Chapter 6: Learning Systems

Rote Learning

- Rote learning in the context of artificial intelligence (AI) and machine learning refers to the process of memorizing information or patterns without understanding their underlying meaning or principles.
- In traditional education, rote learning involves memorizing facts, formulas, or procedures through repetition, without necessarily comprehending their deeper significance.
- In the realm of AI and machine learning, rote learning similarly involves the memorization of patterns or data without a deeper understanding of the concepts behind them.

Examples of Rote Learning in AI and Machine Learning:

1. **Pattern Recognition:** In supervised learning tasks such as image classification, a model may learn to recognize patterns in images without truly understanding the content of the images. For instance, in a dataset of handwritten digits, a machine learning algorithm may memorize the specific pixel arrangements associated with each digit without understanding the concept of numbers.
2. **Text Classification:** In natural language processing (NLP), rote learning can occur in text classification tasks where a model memorizes certain keywords or phrases associated with particular categories without understanding the semantic meaning behind them. For example, a sentiment analysis model might learn to associate positive sentiment with words like "good" or "great" without truly grasping the nuances of sentiment.
3. **Language Translation:** Machine translation systems may sometimes rely on rote learning, especially in statistical or rule-based approaches. These systems may memorize translations of specific phrases or sentences without truly understanding the grammar or semantics of the languages involved.

Applications of Rote Learning in Machine Learning:

1. **Speech Recognition:** In speech recognition systems, rote learning can be employed to recognize specific words or phrases based on their acoustic patterns. While more advanced systems incorporate deep learning techniques for better understanding of language, simpler systems may rely on rote memorization of speech patterns.
2. **Anomaly Detection:** In anomaly detection tasks, where the goal is to identify unusual patterns or outliers in data, rote learning can be useful. By memorizing normal patterns in the data, a machine learning algorithm can detect deviations from these patterns as anomalies.
3. **Recommendation Systems:** Rote learning can play a role in recommendation systems by memorizing user preferences based on past interactions. While more

sophisticated recommendation systems use collaborative filtering or deep learning techniques, simpler systems may rely on memorizing user-item interactions.

While rote learning can be effective in certain contexts, it also has limitations. Models that rely too heavily on rote learning may struggle to generalize to unseen data or adapt to changing environments. Therefore, it's important for machine learning practitioners to strike a balance between memorization and understanding when developing AI systems.

Learning from examples: Inductive learning Methods

- Inductive Learning Algorithm (ILA) is an iterative and inductive machine learning algorithm that is used for generating a set of classification rules, which produces rules of the form "IF-THEN", for a set of examples, producing rules at each iteration and appending to the set of rules.
- There are basically two methods for knowledge extraction firstly from domain experts and then with machine learning. For a very large amount of data, the domain experts are not very useful and reliable. So we move towards the machine learning approach for this work. To use machine learning One method is to replicate the expert's logic in the form of algorithms but this work is very tedious, time taking, and expensive. So we move towards the inductive algorithms which generate the strategy for performing a task and need not instruct separately at each step.

Why should you use Inductive Learning?

The ILA is a new algorithm that was needed even when other reinforcement learnings like ID3 and AQ were available.

- The need was due to the pitfalls which were present in the previous algorithms, one of the major pitfalls was the lack of generalization of rules.
- The ID3 and AQ used the decision tree production method which was too specific which was difficult to analyze and very slow to perform for basic short classification problems.
- The decision tree-based algorithm was unable to work for a new problem if some attributes are missing.
- The ILA uses the method of production of a general set of rules instead of decision trees, which overcomes the above problems

Basic Requirements to Apply Inductive Learning Algorithm

1. List the examples in the form of a table 'T' where each row corresponds to an example and each column contains an attribute value.

2. Create a set of m training examples, each example composed of k attributes and a class attribute with n possible decisions.
3. Create a rule set, R, having the initial value false.
4. Initially, all rows in the table are unmarked.

Necessary Steps for Implementation

- Step 1: divide the table 'T' containing m examples into n sub-tables (t1, t2,…..tn). One table for each possible value of the class attribute. (repeat steps 2-8 for each sub-table)
- Step 2: Initialize the attribute combination count ' j ' = 1.
- Step 3: For the sub-table on which work is going on, divide the attribute list into distinct combinations, each combination with 'j ' distinct attributes.
- Step 4: For each combination of attributes, count the number of occurrences of attribute values that appear under the same combination of attributes in unmarked rows of the sub-table under consideration, and at the same time, not appears under the same combination of attributes of other sub-tables. Call the first combination with the maximum number of occurrences the max-combination ' MAX'.
- Step 5: If 'MAX' == null, increase ' j ' by 1 and go to Step 3.
- Step 6: Mark all rows of the sub-table where working, in which the values of 'MAX' appear, as classified.
- Step 7: Add a rule (IF attribute = "XYZ" –> THEN decision is YES/ NO) to R whose left-hand side will have attribute names of the 'MAX' with their values separated by AND, and its right-hand side contains the decision attribute value associated with the sub-table.
- Step 8: If all rows are marked as classified, then move on to process another sub-table and go to Step 2. Else, go to Step 4. If no sub-tables are available, exit with the set of rules obtained till then.

An example showing the use of ILA suppose an example set having attributes Place type, weather, location, decision, and seven examples, our task is to generate a set of rules that under what condition is the decision.

Example no.	Place type	weather	location	decision
1.	hilly	winter	kullu	Yes
2.	mountain	windy	Mumbai	No
3.	mountain	windy	Shimla	Yes
4.	beach	windy	Mumbai	No
5.	beach	warm	goa	Yes
6.	beach	windy	goa	No
7.	beach	warm	Shimla	Yes

Subset – 1

s.no	place type	weather	location	decision
1.	hilly	winter	kullu	Yes
2.	mountain	windy	Shimla	Yes
3.	beach	warm	goa	Yes
4.	beach	warm	Shimla	Yes

Subset – 2

s.no	place type	weather	location	decision
5.	mountain	windy	Mumbai	No
6.	beach	windy	Mumbai	No

7.	beach	windy	goa	No

- At iteration 1 rows 3 & 4 column weather is selected and rows 3 & 4 are marked. the rule is added to R IF the weather is warm then a decision is yes.
- At iteration 2 row 1 column place type is selected and row 1 is marked. the rule is added to R IF the place type is hilly then the decision is yes.
- At iteration 3 row 2 column location is selected and row 2 is marked. the rule is added to R IF the location is Shimla then the decision is yes.
- At iteration 4 row 5&6 column location is selected and row 5&6 are marked. the rule is added to R IF the location is Mumbai then a decision is no.
- At iteration 5 row 7 column place type & the weather is selected and row 7 is marked. the rule is added to R IF the place type is beach AND the weather is windy then the decision is no.

Finally, we get the rule set:- Rule Set

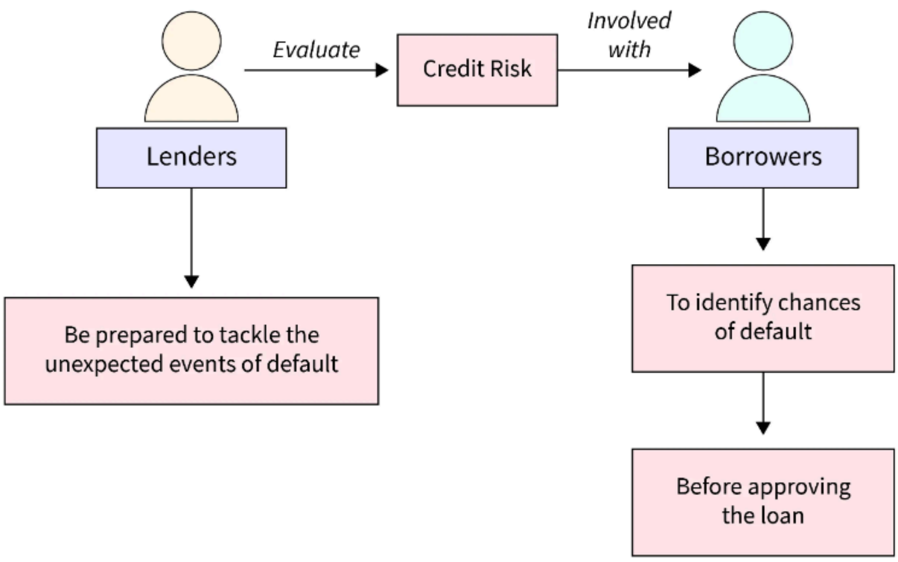
- Rule 1: IF the weather is warm THEN the decision is yes.
- Rule 2: IF the place type is hilly THEN the decision is yes.
- Rule 3: IF the location is Shimla THEN the decision is yes.
- Rule 4: IF the location is Mumbai THEN the decision is no.
- Rule 5: IF the place type is beach AND the weather is windy THEN the decision is no.

The Fundamental Concept of Inductive Learning

Inductive learning algorithms are used in a variety of applications, including credit risk assessment, disease diagnosis, face recognition, and autonomous driving.

Credit Risk Assessment

- One of the most significant inductive learning applications in AI is credit risk assessment. The goal of credit risk assessment is to predict the probability that a borrower will be incapable of repaying a loan (or other kind of credit). By looking at a borrower's financial records and other important factors, lenders can use credit risk assessment to make knowledgeable choices about whether to offer credit and on what conditions.
- Inductive learning algorithms can be used to analyze a wide range of financial data to predict credit risk. Large databases of past borrower data can be used to train these algorithms so they can discover trends and pinpoint the variables that most accurately predict credit risk.

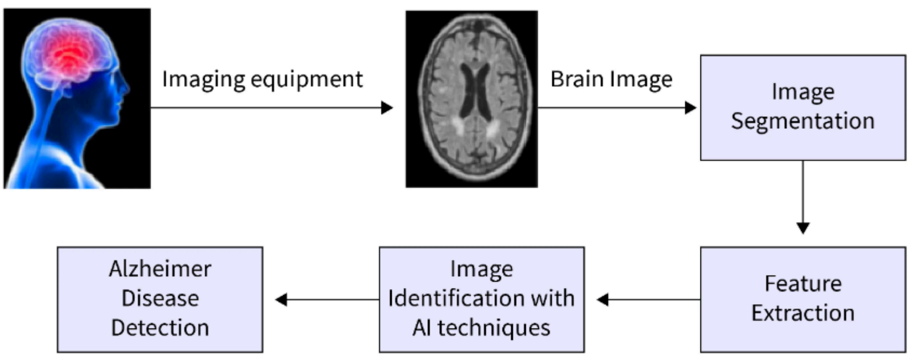


The potential for lenders to consider various criteria when making decisions regarding credit is one of the main advantages of inductive learning for credit risk assessment. Conventional credit risk assessment methods frequently rely on a small collection of unreliable or biased variables, like income and credit score. On the other hand, inductive learning algorithms can examine a considerably wider range of elements, including social media activity, debt-to-income ratios, and employment histories, to produce more precise and sophisticated credit risk evaluations.

Inductive learning algorithms for credit risk assessment may also be more flexible and adaptable in addition to being more accurate. They can be updated with fresh information as it becomes available, enabling lenders to continuously improve their evaluations of credit risk based on the most recent data. This is especially critical given the current economy's rapid economic change and emerging new financial products and services.

Disease Diagnosis

- Disease diagnosis is another important application of inductive learning in AI. With the help of inductive learning algorithms, doctors and other medical professionals can use data to make more accurate and timely diagnoses of various diseases.
- One of the key benefits of inductive learning in disease diagnosis is the ability to analyze large datasets of medical information, including patient histories, lab results, and imaging data. Inductive learning algorithms can find patterns and trends by examining these big datasets that could be challenging for medical professionals to spot on their own.
- For example, huge collections of medical pictures, such as X-rays or MRIs, can be analyzed using inductive learning algorithms to assist in the diagnosis of illnesses like cancer and heart disease. Inductive learning algorithms can learn to recognize patterns and anomalies that may be challenging for human doctors to spot. In the long run, patients would gain since earlier and more accurate diagnoses might arise from this.



We can use inductive learning to examine patient data, including medical histories, symptoms, and lab findings to diagnose various diseases. Inductive learning algorithms can learn to recognize patterns and trends that are suggestive of particular diseases or disorders by examining vast datasets of patient data. This can aid medical professionals in developing more precise diagnoses and efficient treatment methods.

Face Recognition

- Face recognition is a widely used application of inductive learning in AI. Face recognition seeks to recognize people based on their facial features, which can be helpful for various applications, from social media and marketing to security and surveillance.
- We can train inductive learning algorithms on large facial picture data datasets to discover patterns and pinpoint distinctive traits for every person. Inductive learning algorithms can properly identify people with a high degree of accuracy by analyzing face traits such as the distance between the eyes, the curve of the nose, and the curvature of the lips.

One of the key benefits of inductive learning for face recognition is that it can adapt to new data and changing conditions. For example, in a security or surveillance context, inductive learning algorithms can learn to recognize individuals under various lighting and environmental conditions, including different angles, lighting, and facial expressions. This makes them more reliable and effective than traditional face recognition systems, which may struggle to accurately identify individuals under changing conditions.

We can also use inductive learning algorithms to improve the accuracy of facial recognition systems over time. By continually training on new data and incorporating new features into the recognition process, inductive learning algorithms can improve their accuracy and reduce the likelihood of false positives and false negatives.

Automatic Steering (Autonomous Driving)

- Automatic steering, or autonomous driving, is another important application of inductive learning in AI. Autonomous driving aims to develop self-driving vehicles that can operate safely and efficiently on public roads and highways without human intervention.
- We can use inductive learning algorithms in various ways to support autonomous driving. For example, we can use them to analyze large datasets of sensor data, such as Lidar (Light Detection and Ranging), Radar, and camera data, to identify and track objects on the road, including other vehicles, pedestrians, and obstacles. By learning to recognize and respond to these objects, inductive learning algorithms can help self-driving vehicles operate safely and avoid accidents.

- Another important application of inductive learning in autonomous driving is route planning and optimization. By analyzing traffic patterns, road conditions, and other factors, inductive learning algorithms can help self-driving vehicles plan the most efficient and safe route to their destination while avoiding congestion, accidents, and other hazards.
- Inductive learning algorithms can also be used to improve the performance and reliability of self-driving vehicles over time. By continually analyzing data on driving behavior and road conditions, inductive learning algorithms can identify patterns and trends that can be used to improve the performance and safety of self-driving vehicles.

Applications

Inductive learning algorithms are used in a variety of applications, including:

Problems in which no human expertise is available

Inductive learning algorithms are beneficial for solving problems in which no human expertise is available. This is because these algorithms can learn from large datasets and discover patterns and relationships that may take time to become evident to humans. Some examples include:

1. **Bioinformatics:** In the field of bioinformatics, we can use inductive learning algorithms to analyze large datasets of genetic and molecular data to identify relationships between genes and diseases. This is a complex problem that requires a deep understanding of genetics, but we can train inductive learning algorithms to recognize patterns and relationships that may not be immediately apparent to human researchers. For example, consider a scenario where researchers aim to understand the genetic factors contributing to a specific disease, such as cancer. With inductive learning algorithms, they can input genetic data from thousands of patients, including gene expressions, mutations, and clinical outcomes. The algorithms can then analyze this data, identify common patterns, and discover genetic features associated with the disease. This knowledge can further aid in predicting disease risk, optimizing treatment strategies, and even identifying potential therapeutic targets.

2. **Speech Recognition:** Another example of a problem in which no human expertise is available is speech recognition. While humans are naturally skilled at recognizing speech and understanding language, the underlying processes that enable this ability are complex and difficult to understand. Inductive learning algorithms can be used to analyze large datasets of speech and language data to identify patterns and relationships that can be used to improve the accuracy and reliability of speech recognition systems.
3. **Finance and Economics:** We can also use inductive learning algorithms to analyze complex data sets in fields such as finance and economics. For example, we can use these algorithms to analyze stock market data to identify trends and patterns that can be used to make better investment decisions. In certain scenarios, the complexity of the data may surpass human capacity to make precise predictions. Still, we can train inductive learning algorithms to recognize patterns and relationships that we can use to make more accurate predictions.

Humans can complete the task, but no one knows how to do it

Inductive learning algorithms are also helpful in solving problems in which humans can complete the task, but no one knows how to do it. In these cases, the task is often too complex for humans to explain clearly and concisely, but humans can still complete the task successfully through trial and error. Some examples include:

1. **Robotics:** For example, in the field of robotics, we can use inductive learning algorithms to teach robots how to perform complex tasks, such as grasping and manipulating objects, based on trial and error. Humans can complete these tasks naturally, but it is difficult to explain the precise sequence of required movements and actions. By observing humans performing the task and analyzing the data, inductive learning algorithms can learn how to perform it themselves.
2. **Games:** Another example of a problem in which humans can complete a task, but no one knows how to do it is game playing. In games such as Chess and Go, human experts can play at a very high level, but it is difficult to explain exactly how they can do so. Again, we can use inductive learning algorithms to analyze large datasets of

gameplay and identify patterns and strategies that we can use to play the game at a high level.

3. **Nuclear Power Plants:** We can also use inductive learning algorithms to solve problems in which humans are able to complete the task through trial and error, but the task is too dangerous or expensive for humans to perform. For example, in the field of nuclear power plant maintenance, inductive learning algorithms can be used to teach robots how to perform complex maintenance tasks in hazardous environments based on trial and error.

Problems where the desired function is frequently changing

Inductive learning algorithms are particularly well-suited for solving problems where the desired function changes frequently. In these situations, traditional machine learning algorithms may not be effective because they require a fixed training dataset and cannot adapt to changing conditions. Some examples include:

1. **Fraud Detection:** For example, in the field of fraud detection, the task is to identify fraudulent transactions. However, fraudsters are constantly changing their techniques, which means that traditional machine-learning algorithms may be unable to keep up. Inductive learning algorithms can adapt to changing conditions by continually updating their models based on new data, which can help them identify new fraud patterns as they emerge.
2. **Chatbots**: In Natural Language Processing (NLP), we can use inductive learning algorithms to develop chatbots and other conversational AI systems that can adapt to the language and user behavior changes. As users interact with the system and provide feedback, the algorithm can learn from this feedback and update its model to understand the user's intent better and respond more accurately.
3. **E-commerce:** Another example of a problem where the desired function is frequently changing is e-commerce. Online retailers are constantly changing their pricing and promotion strategies, which means that algorithms used to optimize pricing and marketing campaigns must adapt quickly to changing conditions. We can use inductive learning algorithms to analyze real-time data on user behavior and sales

performance, identify new patterns and trends, and update pricing and marketing algorithms accordingly.

Problems where each user requires a unique function

Problems where each user requires a unique function are known as personalized problems. These problems are particularly challenging because the desired function for each user is different, which means that traditional machine-learning algorithms may not be effective.

Inductive learning algorithms, however, are well-suited for solving personalized problems. These algorithms can analyze user behavior and feedback to develop a unique model for each user, which can help to improve the accuracy of predictions and recommendations.

1. **Healthcare:** One example of a personalized problem is in the healthcare field. Each patient has unique medical history, symptoms, and lifestyle factors, which means that the optimal treatment plan for each patient may differ. We can use inductive learning algorithms to analyze patient data and develop personalized treatment plans considering each patient's unique characteristics.
2. **Recommendation Systems:** Another example of a personalized problem is in the field of recommender systems. Traditional recommender systems use collaborative filtering to recommend products or services to users based on the behavior of similar users. However, this approach may not be effective for users with unique preferences or tastes. We can use inductive learning algorithms to develop personalized recommendations that consider each user's preferences and behavior.
3. **Education:** In the field of education, personalized learning is becoming increasingly popular as a way to improve student outcomes. We can use inductive learning algorithms to analyze student data and develop personalized learning plans considering each student's learning style, strengths, and weaknesses.

Inductive Learning for Risk Classification

One of the most common applications of inductive learning is in risk classification, where the goal is to predict the likelihood of a particular event occurring. Risk classification includes business loan evaluation, bond-rating, and bankruptcy prediction.

Business loan evaluation

Business loan evaluation involves assessing a company's creditworthiness before granting a loan. Inductive learning algorithms can analyze a company's financial data, such as revenue, debt, and profit, to predict the likelihood of defaulting on the loan. The algorithm can then provide a risk score that is used to determine the interest rate and terms of the loan.

Bond-rating

Bond rating involves assessing the creditworthiness of a bond before investing in it. Inductive learning algorithms can analyze a bond's financial data, such as the issuer's credit rating, interest rate, and maturity, to predict the likelihood of defaulting on the bond. The algorithm can then provide a risk score that is used to determine the investment strategy.

Prediction of Bankruptcy

Prediction of bankruptcy involves assessing the likelihood of a company filing for bankruptcy. Inductive learning algorithms can be used to analyze a company's financial data, such as profitability, liquidity, and solvency, to predict the likelihood of filing for bankruptcy. The algorithm can then provide a risk score that is used to make investment decisions.

Inductive Learning Algorithm (ILA)

Inductive learning algorithms (ILA) are used to create models that can learn from examples and make predictions based on those examples. The ILA involves the following steps:

1. Data collection: Collect data from a variety of sources, including databases, sensors, and human input.
2. Data preparation: Clean, transform, and pre-process the data to ensure that it is in a format suitable for training the model.
3. Feature extraction: Identify the relevant features of the data that will be used to train the model.
4. Model training: Use the labeled data to train the model to map inputs to outputs.
5. Model testing: Use a separate set of data to test the accuracy and performance of the model.
6. Model deployment: Use the model to make predictions in new situations.

Inductive learning algorithms can be further classified into two categories: instance-based learning and model-based learning.

- Instance-based learning involves storing all the labeled examples in memory and using them to predict new situations.
- Model-based learning involves creating a mathematical model of the data that can be used to make predictions in new situations.

One of the key advantages of ILA is its ability to learn from examples without prior knowledge of the underlying structure or relationships in the data. This makes it well-suited for solving complex problems where one may not well under the relationship between the input variables and the output variables.

Another advantage of ILA is its ability to adapt to changes in the data over time. Since the algorithm is based on learning from examples, it can continue to improve its accuracy and performance as new data becomes available.

Several types of ILA exist, including decision tree algorithms, rule-based algorithms, and neural networks. Each type of algorithm has its own strengths and weaknesses, and the choice of algorithm will depend on the specific problem being addressed.

Conclusion

- Inductive learning is a powerful technique that is widely used in AI to make predictions and decisions based on specific examples.
- We can use inductive learning algorithms in various applications, including credit risk assessment, disease diagnosis, face recognition, and autonomous driving.
- We can also use inductive learning algorithms in risk classification, where the goal is to predict the likelihood of a particular event occurring.
- The ILA involves the collection, preparation, and transformation of data, feature extraction, model training, model testing, and model deployment.
- With the increasing availability of data and advances in computing power, inductive learning algorithms will continue to play a critical role in AI.

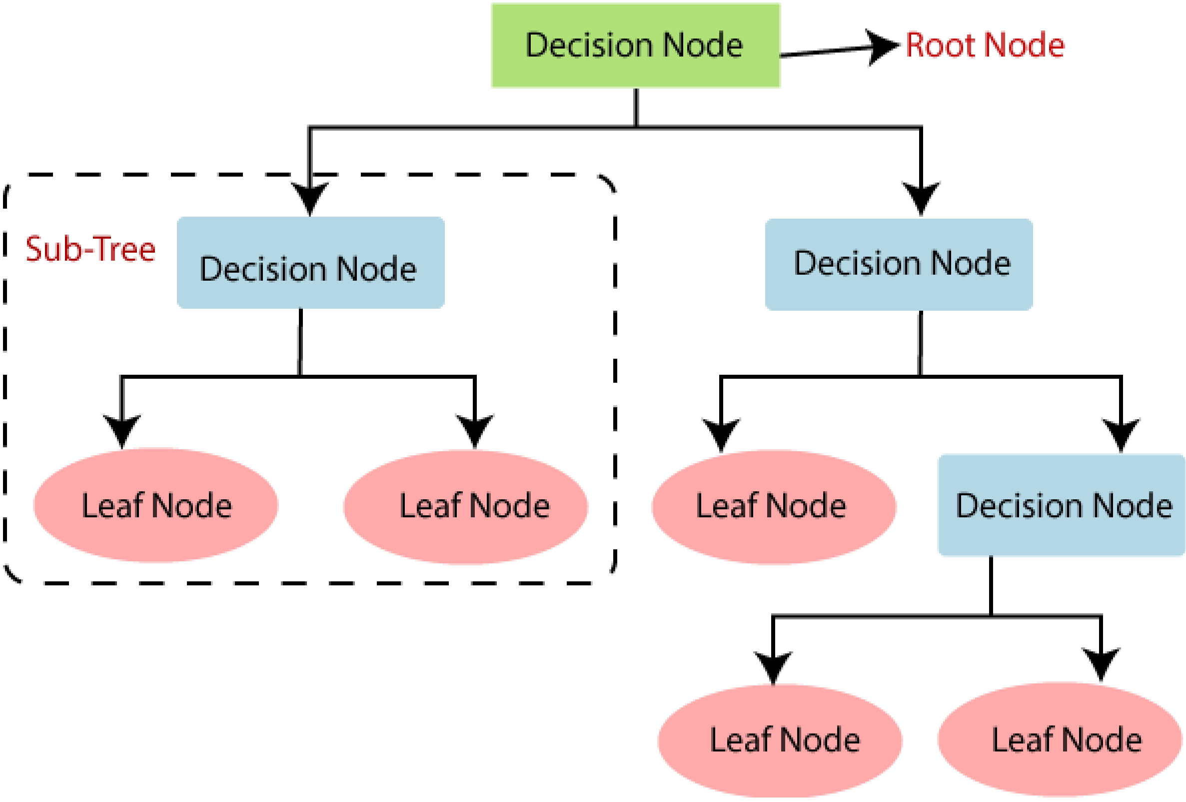
Inductive Learning vs Deductive Learning

	Inductive Learning	**Deductive Learning**
Approach	Bottom-up	Top-down
Data	Specific examples	Logical rules and procedures
Model Creation	Find correlations and patterns in data.	obey clearly stated guidelines and instructions
Training	Adapting model parameters and learning from instances	Programming explicitly and establishing rules
Goal	Using fresh data, generalizing, and making predictions.	Make a model that precisely complies with the given guidelines and instructions.
Examples	Decision trees, neural networks, clustering algorithms	Knowledge-based systems, expert systems, and rule-based systems
Strengths	capable of learning from a variety of complicated data, adaptable, and versatile	accurately when according to established norms and processes, and effective when doing specific duties
Limitations	It may be difficult to manage complex and diverse data and may overfit to specific facts.	limited to well-defined duties and norms, possibly incapable of adjusting to novel circumstances

Decision Tree Classification Algorithm

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

> Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.



Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

- Branch/Sub Tree: A tree formed by splitting the tree.
- Pruning: Pruning is the process of removing the unwanted branches from the tree.
- Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

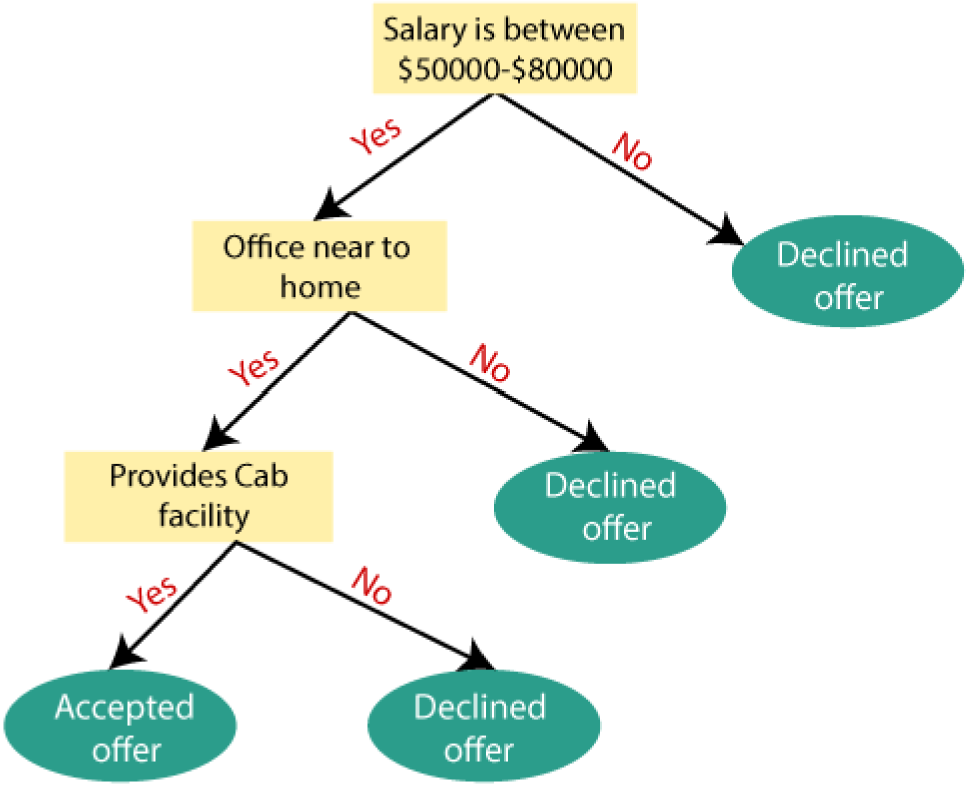
How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contain possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- Information Gain
- Gini Index

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.

- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= 1- $\sum_j P_j^2$

Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree pruning technology used:

- Cost Complexity Pruning
- Reduced Error Pruning.

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

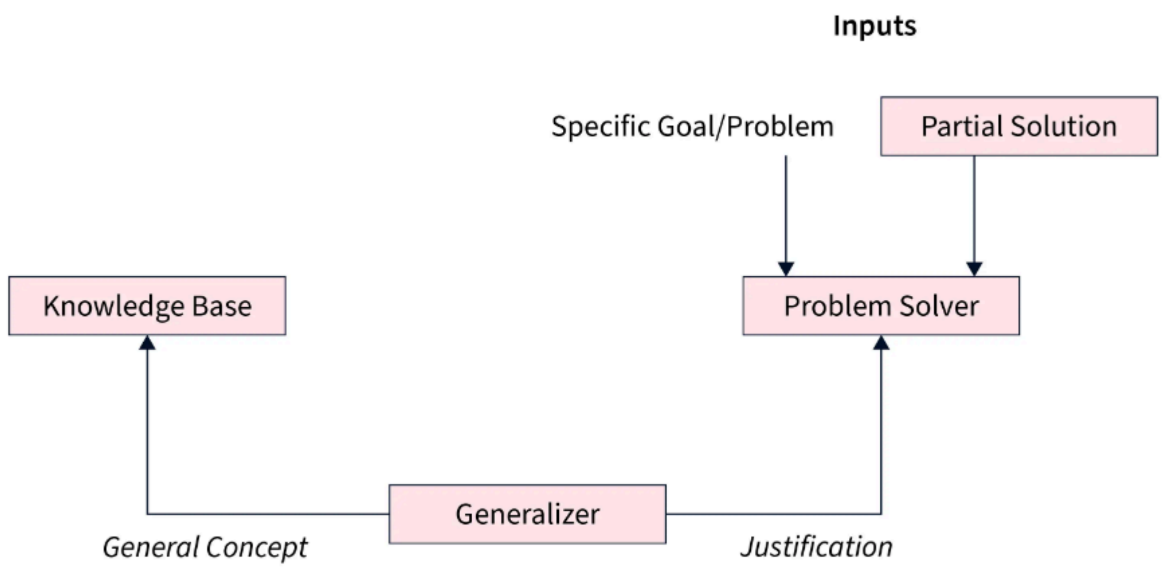
Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

What is Explanation-Based Learning?

- Explanation-based learning in artificial intelligence is a problem-solving method that involves agent learning by analyzing specific situations and connecting them to previously acquired information.
- Also, the agent applies what he has learned to solve similar issues. Rather than relying solely on statistical analysis, EBL algorithms incorporate logical reasoning and domain knowledge to make predictions and identify patterns.

Explanation-based learning architecture:



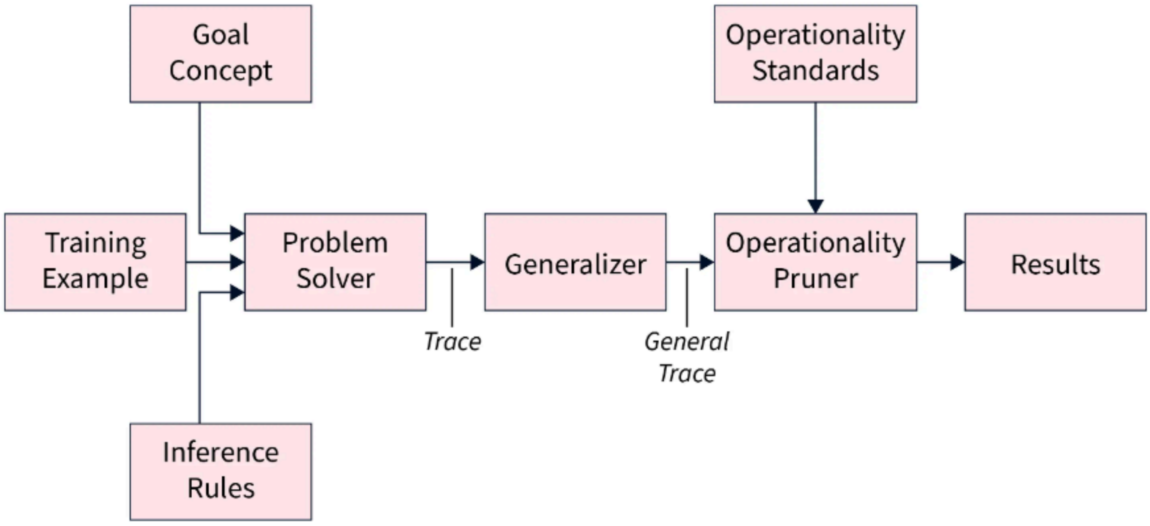
The environment provides two inputs to the EBL architecture:

1. A specific goal, and
2. A partial solution.

The problem solver analyzes these sources and provides reasoning to the generalizer.

The generalizer uses general ideas from the knowledge base as input and compares them to the problem solver's reasoning to come up with an answer to the given problem.

Explanation-based learning System Representation:



- **Problem Solver:** It takes 3 kinds of external inputs: The goal idea is a complex problem statement that the agent must learn. Training instances are facts that illustrate a specific instance of a target idea. Inference rules reflect facts and procedures that demonstrate what the learner already understands.
- **Generalizer:** The problem solver's output is fed into the generalizer, which compares the problem solver's explanation to the knowledge base and outputs to the operational pruner.
- **Operational pruner:** It takes two inputs, one from generalized and the other from operationally standard. The operational standard describes the final concept and defines the format in which the learned concept should be conveyed.

a) The Explanation-Based Learning Hypothesis

According to the Explanation based learning hypothesis, if a system has an explanation for how to tackle a comparable problem it faced previously, it will utilize that explanation to handle the current problem more efficiently. This hypothesis is founded on the concept that learning via explanations is more successful than learning through instances alone.

b) Standard Approach to Explanation-Based Learning

The typical approach to explanation-based learning in artificial intelligence entails the following steps:

1. Determine the problem to be solved
2. Gather samples of previously solved problems that are comparable to the current problem.
3. Identify the connections between the previously solved problems and the new problem.

4. Extraction of the underlying principles and rules used to solve previously solved problems.
5. Apply the extracted rules and principles to solve the new problem.

c) Examples of Explanation-Based Learning

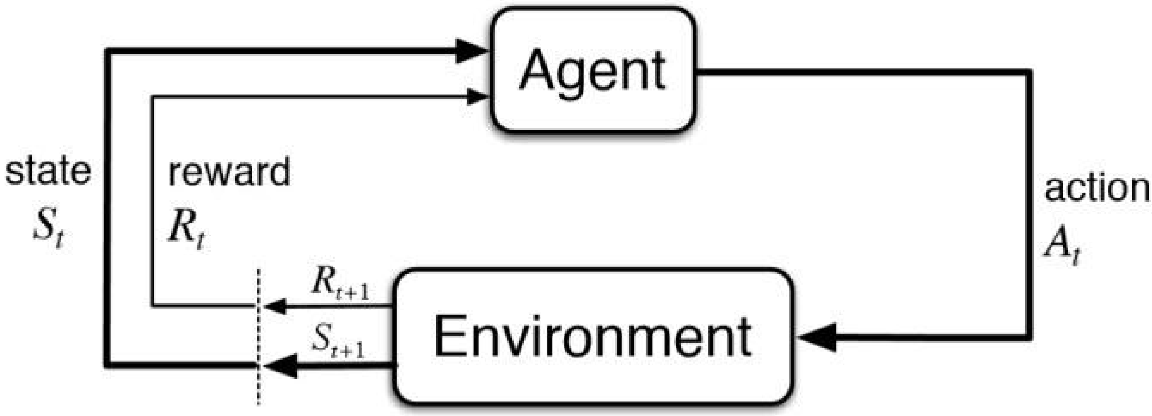
- Medical Diagnosis: Explanation-based learning can be used in medical diagnosis to determine the underlying causes of a patient's symptoms. Explanation-based learning algorithms can find trends and produce more accurate diagnoses by analyzing previously diagnosed instances.
- Robot Navigation: Explanation-based learning may be used to educate robots on how to navigate through complicated settings. Explanation-based learning algorithms can discover the rules and principles that were utilized to navigate those settings and apply them to new scenarios by analyzing prior successful navigation efforts.
- Fraud Detection: Explanation-based learning may be utilized in fraud detection to discover patterns of fraudulent conduct. Explanation-based learning algorithms can find the rules and principles that were utilized to detect prior cases of fraud and apply them to new cases by analyzing previous incidents of fraud.

Conclusion

- Explanation-based learning in artificial intelligence is a powerful tool for solving complex problems efficiently.
- By learning from explanations provided by domain experts, EBL algorithms can identify the underlying principles and rules that govern a particular domain and apply them to new situations.
- Explanation-based learning in the artificial intelligence approach has a wide range of applications, from medical diagnosis to fraud detection, and is poised to play an increasingly important role in the development of advanced AI systems.

Reinforcement learning

- Reinforcement Learning is a part of machine learning. Here, agents are self-trained on reward and punishment mechanisms. It’s about taking the best possible action or path to gain maximum rewards and minimum punishment through observations in a specific situation. It acts as a signal to positive and negative behaviors. Essentially an agent (or several) is built that can perceive and interpret the environment in which it is placed, furthermore, it can take actions and interact with it.

- 

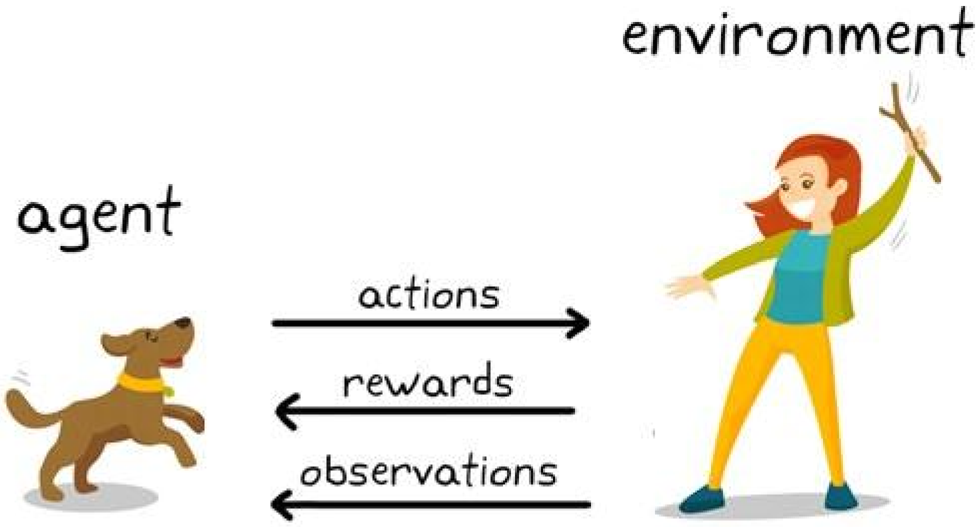
  Basic Diagram of Reinforcement Learning – KDNuggets

- To know the meaning of reinforcement learning, let's go through the formal definition.
- Reinforcement learning, a type of machine learning, in which agents take actions in an environment aimed at maximizing their cumulative rewards – NVIDIA
- Reinforcement learning (RL) is based on rewarding desired behaviors or punishing undesired ones. Instead of one input producing one output, the <u>algorithm</u> produces a variety of outputs and is trained to select the right one based on certain variables – Gartner

- It is a type of machine learning technique where a computer agent learns to perform a task through repeated trial and error interactions with a dynamic environment. This learning approach enables the agent to make a series of decisions that maximize a reward metric for the task without human intervention and without being explicitly programmed to achieve the task – Mathworks
- The above definitions are technically provided by experts in that field however for someone who is starting with reinforcement learning, these definitions might feel a little bit difficult. As this is a reinforcement learning guide for beginners, let's create our reinforcement learning definition in an easier way.
- Simplified Definition of Reinforcement Learning
- Through a series of Trial and Error methods, an agent keeps learning continuously in an interactive environment from its own actions and experiences. The only goal of it is to find a suitable action model which would increase the total cumulative reward of the agent. It learns via interaction and feedback.
- Well, that's the definition of reinforcement learning. Now how we come to this definition, how a machine learns and how it can solve complex problems in the world through reinforcement learning, is something we are going to see further.

How Does Reinforcement Learning Work?

1. Start in a state.
2. Take action.
3. Receive a reward or penalty from the environment.
4. Observe the new state of the environment.
5. Update your policy to maximize future rewards.

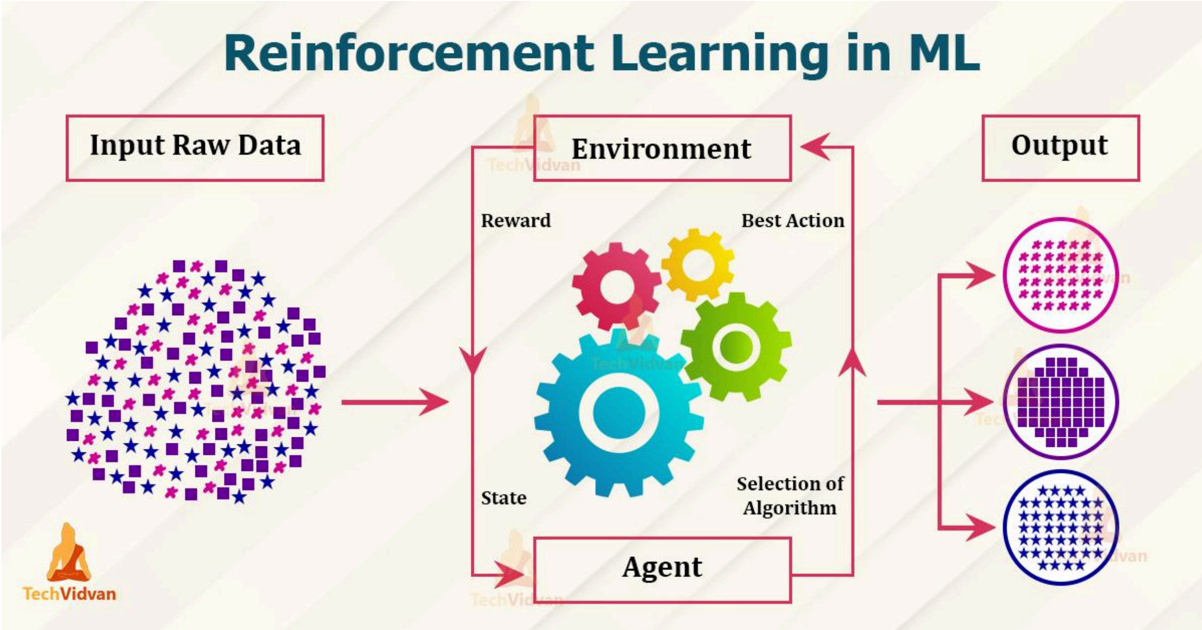


Reinforcement Learning Example – KDNuggets

Here is What do you see?

You can see a dog and a master. Let's imagine you are training your dog to get the stick. Each time the dog gets a stick successfully, you offer him a feast (a bone let's say). Eventually, the dog understands the pattern, that whenever the master throws a stick, it should get it as early as it can to gain a reward (a bone) from a master in a lesser time.

Terminologies used in Reinforcement Learning



Terminologies in RL

Agent – is the sole decision-maker and learner

Environment – a physical world where an agent learns and decides the actions to be performed

Action – a list of action which an agent can perform

State – the current situation of the agent in the environment

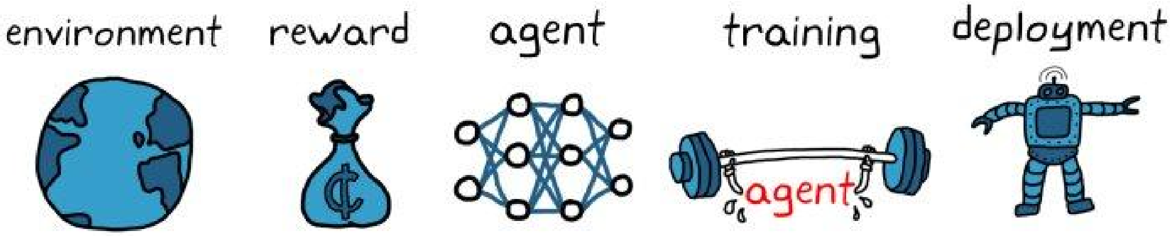
Reward – For each selected action by agent, the environment gives a reward. It's usually a scalar value and nothing but feedback from the environment

Policy – the agent prepares strategy(decision-making) to map situations to actions.

Value Function – The value of state shows up the reward achieved starting from the state until the policy is executed

Model – Every RL agent doesn't use a model of its environment. The agent's view maps state-action pairs probability distributions over the states

Reinforcement Learning Workflow



Reinforcement Learning Workflow – KDNuggets

– **Create the Environment**

– Define the reward

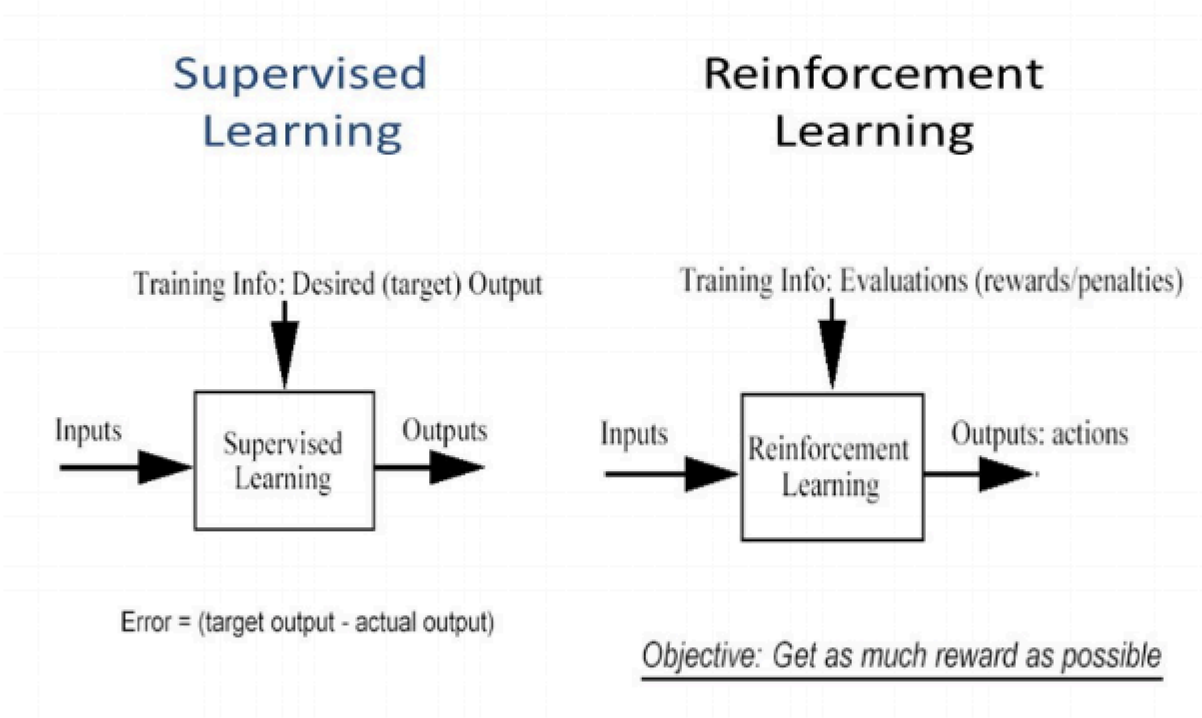
– Create the agent

– Train and validate the agent

– Deploy the policy

How is reinforcement learning different from supervised learning?

● In supervised learning, the model is trained with a training dataset that has a correct answer key. The decision is done on the initial input given as it has all the data that's required to train the machine. The decisions are independent of each other so each decision is represented through a label. Example: Object Recognition



Difference between Supervised and Reinforcement Learning – purestudy

● In reinforcement learning, there isn't any answer and the reinforcement agent decides what to be done to perform the required task. As the training dataset isn't available, the agent had to learn from its experience. It's all about compiling the decisions in a sequential manner. To be said in simpler words, the output relies on the current input state and the next input relies on the output of the previous input. We give labels to the sequence of dependent decisions. Decisions are dependent. Example: Chess Game

Characteristics of Reinforcement Learning

– No supervision, only a real value or reward signal

– Decision making is sequential

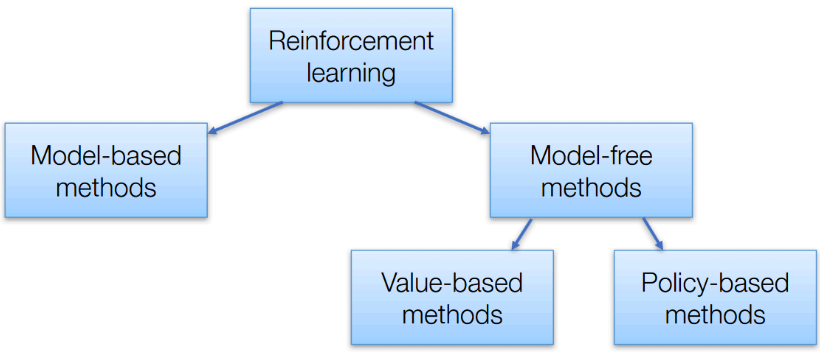
– Time plays a major role in reinforcement problems

– Feedback isn't prompt but delayed

– The following data it receives is determined by the agent's actions

Reinforcement Learning Algorithms

There are 3 approaches to implement reinforcement learning algorithms

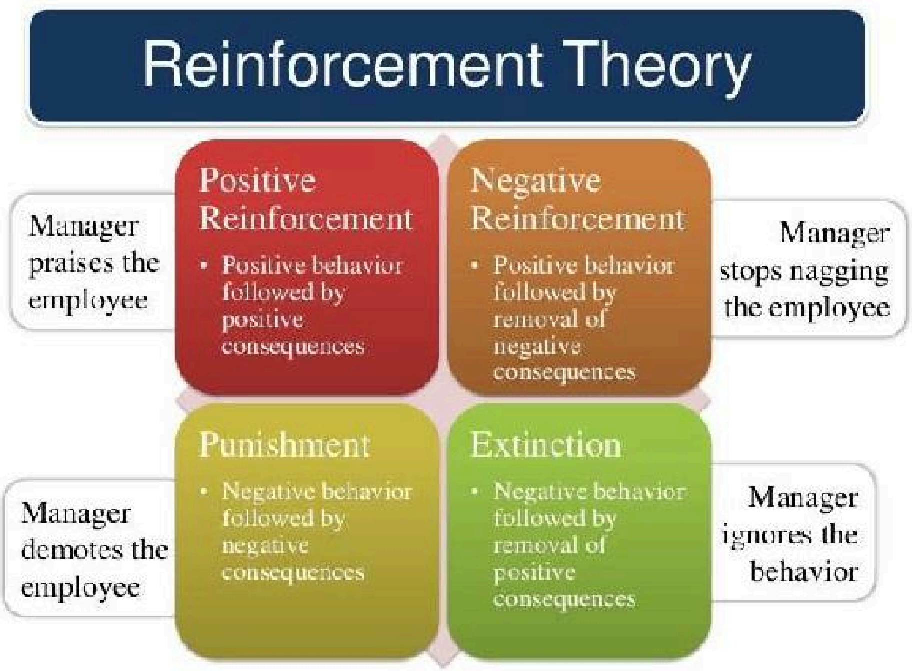


Reinforcement Learning Algorithms – AISummer

- **Value-Based –** The main goal of this method is to maximize a value function. Here, an agent through a policy expects a long-term return of the current state.
- **Policy-Based –** In policy-based, you can come up with a strategy that helps to gain maximum rewards in the future through possible actions performed in each state. Two types of policy-based methods are deterministic and stochastic.
- **Model-Based –** In this method, we need to create a virtual model for the agent to help in learning to perform in each specific environment

Types of Reinforcement Learning

There are two types :



Reinforcement Theory Example – Tutorialspoint

1. Positive Reinforcement

Positive reinforcement is defined as when an event, occurs due to specific behavior, increases the strength and frequency of the behavior. It has a positive impact on behavior.

Advantages

– Maximizes the performance of an action

– Sustain change for a longer period

Disadvantage

– Excess reinforcement can lead to an overload of states which would minimize the results.

2. Negative Reinforcement

Negative Reinforcement is represented as the strengthening of a behavior. In other ways, when a negative condition is barred or avoided, it tries to stop this action in the future.

Advantages

– Maximized behavior

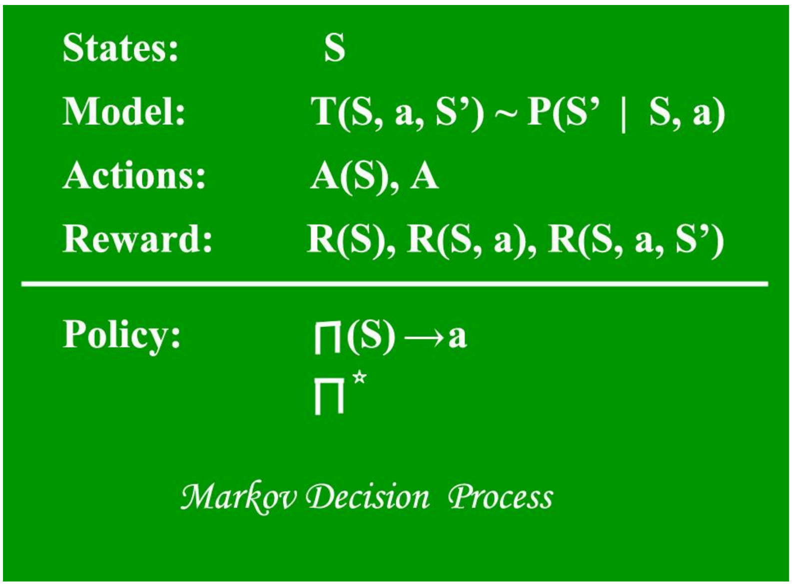
– Provide a decent to minimum standard of performance

Disadvantage

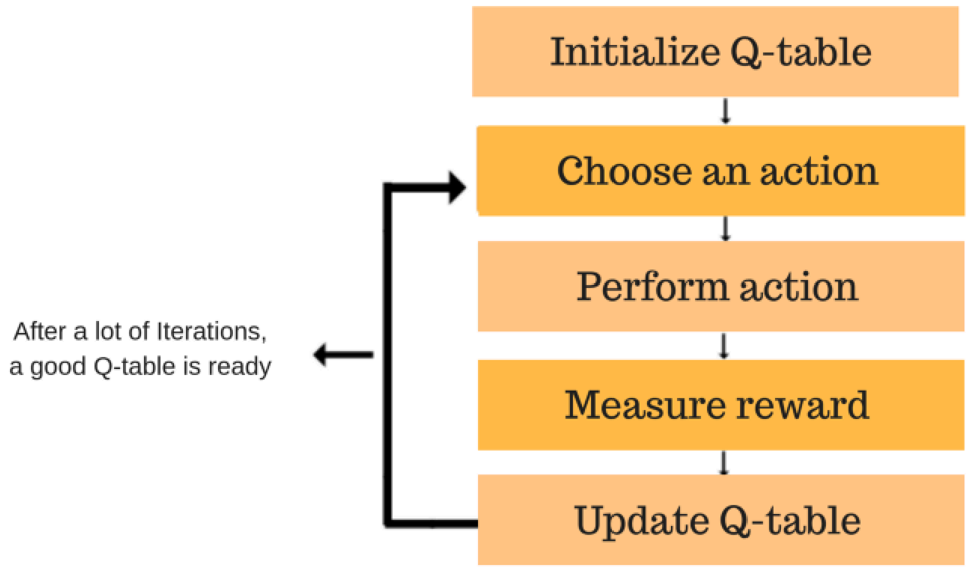
– It just limits itself enough to meet up a minimum behavior

Widely used models for reinforcement learning

1. Markov Decision Process (MDP's) – are mathematical frameworks for mapping solutions in RL. The set of parameters that include Set of finite states – S, Set of possible Actions in each state – A, Reward – R, Model – T, Policy – π. The outcome of deploying an action to a state doesn't depend on previous actions or states but on current action and state.



2. Q Learning – it's a value-based model free approach for supplying information to intimate which action an agent should perform. It revolves around the notion of updating Q values which shows the value of doing action A in state S. Value update rule is the main aspect of the Q-learning algorithm.



QLearning

Practical Applications of reinforcement learning

- – Robotics for Industrial Automation
- – Text summarization engines, dialogue agents (text, speech), gameplays
- – Autonomous Self Driving Cars
- – Machine Learning and Data Processing
- – Training system which would issue custom instructions and materials with respect to the requirements of students
- – AI Toolkits, Manufacturing, Automotive, Healthcare, and Bots
- – Aircraft Control and Robot Motion Control
- – Building artificial intelligence for computer games

Conclusion

Reinforcement learning guides us in determining actions that maximize long-term rewards. However, it may struggle in partially observable or non-stationary environments. Moreover, its effectiveness diminishes when ample supervised learning data is available. A key challenge lies in managing parameters to optimize learning speed.

Difference between Reinforcement learning and Supervised learning:

Reinforcement learning	Supervised learning
Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input	In Supervised learning, the decision is made on the initial input or the input given at the start
In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions	In supervised learning the decisions are independent of each other so labels are given to each decision.

Example: Chess game,text summarization	Example: Object recognition,spam detection

Types of Reinforcement:

There are two types of Reinforcement:

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

   Advantages of reinforcement learning are:

   - Maximizes Performance
   - Sustain Change for a long period of time
   - Too much Reinforcement can lead to an overload of states which can diminish the results
2. **Negative:** Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided.
   Advantages of reinforcement learning:

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

Elements of Reinforcement Learning

Reinforcement learning elements are as follows:

1. **Policy**
2. **Reward function**
3. **Value function**
4. **Model of the environment**

Policy: Policy defines the learning agent behavior for a given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.

Reward function: Reward function is used to define a goal in a reinforcement learning problem.A reward function is a function that provides a numerical score based on the state of the environment

Value function: Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

Model of the environment: Models are used for planning.

Credit assignment problem: Reinforcement learning algorithms learn to generate an internal value for the intermediate states as to how good they are in leading to the goal. The learning decision maker is called the agent. The agent interacts with the environment that includes everything outside the agent.

The agent has sensors to decide on its state in the environment and takes action that modifies its state.

The reinforcement learning problem model is an agent continuously interacting with an environment. The agent and the environment interact in a sequence of time steps. At each time step t, the agent receives the state of the environment and a scalar numerical reward for the previous action, and then the agent then selects an action.

Reinforcement learning is a technique for solving Markov decision problems.

Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. This framework is intended to be a simple way of representing essential features of the artificial intelligence problem.

Various Practical Applications of Reinforcement Learning –

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

Application of Reinforcement Learnings

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.

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

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

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

4. Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.

5. Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.

6. Reinforcement learning is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance.

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

3. Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behavior.

4. Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.