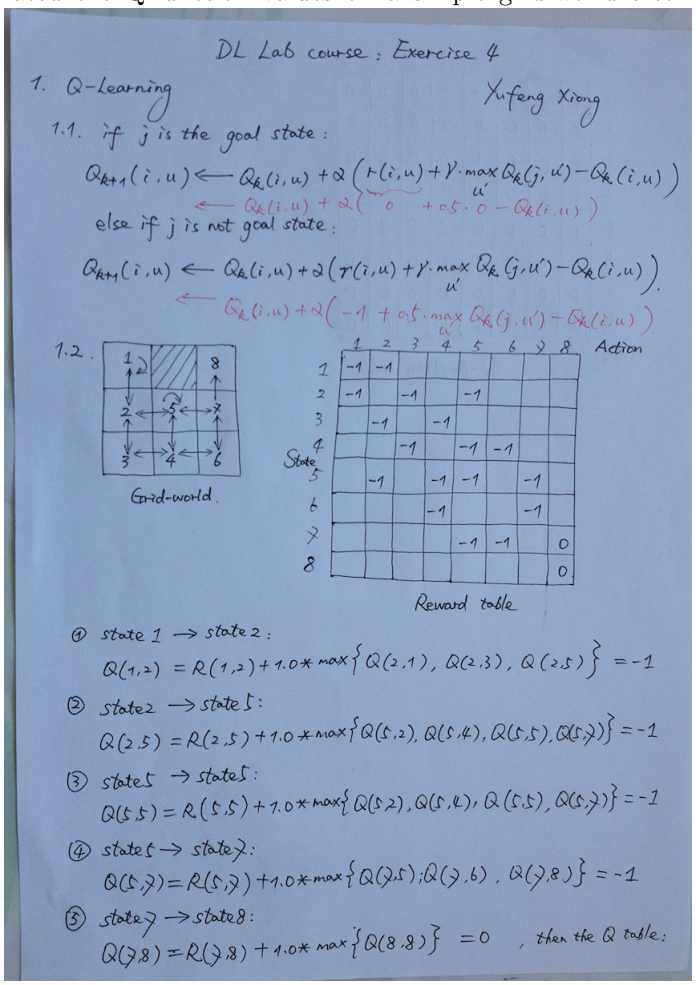
Deep Learning Lab Course: Assignment 04

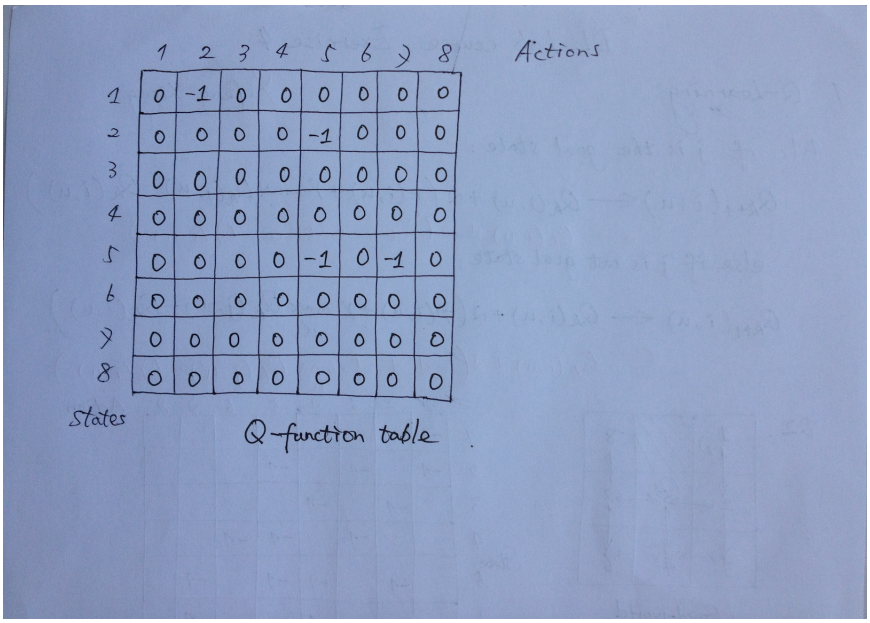
Yufeng Xiong

January 30, 2017

1 Q-learning

In exercise 1, I wrote the update-rule of Q-learning function and calculated the Q-function values on a simple grid-world after the initial episode:





2 Deep Q-learning

2.1 Network Architecture

The implemented deep Q-learning network is a LeNet5-like neural network, used to predict the action of the deep q-learning agent for a simple maze task, the detailed neural network layers are:

a. convolutional layer 1
b. pooling layer 1
c. convolutional layer 2
d. pooling layer 2
e. fully connected layer 1
f. fully connected layer 2

2.2 Implementation details

2.2.1 Some parameters

training steps: 200,000
testing steps: 30,000
batch_size: 128 (to make the model more stable)
epsilon = 0.2

2.2.2 Memory replay

In the first 3000 steps, the deep Q-learning agent will not learn anything, just takes random actions for memory replay.

After the first 3000 steps' memory replay, the agent will start to learn. In addition, I used epsilon-greedy algorithm for agent exploration, to avoid the agent overestimating the Q-function values.

2.2.3 Saved model

In my example, after 120,000 training steps, the deep Q network can reach some kind of convergence. I saved the trained model in the /model directory after 200,000 steps. At then the deep Q agent becomes a very intelligent agent for the sample maze task, just need very a few steps to reach the terminal state.

2.3 Testing results

In testing phase, I run the saved model for 30,000 steps, the accuracy can reach 99.904%.