Malachi Eberly
Assignment 6: Deep Q-Learning

Architecture

I chose to use a neural network with two hidden layers because I think it sufficiently captures the complexity of the model. 128 neurons was a good number because it gave the neural net enough capacity to understand how the model can be broken down. Too many more neurons or another hidden layer, and it may have led to giving the model too many components.

Parameters

Final Parameters:

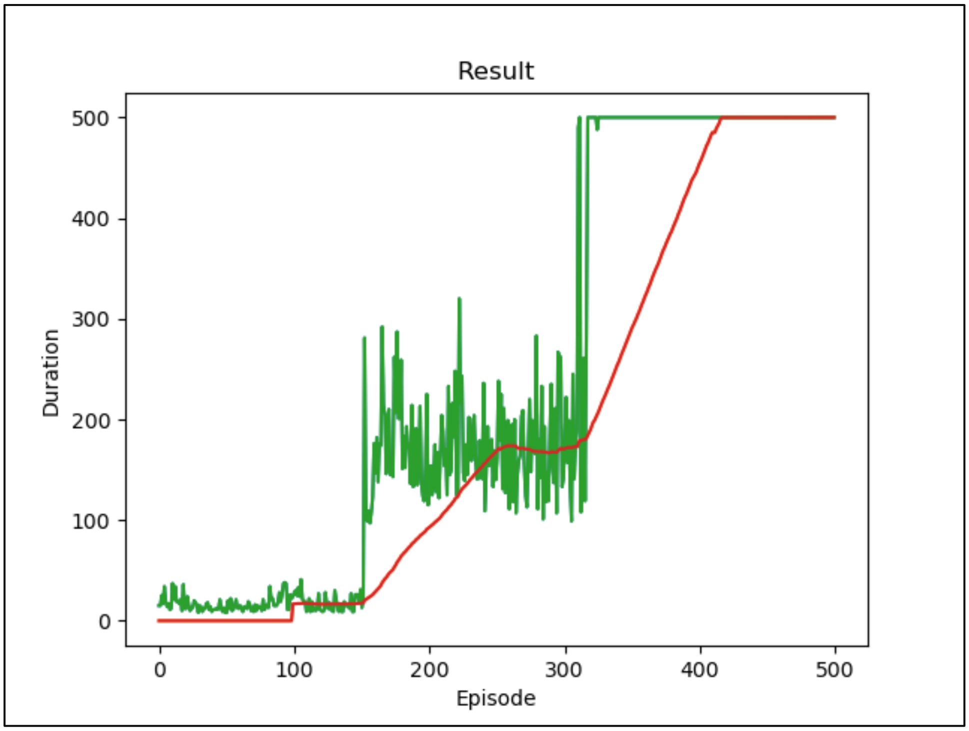
- □ Batch size: 128
- □ Gamma: 0.99
- □ Epsilon start: 0.9
- □ Epsilon end: 0.05
- □ Epsilon decay: 1000
- □ Tau: 0.005
- □ Learning rate: 0.0001

Alternative Parameters:

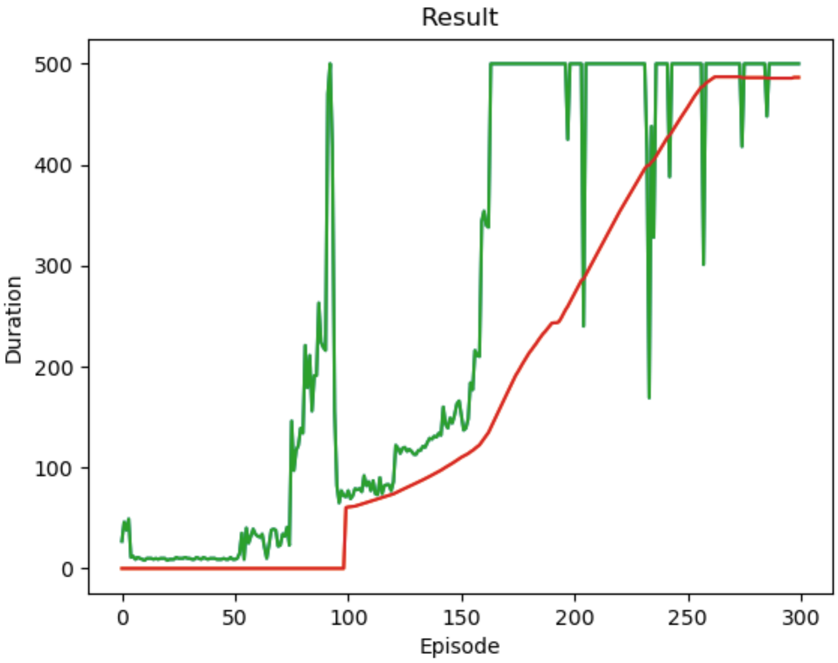
- □ Batch size: 128
- □ Gamma: 0.99
- □ Epsilon start: 0.9
- □ Epsilon end: 0.05
- □ Epsilon decay: 100
- □ Tau: 0.005
- □ Learning rate: 0.001

I chose these values because I think they helped the model train quickly and efficiently. A batch size of 128 made sense to me because it mirrors how many neurons are in the hidden layers. I wanted gamma to be close to 1 because the future rewards are just as important as the current rewards. I decided to have Epsilon start at 0.9 and end at 0.05 because at the beginning I wanted the model to try and learn as many values as possible, but by then end it should be taking optimal actions instead of trying new ones because it's learned what to do. Tau made sense to be something small like 0.005 because the update rate of the target network should be something small. For my final parameters, I used an epsilon decay of 1000 and a learning rate of 0.0001 so that the model could be explored in detail. I decided to also try a decay of 100 and a learning rate of 0.001 to see if I could have the agent learn quicker.

Training Progression Plots



Final parameters – 500 Episodes, average duration 246.284



Alternative Parameters – 300 Episodes, average duration 271.657

Evaluation

With my final parameters, the agent ended up with a stable final duration of 500, which is optimal for the CartPole environment. It took around 150 episodes to reach a duration around 200, and right after episode 300 it found a consistent duration of 500. I decided to try alternative parameters to speed up the learning process. It succeeded in learning quicker but ended up with less stable results. It reached a duration of 500 consistently around episode 150, which is much quicker, but there were many more outliers following that then there were with the final parameters.

Insights and Difficulties

This assignment was helpful for understanding how neural networks can be implemented to solve RL models where the state space is much larger. Most of the difficulties came from learning more about Pytorch's syntax and how changing the hyperparameters and neural network structure affected the model's performance. Having taken CSCI-315 already was a significant help because a lot of the concepts that we covered with deep learning were a review from that course.