

Hierarchical Bayesian Neural Network in Pytorch

This is the code adapted from the Joshi's work, implemented in pytorch.

Required Packages:

- Python 2.7.13
- PyTorch 0.2.0
- torchvision 0.1.9
- Numpy 1.13.3
- Matplotlib 2.1.0
- Scikit-learn 0.19.1
- Scipy 0.19.1

Note:

Pytorch is only supported in linux based systems.

The only usage for scikit-learn is to shuffle the numpy array.

Introduction

I am currently writing a python script to do experiments for me to find the best parameters. Currently, when the experiment is done, the program would send an email to a designated destination (*hxianglong@gmail.com* by default) through Fudan mail system (*xlhu13@fudan.edu.cn* by default because the local internet would not be shut down automatically on a daily base).

Usage

The *Net* class is a data structure designed for Hbnn, why the *Train* class sets up the parameters for the training. *Mail* function is used to send email to notify you when the experiment is done no matter it fails or succeeds. *Test* actually starts the training.

Server

Usually it takes tens of hours for the program to finish. In order to run the program in the background without interruptions, screen command is recommended.

```
# this create a virtual terminal named demo, and it's automatically attached
screen -R demo
# activate the conda environment
source activate Hbnn
# check the status of all the NVIDIA GPUs
```

```

nvidia-smi
# the default GPU is GPU 0
# run the python program on GPU 0 and record the output in a text file
python test.py |& tee test.txt
# in case GPU 0 is full, there would be something like
# 'conda: insufficient memory'
# run the following command instead so that GPU 1(likewise 2, 3, 4) is used
CUDA_VISIBLE_DEVICES=1 python test.py |& tee test.txt

# to detached from the screen, i.e., run the screen in the background
# press ctrl+a d

# you could check the number of existing screens by
screen -ls
# and the terminal would print the screen name with an id number as well as
# the status of the screen
screen -r demo
# or
screen -r id_of_demo
# to reattach the demo screen

```

The previous commands spares you from keeping connected to your server through ssh all the time. Programs running in screens would keep running in the background. And once it's done, you would receive an email with the output and pictures of the experiment.