Introduction to Machine Learning

Supervised Learning:

- Classification
- Regression

Introduction to Machine Learning

Supervised Learning:

- ➢ Classification

  - Used to classify the data using class labels like Yes or No, True or False, Male or Female etc.

  E.g.: Weather Forecasting, Market trends etc.

  Types:

  Random Forest, Decision Trees, Logistic Regression, Support Vector

Introduction to Machine Learning

Supervised Learning:

- ➢ Regression

  - Used to find the relationship between the i/p and o/p variables.

  Types:

  Linear Regression, Non Linear Regression, Regression Trees, Bayesian, Polynomial Non Linear Regression.

Introduction to Machine Learning

Advantages:

- It can predict the output on the basis of prior experiences.
- Contains clear idea about the classes of objects.

Disadvantages:

- Not suitable complex problems.
- It cannot predict the correct output correctly, if the test data is different from the training dataset.
- Training require lots of computation times.

Introduction to Machine Learning

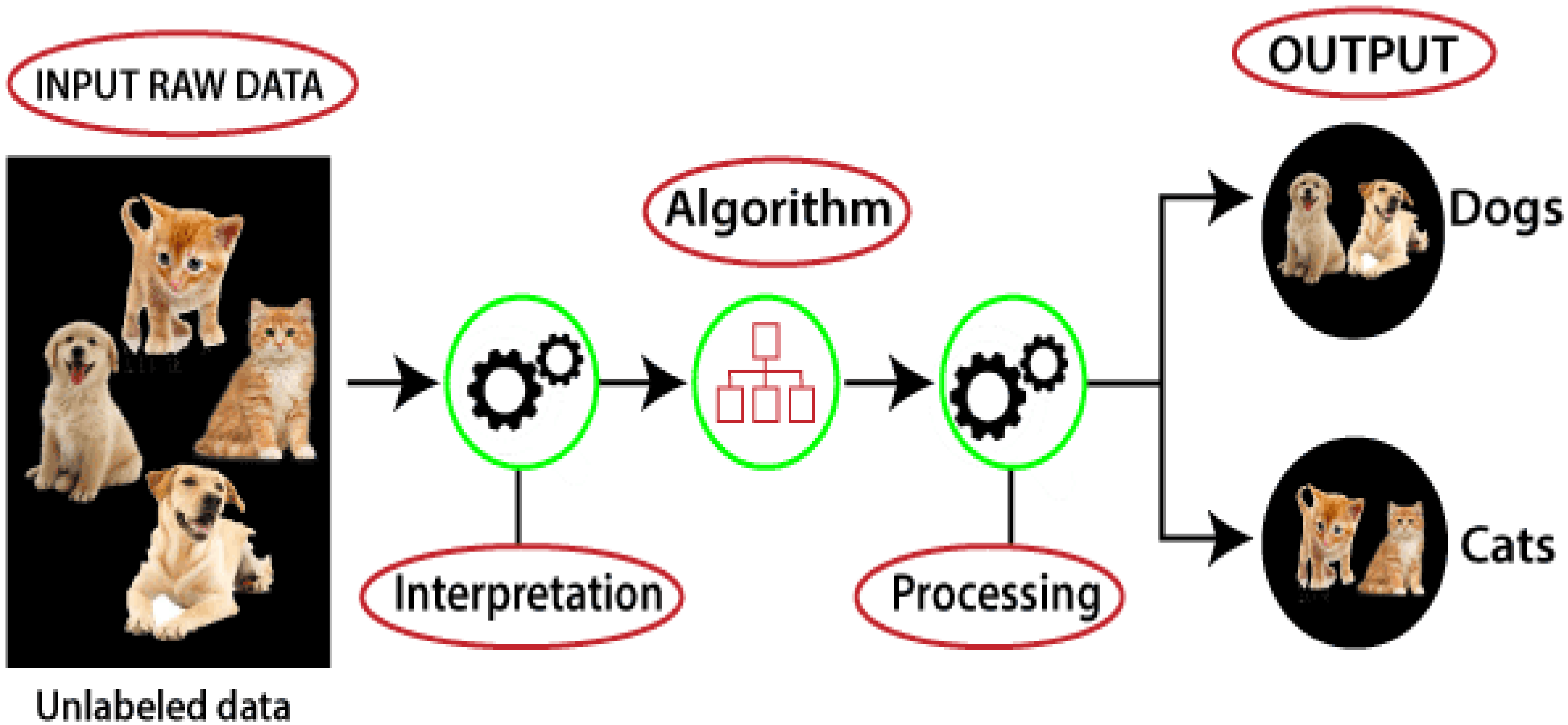
Unsupervised Learning:

- ✓ Unlabelled data.
- ✓ Tries to find useful insights from the huge amount of data.
- ✓ Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings.
- ✓ The creation of these hidden structures is what makes unsupervised learning algorithms versatile.
- ✓ can adapt to the data by dynamically changing hidden structures.

Introduction to Machine Learning

Unsupervised Learning:

- ✓ Clustering
- ✓ Association



Introduction to Machine Learning

Advantages:

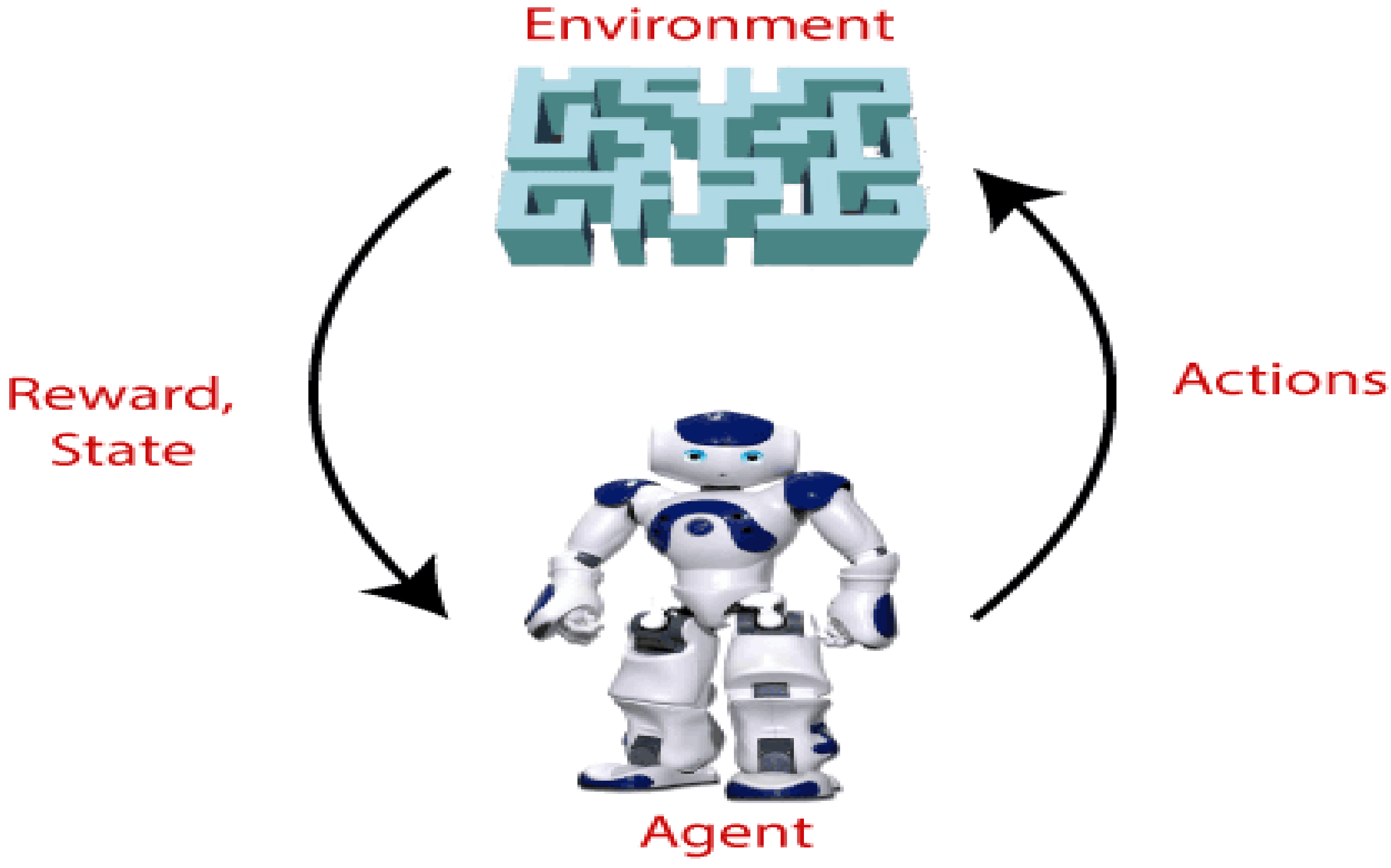
- ✓ Used for more complex tasks.
- ✓ It is preferable one since unlabelled data's are easy to get.

Disadvantages:

- ✓ It does not have corresponding output.
- ✓ Less accurate as input data is not labelled, and algorithms do not know the exact output in advance.

Introduction to Machine Learning

Reinforcement Learning:



Introduction to Machine Learning

Reinforcement Learning:

- ✓ Takes inspiration from how human beings learn from data in their lives.
- ✓ It improves upon itself and learns from new situations using a trial-and-error method.
- ✓ Favourable outputs are encouraged or ‘reinforced’, and non-favourable outputs are discouraged or ‘punished’.
- ✓ In every iteration of the algorithm, the output result is verified whether favourable or not.
- ✓ In the finding of correct solution, it reinforces the solution by providing a reward to the algorithm.
- ✓ It is trained to give the best possible solution for the best possible reward.

Introduction to Machine Learning

Types of Reinforcement learning:

1. **Positive Reinforcement -** It impacts positively on the behaviour of the agent and increases the strength of the behaviour.

2. **Negative Reinforcement -** It increases the tendency that the specific behaviour will occur again by avoiding the negative condition.

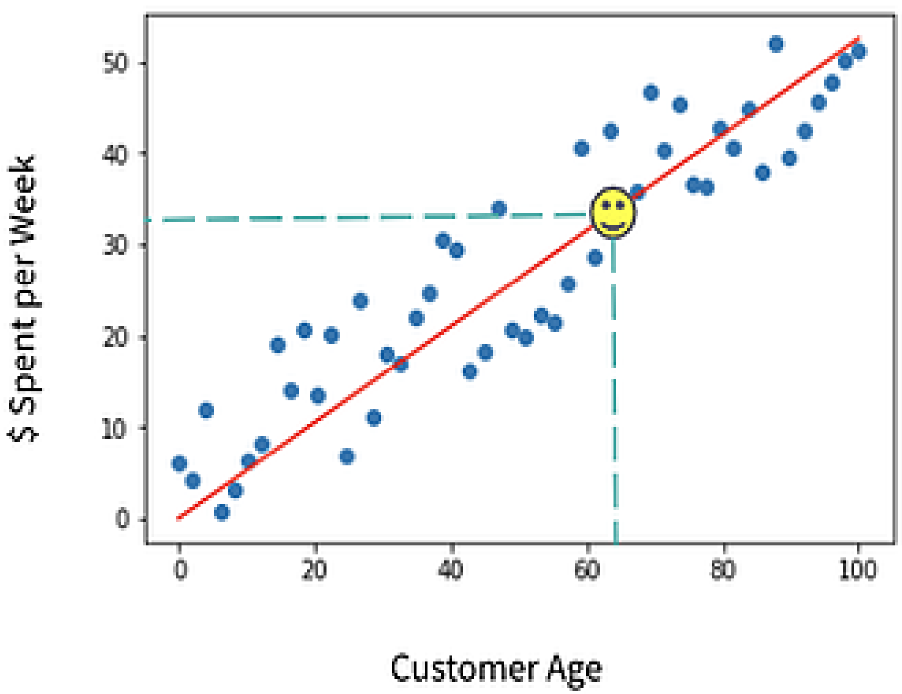
Introduction to Machine Learning

Differences between Supervised Vs Unsupervised Vs Reinforcement Learning

Criteria	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	Learns by using labeled data	Trained using unlabelled data without any guidance	Works on interacting with the environment
Type of data	Labeled data	Unlabeled data	No-predefined data
Type of problems	Regression and Classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra Supervision	No	No
Algorithms	Linear Regression, Logistic Regression, SVM	K-means, C-Means, Apriori	Q-Learning, SARSA
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action
Application	Risk Evaluation, Forecast sales	Recommendation system, Anomaly detection	Self driving cars, Gaming and Healthcare

Linear Models

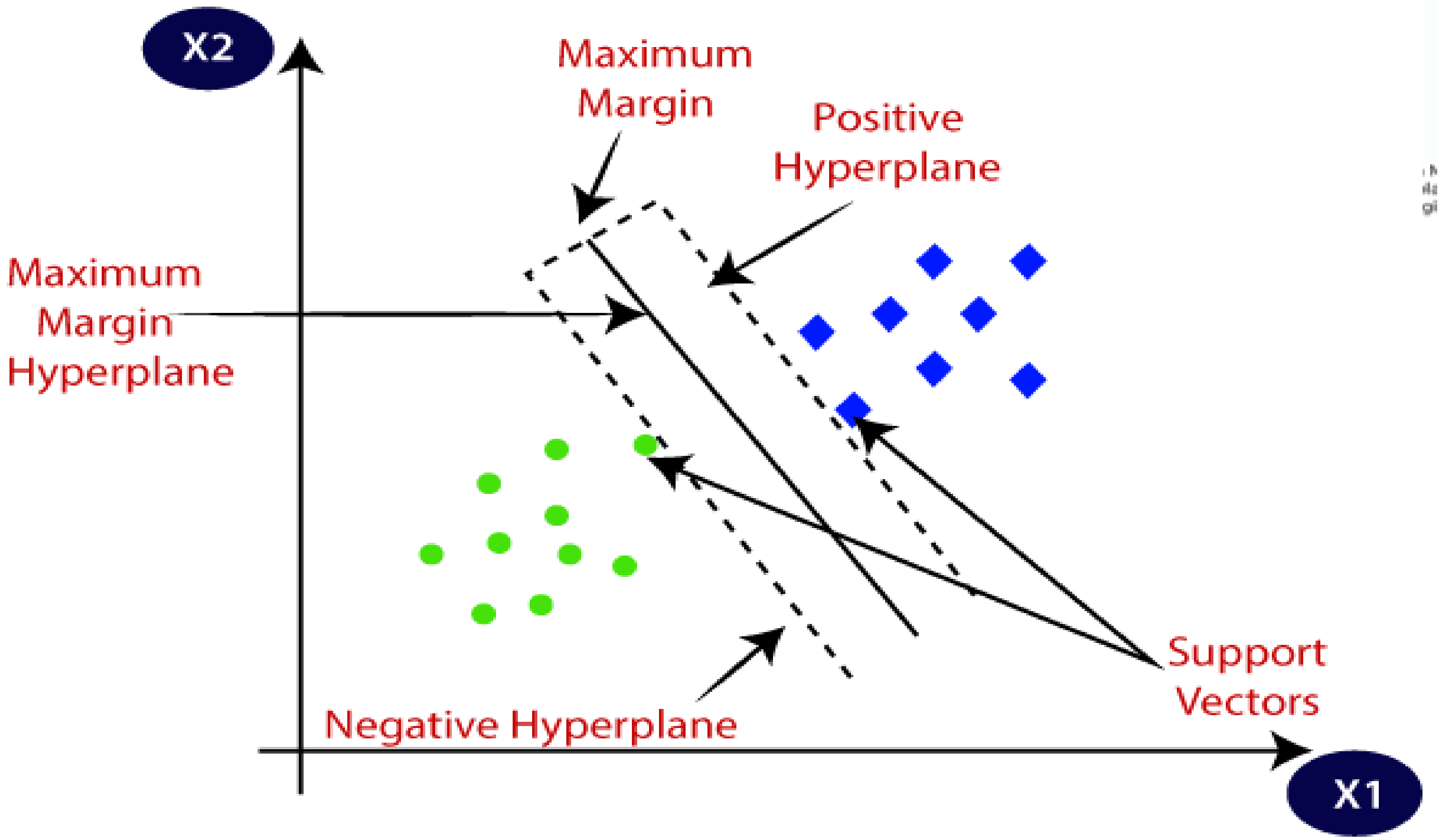
Linear models generate a formula to create a best-fit line to predict unknown values. Linear models are not as predictive as newer algorithm classes, but they can be trained relatively quickly and are generally more straightforward to interpret.



The data points are in blue. The red line is the line of best fit, which the model generated, and captures the direction of those points as best as possible.

Linear Models

Support Vector Machine:



Linear Models

Support Vector Machine:

- ✓ Used for Classification as well as Regression problems.
- ✓ Primarily used for Classification problems in Machine Learning.
- ✓ The goal of the SVM algorithm is to find out the best decision boundary to classify the points.

Ex. Face detection, image classification, text categorization

Linear Models

Hyperplane:

✓ Multiple lines / decision boundaries to segregate the classes in n-dimensional space, but the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

✓ The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

✓ Hyperplane will be created with a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector.

Linear Models

Types of SVM:

1. **Linear SVM** - linearly separable data

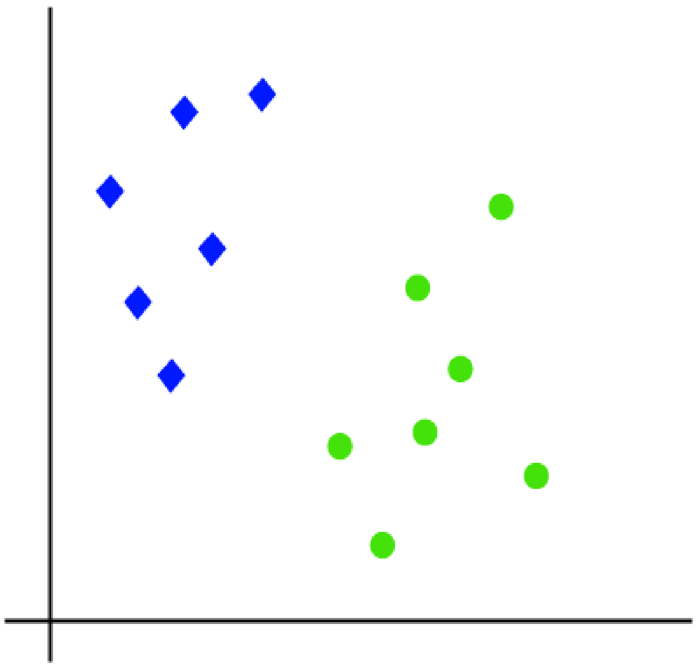
   - if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier used is called as Linear SVM classifier.

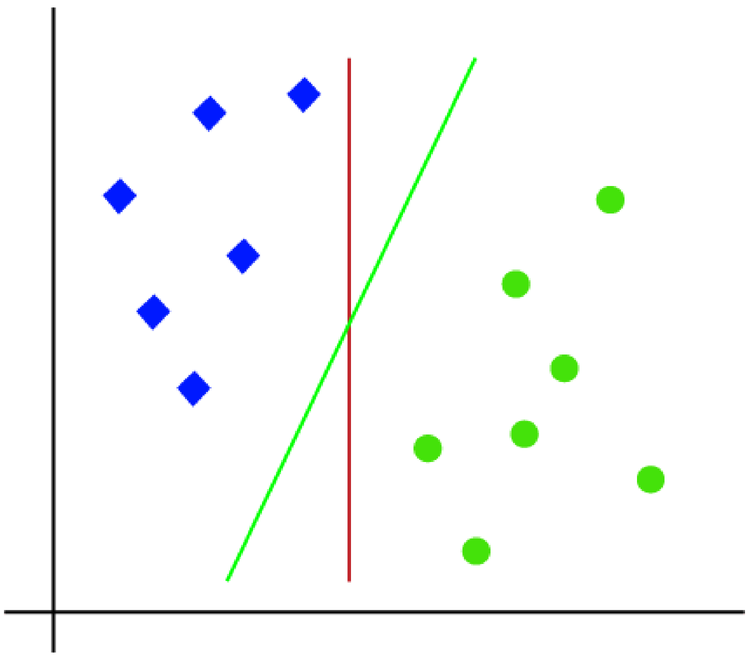
2. **Non-linear SVM** - non-linearly separable data

   - if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Linear Models

Linear SVM:

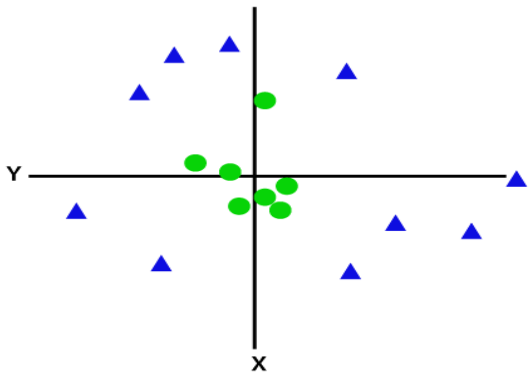




Linear Models

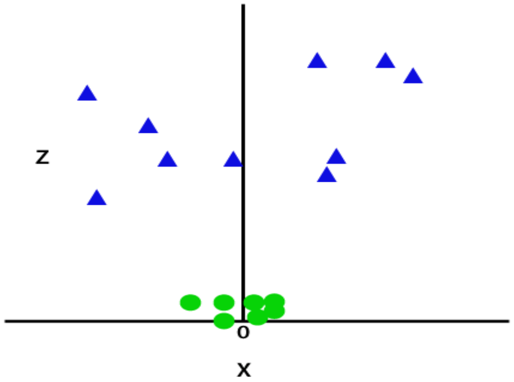
Non-Linear SVM:

1. Cant draw single straight line for non-linear data

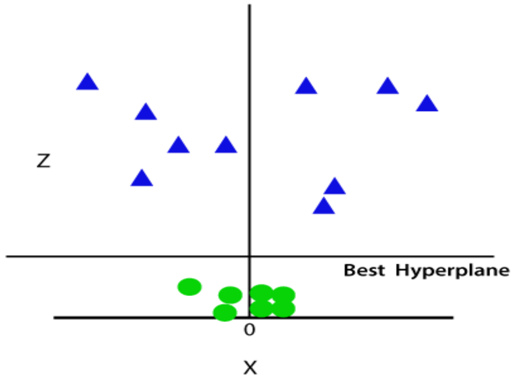


2. One more dimension is added

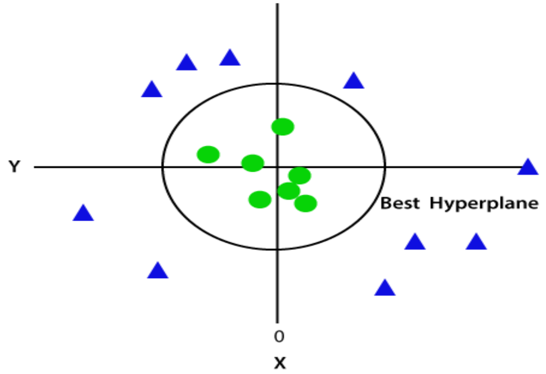
$z = x^2 + y^2$



3. Datasets divided into classes

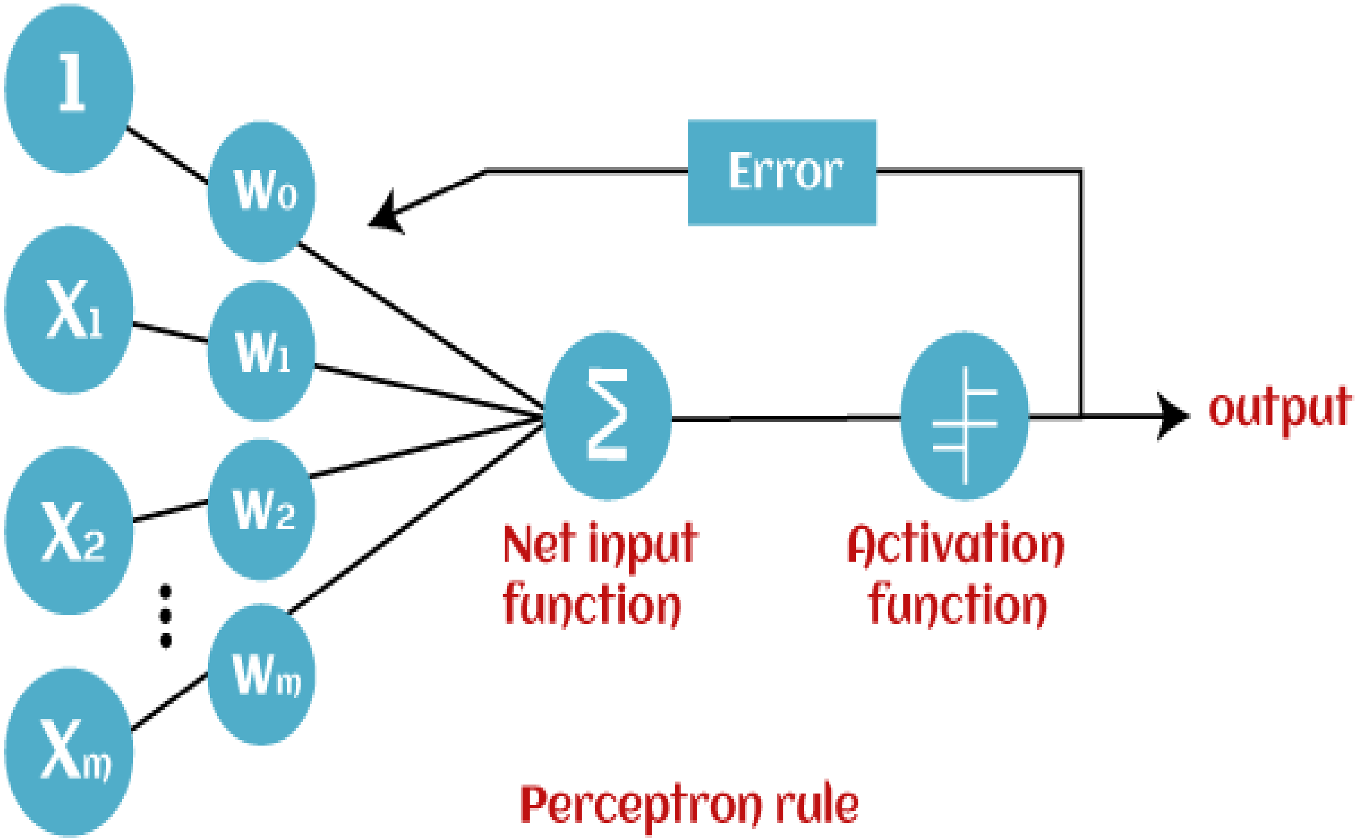


4. Converted into two dimensional one



Linear Models

Perceptron:



Linear Models

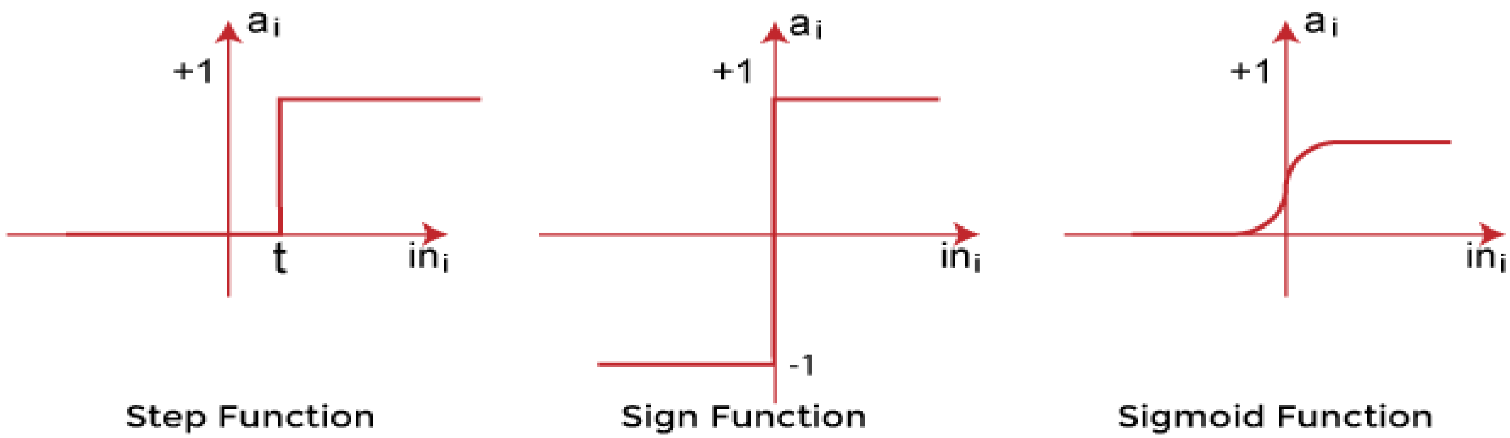
Perceptron:

- ***Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence***.
- One of the best and simplest types of Artificial Neural networks.
- It is a supervised learning algorithm of binary classifiers.
- single-layer neural network with four main parameters, i.e., **input values, weights and Bias, net sum, and an activation function.**

Linear Models

Types of Activation functions:

- Sign function
- Step function, and
- Sigmoid function



Linear Models

Characteristics of Perceptron:

- Supervised learning of binary classifiers.
- Weight coefficient is automatically learned.
- Initially, weights are multiplied with input features, and the decision is made whether the neuron is fired or not.
- The activation function applies a step rule to check whether the weight function is greater than zero.
- Linear decision boundary is drawn, enabling the distinction between the two linearly separable classes +1 and -1.

Linear Models

Limitations of Perceptron Model:

- The output of a perceptron can only be a binary number (0 or 1) due to the hard limit transfer function.
- Perceptron can only be used to classify the linearly separable sets of input vectors. If input vectors are non-linear, it is not easy to classify them properly.

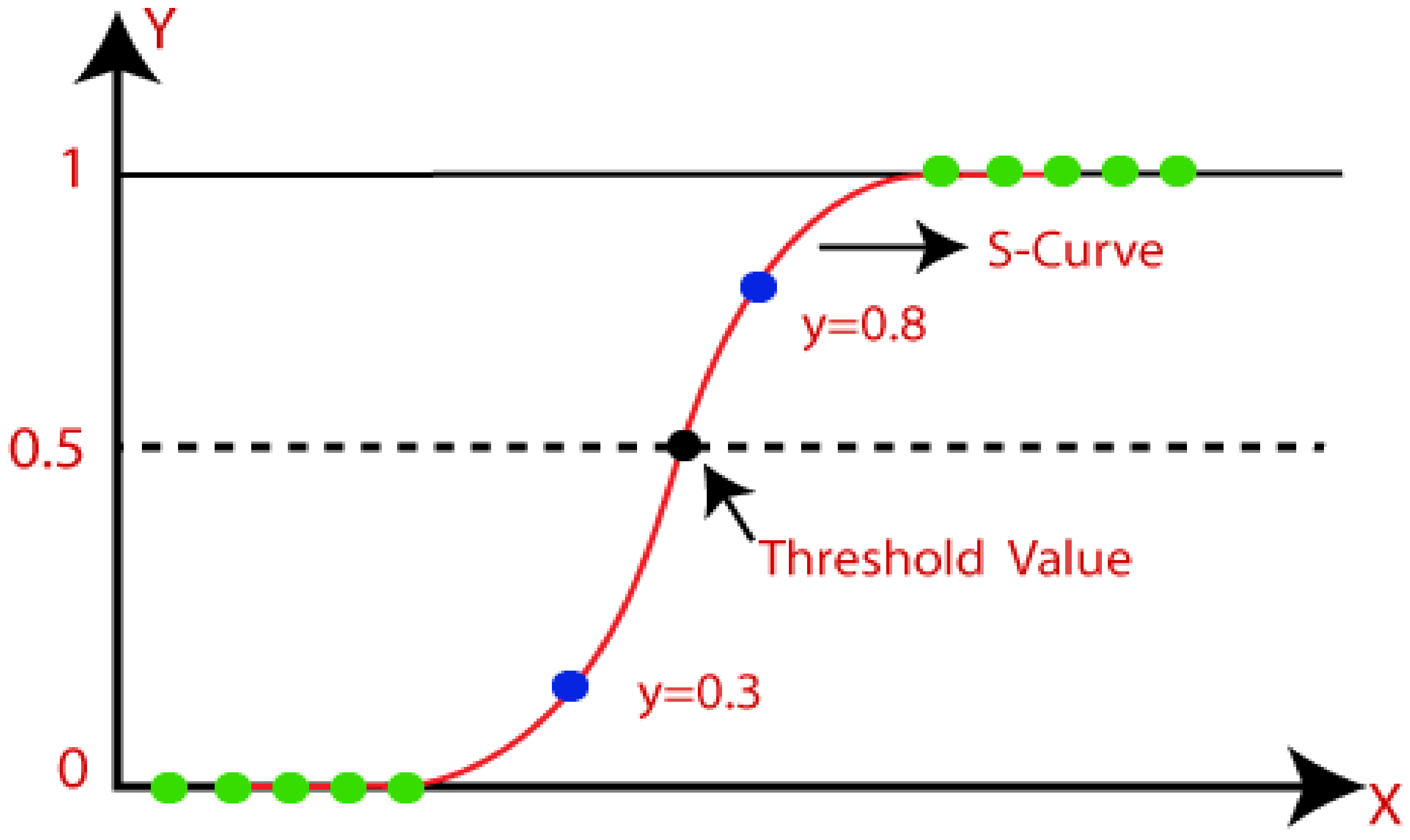
Linear Models

Logistic Regression:

- ✓ It is used for predicting the categorical dependent variable using a given set of independent variables.
- ✓ Outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- ✓ Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- ✓ In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- ✓ It has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Linear Models

Logistic Regression:



Linear Models

Logistic Function (Sigmoid Function):

- It is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- The concept of the threshold value is used such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Linear Models

Steps in Logistic Regression:

1. Data Pre-processing step.
2. Fitting Logistic Regression to the Training set.
3. Predicting the test result.
4. Test accuracy of the result.
5. Visualizing the test set result.

Linear Models

Type of Logistic Regression:

1. **Binomial -** Only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial -** 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
3. **Ordinal -** 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

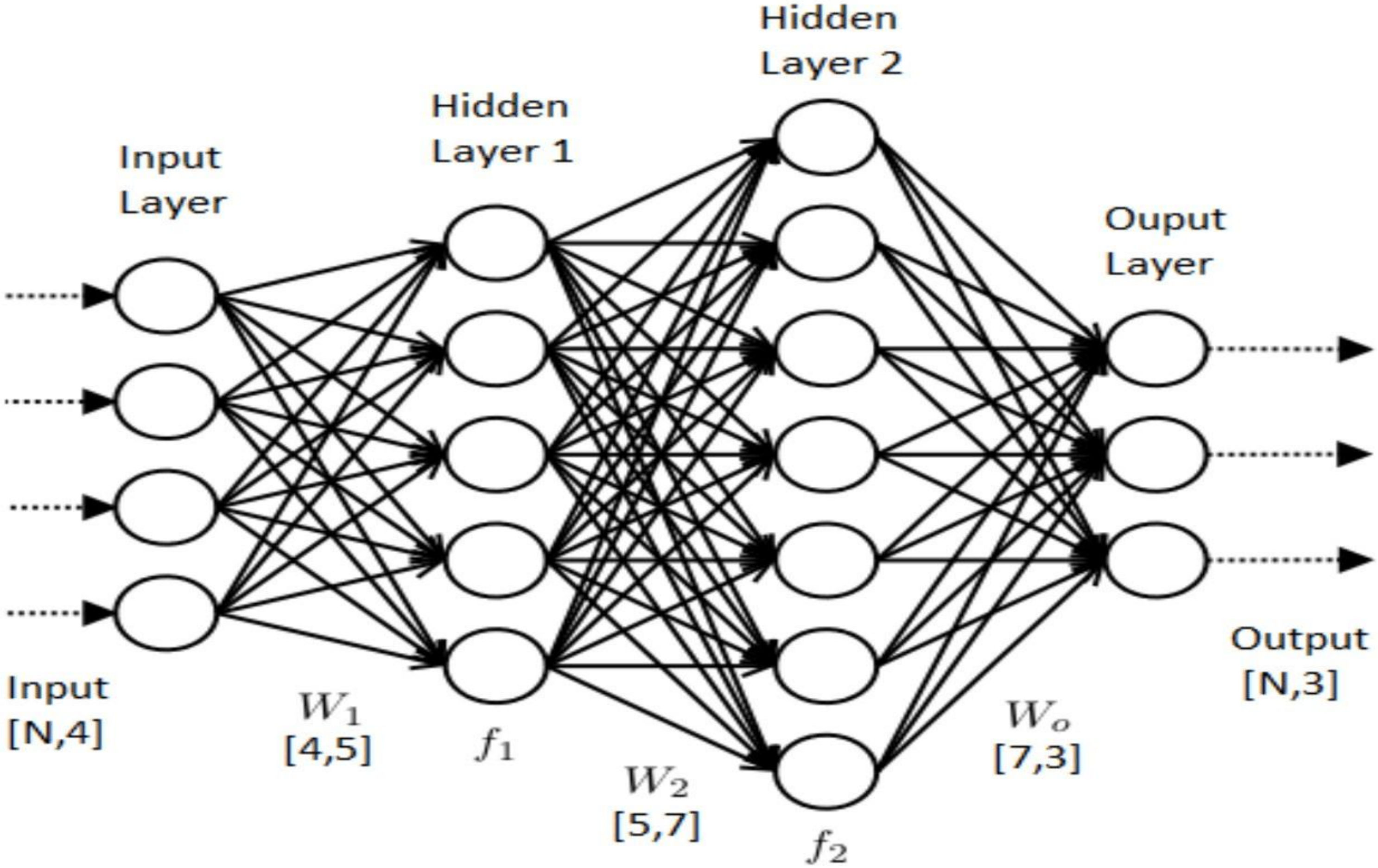
Introduction to Neural Network

A neural network is made of artificial neurons that receive and process input data. Data is passed through the input layer, the hidden layer, and the output layer.

A neural network learns from structured data and exhibits the output. Learning taking place within neural networks can be in three different categories:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Introduction to Neural Network



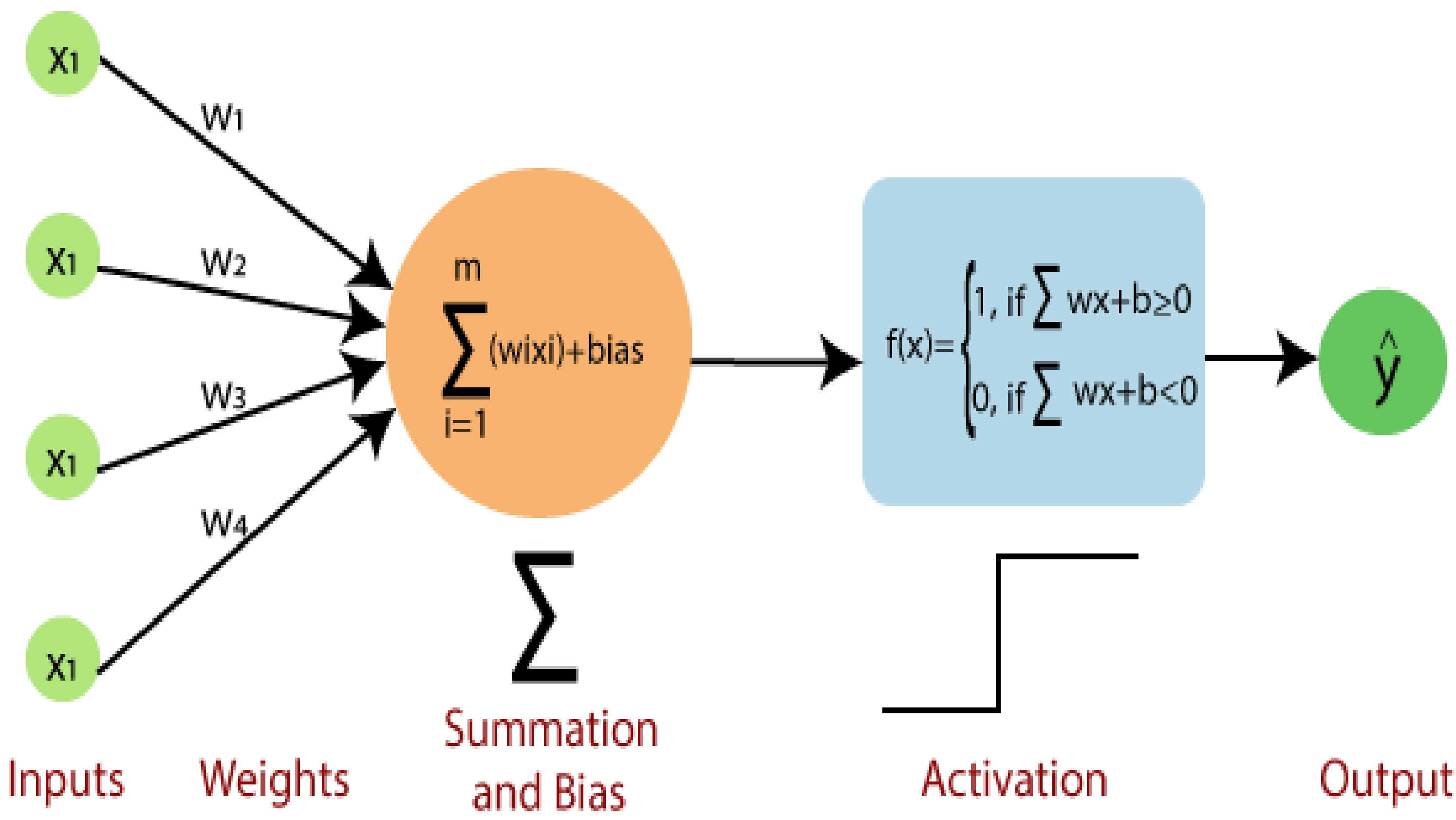
Introduction to Neural Network

How Neural Networks work:

Artificial neurons or perceptron consist of:

- Input
- Weight
- Bias
- Activation Function
- Output

Introduction to Neural Network



Introduction to Neural Network

The neurons receive many inputs and process a single output. Neural networks are comprised of layers of neurons. These layers consist of the following:

- Input layer - receives data represented by a numeric value
- Multiple hidden layers - perform the most computations required by the network
- Output layer - predicts the output

Introduction to Neural Network

Feed Forward Propagation Process:

Feed forward propagation takes place when the hidden layer accepts the input data. Processes it as per the activation function and passes it to the output. The neuron in the output layer with the highest probability projects the result.

1. I/P layer received input data, initialize input with the weights and redirected into the hidden layer.
2. Converts I/P data within Hidden Layers – Taking I/P data, multiplying with it by the weight value , the sum is sent to neuron and then bias is applied to each neuron .
3. Hidden layer transform the I/P data and pass it to the other layer. Finally transmits through the activation function.
4. Activated neuron transfers the information into the output layer.

Introduction to Neural Network

Different Types:

NN's are identified based on mathematical performance and principles to determine the O/P.

1. **Perceptron** – Single Layer Network (I/P & O/P)
2. **Feed Forward NN** – Data moves in a Single Direction
3. **Radial Basis Function (RBF)** – Classify data based on distance of any centered point and interpolation (3 Layers)
4. **Recurrent NN** – Feedback NN
5. **Convolutional NN** – Composed of Convolution, Pooling and Fully connected NN)
6. **Modular NN** – composed of unassociated networks working individually to get the output

Introduction to Neural Network

Applications of NN:

1. Facial Recognition
2. Weather Forecasting
3. Music composition
4. Image processing and Character recognition

Introduction to Neural Network

Advantages of NN:

1. Fault tolerance
2. Real-time Operations
3. Adaptive Learning
4. Parallel processing capacity

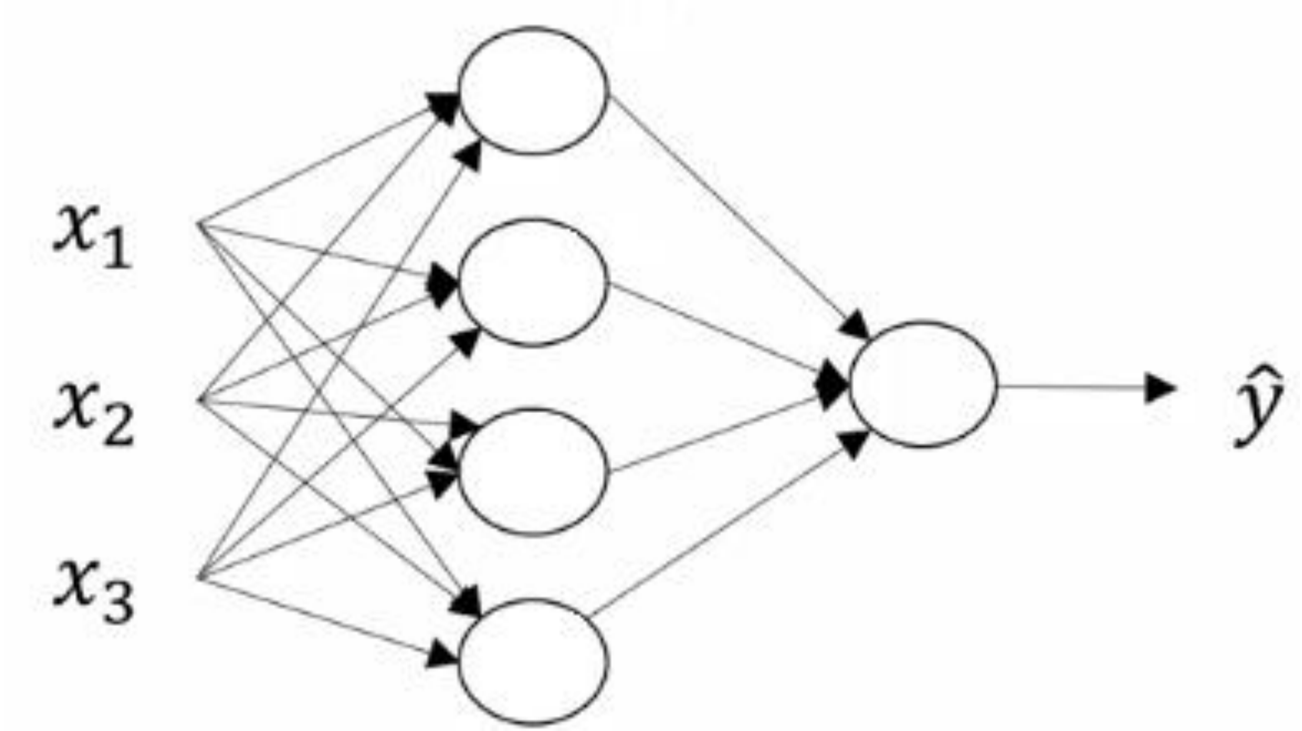
Advantages of NN:

1. Unexplained behavior of the network
2. Determination of appropriate network structure
3. Hardware dependence

SHALLOW NEURAL NETWORKS

Structure of SNN:

- 3 Layers
- 1 or 2 Hidden Layers
- Forward Propagation Process



SHALLOW NEURAL NETWORKS

Steps followed for SNN:

1. The first equation calculates the intermediate output Z[1] of the first hidden layer. $Z[1] = w[1].X + B[1]$
2. The second equation calculates the final output A[1] of the first hidden layer. $A[1] = \sigma (Z[1])$
3. The third equation calculates the intermediate output Z[2] of the output layer. $Z[2] = w[2]. A[1] + B[2]$
4. The fourth equation calculates the final output A[2] of the output layer which is also the final output of the whole neural network.

$$A[2] = \sigma (Z[2])$$

SHALLOW NEURAL NETWORKS

- Takes I/P only as Vectors

- Need more parameters

- Not compatible on complex functions

- No. of units grows exponentially with task complexity.

- More difficult to train with our current algorithms.

TRAINING A NETWORK

1. Select a training dataset.
2. Initialize NN parameters such as weights or gradients.
3. Choose an optimization Algorithm
4. Repeat the following steps:
    1. Forward Propagate an I/P.
        1. Linear Half
        2. Activation Half
    2. Compute the cost function.
    3. Compute the gradients of the cost with respect to parameter using back propagation.
    4. Update each parameter using the gradients according to the optimization algorithm.

TRAINING A NETWORK

Loss Functions :

After completion of activation function, the model performance is verified with the required output. The cost function is defined as the measurement of difference or error between actual values and expected values at the current position

Loss function = Actual O/P – Desired O/P

1. Forward Propagation
   - No change
2. Back Propagation
   - Update the parameters until the model reached required output.

TRAINING A NETWORK

Back Propagation:

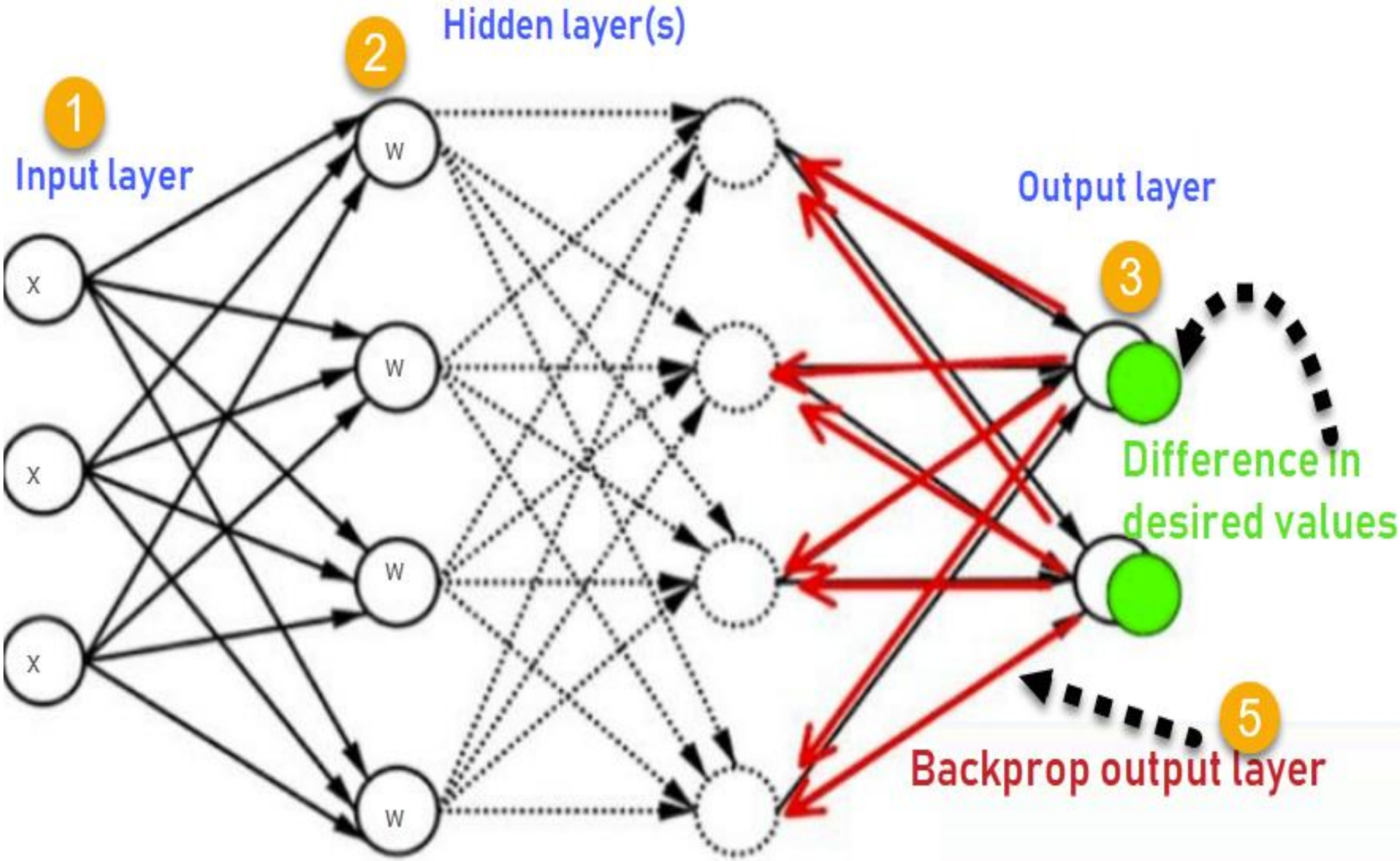
The Back propagation algorithm in neural network computes the gradient of the loss function for a single weight by the chain rule. This includes the step by step procedure:

– The network makes a guess about data, using its parameters.

– The network is measured with a loss function.

– The error is back propagated to adjust the wrong-headed parameters.

BACK PROPAGATION



BACK PROPAGATION

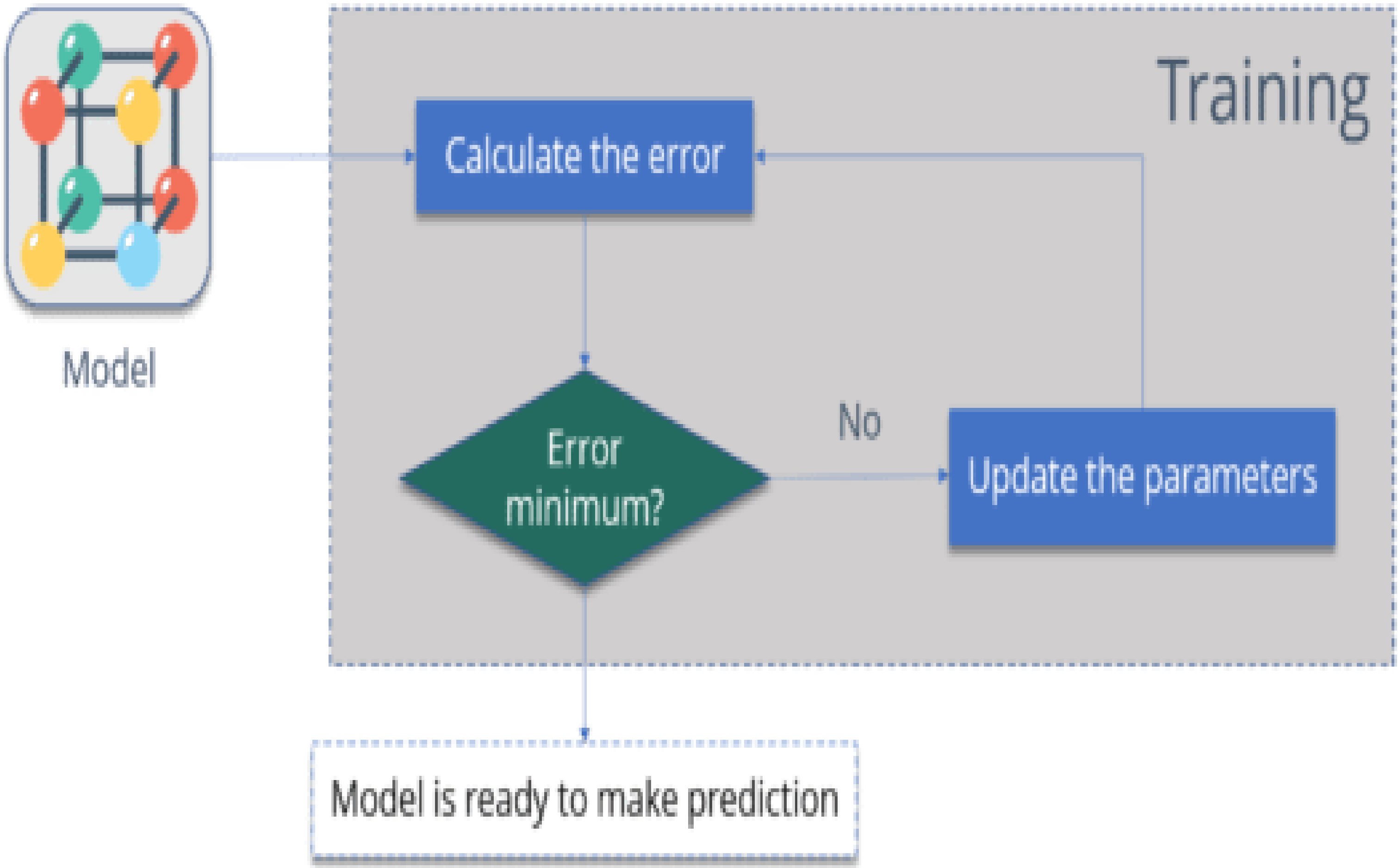
- Inputs X, arrive through the pre connected path.
- Input is modeled using real weights W. The weights are usually randomly selected.
- Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
- Calculate the error in the outputs
- Error = Actual Output – Desired Output
- Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

BACK PROPAGATION

Need for Back Propagation:

- ✓ Fast, simple and easy to program
- ✓ No parameters to tune apart from the numbers of input
- ✓ It does not require prior knowledge
- ✓ Standard method
- ✓ does not need any special mention of the features
- ✓ Train a model using trail and error approach
- ✓ Error becomes minimum

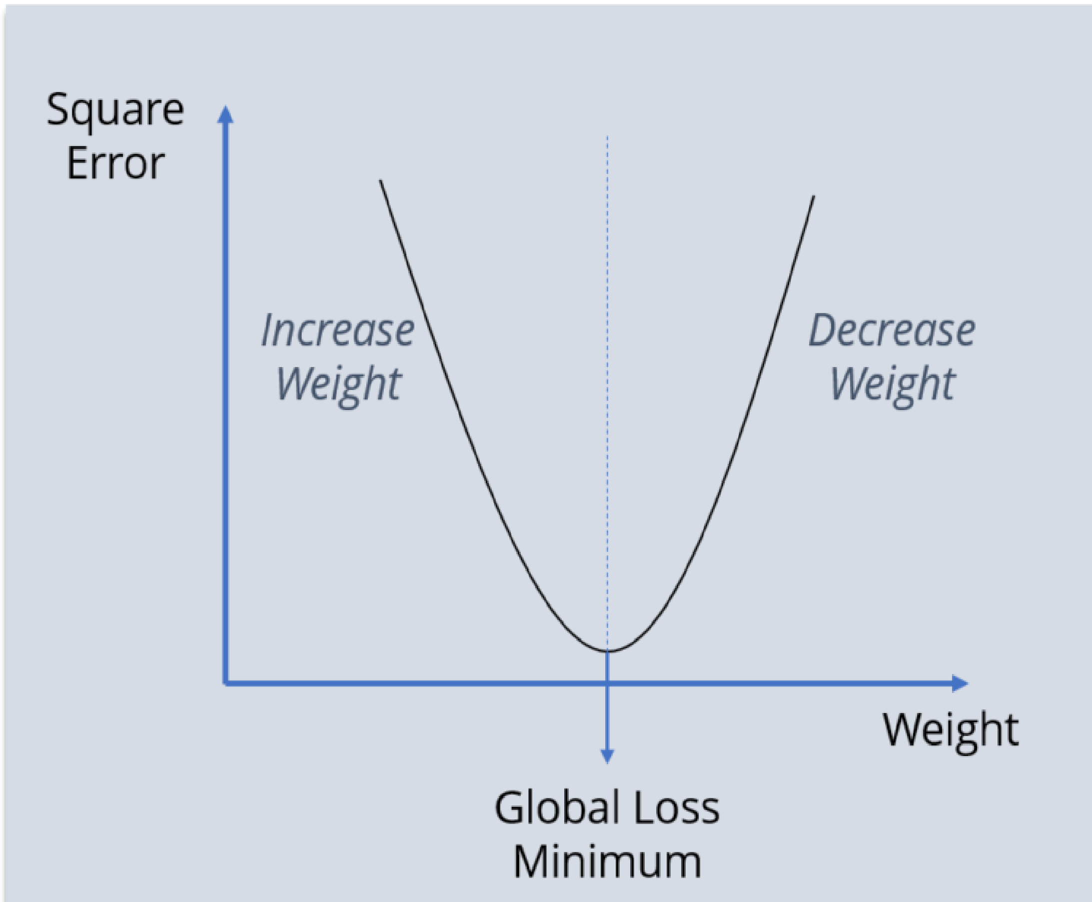
BACK PROPAGATION



BACK PROPAGATION

- Initialize random value to ‘W’ (weight) and propagated forward.
- Then compute the error. To reduce that error, the model propagated backwards.
    - ➢ If error increased, decrease the ‘W’ value.
    - ➢ If error decreased, increase the ‘W’ value.
- Repeat the process until error becomes minimal.

BACK PROPAGATION



BACK PROPAGATION

Types of Back propagation Networks are:

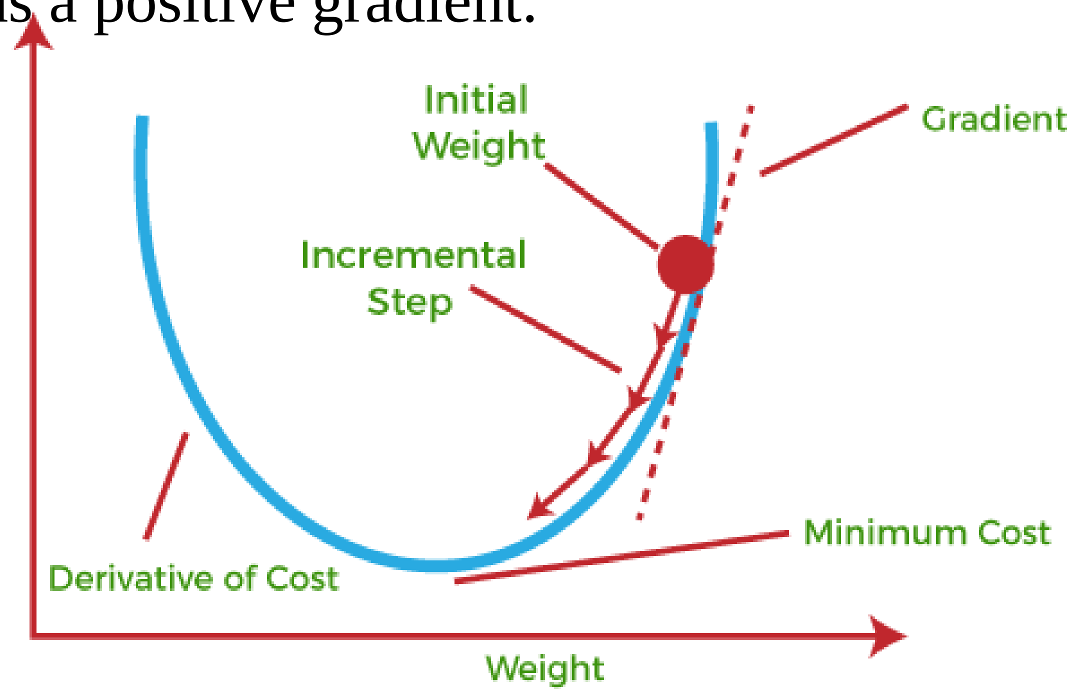
1. Static Back propagation
   - Produces a mapping of a static input for static output
   - Eg. optical character recognition
2. Recurrent Back propagation
   - Feed forward until a fixed value is achieved
   - Error is computed and propagated backward

Gradient Descent

Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning and deep learning models. The best way to define the local minimum of a function.

•Move towards a negative gradient.

•Move towards a positive gradient.



Gradient Descent

- ✓ Generic optimization algorithm
- ✓ Iterative approach
- ✓ Compute step size

    Step size = Gradient * Learning Rate

    New Parameter = Old Parameter – Step size

- ✓ Direction and Learning Rate

Gradient Descent

Types of Gradient Descent :

1. Batch Gradient Descent
2. Stochastic Gradient Descent
3. Mini-Batch Gradient Descent

Gradient Descent

Batch Gradient Descent :

Total number of samples from a dataset that is used for calculating the gradient for iteration.

Advantages of Batch gradient descent:

1. Less noise than other types of gradient descent.
2. Stable gradient descent convergence.
3. Computationally efficient - all training samples.

Disadvantages of Batch gradient descent:

1. Larger Datasets.

Gradient Descent

Stochastic Gradient Descent :

It processes a dataset and updates each parameters one at a time i.e. runs one training example per iteration.

Advantages of Stochastic Gradient Descent:

1. It is easier to allocate in desired memory.
2. Fast to compute.
3. Efficient for large datasets.

Disadvantages of Stochastic Gradient Descent:

1. Computation efficiency loss – More speed and space.
2. Noisy Gradient.

Gradient Descent

Mini-Batch Gradient Descent :

It divides the training datasets into small batch sizes then performs the updates on those batches separately. Combination of both batch gradient descent and stochastic gradient descent.

Advantages of Mini-Batch Gradient Descent:

1. It is easier to fit in allocated memory.
2. It is computationally efficient.
3. Stable gradient descent convergence.

Disadvantages of Mini-Batch Gradient Descent:

1. Less noisy Gradient.

Gradient Descent

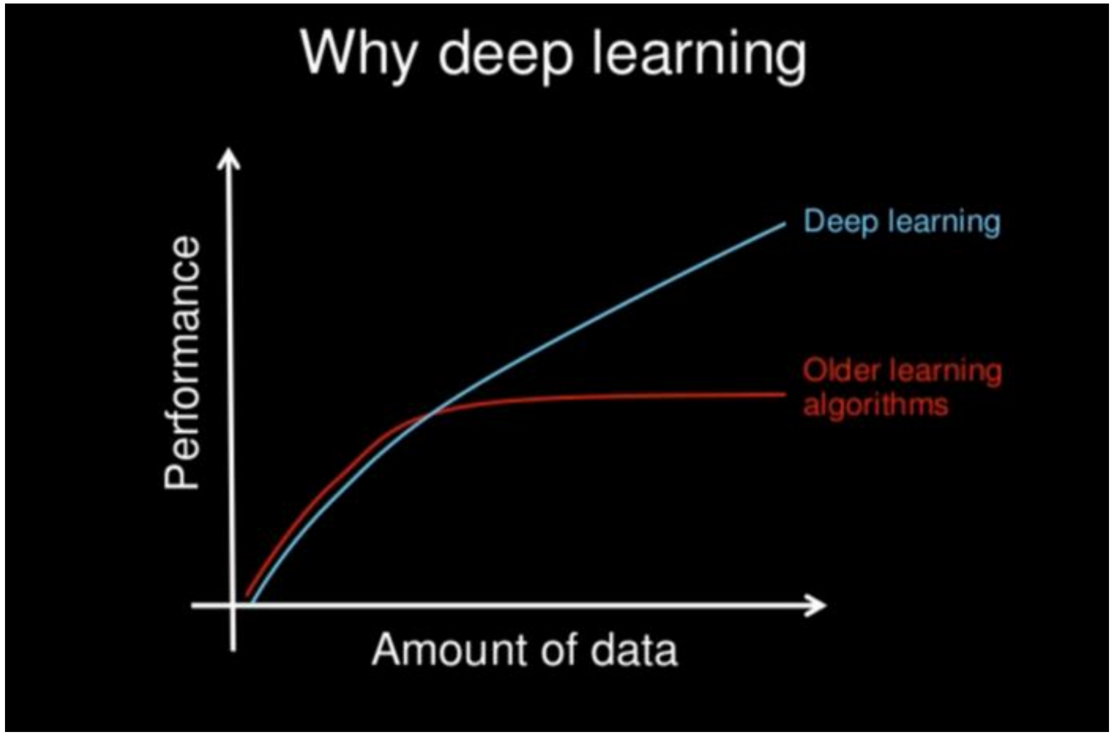
Disadvantages of Gradient Descent:

1. Local Minima and Saddle Point.
   - gradient is smaller than expected
2. Vanishing and Exploding Gradient
   - the Gradient is too large and creates a stable model
   - model weight increases, and they will be represented as NaN.
   - solved using the dimensionality reduction technique

Universal Approximation Theorem

The Universal Approximation Theorem tells us that Neural Networks has a kind of universality i.e. no matter what f(x) is (Mapping attributes to the output), there is a network that can approximately approach the result and do the job! This result holds for any number of inputs and outputs.

THE BOOM OF DEEP NEURAL NETWORKS:



- ✓Data Hungry Models
- ✓minimal preprocessing of data.
- ✓Low level understanding of programming Languages