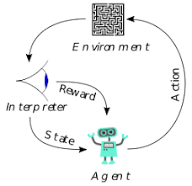
**ML with Graphs assignment4**

**1. What do you understand by reinforcement learning in Machine Learning, also enlist its applications?**

**Ans.**

* Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.
* Main points in Reinforcement learning –
* Input: The input should be an initial state from which the model will start
* Output: There are many possible outputs as there are a variety of solutions to a particular problem
* Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
* The model keeps continues to learn.
* The best solution is decided based on the maximum reward.

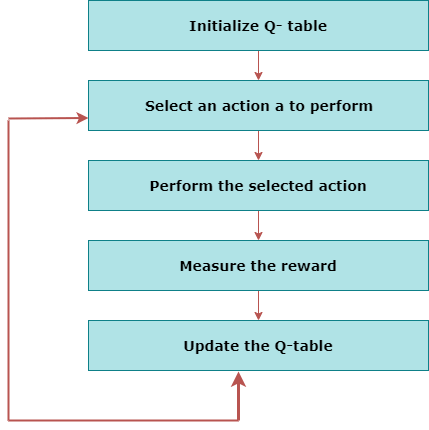


* The environment is typically stated in the form of a Markov decision process (MDP), because many reinforcement learning algorithms for this context use dynamic programming techniques. The main difference between the classical dynamic programming methods and reinforcement learning algorithms is that the latter do not assume knowledge of an exact mathematical model of the MDP and they target large MDPs where exact methods become infeasible.
* Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.
* There are no rules for how to solve the problem, It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills.
* Suppose there is an AI agent present within a maze environment, and his goal is to find the diamond. The agent interacts with the environment by performing some actions, and based on those actions, the state of the agent gets changed, and it also receives a reward or penalty as feedback.
* The agent continues doing these three things (take action, change state/remain in the same state, and get feedback), and by doing these actions, he learns and explores the environment.
* The agent learns that what actions lead to positive feedback or rewards and what actions lead to negative feedback penalty. As a positive reward, the agent gets a positive point, and as a penalty, it gets a negative point.
* There are four main elements of Reinforcement Learning, which are given below:
* 1) Policy: A policy can be defined as a way how an agent behaves at a given time. It maps the perceived states of the environment to the actions taken on those states. A policy is the core element of the RL as it alone can define the behavior of the agent. In some cases, it may be a simple function or a lookup table, whereas, for other cases, it may involve general computation as a search process.
* 2) Reward Signal: The goal of reinforcement learning is defined by the reward signal. At each state, the environment sends an immediate signal to the learning agent, and this signal is known as a reward signal. These rewards are given according to the good and bad actions taken by the agent. The agent's main objective is to maximize the total number of rewards for good actions. The reward signal can change the policy, such as if an action selected by the agent leads to low reward, then the policy may change to select other actions in the future.
* 3) Value Function: The value function gives information about how good the situation and action are and how much reward an agent can expect. A reward indicates the immediate signal for each good and bad action, whereas a value function specifies the good state and action for the future. The value function depends on the reward as, without reward, there could be no value. The goal of estimating values is to achieve more rewards.
* 4) Model: The last element of reinforcement learning is the model, which mimics the behavior of the environment. With the help of the model, one can make inferences about how the environment will behave. Such as, if a state and an action are given, then a model can predict the next state and reward. The model is used for planning, which means it provides a way to take a course of action by considering all future situations before actually experiencing those situations. The approaches for solving the RL problems with the help of the model are termed as the model-based approach. Comparatively, an approach without using a model is called a model-free approach.

**2. Describe about Deep Q-Network in Machine Learning and how it plays its key role.**

**Ans.**

* Q-learning is an Off policy RL algorithm, which is used for the temporal difference Learning. The temporal difference learning methods are the way of comparing temporally successive predictions.
* It learns the value function Q (S, a), which means how good to take action "a" at a particular state "s.“
* The flowchart explains the working of Q- learning.
* As the name suggests, DQN is a Q-learning using Neural networks.
* For a big state space environment, it will be a challenging and complex task to define and update a Q-table.
* To solve such an issue, we can use a DQN algorithm. Where, instead of defining a Q-table, neural network approximates the Q-values for each action and state.



**Q Learning**

Let’s say we know the expected reward of each action at every step. This would essentially be like a cheat sheet for the agent! Our agent will know exactly which action to perform.

It will perform the sequence of actions that will eventually generate the maximum total reward. This total reward is also called the Q-value and we will formalise our strategy as:

The above equation states that the Q-value yielded from being at state s and performing action a is the immediate reward r(s,a) plus the highest Q-value possible from the next state s’. Gamma here is the discount factor which controls the contribution of rewards further in the future.

Q(s’,a) again depends on Q(s”,a) which will then have a coefficient of gamma squared. So, the Q-value depends on Q-values of future states as shown here

Adjusting the value of gamma will diminish or increase the contribution of future rewards.

Since this is a recursive equation, we can start with making arbitrary assumptions for all q-values. With experience, it will converge to the optimal policy. In practical situations, this is implemented as an update:

where alpha is the learning rate or step size. This simply determines to what extent newly acquired information overrides old information.

**Why ‘Deep’ Q-Learning?**

Q-learning is a simple yet quite powerful algorithm to create a cheat sheet for our agent. This helps the agent figure out exactly which action to perform

But what if this cheatsheet is too long? Imagine an environment with 10,000 states and 1,000 actions per state. This would create a table of 10 million cells. Things will quickly get out of control!

It is pretty clear that we can’t infer the Q-value of new states from already explored states. This presents two problems:

First, the amount of memory required to save and update that table would increase as the number of states increases

Second, the amount of time required to explore each state to create the required Q-table would be unrealistic

Here’s a thought – what if we approximate these Q-values with machine learning models such as a neural network? Well, this was the idea behind DeepMind’s algorithm that led to its acquisition by Google for 500 million dollars!

**Deep Q-Networks**

In deep Q-learning, we use a neural network to approximate the Q-value function. The state is given as the input and the Q-value of all possible actions is generated as the output. The comparison between Q-learning & deep Q-learning is wonderfully illustrated below:

So, what are the steps involved in reinforcement learning using deep Q-learning networks (DQNs)?

All the past experience is stored by the user in memory

The next action is determined by the maximum output of the Q-network

The loss function here is mean squared error of the predicted Q-value and the target Q-value – Q\*. This is basically a regression problem. However, we do not know the target or actual value here as we are dealing with a reinforcement learning problem. Going back to the Q-value update equation derived fromthe Bellman equation. we have:

The section in green represents the target. We can argue that it is predicting its own value, but since R is the unbiased true reward, the network is going to update its gradient using backpropagation to finally converge.

**4. Compare the reinforcement learning vs. Supervised and Unsupervised Learning in a tabular form.**

**Ans.**

* Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.
* Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented, and in not needing sub-optimal actions to be explicitly corrected. Instead the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge). Partially supervised RL algorithms can combine the advantages of supervised and RL algorithms.
* supervised learning is when a model learns from a labeled dataset with guidance. And, unsupervised learning is where the machine is given training based on unlabeled data without any guidance. Whereas reinforcement learning is when a machine or an agent interacts with its environment, performs actions, and learns by a trial-and-error method.

| **Criteria** | **Supervised ML** | **Unsupervised ML** | **Reinforcement ML** |
| --- | --- | --- | --- |
| Definition | Learns by using labelled data | Trained using unlabelled data without any guidance. | Works on interacting with the environment |
| Type of data | Labelled data | Unlabelled data | No – predefined data |
| Type of problems | Regression and classification | Association and Clustering | Exploitation or Exploration |
| Supervision | Extra supervision | No supervision | No supervision |
| Algorithms | Linear Regression, Logistic Regression, SVM, KNN etc. | K – Means, C – Means, Apriori | Q – Learning, SARSA |
| Aim | Calculate outcomes | Discover underlying patterns | Learn a series of action |
| Application | Risk Evaluation, Forecast Sales | Recommendation System, Anomaly Detection | Self Driving Cars, Gaming, Healthcare |

**5. How Dynamic Programming help the user of applications based on Machine Learning?**

**Ans.**

* DP is a collection of algorithms that can solve a problem where we have the perfect model of the environment and where an agent can only take discrete actions.
* DP essentially solves a planning problem rather than a more general RL problem. The main difference, as mentioned, is that for an RL problem the environment can be very complex and its specifics are not known at all initially.
* Apart from being a good starting point for grasping reinforcement learning, dynamic programming can help find optimal solutions to planning problems faced in the industry, with an important assumption that the specifics of the environment are known. DP presents a good starting point to understand RL algorithms that can solve more complex problems.
* Dynamic programming algorithms solve a category of problems called planning problems. Herein given the complete model and specifications of the environment (MDP), we can successfully find an optimal policy for the agent to follow. It contains two main steps:
* Break the problem into subproblems and solve it
  + Solutions to subproblems are cached or stored for reuse to find overall optimal solution to the problem at hand
  + To solve a given MDP, the solution must have the components to:
* To solve a given MDP, the solution must have the components to:
  + Find out how good an arbitrary policy is
  + Find out the optimal policy for the given MDP

**7. Briefly discuss the key features and applications of Markov Decision Process in the context of Machine Learning?**

**Ans.**

* MDP is used to describe the environment for the RL, and almost all the RL problem can be formalized using MDP.
* MDP contains a tuple of four elements (S, A, Pa, Ra):
* A set of finite States S
* A set of finite Actions A
* Rewards received after transitioning from state S to state S', due to action a.
* Probability Pa.
* MDP uses Markov property, and to better understand the MDP, we will see about Markov property
* Any random process in which the probability of being in a given state depends only on the previous state, is a markov process.
* In other words, in the markov decision process setup, the environment’s response at time t+1 depends only on the state and action representations at time t, and is independent of whatever happened in the past.
* . diagram1

Diagram

Description automatically generated

* St: State of the agent at time t
* At: Action taken by agent at time t
* Rt: Reward obtained at time t
* The above diagram clearly illustrates the iteration at each time step wherein the agent receives a reward Rt+1 and ends up in state St+1based on its action At at a particular state St. The overall goal for the agent is to maximise the cumulative reward it receives in the long run. Total reward at any time instant t is given by:
* . diagram 2

A picture containing text

Description automatically generated

* where T is the final time step of the episode. In the above equation, we see that all future rewards have equal weight which might not be desirable. That’s where an additional concept of discounting comes into the picture. Basically, we define γ as a discounting factor and each reward after the immediate reward is discounted by this factor as follows:
* . diagram 3

A close up of a calculator

Description automatically generated with low confidence