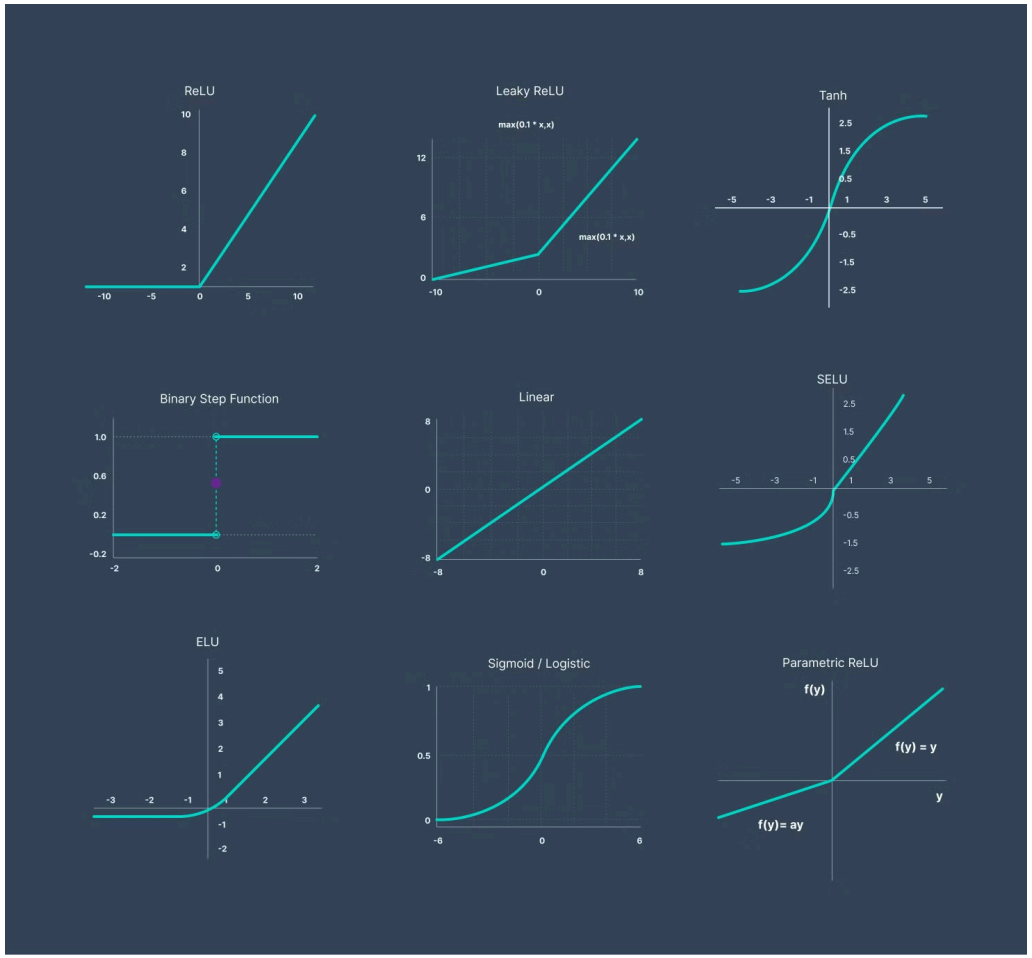
Machine Learning Algorithms

Machine learning algorithms are at the core of artificial intelligence and data science. They allow computers to learn from data and make decisions without being explicitly programmed. Understanding these algorithms is crucial for anyone looking to work in these cutting-edge fields.

Activation Functions

Activation functions play a crucial role in neural networks, determining the output of each neuron. Common activation functions include sigmoid, ReLU, and tanh. Below is a graph illustrating how each activation function behaves.



Supervised Learning Algorithms

Regression

Supervised learning algorithms used to predict continuous-valued outputs based on a set of input features and labeled training data.

Classification

Algorithms that learn to assign input to one of a number of categories or classes, based on labeled training data.

Classification Algorithms

Decision Trees

Classification algorithms that partition the data into subsets based on certain features.

Random Forest

An ensemble learning method for classification, based on constructing multiple decision trees.

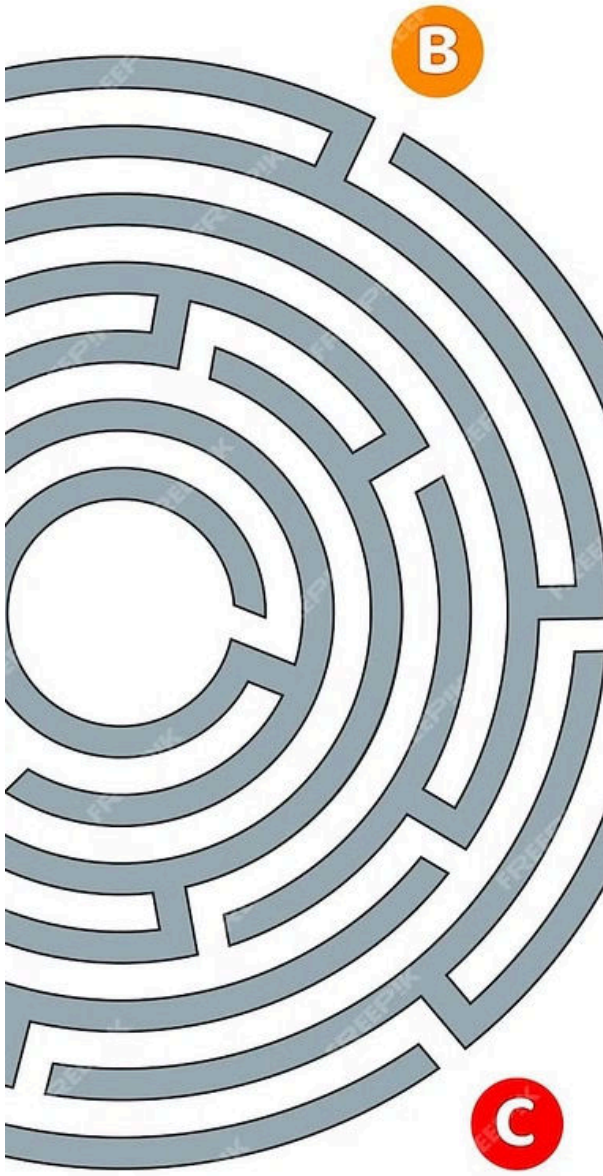
Logistic regression

selected when the dependent variable is categorical(binary output "TRUE" or "FALSE" .

neural networks

learns its mapping through supervised learning , adjusting based on the loss function through the process of gradient descent.

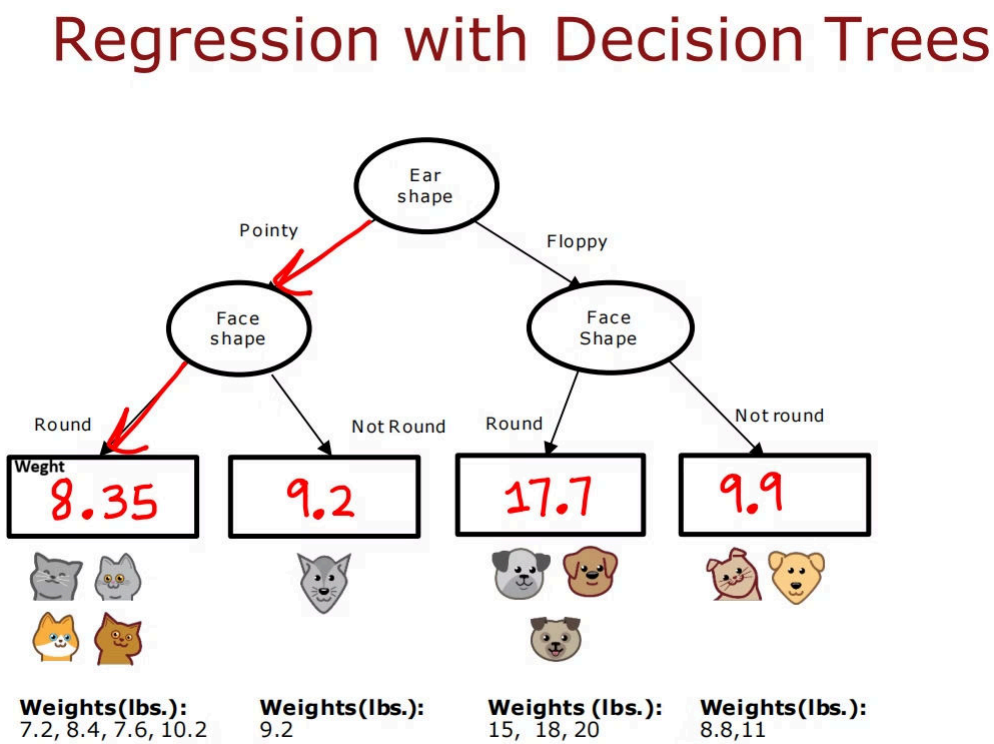
K-nearest neighbor (KNN) , Naive bayes, SVM



Decision Tree Models

Decision tree models are a popular machine learning algorithm used for classification and regression tasks. They are intuitive and easy to interpret, making them a valuable tool for understanding complex decision-making processes in various fields, including finance, healthcare, and marketing.

Decision Tree example :



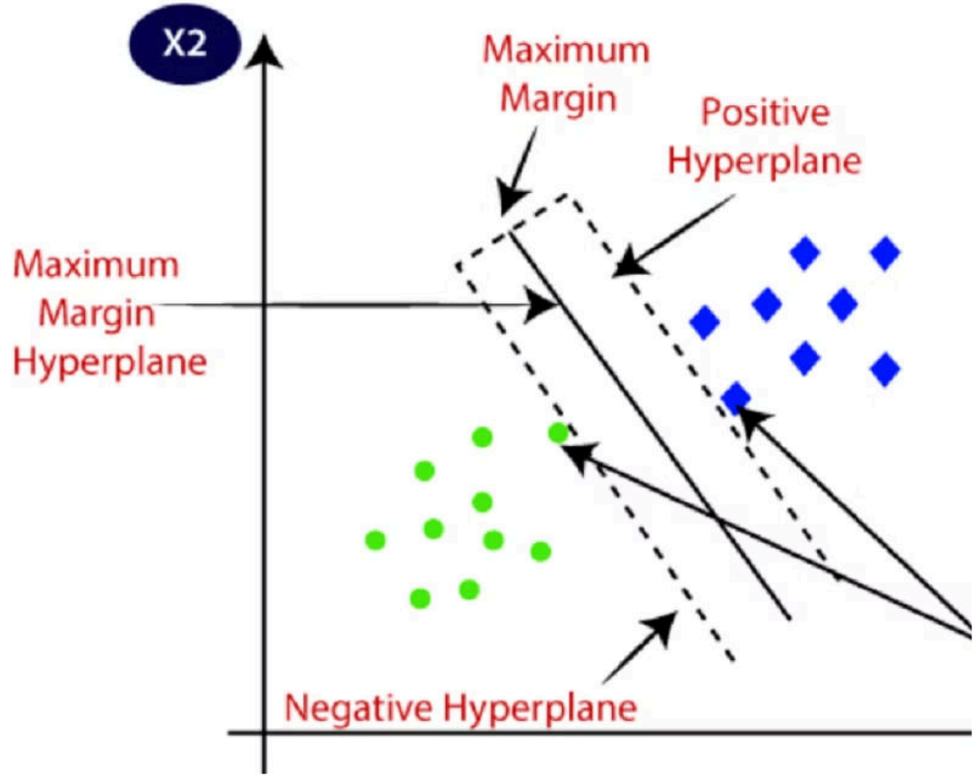
Random Forest Algorithm

The random forest algorithm is a powerful machine learning technique that combines multiple decision trees to make accurate predictions. It is known for its ability to handle complex datasets and handle both classification and regression tasks effectively.

Stock traders use Random Forest to predict a stock's future behavior. It's used by **retail companies** to recommend products and predict customer satisfaction as well.

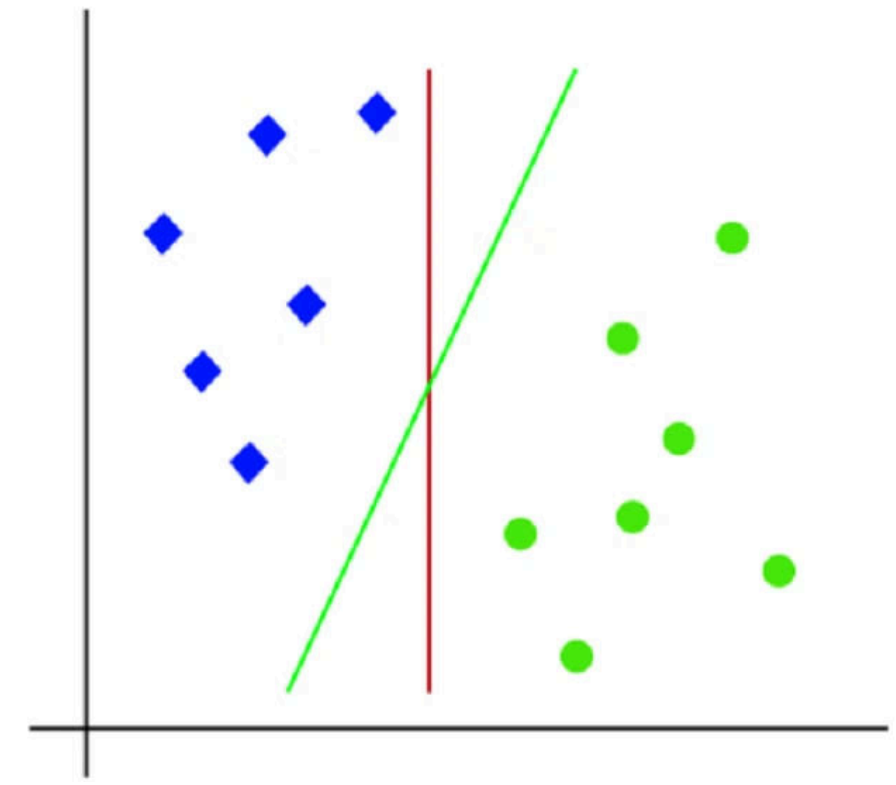
Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems, primarly used for classification. The goal of the SVM algorithm is to create the best line or decision boundary to easily put the new data point in the correct category. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors.

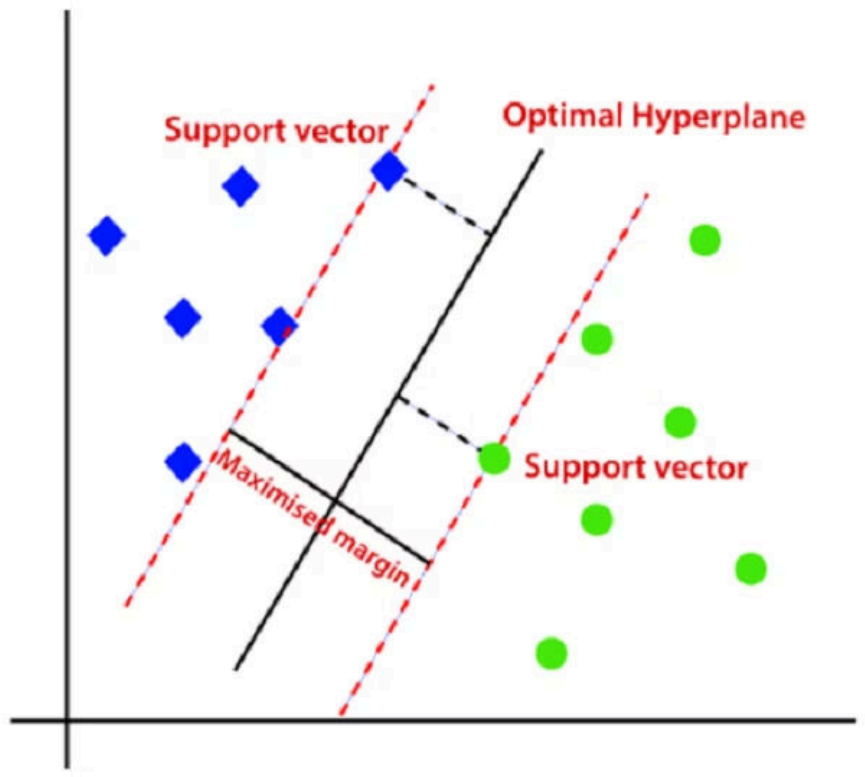


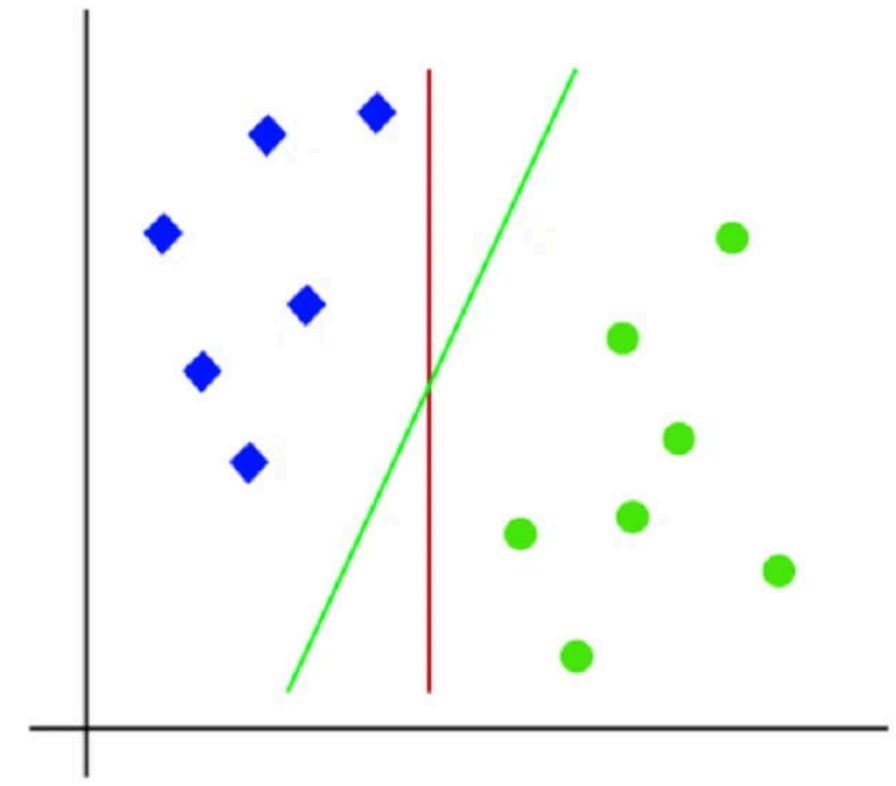
Support Vector Machine Types

Linear : Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can entirely divide the data points into their respective classes. A hyperplane that maximizes the margin between the classes is the decision boundary.



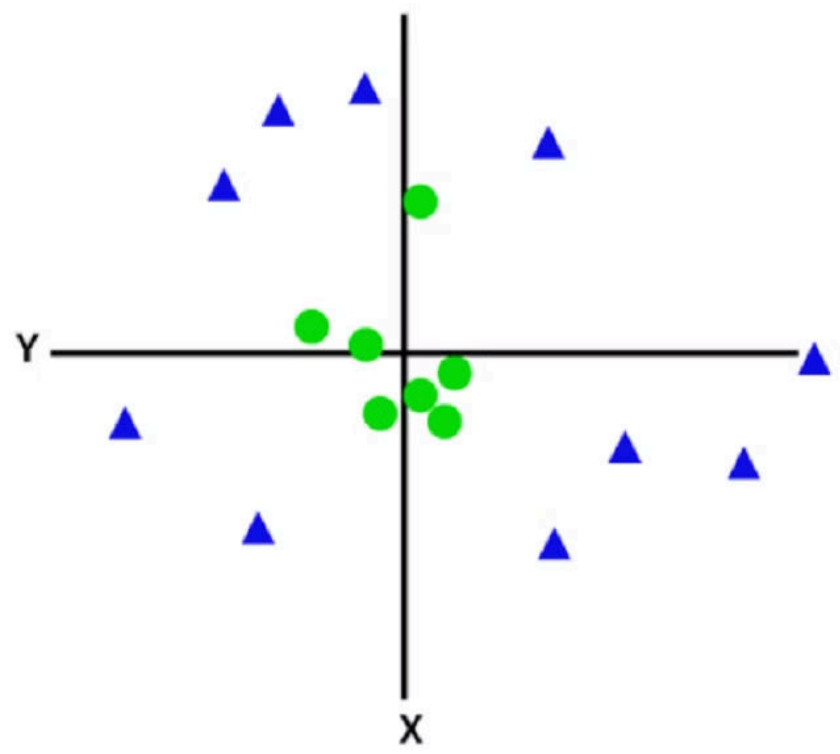
The green line is the best line (hyperplane)



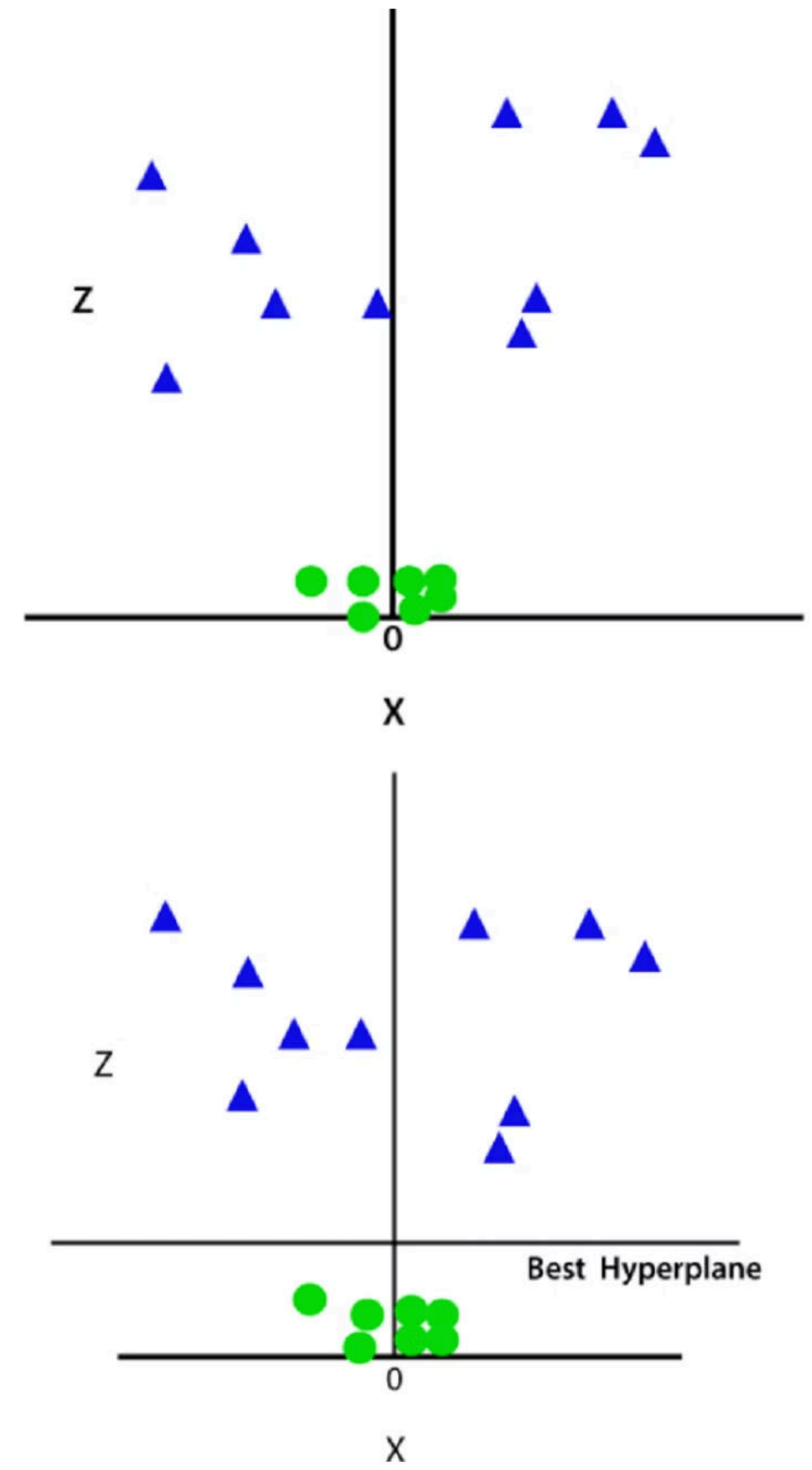


Support Vector Machine Types

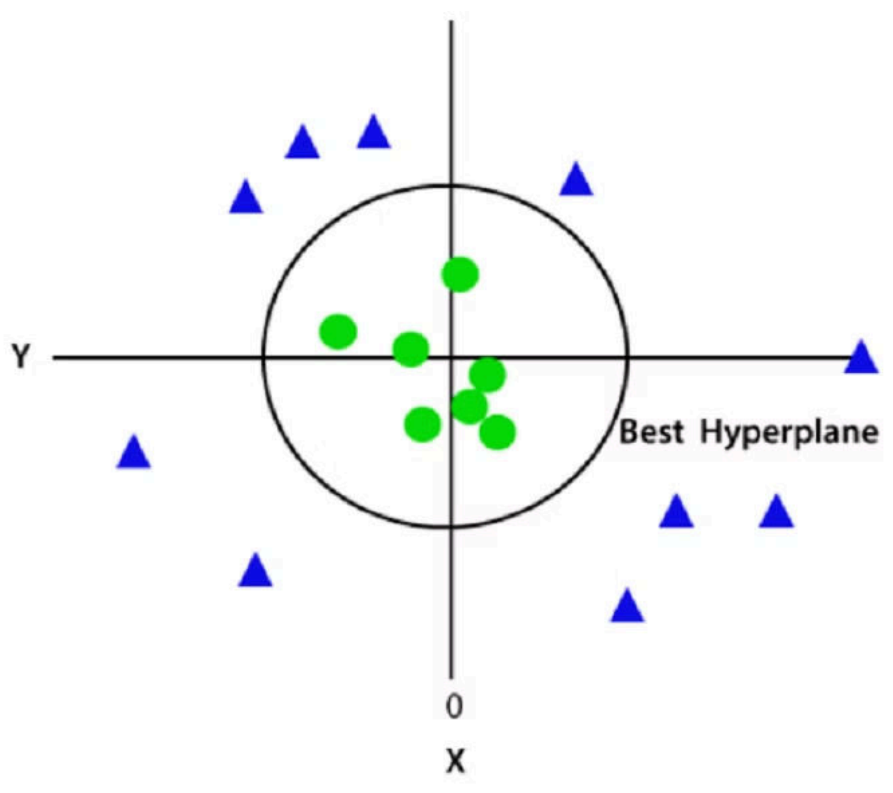
Non - Linear : Non-Linear SVM can be used to classify data when it cannot be separated into two classes by a straight line (in the case of 2D). By using kernel functions, nonlinear SVMs can handle nonlinearly separable data. The original input data is transformed by these kernel functions into a higher-dimensional feature space, where the data points can be linearly separated. A linear SVM is used to locate a nonlinear decision boundary in this modified space.



in order to separate these data points, more dimensions are needed . So, we will add a third dimension z. It can be calculated as: z=x2 +y2



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1



C parameter is a primary regularization parameter in SVMs. It controls the tradeoff between maximizing the margin and minimizing the misclassification of training data. A smaller C enables more misclassification, while a larger C imposes a stricter margin.

Regression Algorithms

1

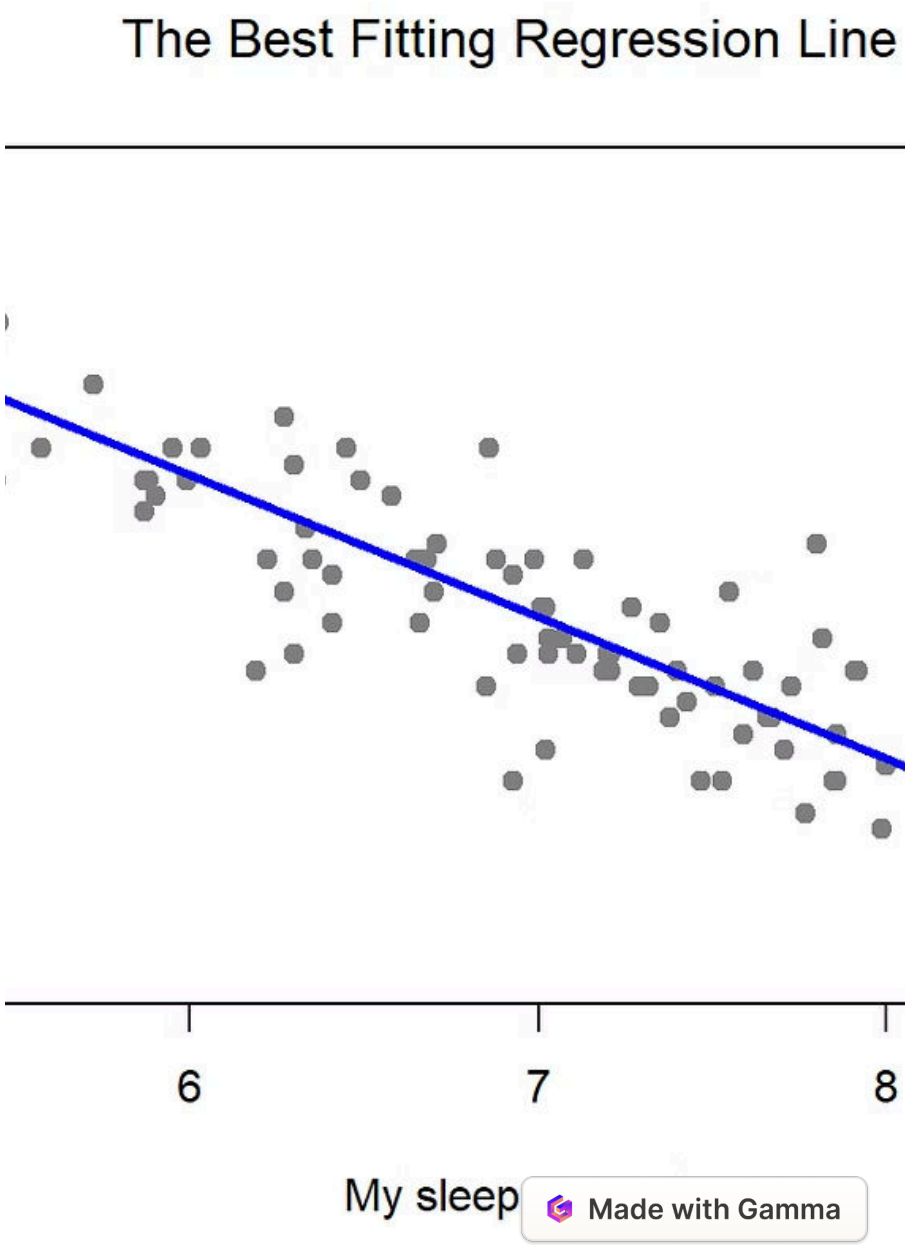
Linear Regression

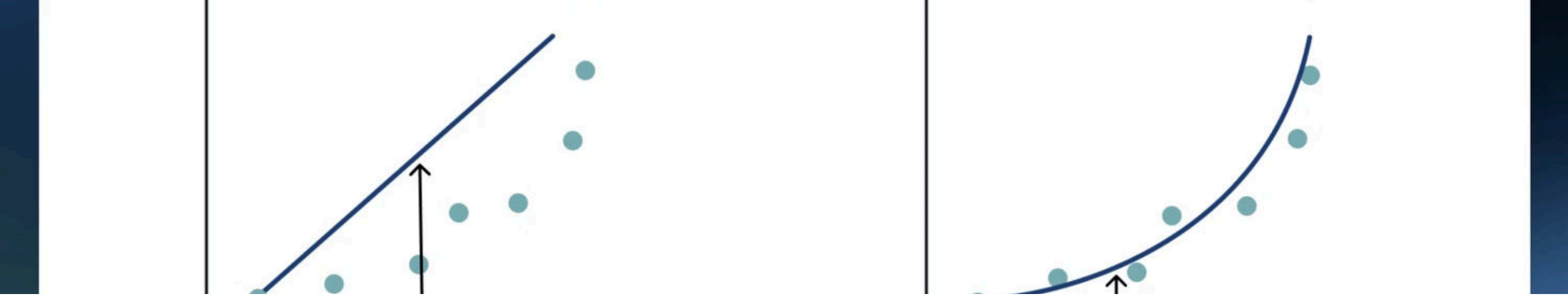
A technique to model the relationship between a dependent variable and one or more independent variables.

2

Polynomial Regression

Regression analysis where the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial.





Polynomial Regression

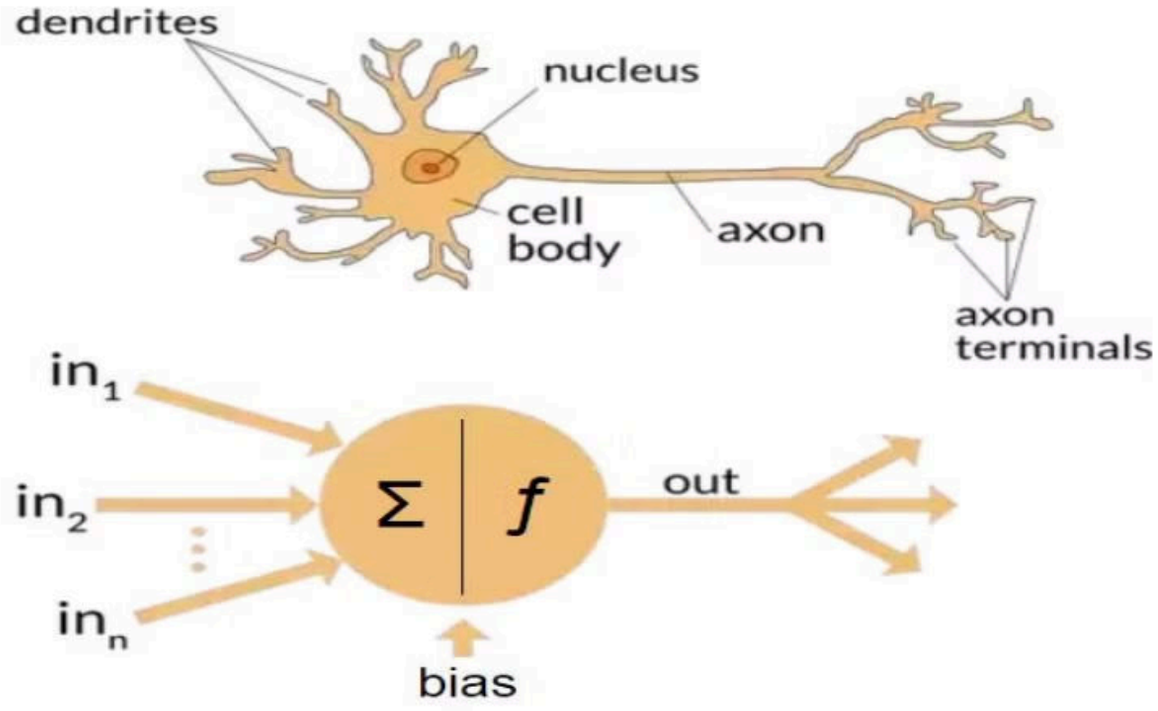
Polynomial regression is a type of regression analysis where the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial. It is used when the relationship between the variables is not linear, and can provide a better fit to the data. Below is an example equation and graph illustrating polynomial regression.

Equation: $y = a + bx + cx^2 + dx^3 + ...$

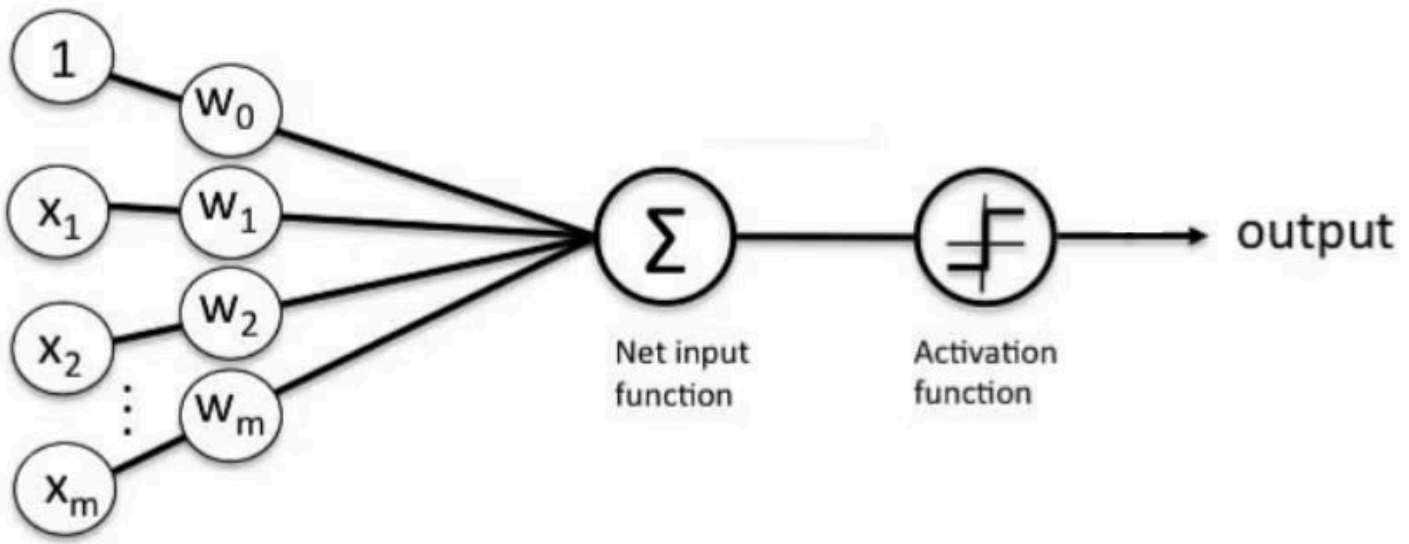
Neural Networks

Neural networks are a key component of modern machine learning. Inspired by the structure of the human brain, they can model complex relationships and make accurate predictions. Neural networks consist of interconnected layers of artificial neurons that learn from data through a process called training.

From biological to artificial :



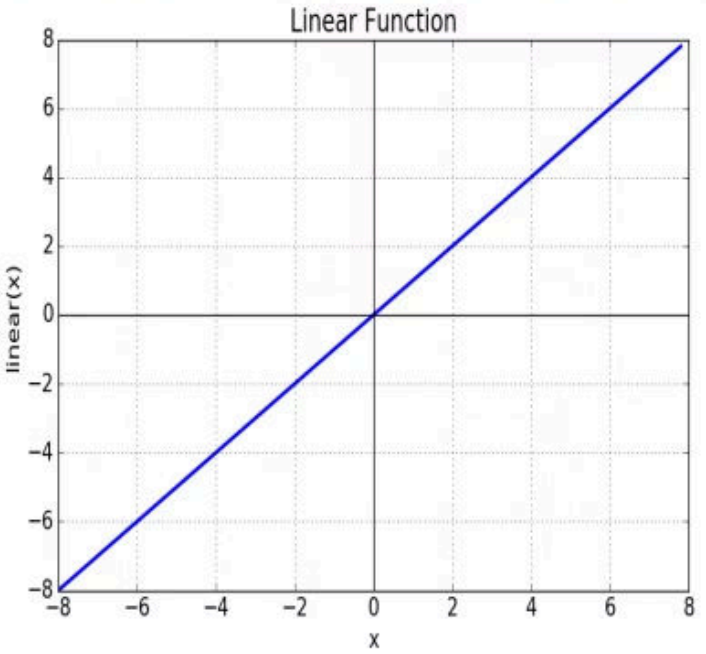
Perceptron model

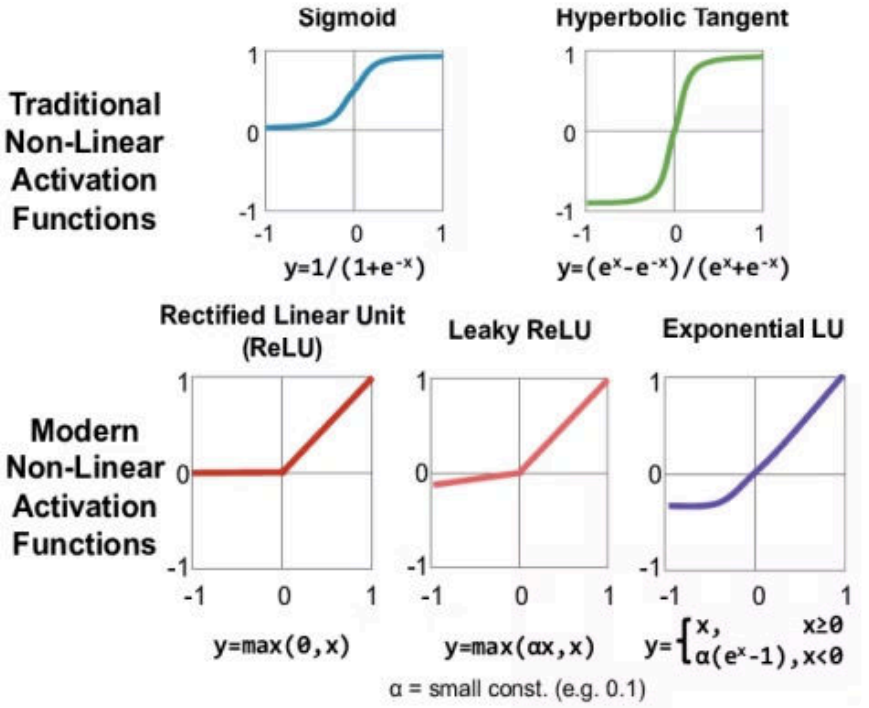


$$\text{Net} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

output = activation(Net)

Linear and non linear activation functions:



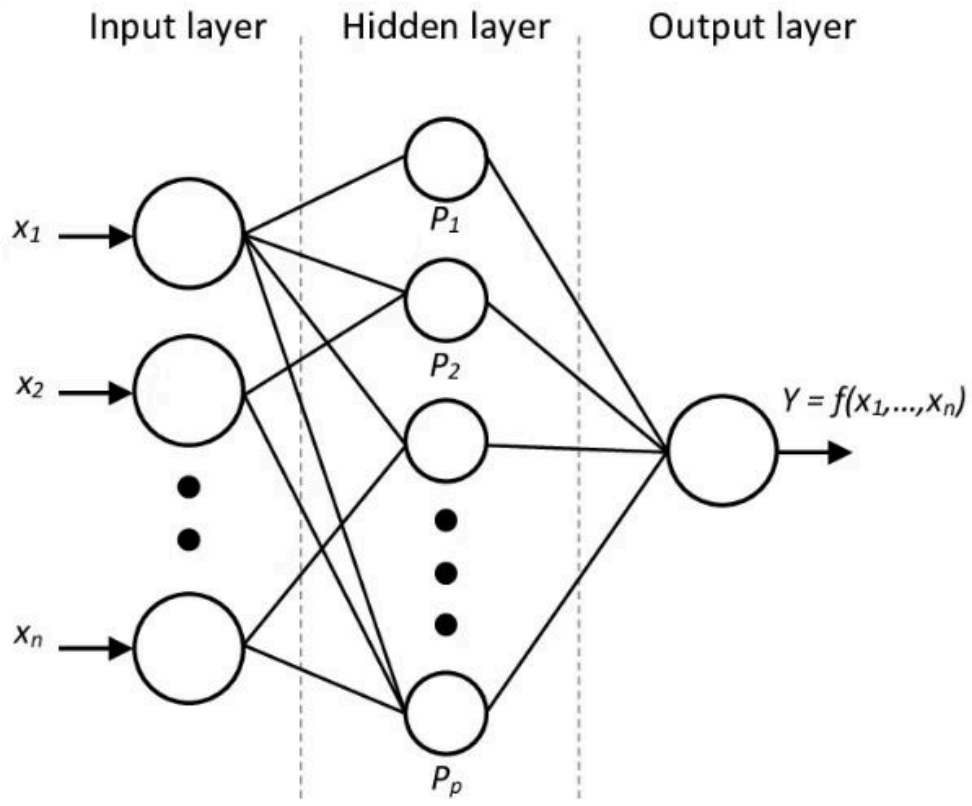


Well, the purpose of an activation function is to add non-linearity to the neural network.

For example we choose Sigmoid

output = 1 / (1 + exp(-net))

Neural Network :



Input : holds the input features (no computation is performed)

Hiddden layer : performs all kinds of computation on the features entered through the input layer and transfers the result to the output layer

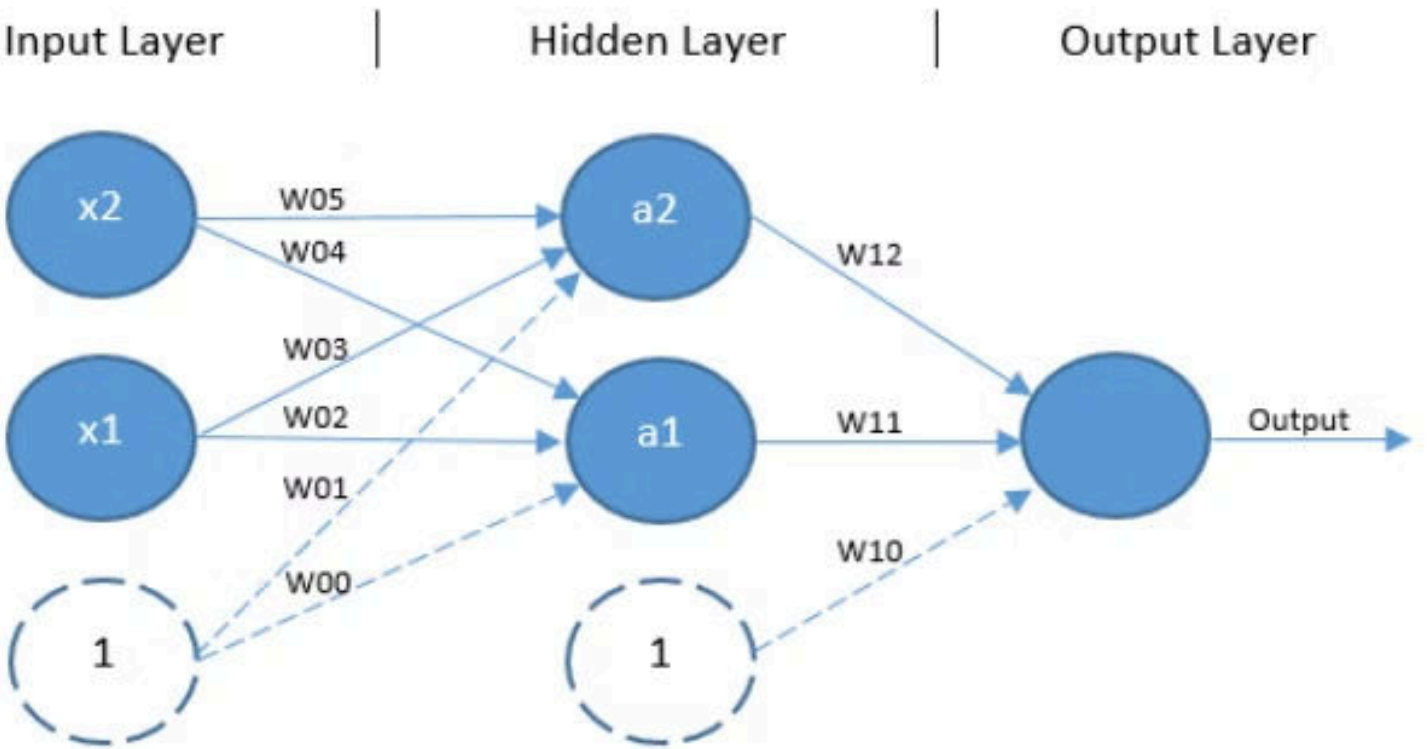
Output layer : It’s the final layer of the network that brings the information learned through the hidden layer and delivers the final value as a result.

Forward propagation? Backpropagation?

💡 **Feedforward Propagation** - the flow of information occurs in the forward direction. The input is used to calculate some intermediate function in the hidden layer, which is then used to calculate an output.

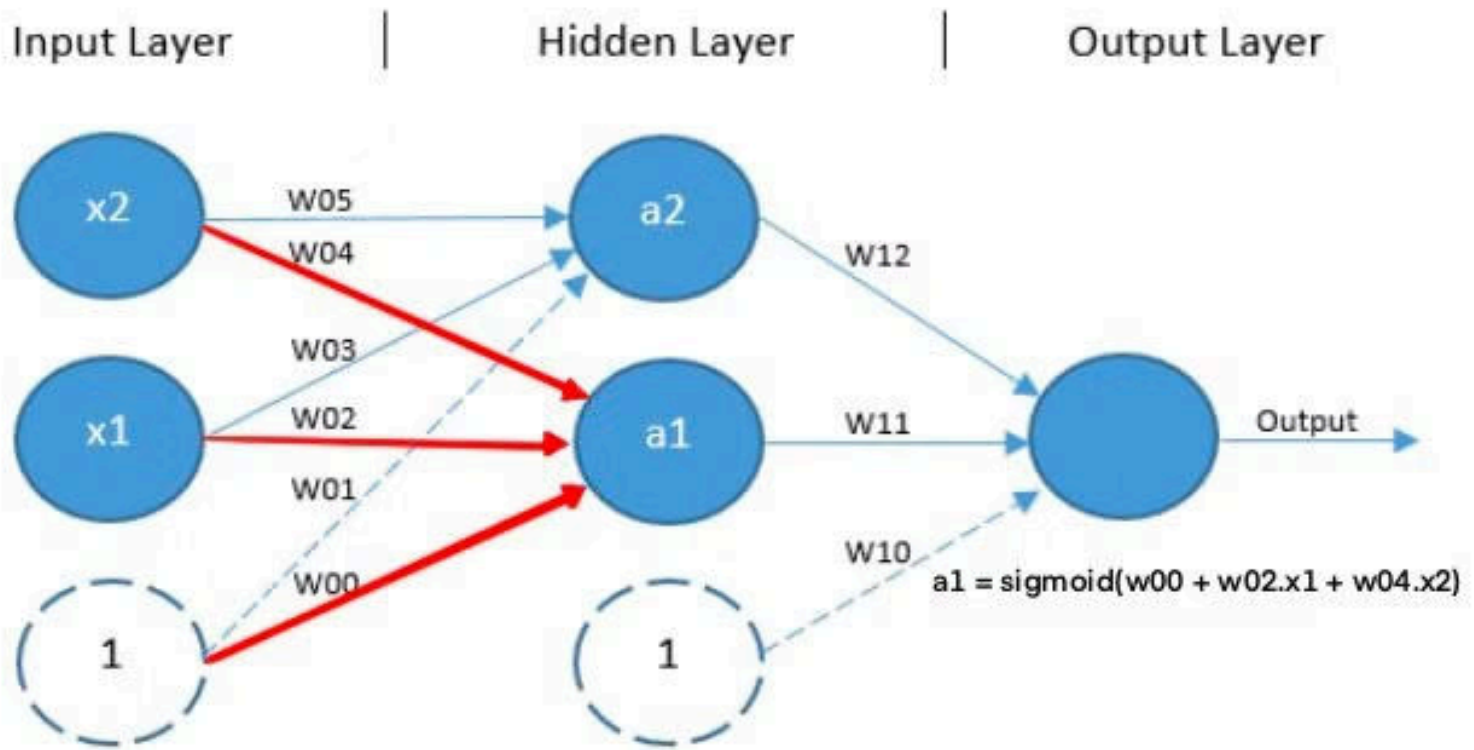
💡 **Backpropagation** - the weights of the network connections are repeatedly adjusted to minimize the difference between the actual output vector of the net and the desired output vector.

FeedForward propagation

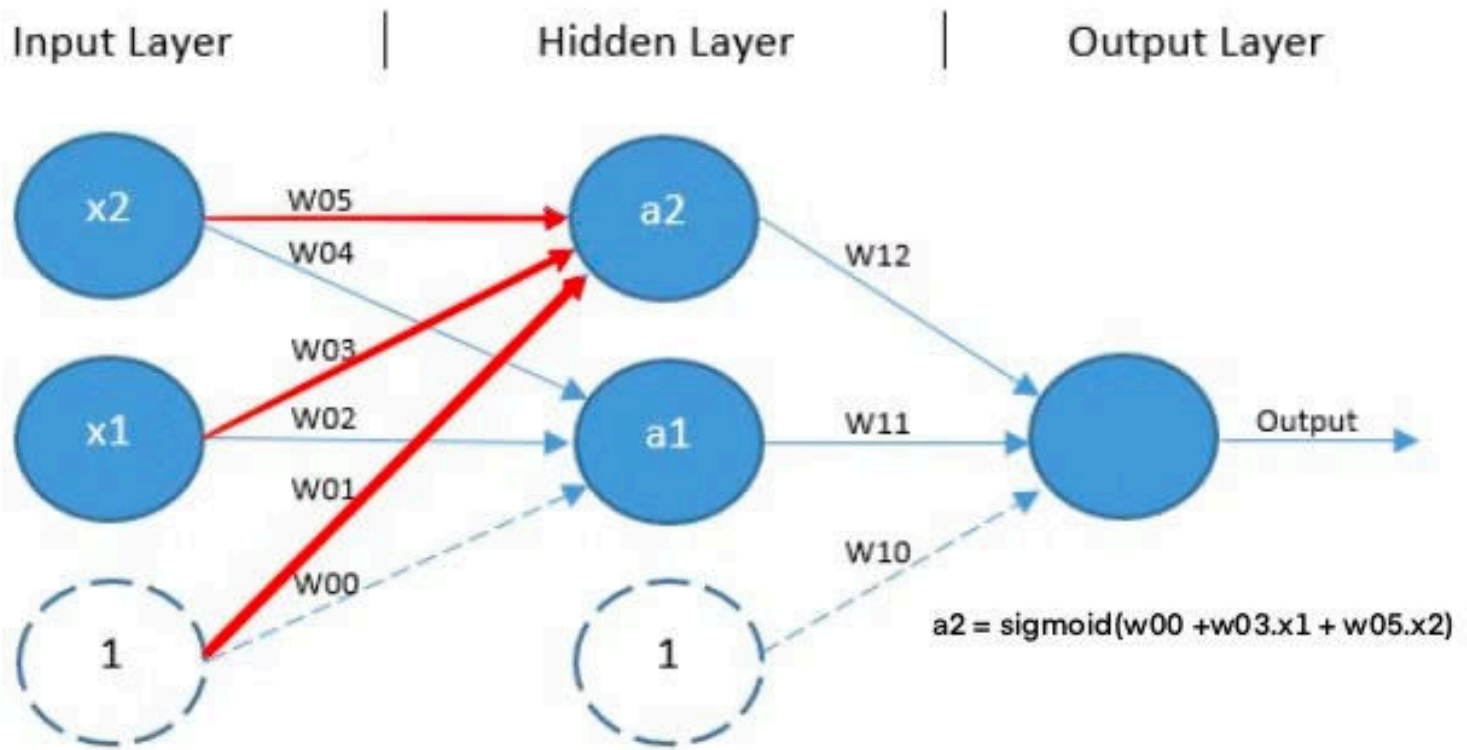


the bias is used to offset the result. It helps the models to shift the activation function towards the positive or negative side.

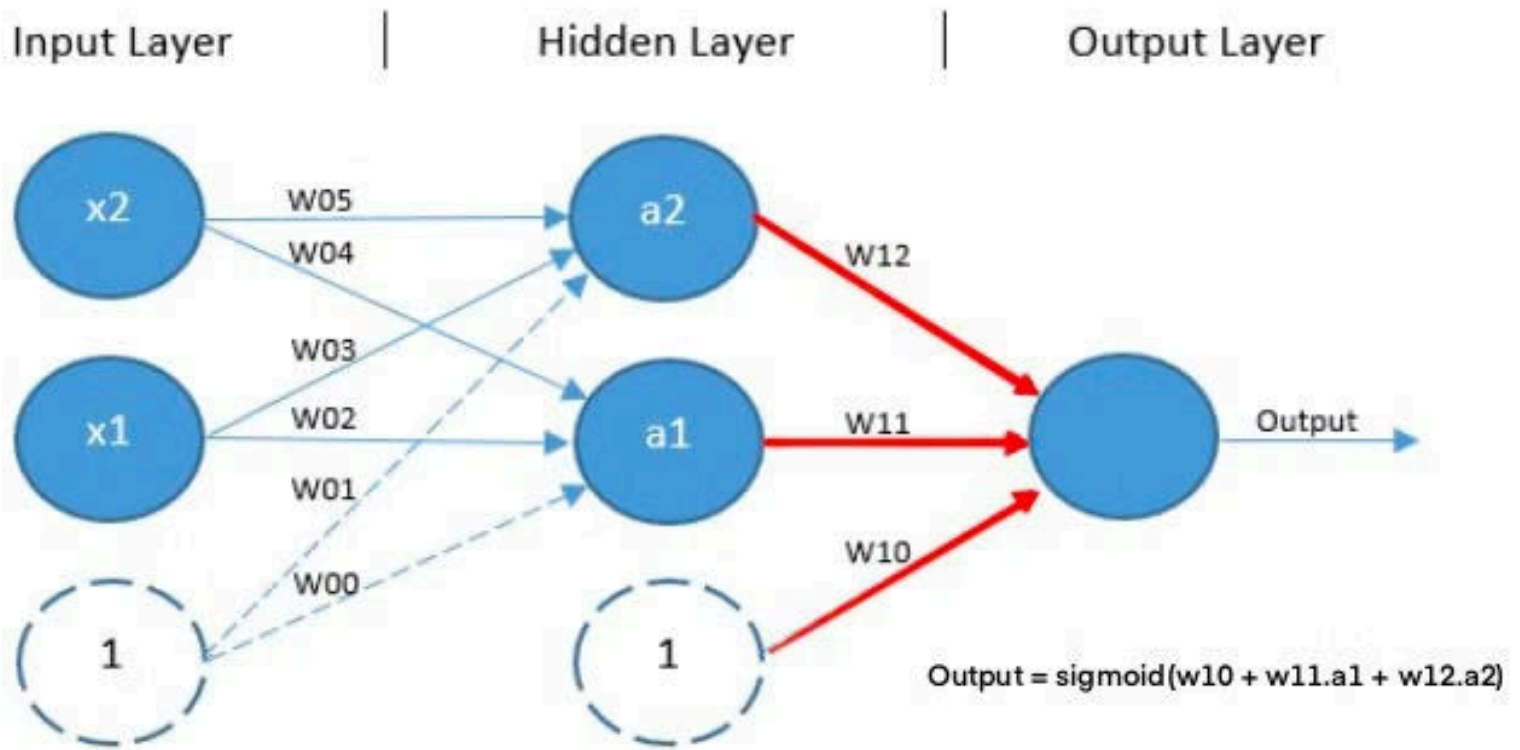
FeedForward propagation



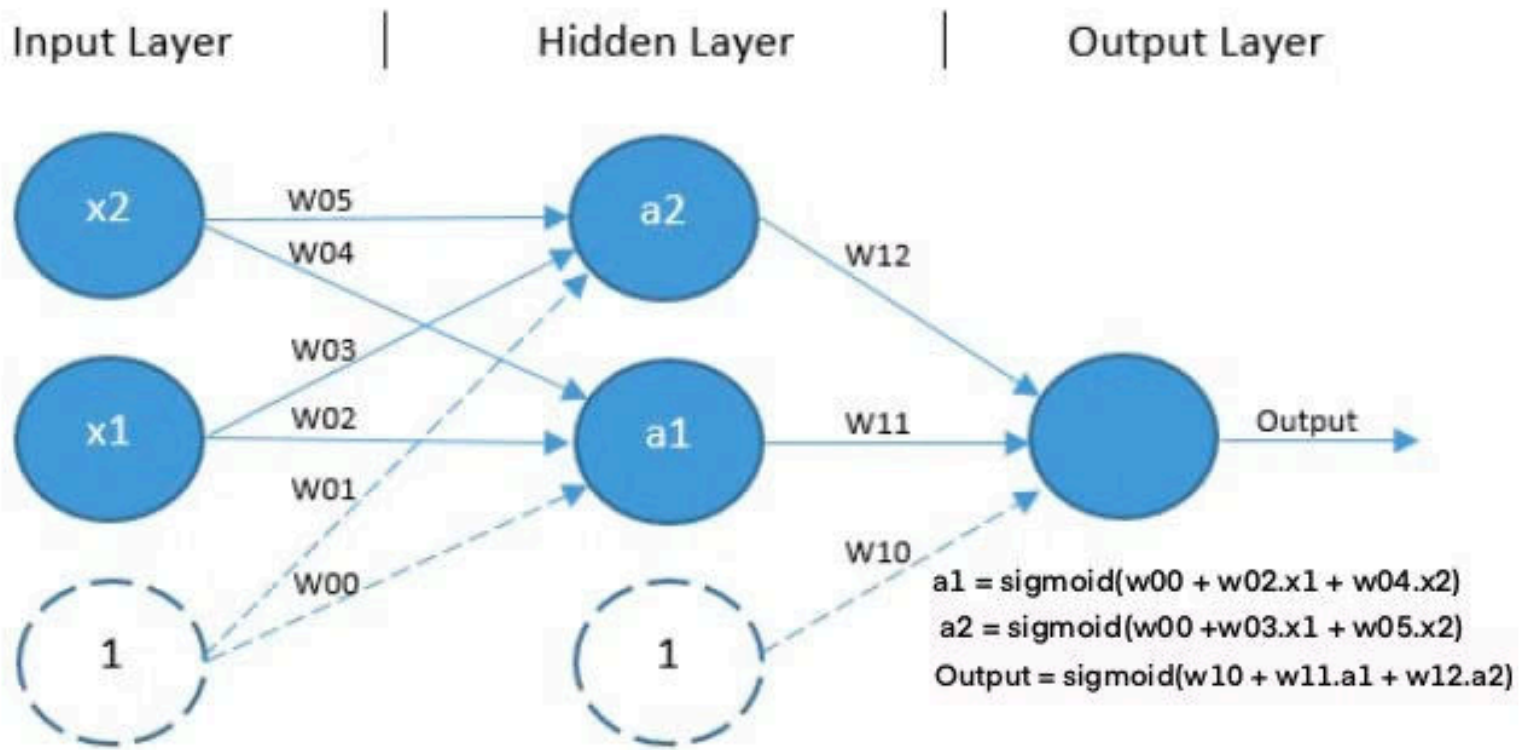
FeedForward propagation



FeedForward propagation



FeedForward propagation



Backpropagation

Our goal with backpropagation is to update each of the weights in The network so that they cause the actual output to be closer to the target Output by minimizing the error for each output neuron and the network as a whole

Backpropagation concepts

Its concepts (from calculus):

- Derivative
- Partial derivative
- Chain rule

i recommend watching this playlist : https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Clustering Algorithms

1 **K-Means**

An unsupervised learning algorithm used to partition data into distinct, non-overlapping subgroups.

2 **Hierarchical Clustering**

Algorithms that build a tree-like hierarchy of clusters by either a bottom-up or top-down approach.

Evaluation Metrics for Classification

Accuracy : measures how often the classifier correctly predicts

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision : explains how many of the correctly predicted cases actually turned out to be positive.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall : explains how many of the actual positive cases we were able to predict correctly with our model.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

F1 score : It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.

$$F1 = 2.\frac{Precision \times Recall}{Precision + Recall}$$

Log loss : assess the performance of a classification problem.

$$logloss_{(\mathrm{N}=1)} = y\log(p) + (1-y)\log(1-p)$$

Evaluation Metrics for Regression

Mean Square Error (MSE) : the squared mean of the difference between the actual values and predictable values.

$$\frac{1}{N}\sum_{i=1}^{n}(\text{actual values} - \text{predicted values})^2$$

Mean Absolute Error (MAE) : the mean size of the mistakes in collected predictions.

$$\textbf{MAE} = \textbf{True values} - \textbf{Predicted values}$$

Root Mean Squared Error (RMSE) : is the square root of MSE