Prasad Jawale D16AD 20

Date
Page

Reinforcement learning lab 1\2

Experiment 1

Aim: To implement the python program to implement a simple grid ~~leve~~ world environment and training an agent using basic Q-learning

Theory:

1) Q learning is a model free reinforcement learning algorithm used to find the optimal action-selection policy for a given finite Markov Decision Process (MDP).

2) The goal of Q-learning is to learn a policy, represented by the Q-values that maximises the cumalative reward over time.

3) Reinforcement learning (RL) is a type of machine learning where an agent learns how to behave in an environment by performing actions and recieving rewards.

4) The agent aims to learn a policy or a strategy that maximises the cumalative reward over time.

5) key concepts of Reinforcement learning are

① Agent - The learner or decision maker that interacts with the environment

② Environment - The external system with which the agent interacts

Teacher's Sign.: ____________

③ State (s) - A representation of a current situation or configuration of environment.

④ Action (a) - The set of possible moves or decisions that the agent can take in a given state

⑤ Reward (R) - A numeric value that the environment provides to the agent after it performs an action

⑥ Policy (π) - A strategy or mapping from states to actions that guides the agent's decision making

⑦ Value function - Describes the cumulative reward or value of being in a particular state or taking a action.

6) Different types of Reinforcement learning algorithms are

1. Q-learning
2. Deep Q Network (DQN)
3. Policy Gradient methods
4. Actor Critic
5. Deep deterministic policy gradients (DDPG)
6. Trust region policy optimisation (TRPO)
7. Soft actor critic (SAC)
8. Twin delayed DDPG (TD3)

7) Q learning works by creating a Q table and iteratively improve its action over time by updating values.

Teacher's Sign.: ____________

Algorithm

Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
Initialize s
Repeat (for each step of episode):
- Choose a from s using policy derived from Q
- Take action a, observe r, s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

$s \leftarrow s'$;
until s' is terminal

Conclusion: Q learning algorithm was implemented.

Vemburaj Konar
28
D16AD

RL

Exp-2

DATE:

Aim: Implementing State Action Reward State Action (SARSA) algorithm using python and compare it with Q Learning

Theory:

State Action Reward State Action is another reinforcement learning algorithm that is similar to Q Learning but differs in the way it updates Q-Values

SARSA Algorithm

1) Initialize Q table : Create a table where rows represent states and columns represent actions. Initialize Q values arbitrarily
2) Exploration vs Exploitation : The agent decides whether to explore new actions or exploit known ones based on policy
3) Action selection : Choose an action based on current state considering exploration and exploitation
4) observation and update : observe the reward received and then the new state resulting from the action taken
5) Update Q-value : update the Q value of the previous state action pair using the SARSA equation
6) Repeat : Repeat steps 3-5 until convergence or a predefined number of iterations

SARSA Equation

The SARSA algorithm updates the Q values of a state action pair using the following equation

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\,(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

- $Q(s,a)$: Q - Value of the state action pair.
- α : Learning rate ($0 < \alpha <= 1$) determines the weight of new information in updating the Q-value
- r : Immediate reward recieved after taking action a in state s
- γ : Discount factor ($0 < \gamma <= 1$) balances the importance of immediate and future rewards
- $Q(s', a')$: Q - Value of the next state s' and the action a' taken in that state

Intution

- The term $r_t + \gamma . Q(s', a')$ represents the total expected reward the agent can achieve from the current state onwards, following the policy
- By subtracting the current $Q(s, a)$ from this total reward, the algorithm updates the Q-value to make it closer to the expected reward
- Like in Q learning, the learning rate α controls how much the new information influences the update process, while the discount factor γ emphasizes the importance of future rewards
- SARSA learns directly from actions taken and their corresponding consequences, making it an on policy algorithm where it evaluates and improves its policy simulteneously

Conclusion:

= In Conclusion both SARSA & Q Learning are powerful reinforcement learning algorithms with their unique characteristics and trade offs. Understanding their differences and performance nuances is crucial for effectively applying them to various reinforcement learning tasks and domains

Shruti Devlekar
D15AD-09

Reinforcement Learning Lab

Experiment - 3

AIM: Experimenting with different exploration strategies and analyzing their impact on the learning performance of an agent in a bandit problem using Epsilon-greedy strategy

THEORY: 'Exploration vs Exploitation dilemma' is one of the key concepts in reinforcement learning.

As in RL, the agent is not aware of the different states, actions for each state, associate rewards, and transition to the next state, but it learns it by exploring the environment. However, the knowledge of an agent about the state, actions, rewards, and resulting states is partial, which results in Exploration-Exploitation Dilemma.

Exploitation is defined as a greedy approach in which agents try to get more rewards by using estimated value but not the actual value. So, in this technique, agents make the best decision based on current information. Unlike exploitation, in exploration techniques, agents primarily focus on improving their knowledge about each action instead of getting more rewards so that they can get long-term benefits. So, in this technique, agents work on gathering more rewards so that they can get long-term benefits. So, in this techniques, agent work on gathering more information

The Seven Exploration strategies in RL are :-

1. Epsilon-Greedy : This is a simple yet effective strategy. Here, the agent chooses the best-known action (exploitation) with probability 1-ε and a random action (exploration) with probability ε. The value of ε typically starts high and decreases over time, allowing for more exploration early on and more exploitation as the agent learns about the environment.

2. Softmax Exploration (Boltzmann Exploration) : Rather than making a hard choice between exploration and exploitation, softmax exploration assigns a probability to each action based on its value estimate, with higher-valued actions being more likely but still allowing for exploration. This is done using the softmax function, which converts the action values into probabilities.

3. Upper Confidence Bound (UCB) : It is more sophisticated and uses uncertainty in the action-value estimates to drive exploration. It selects actions based on both the estimated value and the uncertainty or variance associated with those estimates. This approach naturally balances exploration and exploitation by preferring actions that are either highly rewarding or poorly understood.

4. Thompson Sampling : This Bayesian approach samples from the posterior distributions of the action value estimates to decide which action to take. It inherently balances exploration and exploitation by considering the uncertainty in the value estimates, with actions that have more uncertain estimates being explored more frequently.

5. Optimistic Initial Values : This strategy involves initialising the estimates of the action values optimistically (higher than the maximum possible value). This encourages exploration, as the agent will prefer actions that it knows less about, assuming they might yield the optimistically high rewards until it learns otherwise.

6. Noise Addition : Adding noise to the action-value estimates or directly to the actions chosen by the policy can encourage exploration. For example, parameter noise added to the weights of a neural network policy can lead to exploration in policy space, while action noise can lead to exploration in action space.

7. Intrinsic Motivation : This approach involves adding an intrinsic reward to the extrinsic reward provided by the environment. The intrinsic reward is based on the novelty or information gain of an action or state, encouraging the agent to explore unseen or poorly understood parts of the environment.

CONCLUSION :

NAME:__________ STD.:______ DIV.:________ | DATE :
PAGE :

Aim: Implementing a Multi-Armed bandit Problem using Python and Comparing USB Exploration Strategies using python.

Theory:

IN RL, A MULTI ARMED BANDIT IS A CLASSIC Problem that Serves as Simplified version of Sequential Decision Making tasks. The name "bandit" Comes from the ~~[illegible]~~ analogy of a gambler facing multiple slot machines and trying to maximize their Cumulative reward over time

Setup:

1) The agent is presented with Set of arms or actions, each associated with an unknown reward Distributation

2) Action Selection:
At each ~~time~~ step, the agent selects one action based on its Current policy

3) Reward Feedback:
After Selecting an action the agent receives a reward from the choosen actions reward distrebutation

plt.show()

DATE :
PAGE :

NAME: STD.: DIV.:

4) Learning:
As the Agent interacts with the environment, it learns about the reward distributions associated with each action

5) Exploration vs Exploitation:
One of the key challenges in multi armed bandit problem is balancing exploration and exploitation

6) Regret:
In the context of the multi-armed bandit problem regret measure the opportunity cost of not selecting the optimal action at each time step.

The working of UCB Exploration Strategy:

1) Estimation of Action Value:
The agent maintains estimates of the expected rewards for each action based on its past experience

2) Upper Confidence Bound Calculation:
In addition to the estimated action values, the agent calculates an upper confidence bound (UCB) for each action.

DATE :
PAGE :
NAME: STD.: DIV.:

3) Action Selection:
At each time step, the agent selects action with highest upper confidence bound. By selecting actions based on their UCB's the agent prioritizes action with higher uncertainty, encouraging exploration of less favoured actions

4) Exploration & Exploitation:
Initially actions with high uncertainty are favoured due to their high UCBS, promoting exploration. As the agent gathers more data and reduces uncertainity, the UCBS shrink and the agent tends to exploit action with higher estimated values

5) Trade off:
The UCB exploration strategy balances between exploring new actions to learn more about their rewards and known actions to maximize immediate rewards

6) Regret minimization:
Regret ~~measure~~ the difference between the rewards obtained by the agent and the rewards it would have received if it had choosen the optimal action at each time step

NAME:______________ STD.:______ DIV.:______

DATE :
PAGE :

Conclusion:

The experiment underscores the importance of principled exploration strategies in RL tasks and highlights the potential of UCB exploration in optimizing decision making processes in uncertain environments

RL

Experiment - 05

Aim:→ Implementing a basic grid world environment as a MDP and applying policy evaluation and policy iteration on it.

Theory :→

① Markov decision process (MDP)

→ A MDP is a mathematical framework used to model decision making in situations where outcomes are partly random and partly under the control of a decision maker.

→ The key components of a MDP are:→

① State (s) : A set of all possible situations or configuration in which the system can exist for example on a grid environment, each cell on a grid may represent a state.

② Actions :- A set of all possible decisions or choices that the decision makers can make in each state for example, moving up, down, bottom or right on the grid environment.

③ Transition Probability (P) : A function that describes the probability of transitioning from one state to another after taking a specific action.

(4) Reward (R): A function that assigns a numerical value to each state action pair, representation the immidiate benifit or cost associated by actions.

(5) Discount factor (γ): A parameter that represents relative importance to future rewards compared to immidiate rewards.

(II) Markov property :- It states that the future state of a stochastic process depends only on the present state and is independent of the past states, given the present states

Conclusion :→ Markov decision process was applied to the grid world environment.