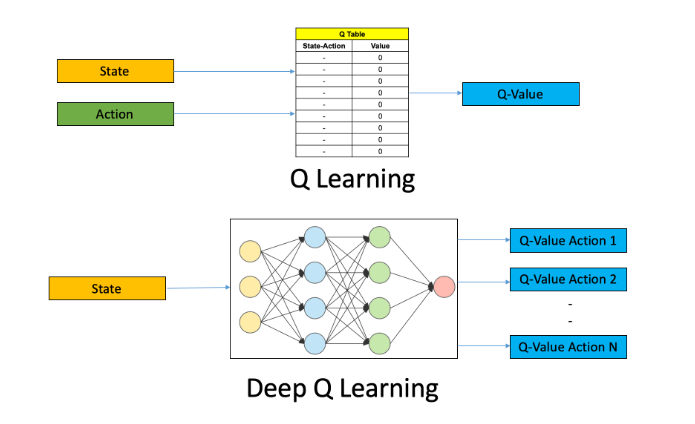
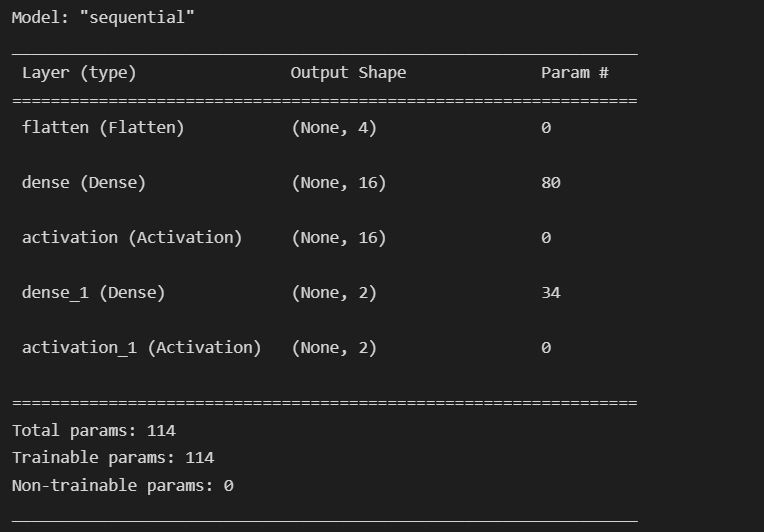
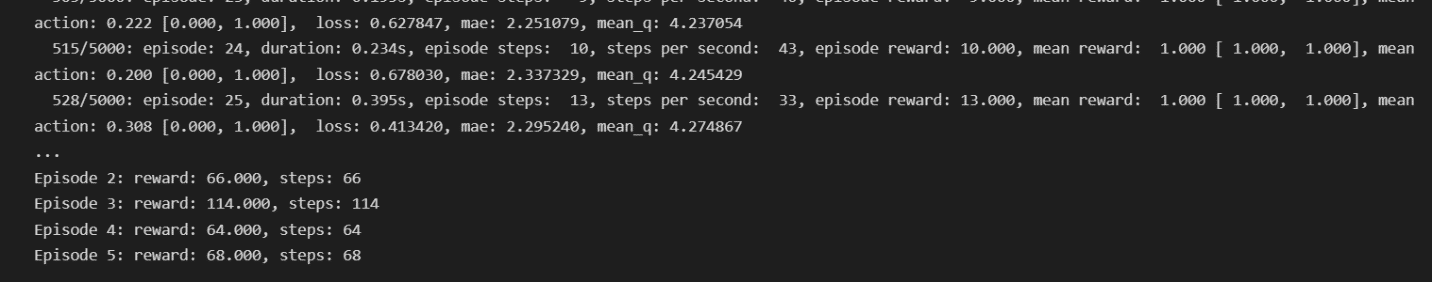
For Week 8 lab, we are implementing deep Q-learning using OpenAI Gym to get cart pole environment and using Keras-rl library to create a neural network. The idea behind deep Q-learning is instead of getting Q-Value as an output with state-action input, we use NN to produce Q-value for N number of actions. Because of this, we eliminate the disadvantage of memory usage in a case of using tremendous number of states and actions. The image below shows how Q-learning and deep Q-learning work.



Now, the lab is using deep q-learning to play cart pole and the agent maintains the pole balanced by using one hidden layer NN model, we also can configure our policy as Epsilon greedy policy, and store Q-values from actions we set the memory as sequential. Then we train the model. And test the model

Here is the output for the training phase:





The training took a long time, even though visualization is turned off for training. I wonder if we can speed up the process by adding layers. I was playing with the code to achieve that. However, it always ends up with errors. I also faced a problem importing Adam from Keras optimization library. I’ve solved it by “from tensorflow.keras.optimizers import Adam”