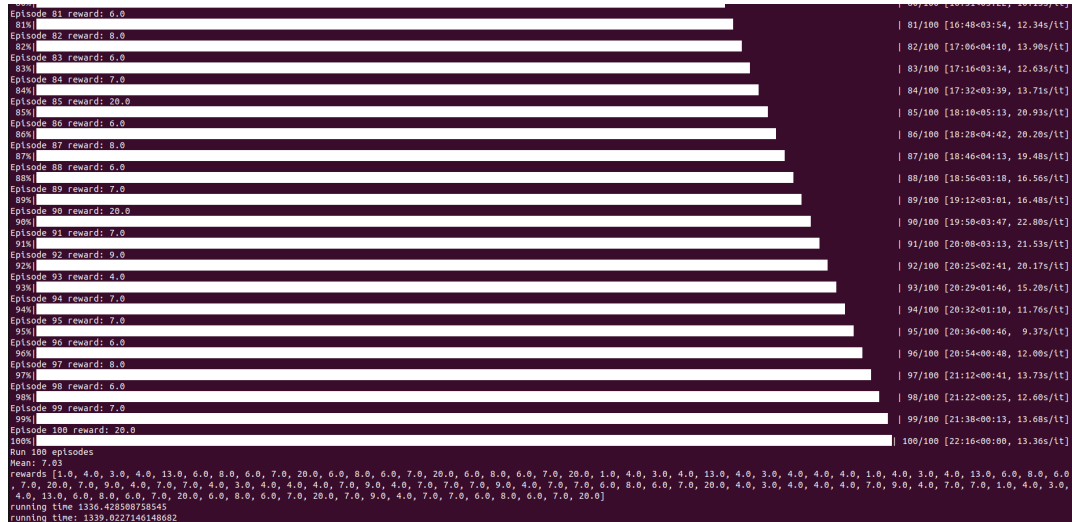


RL Project 3 Report



Results

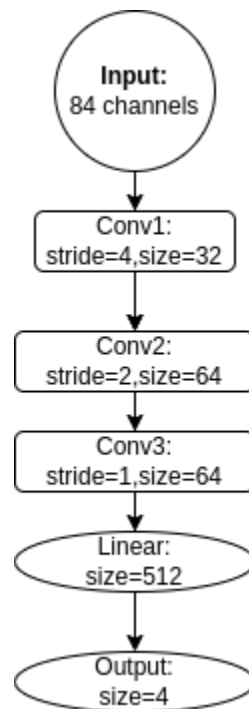
The model I trained gave an average reward of 7.03 for a sample size of 100 episodes. The above is the screenshot of my testing code.

Experiments Performed

I did not achieve the goal reward value using the DQN approach. I have experimented with adding and decreasing the linear layers in the Deep Learning Model and I will present the best performing architecture.

This model is a 5 layered Convolutional + Linear architecture. The Convolutional part is necessary because we are inputting 4 grayscale images as inputs.

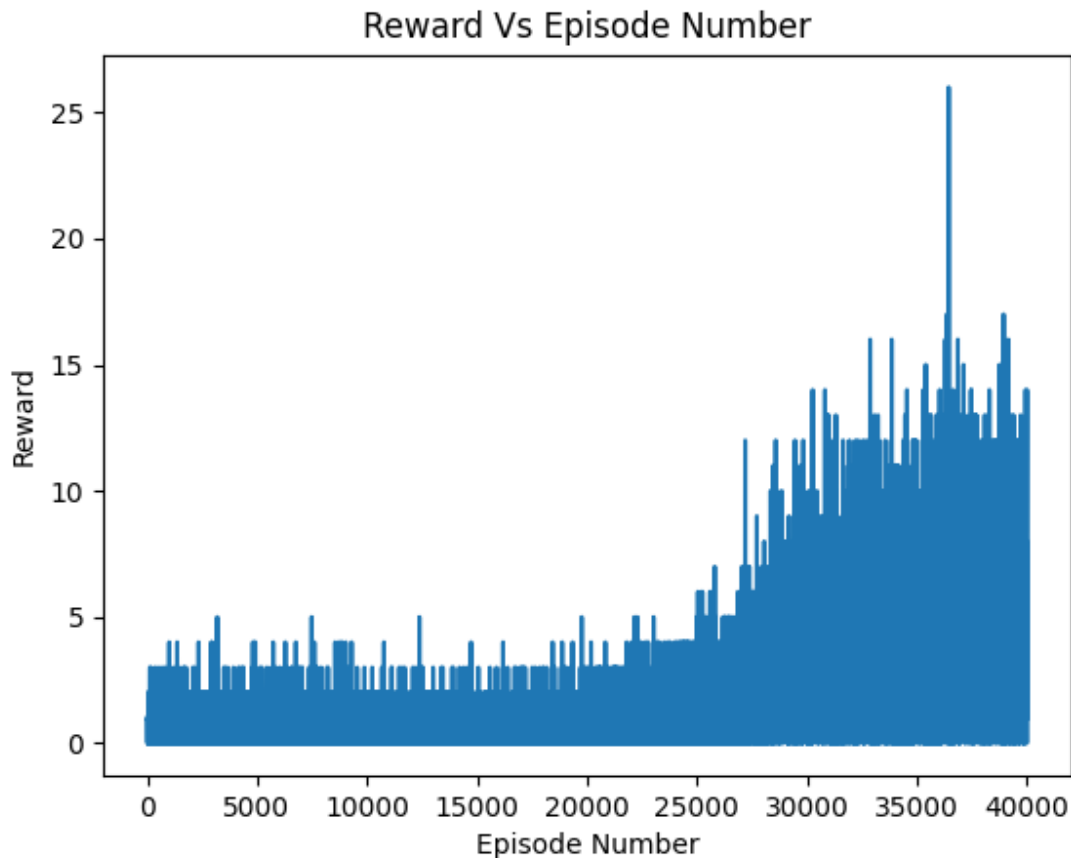
There are 3 Convolutional layers in the network and 2 Linear layers in the network. All except the final layer use the ReLU activation function.



To train this network I set up the environment on the Turing Cluster and trained it there for 40000 iterations.

Variations in Hyperparameters:

- 1) I varied the memory buffer size from 10000 to 5000. This did not lead to any significant reward change. The computation speed was also similar and thus I chose the size 10000 as my size during my final training.
- 2) Increasing the episodes from 10000 to 40000 significantly increased the mean reward I was getting. It changed from 1.34 to 7.03.
- 3) Since convolutional layers extract features from the image, I tried adding one more layer of convolution. This actually reduced my reward and thus I went back to the architecture shown above.



DQN Graph and Future Possibilities

After about 25000 iterations , I saw a jump in the reward. In the future I would like to see how my model does with the recommended 100000 iterations.

I would also like to implement the Duelling DQN model and the Rainbow technique.

This was a particularly interesting and a hard project to understand and implement. This project really helped me understand the nitty gritty of using a DQN model.