COMP9444 – Assignment 3
Tianwei Zhu, z5140081; Haoxiang Zhao, z5084093

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

```
# TODO: Define Network Graph
w1 = tf.Variable(tf.truncated_normal([STATE_DIM, HIDDEN]))
b1 = tf.Variable(tf.constant(0.01, shape=[HIDDEN]))

w2 = tf.Variable(tf.truncated_normal([HIDDEN, ACTION_DIM]))
b2 = tf.Variable(tf.constant(0.01, shape=[ACTION_DIM]))

logits_layer1 = tf.matmul(state_in, w1) + b1
output_layer1 = tf.nn.tanh(logits_layer1)

# TODO: Network outputs
q_values = tf.matmul(output_layer1, w2) + b2
q_action = tf.reduce_sum(tf.multiply(q_values, action_in), reduction_indices=1)

# TODO: Loss/Optimizer Definition
loss = tf.reduce_mean(tf.square(target_in - q_action))
optimizer = tf.train.AdamOptimizer(0.001).minimize(loss)
```

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To keep updating batch and abandon outdated misleading data, we drop previous batch if the batch size is over 10000.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

```
Backend MacOSX is interactive backend. Turning interactive mode on.
WARN: gym.spaces.Box autodetected dtype as <class 'numpy.float32'>. Please provid
2018-10-20 16:26:56.519938: I tensorflow/core/platform/cpu_feature_guard.cc:141]
episode: 100 epsilon: 0.2174232107162985 Evaluation Average Reward: 200.0
episode: 200 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 300 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 400 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 500 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 600 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 700 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 800 epsilon: 0.1 Evaluation Average Reward: 200.0
episode: 900 epsilon: 0.1 Evaluation Average Reward: 200.0
```

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