Part 2 - Project report

Årne, Runar and Lars

April 7, 2024

1 Introduction

In this project we will first consider solving a continuing task using function approximation. We will implement linear approximation and neural network approximation for approximating the action-value function $q_\pi(a, s)$, and implementing the Actor-Critic model for directly learning an optimal policy $\pi(a|s)$.

2 Function Approximation and Policy Gradient

We've implemented Q-learning using Linear approximation and Neural Network approximation with the hyperparameters $\epsilon \in [0.1, 0.99]$, $\epsilon_{\text{decay}} = 0.9995$, $\gamma = 0.95$ and $\alpha = 0.01$ running for 5000 episodes.

Linear function approximation works by storing and iteratively updateing a vector of weights, which are dot-producted with a given feature vector (extracted from state and action) to approximate the Q-value $q_\pi(a, s)$.

For neural networks the weight update is similar, however now we update them through backpropagation. Approximating it using neural networks also allow us to approximate non-linear functions, meaning if $q_\pi(a, s)$ we could possibly approximate it much better than using linear approximation.

For this task, cookie disaster, it seems plausible that the Q-value function $q_\pi(a, s)$ is non-linear, as the neural network does a way better job at the task than the linear approximation. Note that the linear is not very bad, it's just not very good either. The same cannot be said for the neural network, as it consistently chooses the correct actions for maximizing the reward! See figure 1.

We then implement the Actor-Critic model with the same hyperparameters.

The actor-critic model works by having two different networks. The actor network, responsible for approximating the policy $\pi(a|s)$, and a critic network, responsible for approximating the value of a state/action.

These two work together in training the actor network. The actor picks the action it thinks is the optimal, and the critic its own evaluation to either criticize or reward the actors choice of action.

When comparing it to the others we see that it does slightly better than the linear approximation, but worse than the neural network approximation. This is because the actor-critic algorithm suffers from the deadly triad. There are several things one could do to improve the learning of this algorithm, such as parallelism, synchronization, target networks, replay buffers, etc. Including any of these would greatly increase the efficiency of this model.

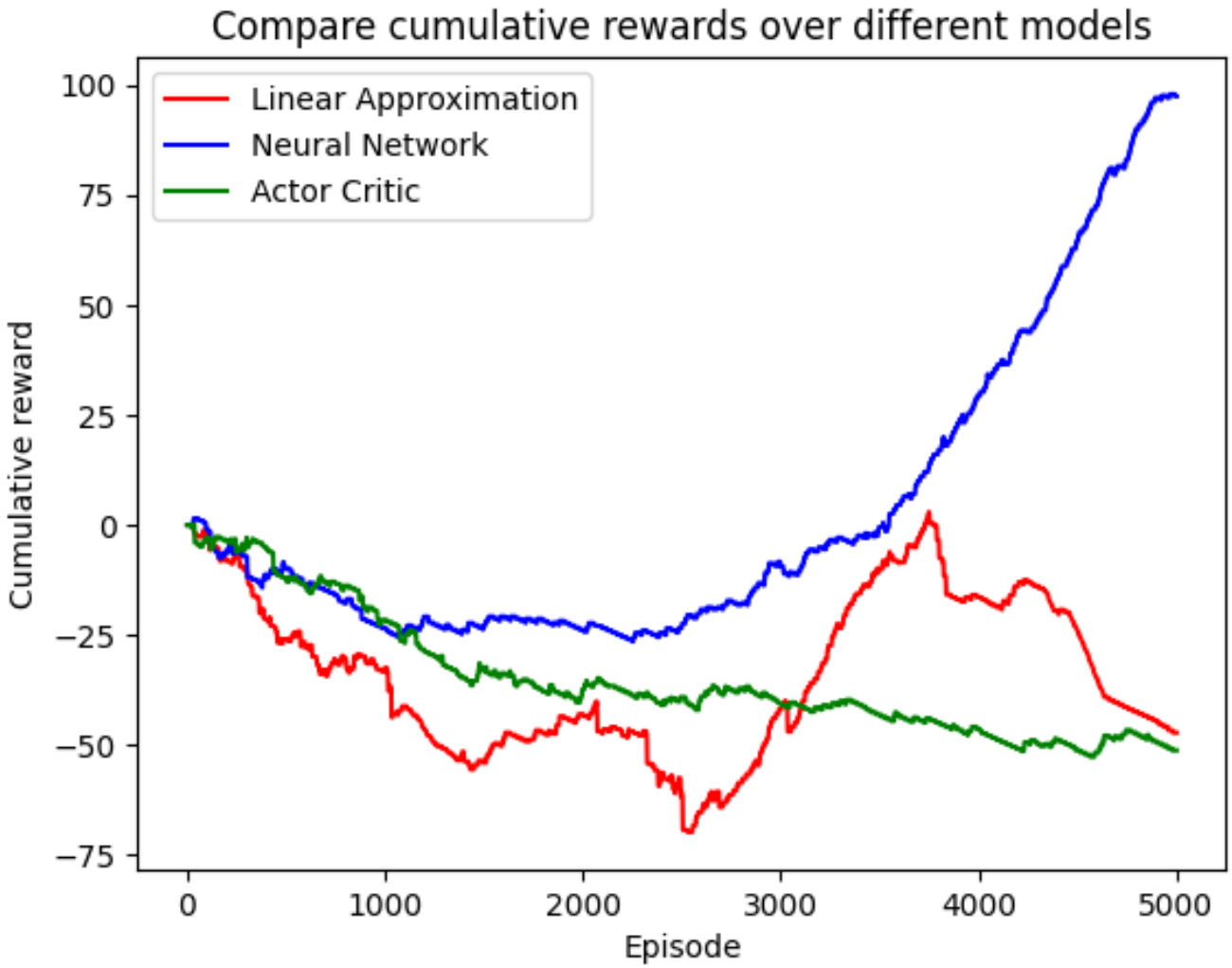

Figure 1: Cumulative reward after different techniques for finding optimal policy $\pi(a|s)$.