**Experiment :- 13**

|  |
| --- |
| **Title:**   **Write a program to implement Actor-critic algorithm** |

# Objective:

# Understand Actor-critic algorithm

# Apply Actor-critic algorithm by implementing it.

# Books/ Journals/ Websites referred:

* Markov Decision Processes in Artificial Intelligence MDPs, Beyond MDPs and Applications, Edited by Olivier Sigaud, Olivier Buffet, Wiley Publications, 2010
* https://pylessons.com/A2C-reinforcement-learning/
* https://medium.com/intro-to-artificial-intelligence/the-actor-critic-reinforcement-learning-algorithm-c8095a655c14
* https://towardsdatascience.com/reinforcement-learning-w-keras-openai-actor-critic-models-f084612cfd69
* https://www.tensorflow.org/tutorials/reinforcement\_learning/actor\_critic
* https://github.com/dennybritz/reinforcement-learning/blob/master/PolicyGradient/CliffWalk%20Actor%20Critic%20Solution.ipynb
* https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/reinforcement\_learning/actor\_critic.ipynb

# Resources used:

# python

# Theory:

# The Actor-Critic algorithm is a reinforcement learning technique that combines elements of both policy-based methods (the actor) and value-based methods (the critic) to learn optimal actions in a given environment.

# In the Actor-Critic algorithm, the actor is responsible for learning the optimal policy by selecting actions that maximize the expected reward. The critic evaluates the actions taken by the actor by estimating the state-value function, which provides an estimate of how good a particular state is in terms of expected future rewards. The critic then provides feedback to the actor, which is used to improve the policy.

# The actor and critic work together to update the policy and the state-value function, respectively. The actor is updated using policy gradients, which are computed using the state-value function as a baseline. The critic is updated using temporal difference learning (TD) to estimate the state-value function.

# The Actor-Critic algorithm has several advantages over other reinforcement learning methods. It can handle high-dimensional action spaces and it is less sensitive to hyperparameters than other methods. It is also computationally efficient and can learn in online, on-policy settings.

# Overall, the Actor-Critic algorithm is a powerful and flexible technique for solving a wide range of reinforcement learning problems, including those with continuous action spaces and stochastic environments.

# The Action-Critic Algorithm combines the advantages of both the actor-critic and Q-learning algorithms. It provides a more stable and efficient learning process by using the Q-function to update the actor's policy and the critic's value function estimates. This algorithm has been successfully used in many applications such as robotics, gaming, and control systems.

# The Actor-Critic algorithm is a reinforcement learning algorithm that combines two models: the actor and the critic. The actor learns to select actions that maximize the expected return, while the critic learns to estimate the value function of a state or state-action pair.

# Here's a more detailed explanation of how the Actor-Critic algorithm works:

# 1) Define the problem: First, we define the problem we want to solve, which is typically framed as a Markov Decision Process (MDP). An MDP is defined as a tuple (S, A, R, P, γ), where S is the set of states, A is the set of actions, R(s,a) is the reward function, P(s'|s,a) is the transition probability function, and γ is the discount factor.

# 2) Initialize the actor and critic: We initialize the actor and critic models, which are typically neural networks that take the current state as input and output the probability distribution over actions (actor) or the value function estimates (critic).

# 3) Interact with the environment: We interact with the environment by repeatedly observing the current state, selecting an action using the actor's policy, receiving a reward, and transitioning to the next state. This process is repeated until a terminal state is reached.

# 4) Update the critic: After each interaction with the environment, the critic updates its value function estimates based on the observed rewards and the predicted values of the next state. This is done using a temporal difference (TD) learning algorithm, such as TD(0) or TD(λ).

# 5) Update the actor: After the critic updates its value function estimates, the actor updates its policy to maximize the expected return based on the value function estimates. This is done using the policy gradient method, which involves computing the gradient of the expected return with respect to the actor's parameters and updating them accordingly.

# Repeat steps 3-5: We repeat steps 3-5 until convergence, which is typically defined as the point at which the value function estimates and the policy have stabilized.

# Simple Example:-

# Suppose you have an agent that is trying to learn how to play a game. The state of the game is represented by a set of features, such as the position of the player, the position of the enemies, and the current score. The agent can take actions that change the state of the game, such as moving the player left or right, or shooting a bullet.

# The Actor-Critic algorithm consists of two parts: the Actor and the Critic.

# The Actor is responsible for choosing actions based on the current state of the game. It takes the current state as input and outputs a probability distribution over the possible actions. For example, if the agent is playing a game where it can move left or right, the Actor might output a probability of 0.7 for moving right and 0.3 for moving left.

# The Critic is responsible for estimating the value of the current state. It takes the current state as input and outputs a value function, which represents the expected total reward the agent will receive from that state onwards. For example, if the agent is playing a game where the goal is to maximize the score, the Critic might output a high value for states where the score is high, and a low value for states where the score is low.

# During training, the agent interacts with the environment and collects a sequence of states, actions, and rewards. The Actor and Critic are updated based on this sequence of experiences, using the following steps:

# 1) Compute the value of the current state using the Critic.

# 2) Use the Actor to choose an action based on the current state.

# 3) Take the chosen action and observe the next state and reward.

# 4) Compute the TD error, which is the difference between the predicted value of the current state and the actual value of the next state plus the reward.

# 5) Update the Critic by minimizing the squared TD error.

# 6) Update the Actor using the TD error as a signal to adjust the probabilities of the chosen action.

# This process is repeated for multiple episodes until the agent has learned a good policy for the game.

# Implementation (Code):

**Kindly find code and output in E13\_actor\_critic\_01/2.ipynb**

# Output Screenshots:

# Conclusion (Students should write in their own words):

# In Conclusion, the Actor-Critic algorithm is a reinforcement learning method that combines the benefits of both policy-based and value-based methods. The Actor learns to choose good actions based on the current state, while the Critic learns to estimate the expected total reward of a state. The two components work together to improve the agent's policy over time.

# Applications:

# The Actor-Critic algorithm has been applied to various domains, some of which include:

# 1) Robotics: Actor-Critic has been used for training robots to perform various tasks, such as grasping objects and navigating through environments.

# 2) Game Playing: Actor-Critic has been used to train agents to play games such as chess, go, and poker.

# 3) Finance: Actor-Critic has been applied to financial applications such as algorithmic trading and portfolio optimization.

# 4) Natural Language Processing: Actor-Critic has been used for tasks such as text classification and machine translation.

# 5) Healthcare: Actor-Critic has been used in healthcare applications such as disease diagnosis and drug discovery.

# Overall, Actor-Critic is a versatile algorithm that can be applied to a wide range of problems that involve decision-making and optimization.