Dominic Drury

CS-370

04/11/2024

6-2 Assignment: Cartpole Revisited

A reinforced algorithm “is a Monte-Carlo variant of policy gradients” where the agent uses the current policy to perform an epoch, and then updates its policy based on the results of the policy (Yoon, 2018). A good flow of the algorithm provided by Chris Yoon in their article Deriving Policy Gradients and Implementing REINFORCE is that the agent performs a trajectory roll-out using the current policy. stores the log probabilities and reward values at each step, gets the discounted value for future rewards at each step, and then updates the policy accordingly before repeating (Yoon, 2018). For the cartpole problem specifically, the agent uses its current policy to predict the best possible move for it to make, then takes and logs the results from that action, next it calculates the discounted future reward to identify its next move, and finally it updates its policy before repeating.

One improvement that can be made over reinforcement learning is using Advantage Actor Critic, or A2C, to increase stability, speed up the convergence, and reduce variance (Yoon, 2019). For solving the cartpole problem I have included the pseudocode provided by Chris Yoon in their article Understanding Actor Critic Methods and A2C which they have adapted from Lilian Weng’s post “Policy Gradient Algorithms (Yoon, 2019):

*Initialize the parameters and learning rates*

*for each episode in range*

*take the current reward sample and calculate the next state based on policy*

*sample the next action based on the results from the previous step*

*update the policy parameters based on the results from the previous step, correcting for the Q-value*

*set current action and reward to the calculated future action and reward*

*repeat until end of range*

The policy gradient approach differs from value-based approaches in that policy gradient calculates the next action based on the maximum cumulative future reward and value-based approaches, like Q-learning, calculates the next action based on what next step provides the best value (Yoon, 2018; Yoon, 2019). Both can be advantageous to the other depending on the model that you would be trying to make depending on if you had proper inputs and resources for a value-based approach, or if you had enough of the variables in the environment known enough to create a policy-based approach. The best of both worlds is the actor-critic model that combines the two. Where value- and policy-based approaches are sperate approaches with their own set of advantages and disadvantages, the actor-critic model is a combination of both (Karagiannakos, 2018).

References

Yoon, C. (2018, December 29). Deriving Policy Gradients and Implementing REINFORCE. Medium.com. Retrieved April 8, 2024, from <https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63>

Yoon, C. (2019, February 5). Understanding Actor Critic Methods and A2C. Medium.com. Retrieved April 8, 2024, from <https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f>

Karagiannakos, S. (2018, November 17). *The idea behind Actor-Critics and how A2C and A3C improve them*. Theaisummer.com. Retrieved April 11, 2024, from <https://theaisummer.com/Actor_critics/>