Project #1 Report

Name: Patrick H. Rupp

Date: 2023/09/08

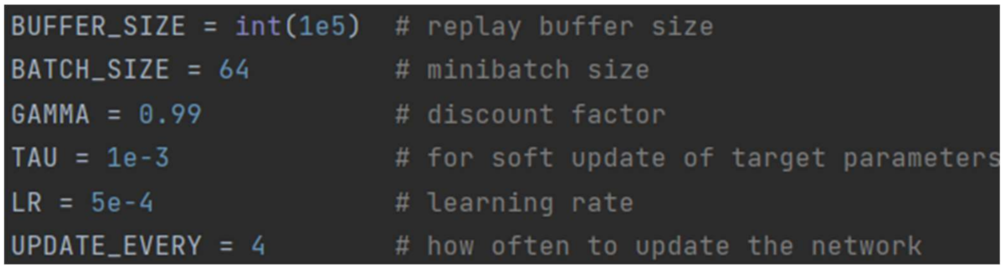
Course: Udacity Deep Reinforcement Learning

Repo: https://github.com/PHRupp/udacity-agent-is-bananas

Learning Algorithm

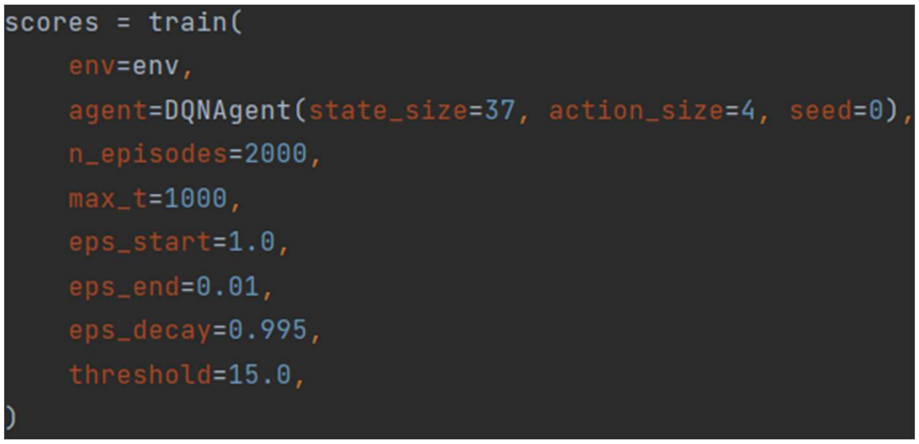
I implemented a Deep Q-Learning (DQN) algorithm using the Torch framework. This neural network is a simple forward pass network with two hidden layers, each using the RELU activation function. I included a memory buffer with 10,000 instances where a batch of 64 is sampled. Additionally, I focused on using a large discount factor to focus on maximizing the value of future rewards. I set a maximum of 2,000 episodes but we reached threshold at 509 and target at 718.

dqn_agent.py



```
BUFFER_SIZE = int(1e5)  # replay buffer size
BATCH_SIZE = 64         # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR = 5e-4               # learning rate
UPDATE_EVERY = 4        # how often to update the network
```

main.py



```
scores = train(
    env=env,
    agent=DQNAgent(state_size=37, action_size=4, seed=0),
    n_episodes=2000,
    max_t=1000,
    eps_start=1.0,
    eps_end=0.01,
    eps_decay=0.995,
    threshold=15.0,
)
```

Plot of Rewards

The results of the agent training is shown below with the score of the last 100 episodes. These results grow fairly linearly from episode 100 until about episode 300 where it begins to plateau slightly with more varying results. The first 100 episodes were slow to learn because of it focusing mostly on random actions to build enough data/experience to learn from. I had set the requirement for average score to be >= 15.00 over a window size of 100 episodes. The plot below shows the actuals scores of each episode and that it consistently grows above the 13.00 threshold starting from episode 509.

Project #1 Report

Log Output:

Results were captured in the excel document saved within the repo.

```
…
Episode 713     Average Score: 14.83
Episode 714     Average Score: 14.87
Episode 715     Average Score: 14.87
Episode 716     Average Score: 14.85
Episode 717     Average Score: 14.86
Episode 718     Average Score: 15.00
INFO:TEST:
Environment solved in 618 episodes!     Average Score: 15.00
INFO:TEST:Exiting...
```

Rewards Plot

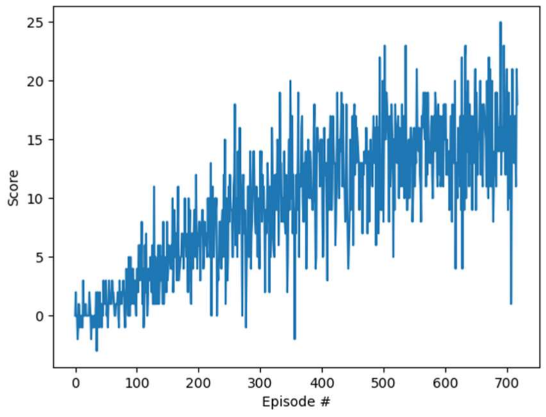


Figure I: Raw scores from each episode until training complete

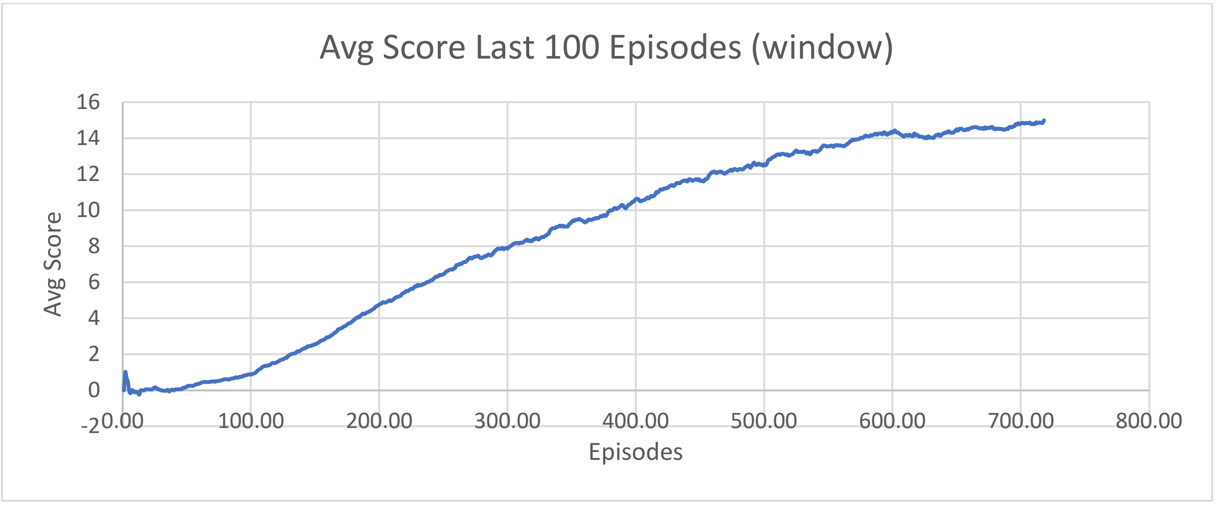


Figure II: Avg Score Over last 100 episodes

Ideas for Future Work

One thing that could make this approach significantly better is using an optimization technique for hyperparameter tuning. For this project, I started with some generic numbers and slightly tweaked by hand. This is inefficient and prone to sub-optimal performance. An optimization technique like particle swarm optimization (PSO) on the hyper parameters to algorithmically search for an optimal hyperparameter set which maximizes the score within a given time period or same score within smallest training cycles. Another method might be to use Ray-tune with its distributed processing capabilities to accelerate results maximizing the compute resources available.