Implementing Double DQN and Dueling DQN in Pytorch

Your Name

1 Introduction

In this tutorial, we will discuss the implementation of Double Deep Q-Network (Double DQN) and Dueling Deep Q-Network (Dueling DQN) algorithms in reinforcement learning using LaTeX.

2 Double DQN Algorithm

The Double Deep Q-Network (Double DQN) algorithm is an extension of the DQN algorithm that aims to reduce overestimation of Q-values. It achieves this by using two separate networks for action selection and evaluation. Here is the algorithm:

Algorithm 1 Double DQN Algorithm

```
Initialize two Q-networks: $Q$ and $Q'$
Initialize target network weights: $Q' \leftarrow Q$
for each episode do
    Initialize state $s$
    for each step in episode do
        Select action $a$ using $Q$ with $\epsilon$-greedy policy
        Execute action $a$, observe reward $r$ and new state $s'$
        Select action $a'$ using $Q'$ with $\epsilon$-greedy policy
        Update $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q'(s',a') - Q(s,a)]$
        Update state $s \leftarrow s'$
    end for
end for
```

3 Dueling DQN Algorithm

The Dueling Deep Q-Network (Dueling DQN) algorithm is another extension of the DQN algorithm that separates the estimation of state values and action advantages. Here is the algorithm:

Algorithm 2 Dueling DQN Algorithm

```
Initialize two networks: $V$ and $A$
for each episode do
    Initialize state $s$
    for each step in episode do
        Compute $V(s)$ and $A(s, a)$ for all actions $a$
        Compute Q-values: $Q(s, a) = V(s) + (A(s, a) - \text{mean}(A(s, a')))$
        Select action $a$ using $\epsilon$-greedy policy based on $Q$
        Execute action $a$, observe reward $r$ and new state $s'$
        Update $V(s)$ and $A(s, a)$ using TD error
        Update state $s \leftarrow s'$
    end for
end for
```

4 implementation

1. This repo is a PyTorch implementation of Vanilla DQN, Double DQN, and Dueling DQN based off these papers.

2. This tutorial walks you through the fundamentals of Deep Reinforcement Learning. At the end, you will implement an AI-powered Mario (using Double Deep Q-Networks) that can play the game by itself.

3. In this Yputube tutorial we'll learn how to implement dueling double deep q learning in the open ai gym.

5 Conclusion

In this tutorial, we have discussed the implementation of Double DQN and Dueling DQN algorithms in reinforcement learning using LaTeX. These algorithms are extensions of the DQN algorithm and aim to improve learning efficiency and stability.