CS525, Assignment 3

Hanshen Yu

Worcester Polytechnic Institute

Student ID: 664892122

Individual Assignment

CS525, Assignment 3

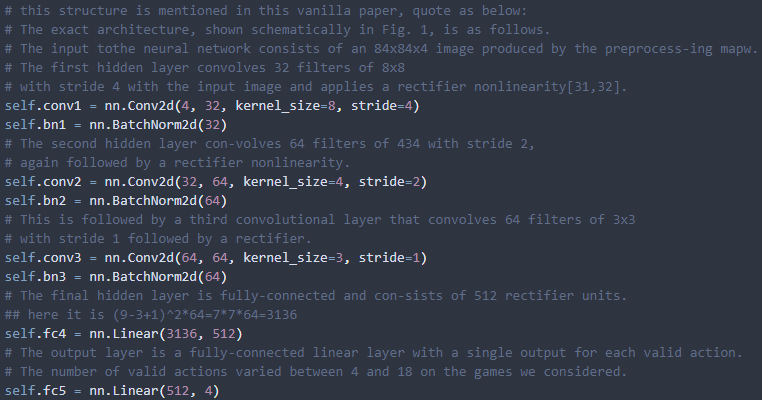
I followed the instructions of in class instructions, pytorch DQN tutorial, paper offered[1] and other online resources[2][3]. I found a blog interesting[4][5], it mentioned how to tune the params to get a similar result from DeepMind breakout results of over 400.

**Include a section describing the set of experiments that you performed**

I trained an agent with Double DQN, with a CNN to learn how to play Atari breakthrough, with the suggested GCP settings.

At first I tried vanilla DQN with all suggested params, but it only gives me around 16 mean reward in test. I later added some improvements, and implemented double DQN to make the performance better, yet still not changing the params. I got a mean reward of 73 in 11.5k episodes and 5 hrs of training time. The training is still going on.

**what structures you experimented with (i.e., number of layers, number of neurons in each layer)**

I strictly followed the suggested paper to set up the structure of neural network, the calculation process is mentioned in the comments. 

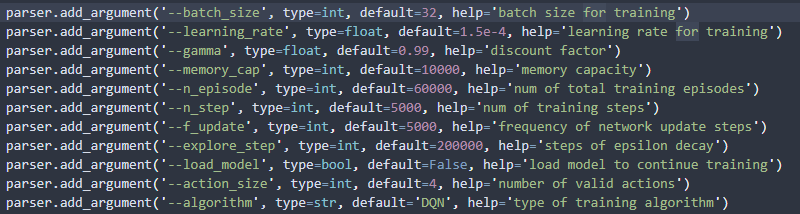
**what hyperparameters you varied (e.g., number of epochs of training, batch size and any other parameter values, weight initialization schema, activation function)**

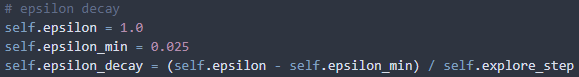
For training, I made a loading mode so that the training could start from where it paused. Fortunately, it helped when I was accidentally disconnected. For this reason, I got more than 1 sessions in training, the graph is split but nothing else is ruined.

During testing, I find that the agent sometimes get stuck in a loop. It might be because the score is so high and it cannot hit a brick, or some bugs in the environment.

I was using the Double DQN algorithm while training. It was performing better than the vanilla DQN in both training time and training results.

My model is trained under these parameters, except for the algorithm. Due to this, I am actually using fairly a large learning rate, a small memory and a short exploring time. I could have a better model if I could change them, but unfortunately the time is limited so I have to accept that.





The activation function is ReLU, with Kaiming He initial function[3]

**what kind of loss function you used and what kind of optimizer you used.**

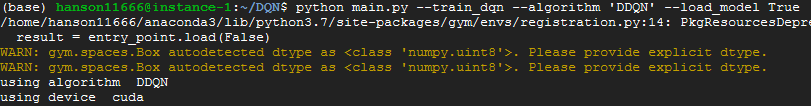
Adam optimizer (learning rate 1.5e-4), huber loss(nn. Smooth-l1-loss)

**Special skills: Include the skills which can improve the generation quality. Here are some [tips](https://arxiv.org/pdf/1710.02298.pdf) may help. (Optional)**

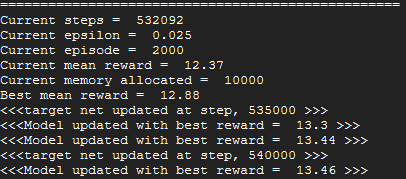
* Using double DQN
* Normalizing the input states by dividing 255
* Changed the initializer for nn to the one mentioned in paper[3].
* I saved an additional flag indicating that the agent lost a life in memory, although it won’t help for this training since it has only one life, it still might help the agent to understand the importance of keeping a life.

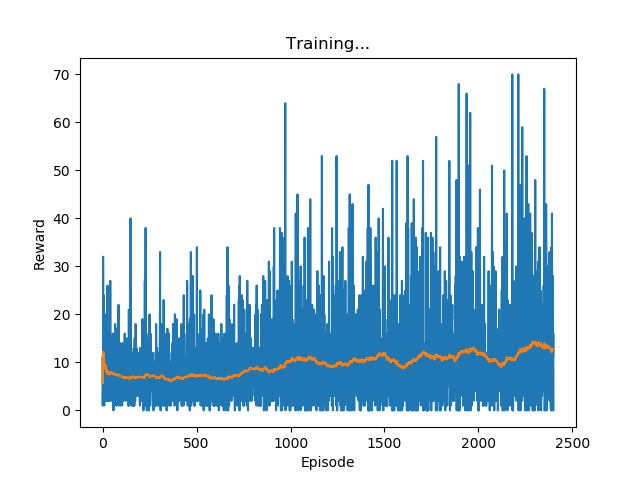
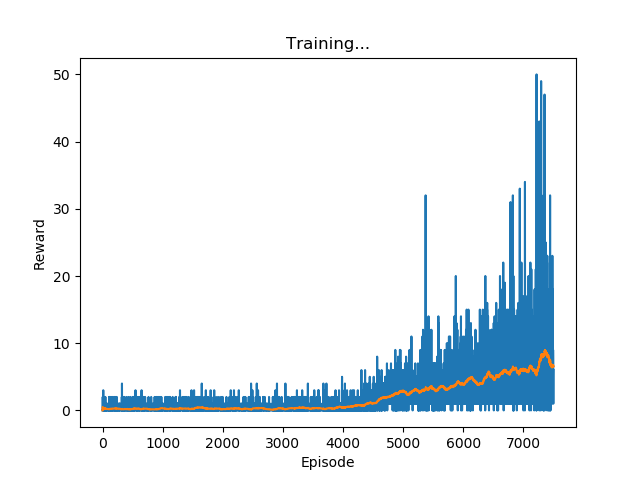
**Visualization: Learning curve of DQN.**

This is the screenshot for session 2, since I have to move my computer around and pause training.

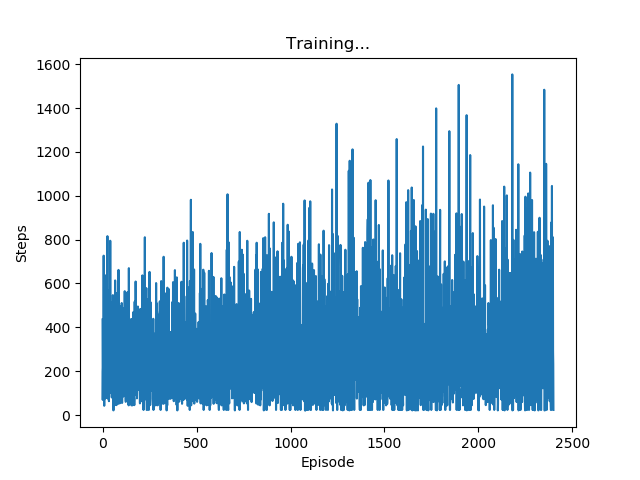
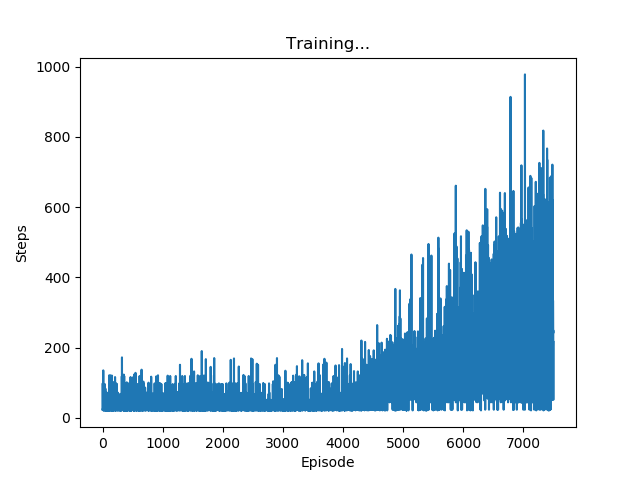


After 3hrs of online training (session 1 and 2 combined), with training reward 13.46,



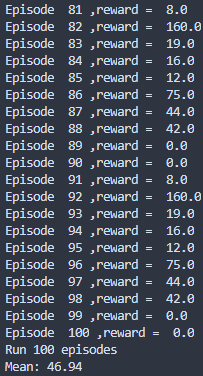
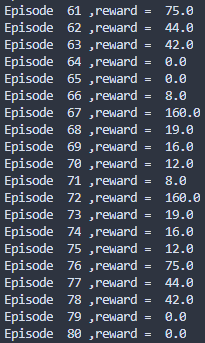
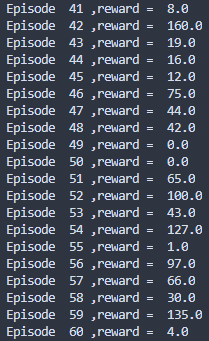
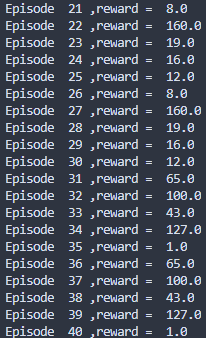
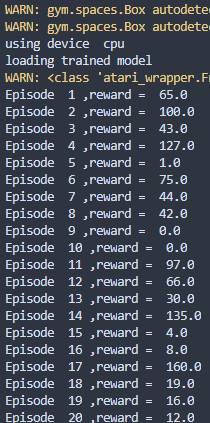


10k episodes with 7500 under session 1 and 2000 under session 2 (which is undergoing)



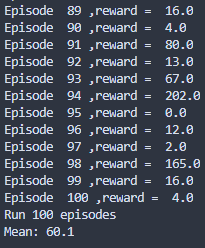
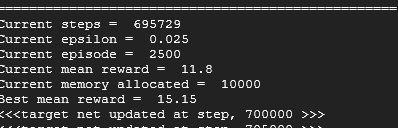
Approximately 2m total steps (sum of session 1 and 2)

And the result is Mean: 46.94, reaches the goal already.

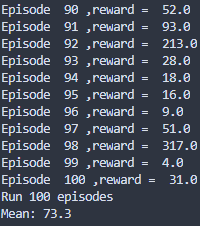
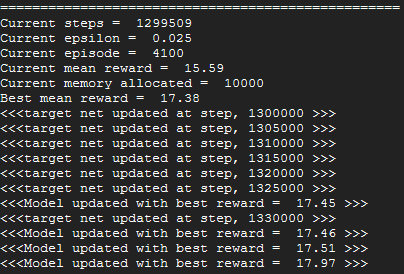


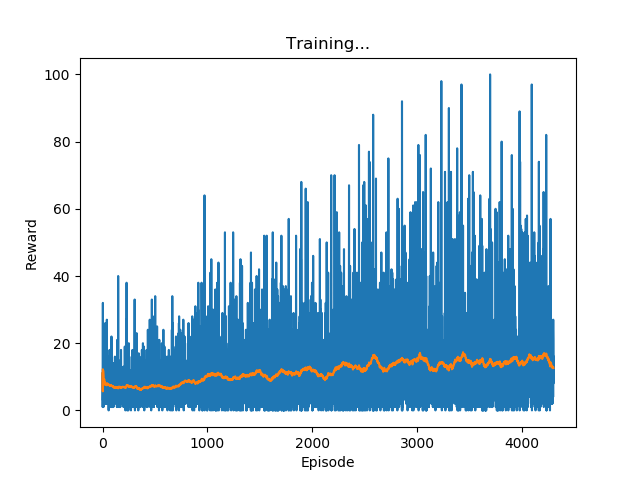
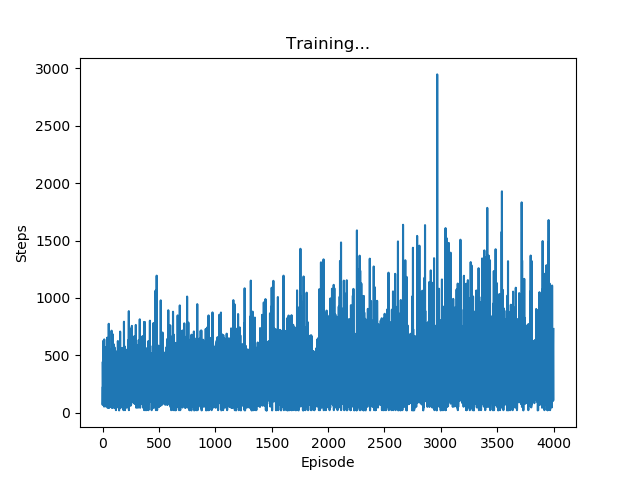
After that I am constantly testing my best model:

3.5hrs, 10k episodes, 15.15 max reward, 2.2m steps with mean 60.1



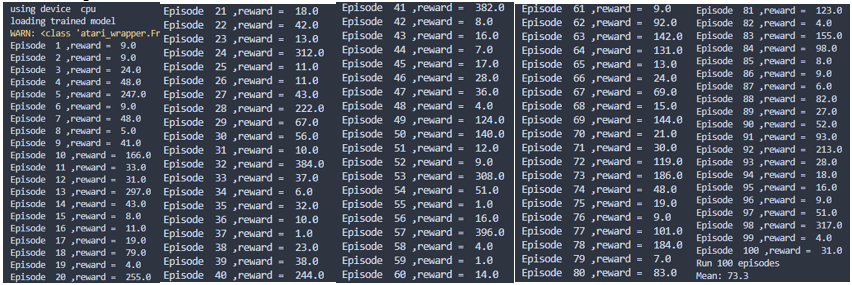
5hrs, 11.5k episodes, 17.97 max reward, 2.8m steps with mean 73.3





Steps and rewards for session 2

To sum up, until the end of session 2, I reached the mean of 73.3.



Reference:

[1] V. Mnih, Playing Atari with Deep Reinforcement Learning

[2] Z. Wang, Dueling Network Architectures for Deep Reinforcement Learning

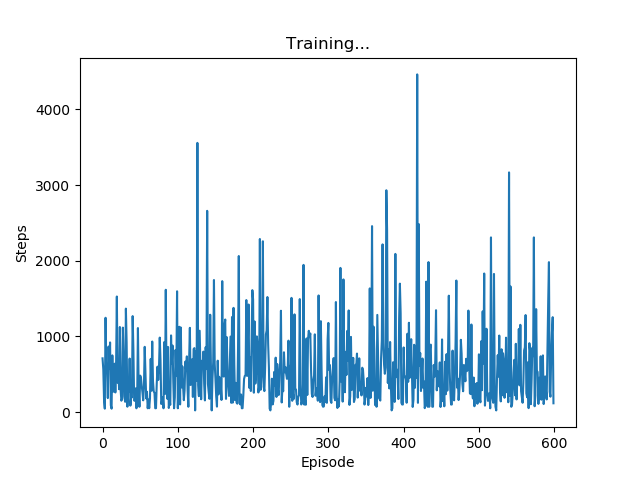
[3] K. He, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

[4] <https://towardsdatascience.com/tutorial-double-deep-q-learning-with-dueling-network-architectures-4c1b3fb7f756>

[5] <https://github.com/dennybritz/reinforcement-learning/issues/30>

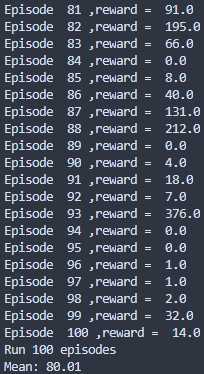
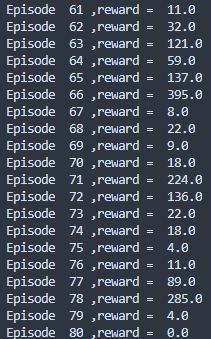
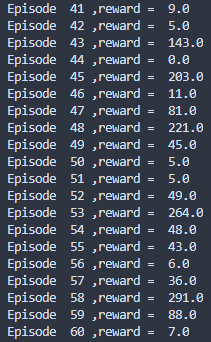
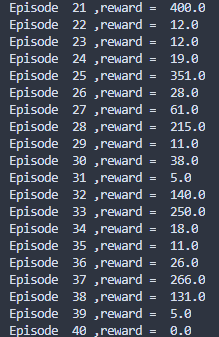
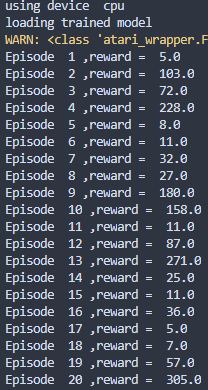
After submission:

Session 3, testing a smaller learning rate lr=1e-5 and slightly a larger memory 30000:

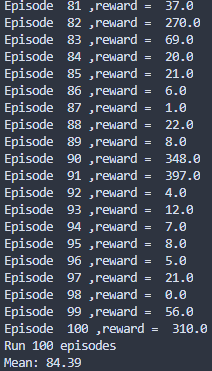
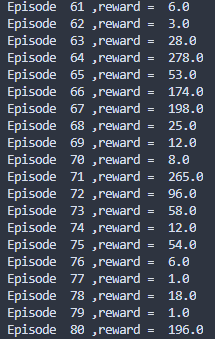
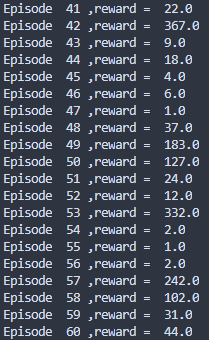
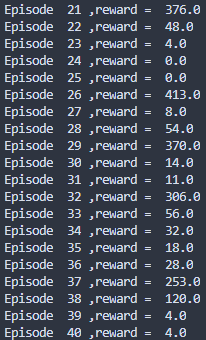
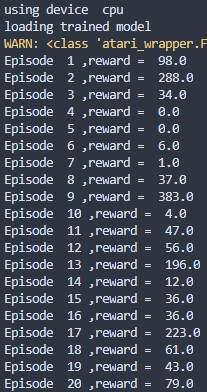


Unfortunately, rewards graph for session 3 is broken. It’s peak is 21.75 and usually around 19.

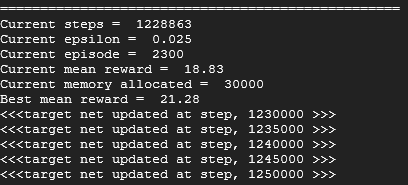
0.5hrs, 0.2k episodes, 100k steps, 21.36 training rewards, mean reward 80.01

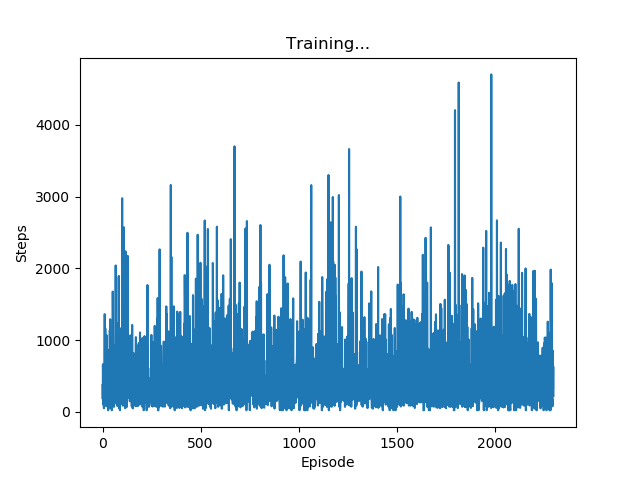
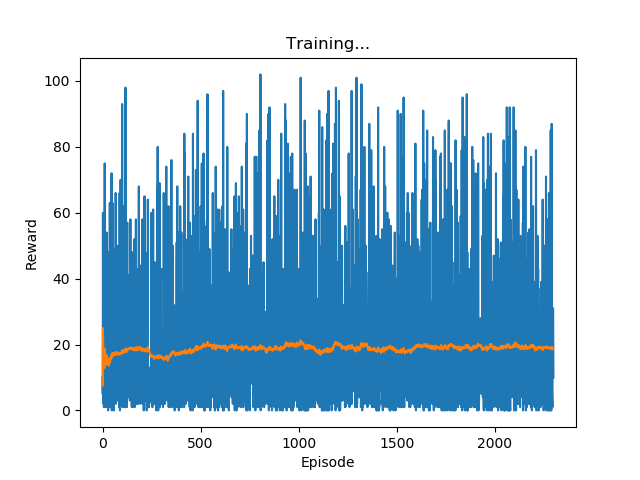


1.5hrs, 0.6k episodes, 300k steps, 21.75 training rewards, mean reward 84.39



Session 4 2hrs, 2.3k episodes, 1.25m steps, could not get a better reward than last session.





Session 5

