Deep Learning Lab Course: Assignment 03

Yufeng Xiong

December 16, 2016

1 Network Architecture

In assignment 03, I implemented a LeNet5-like neural network (similar to assignment 02) to predict the action of a A-star planner in a grid map, 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
g. softmax out layer

2 Implementation

I trained the neural network using AdamOptimizier for 15 epochs, after training, the Minibatch error is around 3.1%, the Validation error is around 6.2%.

2.1 Details

2.1.1 History states

In the first epi_step, I will copy the first state for 4 times as the 4 history states and repeat 32 times to form the input shape [32, 25, 25, 4]. In the next epi_step, the first element will be replaced by the latest state as the 4 history state.

2.1.2 Starting actions

In order to let the A-star planner start to move easier, in the first 10 epi_step, the planner just take random action. After the first 10 epi_step, the planner will take action according to the prediction from the pretrained model.

2.2 Issues

Although the training error is really good, while in the testing phase, the accuracy is very bad (around 10%). It seems after the first 10 epi_step, the planner also takes some 'random' actions thus can not lead to the target position successfully.

Since I got some problems with the saved trained model, so I put the test_agent.py into the train_agent.py to perform the test. Maybe I will fix it later.