Continuous Control with Deep Reinforcement Learning

CSE510 – Introduction to Reinforcement Learning

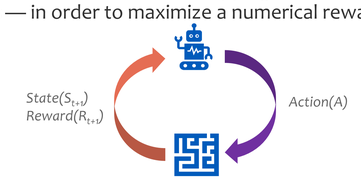
Presented by Vishva Nitin Patel and Leena Manohar Patil under the guidance of Professor Alina Vereshchaka

Introduction

Supervised Learning	Unsupervised Learning	Reinforcement Learning
$y \Rightarrow f(x)$	$f(x)$	$y \Rightarrow f(x), r$
Given: y, x	Given: x	Given: x, r
To Find: f	To Find: f	To Find: f

x→ input, y→ output, f→ function that maps y to x, r→ reward

Reinforcement Learning (RL) is a branch of Machine Learning that deals with methods of mapping the situations/states to actions — in order to maximize a numerical reward.



Markov Decision Process (MDP)

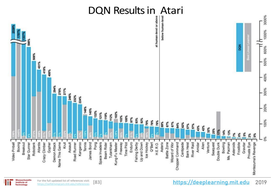
Markov Decision Process is an approach that is only concerned with the next State 'S+1' and the decision of reaching the next state is solely based on the current State at the current time.
Our task is find a policy $\pi : S \rightarrow A$ which our agent will use to take actions in the environment which will maximize the cumulative reward, where $\gamma \in [0, 1]$ is the discounting factor (used to give more weight to more immediate rewards).

Key-points:

- In MDP, *'only the present matters'*
- Components involved in MDP:
  - State(S) – current situation or position of the agent
  - Model(P(S,A,S')) – Transition Probabilities
  - Actions(A) – Things agent can do
  - Rewards(R)

Problem

RL can be used to solve a wide scope of problems, where, the agent has been observed to learn gradually and eventually exceed human control. We can observe below the comparison of numerous Atari game RL is used to solve and it's comparison to human control:



The Primary Challenge in RL

The major challenge in RL is that, we are exposing the agent to an unknown environment where, it doesn't know the consequences of it's actions and also, it has no idea about what state it is in. It just receives the reward when performing an action, and has to make decisions based on them. Hence, it becomes difficult for the agent firstly, to get familiar with the environment and then to optimize it's policy.

Deep - RL Methods

Difference between value and policy:

Value is the estimated total returns for a given state and action pair.

Policy is set of optimal state and action pairs.

We can implement RL Algorithm primarily in two ways:

Value based Learning Approach

1. Find the Q-value for each state-action pair
2. Execute the policy based on the maximum Q value

Policy based Learning Approach

1. Find the optimal policy by calculating the probability distribution, given the state, over all possible actions and then sample the probabilities.

Environments

OpenAI Gym

OpenAI gym provides a comprehensive library of environments which can be used to work with Reinforcement Learning algorithms. It provide a standardize steps for creating and working with any environment.

Following environments were being used:

❖ Pendulum

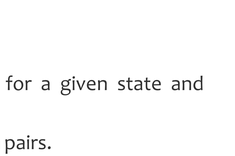
Problem: The pendulum is inverted and randomly swings to any direction.

Goal: To control the pendulum to keep it always in upright direction

❖ Lunar Lander

Problem: The pendulum is inverted and randomly swings to any direction.

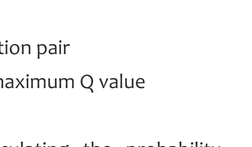
Goal: To control the pendulum to keep it always in upright direction



❖ Bipedal Walker

Problem: This is a robot walker environment learning to move forward.

Goal: To move the robot forward.



Deep Q-Network (DQN)

- Q-Learning, but with Deep Neural Network for function approximation.
- Q-Learning fails because of overestimation of action values. These overestimation result from a positive bias introduced by using the maximum expected action value for approximation.
- In reinforcement learning, both the input and the target change constantly during the process and make training unstable.

DQN overcomes unstable learning by mainly 4 techniques:
- ✓ Experience Replay
- ✓ Target Network
- ✓ Clipping Rewards
- ✓ Skipping Frames

Double Deep Q-Network (DDQN)

We apply the double estimator to Deep Q-learning to construct Double Deep Q-learning (Double DQN), a new off-policy reinforcement learning algorithm.

DDQN uses two separate Q-value estimators, each of which is used to update the other. Using these independent estimators, we can get unbiased Q-value estimates of the actions selected using the opposite estimator.

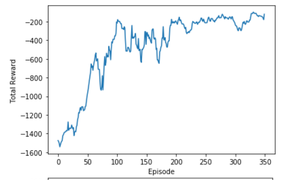
We can thus avoid maximization bias by disentangling our updates from biased estimates.

Deep Deterministic Policy Gradient (DDPG)

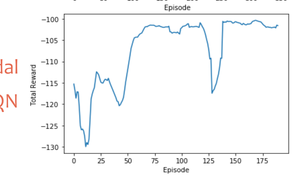
- DDPG is a model-free off-policy actor-critic algorithm, combining Deterministic Policy Gradient (DPG) with Deep Q-Network (DQN).
- DQN stabilizes the learning of Q-function by experience replay and frozen target network.
- DQN works in discrete space, and DDPG extends it to continuous space with the actor-critic framework while learning a deterministic policy.
- DDPG is a policy gradient algorithm that uses a stochastic behavior policy for good exploration but estimates a deterministic target policy.
- It performs policy iteration to evaluate the policy, and then follow the policy gradient to maximize performance.

Reward Graphs

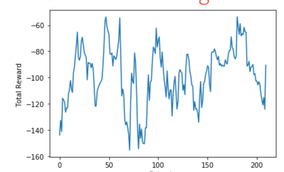
Performance for Pendulum against DDPG

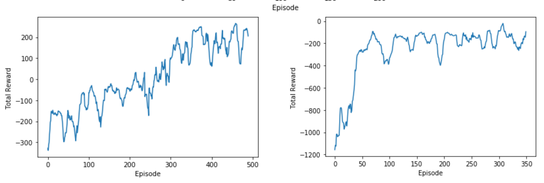


Performance for Bipedal Walker against DDQN



Performance for Lunar Lander against DDQN and DDPG





Conclusion

DDPG on OpenAI gym environments as in Pendulum, Lunar lander and Bipedal walker. It is observed that the agent gradually learns to maximize it's rewards within the span of 300- 350 episodes, by the end of which, it learns to get maximum possible scores. Although, in case of Bipedal Environment, it is observed that the learning is a bit overfitted. This could be adjusted by tuning the hyper-parameters of the neural nets.
When comparing the Lunar Lander environments tested for DDPG, DQN and DDQN Environments, it is quite evident from the graphs that the rate of learning by the agent is comparatively better in DDPG with a continuous action space, rather that the discrete space in case of DQN and DDQN.

References

1. CSE510 Lecture Slides
2. https://gym.openai.com/envs/
3. https://arxiv.org/abs/1509.02971
4. https://www.youtube.com/channel/UC58v9cLitc8VaCjrcKyAbrw
5. https://classroom.udacity.com/courses/ud600

University at Buffalo The State University of New York

Contacts:

leenaman@buffalo.edu

vishvani@buffalo.edu