CHAPTER No. 1

INTRODUCTION TO MACHIN LEARNING (Part1)

Machine learning (ML) is a subset of artificial intelligent (AI), its basic aim is to design a system by using algorithms such that it create ability in computer to learn as human brain do through their experience or observation which lead them to their thinking and predictive action ,finally at the end to make decision that resemble with human action.ML enable to move toward future change which come from improving our past learning experience and observation performance on specific field by time. As many research and development is ongoing in ML ,hence constantly new algorithms and techniques are being created to introduce new capabilities in world. Today, ML application are using in various area such as healthcare, weather forecasting , recommendation system, image or signal processing, data visualization, data compression, natural language processing, speech recognition, financial analysis ,market analysis, autonomous vehicles and many more.

As due to the continuous advancement in technology and growing of challenges ,ML is concern to ensuring data quality, preventing bias, and interpretability of system and play main role to bring change in the future of AI driven application and system.

Machine learning (ML) are broadly categorized into three types:

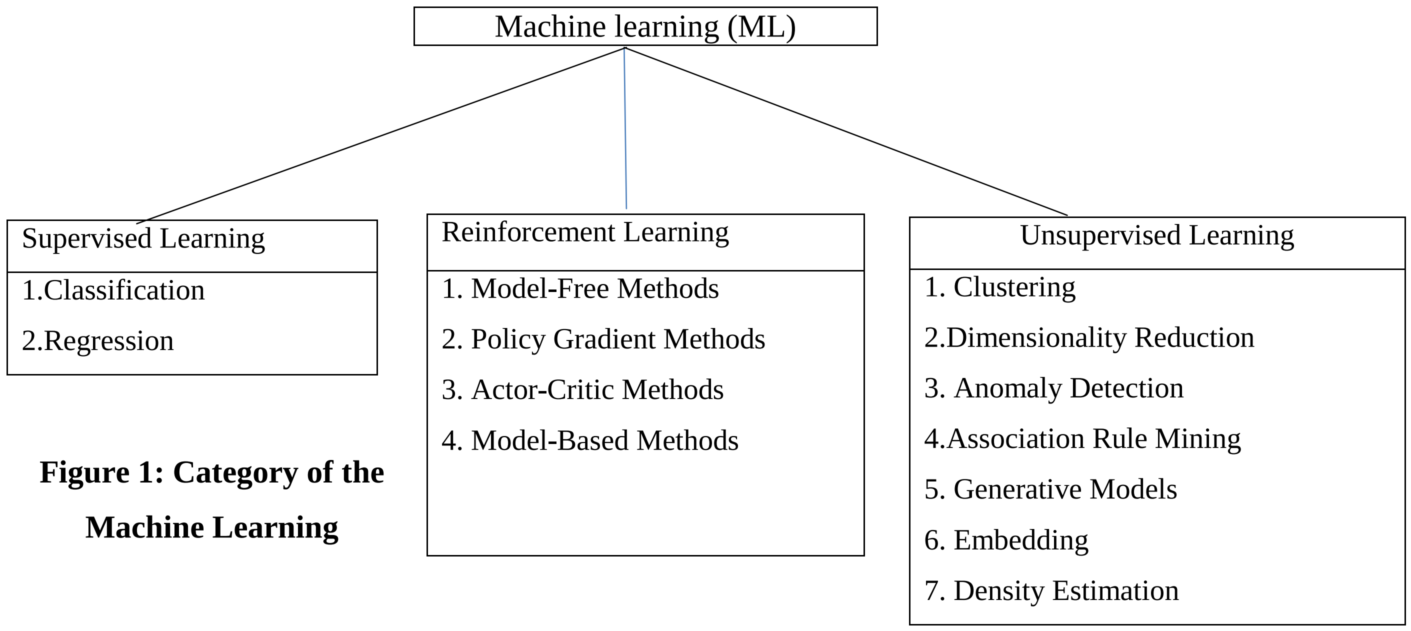

Figure 1: Category of the Machine Learning

1. Types of Machine Learning

1.1. Supervised Learning: In supervised Learning first step is to trained the model by enabling the algorithm to learn from the labeled dataset, so that it easily map input data points based on features to their corresponding output data points based on features. During the training phase , it is important to adjust model parameters , in order to minimize the differences between its prediction and output features. Finally the model is trained to make prediction or decision on new data.

Process of Supervised Learning:

It is a several phase processes which include to build, trained, evaluate and develop a machine learning model that gain the capabilities to predict or to make decision on new data. The processes of Supervised Learning include the following:

1. **Data Collection:**

Data are gathered from labeled dataset which contain input and their corresponding output or target labels. The main role of this dataset is that , it should be represent the problem which you want to solve and in addition to cover a diverse range of examples.

2. **Data Preprocessing:**

It is essential to standardized the data into suitable format before sending it to training phase. This include task such as handling partial content of dataset, converting categorical variables into numerical values and at last ,split the data into two set in ratio of approximately 70 : 30 , first data ratio is for training set and second data ratio is for testing set.

3. **Model Selection:**

Select the most appropriate machine learning algorithm or model which will move your problem towards satisfactory or desired solution . The selection of models depend on the features of data , complexity of problem and characteristic of output. The commonly used algorithms are neural network , decision tree, support vector machines ,etc.

4. **Model Training:**

Model is trained by using the labelled trained data. During this process, parameters of algorithm are adjusted ,in such way that ,it should map input data point with corresponding output data point by minimizing predictive error rate and efficiently increasing accuracy rate.

5. **Model Evaluation:**

Here , the trained model's performance are evaluate on new dataset which are not used during its training process known as Test set or Validation set . The following metrics are using for model evaluation such as accuracy rate , error rate, precision rate , recall , mean squared error (MSE), mean absolute error (MAE), or R-squared (coefficient of determination) etc .

6. **Model Tuning:**

This process is applied when model need improvement in order to meet the satisfy able performance of desired output. Therefore to make it more correct and efficient ,following changes are done by adjusting various model parameter and preventing from condition of over - fitting and under – fitting.

7. **Model Testing:**

To assess the model performance in real world, the trained model and tuned hyper parameters is evaluated in test set.

8. **Model Deployment :**

If model meet the satisfy able performance of desired output, then it can be deployed to make prediction or decision on new ,real world data.

9. **Monitoring and Maintenance :**

Monitoring and maintenance is the essential need of every deploy model, in order ensure accuracy and be update by time, hence this should be done on regular basis.

1.1.1 Types of Supervised Learning

1.1.1.1 Classification: Model labeled a given input data points based on features into separate class or category. Based on number of classes and nature of the output features, it is further classified into the following which are: -

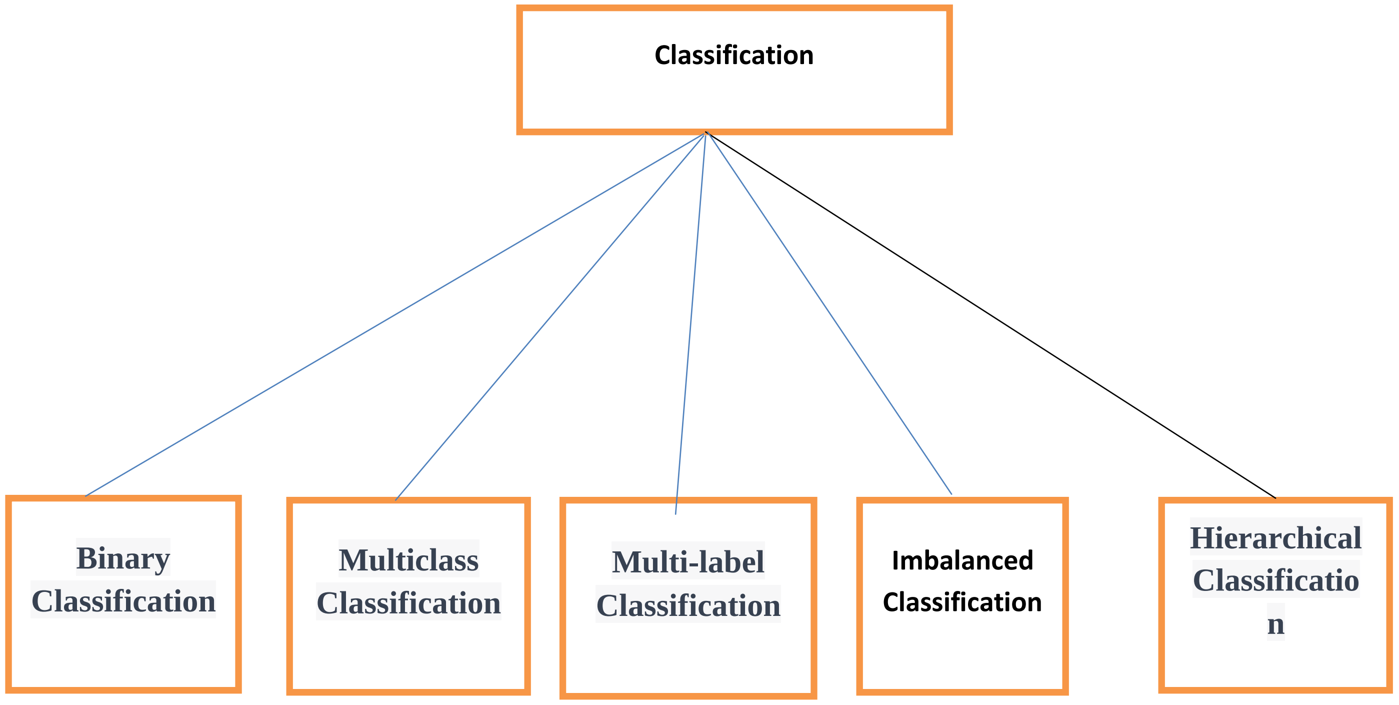
Classification
Binary Classification
Multiclass Classification
Multi-label Classification
Imbalanced Classification
Hierarchical Classificatio n

Figure 2: Types of Classification

Types of Classification

A. **Binary Classification:** In this type ,a given input data point based on features are put either in "0" and "1" class or "Negative" and " Positive" class .This classification is mainly used where there is only two classes of prediction or decision in output data points.

   For example:

   1.Credit card fraud detection ("0" fraudulent transaction or "1" legitimate transaction).

   2.Email spam detection ("0" spam or "1" not spam).

3.Disease diagnosis (“0” disease present or “1” not present).

B. **Multiclass Classification:** In this type ,a given input data point based on features are put in multiple classes or categories. This classification is mainly used where there is minimum three or more than three classes of prediction or decision in output data points.

For example:

1.Handwritten alphabet recognition (recognizing alphabet from 1 to 26).

2.Image classification of discrete object (e.g. cow, goat , bottle, truck ,etc).

3.Color identification (detecting the shades of a given picture among multiple possibilities).

C. **Multi-label Classification:** In this type ,a given input data point based on features are put in a single class or category with assigning multiple label or tag . This classification is mainly used where there is only one class of multiple prediction or decision in output data point.

For example:

1.File categorization (assigning multiple tags to a file based on its content).

2.Image classification (tagging image with multiple descriptive labels based on shapes).

3.Metals tagging (tagging metals with multiple descriptive labels based on its element).

D. **Imbalanced Classification:**

In this type ,a given input data point based on features are not lying in average range of class or category . This classification is mainly used where output data points falls beyond the boundaries of a class or category. Data point which belongs to this classification are counting in “outliers”

For example:

In a class of 100 students where 98 students score between 40 to 90 marks out of 100 marks and rest 2 students in that ,one student score 5 marks and another score 95 marks. Hence score of 5 marks and 95 marks are outliers .

E. **Hierarchical Classification:**

In this type ,a given input data point based on features are put in tree – like structure or hierarchy, ie. Most weighted features are put from top to down in tree. This classification is mainly used where data points large and complex .

For example:

Hierarchical classification for classifying different spices of birds.

Applications of classification using supervised learning. They are:

Fraud Detection, Email Spam Detection, Sentiment Analysis, Image Classification, Handwriting Recognition, Medical Diagnosis , Language Identification, Facial Expression Recognition, etc.

1.1.1.2 Regression:

It is type of machine learning task , where it focus to find predicted output in continuous numerical values. Here the model is trained on labeled data set in such a way that ,it find the relationship between input data point based on features to be associated with corresponding continuous numerical values of target or output data point based on features , enabling it to make prediction on new, unseen data set.

Types of regression techniques, They are :

A. **Linear Regression:**

This model goal is to find the best-fitted line (or hyperplane in higher dimensions) , which represent the relationship between the input features and the target variable as a linear equation . This linear equation help in minimizes the distance between the predicted values and the actual target values .

For this the linear equation is..

$Y= \beta 0 +\beta 1x1 +\beta 2x2 +...+ \beta nxn$

Where:

Y = Target variable (Dependent variable).

X = Target variable (Independent variable).

$\beta 0$, $\beta 1$, $\beta 2$,……. βn= Coefficients of the linear terms that represent intercept and slope of the line.

B. Polynomial Regression:

This regression is useful when data are in non linear pattern. It model the relationship between the input features and the target variable as a polynomial equation.

For this the polynomial equation is:

$Y= \beta 0 + \beta 1 x + \beta 2x2 + \beta 3x3 +……..+\beta ixi+\infty$

Where:

Y = Target variable (Dependent variable).

X = Target variable (Independent variable).

$\beta 0$, $\beta 1$, $\beta 2$,……. βi ,= Coefficients of the polynomial terms

∞ = Error term representing the difference between the actual and predicted values.

C. Ridge Regression (L2 Regularization):

It is also known as Tikhonov regularization, its objective is to find the optimal coefficient values for each features and penalizes the cost function of model for having large coefficient values. Besides this it helps to prevent the model from over- fitting situations . All this is possible by minimizing the combination of the Mean Squared Error (MSE) between the predicted values and the actual target values . And simply it is linear regression with penalty in terms of cost function.

The Ridge Regression cost function can be expressed as:

Cost Function = MSE (Mean Squared Error) + α * Σ(coefficient^2)

Where:

MSE = The difference between the predicted values and the actual target values.

α = The hyper-parameter that controls the strength of the regularization

∑(coefficient^2) = The hyper-parameter that controls the strength of the regularization

D. Lasso Regression (L1 Regularization):

It is same as Ridge Regression but here different penalty method is used. It aims to prevent overfitting and stabilize the model. It is efficient in dealing with high-dimensional data enough if it contains many irrelevant or less important features.

Lasso Regression cost function can be expressed as:

Cost Function = MSE (Mean Squared Error) + α * Σ(|coefficient|)

Where:

MSE = The difference between the predicted values and the actual target values.

α = The hyper-parameter that controls the strength of the regularization.

Σ(|coefficient|) = It calculates the sum of the absolute values of all the model coefficients.

E. Elastic Net Regression:

It is based on linear regression technique that combines L1 (Lasso) and L2 (Ridge) regularization terms. Its objective is to find the optimal coefficient values even when dealing with high-dimensional data and datasets with highly correlated features.

Elastic Net Regression cost function can be expressed as:

Cost Function = MSE (Mean Squared Error) + α * (ρ * Σ(|coefficient|) + (1 - ρ) * Σ(coefficient^2))

Where:

MSE = The difference between the predicted values and the actual target values.

α = The hyper-parameter that controls the strength of the regularization.

ρ = Mix parameter that balances the contribution of L1 and L2 regularization terms.

F. Support Vector Regression (SVR):

Its objective is to find the hyperplane (linear or non-linear) that best fit for the training data by allowing a certain range of error (epsilon) for the target values. The hyperplane has two parallel lines, known as the "ε-tube," whose width is (2 * epsilon).

Support Vector Regression (SVR) is a supervised machine learning algorithm for regression tasks .It is an extension of Support Vector Machines **(SVM)** designed to predict normal numbers and is effective for tasks such as performance estimation**,** time estimation**,** and more. The goal of SVR is to find the general plane that best fits the data while maximizing the margin somehow.

A description of how Support Vector Regression works:

Data Preparation:
A dataset is prepared using input data (attributes) and corresponding results (results to be estimated).

Kernel options:
Select functions (linear, polynomial, radial basis functions, etc.) maps the data to a higher-
order field that can capture more relationships.

Feature Scale:
Scale the input features to make sure they are in the same range. This is usually done using normalization or normalization.
Theformulation as an optimization problem: TheSVR aims to find a hyperplane (or a hyperplane-
like surface in high-
dimensional space) that fits the data points in one of many things. The goal is to minimize the error by all

owing some exceptions to the predicted values.

Unemployment:

SVR uses unemployment to account for the difference and margin between forecast and actual. Edges can be changed using a hyperparameter called the "C" parameter.

Solution:

solves the optimization problem to find the support vectors and indirect time coefficients that describe the plane mass (or surface). Support vectors are the data points closest to the edges.

Prediction:

To make predictions for new data, the model uses support vectors and their coefficients to calculate predi cted values.

Hyper-parameter Tuning:

Tune hyperparameters such as kernel type, constant constant (C), and kernel-specific parameters to improve the performance of the model.

Measurement:

Evaluate the performance of the model using an appropriate regression metric such as mean square error (MSE), root mean square error (RMSE), or R-squared.

Visualization (optional):

For 2D or 3D data, you can visualize SVR hyperplanes and support vectors to understand the behavior of your model.

Support Vector Regression is particularly useful when dealing with nonlinear relationships and complex d ata structures. It's worth noting that while the SVR is working well, careful consideration and consideratio n of key tasks is required to achieve the best results.

Libraries like scikit-learn provide implementations of support vector regression that you can use in your projects.

Advantages of Support Vector Regression:

1. SVR works efficiently even when the target variable exhibits non-linear behavior and enables to capture complex and non-linear relationships in the data.

2. SVR can handle high-dimensional data through the use of the kernel trick

Limitations of Support Vector Regression:

1. It is computationally expensive to train SVR on large dataset.
2. Choosing appropriate hyper-parameters , by using techniques like cross-validation can be challenging.

G. Decision Tree Regression:

It is a non-linear regression technique that uses a tree-like model to make predictions on input features and continuous numerical target values. The decision tree are formed by iteratively splitting the data into subsets based on feature values, so that minimizes the variance of the target values within each subset. Each internal node in tree denotes a decision based on a feature, and each leaf node denote the target predicted value. In order to make more accurate and stable predictions ,techniques like Random Forest and Gradient Boosting are often used .

Decision tree regression is a machine learning algorithm for regression functions. It is a supervised learning algorithm that constantly predicts results. Decision tree regression creates a tree model that iteratively divides a dataset into subsets and makes predictions based on the mean or weighted average of the target values in each subset. Here is the summary of Decision Tree Regression algorithm:

- Data Preparation:

Prepare data using inputs (features) and similar results (probability to predict).

- Tree Reference:

Decision tree regression begins with selecting the best features (attributes) to classify data according to a set of criteria aimed at minimizing the variance or similarity of squared error (i.e. mean squared error or MSE).

- Segmentation:

Selective features are used to subset data according to a certain threshold value. The division operation aims to create as homogeneous subsets as possible according to the target value.

- Recursive split:

Then iteratively repeat the split for all sites created in the previous step. This results in a tree structure where nodes represent individual elements and branches represent events.

- Stop Patterns:

Recursive splitting will continue until the stop is complete.
This measure can be the maximum height of the tree, the minimum number of samples in the number of leaves, or the variance/error reduction threshold.

- Leaf node prediction:

After a tree is built, the leaf nodes (terminal nodes) contain a subset of the data points. The estimate for each leaf is the mean (or weighted average) of the target values in that problem.

- Prediction:

Walk the tree evaluating the state at each node to predict for new data. When a page is reached, the estimate is the value associated with that page.

- Rating:

Evaluate the performance of a decision tree regression model using an appropriate regression metric such as mean square error (MSE), root mean square error (RMSE), or R - squared.

- Pruning (Optional):

Pruning will result in the removal of some branches from the tree to prevent overfitting. This can be done by considering the impact of each branch on the predicted performance of the model.

Decision tree regression has several advantages, including its interpretability, ability to manage nonlinear relationships, and robustness against outliers. But it can be very effective if the tree is very deep or the data is unpopular.

To implement decision tree regression, you can use libraries such as scikit-learn (Python), which provide simple classes and functions for building, training, and evaluating the decision tree.

Advantages of Decision Tree Regression

1. It can easily extract insights from decision tree.
2. It can handle large datasets and scale efficiently.
3. It effectively work on both numerical and categorical features.

Limitations of Decision Tree Regression:

1. A minor change in data can be lead to major change in prediction.
2. It is necessary to use Pruning and regularization techniques to control overfitting.
3. Its performance degrade when it captured subtle and complex relationships in the data.

H. Random Forest Regression:

It is formed by combining multiple decision tree regressions to improve predictive accuracy and reduce the challenges of overfitting. It is suitable in a variety of applications, including finance, healthcare, environmental sciences, etc. To achieve optimal performance on the specific dataset, tuning hyper-parameters are adjusted regularly , such as the number of trees, the maximum depth of the trees, and the number of features considered at each split.

Advantages of Random Forest Regression:

1. It can handle a large data set without the need for extensive feature selection or feature engineering.
2. It is robust in outliers and noisy data.
3. It reduces the risk of overfitting conditions.

Limitations of Random Forest Regression:

1. It is computationally expensive to trained a large number of decision tree.
2. It cannot always capture subtle patterns in the data .

3. In this regression it is quite challenging to trace back the decision-making process from various prediction.

1.2 Reinforcement Learning:

Reinforcement learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with the environment to achieve the best possible outcome. It was inspired by the way humans and animals learn through trial and error. In incentive learning, the agent learns how to behave in various situations to maximize reward over time and penalize them for their error.

Reinforcement learning is one of many methods developers use to train learning machines. The importance of this method lies in its ability to allow an agent (whether it is a work in a movie or a robot in a production) to learn how to act in the difficult environment it creates. The offender learns from his environment and improves his behavior over time, often through feedback that includes rewards and punishments.

Basics keys of RL work. They are:

1. **Agent /Caregiver**: A student or decision maker who interacts and works with the environment.

2. **Environment:** The external processes with which the agent interacts. Provides feedback to the agent in the form of rewards or punishments based on their behavior.

3. **State/ Status**: The current state or context the agent is in. The actions and decisions of the agent depend on the situation in which he lives.

4. **Action:** The choices an agent can make in a situation. Possible methods depend on the environment.

5. **Reward:** A scalar value given to the agent by the environment after each action. Indicates the direct benefits or costs of working in a state.

6. **Penalty/ Punishment**: The opposite of the reward.

The agent's goal is to learn a policy that is a map from situations to action and leads to the best results over time. The agent explores different actions in different situations and learns which ones are more profitable. There are many techniques to achieve this, which can be divided into two groups:

Model-based methods: These include creating models of environmental variables, keeping change in the state's environment, and how the reward is received. The agent then uses the model to plan and make decisions.

Model-free methods: As explained in the previous answer, this model focuses on learning directly by interacting with the environment.

1.2.1 Types of Reinforcement learning .They are as follows:

1. **Model-Free Methods**

   Model-free models are a class of techniques used in reinforcement learning (RL), where an agent learns to make decisions by interacting directly with the environment without an environmental energy design model. In other words, the freelance model seeks to learn performance based on the agent's experience, rather than trying to build a model of the environment's behavior.

   The Monte Carlo methods and Temporal Difference (TD) methods are the two main types of model-free methods in reinforcement learning:

   . Let's explore each of these:

   i) The Monte Carlo methods:

- The Monte Carlo method estimates the value of a state or state function by averaging its observed values (multiple rewards).
- They do not require strong environmental knowledge and are often used in situational tasks (working with different situations or different interactions).
- One of the most common Monte Carlo methods is the Monte Carlo Law Assessment, which estimates the value of a given situation by averaging repeated observations of different events.
- Another method is the Monte Carlo control, which develops the policy by estimating the values of the actions and choosing the desired action based on these results.

The Monte Carlo method is a class of computational methods used to estimate complex numbers or simulate random sampling. They are widely used in many fields, including physics, statistics, finance, and machine learning. Here is a general algorithmic summary of the Monte Carlo method:

Define the problem:

Clearly define the problem to be solved or the amount to be estimated using the Monte Carlo method.

Sources and Inputs:
Show the input, parameters, and insights needed for the problem. Decide on a name to use as an example.

Random Sampling:
Generate a random sample from the specified list. Randomness ensures that the sample represents the entire field.

Evaluate a Function or Model:
For each model, rate the function, model, or process you are working on. This may include solving equations, simulating physical systems, or using measurement models.

Results collected:
Results obtained (Operation evaluation, Results, etc.)
) from a random sample. These results will be used to estimate the required amount.

Approx :
Estimate the interest rate using the cumulative effect. This may include calculating means, compounds, probabilities, or other statistical measures.

Uncertainty Analysis:
Monte Carlo methods essentially provide an estimate that includes a measure of uncertainty, such as a confidence interval or standard error, of how the estimate will turn out.

Refinement and Convergence:
Increase sample size to refine estimates and reduce uncertainty. Convergence appears to be a stable approach for many models.

Analysis and Interpretation:
Estimate quantities and uncertainties in the context of analytical problems. Interpret the results and draw conclusions.

Visualization (optional):
Visualize results using histograms, graphs, or other visualization tools to understand the forecast.

Iteration and Refinement (Optional):

Depending on the complexity of the problem and the quality of the estimate, you can iterate, refine, or adjust the Monte Carlo method to improve accuracy or efficiency.

Monte Carlo methods can be used in many ways, including integration (Monte Carlo integration), optimization (simulated annealing), and simulation (Monte Carlo simulation). The main idea behind all these methods is to use statistical models to predict solutions that may be difficult to describe or measure.

Note that while the algorithm overview is general, the details of implementation and techniques may vary depending on the problem being solved and where the Monte Carlo method is used.

Let's use a simple example of using the Monte Carlo method in machine learning to estimate the value of π (pi) using a technique called Monte Carlo π estimation.

Example: Estimating π Using Monte Carlo

Problem:

We want to estimate the value of π using random sampling and the product of the inscribed square circle.

Setup:

Consider a square with a side length of 2 and a center (0, 0). A quarter of a circle of radius 1 is located at the origin of a square.

Random Sampling:
Generate random points (x, y) in a square.

ii). Temporal Difference (TD) methods :

- The physical difference method combines the ideas of the efficient function and the Monte Carlo method.
- TD's revised cost estimates based on the predicted state of future outcomes make them eligible for online learning in non-participating programs.
- The most famous TD method is the Q-learning algorithm that directly estimates the best value and iterates the Q value using the Bellman equation.
- Another popular TD method is SARSA (Status-Action-Reward-State-Action), which changes the Q value while following a rule given by the Q value itself.

Both Monte Carlo and TD methods have their own advantages and disadvantages. Monte Carlo methods may differ in that they depend on all returns, while TD methods allow more learning and are generally faster tied. Model-free methods are often appropriate in real-world situations, when building appropriate models of complex environments or when focusing on learning from direct interactions.

1. **Policy Gradient Methods :**

   The gradient rule method is a class of reinforcement learning algorithms that optimizes an agent's rule to maximize its reward. Unlike Q-learning or other useful methods that focus on estimating the value of the value, the gradient rule method directly improves the rule itself. This technique is particularly useful when the workspace is continuous and the agent needs to learn stochastic rules.

Here's basics overview of how Policy Gradient Methods works:

i). Policy Representation: The policy of an agent is usually represented by a parametric function (such as a neural network) that takes the state as input and outputs the probability distribution. This allows the agent to learn a map from situations to actions.

ii). Objective Function: The agency's goal is to maximize profits. This is often done as an objective function, often referred to as the "objective" or "objective function", that evaluates how good or bad the policy is. Often this purpose is the need for gifts, also known as "gifts" or "gifts" at the time.

iii). Gradient Ascent : The gradient rule method uses gradient descent to change the limitless rule to increase the desired reward. The gradient of the objective function with rules is not calculated using techniques such as the REINFORCE algorithm (gradient rule) or more methods such as the Actor-Critic architecture.

iv). Monte Carlo Estimation :It uses algorithm sampling method to calculate gradients. The agent interacts with the environment, writes the system according to the current policy, and then uses this system to change the policy. Gradients are estimated using the probability of action in trajectories scaled with reward.

v). Update Rule: The updated rule for the rule does not include the evaluation of the gradient from the learning value (small degree) and is used in the rule network using techniques such as stochastic gradient descent (SGD) or multiple optimization algorithms.

vi). Exploration and Exploitation: The gradient rule method involves searching by nature, as the broker's rule is stochastic. This allows the agent to explore different behaviors and understand their impact on the environment.

There are many algorithms and variants within the framework of the gradient rule. Here are some key points:

A. **REINFORCE (Monte Carlo Policy Gradient):** REINFORCE, which stands for "Reinforcement = No Error × Reinforcement Program × Efficiency", is a classical gradient algorithm used for reinforcement learning to improve the operator's method. This is a Monte Carlo Gradient method; that is, it estimates gradients based on Monte Carlo sampling of orbits.

Here is a sequence of the REINFORCE algorithm:

i).Policy Parameterization: Represent a rule as a parameterized function that specifies a function. This can be a neural network, linear function, or some other approximation function.

ii). Write Trajectories: Interact with the environment using existing policy to create a sequence (series of states, actions, and rewards).
All methods start from initial state, follow rules to choose actions and collect rewards till the end.

iii). Calculate Reward: Calculate the discount (reward) from that moment for each time on each road. The reward is the sum of future rewards weighted by a discount factor that takes into account the reduction in future rewards.

iv). Calculation Policy Gradient: For each time in each path, calculate the gradient of policy probability with the logarithm of the policy parameter. This slope shows how much constraints need to be adjusted in order to have the possibility to choose the action of that situation.
v). Update Rules Automatically: Automatically update rules, including gradients and return values. Policy change usually involves equalizing the slope from the reward and adjusting the unlimited policy to increase the probability of higher reward.

Mathematically, the update rule for the rule parameter θ using the REINFORCE algorithm can be expressed as:
$\Delta\theta = \alpha * \nabla\theta \ln(\pi(as, \theta)) * G$

where:

α is the learning rate.
$\nabla\theta \ln(\pi(as, \theta))$ is the slope of the logarithm of the law with the parameter θ.
G is revenue.
Repeat: Keep typing sequences, counting down, calculating gradients, and editing unlimited rules. An important feature of

REINFORCE is that the trajectories have high variability in updates due to Monte Carlo sampling. This can be slow and unstable. Variable reduction techniques such as fundamental subtraction (reducing the estimated value of the reward) and normalizing the reward are often used to solve this problem.

B. Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO): These are policy gradient algorithms that address some of the limitations of the REINFORCE algorithm. Both A2C and PPO are used to enhance the agent's authority to support learning, but they include additional methods to quickly improve security, model performance, and connectivity.

i). Best Actor-Critic (A2C):

A2C is an algorithm that does well in terms of policy and value. It uses two main concepts:

- **Actor:** The actor is responsible for learning and developing policy. It interacts with the environment, populating the system and changing the law without using law gradients.

This is similar to the network rule in other gradient rule algorithms.

- **Critic (Value Network):** The critic is responsible for estimating the value of the function that estimates the expected reward by following the policy starting from the given situation. This will help gauge how good or bad the participants' performance is compared to the desired reward. The cost function is often used to calculate the "quality" of the order, that is, the difference between the observed reward and the estimated cost. The main advantage of

A2C is that it uses predictive value to reduce the variance of the gradient update law.
By removing the forecast from retrospective analysis, the update becomes more predictable and less likely to change.

ii). Proximal Policy Optimization (PPO):

PPO is a state-of-the-art policy optimization algorithm based on the Progressive Policy framework. Its purpose is to improve policy while ensuring that updates remain in a safe zone to prevent major changes that could lead to instability. This is done by imposing restrictions on the update policy using a backup target.

PPO works in two steps: sampling rows and updating rules.

The main concept of PPO is "imminent" updating, which limits changes in distribution policy. **There are two types of PPO**: PPO-Clip and PPO-Punishment. PPO-Clip cuts to determine the ratio between old and new rules, while PPO-Punishment adds a penalty to the target function to check the limit. The advantages of the

PPO include model performance and more stable adjustment compared to simple tilt methods such as REINFORCE. It also has parameters that govern the balance between policy development and regulatory constraints.
In summary, both A2C and PPO are high gradient policies compared to REINFORCE. A2C uses a cost function to reduce divergence of policy updates, while PPO imposes restrictions on security and control of policy updates. These algorithms have performed well in many supportive learning applications and are widely used in research and practice.

C. **Trust Region Policy Optimization (TRPO)** :Trust Zone Policy Optimization (TRPO) is one of the best policy optimization methods for further studies. He mentions some of the challenges of development policy, especially in terms of security and innovation management. TRPO aims to ensure that the updated policy remains "safe" to prevent major changes that could lead to instability. The main idea behind

TRPO is to improve the law by following the constraints that limit the KL difference (measurement difference) between the new law and the old law. This restriction ensures that policy changes are not too aggressive, preventing performance degradation or instability.
This is how TRPO works:

- **Write Trajectories:** Interact with the environment to write trajectories using the current strategy.

- **Calculate Ratio Estimates**: Estimate the probability function for each state-function pair. Dominance indicates how good or bad the action is compared to the average action in that situation. Dominance estimation can be done using quantitative estimation or more complex methods.

- **Alternative Calculation Target:** Calculate an alternative target aimed at maximizing profit while staying within some variant of KL.

This representative objective anticipates policy development while determining regional confidence.

- **Bad policy update:** solve the optimization problem to find the update policy whose purpose of the agent is to ensure that the KL difference between the old policy and the New policy is at a starting point.

TRPO Ensure policy reforms are effective to prevent radical changes that could lead to inadequate or unsustainable education. This helps sustain progress while minimizing the risk of major policy changes. The advantages of TRPO include better performance standards compared to other policy development methods, and a clear theoretical foundation that allows for policy updates.
However, TRPO can be computationally expensive as it solves the optimization problem at every iteration. It is worth noting that although TRPO works well, the optimization problem is difficult to solve correctly. This led to the development of a more practical strategy that took the ideas of the TRPO but had a more refined approach. One such algorithm is Proximal Policy Optimization (PPO), a successor to the widely used TRPO due to its ease of use, improved model performance and stability.

In summary, Trusted Zone Policy Optimization (TRPO) is a powerful policy optimization algorithm that focuses on managing trust zones to maintain security and maintain new policies, but its computation should lead to the development of many successor algorithms such as Proximal Policy. Optimization (PPO).

D. Soft Actor-Critic (SAC): Soft Actor-Critic (SAC) is an advanced academic support program that addresses the challenge of continually optimizing code at work. SAC is designed to handle stochastic rules and environments with high state and function settings. It combines the principles of the gradient method and Q-learning to provide stable and effective learning. The main features and definitions of the SAC include:

i). Entropy Regulation: One of the distinguishing features of the SAC is the inclusion of an entropy normalization term in the objective function. Entropy is a measure of uncertainty in an action rule.
SAC encourages the rule to be stochastic by adding a negentropy term to the reward, which simplifies the search and helps prevent premature convergence to the right decision.

ii). Actor-Critic architecture: Similar to other gradient algorithms, SAC uses Actor-Critic architecture. The participant is responsible for learning the rule, while the critic learns the value to predict the expected reward. There are two useful functions in SAC that help reduce uncertainty in price forecasting.

iii). Non-policy learning: SAC is a feedback algorithm; that is, it learns from past experiences that are stored iteratively.

This improves model performance as previous information can be reused for multiple updates.

iv). Decision rule update: SAC updates the decision rule when it supports a stochastic rule with constant entropy. This means that the policy update is based directly on the best decision based on current policy and cost performance.

v). Soft Bellman Equation: SAC uses a modified Bellman equation to calculate term entropy. This results in a "smooth" update rate that takes into account the expected reward and the entropy rule.
The SAC algorithm aims to achieve a combination of expected rewards and entropy. This encourages exploration while seeking higher rewards. The term entropy in the target avoids over determination of the law, which is particularly useful in complex and high-dimensional environments. The SAC performs in a variety of tasks, including robot control, robot handling, and continuous control issues. Ideal for areas where search is paramount and finding the right balance between search and use is difficult.

In summary, Soft Actor-Critic (SAC) is a knowledge-based support learning algorithm that combines regular entropy, decision-making policy updates, and learning exit policies to improve the good and stable in the permanent workplace. It supports search and randomization rules while aiming for high rewards.

3.Actor-Critic Methods :

The Actor-Critic method is a class of support learning algorithms that combines value-based and policy-based concepts. They belong to two directions: "actor" and "critic". These components work together to optimize the broker's policy when estimating the value of different states or pairs of states.

An example of how the Actor-Critic approach works:

Actor:

The actor is responsible for learning and developing policies. It is a policy function that uses the state as input and outputs the probability of action.

The aim of the participants is to learn the most profitable policy over time. A rule can be stochastic, i.e. it chooses actions based on probability.

Critic:

Critic is responsible for estimating the value of a state or state-function pair. It helps the agent understand the long-term needs of staying in a situation and making a plan of action.

The critic estimates the value, giving feedback to the participants and directing them to actions that will lead to better results.

The interaction between the performer and the critics is an important part of the performance improvement process. The price index allows actors to understand the need for different operations in different states. This allows stakeholders to modify their code to support functionality with higher requirements.

There are several variants of the Actor-Critic method, each with its own characteristics:

A. **.Deep Deterministic Policy Gradient (DDPG):** Deep Deterministic Policy Gradient (DDPG) is a reinforcement learning algorithm designed to solve continuous problems in space. It is an Actor-Critic algorithm that combines the true gradient method and Q-learning ideas. DDPG is particularly useful for activities where actions are not limited to discrete options but take place in multiple continuities such as robot control and autonomous driving.

DDPG works like this:

i). Actor Network (Policy):

Actor Network uses state to process entry and exit decision. The goal of players is to learn the rules that show the actions that give the best results.

ii). Critic Network (Value Function): The critical mesh estimates the value of a state pair.

iii). Experience Replay Buffer:

The DDPG uses the Broadcast Replay Buffer to store information (status, action, reward, subsequent states) collected during interaction with the environment. The replay buffer helps to disrupt the information environment and make learning more effective by reusing new information.

iv). Target Network:

DDPG uses a target network to secure learning. These are back-to-back communications that are on the same stage with the main actors and critics, but are rarely updated. The target network provides stable results for updates and good ideas for updated actors.

v). Noise Research:

To support research in the field of continuous operations, DDPG adds noise research to the actions of network actors. This noise helps the agent search different areas of the workplace.

vi). Drop and Update:

It uses the gradient update rule to update participants to increase the critic's estimated Q for the selection. The critic is updated using the squared Bellman error, which minimizes the difference between the estimated Q value and the target Q value obtained from the target network.

In summary, DDPG addresses specific issues for the permanent workplace by combining decision-making policy updates with Q-learning-like updates. It uses an irreplaceable experience and a purposeful network to improve the security and performance of education. The success ofDDPG has led to the development of several methods, such as Double Delayed Deep Deterministic Policy Shifts (TD3), which further enhance stability by reducing overestimation bias and introducing policy smoothing. DDPG and its variants are widely used in many fields, including robotics, self-driving cars, and financial markets.

B. **Twin Delayed Deep Deterministic Policy Gradient (TD3)** : Double Delayed Deep Deterministic Policy Gradient (TD3)is a reinforcement learning algorithm and an extension of the Deep Deterministic Policy Gradient (DDPG) algorithm. TD3 is reported to address some of the limitations of DDPG, in particular the limitations of Q overestimation and law instability. TD3 aims to improve the stability, model performance and integration of the DDPG algorithm.

TD3 is designed for DDPG as follows:

i). Dual Q Study with Binary Critics :

The DDPG may suffer from overestimation bias in the Q value estimation, resulting in poor tuning. To solve this problem, TD3 offers a network of Siamese critics. TD3 retains two independent functions for estimating the Q value. This helps solve the overestimation problem by choosing the smallest estimate of the two critics' Q-values as the target for price adjustment.

ii). Target Policy Smoothing:

TD3 uses target policy smoothing to stabilize training and prevent aggressive policy updates. This includes adding noise to the action plan established by the target network policy. The TD3 optimizes the target function, reducing the bias of Q value estimates and avoiding cost bias.

iii). Policy Update Delay:

TD3 shows the network policy update delay. TD3 updates the rule less frequently than updating the rule with each update of the critic. Delayed policy updates help prevent unwanted policy swings and improve learning stability.

iv). Clipped Double Q Target:

TD3 shows the clipped Q target for the update value. This helps prevent the Q value from being overestimated and ensures that the update value remains constant.

v). Noise Detection and Repeatability:

Similar to the DDPG, the TD3 uses noise detection in the player's movement to support detection in the continuous action area. The also uses recycling to store and reuse information for better learning outcomes. Developing the TD3 over DDPG makes operation more stable, reduces overestimation bias, and improves performance of continuous control challenge. Addressing the limitations of DDPG, the TD3 has become a popular choice for solving the continuous workspace problem in further education. It is worth noting that although theTD3 has improved stability and performance, hyper-parameter tuning is still very important to achieve the best results .Also, the TD3 approach using twin critical, objective right smoothing and delay updating has led to further research in this area and the development of other algorithms and their variants.

C. **Advantage Actor-Critic (A2C):** Refer point Bin **Policy Gradient Methods :**

D. **Soft Actor-Critic (SAC):** Refer point D in **Policy Gradient Methods :**

4. Model-Based Methods: Model-based methods in learning support (RL) involve creating clear models of the work environment to help employees learn to make decisions. This system uses learning models to simulate performance, predict and optimize results. Model-based reinforcement learning contrasts with model-less learning, where the agent learns the rules simply by interacting with the environment, without design.

Here is the explanation of the model in further education:

i). Model Building:

Workers create models for the changing environment. The model captures how the environment for an action changes between situations and how rewards are created.

Models can be used in many ways, such as transition probability matrices, state transitions, or neural networks that predict the next state and reward.

ii). Planning and Simulation:

Operators can simulate trajectories by selecting actions and predicting future states and rewards using learning models. Planning algorithms such as

Monte Carlo Tree Search (MCTS) or dynamic programming can be used to explore the possibilities and state space to optimize decision making.

iii). Optimization Rules:

Model-Based Optimization Rules optimize rules by incorporating predictions of learning models. The broker simulates various bids in different ways and chooses the action that produces the best value based on the model.

iv). Trade-offs and challenges:

The standards-based model has the advantage of being effective in planning and thinking about the environment. They often need to interact with the real environment to learn good rules.

However, the quality of the study sample is important. Correct operation would be best if the model cannot represent the true nature of the environment.

Also, planning in complex environments can be computationally expensive, especially when using detailed neural network models.

v). Model Learning:

Model-based methods require learning the model itself. This can be done using various techniques such as supervised learning, dynamic matching or incremental learning, where the agent gathers information from interactions with the environment.

vi). Combination of Model-Free Methods:

Many modern methods combine elements of model-based and model-independent methods. These combinations aim to balance the strengths of each method.

Model-based approaches can provide a good starting point for early stages of learning where the agent knows little about the environment.

As the agent collects more data, it can switch to using free models that learn from real interactions.

Model-based approaches are particularly useful where real-world environments are costly or risky (for example, in robotics or healthcare) and where a good understanding of the work environment is required. However, the quality of the study sample is still an important factor affecting the effectiveness of these models.