INDUSTRIAL TRAINING REPORT

Amazon ML Summer School

Submitted in partial fulfilment of requirement of the Degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

SUBMITTED BY

Dhananjay Porwal

EN19CS301110

SUBMITTED TO

Mrs. Shredha Parmar

Department of Computer Science & Engineering

Faculty of Engineering

MEDI-CAPS UNIVERSITY, INDORE- 453331

Aug-Dec 22

Report Approval

The Industrial Training Report entitled "Amazon ML Summer School" is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted. It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the "Industrial Training Report" only for the purpose for which it has been submitted.

Internal Examiner Name:

Designation

Affiliation

External Examiner Name:

Designation

Affiliation

Declaration

I hereby declare that the Online Internship entitled "Amazon ML Summer School" submitted in partial fulfilment for the award of the degree of Bachelor of Technology in 'Computer Science & Engineering' completed under the supervision of Prof. Shredha Parmar, Department of Computer Science, Medi-Caps University and Amazon ML Summer School Training from 01-07-2022 to 31-07-2022.

Further, I declare that the content of this Industrial Training, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Dhananjay Porwal

Signature

20-11-2022

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, Shri R C Mittal, who has provided me with every facility to successfully carry out this Industrial Training, and my profound indebtedness to Prof. (Dr.) Dilip Kumar Patnaik, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank Dr. Suresh Jain, Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this Industrial Training. I would also like to thank my Head of the Department Prof. (Dr.) Pramod S. Nair for his continuous encouragement for betterment of the Industrial Training.

I express my heartfelt gratitude to my Instructor and Guide Prof. Shredha Parmar, Department of Computer Science & Amazon India Inc, without whose continuous help and support, this Industrial Training would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Name: Dhananjay Porwal

E. No.: EN19CS301110

B.Tech. IV Year

Department of

Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

Table of Contents

Introduction

Definition: Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term “Machine Learning” in 1959 while at IBM. He defined machine learning as “the field of study that gives computers the ability to learn without being explicitly programmed “. However, there is no universally accepted definition for machine learning. Different authors define the term differently. We give below two more definitions.

Machine learning is programming computers to optimize a performance criterion using example data or past experience. We have a model defined up to some parameters, and learning is the execution of a computer program to optimize the parameters of the model using the training data or past experience. The model may be predictive to make predictions in the future, or descriptive to gain knowledge from data.

The field of study known as machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

Definition of learning: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E.

Examples:

- Handwriting recognition learning problem
    - Task T: Recognizing and classifying handwritten words within images
    - Performance P: Percent of words correctly classified
    - Training experience E: A dataset of handwritten words with given classifications
- A robot driving learning problem
    - Task T: Driving on highways using vision sensors
    - Performance P: Average distance travelled before an error
    - Training experience E: A sequence of images and steering commands recorded while observing a human driver

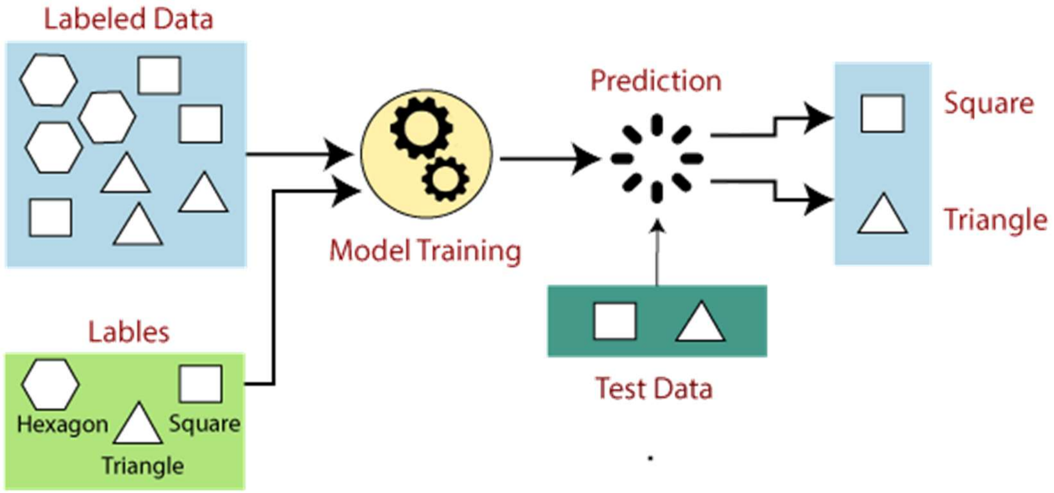
Definition: A computer program which learns from experience is called a machine learning program or simply a learning program.

Classification of Machine Learning

Machine learning implementations are classified into four major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:

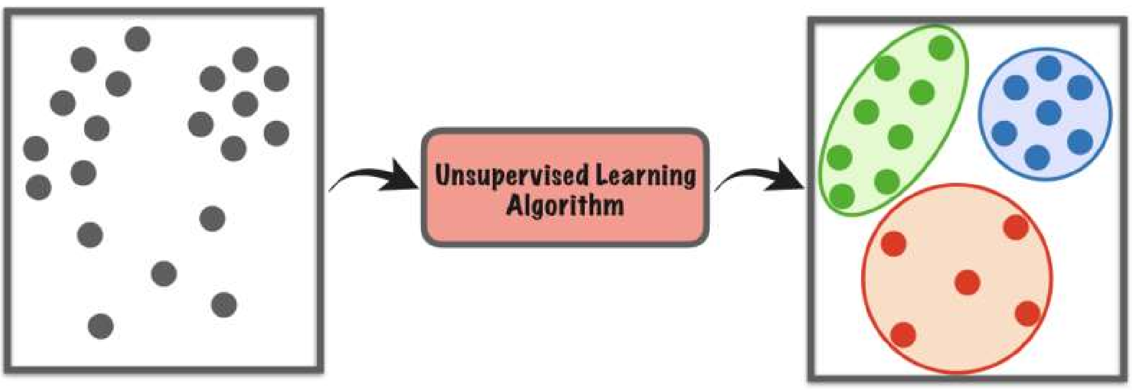
Supervised learning:

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. The given data is labelled. Both classification and regression problems are supervised learning problems.



Unsupervised learning:

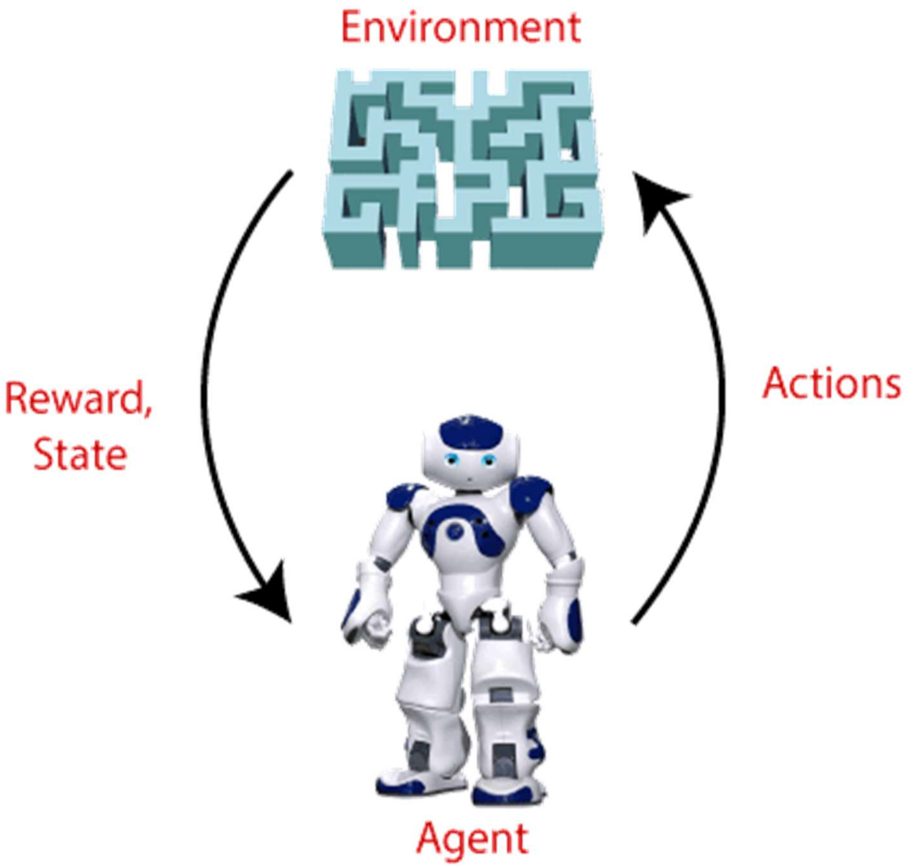
Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses. In unsupervised learning algorithms, classification or categorization is not included in the observations. Example: Consider the following data regarding patients entering a clinic. The data consists of the gender and age of the patients.



Reinforcement learning:

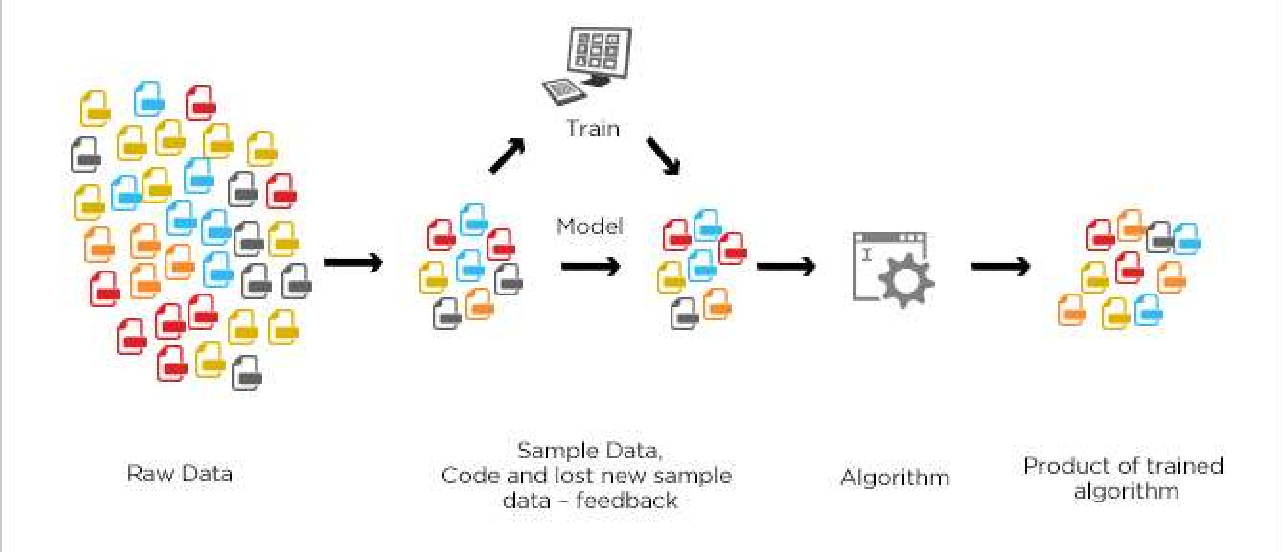
Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards.

A learner is not told what actions to take as in most forms of machine learning but instead must discover which actions yield the most reward by trying them. For example — Consider teaching a dog a new trick: we cannot tell it what tell it to do what to do, but we can reward/punish it if it does the right/wrong thing.



Semi-supervised learning:

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing. Semi-supervised learning is an approach to machine learning that combines small labelled data with a large amount of unlabelled data during training. Semi-supervised learning falls between unsupervised learning and supervised learning.



A. Supervised Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

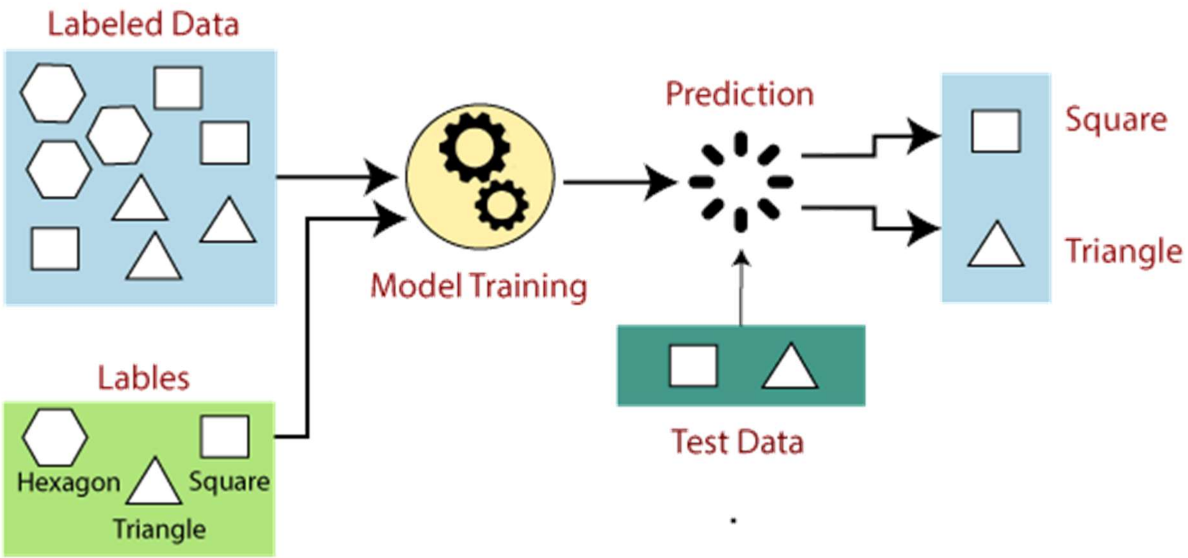
Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

- How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

The working of Supervised learning can be easily understood by the below example and diagram:



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- ❖ If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
- ❖ If the given shape has three sides, then it will be labelled as a triangle.
- ❖ If the given shape has six equal sides, then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

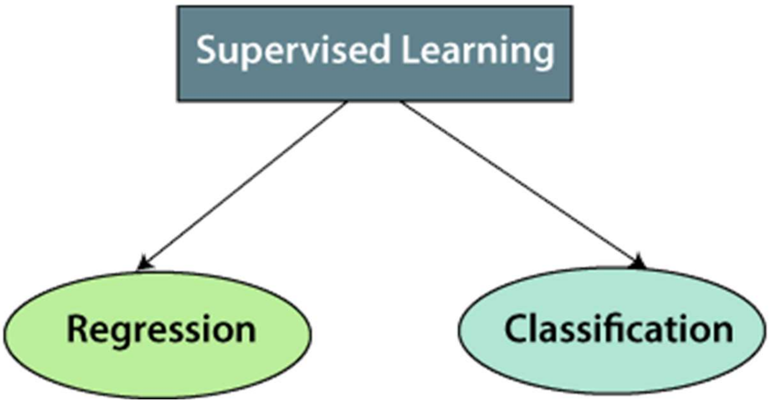
The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Steps Involved in Supervised Learning:

1. First Determine the type of training dataset
2. Collect/Gather the labelled training data.
3. Split the training dataset into training dataset, test dataset, and validation dataset.
4. Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
5. Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
6. Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

7. Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:



1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Disadvantages of supervised learning:

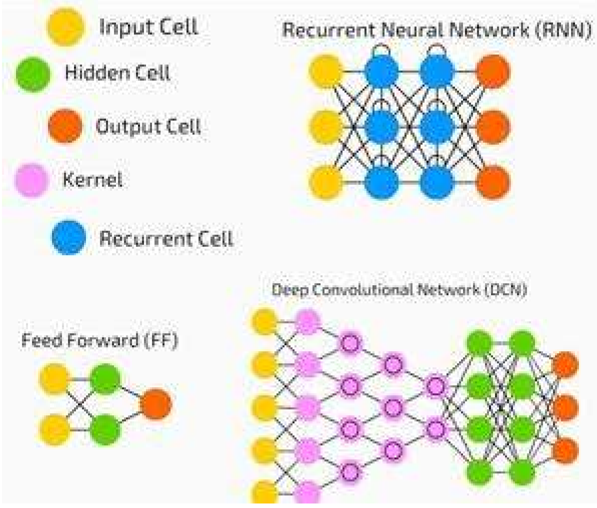
- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

B. Deep Neural Networks

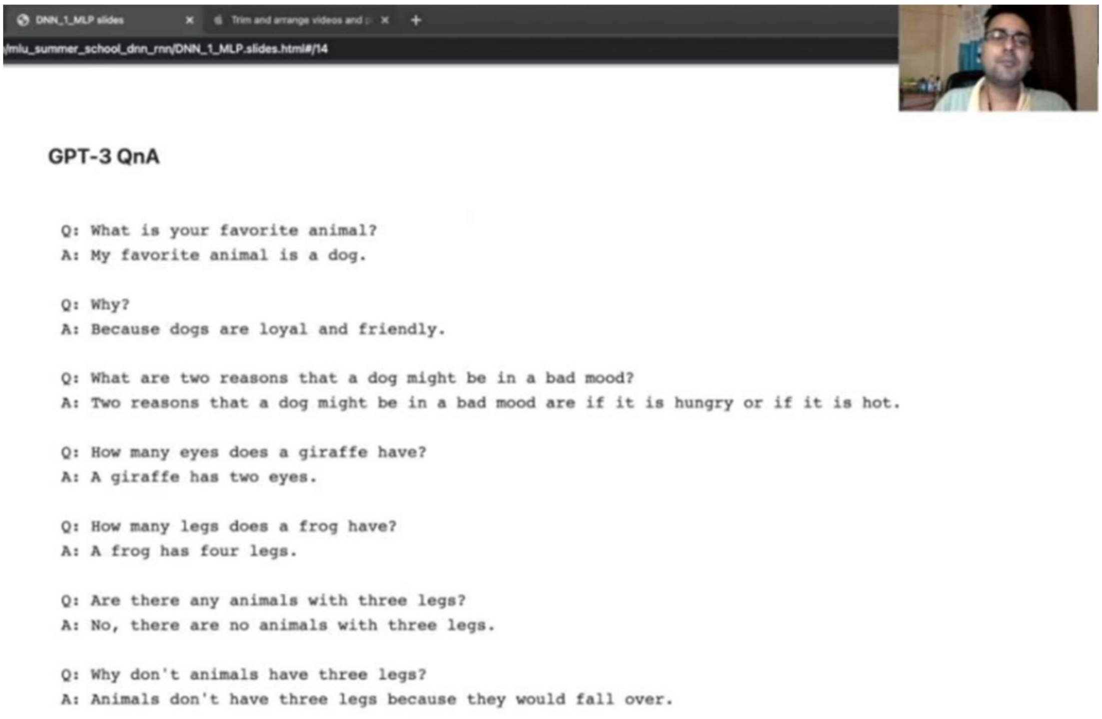
A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.

We have an input, an output, and a flow of sequential data in a deep network.



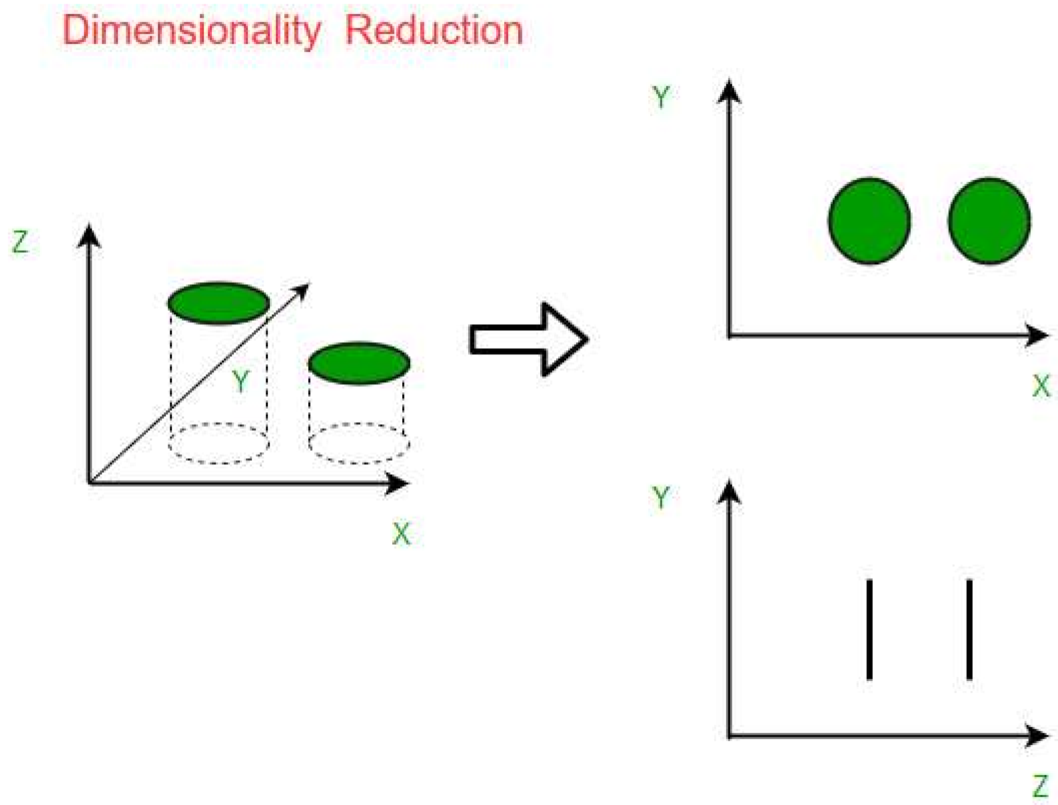
Neural networks are widely used in supervised learning and reinforcement learning problems. These networks are based on a set of layers connected to each other. In deep learning, the number of hidden layers, mostly non-linear, can be large; say about 1000 layers. DL models produce much better results than normal ML networks. We mostly use the gradient descent method for optimizing the network and minimising the loss function. We can use the Imagenet, a repository of millions of digital images to classify a dataset into categories like cats and dogs. DL nets are increasingly used for dynamic images apart from static ones and for time series and text analysis. Training the data sets forms an important part of Deep Learning models. In addition, Backpropagation is the main algorithm in training DL models. DL deals with training large neural networks with complex input output transformations.



C. Dimensionality Reduction

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

An intuitive example of dimensionality reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not. This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc. However, some of these features may overlap. In another condition, a classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both of the aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2-dimensional space, and a 1-D problem to a simple line. The below figure illustrates this concept, where a 3-D feature space is split into two 2-D feature spaces, and later, if found to be correlated, the number of features can be reduced even further.

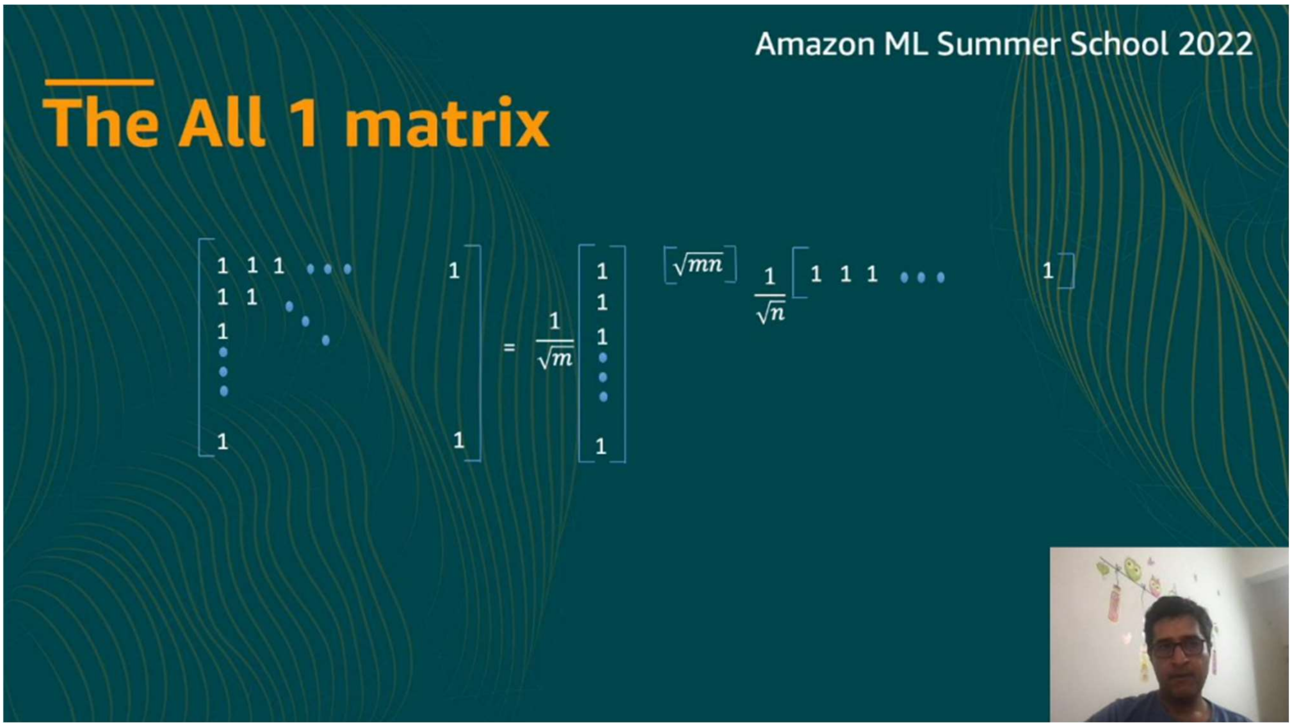


There are two components of dimensionality reduction:

- Feature selection: In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:
  - Filter
  - Wrapper
  - Embedded
- Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e., a space with lesser no. of dimensions.

The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Generalized Discriminant Analysis (GDA)



Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.

Disadvantages of Dimensionality Reduction

- It may lead to some amount of data loss.
- PCA tends to find linear correlations between variables, which is sometimes undesirable.
- PCA fails in cases where mean and covariance are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.

D. Unsupervised Learning

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

"Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision."

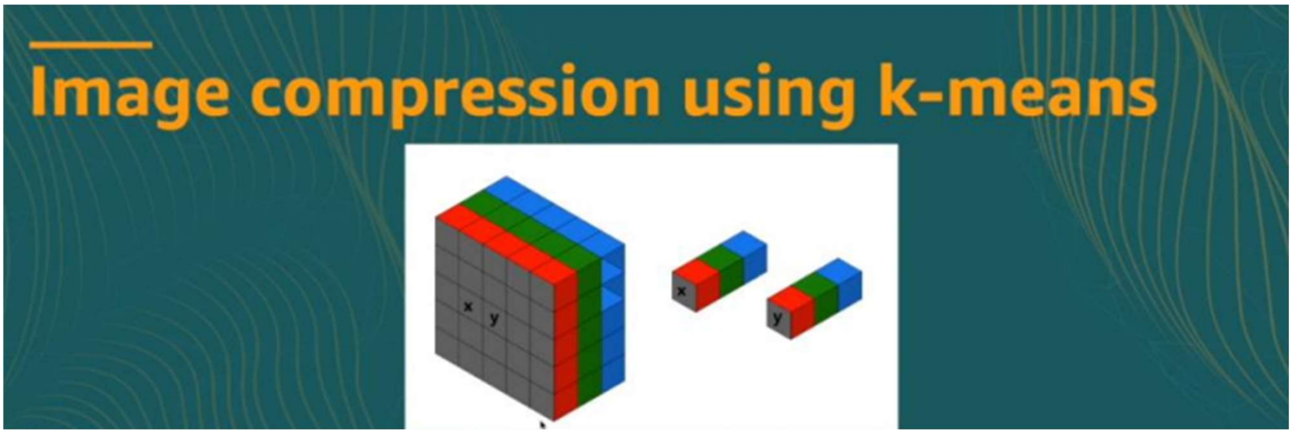
Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Why use Unsupervised Learning?

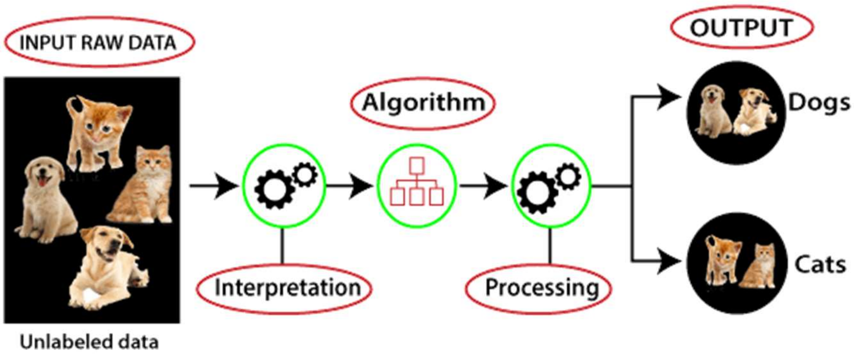
Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabelled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

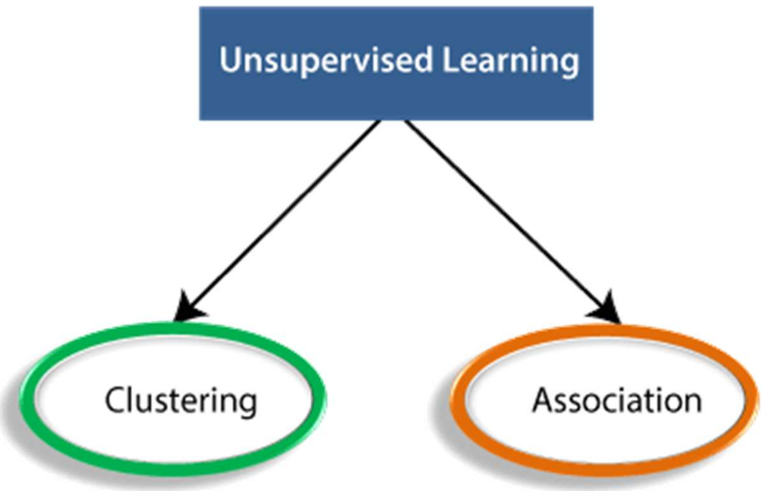


Working of Unsupervised Learning

Working of unsupervised learning can be understood by the below diagram:



Types of Unsupervised Learning Algorithm:



- **Clustering**: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
- **Association**: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Unsupervised Learning algorithms:

Below is the list of some popular unsupervised learning algorithms:

- **K-means clustering**
- **KNN (k-nearest neighbors)**
- **Hierarchal clustering**
- **Anomaly detection**
- **Neural Networks**
- **Principle Component Analysis**
- **Independent Component Analysis**
- **Apriori algorithm**
- **Singular value decomposition**

Advantages of Unsupervised Learning

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labelled input data.
- Unsupervised learning is preferable as it is easy to get unlabelled data in comparison to labelled data.

Disadvantages of Unsupervised Learning

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.

- The result of the unsupervised learning algorithm might be less accurate as input data is not labelled, and algorithms do not know the exact output in advance.

E. Probabilistic Graphical Models

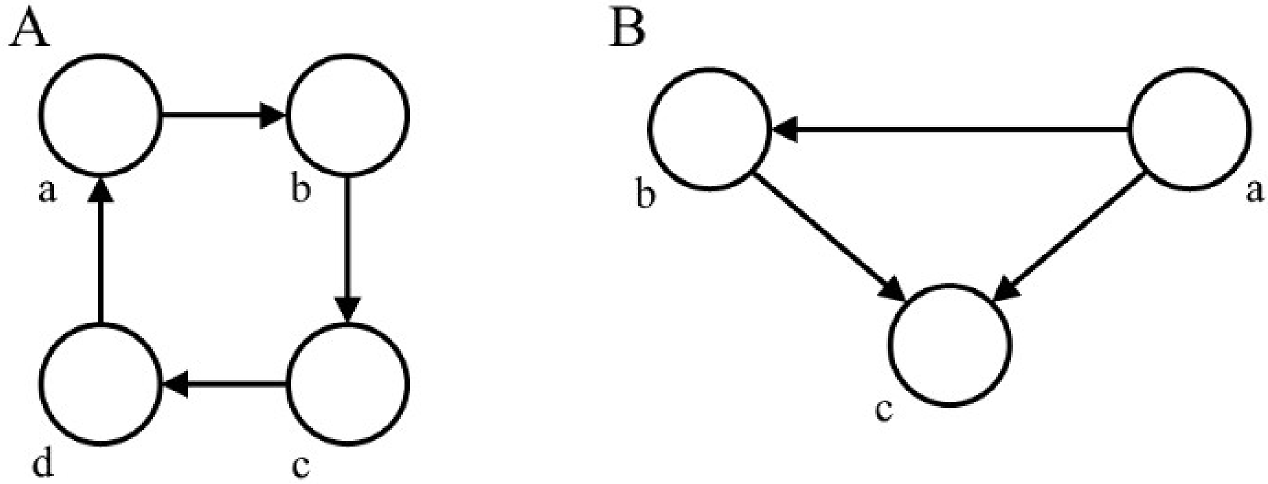
Probabilistic Graphical models (PGMs) are statistical models that encode complex joint multivariate probability distributions using graphs. In other words, PGMs capture conditional independence relationships between interacting random variables. This is beneficial since a lot of knowledge on graphs has been gathered over the years in various domains, especially on separating subsets, cliques and functions on graphs. This knowledge can be reused in PGMs. Furthermore, one can easily visualize PGMs and get a quick overview of the model structure.

By knowing the graph structure of a PGM, one can solve tasks such as inference (computing the marginal distribution of one or more random variables) or learning (estimating the parameters of probability functions). One can even try to learn the structure of the graph itself, given some data.

In this post, I will give a concise description on the two flavours of PGMs, Directed Graphical Models (DGMs), otherwise known as Bayesian Networks (BNs) and Undirected Graphical Models (UGMs) or Markov Random Fields (MRFs). I will explain the differences between these models and provide examples for both. In the last part of this post, I will look at the conversion of BNs into MRFs and back. I will also briefly touch on inference and parameter estimation in PGMs. I will provide links to more information about the topics where applicable.

Directed Graphical Models (DGMs)

As the name already suggests, directed graphical models can be represented by a graph with its vertices serving as random variables and directed edges serving as dependency relationships between them (see figure below).



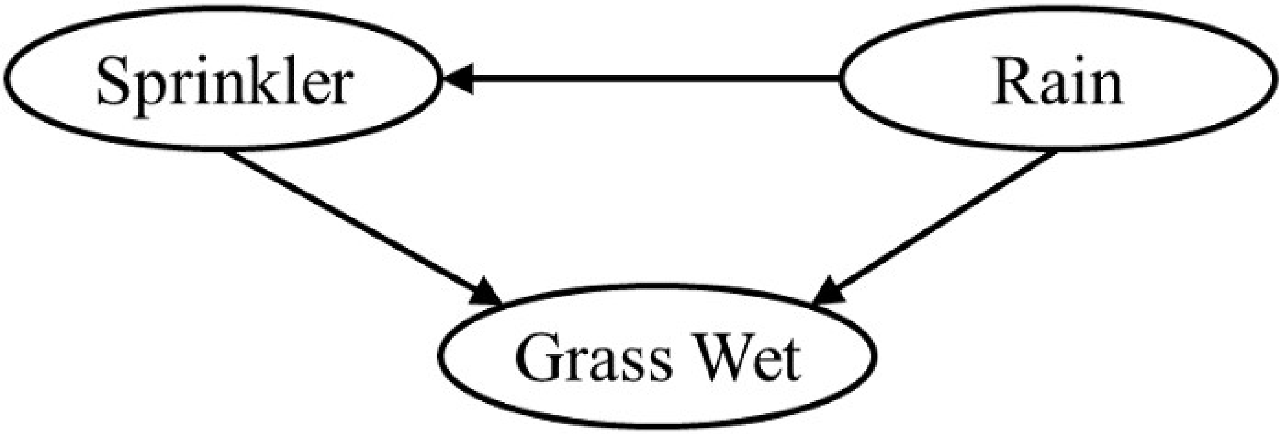
The direction of the edges determines the influence of one random variable on another. If the graph does not contain cycles (a number of vertices connected in a closed chain), it is usually referred to as a Directed Acyclic Graph (DAG). Inference on these graphs may be performed exactly using algorithms such as Belief Propagation (BP) or variable elimination.

Bayesian Networks (BNs)

An example of a DGM is the Bayesian network (BN). The Bayesian Network is a DAG with vertices (random variables) representing observable or latent variables of the model.

The directed edges ("arrows") of a BN represent conditional distributions. If the values of the vertices are binary, for example, the conditional distributions may be Bernoulli distributions. In case of continuous values, the conditional distributions may be Gaussian. The joint probability distribution is formulated as a product of conditional or marginal probabilities.

For instance, when modelling the probability of wet grass given if it is raining or if the sprinkler is on, we might represent it using a DAG such as this one:



This DAG represents the (factorized) probability distribution,

$$p(S, R, G) = p(R)p(S \mid R)p(G \mid S, R)$$

where R is the random variable for rain, S for the sprinkler and G for the wet grass. By examining the graph, you quickly see that the only independent variable in the model is R. The other two variables are conditioned on the probability of rain and/or the sprinkler. In general, the joint distribution for a BN is the product of the conditional probabilities for every node given its parents:

$$p(X) = \prod_{i=1}^{N} p(X_i \mid Parents(X_i))$$

Since most nodes have far fewer parents than the total number of nodes, the graph is usually sparse.

F. Sequential Learning

Machine learning models that input or output data sequences are known as sequence models. Text streams, audio clips, video clips, time-series data, and other types of sequential data are examples of sequential data. Recurrent Neural Networks (RNNs) are a well-known method in sequence models.

The analysis of sequential data such as text sentences, time-series, and other discrete sequence data prompted the development of Sequence Models. These models are better suited to handle sequential data, whereas Convolutional Neural Networks are better suited to treat spatial data.

The crucial element to remember about sequence models is that the data we're working with are no longer independently and identically distributed (i.i.d.) samples, and the data are reliant on one another due to their sequential order. For speech recognition, voice recognition, time series prediction, and natural language processing, sequence models are particularly popular.

Understanding Sequential Modelling

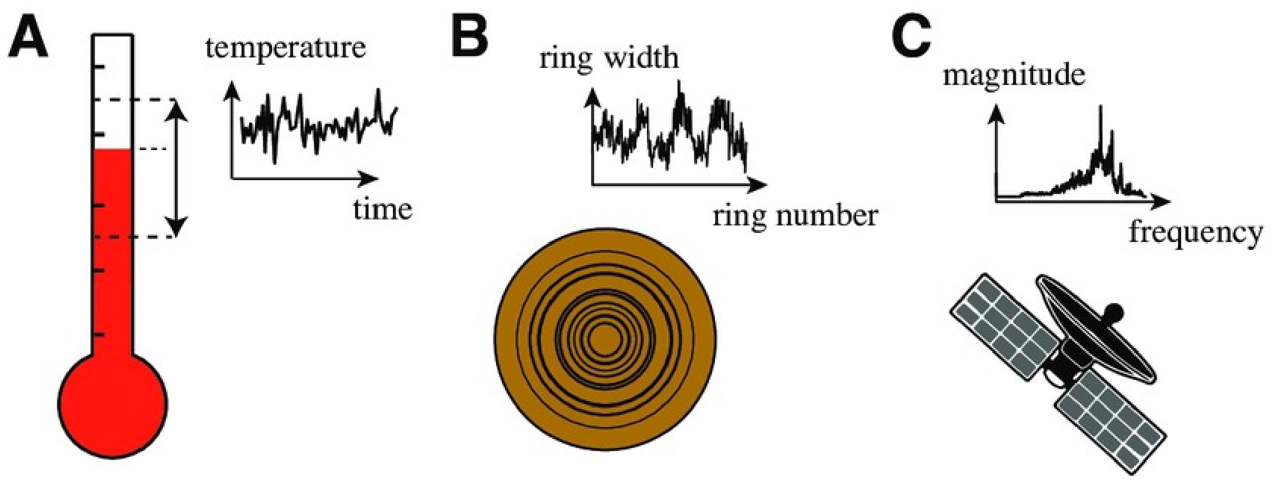
Simply described, sequence modelling is the process of producing a sequence of values from a set of input values. These input values could be time-series data, which shows how a certain variable, such as demand for a given product, changes over time. The production may be a forecast of demand for future times.

Another example is text prediction, in which the sequence modelling algorithm predicts the next word based on the sequence of the previous phrase and a set of pre-loaded conditions and rules. Businesses may achieve more than just pattern production and prediction by employing sequence modelling.

What is Sequential Data?

When the points in the dataset are dependent on the other points in the dataset, the data is termed sequential. A Timeseries is a common example of this, with each point reflecting an observation at a certain point in time, such as a stock price or sensor data. Sequences, DNA sequences, and meteorological data are examples of sequential data.

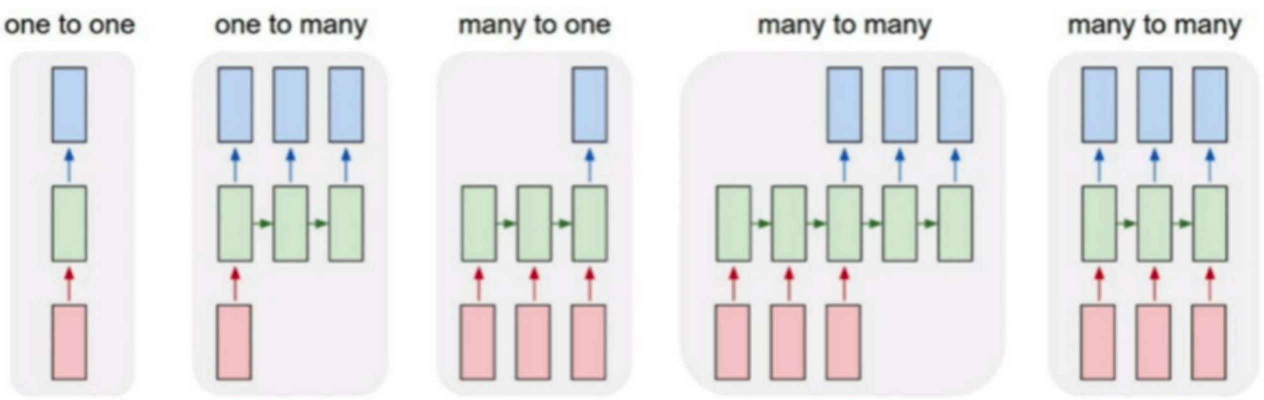
In other words, sequential we can term video data, audio data, and images up to some extent as sequential data. Below are a few basic examples of sequential data.



Below I have listed some popular machine learning applications that are based on sequential data,

- Time Series: a challenge of predicting time series, such as stock market projections.
- Text mining and sentiment analysis are two examples of natural language processing (e.g., Learning word vectors for sentiment analysis)
- Machine Translation: Given a single language input, sequence models are used to translate the input into several languages. Here's a recent poll.
- Image captioning is assessing the current action and creating a caption for the image.
- Deep Recurrent Neural Network for Speech Recognition Deep Recurrent Neural Network for Speech Recognition.
- Recurrent neural networks are being used to create classical music.
- Recurrent Neural Network for Predicting Transcription Factor Binding Sites based on DNA Sequence Analysis

In order to efficiently model with this data or to get as much information, it contains a traditional machine algorithm that will not help as much. To deal with such data there are some sequential models available and you might have heard some of those.

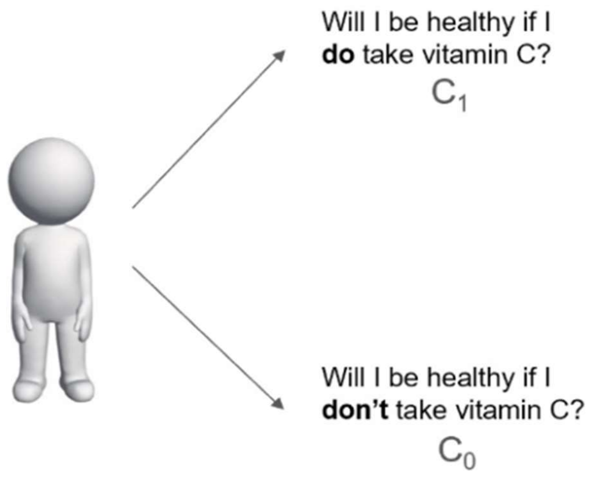


G. Casual Inference

The rules of causality play a role in almost everything we do. Criminal conviction is based on the principle of being the cause of a crime (guilt) as judged by a jury and most of us consider the effects of our actions before we make a decision. Therefore, it is reasonable to assume that considering causality in a world model will be a critical component of intelligent systems in the future. However, the formalisms, mechanisms, and techniques of causal inference remain a niche subject few explore. In this blog we formally consider the statement "association does not equal causation", review some of the basics of causal inference, discuss causal relationship discovery, and describe a few examples of the benefits of utilizing causality in AI research.

Association does not equal Causation:

To formally illustrate this concept, we will attempt to determine the causal effect of vitamin C intake on resistance to sickness. Let X be defined as a binary indicator representing if this subject intake vitamin C and let Y be a binary indicator of being healthy (not getting sick). X is also referred to as the 'treatment' in a more general setting. Now, let C1 be the value of Y if X=1 (vitamin C is taken) and C0 be the value of Y if X=0 (vitamin C is not taken). We call C0 and C1 the potential outcomes of this experiment.



X	Y	C_0	C_1
0	0	0	0*
0	0	0	0*
0	0	0	0*
0	0	0	0*
1	1	1*	1
1	1	1*	1
1	1	1*	1
1	1	1*	1

For a single person, the causal effect of taking vitamin C in this context would be the difference between the expected outcome of taking vitamin C and the expected outcome of not taking vitamin C.

$$\text{Causal Effect} = E(C_1) - E(C_0)$$

Unfortunately, we can only ever observe one of the possible outcomes C_0 or C_1. We cannot perfectly reset all conditions to see the result of the opposite treatment. Instead, we can use multiple samples and calculate the association between Vitamin C and being healthy.

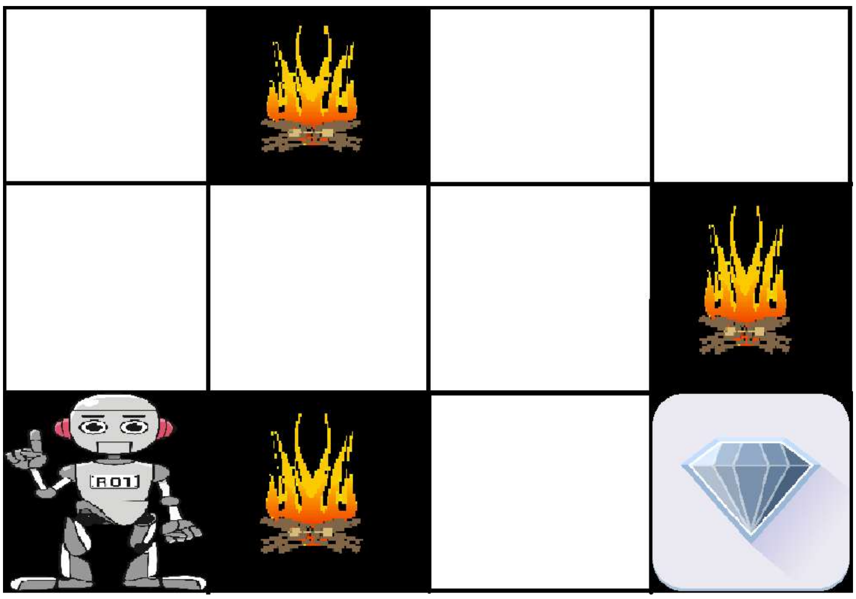
$$\text{Association} = E(Y|X=1) - E(Y|X=0)$$

From the table in Figure 3, we can calculate the association as being (1+1+1+1)/4 – (0+0+0+0)/4 = 1 and the causal effect, using the unobserved outcomes*, as being (4*0 + 4*1)/4 – (4*0 + 4*1)/4 = 0. We just calculated that, in this case, association does not equal causation. Observationally, there seems to be a perfect association between taking Vitamin C intake and being healthy. However, we can see there is no causal effect because we are privileged with the values of the unobserved outcomes. This inequality could be explained by considering that the people that stayed healthy practiced healthy habits which included taking Vitamin C. While it is easy to see the inequality in this instance, let us look at a real-world example of when this is not obvious.

H. Reinforcement Learning

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Example: The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.



The above image shows the robot, diamond, and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement learning –

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input. The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

Types of Reinforcement: There are two types of Reinforcement:

1. Positive –

Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior.

Advantages of reinforcement learning are:

- Maximizes Performance
- Sustain Change for a long period of time
- Too much Reinforcement can lead to an overload of states which can diminish the results

2. Negative –

Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided.

Advantages of reinforcement learning:

- Increases behavior
- Provide defiance to a minimum standard of performance
- It Only provides enough to meet up the minimum behavior

Various Practical applications of Reinforcement Learning –

- ✓ RL can be used in robotics for industrial automation.

- ✓ RL can be used in machine learning and data processing.
- ✓ RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

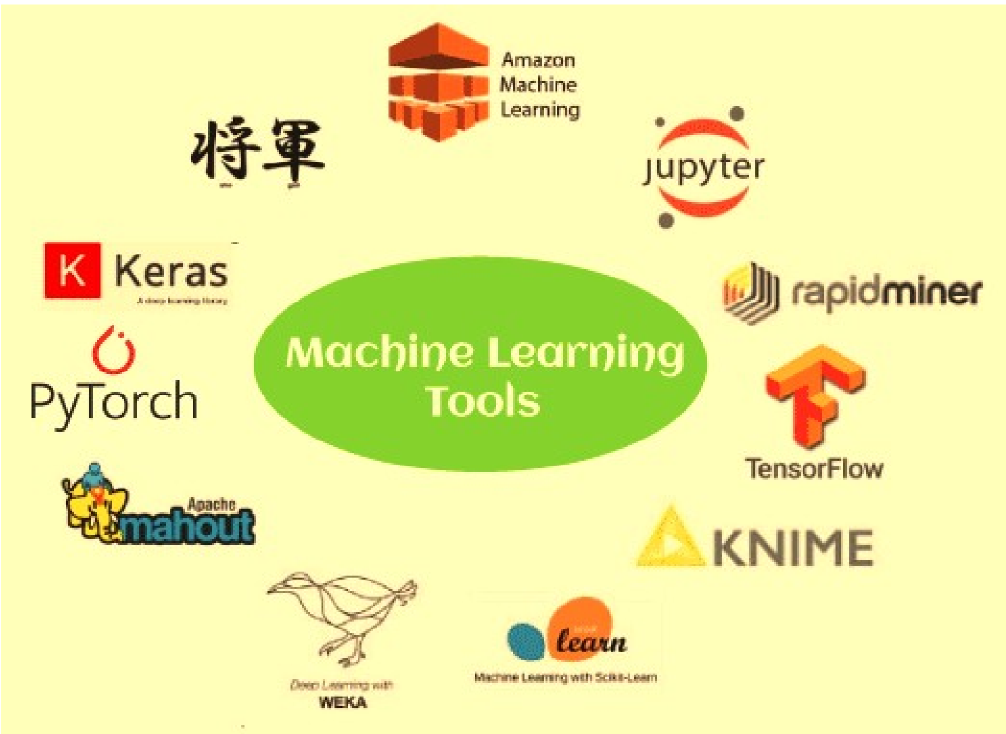
RL can be used in large environments in the following situations:

a. A model of the environment is known, but an analytic solution is not available;
b. Only a simulation model of the environment is given (the subject of simulation-based optimization)
c. The only way to collect information about the environment is to interact with it.

Tools and Technology Used

Machine learning is one of the most revolutionary technologies that is making lives simpler. It is a subfield of Artificial Intelligence, which analyses the data, build the model, and make predictions. Due to its popularity and great applications, every tech enthusiast wants to learn and build new machine learning Apps. However, to build ML models, it is important to master machine learning tools. Mastering machine learning tools will enable you to play with the data, train your models, discover new methods, and create algorithms.

There are different tools, software, and platform available for machine learning, and also new software and tools are evolving day by day. Although there are many options and availability of Machine learning tools, choosing the best tool per your model is a challenging task. If you choose the right tool for your model, you can make it faster and more efficient. In this topic, we will discuss some popular and commonly used Machine learning tools and their features.



1. TensorFlow

TensorFlow is one of the most popular open-source libraries used to train and build both machine learning and deep learning models. It provides a JS library and was developed by **Google Brain Team.** It is much popular among machine learning enthusiasts, and they use it for building different ML applications. It offers a powerful library, tools, and resources for numerical computation, specifically for large scale machine learning and deep learning projects. It enables data scientists/ML developers to build and deploy machine learning applications efficiently. For training and building the ML models, TensorFlow provides a high-level Keras API, which lets users easily start with TensorFlow and machine learning.

Features:

Below are some top features:

- TensorFlow enables us to build and train our ML models easily.
- It also enables you to run the existing models using **the TensorFlow.js**
- It provides multiple abstraction levels that allow the user to select the correct resource as per the requirement.
- It helps in building a neural network.
- Provides support of distributed computing.
- While building a model, for more need of flexibility, it provides eager execution that enables immediate iteration and intuitive debugging.
- This is open-source software and highly flexible.
- It also enables the developers to perform numerical computations using data flow graphs.
- Run-on GPUs and CPUs, and also on various mobile computing platforms.
- It provides a functionality of auto diff (Automatically computing gradients is called automatic differentiation or auto diff).
- It enables to easily deploy and training the model in the cloud.

- It can be used in two ways, i.e., by installing through NPM or by script tags.
- It is free to use.

2. PyTorch

PyTorch is an open-source machine learning framework, which is based on **the Torch** library. This framework is free and open-source and developed by **FAIR(Facebook's AI Research lab)**. It is one of the popular ML frameworks, which can be used for various applications, including computer vision and natural language processing. PyTorch has Python and C++ interfaces; however, the Python interface is more interactive. Different deep learning software is made up on top of PyTorch, such as PyTorch Lightning, Hugging Face's Transformers, Tesla autopilot, etc.

It specifies a Tensor class containing an n-dimensional array that can perform tensor computations along with GPU support.

Features:

Below are some top features:

- It enables the developers to create neural networks using Autograde Module.
- It is more suitable for deep learning researches with good speed and flexibility.
- It can also be used on cloud platforms.
- It includes tutorial courses, various tools, and libraries.
- It also provides a dynamic computational graph that makes this library more popular.
- It allows changing the network behaviour randomly without any lag.
- It is easy to use due to its hybrid front-end.
- It is freely available.

3. Google Cloud ML Engine

While training a classifier with a huge amount of data, a computer system might not perform well. However, various machine learning or deep learning projects requires millions or billions of training datasets. Or the algorithm that is being used is taking a long time for execution. In such a case, one should go for the Google Cloud ML Engine. It is a hosted platform where ML developers and data scientists build and run optimum quality machine, learning models. It provides a managed service that allows developers to easily create ML models with any type of data and of any size.

Features:

Below are the top features:

- Provides machine learning model training, building, deep learning and predictive modelling.
- The two services, namely, prediction and training, can be used independently or combinedly.
- It can be used by enterprises, i.e., for identifying clouds in a satellite image, responding faster to emails of customers.
- It can be widely used to train a complex model.

4. Amazon Machine Learning (AML)

Amazon provides a great number of machine learning tools, and one of them is **Amazon Machine Learning** or AML. Amazon Machine Learning (AML) is a cloud-based and robust machine learning software application, which is widely used for building machine learning models and making predictions. Moreover, it integrates data from multiple sources, including **Redshift, Amazon S3, or RDS.**

Features

Below are some top features:

- AML offers visualization tools and wizards.
- Enables the users to identify the patterns, build mathematical models, and make predictions.
- It provides support for three types of models, which are multi-class classification, binary classification, and regression.
- It permits users to import the model into or export the model out from Amazon Machine Learning.
- It also provides core concepts of machine learning, including ML models, Data sources, Evaluations, Real-time predictions and Batch predictions.
- It enables the user to retrieve predictions with the help of batch APIs for bulk requests or real-time APIs for individual requests.

5. NET

Accord.Net is .Net based Machine Learning framework, which is used for scientific computing. It is combined with audio and image processing libraries that are written in C#. This framework provides different libraries for various applications in ML, such as **Pattern Recognition, linear algebra, Statistical Data processing.** One popular package of the Accord.Net framework is **Accord. Statistics, Accord.Math, and Accord.MachineLearning**.

Features

Below are some top features:

- It contains 38+ kernel Functions.
- Consists of more than 40 non-parametric and parametric estimation of statistical distributions.
- Used for creating production-grade computer audition, computer vision, signal processing, and statistics apps.
- Contains more than 35 hypothesis tests that include two-way and one way ANOVA tests, non-parametric tests such as the Kolmogorov-Smirnov test and many more.

6. Apache Mahout

Apache Mahout is an open-source project of Apache Software Foundation, which is used for developing machine learning applications mainly focused on Linear Algebra. It is a distributed linear algebra framework and mathematically expressive Scala DSL, which enable the developers to promptly implement their own algorithms. It also provides Java/Scala libraries to perform Mathematical operations mainly based on linear algebra and statistics.

Features:

Below are some top features:

- It enables developers to implement machine learning techniques, including recommendation, clustering, and classification.
- It is an efficient framework for implementing scalable algorithms.
- It consists of matrix and vector libraries.
- It provides support for multiple distributed backends(including Apache Spark)
- It runs on top of Apache Hadoop using the MapReduce paradigm.

7. Shogun

Shogun is a free and open-source machine learning software library, which was created by **Gunnar Raetsch and Soeren Sonnenburg** in the year **1999**. This software library is written in C++ and supports interfaces for different languages such as Python, R, Scala, C#, Ruby, etc., using **SWIG**(Simplified Wrapper and Interface Generator). The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems. It also provides the complete implementation of Hidden Markov Models.

Features:

Below are some top features:

- The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems.
- It provides support for the use of pre-calculated kernels.
- It also offers to use a combined kernel using Multiple kernel Learning Functionality.
- This was initially designed for processing a huge dataset that consists of up to 10 million samples.
- It also enables users to work on interfaces on different programming languages such as Lua, Python, Java, C#, Octave, Ruby, MATLAB, and R.

8. Oryx2

It is a realization of the lambda architecture and built on **Apache Kafka** and **Apache Spark**. It is widely used for real-time large-scale machine learning projects. It is a framework for building apps, including end-to-end applications for filtering, packaged, regression, classification, and clustering. It is written in Java languages, including Apache Spark, Hadoop, Tomcat, Kafka, etc. The latest version of Oryx2 is Oryx 2.8.0.

Features:

Below are some top features:

- It has three tiers: specialization on top providing ML abstractions, generic lambda architecture tier, end-to-end implementation of the same standard ML algorithms.
- The original project of Oryx2 was Oryx1, and after some upgrades, Oryx2 was launched.
- It is well suited for large-scale real-time machine learning projects.
- It contains three layers which are arranged side-by-side, and these are named as Speed layer, batch layer, and serving layer.
- It also has a data transport layer that transfer data between different layers and receives input from external sources.

9. Apache Spark MLlib

Apache Spark MLlib is a scalable machine learning library that runs on Apache Mesos, Hadoop, Kubernetes, standalone, or in the cloud. Moreover, it can access data from different data sources. It is an open-source cluster-computing framework that offers an interface for complete clusters along with data parallelism and fault tolerance.

For optimized numerical processing of data, MLlib provides linear algebra packages such as Breeze and netlib-Java. It uses a query optimizer and physical execution engine for achieving high performance with both batch and streaming data.

Features

Below are some top features:

- MLlib contains various algorithms, including Classification, Regression, Clustering, recommendations, association rules, etc.
- It runs different platforms such as Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud against diverse data sources.
- It contains high-quality algorithms that provide great results and performance.
- It is easy to use as it provides interfaces In Java, Python, Scala, R, and SQL.

10. Google ML kit for Mobile

For Mobile app developers, Google brings ML Kit, which is packaged with the expertise of machine learning and technology to create more robust, optimized, and personalized apps. This tools kit can be used for face detection, text recognition, landmark detection, image labelling, and barcode scanning applications. One can also use it for working offline.

Features:

Below are some top features:

- The ML kit is optimized for mobile.
- It includes the advantages of different machine learning technologies.
- It provides easy-to-use APIs that enables powerful use cases in your mobile apps.
- It includes Vision API and Natural Language APIS to detect faces, text, and objects, and identify different languages & provide reply suggestions.

Conclusion

Machine Learning can be a Supervised or Unsupervised. If you have lesser amount of data and clearly labelled data for training, opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If you have a huge data set easily available, go for deep learning techniques. You also have learned Reinforcement Learning and Deep Reinforcement Learning. You now know what Neural Networks are, their applications and limitations.

Finally, when it comes to the development of machine learning models of your own, you looked at the choices of various development languages, IDEs and Platforms. Next thing that you need to do is start learning and practicing each machine learning technique. The subject is vast, it means that there is width, but if you consider the depth, each topic can be learned in a few hours. Each topic is independent of each other. You need to take into consideration one topic at a time, learn it, practice it and implement the algorithm/s in it using a language choice of yours. This is the best way to start studying Machine Learning. Practicing one topic at a time, very soon you would acquire the width that is eventually required of a Machine Learning expert.

We have discussed some popular machine learning tools and technology. However, there are many more other ML tools and technologies, but choosing the tool and technology can be completely depends on the requirement for one's project, skills, and price to the tool. Most of these tools are freely available, except for some tools such as Rapid Miner. Each tool works in a different language and provides some specifications.

Reference

- Machine Learning Tools - Javatpoint
- Deep Neural Networks (tutorialspoint.com)
- Introduction to Dimensionality Reduction - GeeksforGeeks
- Unsupervised Machine learning - Javatpoint
- Introduction to Probabilistic Graphical Models | by Branislav Holländer | Towards Data Science
- A Tutorial on Sequential Machine Learning (analyticsindiamag.com)
- Sequential Learning | SpringerLink
- Introduction to Semi-Supervised Learning - Javatpoint
- What is Unsupervised Learning? | IBM