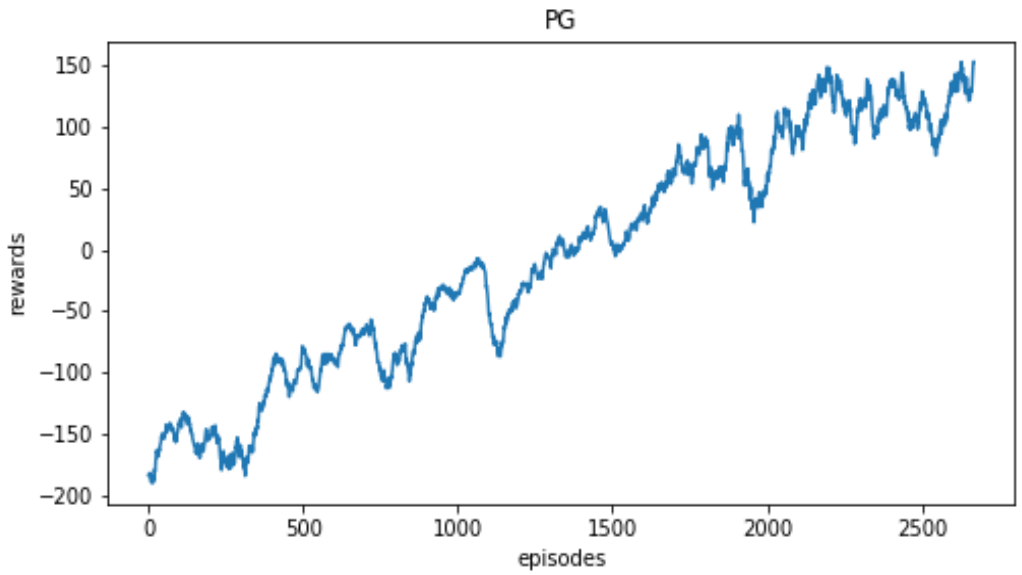
ADL HW3

生醫電資所碩一 R07945029 王思敏

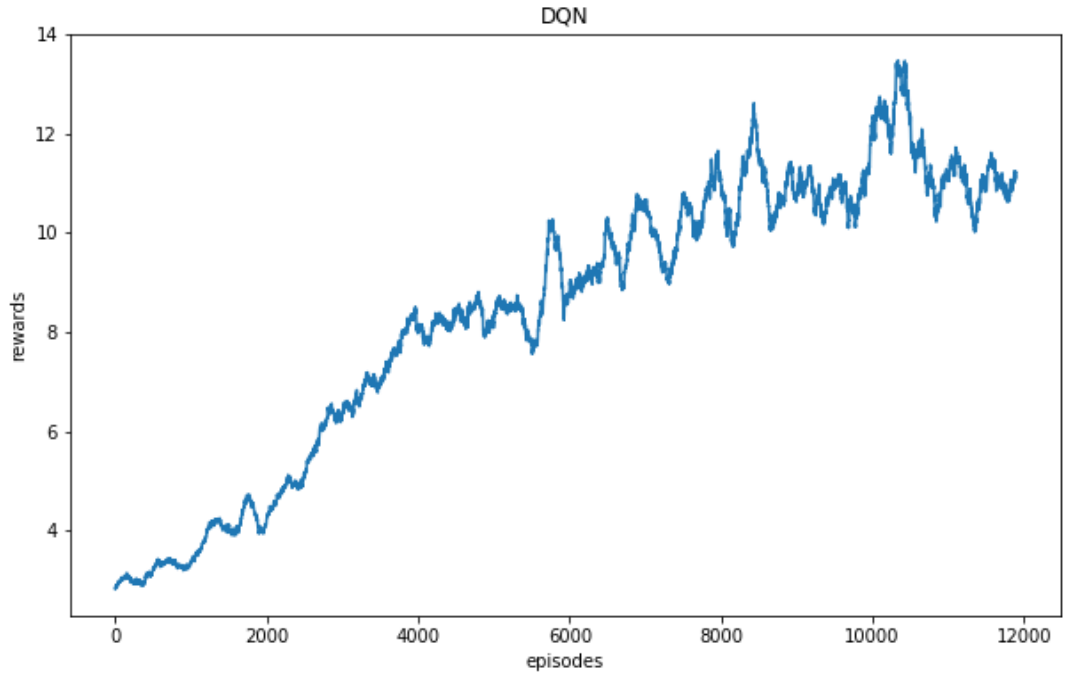
1. Basic Performance
a. Describe your Policy Gradient & DQN model
b. Plot the learning curve to show the performance of your Policy Gradient on LunarLander
c. Plot the learning curve to show the performance of your DQN on Assualt

The policy gradient model consists of 2 linear layers with hidden size 64 which is activated by ReLU layer and with hidden size equals to environment action space size separately. Softmax will be used on output to normalize the probability distribution to 1. The parameters of policy gradient model are set to: learning rate = 1e-3, gamma = 0.99, optimizer=adam, and training epoch = 10000.
The DQN model consists of 3 convolution layers with hidden size 32, 64, and 64 and stride 4, 2, and 1 which are also activated by ReLU separately and 2 linear layers with hidden size 512 and environment action space size. The former is activated by ReLU. The parameters of DQN model are set to: learning rate=1e-4, gamma = 0.99, optimizer=RMSProp, memory capacity=10000, maximum training epoch=3000000.
X-axis: episodes
Y-axis: average reward in last 50 episodes on LunarLander / average reward in last 200 episodes on LunarLander

The test result is 148 on pg.



The test result is 180 on DQN.



2. Experimenting with DQN hyperparameters

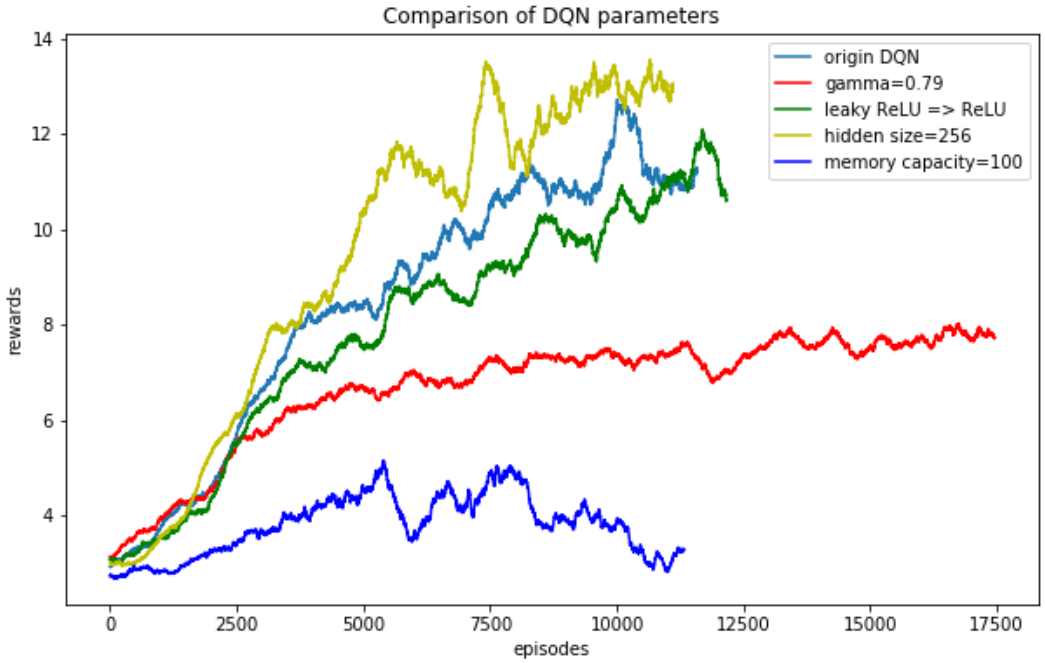
a. Choose one hyperparameter of your choice and run at least three other settings of this hyperparameter

b. Plot all four learning curves in the same figure

c. Explain why you choose this hyperparameter and how it affect the results

The chosen hyperparameters are gamma, activated function, hidden size, and memory capacity. Lower hidden size shows positive effect on DQN model. The reason may be that original hidden size make the model contains unnecessary information and hidden size of 256 is more appropriate for the model. The ReLU layer seems similar to original DQN on training while getting high score on testing. ReLU will not keep some negative information as leaky ReLU, thus may cause some neurons dead during training while the characteristic make the testing more informative. On the other hand, lower gamma, and lower replace memory show negative effect on the model. Lower gamma will make the model more concentrate on the current situation than the future and may cause the outcome not as well as the higher. When replay memory is too small, then the learning starts to oscillate - once it learns to play well, the replay memory is dominated by middle gameplay experiences and it forgets how to play the beginning.

The results are 180, 161, 416, 376, and 57 on origin, gamma=0.79, leaky ReLU → ReLU, hidden size=256, and memory capacity=100.



3. Improvements to Policy Gradient & DQN / Other RL methods

Choose two improvements to PG & DQN or other RL methods.

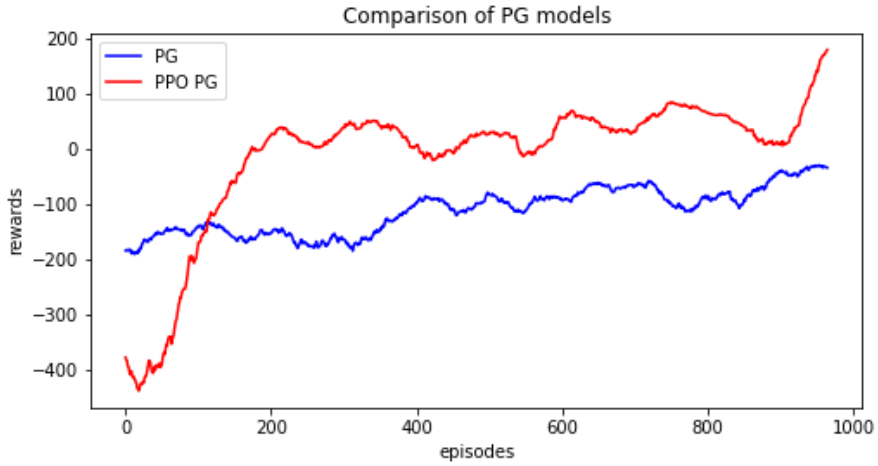
a. Describe why they can improve the performance

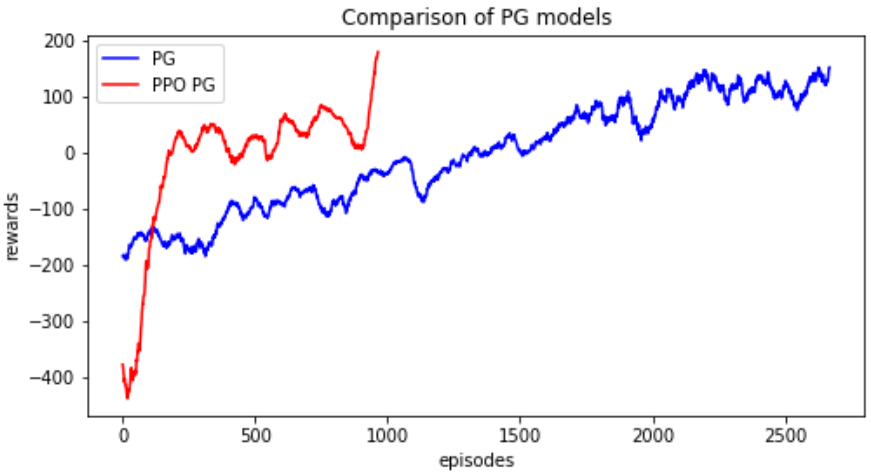
b. Plot the graph to compare results with and without improvement

The chosen improvement for policy gradient is proximal policy optimization (PPO), and the improvements for DQN are dueling DQN and double DQN.

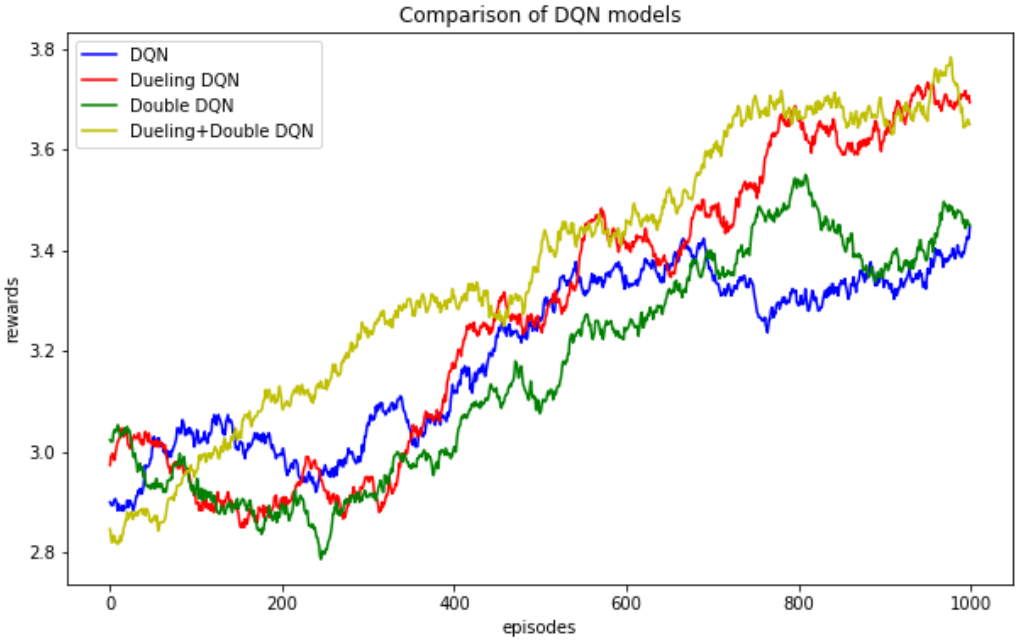
Policy gradient is strongly influenced by step size because the new policy is probably significantly different from the old policy. PPO utilize the clip function and restrict the KL divergence in a range to realize the small batch updating and avoid the problem above. In DQN model, the action value may be overestimated and make the best strategy hard to be found. In double DQN, the max action value is first found in online net. Then, the action value will be calculated in target net based on the previous max action value. In dueling DQN, the extracted feature from the convolution layers will diverse to 2 linear layers which are state function and advantage function and represent the value itself and the extra value separately. The final action value will be calculated by outputs from the 2 function to avoid the overestimating problem.

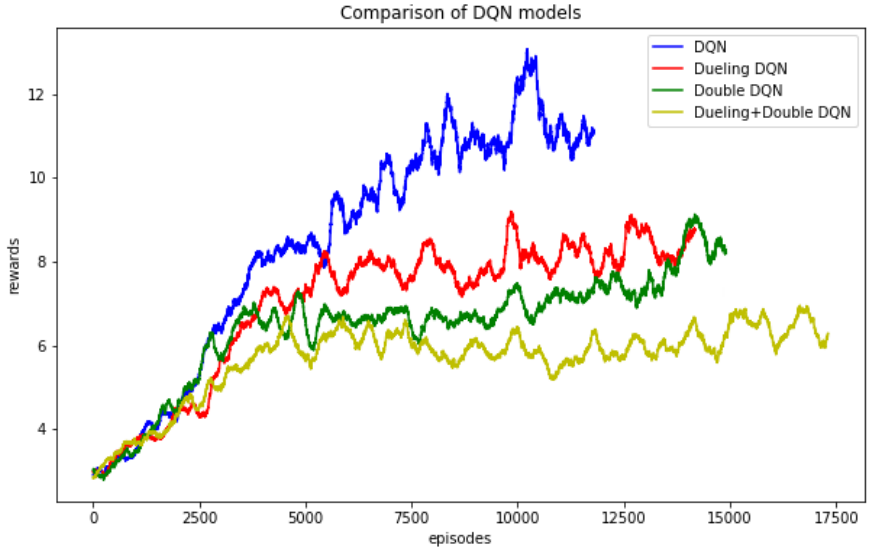
The results are 148 and 206 on origin PG and PPO PG.





The results are 180, 191, 208, and 156 on origin DQN, dueling DQN, double DQN, and dueling+double DQN.





4. Train on SuperMarioBros

a. Describe the RL algorithm you used

b. Plot the learning curve to show the performance

The used RL algorithm is A2C. The model consists of 3 convolution layers with 32, 64, and 32 hidden size and stride 4, 2, and 1 which are activated by ReLU layers. After flattening, the output will be input a linear layer with hidden size 512 activated by ReLU and a GRU layer with hidden size 512. The output will eventually input the actor and critic models which are linear layers with output size of environment action space and 1 separately. The probability distribution output from actor will be normalized by softmax.

The test result is 4331 on A2C.