ML-UNIT 1

Comparison of Machine Learning with Traditional Programming:

Aspect	Machine Learning	Traditional Programming
Approach	Learns from data to create algorithms that make predictions.	Developers write rule-based code to solve specific problems.
Learning	Self-learning from historical data.	No self-learning; follows predefined rules.
Decision-Making	Data-driven; adapts based on new data.	Rule-based; decisions are fixed and based on developer logic.
Flexibility	Can handle complex tasks by finding patterns in large datasets.	Limited to the developer's instructions; less flexible.
Examples	Used in chatbots, self-driving cars, recommendation systems, etc.	Used to build applications like calculators, inventory systems, etc.
Capability	Can discover insights that may be difficult for humans.	Limited to the intelligence and creativity of the developer.
Adaptability	Continuously improves as more data is available.	Does not improve unless the developer updates the code.
Field	A subset of Artificial Intelligence (AI).	A fundamental part of software development, but not AI.

- Fig 1.2.1 shows machine learning and traditional programming.

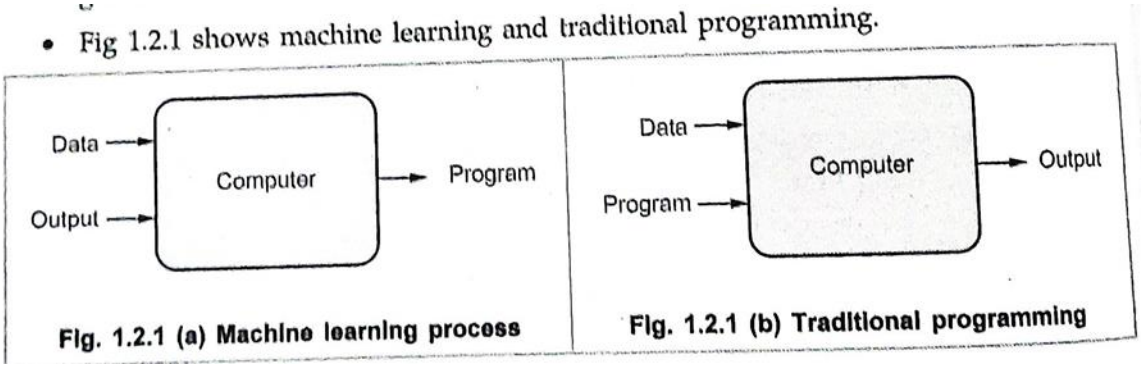


Fig. 1.2.1 (a) Machine learning process

Fig. 1.2.1 (b) Traditional programming

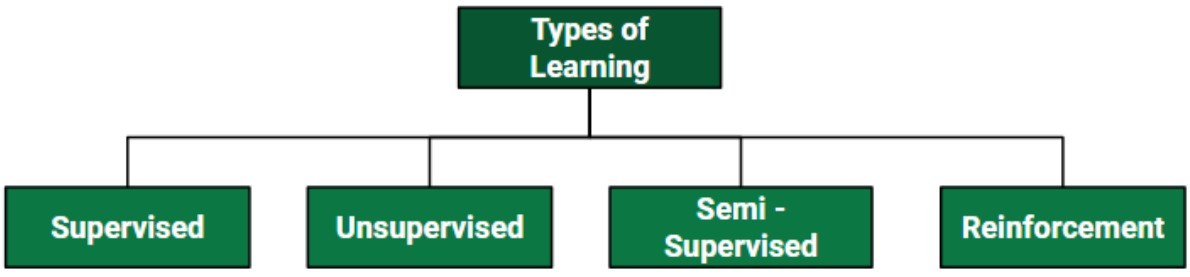
ML vs Traditional Programming vs AI

Aspect	Machine Learning	Traditional Programming	Artificial Intelligence
Definition	A part of AI that learns from data to make predictions.	Developers write code based on specific rules and instructions.	A broad field that makes machines capable of tasks needing human intelligence.
Learning Method	Learns from data to improve over time.	Follows fixed rules set by the developer.	Can include both learning from data (like ML) and using fixed rules.
Decision-Making	Uses data to make predictions and decisions.	Decisions are based on pre-written rules.	Combines data and rules for complex decision-making.
Flexibility	Adapts and improves with more data.	Limited to the instructions given by the developer.	Can handle a wide range of tasks by using different techniques.
Capability	Finds patterns in large datasets that humans might miss.	Limited to what the developer programs.	Solves complex tasks with accuracy that may be impossible for humans alone.
Use Cases	Used in chatbots, self-driving cars, and recommendation systems.	Used to build software with specific functions, like calculators or websites.	Applied in areas like language processing, computer vision, and robotics.
Role in AI	A subset of AI, focused on learning and prediction.	A fundamental part of building applications but not AI.	AI includes ML and other methods to make machines intelligent.

Machine Learning	Artificial Intelligence	Data Science
Focuses on providing a means for algorithms and systems to learn from experience with data and use that experience to improve over time.	Focuses on giving machines cognitive and intellectual capabilities similar to those of humans.	Focuses on extracting information needles from data haystacks to aid in decision-making and planning.
Machine Learning uses statistical models.	Artificial Intelligence uses logic and decision trees.	Data Science deals with structured data.
A form of analytics in which software programs learn about data and find patterns.	Development of computerized applications that simulate human intelligence and interaction.	The process of using advanced analytics to extract relevant information from data.
Objective is to maximize accuracy.	Objective is to maximize the chance of success.	Objective is to extract actionable insights from the data.
ML can be done through supervised, unsupervised or reinforcement learning approaches.	AI encompasses a collection of intelligence concepts, including elements of perception, planning and prediction.	Uses statistics, mathematics, data wrangling, big data analytics, machine learning and various other methods to answer analytics questions.
ML is concerned with knowledge accumulation.	AI is concerned with knowledge dissemination and conscious machine actions.	Data science is all about data engineering.

Q1) a) Compare machine learning vs Artificial Intelligence. **[5]**

	ML	AI
Definition	A part of AI that learns from data to make predictions.	A broad field that aims to make machines do tasks that need human intelligence.
Scope	Focuses on creating algorithms that learn from data.	Includes ML and other techniques like rule-based systems and robotics.
Learning	Learns and improves over time using data.	Can learn from data (like ML) or use fixed rules.
Techniques Used	Uses methods like supervised, unsupervised, and reinforcement learning.	Uses many methods, including ML, deep learning, and rule-based systems.
Decision-Making	Makes decisions based on patterns found in data.	Combines different methods to make smart decisions.
Examples	Used in tasks like recommendations, fraud detection, and speech recognition.	Used in tasks like self-driving cars, smart assistants, and game playing.
Flexibility	Adapts and gets better with more data.	Can perform a wide range of tasks using different techniques.
Goal	To create systems that learn and make predictions from data.	To create intelligent systems that can do tasks that usually require human thinking.
Role in AI	ML is a key part of AI.	AI includes ML as one of its ways to make machines intelligent.



Q2) a) Explain supervised, unsupervised and semi supervised learning. [7]

Supervised Machine Learning

← Prev Next →

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.

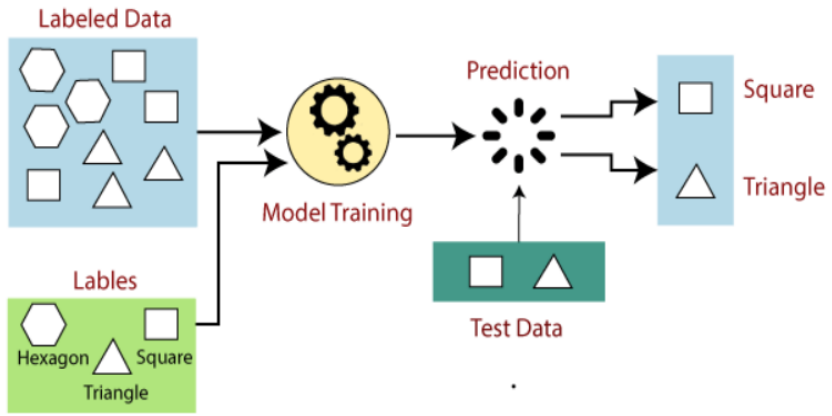
Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

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:

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

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

What is 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:

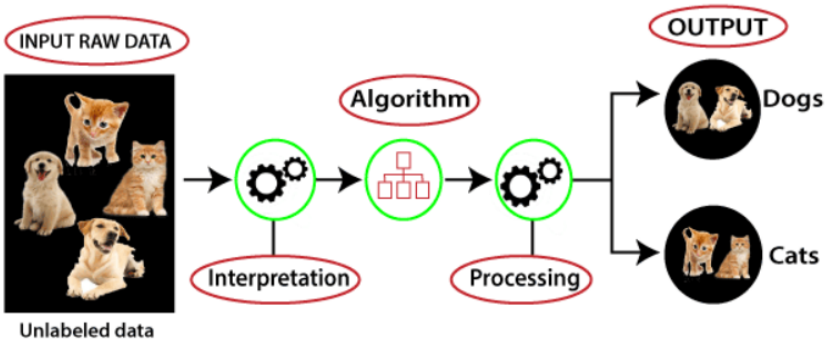
> *Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.*

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.

Working of Unsupervised Learning

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

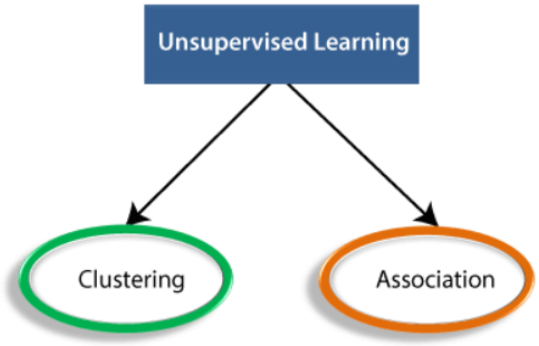


Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



- **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.

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 labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled 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 labeled, and algorithms do not know the exact output in advance.

1. *Explain supervised learning with example.* **SPPU : March-20, In Sem, Marks 5**
2. *Explain data formats for supervised learning problem with example.* **SPPU: June-22, End Sem, Marks 6**

1) Supervised Learning with example

Supervised learning is a type of machine learning where a model is trained on labeled data. This means that the data provided to the model includes both the input features and the correct output (label). The goal is for the model to learn the relationship between the input and the output so that it can accurately predict the output for new, unseen data.

Example: Predicting House Prices

Let's say you want to predict the price of a house based on its size, number of bedrooms, and location. First, you collect data on houses that have already been sold. This data includes the size of the house, the number of bedrooms, the location, and the actual price for each house.

- **Training Data:**
  The data you collected is your labeled training data. Each house in the dataset has both input features (size, bedrooms, location) and a label (price).
- **Training the Model:**
  You feed this labeled data into a supervised learning algorithm, like Linear Regression. The algorithm analyzes the data and learns the relationship between the features and the price.
- **Making Predictions:**
  Once the model is trained, you can use it to predict the price of a new house by inputting its size, number of bedrooms, and location. The model will output a price based on what it learned from the training data.

Types of Supervised Learning:

- **Classification:** If the output is a category, like "spam" or "not spam," this is a classification problem. For example, an email spam filter uses supervised learning to classify emails as spam or not spam.
- **Regression:** If the output is a continuous value, like predicting house prices, this is a regression problem.

Conclusion:
In supervised learning, the model learns from labeled examples to make accurate predictions. It's widely used in applications like spam detection, medical diagnosis, and price prediction.

1.5.2 Difference between Supervised and Unsupervised Learning

Sr. No.	Supervised learning	Unsupervised learning
1.	Desired output is given.	Desired output is not given.
2.	It is not possible to learn larger and more complex models than with supervised learning.	It is possible to learn larger and more complex models with unsupervised learning.
3.	Use training data to infer model.	No training data is used.
4.	Every input pattern that is used to train the network is associated with an output pattern.	The target output is not presented to the network.
5.	Trying to predict a function from labeled data.	Try to detect interesting relations in data.
6.	Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given.	For unsupervised learning typically either the target variable is unknown or has only been recorded for too small a number of cases.
7.	Example : Optical character recognition.	Example : Find a face in an image.
8.	We can test our model.	We can not test our model.
9.	Supervised learning is also called classification.	Unsupervised learning is also called clustering.

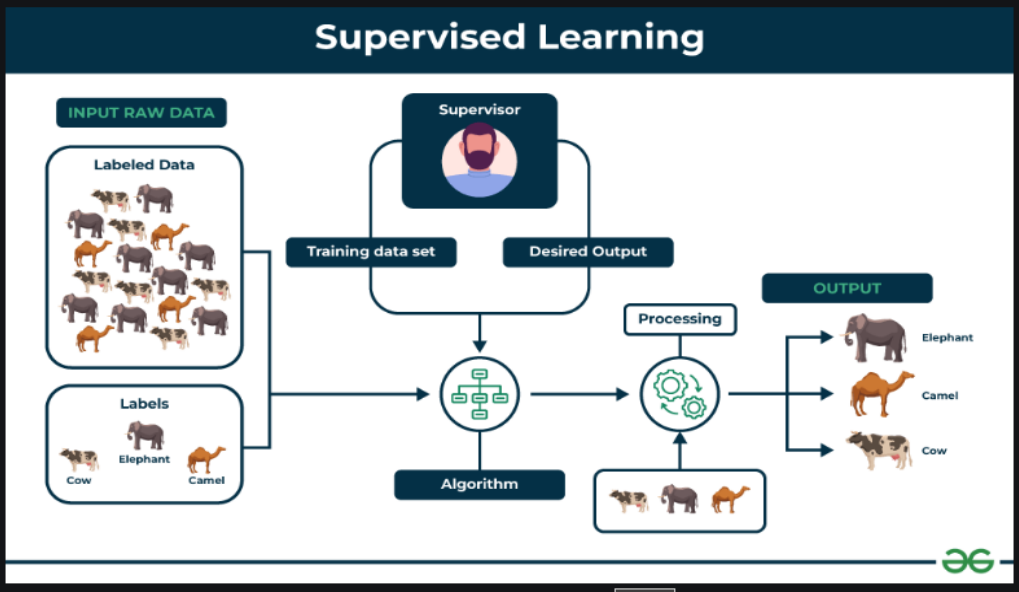
1.6.1 Comparision between Supervised, Unsupervised, Semi-supervised Learning

Sr. No.	Supervised learning	Unsupervised learning	Semi-supervised learning
1.	Input data is labeled.	Input data is unlabeled.	A large amount of input data is unlabeled while a small amount is labeled.
2.	Trying to predict a specific quantity.	Trying to understand the data.	Using unsupervised methods to improve supervised algorithm.
3.	Used in Fraud detection.	Used in Identity management.	Used in spam detection.
4.	Subtype : Classification and regression.	Subtype : Clustering and association.	Subtype : Classification, regression, clustering and association.
5.	Higher accuracy.	Lesser accuracy.	Lesser accuracy.

READ IT ONCE...

1. 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.



Supervised Learning

Let's understand it with the help of an example.

Example: Consider a scenario where you have to build an image classifier to differentiate between cats and dogs. If you feed the datasets of dogs and cats labelled images to the algorithm, the machine will learn to classify between a dog or a cat from these labeled images. When we input new dog or cat images that it has never seen before, it will use the learned algorithms and predict whether it is a dog or a cat. This is how **supervised learning** works, and this is particularly an image classification.

There are two main categories of supervised learning that are mentioned below:

- Classification
- Regression

Advantages of Supervised Machine Learning

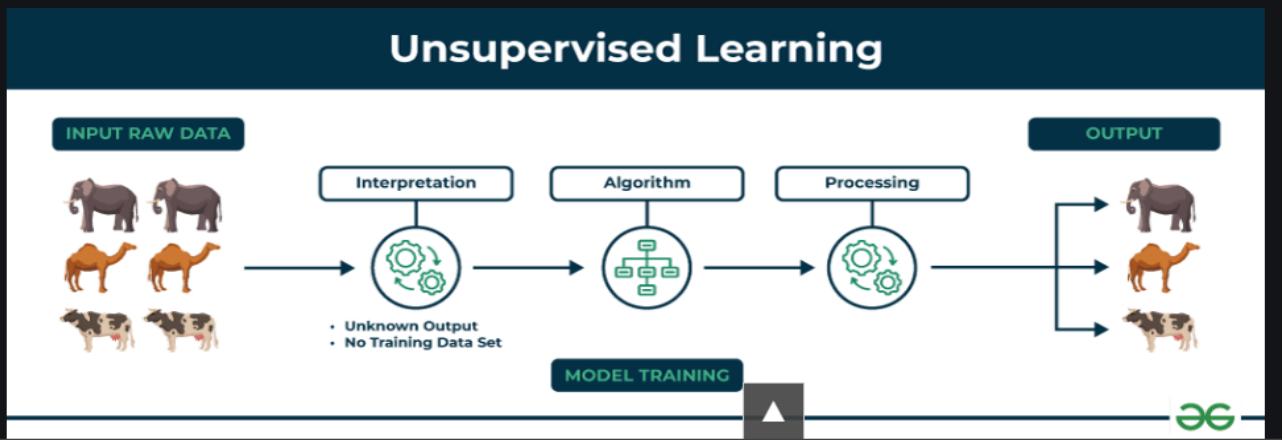
- **Supervised Learning** models can have high accuracy as they are trained on **labelled data**.
- The process of decision-making in supervised learning models is often interpretable.
- It can often be used in pre-trained models which saves time and resources when developing new models from scratch.

Disadvantages of Supervised Machine Learning

- It has limitations in knowing patterns and may struggle with unseen or unexpected patterns that are not present in the training data.
- It can be time-consuming and costly as it relies on **labeled** data only.
- It may lead to poor generalizations based on new data.

2. Unsupervised Machine Learning

Unsupervised Learning Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data. Unlike supervised learning, unsupervised learning doesn't involve providing the algorithm with labeled target outputs. The primary goal of Unsupervised learning is often to discover hidden patterns, similarities, or clusters within the data, which can then be used for various purposes, such as data exploration, visualization, dimensionality reduction, and more.



Let's understand it with the help of an example.

Example: Consider that you have a dataset that contains information about the purchases you made from the shop. Through clustering, the algorithm can group the same purchasing behavior among you and other customers, which reveals potential customers without predefined labels. This type of information can help businesses get target customers as well as identify outliers.

There are two main categories of unsupervised learning that are mentioned below:

- Clustering
- Association

Advantages of Unsupervised Machine Learning

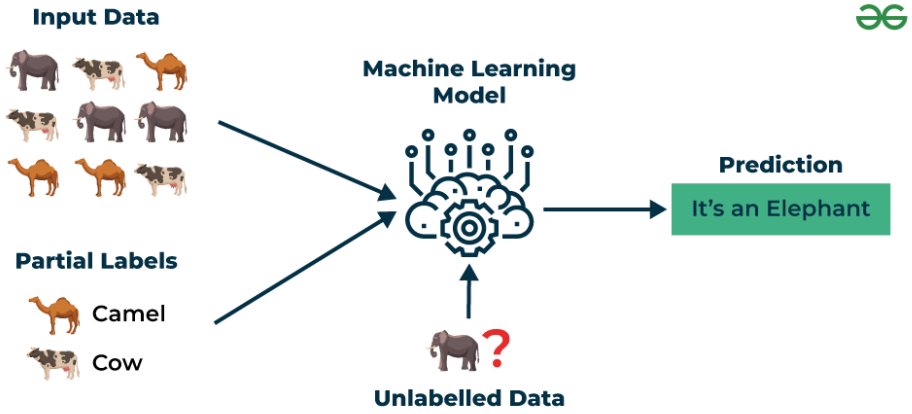
- It helps to discover hidden patterns and various relationships between the data.
- Used for tasks such as **customer segmentation, anomaly detection,** and **data exploration**.
- It does not require labeled data and reduces the effort of data labeling.

Disadvantages of Unsupervised Machine Learning

- Without using labels, it may be difficult to predict the quality of the model's output.
- Cluster Interpretability may not be clear and may not have meaningful interpretations.
- It has techniques such as autoencoders and dimensionality reduction that can be used to extract meaningful features from raw data.

Semi-Supervised learning is a machine learning algorithm that works between the supervised and unsupervised learning so it uses both **labelled and unlabelled** data. It's particularly useful when obtaining labeled data is costly, time-consuming, or resource-intensive. This approach is useful when the dataset is expensive and time-consuming. Semi-supervised learning is chosen when labeled data requires skills and relevant resources in order to train or learn from it.

We use these techniques when we are dealing with data that is a little bit labeled and the rest large portion of it is unlabeled. We can use the unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in the case of image data sets where usually all images are not labeled.



Example: Consider that we are building a language translation model, having labeled translations for every sentence pair can be resources intensive. It allows the models to learn from labeled and unlabeled sentence pairs, making them more accurate. This technique has led to significant improvements in the quality of machine translation services.

Types of Semi-Supervised Learning Methods

There are a number of different semi-supervised learning methods each with its own characteristics. Some of the most common ones include:

- **Graph-based semi-supervised learning:** This approach uses a graph to represent the relationships between the data points. The graph is then used to propagate labels from the labeled data points to the unlabeled data points.
- **Label propagation:** This approach iteratively propagates labels from the labeled data points to the unlabeled data points, based on the similarities between the data points.
- **Co-training:** This approach trains two different machine learning models on different subsets of the unlabeled data. The two models are then used to label each other's predictions.

- **Self-training:** This approach trains a machine learning model on the labeled data and then uses the model to predict labels for the unlabeled data. The model is then retrained on the labeled data and the predicted labels for the unlabeled data.
- Generative adversarial networks (GANs): GANs are a type of deep learning algorithm that can be used to generate synthetic data. GANs can be used to generate unlabeled data for semi-supervised learning by training two neural networks, a generator and a discriminator.

Advantages of Semi- Supervised Machine Learning

- It leads to better generalization as compared to **supervised learning,** as it takes both labeled and unlabeled data.
- Can be applied to a wide range of data.

Disadvantages of Semi- Supervised Machine Learning

- **Semi-supervised** methods can be more complex to implement compared to other approaches.
- It still requires some **labeled data** that might not always be available or easy to obtain.
- The unlabeled data can impact the model performance accordingly.

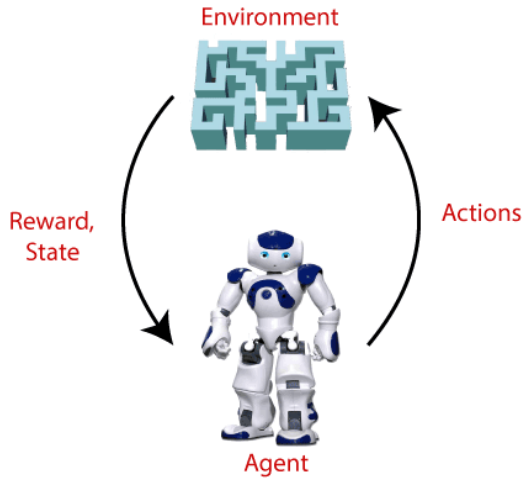
Review Question

1. *Discuss the reinforcement learning and write the brief applications.*

SPPU : March-20, In Sem, Marks 5

What is Reinforcement Learning?

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning.
- Since there is no labeled data, so the agent is bound to learn by its experience only.
- RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as **game-playing, robotics**, etc.
- The agent interacts with the environment and explores it by itself. The primary goal of an agent in reinforcement learning is to improve the performance by getting the maximum positive rewards.
- The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that ***"Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that."*** How a Robotic dog learns the movement of his arms is an example of Reinforcement learning.
- It is a core part of Artificial intelligence, and all AI agent works on the concept of reinforcement learning. Here we do not need to pre-program the agent, as it learns from its own experience without any human intervention.
- **Example:** Suppose there is an AI agent present within a maze environment, and his goal is to find the diamond. The agent interacts with the environment by performing some actions, and based on those actions, the state of the agent gets changed, and it also receives a reward or penalty as feedback.
- The agent continues doing these three things (**take action, change state/remain in the same state, and get feedback**), and by doing these actions, he learns and explores the environment.
- The agent learns that what actions lead to positive feedback or rewards and what actions lead to negative feedback penalty. As a positive reward, the agent gets a positive point, and as a penalty, it gets a negative point.



Terms used in Reinforcement Learning

- **Agent():** An entity that can perceive/explore the environment and act upon it.
- **Environment():** A situation in which an agent is present or surrounded by. In RL, we assume the stochast environment, which means it is random in nature.
- **Action():** Actions are the moves taken by an agent within the environment.
- **State():** State is a situation returned by the environment after each action taken by the agent.
- **Reward():** A feedback returned to the agent from the environment to evaluate the action of the agent.
- **Policy():** Policy is a strategy applied by the agent for the next action based on the current state.
- **Value():** It is expected long-term retuned with the discount factor and opposite to the short-term reward.
- **Q-value():** It is mostly similar to the value, but it takes one additional parameter as a current action (a).

1.7.2 Application of Reinforcement Learning

1. Robotics : Robots with pre-programmed behavior are useful in structured environments, such as the assembly line of an automobile manufacturing plant, where the task is repetitive in nature.
2. A master chess player makes a move. The choice is informed both by planning, anticipating possible replies and counter replies.
3. An adaptive controller adjusts parameters of a petroleum refinery's operation in real time.

1.7.3 Advantages and Disadvantages of Reinforcement Learning

Advantages of Reinforcement learning

1. Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
2. The model can correct the errors that occurred during the training process.
3. In RL, training data is obtained via the direct interaction of the agent with the environment

Disadvantages of Reinforcement learning

1. Reinforcement learning is not preferable to use for solving simple problems.
2. Reinforcement learning needs a lot of data and a lot of computation

Applications of Reinforcement Learning (RL)
Gaming: Used in AI for games (e.g., AlphaGo) and automating game design/testing.
Robotics: Powers autonomous robots and optimizes industrial automation.
Healthcare: Personalizes treatment plans and aids in medical imaging.

Advantages of Reinforcement Learning
Autonomy: Learns through trial and error without needing labeled data.
Adaptability: Adapts to changing environments dynamically.
Optimizes Long-Term Goals: Focuses on maximizing cumulative rewards over time.

Disadvantages of Reinforcement Learning
Complexity: Requires significant computational resources and time.
Exploration vs. Exploitation: Balancing exploration and exploitation is challenging.
Unpredictability: Outcomes can be uncertain, especially in complex environments.

MODELS OF ML:

1) ### Geometric Models in Machine Learning

Definition:

Geometric models use geometric concepts to represent and analyze data. These models help in understanding data structures, connections, and patterns through spatial relationships and geometrical representations.

Key Examples:

1. **Convolutional Neural Networks (CNNs):**
   - **Purpose:** Primarily used for image recognition.
   - **How It Works:** Applies filters to capture local features and spatial patterns in images.

2. **Graph Neural Networks (GNNs):**
   - **Purpose:** Analyzes data structured as graphs (nodes and edges).
   - **How It Works:** Processes the graph's structure to make predictions, useful in recommendation systems and social network analysis.

3. **Support Vector Machines (SVMs):**
   - **Purpose:** Used for classification and regression.
   - **How It Works:** Finds the best hyperplane to separate different classes or fit the data, even in high-dimensional spaces.

4. **Self-Organizing Maps (SOMs):**
   - **Purpose:** Useful for clustering and visualization.
   - **How It Works:** Creates a low-dimensional map from high-dimensional data, preserving its geometric structure.

5. **Principal Component Analysis (PCA):**
   - **Purpose:** Reduces data dimensionality.

- **How It Works:** Identifies principal components (directions of greatest variance) to simplify data while retaining key features.

Challenges:

- **High Dimensionality:** Models can become complex and computationally expensive as dimensions increase.

- **Sensitivity to Representation:** Model performance can vary with how data is represented.

- **Scalability Issues:** Large datasets can overwhelm some models, leading to high computational costs.

- **Noise and Outliers:** Small disturbances in data can significantly impact model accuracy.

Applications:

- **Feature Extraction:** Identifying and using features like edges or shapes in images.

- **Object Detection:** Locating and identifying objects in images or videos.

- **Pose Estimation:** Determining the position and orientation of objects or people.

- **Shape Analysis:** Comparing and classifying shapes in various applications.

- **Dimensionality Reduction:** Simplifying data while preserving important structures.

Advantages:

- **Handles Nonlinearity:** Can capture complex, nonlinear relationships in data.

- **Reduced Feature Engineering:** Automatically learns complex features.

- **Interpretability:** Provides understandable representations of data.

- **Transferability:** Learned features can be applied to different tasks or datasets.

Conclusion:

Geometric models are crucial in machine learning for their ability to represent and analyze complex data structures. They play a significant role in tasks such as classification, clustering, and dimensionality reduction, offering insights and accurate predictions based on geometric relationships in the data.

2)### Probabilistic Models in Machine Learning

What are Probabilistic Models?

Probabilistic models are used in machine learning to make predictions based on probabilities rather than fixed outcomes. They help in understanding and handling uncertainty in real-world data.

Types of Probabilistic Models:

1. **Generative Models:**

- **What They Do:** Generate new data by learning the overall distribution of input and output data.

- **Examples:** Models that can create new images or text similar to the training data.

- **Uses:** Image generation, speech synthesis.

2. **Discriminative Models:**

- **What They Do:** Focus on predicting the output based on the input, often by drawing boundaries between different classes.

- **Examples:** Models used for tasks like identifying objects in images.

- **Uses:** Image recognition, sentiment analysis.

3. **Graphical Models:**

- **What They Do:** Use graphs to show relationships between different variables.

- **Examples:** Models that represent how different factors are related in a network.

- **Uses:** Understanding relationships in data, like in language processing.

Naive Bayes Algorithm:

- **What It Is:** A simple and fast algorithm used for classification tasks.

- **How It Works:**

1. **Calculate Probabilities:** For each feature, figure out how likely it is to appear with each class.

2. **Predict Class:** Based on these probabilities, predict the class that is most likely.

- **Uses:** Email spam detection, medical diagnoses.

Probabilistic Models in Deep Learning:

- **Purpose:** Improve accuracy by considering uncertainty in predictions.

- **How It Helps:** It allows deep learning models to better generalize by accounting for uncertainty.

Advantages:

- **Manages Uncertainty:** Handles unpredictable data well.
- **Reveals Insights:** Helps understand how different factors affect outcomes.
- **Adapts to New Data:** Can update predictions based on new information.

Disadvantages:

- **Risk of Overfitting:** May work well on training data but not on new data.
- **Computationally Demanding:** Can require a lot of computing power.
- **Not Always Suitable:** Some data types may not work well with probabilistic models.

Conclusion:

Probabilistic models are essential in machine learning for dealing with uncertainty and making more accurate predictions. They are powerful tools but need careful handling to avoid issues like overfitting and high computational demands.

Distance Based Models:

Review Question

1. *What do you mean by distance metric and exemplar ? Explain different types of distances, measures.* **SPPU : Dec.-19, Marks 9**

1. Distance Metric:

A distance metric is a way to measure how far or close two points are from each other. It's used in algorithms like K-Nearest Neighbors (KNN) to determine the similarity between data points.

2. Exemplar:

An exemplar is a data point from the training set that is used as a reference to predict the class or value of new data points. In KNN, exemplars are the data points that are closest to the new, unclassified data point.

3. Different Types of Distances (Measures) with Formulas:

a) Euclidean Distance:

- **Explanation:** It is the straight-line distance between two points in a space. It's the most common distance metric used in KNN.
- **Formula:**

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where:

- x_i and y_i are the coordinates of the two points in n-dimensional space.

b) Manhattan Distance:

- **Explanation:** It is the total distance between two points if you only move along the axes (like walking along streets in a grid). It's also known as "taxicab" or "L1" distance.
- **Formula:**

$$\text{Manhattan Distance} = \sum_{i=1}^{n} |x_i - y_i|$$

Where:

- x_i and y_i are the coordinates of the two points in n-dimensional space.

c) Minkowski Distance:

- **Explanation:** It is a generalized distance metric that includes both Euclidean and Manhattan distances as special cases. The value of p determines the type of distance.
- **Formula:**

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Where:

- p is a parameter that defines the type of distance (p=1 gives Manhattan Distance, p=2 gives Euclidean Distance).

d) Hamming Distance:

- **Explanation:** It counts the number of positions at which the corresponding symbols (bits, letters, etc.) are different. It's often used for categorical data, like strings or binary vectors.
- **Formula:**

$$\text{Hamming Distance} = \sum_{i=1}^{n} \delta(x_i, y_i)$$

Where:

- $\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$

Conclusion:

These distance metrics are essential tools in machine learning, particularly for algorithms like KNN, where understanding the similarity or difference between data points is crucial for making accurate predictions.

1.11 Grouping and Grading Models

- Grading vs grouping is an orthogonal categorization to geometric - probabilistic - logical-compositional model. Difference between grouping and grading models is the way they handle the instance space.

Grouping Model :

- Grouping models break the instance space up into groups or segments and in each segment apply a very simple method. Example : Decision tree, KNN.
- Grouping models have fixed resolution. They cannot distinguish instances beyond a resolution. At the finest resolution, grouping models assign the majority class to all instances that fall into the segment. Find the right segments and label all the objects in that segment.

Grading Model :

- Grading models form one global model over the instance space. They don't use the notion of segment.
- Grading models are usually able to distinguish between arbitrary instances, no matter how similar they are.

b) Describe parametric and Non-parametric machine learning models. [5]

Parametric and Non-Parametric Learning Models

Parametric Learning Models:

1. **What Are They?**

- These models assume that data fits a specific type of pattern or distribution, like a normal curve.

- They use a set number of parameters (e.g., mean and standard deviation) to define this pattern.

2. **Examples:**

- **Linear Regression:** Finds a straight line that best fits the data to predict outcomes.

- **Logistic Regression:** Predicts if something falls into one of two categories (like yes or no).

3. **Advantages:**

- **Efficient:** Works well with less data if the assumptions are correct.

- **Effective:** Good for simpler problems with clear patterns.

4. **Disadvantages:**

- **Assumptions:** If the data doesn't fit the assumed pattern, the model might not work well.

- **Less Flexible:** Struggles with complex or unusual patterns in data.

Non-Parametric Learning Models:

1. **What Are They?**

- These models don't assume a specific pattern or distribution. They adapt to whatever pattern the data shows.

- They don't use a fixed number of parameters and can adjust as needed.

2. **Examples:**

- **K-Nearest Neighbors (KNN):** Classifies data by looking at the most common category among the nearest neighbors.

- **Decision Trees:** Makes decisions by asking a series of yes/no questions based on the data features.

3. **Advantages:**

- **Flexible:** Can handle a wide range of patterns and relationships in data.

- **Robust:** Works well even if the data is messy or has outliers.

4. **Disadvantages:**

- **Data Hungry:** May need a lot of data to make accurate predictions.

- **Computationally Heavy:** Can be slower and more complex, especially with large datasets.

In Summary:

- **Parametric models** work best when you know or assume a specific pattern in the data. They are simpler and faster but less flexible.

- **Non-parametric models** are more flexible and can handle more complex data patterns, but they may need more data and be slower to process.

A parametric model in machine learning is a type of model that summarizes data through a set of fixed parameters of a predetermined size. Unlike non-parametric models, which have an unlimited number of parameters that grow with the amount of data, parametric models have a fixed number of parameters that do not change regardless of the size of the data set.

Key Characteristics of Parametric Models:
Fixed Number of Parameters:

Parametric models assume that the underlying data distribution can be described by a specific set of parameters. These parameters are fixed in number, regardless of the size of the dataset.
Examples include the coefficients in linear regression, the mean and variance in a Gaussian distribution, and weights in a neural network.
Assumptions About the Data:

Parametric models make specific assumptions about the form of the data distribution. For example, linear regression assumes that the relationship between the independent and dependent variables is linear.
These assumptions simplify the model but can limit its flexibility to capture complex patterns in the data.
Training Process:

The training process in a parametric model involves estimating the values of the model's parameters that best fit the training data. This is often done using optimization techniques like gradient descent to minimize a loss function, such as the mean squared error in linear regression.
Efficiency:

Parametric models are computationally efficient because they summarize the data with a fixed number of parameters. This makes them easier to implement and faster to compute, especially on large datasets.
Interpretability:

Parametric models are often more interpretable because the effect of each parameter on the output can be directly understood. For example, in linear regression, each coefficient represents the change in the output variable for a one-unit change in the corresponding input variable.
Examples of Parametric Models:

Linear Regression: Predicts the output by a linear combination of the input features.
Logistic Regression: Used for binary classification problems, predicting the probability of a binary outcome.
Naive Bayes: A probabilistic model that assumes independence between features.
Neural Networks: Though complex, neural networks are parametric because the number of parameters (weights and biases) is fixed once the architecture is defined.
Advantages:

Simplicity: The fixed number of parameters makes parametric models simpler to understand and use.
Speed: These models are usually faster to train and predict, making them suitable for real-time applications.

1.12 Parametric Models

- Model can be represented using a pre - determined number of parameters. These methods in Machine Learning typically take a model - based approach. We make an assumption there with respect to the form of the function to be guessed. Then we choose an appropriate model based on this assumption correct to estimate the set of parameters.
- The advantage of the parametric approach is that the model is defined up to a small number of parameters, for example mean and variance, the sufficient statistics of the distribution. Once those parameters are estimated from the sample, the whole distribution is known.

- We estimate the parameters of the distribution from the given sample, plug in these estimates to the assumed model and get an estimated distribution, which we then use to make a decision. The method we use to estimate the parameters of a distribution is maximum likelihood estimation.
- Examples of parametric machine learning algorithms are Logistic Regression, Linear Discriminant Analysis, Perceptron, Naive Bayes and Simple Neural Networks.
- **Advantages :**
  1. These methods are simpler and easier to understand.
  2. These models are very rapid to learn from data.
  3. They do not need as much training data.
  4. The methods are well - matched to simpler problems.

Aspect	Parametric Models	Non-Parametric Models
Assumptions	Assume a specific pattern or distribution (e.g., normal curve).	Don't assume a specific pattern or distribution.
Parameters	Use a fixed number of parameters (e.g., mean, variance).	Don't use a fixed number of parameters.
Examples	Linear Regression, Logistic Regression	K-Nearest Neighbors (KNN), Decision Trees
Flexibility	Less flexible, better for simpler patterns.	More flexible, adapts to complex patterns.
Data Requirements	Works well with less data if assumptions are correct.	May need more data to work effectively.
Handling Complexity	Struggles with complex or unusual patterns.	Handles complex or noisy data better.
Computational Cost	Generally faster and simpler to compute.	Can be slower and more computationally intensive.
Advantages	Efficient and effective for well-defined patterns.	Flexible and robust, can handle a variety of data types.
Disadvantages	Sensitive to incorrect assumptions, less adaptable.	May require large datasets and can be computationally heavy.

1.13.1 Difference between Non-parametric Methods and Parametric Methods

Sr. No.	Non-parametric method	Parametric methods
1.	Algorithms that do not make particular assumptions about the kind of mapping function are known as non-parametric algorithms.	Parametric model is a learner that summarizes data through a collection of parameters.
2.	Non-parametric analysis to test group medians.	Parametric analysis to test group means.
3.	It can be used on small samples.	Tend to need larger samples.
4.	No information about the population is available.	Information about population is completely known.
5.	It can be used on ordinal and nominal scale data.	Used mainly on interval and ratio scale data.
6.	Not necessarily the samples are independent.	Samples are independent.
7.	K-nearest neighbors is an example of a non - parametric algorithm.	Examples of parametric models include logistic regression and linear SVM.

c) Explain various Data formats that conform ML elements. [5]

Based on the provided information, here's a concise 5-mark answer on different data formats in the context of machine learning:

Different Data Formats in Machine Learning

In machine learning, the data format significantly impacts how algorithms process and learn from the data. Here are key data formats relevant to machine learning:

1. **Numeric Data**:

- **Format**: Data represented in numerical values.

- **Example**: Age, income, or sensor readings.

- **Usage**: Used in various algorithms including regression and classification to find patterns and make predictions.

2. **Categorical Data**:

- **Format**: Data representing categories or labels.

- **Example**: Gender, type of fruit.

- **Usage**: Useful for classification tasks where data is categorized into distinct groups.

3. **Ordinal Data**:

- **Format**: Categorical data with an inherent order.

- **Example**: Clothing sizes (small, medium, large), satisfaction ratings.

- **Usage**: Applied in models that need to account for the ranking or order of categories.

4. **Text Data**:

- **Format**: Raw or structured text.

- **Example**: Customer reviews, social media posts.

- **Usage**: Essential for natural language processing tasks like sentiment analysis and text classification.

5. **Time-Series Data**:

- **Format**: Data points indexed in time order.

- **Example**: Stock prices over time, daily temperature readings.

- **Usage**: Used for forecasting and analyzing temporal patterns in time-series analysis.

6. **Image Data**:

- **Format**: Pixel values representing images.

- **Example**: Photographs, medical scans.

- **Usage**: Critical for computer vision tasks including image classification and object detection.

Splitting Data:

- **Training Data**: Used to train the model.

- **Validation Data**: Used to tune hyperparameters and validate the model during training.

- **Testing Data**: Used to evaluate the final model's performance and ensure it generalizes well to unseen data.

Understanding these data formats and how to preprocess them is crucial for effective machine learning model training and evaluation.

Common Data Formats:
NHWC (Batch Size, Height, Width, Channels):

Explanation: In the NHWC format, data is stored as a 4D array where the dimensions are ordered as Batch size, Height, Width, and Channels. This format is commonly used by frameworks like TensorFlow, particularly on CPUs and some GPUs, as it aligns well with how data is processed in row-major order.
Use Case: It is particularly useful when operations like convolution require accessing pixels across channels, as consecutive memory access is faster.
NCHW (Batch Size, Channels, Height, Width):

Explanation: In the NCHW format, data is stored as a 4D array with dimensions ordered as Batch size, Channels, Height, and Width. This format is often preferred by deep learning frameworks like PyTorch and some GPUs, as it can optimize memory access patterns for certain convolution operations.
Use Case: It's particularly efficient on GPUs where accessing channel data sequentially can lead to performance improvements.
NCDHW (Batch Size, Channels, Depth, Height, Width):

Explanation: The NCDHW format is an extension to 5D data, often used in 3D convolutional neural networks (CNNs) that work with volumetric data, such as medical imaging or video processing. Here, the dimensions are ordered as Batch size, Channels, Depth, Height, and Width.
Use Case: Useful in 3D image processing tasks where each data point is a 3D volume rather than a 2D image.
NDHWC (Batch Size, Depth, Height, Width, Channels):

Explanation: The NDHWC format is another 5D data format where the dimensions are ordered as Batch size, Depth, Height, Width, and Channels. This format can also be used in 3D CNNs, particularly in systems optimized for this memory layout.
Use Case: It is sometimes used for 3D image processing when the system or framework is optimized for this data layout.

1.14.1 Data Formats

- Supervised learning always use a dataset, defined as a finite set of real vectors with m features.
- In training data, data are assign the labels. In test data, data labels are unknown but not given. The training data consist of a set of training examples.
- The real aim of supervised learning is to do well on test data that is not known during learning. Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.
- The training error is the mean error over the training sample. The test error is the expected prediction error over an independent test sample.
- **Training set :** A set of examples used for learning, where the target value is known.
- **Test set :** It is used only to assess the performances of a classifier. It is never used during the training process so that the error on the test set provides an unbiased estimate of the generalization error.
- Training data is the knowledge about the data source which we use to construct the classifier.
- Data format is expressed as

  $$X = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\}$$

- Machine learning can be summarized as learning a function (f) that maps input variables (X) to output variables (Y).

  $$Y = f(x)$$

- An algorithm learns this target mapping function from training data.

- The form of the function is unknown, so our job as machine learning practitioners is to evaluate different machine learning algorithms and see which is better at approximating the underlying function.
- Different algorithms make different assumptions or biases about the form of the function and how it can be learned.
- Parametric : "A learning model that summarizes data with a set of parameters of fixed size is called a parametric model. No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs." Basically it includes normal distribution and other known distributions.
- Non-parametric : "Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features."
- Parametric : Data are drawn from a probability distribution of specific form up to unknown parameters.
- Nonparametric : Data are drawn from a certain unspecified probability distribution.

Learnability in Machine Learning

Learnability refers to a model's ability to improve its performance by learning from data. It involves several key factors:

1. **Quality and Quantity of Data**: High-quality and sufficient data help models learn patterns more effectively.

2. **Feature Selection**: Choosing relevant features enhances the model's ability to identify important patterns.

3. **Model Complexity**: Balancing complexity prevents overfitting (too complex) or underfitting (too simple) of the model.

4. **Training Algorithms**: The choice of algorithms and their parameters affects how well the model learns.

Types of Learnability:

- **Supervised Learning**: Learning from labeled data to predict outcomes.

- **Unsupervised Learning**: Identifying patterns in unlabeled data.

- **Semi-Supervised Learning**: Combining labeled and unlabeled data to improve learning.

Challenges:

- **Overfitting**: Model learns noise in the data, reducing performance on new data.

- **Underfitting**: Model is too simple to capture the underlying patterns.

In essence, learnability is about how well a model can adapt and make accurate predictions based on the data it is trained on.

Underfitting and Overfitting in Machine Learning

Underfitting:

- **Definition**: Occurs when a model is too simple to capture the underlying patterns in the data, leading to poor performance on both training and testing datasets.

- **Example**: Using a linear regression model to fit a dataset with a complex, non-linear relationship. For instance, predicting house prices with just one feature (e.g., number of rooms) when the actual relationship involves multiple features (e.g., location, age of the house).

Overfitting:

- **Definition**: Happens when a model is too complex and learns not only the underlying patterns but also the noise in the training data. This results in excellent performance on training data but poor performance on new, unseen data.

- **Example**: Using a very deep neural network with many layers for a simple dataset. For instance, a model that performs extremely well on a training set of handwritten digits but fails to generalize to new digits because it has memorized the training data rather than learning the general patterns.

Summary:

- **Underfitting**: Model is too simple; fails to capture trends.

- **Overfitting**: Model is too complex; captures noise and performs poorly on new data.

Statistical learning is a powerful tool in machine learning that allows us to build models from data. It's all about understanding the relationship between input data and outputs, finding the right balance between model complexity and performance, and making sure the model can generalize well to new data. By mastering these concepts, you can build effective machine learning models that make accurate predictions or reveal hidden patterns in the data.

Statistical Learning

1. **Definition**: Statistical learning is a technique in machine learning used to model and understand relationships between variables in a dataset. It aims to make predictions or identify patterns using statistical methods.

2. **Goal**: The goal is to build a model that can learn from data and make accurate predictions or inferences.

3. **Approach**: It involves analyzing data to estimate relationships and make predictions based on statistical methods.

Maximum Likelihood Estimation (MLE)

1. **Definition**: MLE is a method used to estimate the parameters of a statistical model. It finds the parameter values that make the observed data most probable.

2. **Steps**:

   1. **Define the Likelihood Function**: Formulate a likelihood function based on the probability distribution of the data. This function represents how likely it is to observe the given data under different parameter values.

   2. **Collect Data**: Gather the data that will be used to estimate the parameters. This data could be anything relevant to the model you are working on.

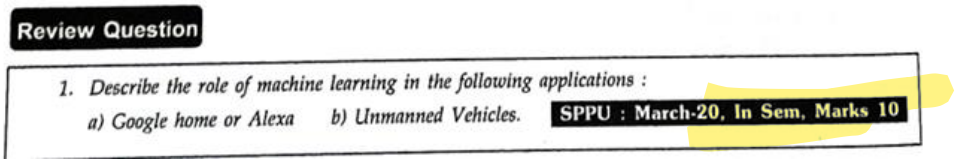
   3. **Estimate Parameters**: Use the likelihood function to find the parameter values that maximize the likelihood. This involves finding the parameter values that make the observed data most likely.

   4. **Optimize**: Solve the optimization problem, usually through mathematical techniques or numerical methods, to find the parameter values that maximize the likelihood function.

3. **Example**: Suppose you have data on the heights of people and want to estimate the mean height and variance of the population. MLE will help you find the mean and variance that make the observed heights most probable, based on the normal distribution model.

In summary, **Statistical Learning** helps in building predictive models, and **MLE** is a technique used within this framework to find the best model parameters by maximizing the probability of observing the given data.

Application of ml:


Review Question

1. *Describe the role of machine learning in the following applications :*
   a) Google home or Alexa b) Unmanned Vehicles. **SPPU : March-20, In Sem, Marks 10**

1. Google Home and Alexa

Introduction:

Google Home and Amazon Alexa are advanced voice-activated assistants that leverage machine learning (ML) to understand and respond to user commands. These smart devices use ML to process and analyze voice data, providing accurate and personalized interactions.

Roles:

1. **Speech Recognition**:

   - **Function**: Converts spoken language into text.

   - **Machine Learning Role**: Utilizes Deep Learning models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) to transcribe speech accurately, handling various accents and noisy environments.

2. **Natural Language Processing (NLP)**:

   - **Function**: Interprets and understands the meaning of user commands.

   - **Machine Learning Role**: Applies NLP techniques to parse and understand user queries. Algorithms like Named Entity Recognition (NER) help identify key elements of commands.

3. **Intent Recognition**:

   - **Function**: Determines the user's intention behind a command.

- **Machine Learning Role**: Uses classification algorithms to map user inputs to specific intents. For example, distinguishing between a request to "play music" and "set a reminder."

4. **Personalization**:

- **Function**: Customizes responses and actions based on user preferences.

- **Machine Learning Role**: Implements algorithms to analyze user interaction patterns and preferences, providing personalized responses and recommendations.

5. **Contextual Understanding**:

- **Function**: Maintains context over interactions for coherent responses.

- **Machine Learning Role**: Utilizes models to understand and remember previous interactions, allowing for more natural and relevant conversations.

Example:

- **Google Home** uses ML to manage tasks such as setting reminders and controlling smart home devices. **Amazon Alexa** employs ML to play music, manage lists, and provide answers based on voice commands.

2. Unnamed Autonomous Vehicles

Introduction:

Unnamed autonomous vehicles (self-driving cars) use machine learning to navigate and operate without human intervention. ML processes data from various sensors to ensure safe and efficient driving.

Roles:

1. **Perception**:

- **Function**: Detects and identifies objects and obstacles around the vehicle.

- **Machine Learning Role**: Employs Convolutional Neural Networks (CNNs) to analyze camera images and integrate data from LiDAR and radar for a comprehensive view of the environment.

2. **Object Detection and Classification**:

- **Function**: Identifies and categorizes objects such as pedestrians, vehicles, and road signs.

- **Machine Learning Role**: Uses algorithms like YOLO (You Only Look Once) and Faster R-CNN to detect and classify objects in real-time.

3. **Localization**:

- **Function**: Determines the vehicle's precise location on the map.

- **Machine Learning Role**: Applies techniques like map matching and Simultaneous Localization and Mapping (SLAM) to compare sensor data with detailed maps for accurate positioning.

4. **Decision Making**:

- **Function**: Makes driving decisions based on road conditions and traffic rules.

- **Machine Learning Role**: Utilizes Reinforcement Learning (RL) and Decision Trees to evaluate different driving scenarios and choose safe and effective actions.

5. **Control and Navigation**:

- **Function**: Manages vehicle movement, including steering, acceleration, and braking.

- **Machine Learning Role**: Implements control algorithms and deep learning models to translate driving decisions into precise vehicle actions for smooth operation.

Example:

- An **unnamed autonomous vehicle** uses machine learning to safely navigate through traffic, avoid obstacles, and adjust driving behavior based on real-time data from its sensors.

Conclusion:

Machine learning is essential in enhancing the functionality of smart assistants like Google Home and Alexa by processing and understanding voice commands. For autonomous vehicles, ML enables safe and efficient driving by analyzing sensor data, making decisions, and controlling vehicle movements.