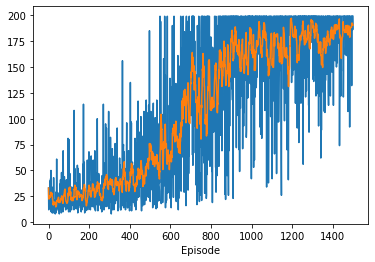
Brian Engel

Module 6 Assignment

Using the REINFORCE algorithm is a good way to solve the cartpole problem. Basically, the REINFORCE algorithm is trial and error. It consists of an agent which is the cart, an environment which is where the cart operates (two dimensional area that the cart can only move right and left in), a goal which is keeping the pole upright, and the policy or strategy that the agent uses that is updated every round to try and maximize the possible rewards. I wanted to see it in action and copied the code from the reading on REINFORCE (Yoon 2018) with a couple of minor changes. I changed the episodes down to 1500 since that is where it looked like the learning leveled off, mainly because it was averaging close to 200 and that was the maximum score. I am including this as Engel\_Brian\_Cartpole\_REINFORCE.ipynb. It seemed like the learning was much smoother than it was with Q learning. There was a steady slope uphill on the minimum and maximum for scores instead of giant differences in Q learning.



I was happy with the results because they look very similar to the chart in the reading (at least the first 1500 episodes). The basic strategy for the REINFORCE algorithm is fairly simple. Actions that lead to higher rewards should have a higher future probability, so the policy is updated to reflect that. The actions with higher rewards get higher probabilities and the actions with lower rewards get lower probabilities, so the outcome should have a steady slope up like the chart above. For stability, it normalizes the updates against an average performance. Once the policy has been updated, it repeats to constantly improves the performance.

Another way to solve this problem is with the A2C algorithm. In this system, there is an actor that receives input (the state of the cart and pole) and then chooses an action based on the policy for the actor. This produces a sequence of state, action, and reward. The critic then uses this to estimate how much the state influences the reward. An advantage is then assigned which is the difference between the estimated rewards and the actual rewards. After this both the actor and the critic policy are updated. The actor is updated to favor actions with a higher advantage, and the critic is updated to better estimate rewards. Then the process repeats. I decided to try out the code in the article pertaining to A2C as well (Yoon 2018). I am including this as Engel\_Brian\_Cartpole\_A2C.ipynb. It seemed like it was much quicker, but way choppier in the chart. I left the code as is and noticed that the gamma values had increased to 0.99 from 0.9 and the number of steps was significantly reduced to 300 from 10000.

A blue and orange lines

Description automatically generatedA blue and orange line

Description automatically generated

Q-learning and other value-based approaches figure out the best actions by estimating the actions values. To do this they predict how good different actions or states are in terms of future rewards, and then choose actions based on the highest estimated values. Policy gradient approaches learn the best actions directly without worrying about estimating values. It changes the strategy to get more rewards over time. This makes policy gradient approaches better at solving more complex problems, since it is more concerned with the big picture as opposed to Q-learning which focuses each step. Actor-critic approaches try to be the best of both worlds. The actor is more of a policy gradient approach and the critic is more concerned with the values of the rewards for the policy. In other words, the actor sets the policy and chooses the action, and the critic tries to predict the value of the action and lets the actor know the result.

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

Yoon, C. (2019, February 5). Understanding Actor Critic Methods and A2C: Important Concepts in Deep Reinforcement Learning. Towards Data Science. https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f