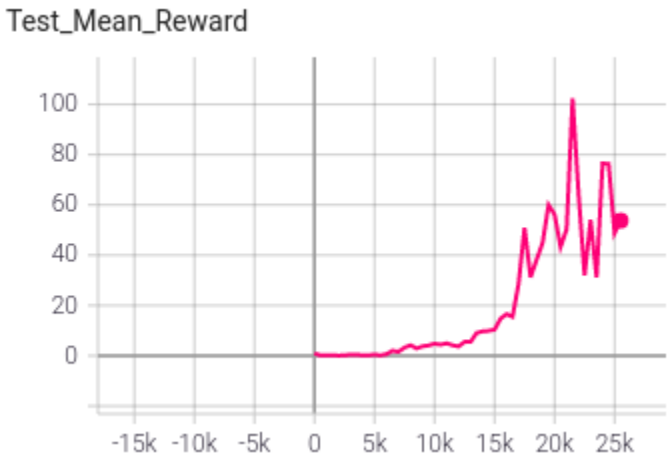
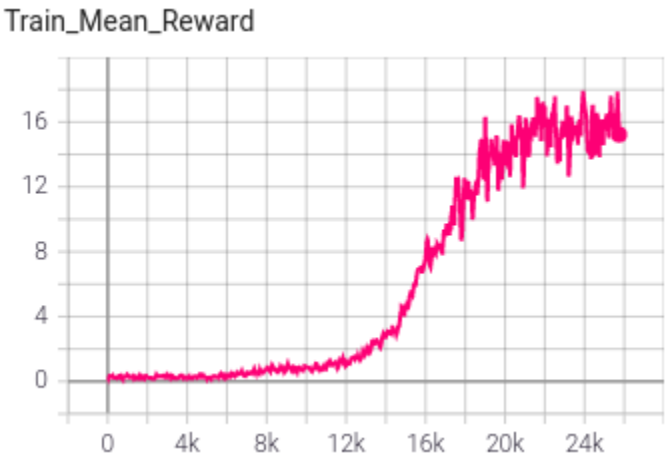
Project 3 : Report

- Sanjeev Kannan

Maximum Test Score achieved : 102.4

Graphs for Test Score and Training Score
(vs number of episodes, approx ~ across 25k episodes)





The rendered video after training can be found within the Project Videos folder in zip file. The video isn't complete, since the agent gets stuck in a loop in certain positions for long periods.

Trained model can be found in the folder : best_weights_model.pt

Set of experiments performed :

Varying network used, decay, hyper parameters, optimizer, memory size, and varying storage from tensors to numpy

In the beginning, I started with using 3 convolutional layers and 1 linear. Towards the end, better results were obtained on increasing the number of linear layers.

I also started out with an exponential Decay system, and later switched to linear decay.

I was updating my target model every 5,000 episodes initially, finally settled on updating every 10,000 steps.

Optimizers used : RMSProp and Adam

Memory size : Started with storing 500,000 images. That consumed more than 12 GB and consumed the GPU's RAM, finally chose to go with a mere 50,000.

A lot of conversions from Tensors to Numpy arrays since they were faster for performing operations and converted them to Tensors when uploading to the GPU.

Final Architecture

3 Convolutional Layers + 2 Linear Layers
Activation Function : ReLU

HyperParameters

Batch Size : 32
Learning Rate : 1e-4
Target Update : Every 10,000 steps
Epsilon Start : 1.00
Epsilon End : 0.01
Epsilon Decay : 1e6
Decay Type : Linear
Activation : ReLU

Loss Function : Huber Loss
Optimizer : Adam