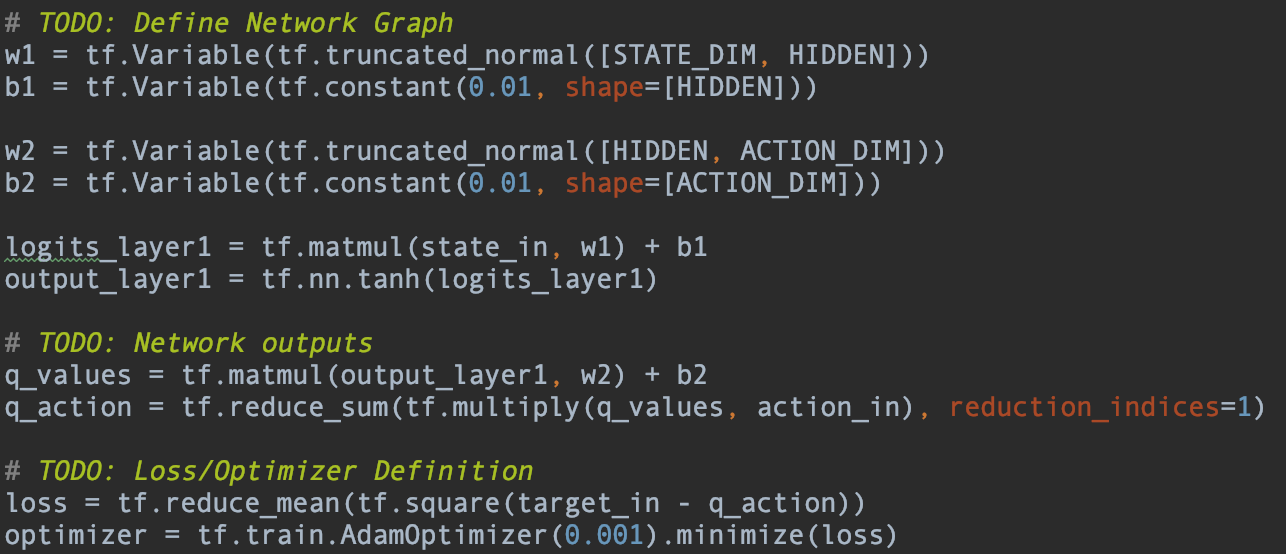
**COMP9444 – Assignment 3**

**Tianwei Zhu, z5140081; Haoxiang Zhao, z5084093**

We used double Deep Q-Learning Network (DQN) to fit the model.



On each step, we put “state”, “action”, “reward”, “next\_state” and “done” into batch and use the batch to get the target q-value. Once the train is “done”, then the target value will be “reward” (1.0) itself:

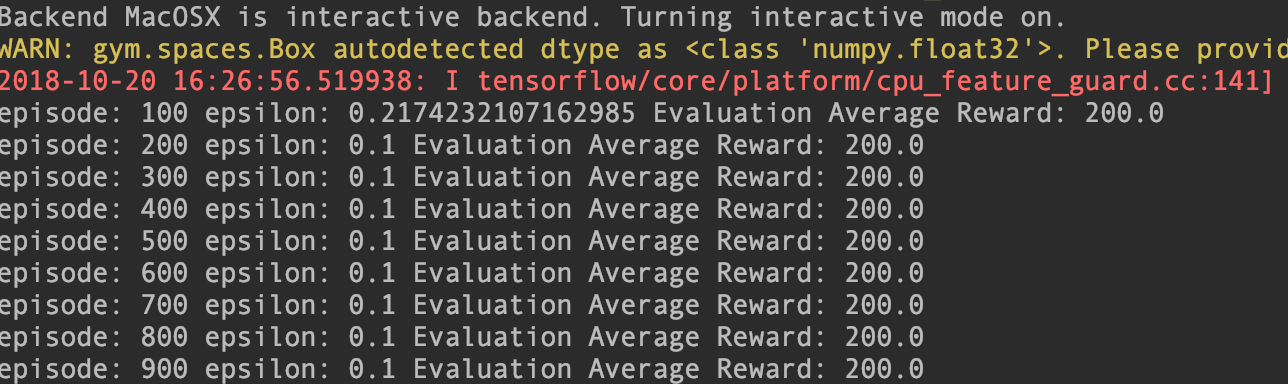
target\_batch.append(reward\_batch[i])

Otherwise target value will be calculated by Bellman algorithm:

target\_batch.append(reward\_batch[i] + GAMMA \* np.max(nextstate\_q\_values[i]))

To keep updating batch and abandon outdated misleading data, we drop previous batch if the batch size is over 10000.

Here is the result of our model:



**Hyperparameters:**

GAMMA = 0.9 # discount factor

INITIAL\_EPSILON = 0.6 # starting value of epsilon

FINAL\_EPSILON = 0.1 # final value of epsilon

EPSILON\_DECAY\_STEPS = 100 # decay period

HIDDEN = 30 # hidden layer of double DQN

BATCH\_SIZE = 128 # each step’s batch size