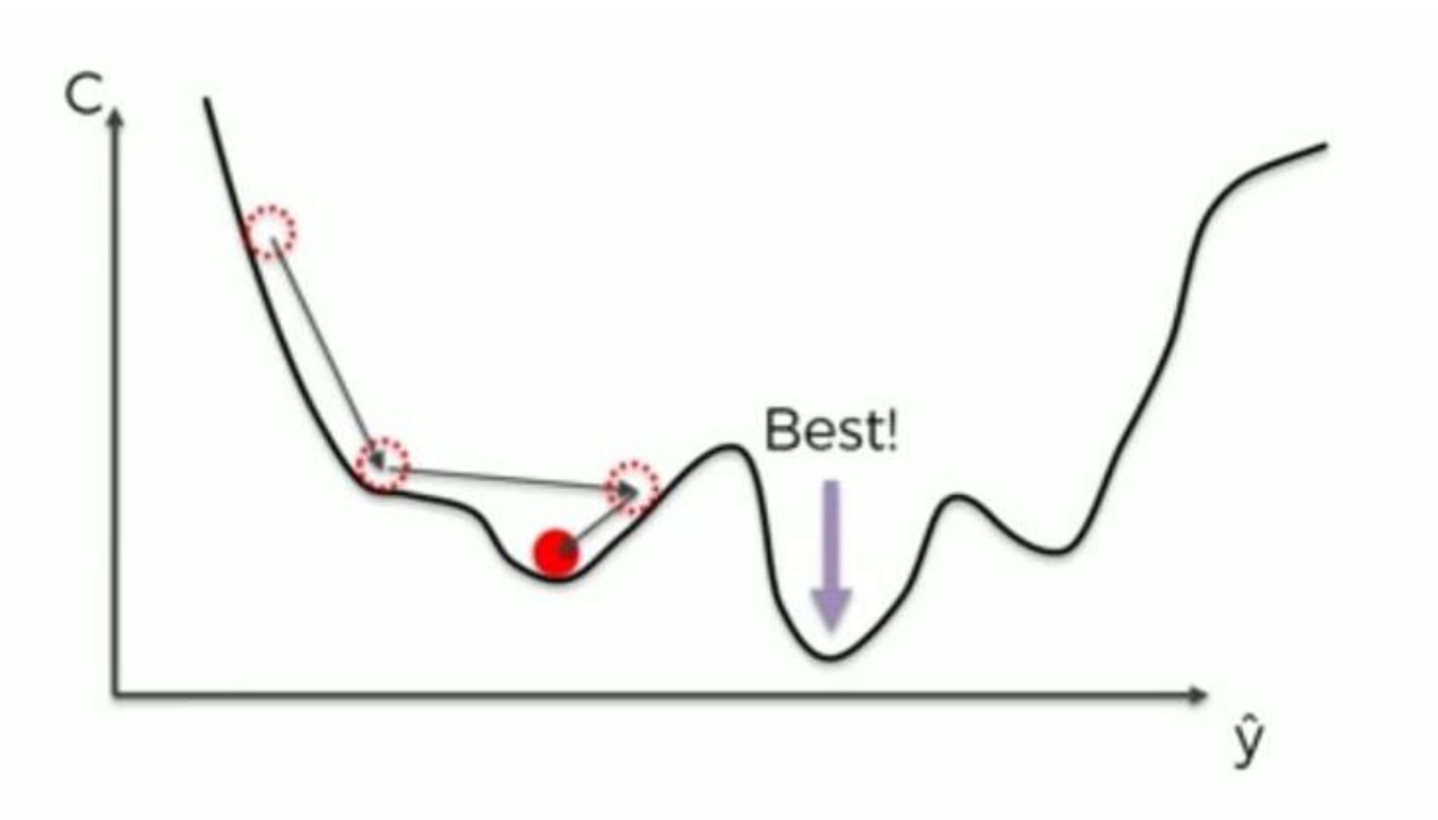
Optimization technique

- Batch Gradient Descent
- Stochastic Gradient Descent
- Mini Batch Gradient Descent

Gradient Descent



Stochastic Gradient Descent



Backpropagation

Reference:

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Steps

1. Randomly initialise the weights to small numbers close to 0
2. Input the first observation in the input layer, each feature in 1 node
3. Forward Propagation: Neurons are activated by the weights and propagates to make the prediction
4. Compare Actual to Predicted and measure error
5. Back Propagation: Weights updated based on the error. Learning rates decide the amount to vary
6. Repeat steps 1 to 5, and update weights after each iteration (SGD), or Repeat steps 1 to 5, and update weights after a batch of observations (GD)
7. When the whole training set passes through the ANN, this is 1 epoch
8. Repeat for more epochs