README for A3C

Shiyi Cao 516021910391

SJTU

1 Introduction

This is a README document for A3C implementations. A comprehensive demonstration of training and testing A3C on the Pantheon platform is included in this document. Moreover, the code for A3C also supports further improvement by changing the settings of state, actions and rewards. An instruction on how to make different settings for A3C implementations will be illustrated in this document as well.

2 Running A3C

Make sure your environment is Python 2.7 and tensorflow 1.12.0. There is already a pre-trained model included under *a3c/logs*, to test without training, run "src/experiments/test.py -local -schemes a3c –data-dir DIR". And then run "src/analysis/analyze.py –data-dir DIR" for analysis result. For more detailed usage, run "src/experiments/test.py -local -h".

To train a new model, go to *thirdparty/indigo/a3c* and run "python train.py –ps-hosts 127.0.0.1:9000 –worker-hosts 127.0.0.1:8000 –username YOUR USERNAME –rlcc-dir /pantheon/third_party/indigo". After training, copy all the model files under $log/TRAINING\_TIME$ into *log* and delete the original model files under *log*.

3 Modified Files

The main model file for A3C algorithm is *thirdparty/indigo/a3c/models.py*, where you can change the network settings such as the number of layers and the number of LSTM cells.

thirdparty/indigo/a3c/a3c.py is the training file for a3c algorithm. Related training settings like learning rate, max training epoch and other things can be modified according to your own need in this file.

To modify the settings of the DRL formulation, such as the state, action and rewards, go to *indigo/env/sender.py*. State is defined in *recv*(), rewards is defined in $compute\_perfomance()$ and actions are defined at the beginning of "Class Sender".