**Machine Learning**

Machine Learning involves training of models to solve a problem on already available data to make accurate predictions or decisions on new data.

Training a model involves processing your data into useful feature vectors that will be given as inputs and also the design of robust algorithms that provide a computationally efficient solution to the problem.

There are many types of Machine Learning which include Supervised, Unsupervised, Semi-supervised and Reinforcement Learning.

**1. Supervised Learning**

Supervised Learning is a subset of Machine Learning where we have labeled data where each training example is mapped to an output label. The goal of Supervised Learning is to learn a mathematical curve from the training dataset which predicts outputs for new labels to some degree of accuracy.

**1.1 Feature Vectors and Target Labels**

Each training example has to be modified such that it contributes significantly to the performance of the model and has high predictive power.

Feature – A variable or property of the data that is being used for analysis.

We can identify the relevant features by plotting their dependence with the target values and then pick the features with highest correlation.

We can also engineer new features that are a mathematical function of the given features and which have high predictive power.

After the identification of important features, each data point is assembled into one vector called a feature vector which is the input to our model and each feature vector has a labeled output called a target label.

**1.2 Feature Engineering and Scaling**

The identifications of relevant features and engineering of new features which are relevant to the problem at hand is known as Feature Engineering.

Feature Scaling involves scaling of features by techniques like mean normalization or Z-score normalization which enhances the numerical stability of computations and ensures that all features contribute equally. Empirical observations suggest that it also improves the model performance and also gives faster and assured convergence in optimization algorithms like Gradient Descent.

**1.3 Cost Function**

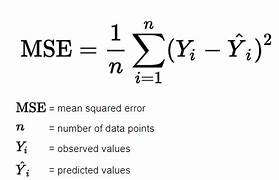
Cost function, also known as loss or objective function, is a mathematical function used to measure the performance of the model. It quantifies the difference between predicted and target values for each data point in the dataset. The main aim during training is to minimize this function which improves model’s accuracy.

This is done using optimization algorithms like Gradient Descent.

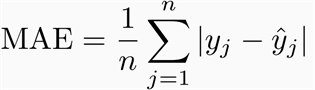
There are many types of cost functions and we choose one for our model depending on the problem at hand.

Some of the common loss functions are –

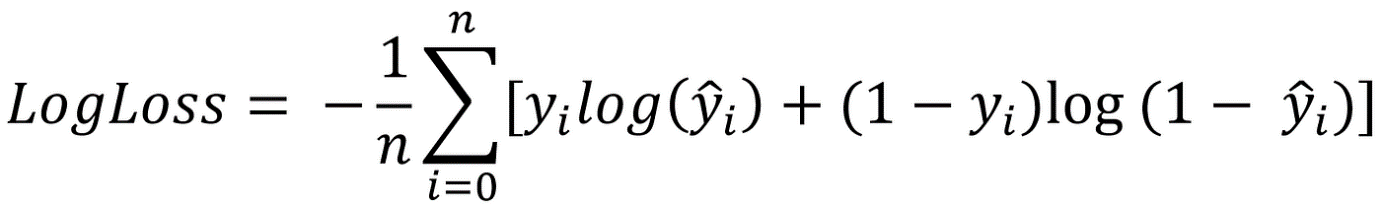
1. Mean Squared Error (MSE) – Average of sum of the square of the difference between predicted and target values. Mainly used for regression tasks.



1. Mean Absolute Error (MAE) – Average of sum of absolute difference between predicted and target values. Also used for regression tasks.



1. Cross-Entropy Loss (log loss) – Measure of difference between the probability distributions of true labels and predicted probabilities. Mainly used for classification tasks.



**1.4 Optimization Algorithms**

Minimizing the cost functions while training a model is done using optimization algorithms like Gradient Descent, Adam Optimizer etc.

The most common algorithm is Gradient Descent.

**Gradient Descent:**

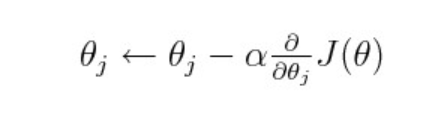
We assume the output label to be a function of the features. The expression would turn out to be a sum involving all the features multiplied by some coefficients called weights. These weights are what known as tunable parameters which means we can alter them to improve our model’s accuracy.

There is another thing called the learning rate which determines the amount by which every tunable parameter changes in every iteration of Gradient Descent.

Steps of Gradient Descent:

1. Initialization – Start with some random values for the parameters.

2. Cost function – Calculate the current value of cost function.

3. Update Parameters – Adjust the parameters to reduce the cost. For Gradient Descent this means decreasing the weight in the direction of the gradient of cost function with respect to that weight.

In the above figure, J is the cost function, α the learning rate, and θj the weight.

Repeat steps 2 and 3 until convergence that is the loss function no longer decreases significantly.

Some things to keep in mind while performing Gradient Descent:

1. Set a low learning rate. This helps the model converge faster and prevents any oscillations.



2. Use Regularization to prevent overfitting to data 

3. Can also modify batch size, use momentum and advanced optimizers like Adam which are offered by many Machine Learning libraries like TensorFlow and PyTorch.

**1.5 Regularization**

Regularization is a technique in Machine Learning which helps to prevent overfitting to training data. Regularization adds a penalty term to the cost function.

**Regularization Techniques:**

1. L1 Regularization – Adds absolute values of weights to cost function.

Cost = Loss + λ

L1 Regularization can help drive some features to zero which removes unnecessary features.

2. L2 Regularization – Adds squares of weights to the cost function.

Cost = Loss + λ

L2 Regularization shrinks all the weights of unnecessary features to values close to zero.

In above expressions, λ is known as the regularization parameter.

We can also use both L1 and L2 Regularization simultaneously.

**1.6 Regression and Classification**

A problem in Supervised Learning is mainly of two types, either of Regression or of Classification.

**Regression:**

The main task in regression is to identify a mathematical relationship between features and output labels.

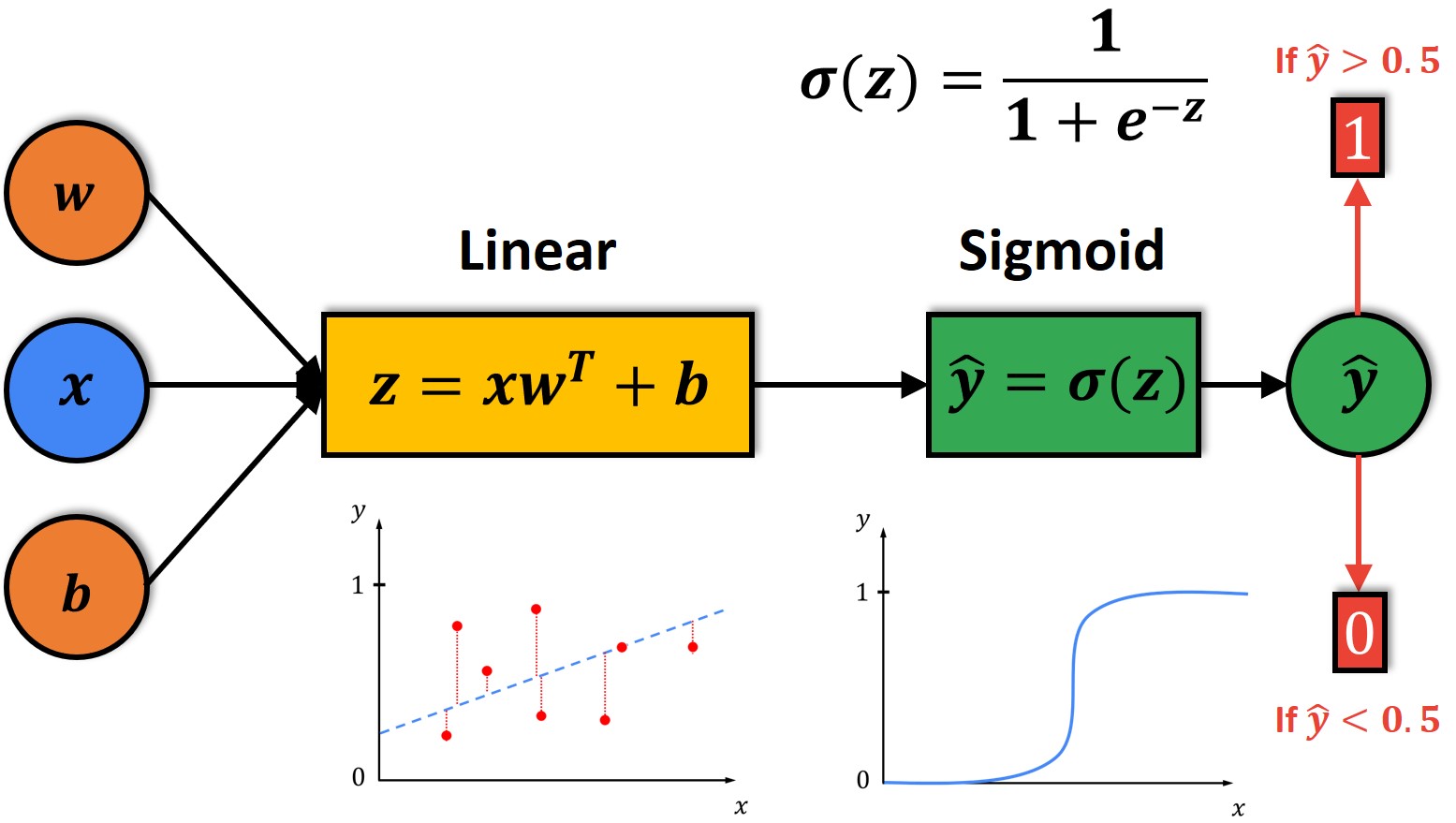
Regression can be Linear, Multiple Linear, or Polynomial regression.

**Classification:**

The main task in classification is to classify an input into predefined categories based on its features. Classification can be Binary or Multi-Class Classification.

Usually, a decision boundary or a classification threshold is used to classify inputs.

For binary classification, we mainly use Logistic Regression whose overall structure is as follows.



**1.6 Train-Test-Validation Split**

It is good practice to break the available data into training data that is used to train the model, test data which is used to check the model’s accuracy and performance, and finally validation data that checks if the model hasn’t overfit the training data.

The split ratio depends from model to model but an 80:10:10 split is considered good.

**2. Unsupervised Learning**

In Unsupervised Learning the data we have does not have any target labels.

There are many ways to deal with unlabeled data, the most common being the K-means clustering. In this algorithm our main goal is to create clusters of closely related data points in space. There is an underlying assumption that close points in space are closely related to each other, hence points within a cluster must have the same output label.

**2.1 The K-means clustering algorithm**

The main aim of this algorithm is to map all of the data points into K mutually exclusive clusters with the underlying assumption that points in the same cluster are similar and points of two different clusters are dissimilar.

**Steps:**

1. Initialization – Choose the number K of clusters that you want to make and also initialize the K cluster centroids randomly. A centroid is a point representing the center of a cluster.

2. Assign points to clusters – A datapoint which is closest to a particular centroid is assigned to the cluster whose center is that centroid, breaking ties arbitrarily. Here the term close is with respect to the distance of the feature vector of the data point from the cluster centroid in a suitable vector space.

3. Update cluster centroids – Compute the centroids of all the clusters by taking the average of the features of all the data points in that cluster and update the value of the centroid to this average vector.

Repeat steps 2 and 3 until convergence that is there is no change in points among clusters and the centroids have stabilized or after a certain number of iterations.

**3. Reinforcement Learning**

Reinforcement Learning involves training a model while providing constant feedback so that the model makes constant improvements over time.

**Deep Learning**

Deep Learning involves training neural networks that understand complex patterns and representations in data. The beauty of these networks is that they don’t rely completely on human crafted features but create new features by using patterns in the data.

**Neural Networks**

Neural networks are the foundation of deep learning. Their architecture is similar to the neurons in a biological brain.

**Components of a Neural Network:**

1. Neuron (Node) – The very basic unit of a neural network that performs computation on the input and applies an activation function to produce an output.

2. Layers – Neurons are organized into layers within a neural network.

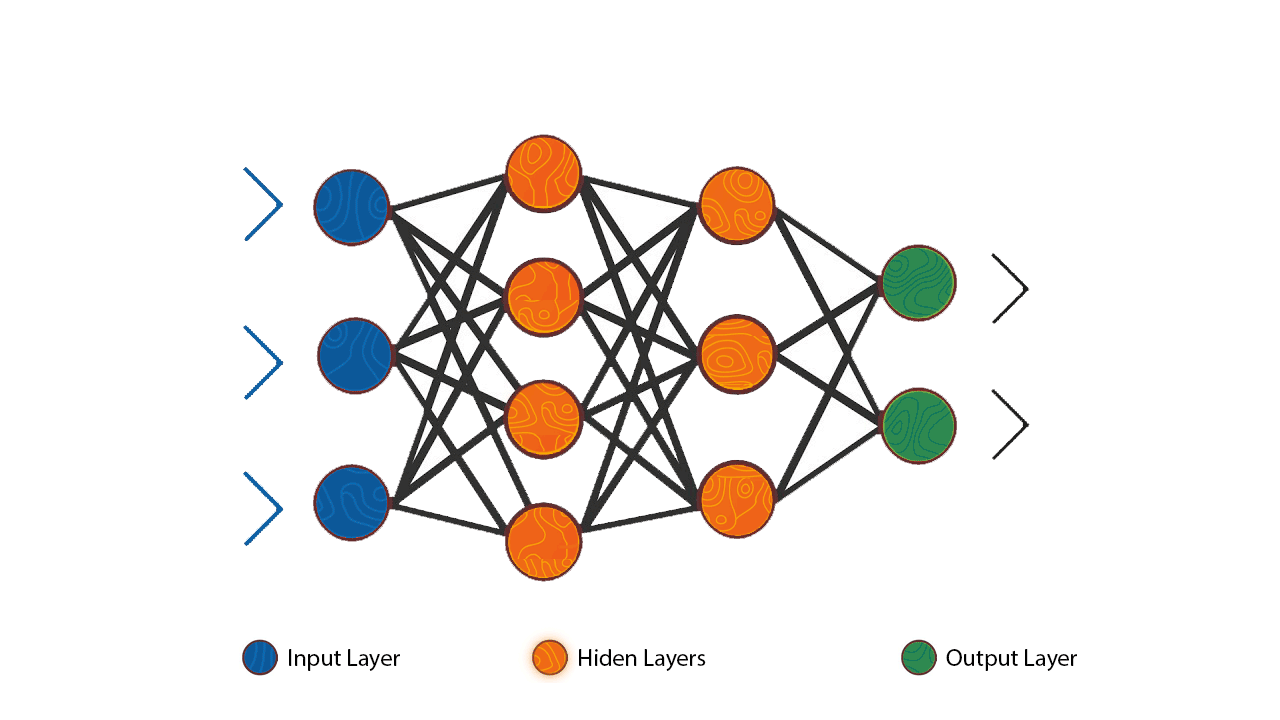
There are three types of layers namely:

* Input layer – Receives input.
* Hidden layer(s) – Intermediate layers between input and output layer of a neural network which perform computations.
* Output layer – Receives the final output of a network.

3. Weights and Biases – Each connection between any two neurons of adjacent layers has a weight associated to it which determines the strength of the influence of one neuron on another. Biases are constant parameters added to the input in the neurons to help the model fit the data better.

4. Activation Function – After multiplying features with respective weights and adding the bias term, this weighted sum of inputs is put into an activation function like sigmoid, tanh (hyperbolic tangent), ReLU (Rectified Linear Unit), Leaky ReLU, and Softmax (mainly used in the output layer in a classification problem).

**5. Feedforward and Backpropagation –** Neural networks use feedforward propagation to pass data through the network from input to output. Backpropagation is used to update weights and biases during training by propagating the error backward through the network, adjusting weights to minimize the difference between predicted and actual outputs.



Above illustration depicts how the neurons are fired in a neural network ultimately producing the final output.

There are many different types of neural networks used in many applications.

**Natural Language Processing**

The main aim of NLP is to interpret day to day human language and also respond to it in a meaningful way. This has led to the development of many algorithms and methods.