AI, ML, Deep Learning and Data Mining Methods for Healthcare

Chapter-02

Outline

- Knowledge discovery and Data Mining
- ML, Multi classifier Decision Fusion
- Ensemble Learning
- Meta-Learning
- Evolutionary Algorithms
- Deep Learning
- Dimensionality reduction algorithms

Overview of Machine Learning and Data Mining

- Machine learning and data mining are major fields within artificial intelligence that are heavily used in many domains for a variety of applications.

Knowledge Discovery and Data Mining

- The goal of data mining is to discover inherent and previously unknown information from data to represent as knowledge.
- In many cases, such methods are used to summarize and model very large data sets, capturing what salient (high support or occurring frequently) and interesting patterns reside in the data that may not have otherwise been discovered.
- Many of these methods were motivated by the growing demand to mine consumer transaction data to determine which items were purchased together, so that retail stores could better position their products within the store and track buying patterns.
- Two such methods are **Association Rule Mining and Sequential Pattern Discovery**.

1. Association Rule Mining

- The methodology of association rule mining produces knowledge in rule form, called an association rule.
- An association rule describes a relationship among different attributes.
- Association rule mining was introduced by Agrawal et al. [1] as a way to discover interesting co-occurrences in supermarket data (the market basket analysis problem).
- It finds frequent sets of items (i.e., combinations of items that are purchased together in at least N database transactions) and generates from the frequent itemsets (such as {X, Y}) association rules of the form $X \rightarrow Y$ and/or $Y \rightarrow X$.

More formally, let

- D = {t1, t2,..., tn} be the transaction database,
- ti represent the ith transaction in D.
- Let I = {i1, i2,..., im} be the universe of items.
- A set $X \sqsubseteq I$ of items is called an itemset.
- When X has k elements, it is called a k-itemset.
- An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$.

- Various statistical measures exist that offer metadata about the rule. Support, confidence, and lift are three such measures.

The *support* of an itemset X is defined as

$$\text{Support}(X) = \frac{\text{number records with } X}{\text{number records in } D}$$

The *support* of a rule $(X \rightarrow Y)$ is defined as

$$\text{Support}(X \rightarrow Y) = \frac{\text{number records with } X \text{ and } Y}{\text{number records in } D}$$

The *confidence* of a rule $(X \rightarrow Y)$ is defined as

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{number records with } X \text{ and } Y}{\text{number records with } X}$$

- A high confidence value suggests a strong association rule. However, this can be deceptive.
- For example, if the antecedent (X) or consequent (Y) has a high support, it could have a high confidence even if it was independent.
- This is why the measure of **lift** was suggested as a useful metric.
- The lift of a rule ($X \rightarrow Y$) measures the deviation from independence of X and Y.
- A lift greater than 1.0 indicates that transactions containing the antecedent (X) tend to contain the consequent (Y) more often than transactions that do not contain the antecedent (X).
- The higher the lift, the more likely that the existence of X and Y together is not just a random [illegible] due to the relationship between them.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)}$$

The Apriori algorithm

- Apriori [1] is the most widely used algorithm for finding frequent k-itemsets and association rules.

- Calculate the support of all 1-itemsets and prune (i.e., remove) any that fall below the minimum support, specified by the user.
- Loop:
  1. Form candidate k-itemsets by taking each pair (p, q) of (k-1)-itemsets, where all but one item match. Form each new k-itemset by adding the last item of q onto the items of p.

  2. Prune the candidate k-itemsets by eliminating any itemset that contains a subset not in the frequent (k-1)-itemsets.

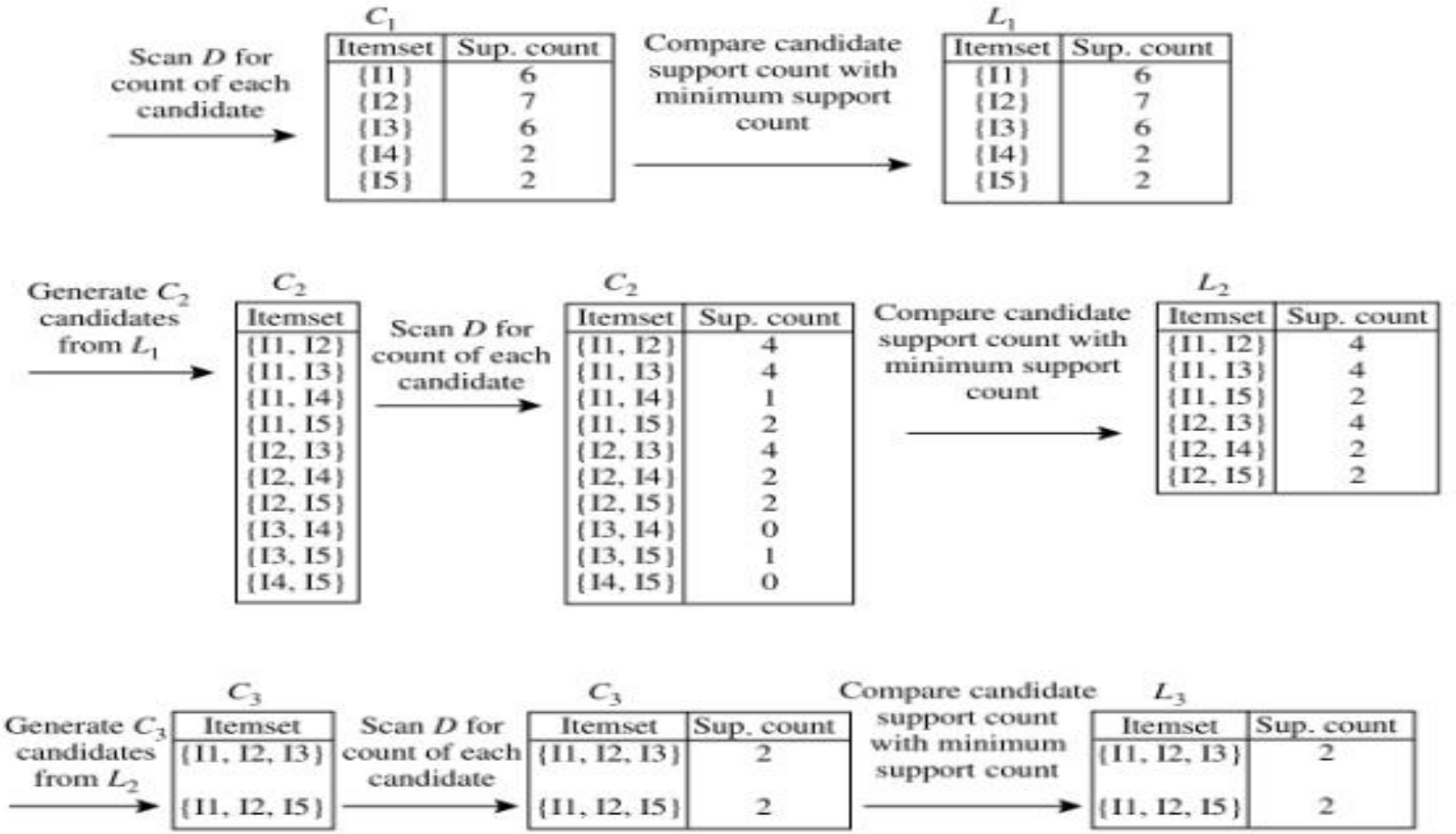
  3. Calculate the supports of the remaining candidate k-itemsets and eliminate any that fall below the specified minimum support threshold. The result is the frequent k-itemsets.

The Apriori Example

- Example, based on the AllElectronics transaction database, D.
- There are nine transactions in this database, that is, |D| = 9.

Transactional Data for an *AllElectronics* Branch

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



6.2 Generation of the candidate itemsets and frequent itemsets, where the minimum support count is 2.

Frequent pattern (FP)-growth algorithm

- The frequent pattern (FP)-growth algorithm was proposed by Han et al. [5] as a fast method for generating frequent itemsets.
- When generating frequent itemsets, FP-growth avoids generating a significant amount of candidates that are generated by apriori.
- FP-growth builds and operates on a data structure called the FP-tree and passes over the data set only twice.
- The FP-growth algorithm operates as follows:

- **FP-tree construction:**
- 1. Calculate the support of all 1-itemsets and prune any that fall below the minimum support threshold, specified by the user.
  2. Sort the remaining frequent 1-itemsets in descending global frequency.
  3. Build the FP-tree, tuple by tuple, from the first transaction to the last.

- **Deriving frequent itemsets from the FP-tree:**
  1. A conditional FP-tree is generated for each frequent itemset, and from that
  tree, the frequent itemsets with the processed item can be recursively derived.

2. Sequential Pattern Discovery

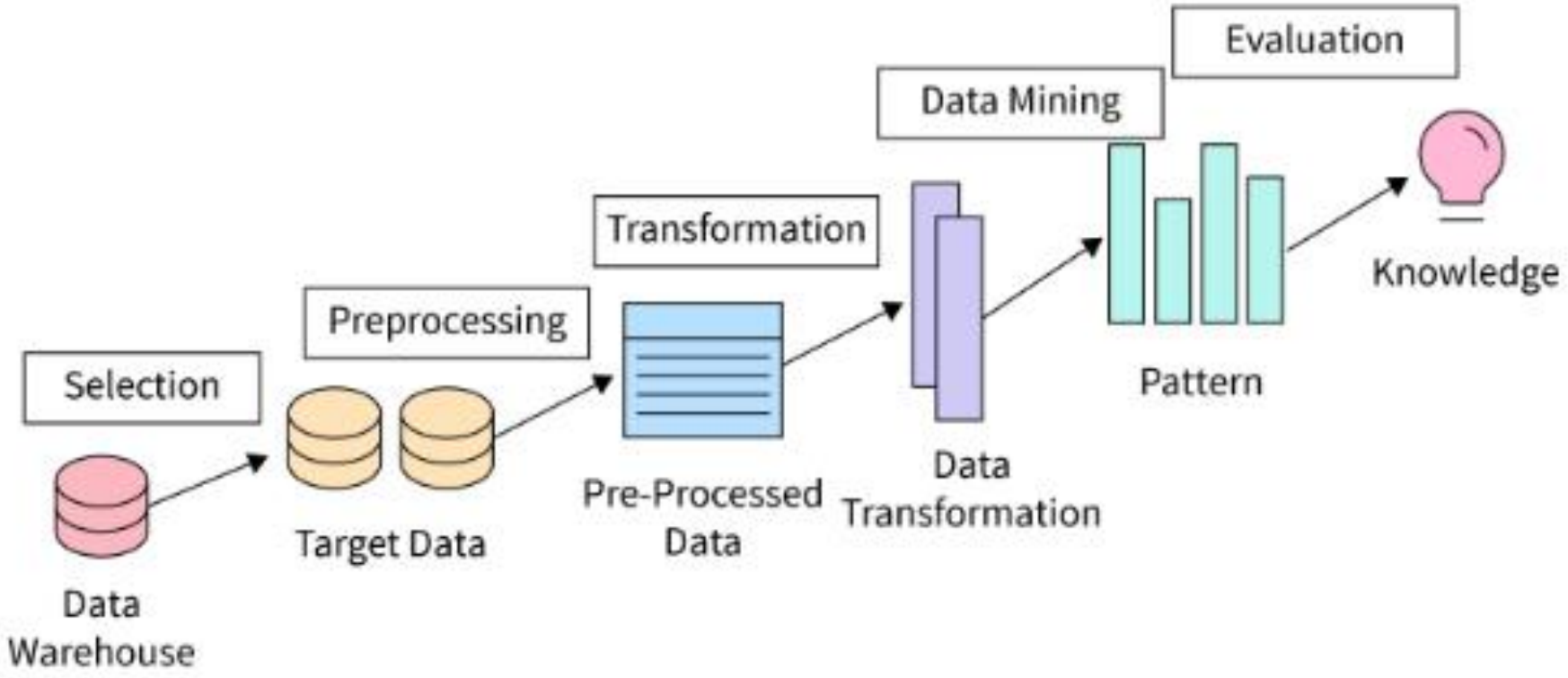
- Association rule mining becomes inefficient for temporal and sequential applications that require ordered matching rather than simple subset testing.
- Although comparatively less mature, sequential pattern discovery methods have been developed specifically for these types of problems.
- Potential medical applications of sequential pattern discovery include, but are not limited to, the following:

- • DNA sequence alignment
- • Motifs and tandem repeats in DNA sequences
- • Modeling clinical visit patterns
- • Analyzing infectious disease and micro-level disease observations
- • Studying the evolution of a disease and agent/host interaction patterns
- • Analysis of multi-omics experiment data involving temporal treatments

Knowledge Discovery and Data Mining

- Data mining, also known as knowledge discovery in databases, is defined as the process of exploring data to extract novel information (including patterns and relationships) using sophisticated algorithms.
- In data mining, the task is of unknown knowledge discovery, whereas machine learning performance is evaluated with respect to reproducing known knowledge.
- For instance, if you had a dataset of a patient's blood pressures, you can perform anomaly detection, which is considered a data mining task to identify previously unknown outliers. The task may make use of machine learning techniques, perhaps a k-means algorithm for cluster analysis, to identify the anomalous data and assist the algorithm's learning.

Knowledge Discovery and Data Mining



Knowledge Discovery and Data Mining

- **Data Selection** - The first step in the KDD process is identifying and selecting the relevant data for analysis. This involves choosing the relevant data sources, such as databases, data warehouses, and data streams, and determining which data is required for the analysis.
- **Data Preprocessing** - After selecting the data, the next step is data preprocessing. This step involves cleaning the data, removing outliers, and removing missing, inconsistent, or irrelevant data. This step is critical, as the data quality can significantly impact the accuracy and effectiveness of the analysis.
- **Data Transformation** - Once the data is preprocessed, the next step is to transform it into a format that data mining techniques can analyze. This step involves reducing the data dimensionality, aggregating the data, normalizing it, and discretizing it to prepare it for further analysis.

Knowledge Discovery and Data Mining

- **Data Mining** - This is the heart of the KDD process and involves applying various data mining techniques to the transformed data to discover hidden patterns, trends, relationships, and insights. A few of the most common data mining techniques include clustering, classification, association rule mining, and anomaly detection.
- **Pattern Evaluation** - After the data mining, the next step is to evaluate the discovered patterns to determine their usefulness and relevance. This involves assessing the quality of the patterns, evaluating their significance, and selecting the most promising patterns for further analysis.
- **Knowledge Representation** - This step involves representing the knowledge extracted from the data in a way humans can easily understand and use. This can be done through visualizations, reports, or other forms of communication that provide meaningful insights into the data.
- **Deployment** - The final step in the KDD process is to deploy the knowledge and insights gained from the data mining process to practical applications. This involves integrating the knowledge into decision-making processes or other applications to improve organizational efficiency and effectiveness.

Machine Learning:

Machine learning involves showing a large volume of data to a machine to learn, make predictions, find patterns, or classify data.

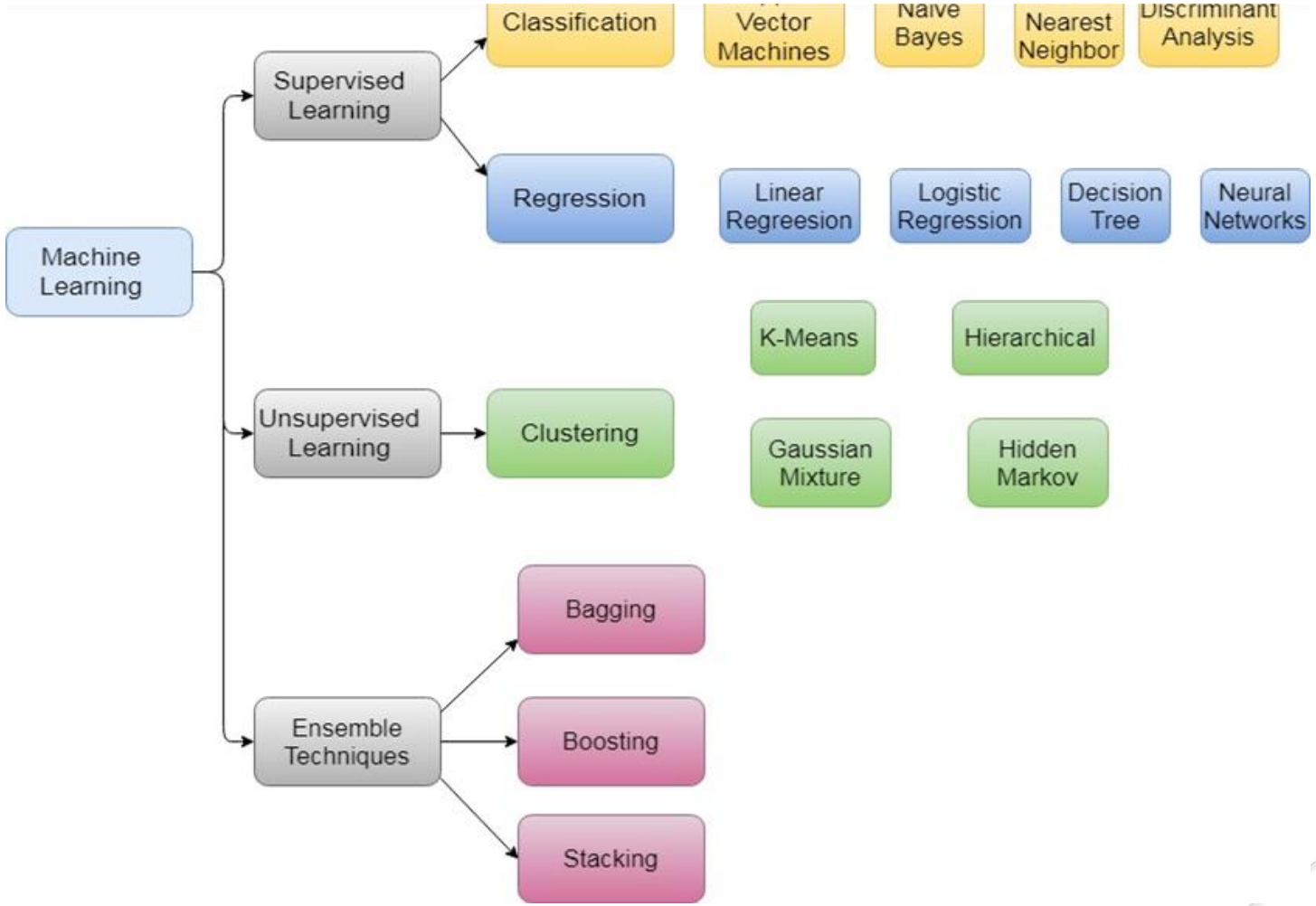


Figure 1. Classification of Machine Learning Algorithms

Supervised Machine Learning

- Supervised learning is defined as when a model gets trained on a **“Labelled Dataset”**. Labelled datasets have both input and output parameters.
- In **Supervised Learning** algorithms learn to map points between inputs and correct outputs. It has both training and validation datasets labelled.
- Within supervised machine learning, there are two main approaches:
- **Regression-based systems and Classification-based systems.**
- Some examples of use cases include:
- Predicting real estate prices
- Classifying whether bank transactions are fraudulent or not
- Finding disease risk factors

Classification

- **Classification** deals with predicting **categorical** target variables, which represent discrete classes or labels.
- For instance, classifying emails as spam or not spam, or predicting whether a patient has a high risk of heart disease.
- Classification algorithms learn to map the input features to one of the predefined classes.
- Algorithms that perform classification include decision trees, Naïve Bayesian, logistic regression, kNNs (k-nearest neighbors), support vector machine, and so forth.
- Common use cases of classification algorithms are the following:

  • Suggesting patient diagnosis based on health profile—that is, does patient have retinopathy?

  • Recommending the most suitable treatment pathway for a patient

  • Defining the period in the day in a clinic that there are peak admissions

  • Determining patient risk of hospital readmission

  • Classify imagery to identify disease—for example, determining whether a patient is overweight or not

  from a photograph

Decision Tree

- A **decision tree** is a flowchart-like structure used to make decisions or predictions.
- It consists of :
- **Root Node**: Represents the entire dataset and the initial decision to be made.
- **Internal Nodes**: Represent decisions or tests on attributes. Each internal node has one or more branches.
- **Branches**: Represent the outcome of a decision or test, leading to another node.
- **Leaf Nodes**: Represent the final decision or prediction. No further splits occur at these nodes.

They are simple in the way that outcomes can be expressed as a set of rules.

Logistic Regression

- Logistic regression is often used for binary classification problems.
- For example, logistic regression can be used to predict whether a patient will experience an adverse event given a particular medication; whether a patient is likely to be readmitted to a hospital or not; or whether a patient has a particular disease diagnosis.
- Unlike linear regression, the output of the model comes in the form of a probability between 0 and 1.
- The predicted output is generated by log-transforming the x input and using the logistic function h(x) = 1/(1 + e^ -x).
- A specified threshold is used to transform this probability into a binary classification.
- Logistic regression uses the sigmoid function, an S-shaped curve (see Figure 4-5) that can take any continuous value and map it into a probability value between 0 and 1 for a default class.

Regression

- Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables.
- These are used to predict continuous output variables, such as market trends, weather prediction, etc.
- Regression algorithms include linear regression, regression trees, support vector machines, k nearest neighbor, and perceptrons.
- Common use cases of regression algorithms are the following:

  • Estimating patient HbA1c from previous blood glucose readings (time series forecasting)
  • Predicting the days before hospital readmission (which may be received as a time-series regression problem)
  • Approximating patient mortality risk based on clinical data
  • Forecasting the risk of developing health complications
  • Forecasting when the office printer will be queued with requests from other employees

Linear Regression

- Linear regression is a continuous statistical technique to understand the relationship between input (x: predictors, explanatory or independent variables) and output (y: response or dependent variable).
- Here, y be calculated from a linear combination of input variables.
- The aim is to place a line of best fit through all of the data points, which enables easy understanding of predictions by minimizing margin or residual variation.
- where yi is the actual observed response value; yˆi is the predicted value of response variable; and their subtraction is the residual variation or difference between observed and predicted values of y

SVM

- Support vector machine (SVM) is a nonprobabilistic binary linear classifier used with both classification and regression problems.
- The algorithm finds a hyperplane, or line of best fit, between two classes determined by the support vectors.
- Support vectors are the data points nearest the hyperplane that would alter the position of the hyperplane if removed.
- The greater the margin value, or distance from the data points to the hyperplane, the more confidence there is that the data is appropriately classified.
- The line of best fit is learned from an optimization procedure that seeks to maximize the margin.

Unsupervised Learning

- As its name suggests, there is no need for supervision.
- It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.
- the models are trained with the data that is neither classified nor labelled.
- **The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences.**
- Machines are instructed to find the hidden patterns from the input dataset.
- Unsupervised learning is composed of two main problem concepts: **clustering and association.**

Clustering

- Clustering refers to the process of discovering relationships within the data.
- Clustering is used for a variety of healthcare uses including the following:
  - Grouping patients of similar profiles together for monitoring
  - Detecting anomalies or outliers in claims or transactions
  - Defining treatment groups based on medication or condition
  - Detecting activity through motion sensors

K-Means Clustering

- K-means aims to find k similar groups within the data.
- K-means is an iterative algorithm that calculates the centroids of the defined k clusters, with all training data assigned to a cluster.
- Data points are assigned by the distance between the data point and its centroid.
- A cluster centroid is a collection of attribute values that define the resulting clusters.
- arg , min c Ciidist c x Î( ) , 2
- where ci is the collection of centroids in set C, and distance (dist) is standard Euclidean distance.

K-Means Clustering

- Steps of the **K-means algorithm** include the following:

  - Step 1: Set a value of k. Here, let us take k = 2.
    - Assign each data point randomly to any of the k = 2 clusters.
    - Determine the centroid for each of the clusters.
  - Step 2: For each data point, associate it to the closest cluster centroid.
  - Step 3: Recalculate centroids for the new clusters.
  - Step 4: Continue
    - this process until the centroid clusters remain unchanged.

Association

- Association rule learning methods extract rules that best explain perceived relationships between variables in data.
- These rules can discover useful associations in large multidimensional datasets that can be used to drive and optimize.
- In healthcare, in particular, associative symptoms can be understood to predict better and diagnose disease and adverse events.
- Potential adverse effects based on medication and associative patient comorbidity pathways could lead to improved care and treatment pathways.
- Explain Support, Confidence, Lift
- Apriori Algorithm

Multi-Classifier Decision Fusion

- Each machine learning paradigm has its own strengths and weaknesses stemming from its underlying theory, mathematical underpinning, and assumptions about the data and/or decision space.
- Different algorithms achieve different levels of accuracy on the same data, but specific classifiers can develop levels of expertise on portions of the decision space.
- Since different learning methods typically converge to different solutions, combining decisions from different learning paradigms can be leveraged for improved accuracy.
- Classifier combination is one such area in machine learning that has offered advances in classification accuracy for complex data sets.
- Generally, when predictions from multiple classifiers are combined, they are said to form an ensemble that is then used to classify new examples. Figure 3.2 illustrates the concept of combining decisions from N classifiers to arrive at a fused, consensus decision and confidence.

Multi-Classifier Decision Fusion

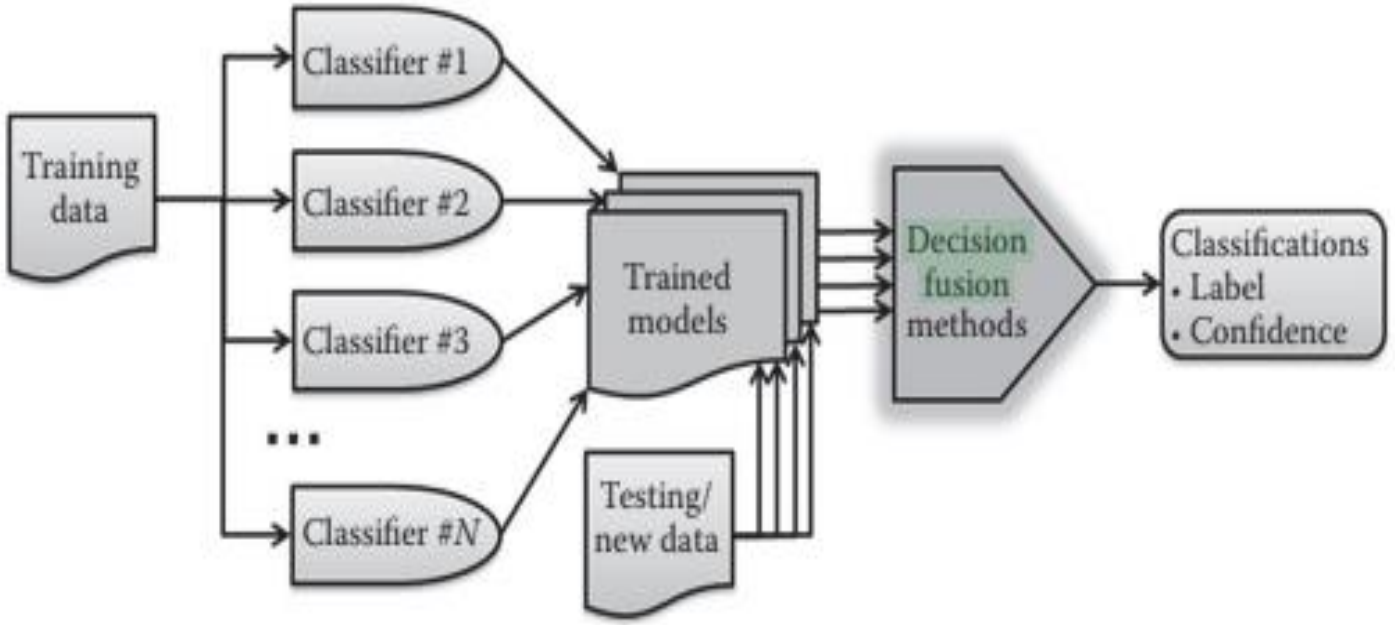


FIGURE 3.2

Decision fusion process for a multi-classifier system, where decisions are collected from a set of classifiers and combined in some way to arrive at a single classification label and confidence for each instance.

Multi-Classifier Decision Fusion

- When developing a multi-classifier system, its members can be a mixture of weak (i.e.,high-error-rate) and strong (i.e., low-error-rate) classifiers.
- Weak classifiers are typically simple to create, at the expense of their accuracy on complex data sets.
- Strong classifiers are typically time consuming and expensive to create, as their parameters are fine-tuned and tweaked for optimal performance.
- Combining weak/strong or homogeneous/heterogeneous classifiers offers the benefit of encompassing different levels of expertise and knowledge bases.

Multi-Classifier Decision Fusion- Types

- Various approaches exist for combining decisions from multiple classifiers:

1. **Voting methods**:

Each entity classifies (votes)that an input instance belongs to one of multiple classes. These votes are tallied in one of a number of ways to arrive at the final decision of the classifiers being fused in the form of a single class label. Examples of pure voting methods are majority vote, maximum confidence, average confidence, and product of confidences.

1. **Accuracy-driven voting/Weighted voting:**

A popular accuracy-driven voting method is weighted voting, where each classifier is assigned a voting weight (usually proportional to its operational accuracy), and the class with the highest weighted vote is used for classification.

1. **Data manipulation methods:**

Which manipulate the training and testing data to attempt to achieve optimal classification accuracy, are by far the most popular means to train and combine multiple classifiers.

Data manipulation methods - Types

- **1. Input decimation:**

  Those features that have the least effect on the output can be selected for removal without compromising overall classifier performance.
  One example is to have one classifier per class, where each classifier only uses the features with high correlation to that class.

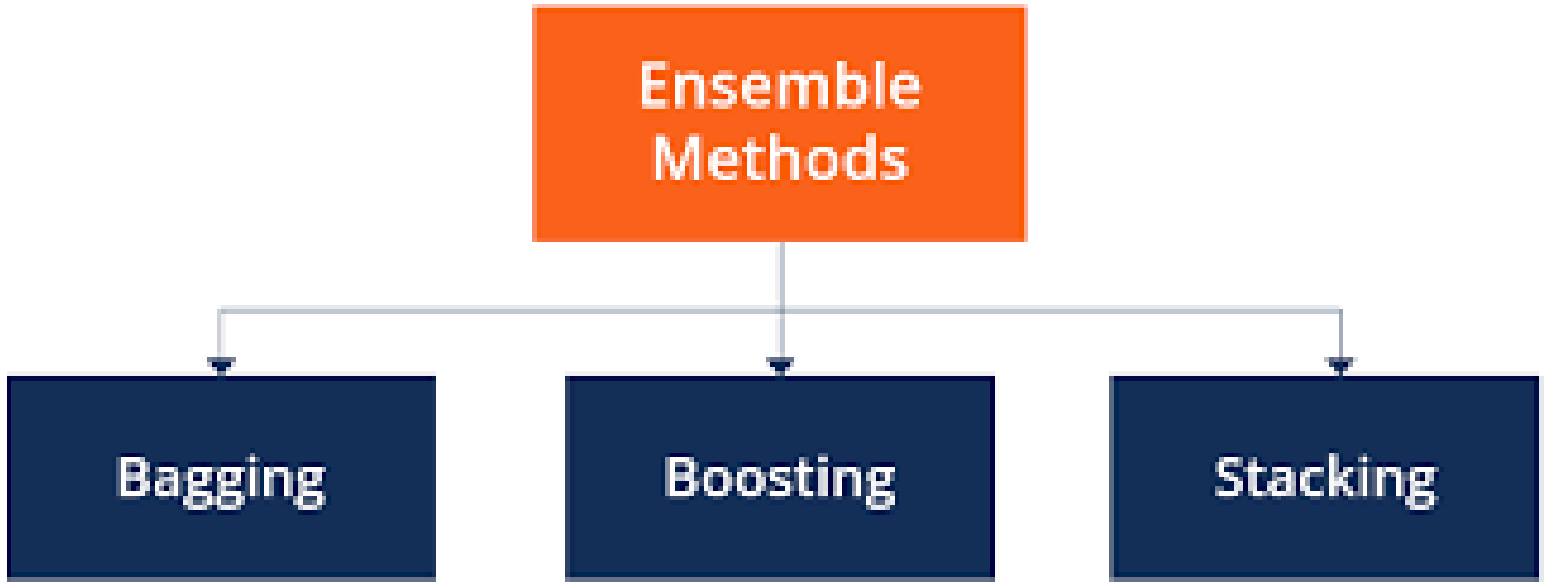
- **2. Boosting:**
  Boosting adaptively changes the training set distribution based on performance of previous classifiers, attempting to reduce both bias and variance.
  At each iteration, instances incorrectly classified are given greater weight in the next iteration.

- **3. Bagging:**
  Bagging creates a family of classifiers by training on stochastically different portions of the training set.
  N training “bags” are initially created, each obtained by taking a training set of size S and sampling the training set S times with replacement.
  Some instances could occur multiple times, while others may not appear at all.
  Each bag is then used to train a classifier, and classifiers are then

Ensemble Learning

- The ensemble methods in machine learning combine the insights obtained from multiple learning models to facilitate accurate and improved decisions.
- In learning models, noise, variance, and bias are the major sources of error. The ensemble methods in machine learning help minimize these error-causing factors, thereby ensuring the accuracy and stability of machine learning (ML) algorithms.
- Ensemble learning improves a model's performance in mainly three ways:
  - By reducing the variance of weak learners
  - By reducing the bias of weak learners,
  - By improving the overall accuracy of strong learners.

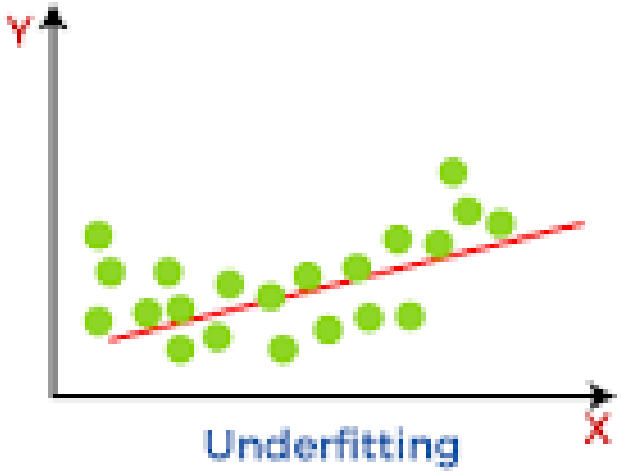
Ensemble Learning

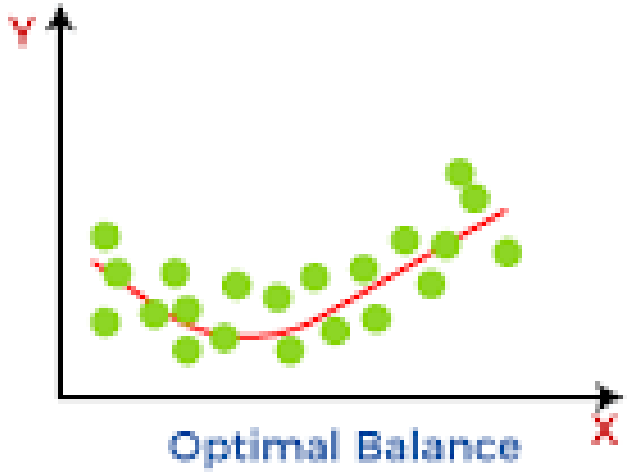


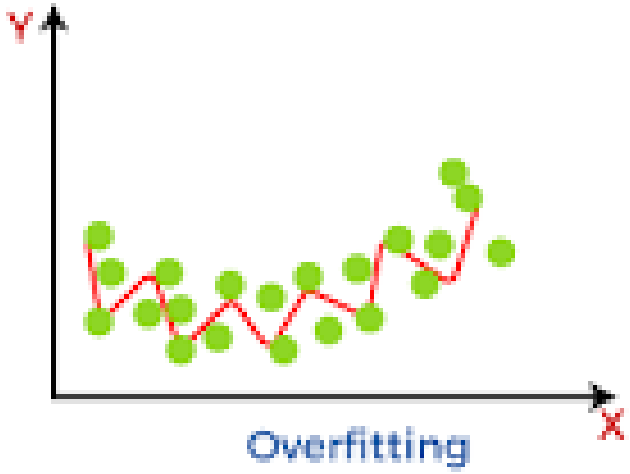
- Bagging is used to reduce the variance of weak learners.
- Boosting is used to reduce the bias of weak learners.
- Stacking is used to improve the overall accuracy of strong learners.

Ensemble Learning

Bias-Variance Trade-off







Ensemble Learning

1.Max Voting
2.Averaging
3.Weighted Averaging

Max Voting: The max voting method is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The predictions which we get from the majority of the models are used as the final prediction. For example, when you asked 5 of your colleagues to rate your movie (out of 5); we'll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4.

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final rating
5	4	5	4	4	4

Ensemble Learning

Averaging: In this method, we take an average of predictions from all the models and use it to make the final prediction. Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.

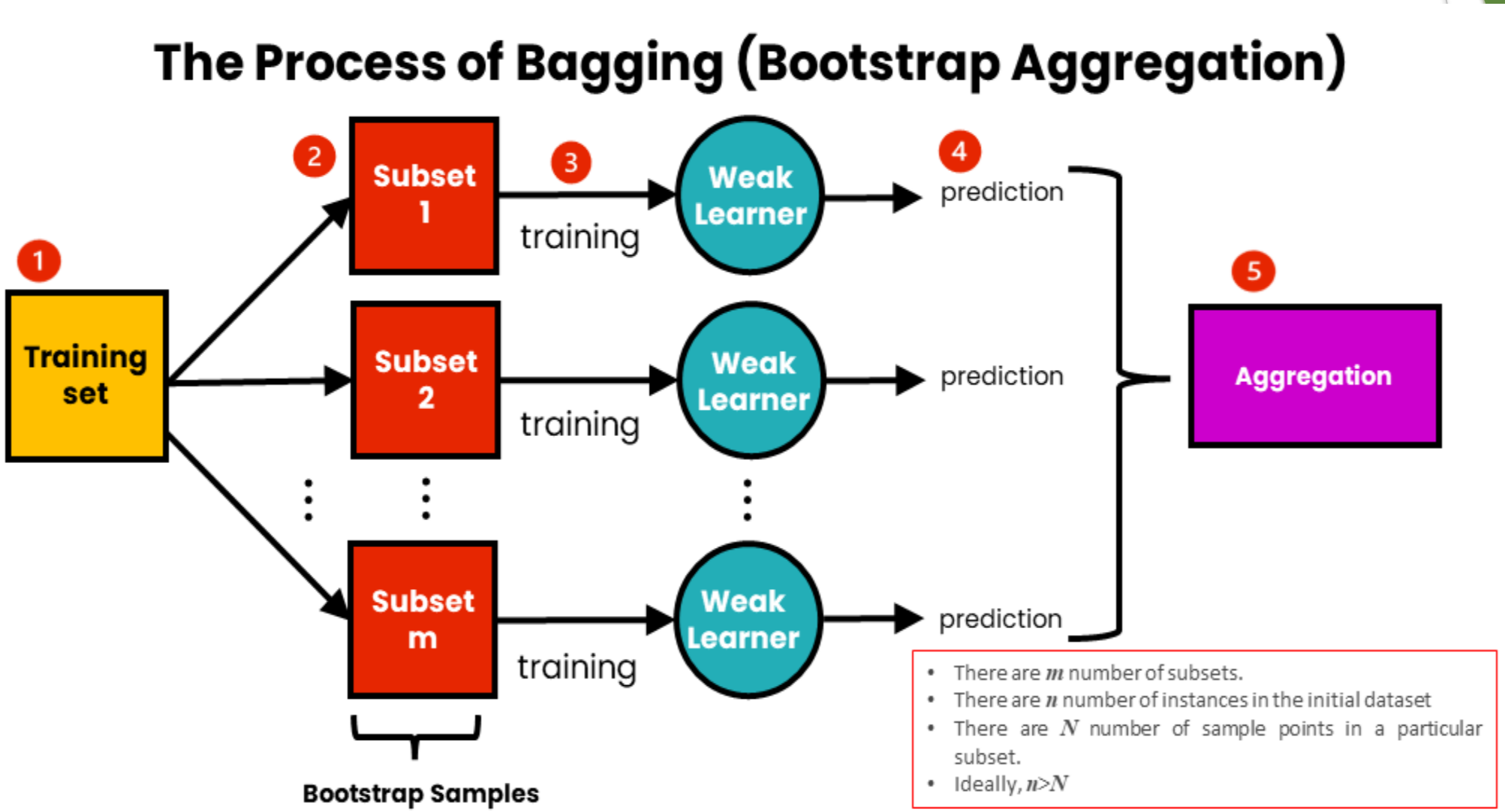
- For example, refer previous example

i.e. (5+4+5+4+4)/5 = 4.4

Weighted Average: This is an extension of the averaging method. All models are assigned different weights defining the importance of each model for prediction. For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.

- The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

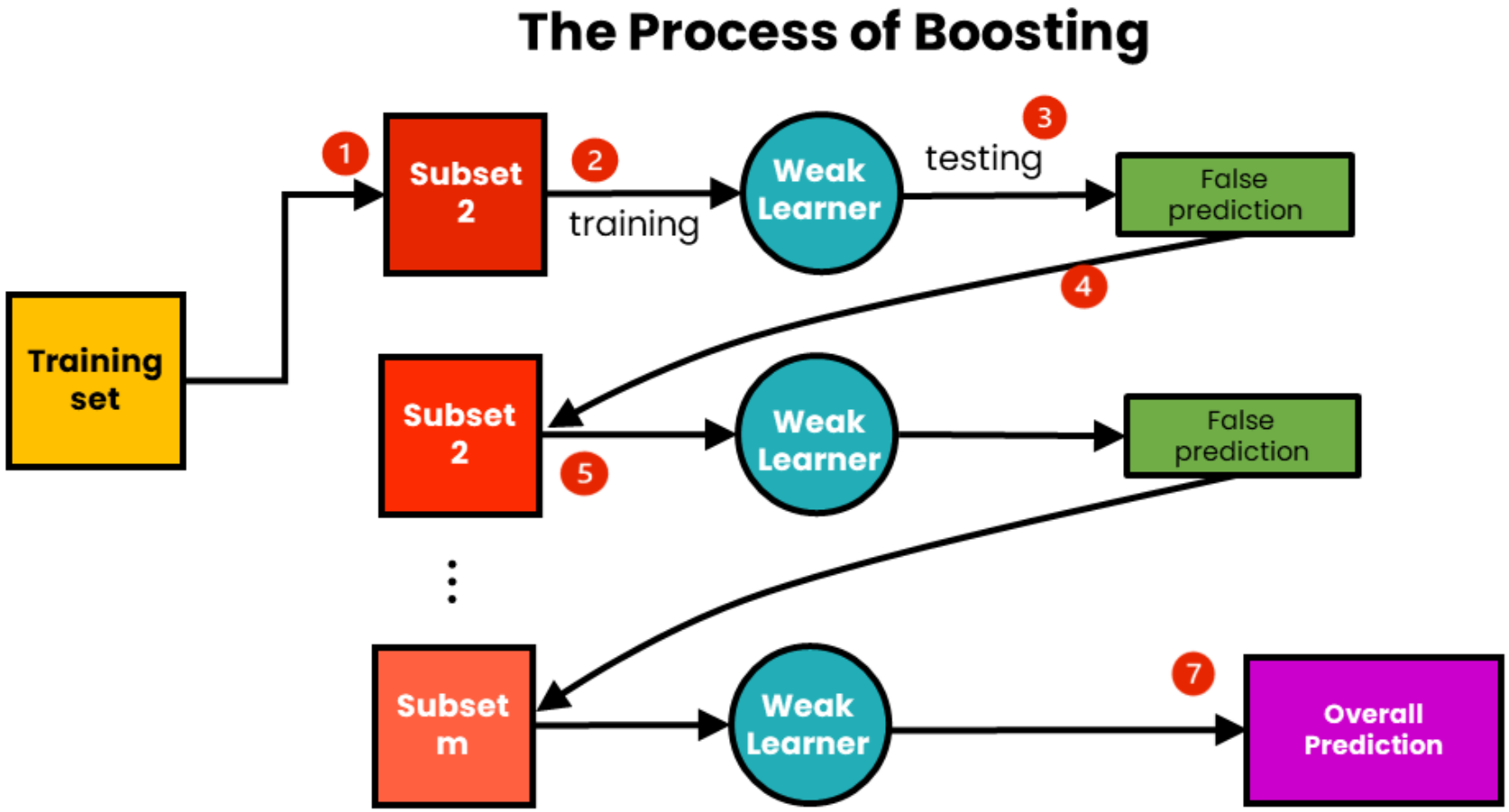
Ensemble Learning: Bagging(Bootstrap Aggregation)



Ensemble Learning: Bagging(Bootstrap Aggregation)

- **The steps of bagging are as follows:**
- We have an initial training dataset containing n-number of instances.
- We create a m-number of subsets of data from the training set. We take a subset of N sample points from the initial dataset for each subset. Each subset is taken with replacement. This means that a specific data point can be sampled more than once.
- For each subset of data, we train the corresponding weak learners independently. These models are homogeneous, meaning that they are of the same type.
- Each model makes a prediction.
- The predictions are aggregated into a single prediction. For this, either max voting or averaging is used.

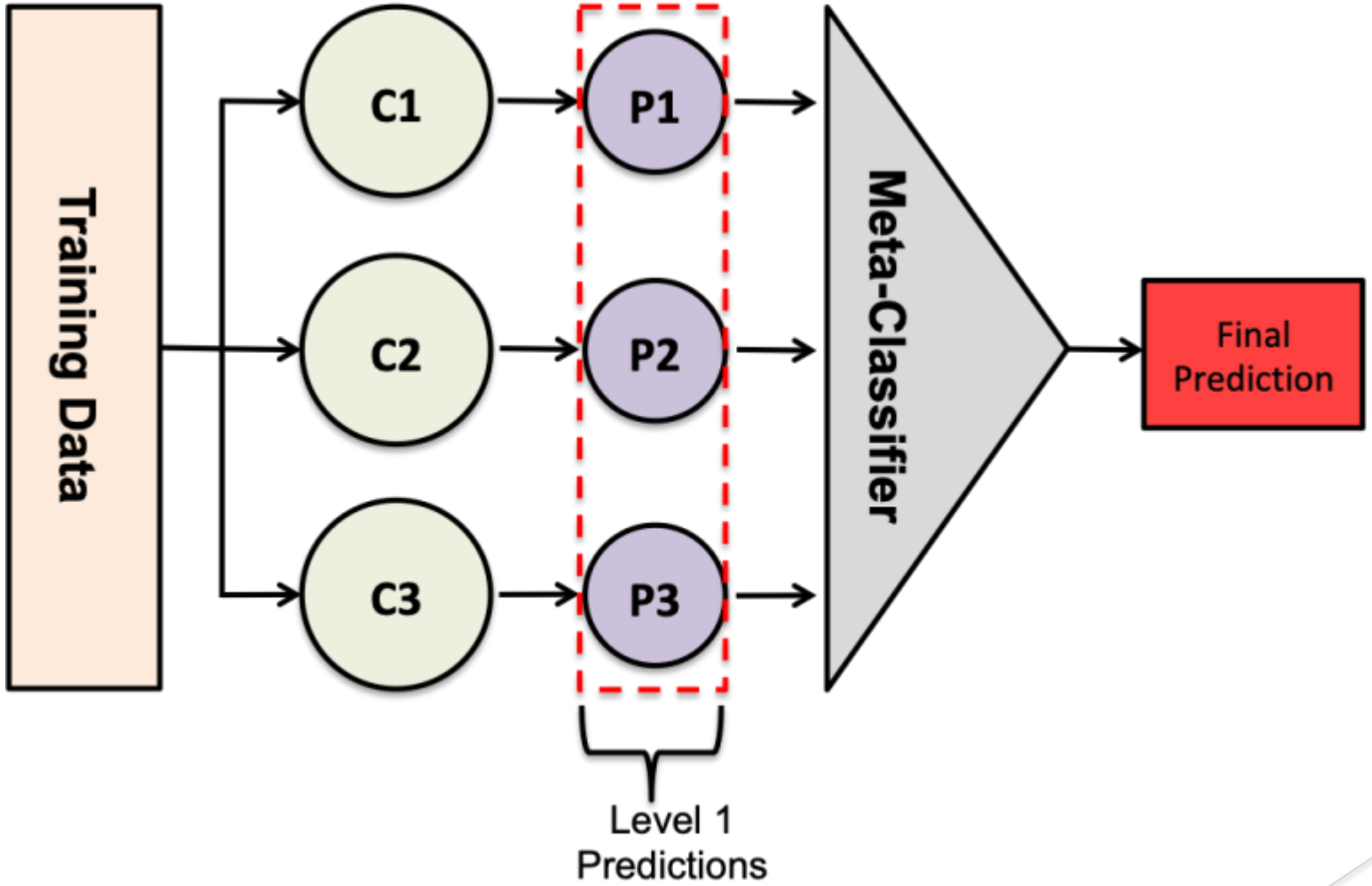
Ensemble Learning: Boosting



Ensemble Learning: Boosting

- **Boosting works with the following steps:**
- We sample m-number of subsets from an initial training dataset.
- Using the first subset, we train the first weak learner.
- We test the trained weak learner using the training data. As a result of the testing, some data points will be incorrectly predicted.
- Each data point with the wrong prediction is sent into the second subset of data, and this subset is updated.
- Using this updated subset, we train and test the second weak learner.
- We continue with the following subset until the total number of subsets is reached.
- We now have the total prediction. The overall prediction has already been aggregated at each step, so there is no need to calculate it.

Ensemble Learning: Stacking



* C1, C2, and C3 are considered level 1 classifiers.

Ensemble Learning: Stacking

The steps of Stacking are as follows:

- We use initial training data to train m-number of algorithms.
- Using the output of each algorithm, we create a new training set.
- Using the new training set, we create a meta-model algorithm.
- Using the results of the meta-model, we make the final prediction. The results are combined using weighted averaging.

When to use Bagging vs Boosting vs Stacking?

	Bagging	Boosting	Stacking
Purpose	Reduce Variance	Reduce Bias	Improve Accuracy
Base Learner Types	Homogeneous	Homogeneous	Heterogeneous
Base Learner Training	Parallel	Sequential	Meta Model
Aggregation	Max Voting, Averaging	Weighted Averaging	Weighted Averaging

Meta Learning

- Meta-learning is essentially learning how to learn. Meta-learning algorithms learn from the outputs of other learning algorithms which learn from data.
- Meta-learning algorithms typically refer to ensemble learning algorithms like stacking that learn how to combine the predictions from ensemble members.
- Meta-learning also refers to algorithms that learn how to learn across a suite of related prediction tasks, referred to as multi-task learning.

Importance of Meta Learning

Machine learning algorithms have some problems, such as

- The need for large datasets for training
- High operating costs due to many trials/experiments during the training phase
- Experiments/trials take a long time to find the best model that performs best for a given data set.

Meta-learning can help machine learning algorithms deal with these challenges by optimizing and finding learning algorithms that perform better.

Meta Learning Approaches

Metric-based meta-learning: This approach basically aims to find a metric space.

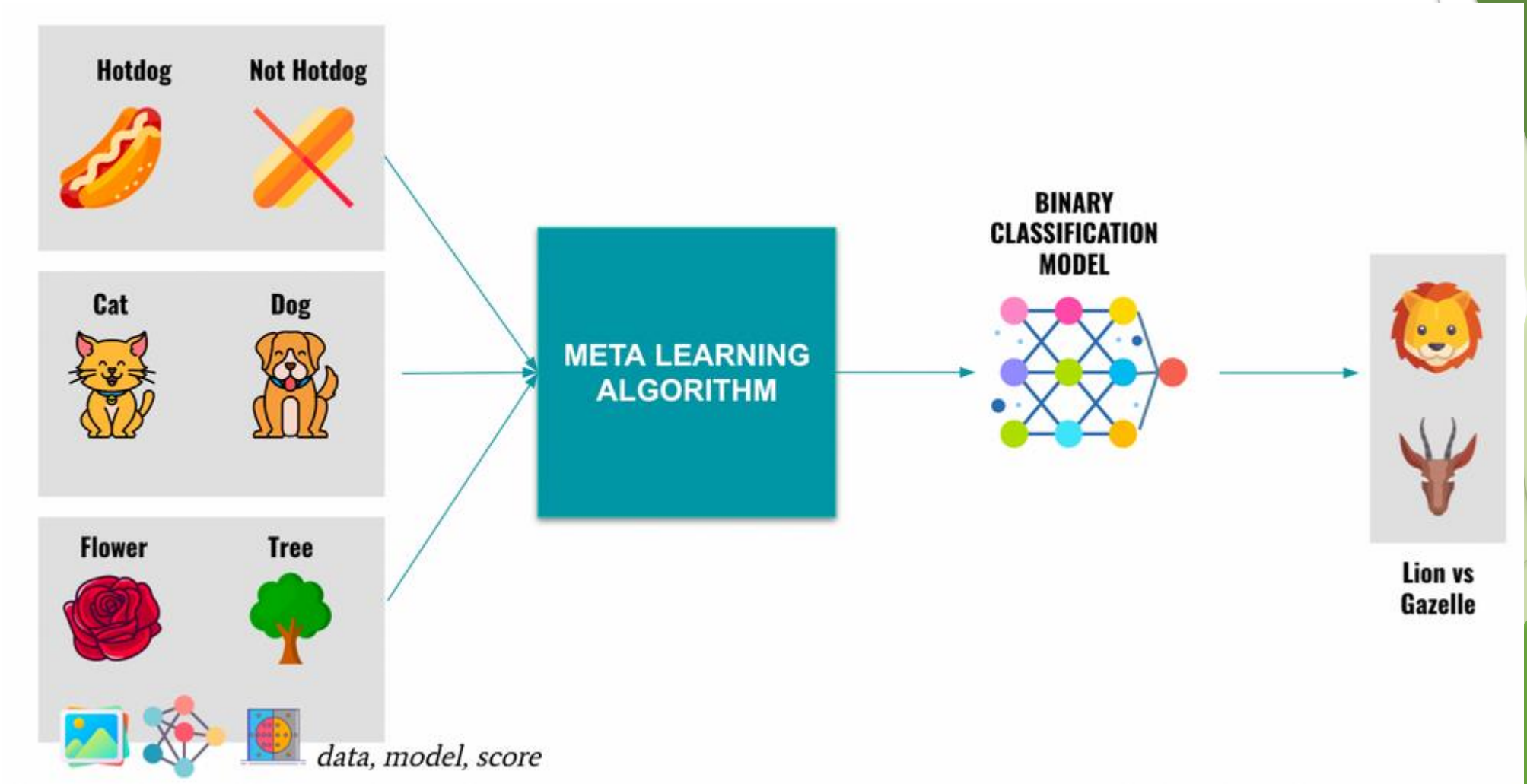
- It is similar to the nearest neighbor algorithm which measures the similarity or distance to learn the given examples.
- The goal is to learn a function that converts input examples into a metric space with labels that are similar for nearby points and dissimilar for far-off points.
- The success of metric-based meta-learning models depends on the selection of the kernel function, which determines the weight of each labeled example in predicting the label of a new example.

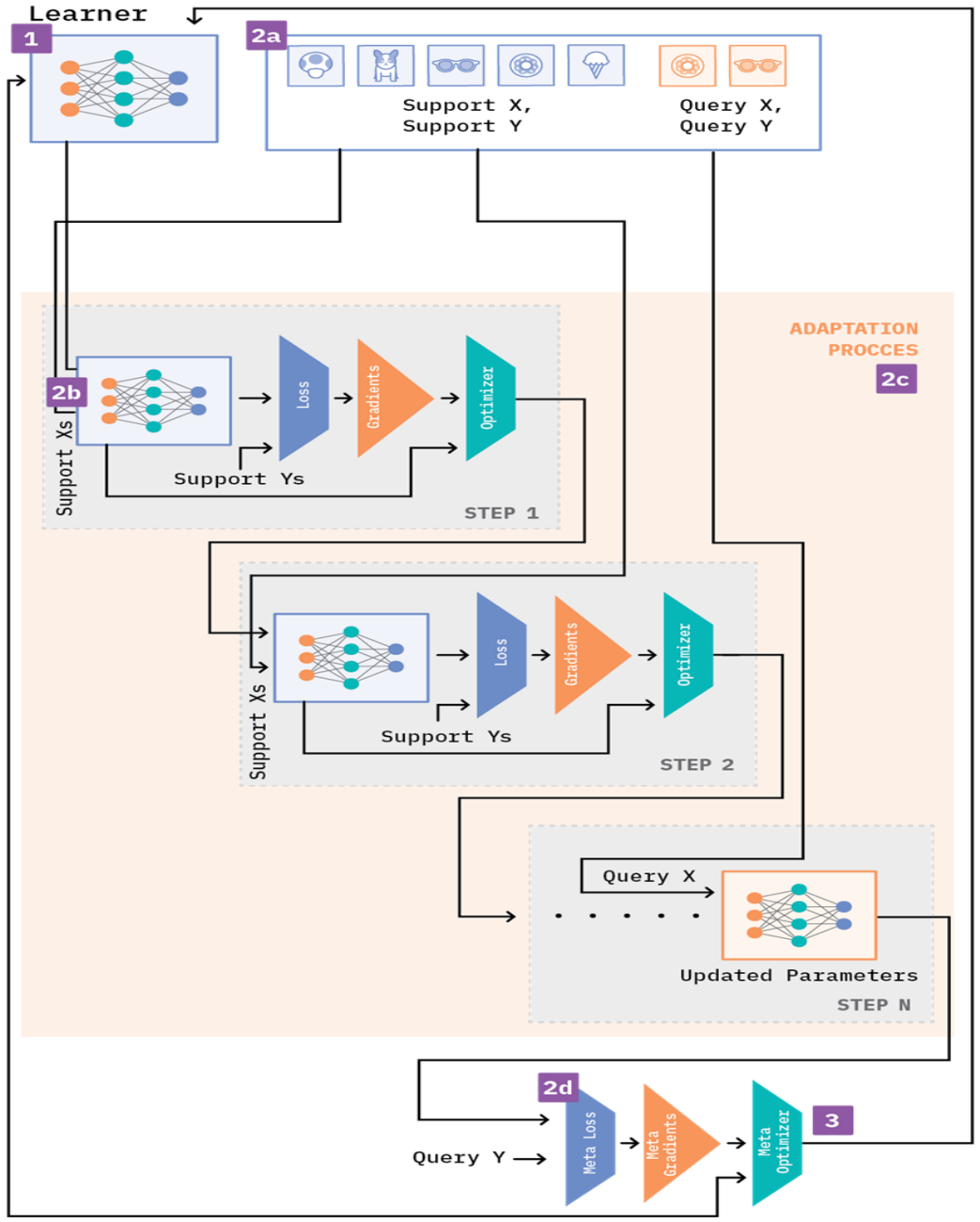
Model-Agnostic Meta Learning (MAML)

- It is an optimization-based meta-learning framework that enables a model to quickly adapt to new tasks with only a few examples by learning generalizable features that can be used in different tasks.

- In MAML, the model is trained on a set of meta-training tasks, which are similar to the target tasks but have a different distribution of data. The model learns a set of generalizable parameters that can be quickly adapted to new tasks

Model-Agnostic Meta Learning (MAML)




Learner
1
2a
Support X,
Support Y
Query X,
Query Y
ADAPTATION
PROCCES
2c
2b
Loss
Gradients
Optimizer
Support Xs
Support Ys
STEP 1
Loss
Gradients
Optimizer
Support Xs
Support Ys
STEP 2
Query X
Updated Parameters
STEP N
2d
Meta Loss
Meta Gradients
Meta Optimizer
3
Query Y

Model-Agnostic Meta Learning (MAML)

- Step 1: Randomly initialize the learner
- Step 2: Repeat the entire process from Step 2.a to Step 3 for all the episodes of the meta-training dataset (or for a certain number of epochs) until the learner converges to a good set of “meta-parameters”
  - Step 2.a: Sample a batch of episodes from the meta-training dataset
  - Step 2.b: Initialize the adapter with the learner’s parameters
  - Step 2.c: While number of inner training steps is not equal to zero
    - Step 2.c.1: Train the adapter based on the support set(s) of the batch, compute the loss and the gradients, and update the adapter’s parameters
  - Step 2.d: Use the updated parameters of the adapter to compute the “meta-loss” based on the query set(s) of the batch
- Step 3: Compute the “meta-gradients”, followed by the “meta-parameters” based on the “meta-loss,” and update the learner’s parameters

Recurrent Neural Network (RNN)

- RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

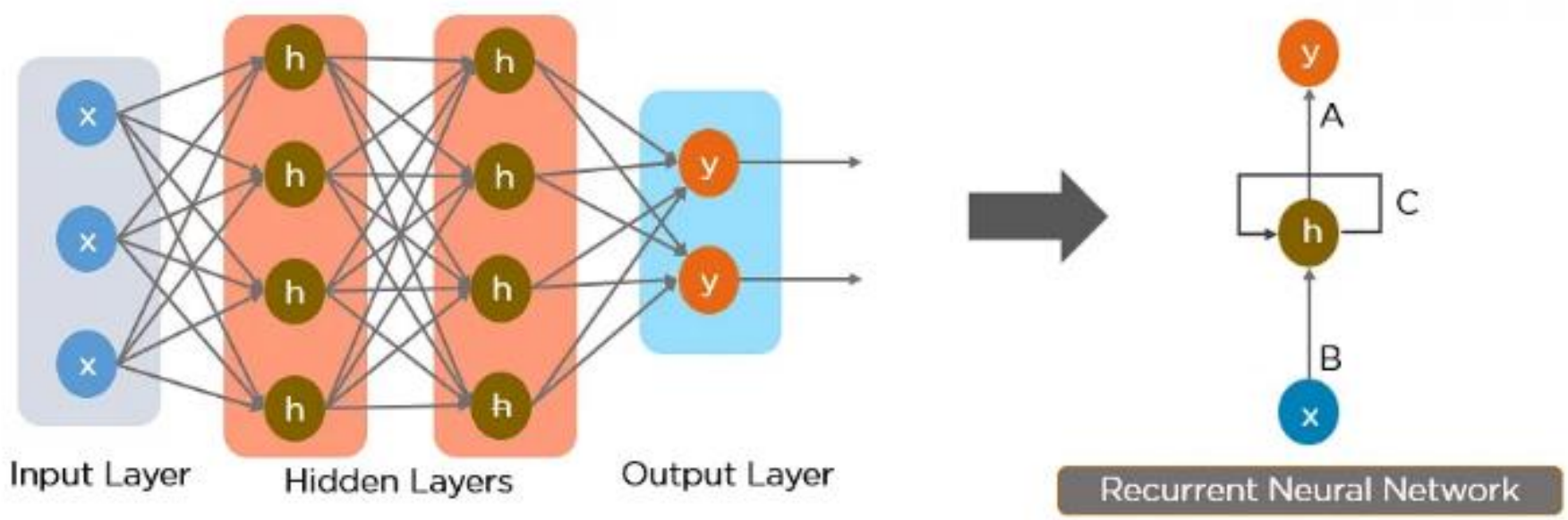


Fig: Simple Recurrent Neural Network

Recurrent Neural Network (RNN)

- Recurrent neural networks (RNN) are a class/type of artificial neural networks, and they are applied to different machine learning problems, such as problems that have sequential data or time series data.

- RNN models are commonly used for language translation, speech recognition, and handwriting recognition tasks.

- In meta learning, RNNs are used as an alternative to creating a recurrent model which can gather data sequentially from datasets and process them as new inputs.

Why Recurrent Neural Network (RNN)?

RNN were created because there were a few issues in the feed-forward neural network:

- Cannot handle sequential data
- Considers only the current input
- Cannot memorize previous inputs

The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory.

Advantages of Meta Learning

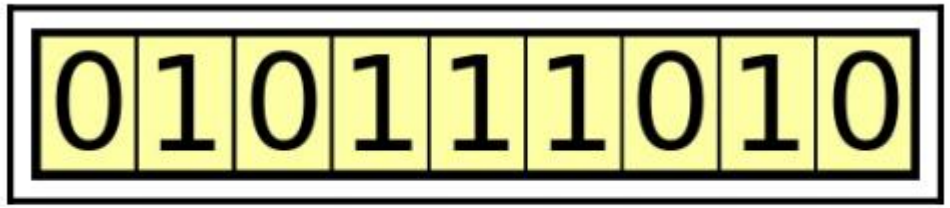
- **Higher model prediction accuracy:** Optimization of learning algorithms: For example, optimization of hyperparameters to achieve the best results.
  - Helps to learn algorithms better adapt to changing conditions
  - Identifying clues for designing better learning algorithms
- **The faster and cheaper training process**
  - Supporting learning from fewer examples
  - Increase the speed of learning processes by limiting the necessary experiments
- **Building more generalized models**: learning to solve many tasks, not just one task: meta-learning does not focus on training one model on one specific data set

Evolutionary Algorithm: Genetic Algorithms

- Genetic Algorithms are search algorithms inspired by Darwin's Theory of Evolution in nature.
- By simulating the process of natural selection, reproduction and mutation, the genetic algorithms can produce high-quality solutions for various problems including search and optimization.
- According to Darwin's theory of evolution, an evolution maintains a population of individuals that vary from each other (variation). Those who are better adapted to their environment have a greater chance of surviving, breeding, and passing their traits to the next generation (survival of the fittest).

Basic Terminologies of Genetic Algorithms

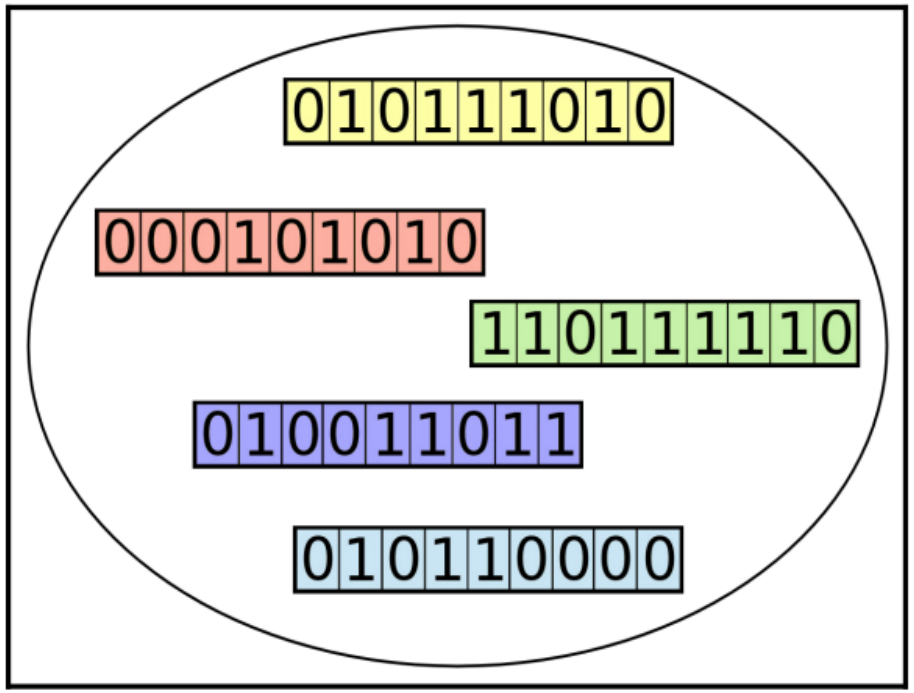
- Chromosome/Individual: A chromosome is a collection of genes. For example, a chromosome can be represented as a binary string where each bit is a gene.



Simple binary-coded chromosome

Basic Terminologies of Genetic Algorithms

- Population:Since an individual is represented as a chromosome, a population is a collection of such chromosomes.



A population of individuals represented by binary-coded chromosomes

Basic Terminologies of Genetic Algorithms

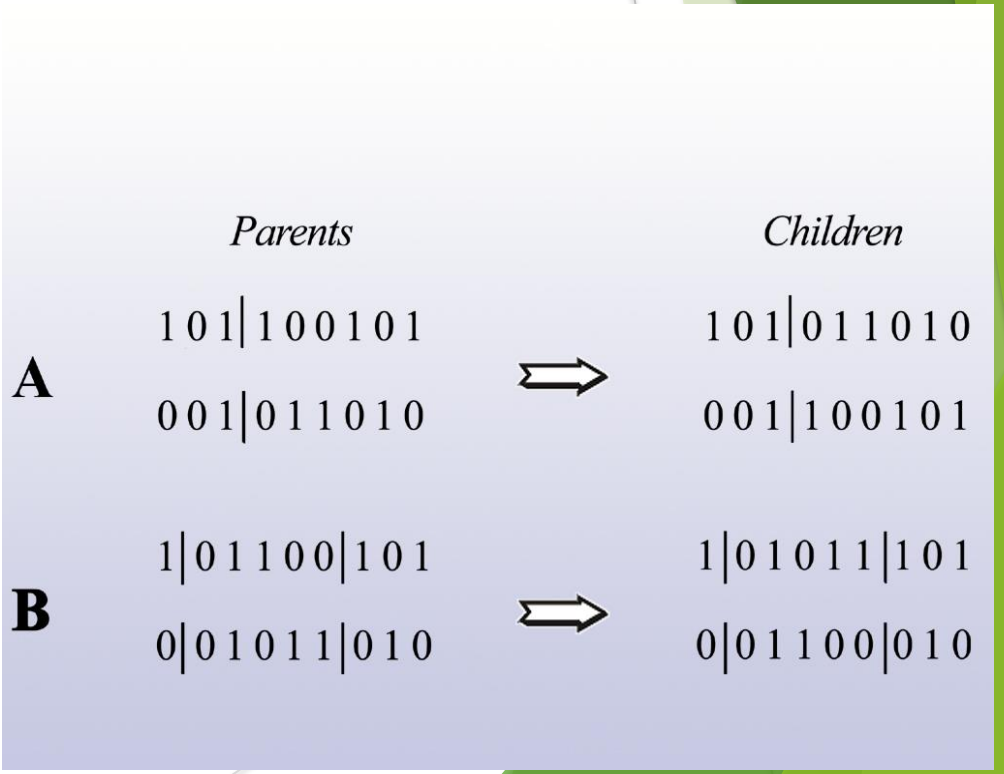
- Fitness Function:In every iteration, the individuals are evaluated based on their fitness scores which are computed by the fitness function.
- Individuals who achieve a better fitness score represent better solutions and are more likely to be chosen to crossover and passed on to the next generation.

Basic Terminologies of Genetic Algorithms

- Selection: After calculating the fitness of every individual in the population, a selection process is used to determine which of the individuals in the population will get to reproduce and create the offspring that will form the next generation.
- Types of selection methods available,
- Roulette wheel selection
- Tournament selection
- Rank-based selection

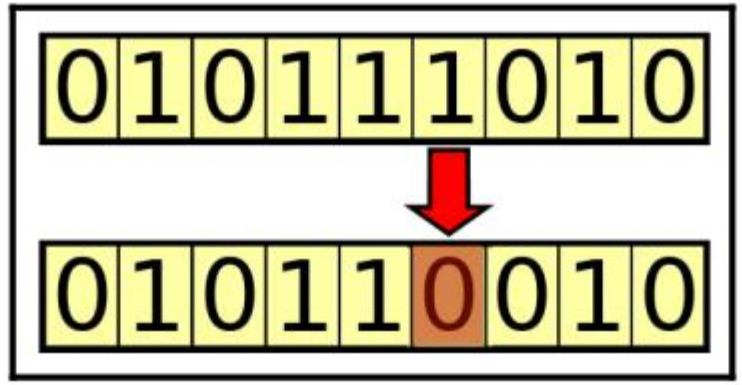
Basic Terminologies of Genetic Algorithms

- Crossover:Generally, two individuals are chosen from the current generation and their genes are interchanged between two individuals to create a new individual representing the offspring. This process is also called mating or crossover.
- Types of crossover methods available,
- One point crossover
- Two-point crossover
- Uniform crossover



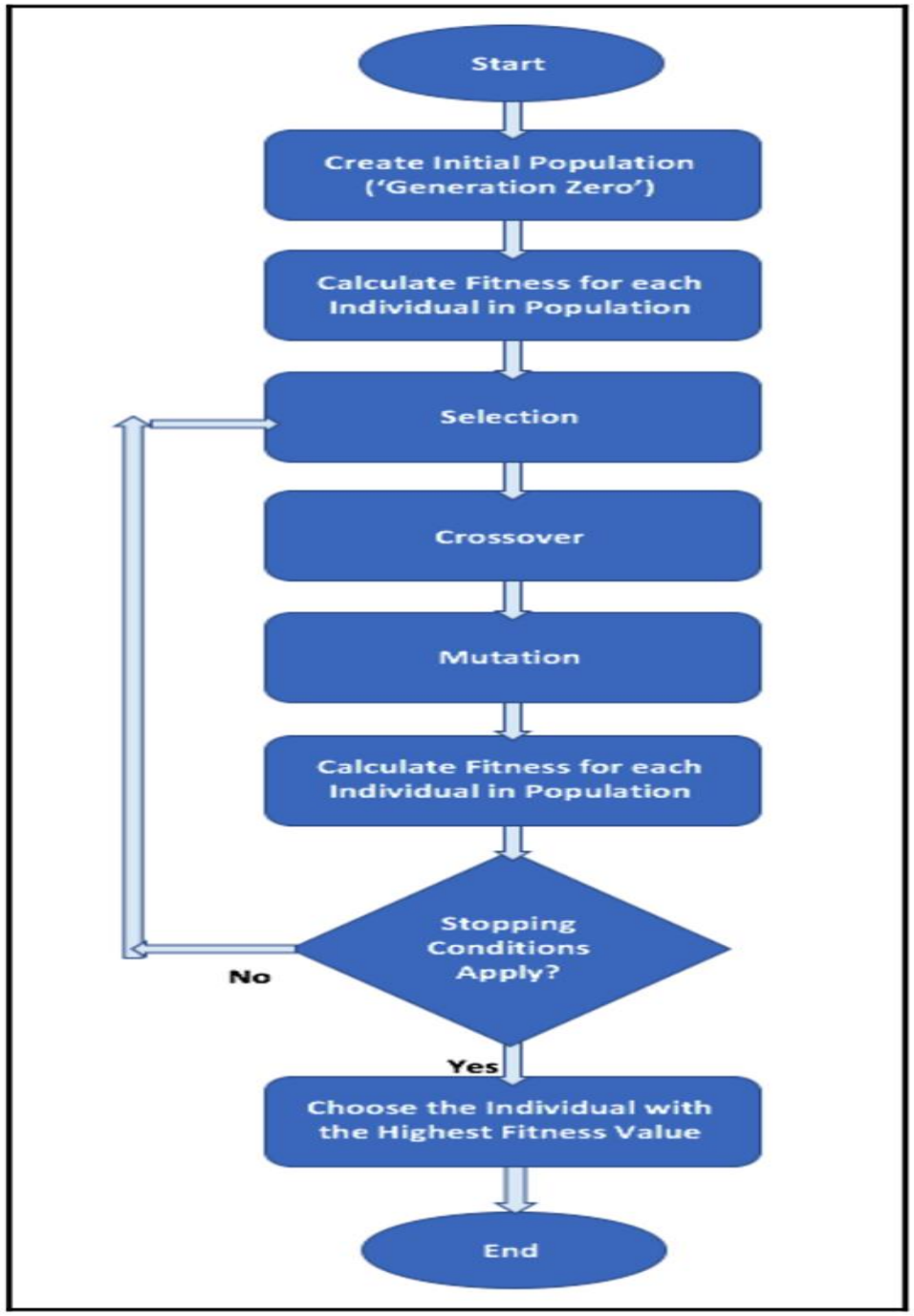
Basic Terminologies of Genetic Algorithms

- Mutation:The mutation is a random change in a chromosome to introduce new patterns to a chromosome. For example, flipping a bit in a binary string.
- Types of mutation methods available,
- Flip bit mutation
- Gaussian mutation
- Swap mutation



Mutation operator applied to a binary-coded chromosome

General workflow a simple genetic algorithm



Basic flow of a genetic algorithm

Deep Learning

- Deep learning can be considered as a subset of machine learning. It is a field that is based on learning and improving on its own by examining computer algorithms.
- In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer.
- The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

Deep Learning

- Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning.
- **Supervised Machine Learning:** the neural network learns to make predictions or classify data based on the labeled datasets. Here we input both input features along with the target variables.
- Deep learning algorithms like Convolutional neural networks, Recurrent neural networks are used for many supervised tasks like image classifications and recognization, sentiment analysis, language translations, etc.

Deep Learning

- **Unsupervised Machine Learning:** the neural network learns to discover the patterns or to cluster the dataset based on unlabeled datasets. Here there are no target variables.
- The machine has to self-determined the hidden patterns or relationships within the datasets.
- Deep learning algorithms like autoencoders and generative models are used for unsupervised tasks like clustering, dimensionality reduction, and anomaly detection.

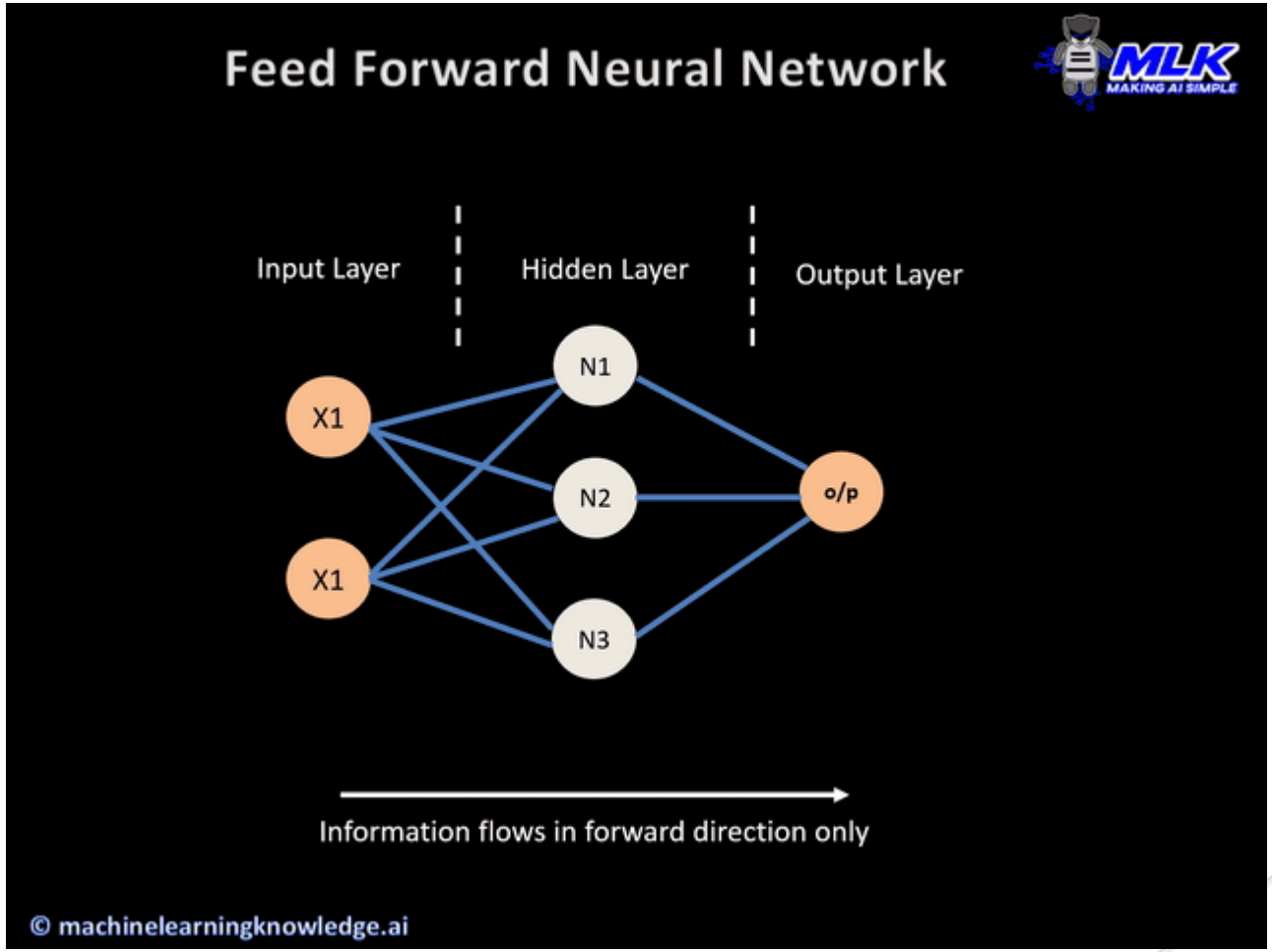
Deep Learning

- **Reinforcement Machine Learning**: an agent learns to make decisions in an environment to maximize a reward signal.
- In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.

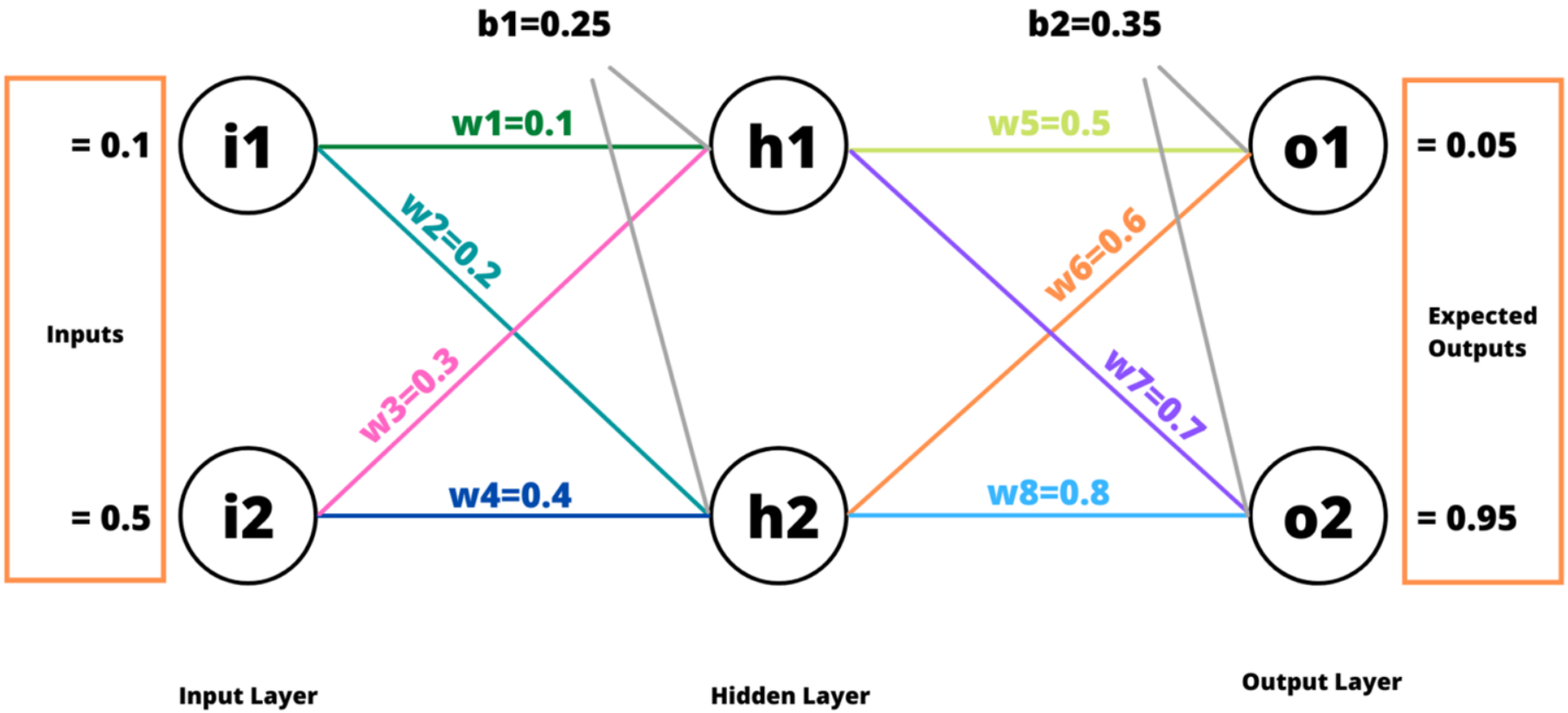
Feedforward Neural Network

- Neural networks feedforward, also known as **multi-layered networks of neurons**, are called "feedforward," where information flows in one direction from the input to the output layer without looping back.
- It is composed of three types of layers:
- **Input Layer:** The input layer accepts the input data and passes it to the next layer.
- **Hidden Layers:** One or more hidden layers that process and transform the input data..
- **Output Layer:** The output layer generates the final output. Depending on the type of problem, the number of neurons in the output layer may vary. For example, in a binary classification problem, it would only have one neuron. In contrast, a multi-class classification problem would have as many neurons as the number of classes.

Feedforward Neural Network



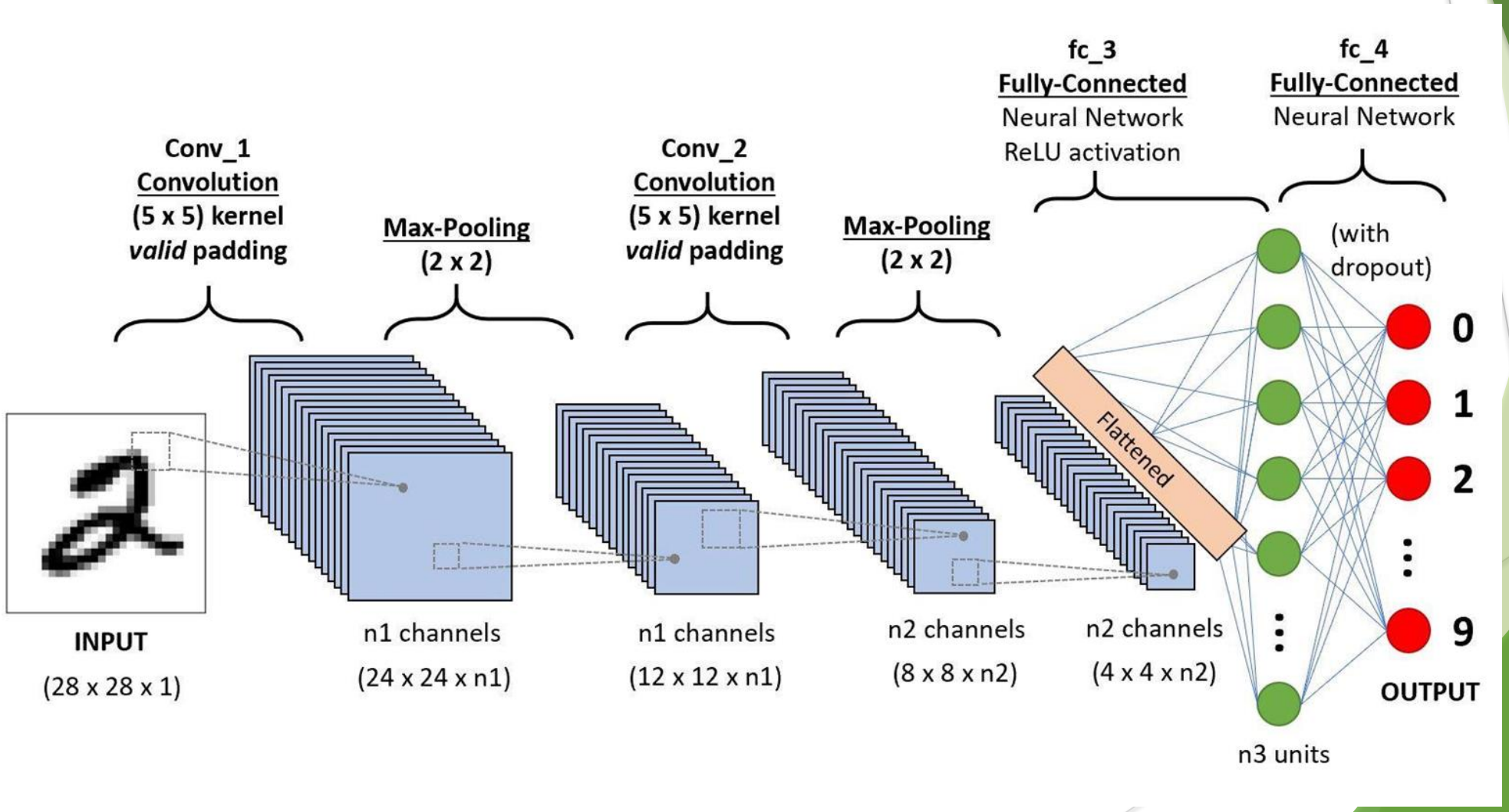
Feedforward Neural Network



Convolution Neural Network

- A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It's also known as a ConvNet.
- A convolutional neural network is used to detect and classify objects in an image.
- A convolution neural network has multiple hidden layers that help in extracting information from an image. The important layers in CNN are:
  - Convolution layer
  - Pooling layer
  - Fully connected layer

Convolution Neural Network



Convolution Neural Network

- **Sample:** A sample is a single row of data. A training dataset is comprised of many rows of data, e.g. many samples. A sample may also be called an instance, an observation, an input vector, or a feature vector.
- **Epoch:** The number of epochs is the number of complete passes through the training dataset.
- **Batch:** The batch size is a number of samples processed before the model is updated. The size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset.

Convolution Neural Network

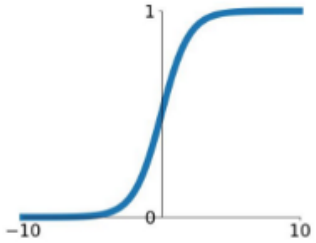
- **Learning rate:** Learning rate (λ) is one such hyper-parameter that defines the adjustment in the weights of our network with respect to the loss gradient descent. It determines how fast or slow we will move towards the optimal weights.
- **Activation Function:** The purpose of the activation function is to introduce non-linearity into the output of a neuron.

Convolution Neural Network

Activation Functions

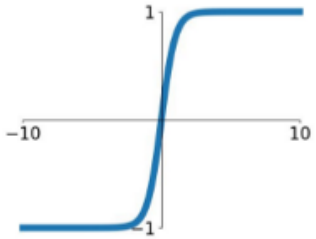
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



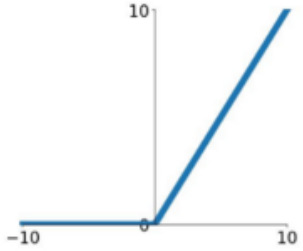
tanh

$$\tanh(x)$$



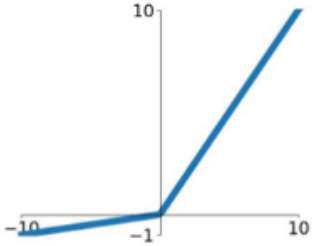
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

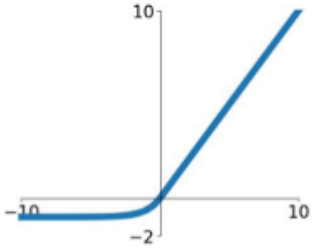


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

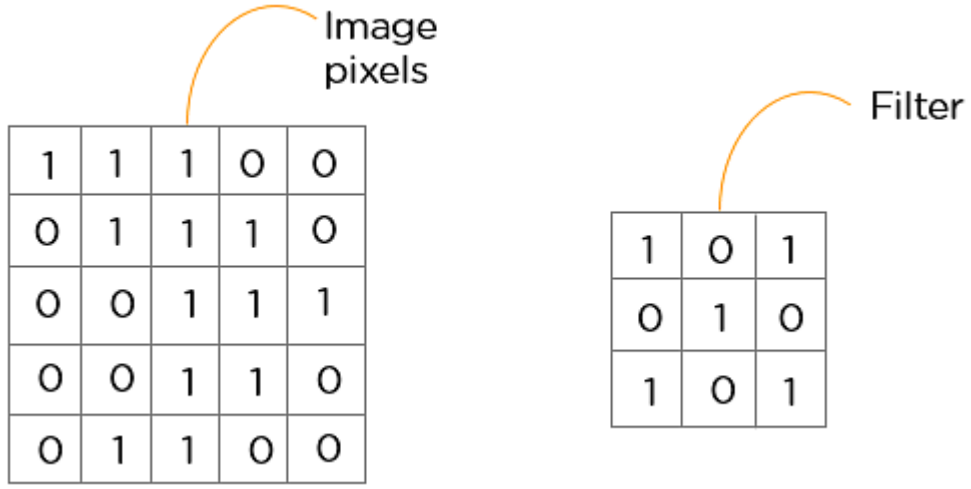
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



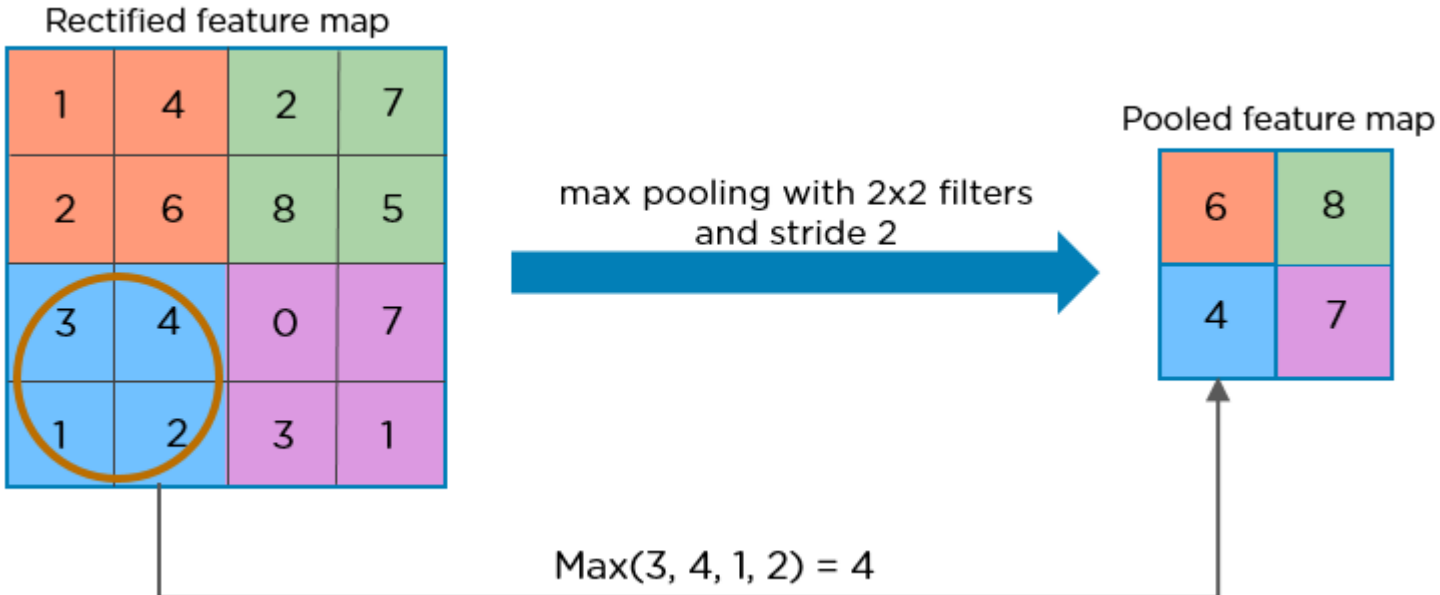
Convolution Layer

- This is the first step in the process of extracting valuable features from an image.
- A convolution layer has several filters that perform the convolution operation.
- Every image is considered as a matrix of pixel values.



Pooling Layer

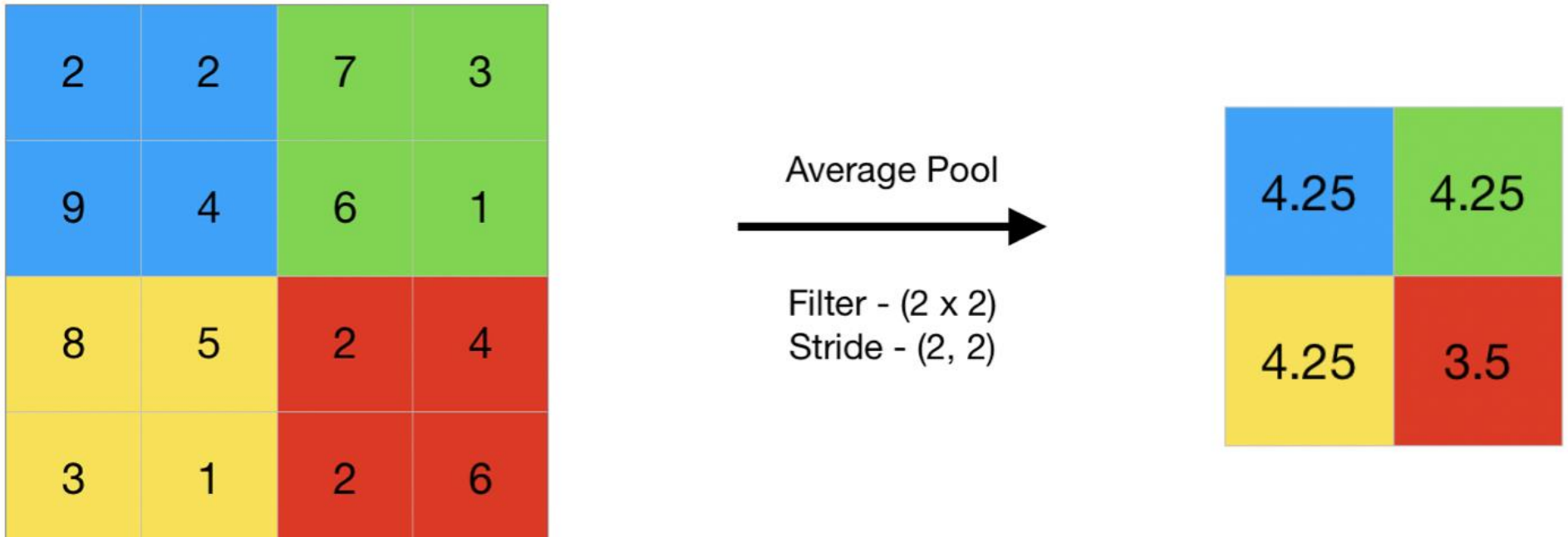
- Pooling is a down-sampling operation that reduces the dimensionality of the feature map. The rectified feature map now goes through a pooling layer to generate a pooled feature map.
- Max Pooling: Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.



Pooling Layer

- Average pooling computes the average of the elements present in the region of feature map covered by the filter.

- Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of features present in a patch.



Pooling Layer

Advantages of Pooling Layer:

1. Dimensionality reduction: The main advantage of pooling layers is that they help in reducing the spatial dimensions of the feature maps. This reduces the computational cost and also helps in avoiding overfitting by reducing the number of parameters in the model.

1. Feature selection: Pooling layers can also help in selecting the most important features from the input, as max pooling selects the most salient features and average pooling preserves more information.

Pooling Layer

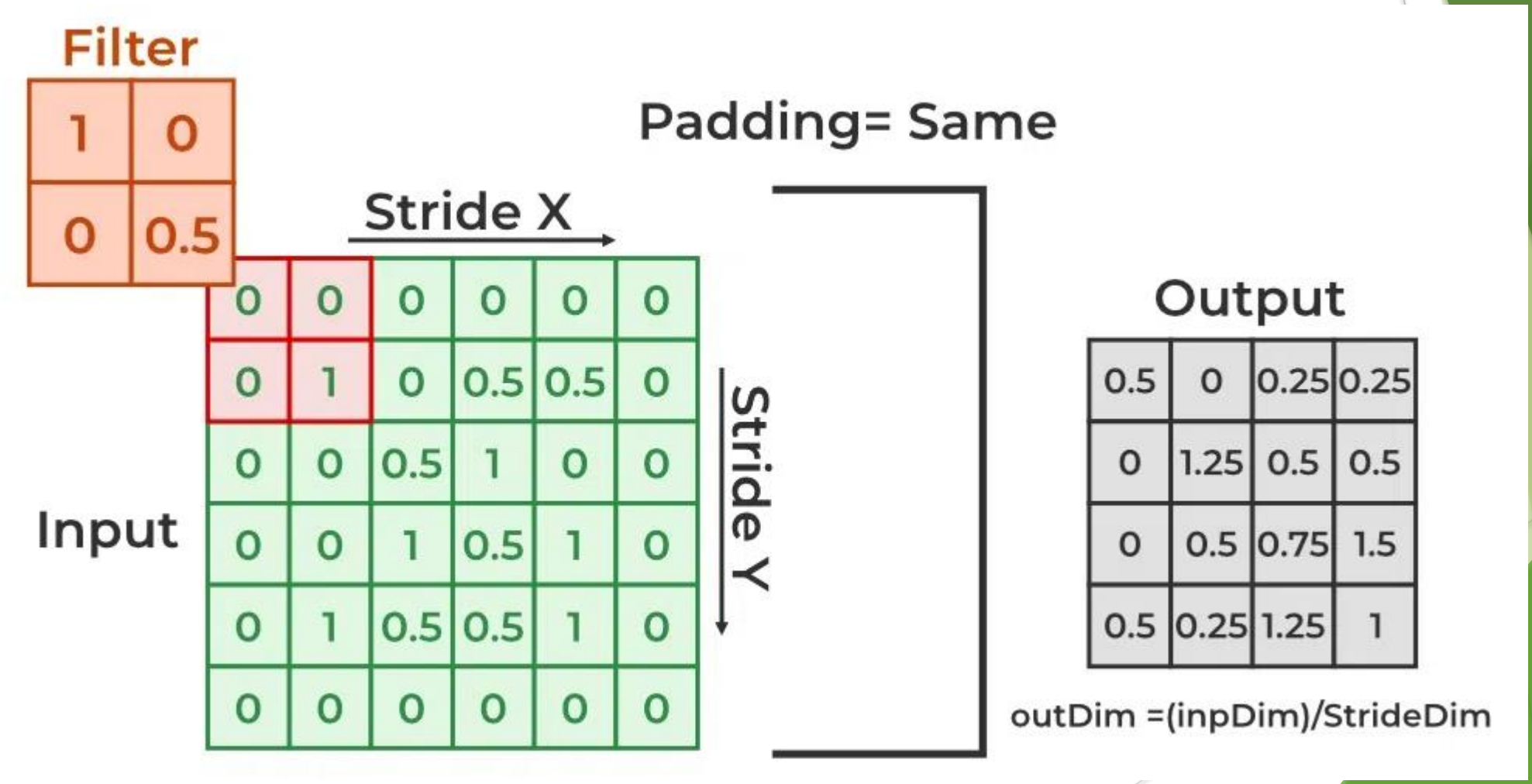
Disadvantages of Pooling Layer:

1. Information loss: One of the main disadvantages of pooling layers is that they discard some information from the input feature maps, which can be important for the final classification or regression task.
2. Over-smoothing: Pooling layers can also cause over-smoothing of the feature maps, which can result in the loss of some fine-grained details that are important for the final classification or regression task.

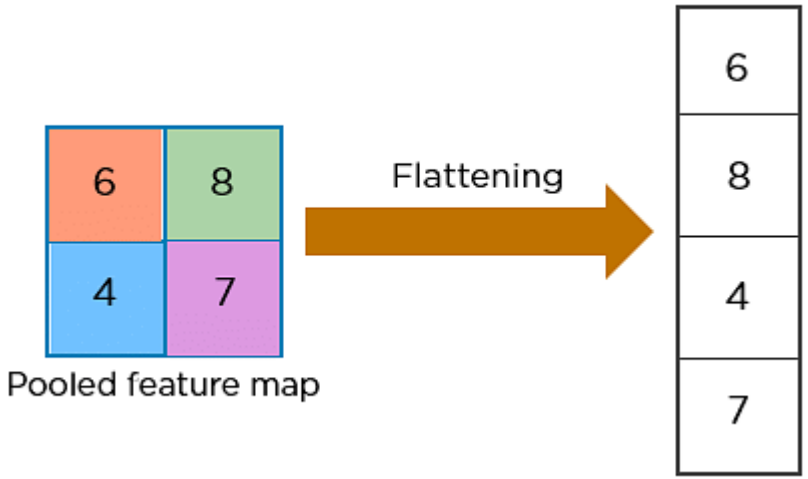
Padding

- Padding is a technique used to preserve the spatial dimensions of the input image after convolution operations on a feature map.
- **Valid Padding**: In the valid padding, no padding is added to the input feature map, and the output feature map is smaller than the input feature map. This is useful when we want to reduce the spatial dimensions of the feature maps.
- **Same Padding**: In the same padding, padding is added to the input feature map such that the size of the output feature map is the same as the input feature map. This is useful when we want to preserve the spatial dimensions of the feature maps.

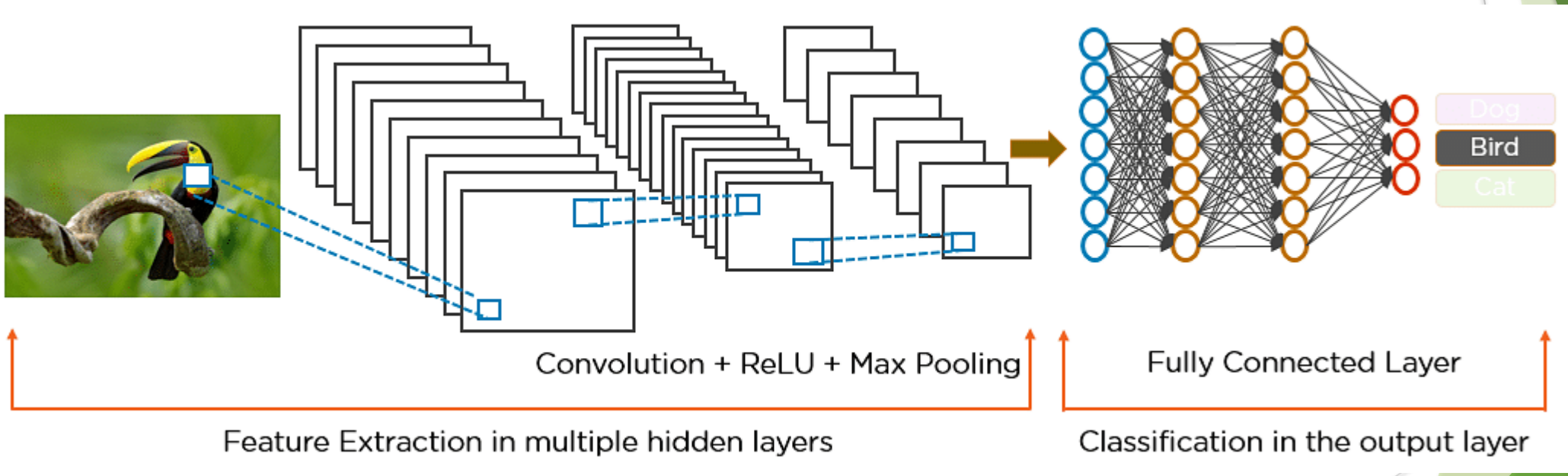
Padding


Filter
1 0
0 0.5
Padding= Same
Stride X
Stride Y
Input
0 0 0 0 0 0
0 1 0 0.5 0.5 0
0 0 0.5 1 0 0
0 0 1 0.5 1 0
0 1 0.5 0.5 1 0
0 0 0 0 0 0
Output
0.5 0 0.25 0.25
0 1.25 0.5 0.5
0 0.5 0.75 1.5
0.5 0.25 1.25 1
outDim =(inpDim)/StrideDim

- The next step in the process is called flattening. Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

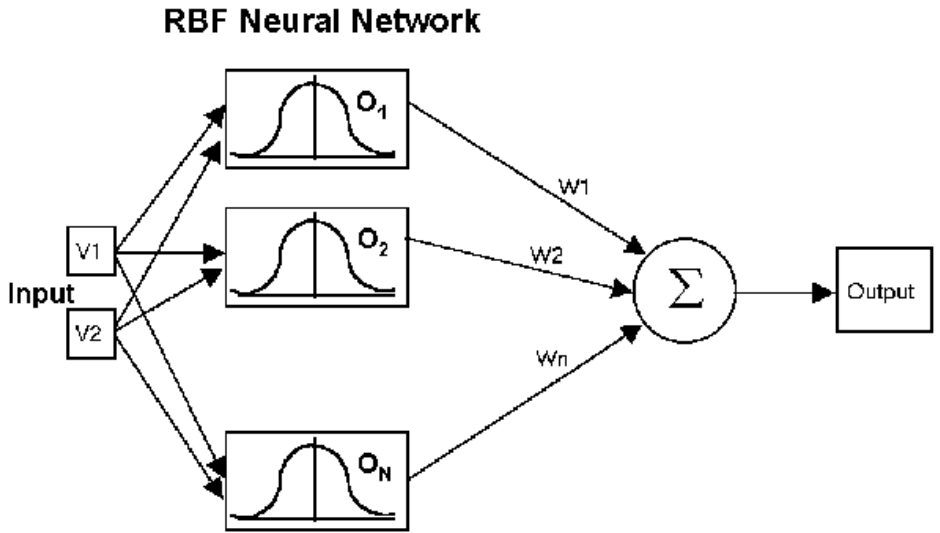


Final Step


Dog
Bird
Cat
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature Extraction in multiple hidden layers
Classification in the output layer

Radial Basis Neural Network

- Radial Basis Functions are a special class of feed-forward neural networks
- A radial basis function network is an artificial neural network that uses radial basis functions as activation functions.
- The typical architecture of a radial basis functions neural network consists of an input layer, hidden layer, and summation layer.



Radial Basis Neural Network

- **Input layer:** There is one neuron in the input layer for each predictor variable. In the case of categorical variables, N-1 neurons are used where N is the number of categories.
- **Hidden Layer:** The number of neurons in the hidden layer should be greater than the number of the input neuron. The number of neurons in the hidden layer should be less than or equal to the number of samples in the training set.
- Each neuron computes the similarity between the input vector and its prototype vector. The computation in the hidden layer can be mathematically written as follow:

$$\Phi_i = \mathrm{e}^{\left(-\frac{\|\bar{X}-\bar{u}_i\|^2}{2\cdot\sigma_i^2}\right)}$$

Radial Basis Neural Network

- **Output Layer :**The output layer uses a linear activation function for both classification or regression tasks.
- The main advantage of the RBF network is that it has only one hidden layer and it uses the radial basis function as the activation function. These functions are very powerful in approximation.

Difference between Machine Learning and Deep Learning :

Machine Learning	Deep Learning
Apply statistical algorithms to learn the hidden patterns and relationships in the dataset.	Uses artificial neural network architecture to learn the hidden patterns and relationships in the dataset.
Can work on the smaller amount of dataset	Requires the larger volume of dataset compared to machine learning
Better for the low-level task.	Better for complex task like image processing, natural language processing, etc.
Takes less time to train the model.	Takes more time to train the model.
A model is created by relevant features which are manually extracted from images to detect an object in the image.	Relevant features are automatically extracted from images. It is an end-to-end learning process.
Less complex and easy to interpret the result.	More complex, it works like the black box interpretations of the result are not easy.
It can work on the CPU or requires less computing power as compared to deep learning.	It requires a high-performance computer with GPU.

Challenges in Deep Learning

- Here are some of the main challenges in deep learning:
- Data availability: It requires large amounts of data to learn from. For using deep learning it's a big concern to gather as much data for training.
- Computational Resources: For training the deep learning model, it is computationally expensive because it requires specialized hardware like GPUs and TPUs.
- Time-consuming: While working on sequential data depending on the computational resource it can take very large even in days or months.
- Interpretability: Deep learning models are complex, it works like a black box. it is very difficult to interpret the result.
- Overfitting: when the model is trained again and again, it becomes too specialized for the training data, leading to overfitting and poor performance on new data.

Advantages of Deep Learning:

- High accuracy: Deep Learning algorithms can achieve state-of-the-art performance in various tasks, such as image recognition and natural language processing.
- Automated feature engineering: Deep Learning algorithms can automatically discover and learn relevant features from data without the need for manual feature engineering.
- Scalability: Deep Learning models can scale to handle large and complex datasets, and can learn from massive amounts of data.
- Flexibility: Deep Learning models can be applied to a wide range of tasks and can handle various types of data, such as images, text, and speech.
- Continual improvement: Deep Learning models can continually improve their performance as more data becomes available.

Dimensionality Reduction

- The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.
- It is commonly used in the fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc. It can also be used for data visualization, noise reduction, cluster analysis**, etc.

Dimensionality Reduction

- If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex.
- As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases.
- If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

Why Dimensionality Reduction?

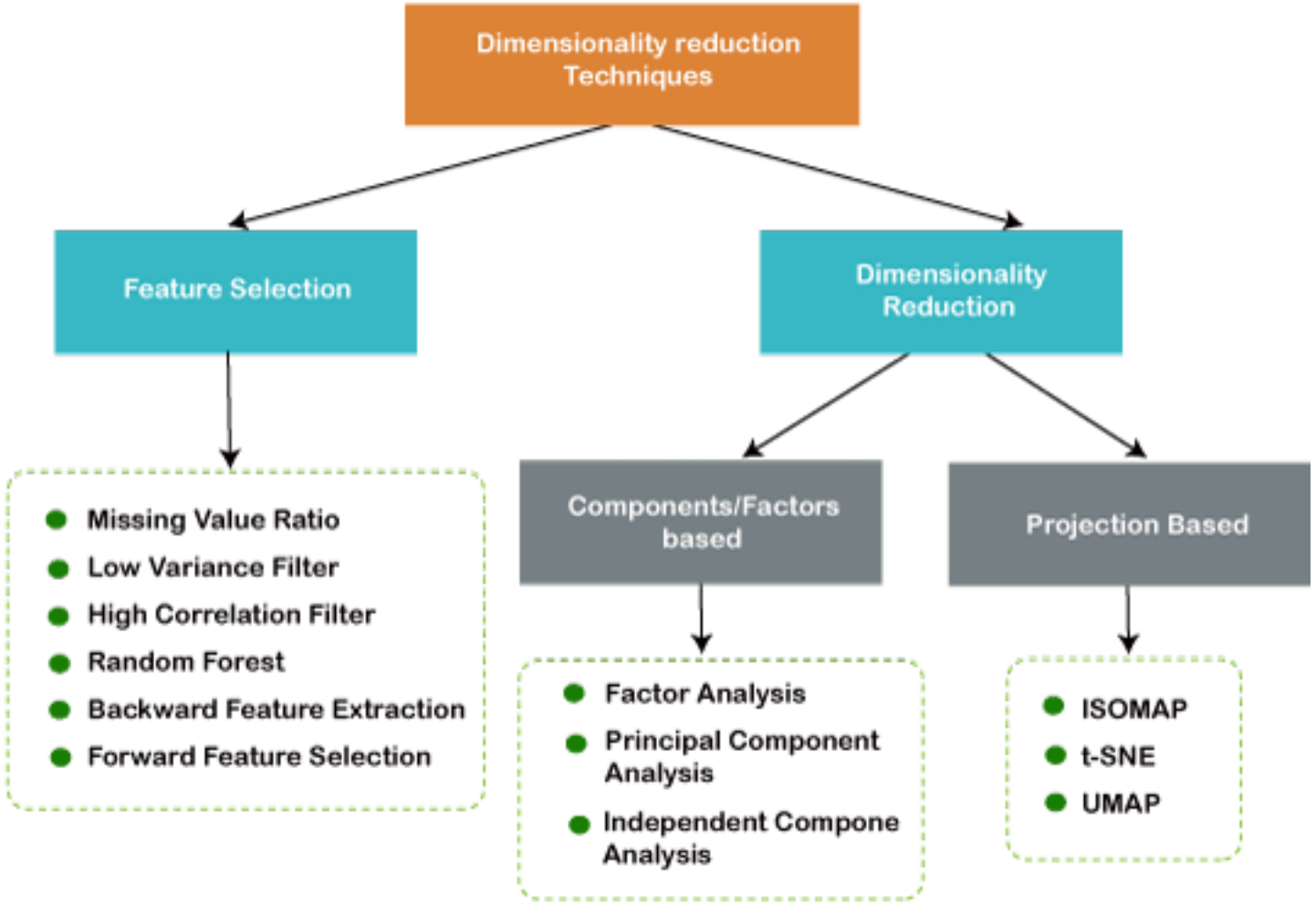
- Space required to store the data is reduced as the number of dimensions comes down
- Less dimensions lead to less computation/training time
- Some algorithms do not perform well when we have a large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful
- It takes care of multicollinearity by removing redundant features. For example, you have two variables – ‘time spent on treadmill in minutes’ and ‘calories burnt’. These variables are highly correlated as the more time you spend running on a treadmill, the more calories you will burn. Hence, there is no point in storing both as just one of them does what you require
- Visualization: projection of high-dimensional data onto 2D or 3D.

Dimensionality Reduction

Dimensionality reduction can be done in two different ways:

- By only keeping the most relevant variables from the original dataset (this technique is called feature selection)
- By finding a smaller set of new variables, each being a combination of the input variables, containing basically the same information as the input variables (this technique is called dimensionality reduction)

Dimensionality Reduction



Missing/Null Values

- Missing data and null values are not a huge problem in isolation, but a growing number of empty values may help determine whether to drop a variable, ignore missing values, or compute a predicted value.
- Most data scientists support the dropping of variables if an attribute has upward of 50% empty or null values.
- This threshold does vary.

Low Variance Filter

- Consider a variable in our dataset where all the observations have the same value, say 1. If we use this variable, do you think it can improve the model we will build?
- Data attributes that are very similar to one another do not carry much information.
- As a result, a lower variance threshold can be set and dimensions removed based on this.
- As variance is dependent on range, data normalization should take place first.

High Correlation

- High correlation between two variables means they have similar trends and are likely to carry similar information.
- This can bring down the performance of some models drastically.
- Multicollinearity is the carrying of similar information that can reduce the performance of the model.

Random Forest

- Random Forest is one of the most widely used algorithms for feature selection.
- It comes packaged with in-built feature importance so you don’t need to program that separately.
- This helps us select a smaller subset of features.

Backward Feature Elimination

Following steps are followed in ‘Backward Feature Elimination’ technique:

- We first take all the n variables present in our dataset and train the model using them
- We then calculate the performance of the model
- Now, we compute the performance of the model after eliminating each variable (n times), i.e., we drop one variable every time and train the model on the remaining n-1 variables
- We identify the variable whose removal has produced the smallest (or no) change in the performance of the model, and then drop that variable
- Repeat this process until no variable can be dropped

Forward Feature Selection

- This is the opposite process of the Backward Feature Elimination .
- Instead of eliminating features, we try to find the best features which improve the performance of the model.
- This technique works as follows:
  - We start with a single feature. Essentially, we train the model n number of times using each feature separately
  - The variable giving the best performance is selected as the starting variable
  - Then we repeat this process and add one variable at a time. The variable that produces the highest increase in performance is retained
  - We repeat this process until no significant improvement is seen in the model's performance

Principal Component Analysis (PCA)

- PCA is a technique which helps us in extracting a new set of variables from an existing large set of variables. These newly extracted variables are called Principal Components.
- A principal component is a linear combination of the original variables.
- Principal components are extracted in such a way that the first principal component explains maximum variance in the dataset.
- Second principal component tries to explain the remaining variance in the dataset and is uncorrelated to the first principal component.
- Third principal component tries to explain the variance which is not explained by the first two principal components and so on.

Step By Step Computation Of PCA

- The below steps need to be followed to perform dimensionality reduction using PCA:

1. Standardization of the data: Standardization is carried out by subtracting each value in the data from the mean and dividing it by the overall deviation in the data set.

- It can be calculated like so:

2. Computing the covariance matrix: A covariance matrix expresses the correlation between the different variables in the data set.

Consider a case where we have a 2-Dimensional data set with variables a and b, the covariance matrix is a 2×2 matrix as shown below:

$$Z = \frac{Variable\ value - mean}{Standard\ deviation}$$

Step By Step Computation Of PCA

2. Computing the covariance matrix: A covariance matrix expresses the correlation between the different variables in the data set.

- Consider a case where we have a 2-Dimensional data set with variables a and b, the covariance matrix is a 2×2 matrix as shown below:

$$\begin{bmatrix} Cov(a, a) & Cov(a, b) \\ Cov(b, a) & Cov(b, b) \end{bmatrix}$$

- The covariance value denotes how co-dependent two variables are with respect to each other
- If the covariance value is negative, it denotes the respective variables are indirectly proportional to each other
- A positive covariance denotes that the respective variables are directly proportional to each other

Step By Step Computation Of PCA

3. Calculating the eigenvectors and eigenvalues: Eigenvectors and eigenvalues computed from the covariance matrix in order to determine the principal components of the data set.

4. Computing the Principal Components: Once we have computed the Eigenvectors and eigenvalues, order them in the descending order, where the eigenvector with the highest eigenvalue is the most significant and thus forms the first principal component.

The principal components of lesser significances can thus be removed in order to reduce the dimensions of the data.

Step By Step Computation Of PCA

5. Reducing the dimensions of the data set: The last step in performing PCA is to rearrange the original data with the final principal components which represent the maximum and the most significant information of the data set.