Introduction of Computer

Homework 5 - DNN

Due 23:59, June 18, 2019

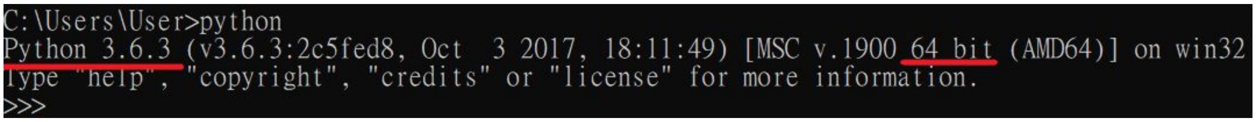
Neural network has rapidly evolved in the past decade. In this exercise, you should implement some DNN models with Keras and get a glimpse into DNN.

Framework Installation

In this part, you are required to install the DNN tool tensorflow and a high level API Keras (python version >= 3.5). If you have already install Keras(version >= 2.0.0) and it works well, do not need to reinstall.

Tensorflow can be installed with or without GPU support. You only need to install tensorflow without GPU in this homework. However, you may try the tensorflow with GPU support if you have a Nvidia GPU with CUDA Compute Capability 3.5 or higher. See the CUDA Compute Capability.

Prerequisites to install tensorflow (with or without GPU support) and Python version >= 3.5, 64-bit



```
C:\Users\User>python
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Open python shell in the command line prompt to check the python version. If your python version is lower than needed or is 32-bit, uninstall it and install the latest one.

- (For interested students only) Install tensorflow with GPU support
  - CUDA 10.0
    Remove the current version of CUDA from the control panel.
    Download the CUDA installation file and install it.
  - cuDNN 7.4.1 or higher
    Register a Nvidia Account and login
    Download the cuDNN library.
    Follow the Installation Guide and copy the binary, library, and header file to the specified location.

After installing all the tools, you can now install the tensorflow using the native pip.
Pip is a package management system used to install and manage software packages written in Python.

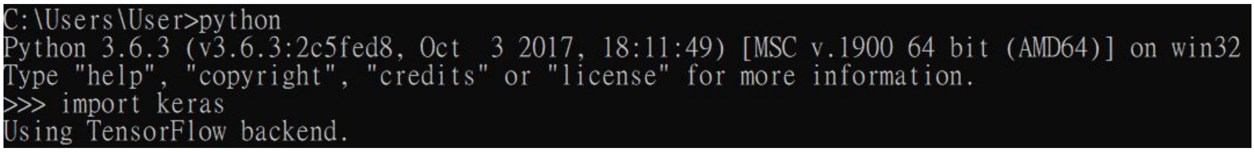
- For tensorflow without GPU support, install with:
  python –m pip install --upgrade tensorflow

- For tensorflow with GPU support, install with:
  python –m pip install --upgrade tensorflow-gpu

- Finally, you can install Keras with:
  python –m pip install --upgrade keras

Exercise Requirements

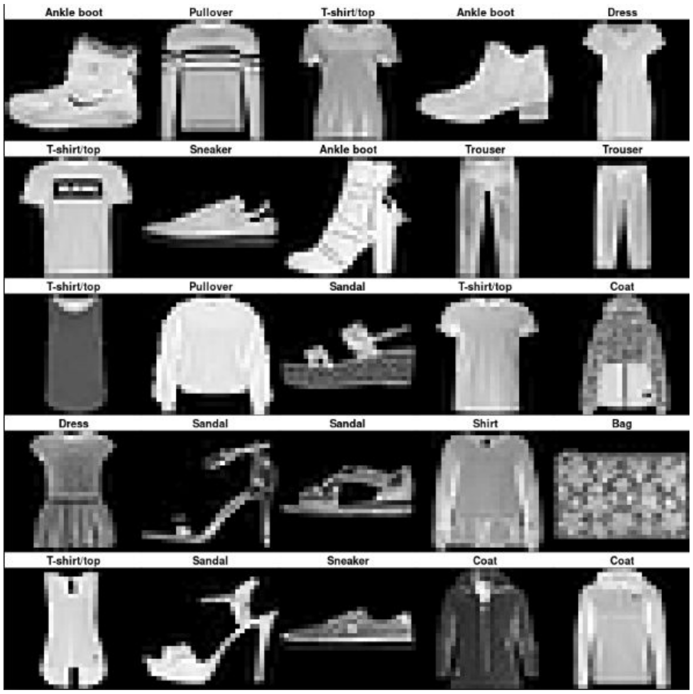
- **Verify Installation (30%)**
  To verify your tensorflow installation, open the python shell and import the package “keras”. There should be a message showing that keras is using the tensorflow backend. Take the screenshot, and then paste the results in your report.



```
C:\Users\User>python
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
```

- **Classification with Neural Network (70%)**

In this part, you are going to build your own network to classify the Fashion MNIST dataset. The dataset contains 28 * 28 pixel greyscale images in 10 clothing categories.



In the provided code hw5.py, we have done the data preprocessing for you. The pixels have been normalize to [0, 1] and are in one dimensional array.
You should build a keras sequential model to predict the class of the testing data. We provide 60000 images for training and 10000 images for testing. Labels are only provided for training. You can validate the accuracy of your model by splitting the validation dataset by Keras built-in function. We will test your model with testing labels which are not provided.
Data download link

Performance evaluation (You can freeze random seed if you want.)

1. Exceed simple baseline 85% for testing data (20%)
2. Exceed strong baseline 88% for testing data (10%)
3. Test accuracy top 10% in class (5%)

Write a short report about how you build your model and answer the following questions:

1. The structure of your best model. (10%)
2. Training accuracy (5%)
3. How do you determine the termination of training ? (10%)
4. How do you determine which model structure is better ? (10%)

Bonus(7%)
You are encouraged to implement a model with convolutional neural network (CNN). You can submit a hw5_cnn.py and describe your model structure and observation of CNN model in report (3 points for bonus). If your CNN model exceeds the baseline 91% on testing data, you will get 4 points for bonus.

File Usage
TA offers a code template for data preprocessing. Please use it and implement the exercise. You can run with command:

```
python3 hw5.py ./data output.csv
```

./data: data path
output.csv: output file name. The first column is id and the second column is their class prediction

We also provide one for bonus. You can run with command:

```
python3 hw5_cnn.py ./data output.csv
```

We will run your models with GeForce GTX 1080(8G), be aware of the computational limitation.

Submission rules

- TA offers a report template, please use it for your report. https://docs.google.com/document/d/1jWOU9SviteJqO8hsIl88eWSggIuKqiNojG1YoewOhUE/edit?usp=sharing
- You should submit your best model in hw5.py. Compress your report.pdf and code (hw5.py, hw5_cnn.py, reader.py) into a .zip file and submit it to CEIBA before deadline.
- DO NOT upload data and well-trained model.
- No Plagiarism.
- Accept late submission for 2 days after the deadline. Late submission penalty is 15 points per day.
- Wrong submitted format will get 10 points penalty.
- It is your responsibility to make sure the submission is completed. Showing an unsubmitted set of homework after the due date will not work.

There are plenty materials on the Internet, for example:
1. [資料分析&機器學習] 第 5.1 講: 卷積神經網絡介紹 (Medium)
2. https://morvanzhou.github.io/tutorials/machine-learning/keras/2-3-CNN/
3. https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py