Stephen Cardone

Professor Balogun

CS-370

February 14, 2021

Comparing Reinforcement Learning Algorithms

The “Cartpole Problem” is a problem where there is a weight held by a single pole above a cart. If the cart does not move, the weight will fall. By moving the cart, we can keep the cart below the weight and keep the weight suspended. In this paper, we will discuss how the Cartpole Problem can be solved by the Reinforce algorithm, as well as an A2C algorithm. We will compare policy gradient algorithms to Q-Learning algorithms, and compare policy Gradient algorithms to actor-critic algorithms.

Firstly, we have the Reinforce algorithm. The Reinforcement algorithm uses a policy gradient methodology. “In essence, policy gradient methods update the probability distribution of actions so that actions with higher expected reward have a higher probability value for an observed state.” (Yoon 1) The policy gradient of the Reinforce algorithm is also known as a “Monte Carlo Policy gradient” this is another way of saying “taking random samples” (Yoon 2) So in essence, we utilize a table to analyze the potential future reward outcomes, then the actions required for an outcome are weighted by a policy gradient. Since the policy gradient favors higher potential future rewards, than lower rewards, the system learns to follow paths with higher rewards. With many, many iterations, the system learns and hones in to be able to successfully solve the problem given to it. This has proved to be a successful algorithm for solving problems including the cartpole problem.

Previously in this course we have discussed Q-Learning algorithms. How are policy gradient algorithms different from Q-Learning? To refresh on Q-learning algorithms, the algorithm creates a table that contains the probable sum of all the rewards for taking a certain action. The actor will then take the action that leads to the highest probable rewarding future. A policy gradient approach, on the other hand applies weights to potential actions and randomly selects based on the weights. Since the policy gradient allows for more randomization by nature, the policy gradient approach proves to be a productive choice in a reinforcement learning algorithm.

An alternative to the Reinforce policy gradient algorithm is that of A2C algorithm. In the A2C algorithm, there is an actor, similar to in Reinforce, but we also introduce the concept of a critic. Like a critic in performing arts, or media, the critic is someone that is not actively making decisions on the actions taken, but its someone who uses experience to give feedback to the actor to inform their decision making. So, in this Actor-critic format, there are in fact two separate models that are being trained, that of the actor, and that of the critic. Fundamentally, the Actor model works much in the same way as the Reinforce algorithm. The Critic, however is different. Rather than take in data about the environment, and outputting an action, the critic will evaluate the action taken by the actor and compute the value function. Essentially, the critic watches the action that the actor takes, and then tells the actor that it made a good decision, or a bad decision.

What are the primary differences when these 2 algorithms are actually used? The Reinforce algorithm is a solid machine learning algorithm that can be very successful. However, there are some issues with the Reinforce algorithm that the A2C actor-critic approach can improve. For, example: “As in the REINFORCE algorithm, we update the policy parameter through Monte Carlo updates (i.e. taking random samples). This introduces in inherent high variability in log probabilities (log of the policy distribution) and cumulative reward values, because each trajectories during training can deviate from each other at great degrees.” (Yoon 2) This can lead to unstable learning that has unnecessary noise and skew that may not be in an optimal direction. The Introduction of the A2C critic supplies the feedback needed to keep the skew on-line with success and prevent the non-optimal skew. Additionally, there is a fundamental issue with the idea of using a policy gradient alone. “another problem with policy gradients occurs trajectories have a cumulative reward of 0. The essence of policy gradient is increasing the probabilities for ‘good’ actions and decreasing those of ‘bad’ actions in the policy distribution; both ‘goods’ and “bad” actions with will not be learned if the cumulative reward is 0.” (Yoon 2) which indicates that the nature of a policy gradient can lead sub optimal learning.

Works Cited

Yoon, C. (2019, May 23). Deriving policy gradients and implementing reinforce. Retrieved February 14, 2021, from https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63

Yoon, C. (2019, July 17). Understanding actor critic methods. Retrieved February 14, 2021, from https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f

Sergios Karagiannakos. (2018, November 16). The idea behind Actor-Critics and HOW A2C AND A3c improve them. Retrieved February 14, 2021, from https://theaisummer.com/Actor\_critics/