Assignment 6-2: Cartpole Revisited

Name: Matthew Rix

Date: 02/10/2023

REINFORCE algorithm pseudocode for solving the cartpole problem:

1. Initialize the policy network with random weights
2. Repeat the following steps until the performance is satisfactory:
   1. Reset the environment to its initial state
   2. Until the episode terminates:
      1. Choose an action based on the current policy network, by passing the current state as input and computing the output probabilities for each action
      2. Take the chosen action and observe the next state and reward
      3. Store the state, action, reward in a memory buffer
   3. Compute the discounted returns for each time step using the rewards stored in the memory buffer
   4. Use the policy network to compute the log probabilities of the actions taken in the episode
   5. Compute the policy loss by taking the element-wise product of the log probabilities and the discounted returns, and summing over all time steps
   6. Backpropagate the policy loss through the policy network to update its weights

A2C uses value-based methods and policy-based methods along with having a single neural network to estimate both the policy and the value function (Yoon 2019).

A2C algorithm pseudocode:

1. Initialize the neural network with random weights
2. Initialize the reward R and the episode number t
   1. Repeat (for each episode):
   2. Initialize the state of the Cart Pole
   3. Repeat (for each step in the episode):
      1. Choose an action using the policy network
      2. Take the action and observe the reward and the next state
      3. Update the reward R
      4. Store the experience (state, action, reward, next state) in memory
      5. If the pole has fallen or the episode has reached a maximum number of steps:
         1. Update the network weights using the stored experiences
         2. Reset the reward R and the episode number t

In this pseudocode, the neural network is used to estimate both the policy and the value function. The policy is used to choose actions, and the value function is used to estimate the expected reward. The network weights are updated using the stored experiences at the end of each episode. The process continues until the agent has learned to balance the pole for a sufficient number of steps.

Policy gradient approaches, estimate the value of a policy directly by computing the gradient of the expected cumulative reward with respect to the policy parameters (Hui 2020). The policy is updated by taking a step in the direction of the gradient that increases the expected cumulative reward. This allows the policy to be updated in a more direct manner, without the need to estimate state-values.

Value-based approaches, on the other hand, estimate the value of a policy by using a state-value function (Or 2021), which estimates the expected cumulative reward for a given state following a policy. In Q-learning, this state-value function is updated using the Bellman equation (TORRES 2021), which takes into account the immediate reward, the discounted future reward, and the maximum expected reward for the next state. The policy is then updated by selecting the action with the maximum expected cumulative reward for each state.

Actor-critic approaches are a combination of value-based and policy-based methods in reinforcement learning (Karagiannakos 2018). In an actor-critic approach, two neural networks are used: one network, called the actor network, is used to generate a policy, and the other network, called the critic network, is used to estimate the value function.

In summary, an actor-critic approach combines the strengths of both value-based and policy-based methods to provide a more stable and efficient reinforcement learning algorithm. The actor network is used to generate a policy, and the critic network is used to estimate the value function, providing a target for the policy network to improve upon.

References:

Yoon, C. (2019, July 17). *Understanding actor critic methods*. Medium. Retrieved February 10, 2023, from https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f

Hui, J. (2020, August 27). *RL - policy gradient explained*. Medium. Retrieved February 10, 2023, from https://jonathan-hui.medium.com/rl-policy-gradients-explained-9b13b688b146

Or, B. (2021, January 31). *Value-based methods in deep reinforcement learning*. Medium. Retrieved February 10, 2023, from https://towardsdatascience.com/value-based-methods-in-deep-reinforcement-learning-d40ca1086e1

TORRES.AI, J. (2021, September 24). *The bellman equation*. Medium. Retrieved February 10, 2023, from https://towardsdatascience.com/the-bellman-equation-59258a0d3fa7

Karagiannakos, S. (2018, November 17). *The idea behind actor-critics and how A2c and A3C improve them*. AI Summer. Retrieved February 10, 2023, from https://theaisummer.com/Actor\_critics/