CS145 Homework 3: KNN and Neural Networks

Instructor: Yizhou Sun | TA: Junheng Hao (in charge of HW3), Yunsheng Bai, Shengming Zhang

Due Date: 11:59pm, Nov. 9, 2018 (Monday)

Please read this instructions carefully, especially submission, grading and notes!

Setup Instructions

Before you start Homework 3, make sure you have finished the preparation step to set up the environment. Note that you may use different operating systems (Windows, Mac or Linux). The setup instuctions are different. If you have problems in this step, please come to TAs for help.

Code of HW3 has been tested to be compatible with python version 3.6 (it may work with other versions of 3.X, but we won't be officially supporting them.). You will need to make sure that during your virtual environment setup that the correct version of `python` is used. If you would like to use Python 2.7, it is also OK but keep in mind that you may have problems or errors, which may take extra efforts.

Set up Python 3 (For Mac OS/Linux)

Most of you have already installed Python 3 for previous assignments. If you have done that, you can skip this step. It is recommended to install Anaconda to manage Python and packages. Other than that, for Mac OS X, you can also install Python 3 via Homebrew and call `brew install python3`. For Linux, you can install through `sudo apt-get`.

Set up Virtual Environment (For Mac OS/Linux)

In case you have different python package version and other issues, we recommend to use Python virtualenv which sets up a vitural environment. At a high-level, a virtual environment creates an environment where you may install packages that aren't installed on your main system Python. This is useful for separating out projects. Further, if you accidentally break your Python during some package installation (it can happen and is not fun), but you were in a virtual environment, you can simply delete the virtual environment and start a new. Use `pip` to install it: `sudo pip install virtualenv`.

Please run the following in the terminal to start visual environment and install required packages of HW3 after installation:

```
cd HW3
virtualenv -p python3 .env       # Create a virtual environment (python3)
# Note: you can also use "virtualenv .env" to use your default python
source .env/bin/activate         # Activate the virtual environment
pip install -r requirements.txt  # Install dependencies (this takes a while)
# Work on the homework for a while ...
deactivate                       # Exit the virtual environment
```

Alternatively, you can use Anaconda to set up virtual environment if you have installed it. To set up a virtual environment with Anaconda, run (in a terminal):

```
conda create -n .env python=3.6 anaconda    # Create an environment called
.env
source activate .env                        # Activate and enter the
environment
# Work on the homework for a while ...
source deactivate .env                      # Exit the virtual environment
```

Check here for more details on managing virtual environments with Anaconda.

Do I have to use virtual environment? The answer is "No, probably you don't". Be advised, however, that if you choose not to use virtual environment, it is up to you to make sure all dependencies in the code are installed globally on your machine. However, if things break, it is not possible for us to help debug each student's unique installation, as each computer setup is different, and the bug may be any package you have previously used or installed interacting negatively. That is why we prefer to have you use virtual environments and have provided you the `requirements.txt` file for the homework.

Installing Python and virtualenv for Windows

Download Python package from https://www.python.org/downloads/release/python-364/ and install it. Run cmd as administrator and run these commands:

```
pip install virtualenv
cd HW3/
virtualenv -p path to python3/python.exe .env # This creates the virtual
environment
.env\Scripts\activate.bat # This activates the virtual environment
pip install -r requirements.txt # This installs all the packages
# Do some work...
.env\Scripts\deactivate.bat # This deactivates the virtual environment
```

Note: We find the setup instructions from online (but all TAs do not have Windows sorry). Please tell us if you have problems. We suggest to search on StackOverflow or post your problem on Piazza first.

Set up and Launch Jupyter Notebook

To finish this homework, we use `jupyter notebook` for coding and report. An IPython notebook lets you write and execute Python code in your web browser. IPython notebooks make it very easy and interactive to tinker with code and execute it in steps. Check here for more details if you are not familiar with Jupyter Notebook and how it works.

If your virtual environment installed correctly, you should **NOT** have to manually install it from the instructions on the website. Just remember to run `source .env/bin/activate` in your homework folder.

Run:

```
cd HW3/
jupyter notebook
```

Also you can use `jupyter lab` which is basically the same thing as `jupyter notebook` but probably a little bit more convenient .

This will launch a new browser window (or a new tab) showing the notebook dashboard. Or you can visit `http://localhost:8888/` to open the dashboard.

When started, the jupyter notebook can access only files within its start-up folder (including any sub-folder). Make sure your relevant files are on the desired path.

Note: To write down the answers in the notebook, you may need a little knowledge of **markdown** syntax, which is basically like txt but more flexible and readable. Here is a brief tutorial. (Fun fact: You can even use it when posting on Piazza!)

Congratulations for completing the setup for your coming exciting homework! Now let's start! ^_^

Data Preparation

Download the CIFAR-10 dataset (file size: ~163M). Run the following from the `HW3` directory:

```
cd cs145/datasets
./get_datasets.sh
```

Also you can download the dataset here, and put it under `cs145/datasets` folder.

After downloading the dataset, you can start your notebook from the `HW3` directory.

Task 1: KNN (30 points)

Task: Complete the k-nearest neighbors Jupyter notebook including codes and questions (following the instructions and notes in `knn.ipynb`). The goal of this workbook is to give you experiences with the CIFAR-10 dataset, training and evaluating a simple classifier, and k-fold cross validation. Print out the entire workbook and submit jupyter notebook together with all source codes.

Task 2: Two-layer Neural Networks (70 points + 20 bonus points)

Task: Compete the two-layer neural network Jupyter notebook including codes and questions (following the instructions and notes in `toy_nn.ipynb`). Print out the entire workbook and submit jupyter notebook together with all source codes.

Submission

- **Step 0:** Make sure to finish the missing lines and questions in notebook.
- **Step 1:** Convert your `ipynb` files to PDF files. If you have problems to do that from the Jupyter Notebook in the browser, try to convert all your `ipynb` notebook files to HTML by running `ipython nbconvert --to html FILE.ipynb` and print as PDF.
- **Step 2:** Confirm that you have required files for submission: two python files (`knn.py` and `neural_net.py`) and two notebook files (`knn.ipynb` and `toy_nn.ipynb`), which have been converted to PDF already (you may name them as `knn.pdf` and `toy_nn.pdf`).
- **Step 3:** Submit a zip file of your homework to CCLE using `zip_hw3.sh` . Make sure you have include all your code (`.py` files) and notebook. Note: **DO NOT** submit dataset files.
- **Step 4:** Check again your files can be sucessfully unzipped and click "Submit for Grading" in CCLE. **If you do not click "Submit for grading" and we probably notice a late submission in your submission, we will apply late-time penalty. No email submission will be accepted this time.**

Grading

- Grading policy of HW3: Code implementation (approx. 40%) + Results and Analysis (approx. 60%) in the notebook (excluding bonus questions) Points will be significantly deducted if you only provide results with no required analysis in the notebook!

  Points will be significantly deducted if you only provide results with no required analysis in the notebook!

- Keep the academic honor code in mind.

  With its status as a world-class research institution, it is critical that the University uphold the highest standards of integrity both inside and outside the classroom. As a student and member of the UCLA community, you are expected to demonstrate integrity in all of your academic endeavors. Accordingly, when accusations of academic dishonesty occur, The Office of the Dean of Students is charged with investigating and adjudicating suspected violations. Academic dishonesty, includes, but is not limited to, cheating, fabrication, plagiarism, multiple submissions or facilitating academic misconduct.

Notes (Keep Updating)

1. Be aware of your installation if you have Python 2 and Python 3 at the same time on your machine with diffrent python and package file path.
2. Pay attention to the indentation of code. Different code editors may have different default indentation settings. Please make sure this does not bring errors or affect your code before you start.
3. If you installed homebrew previously and then updated your OS system, you may have to

uninstall and reinstall homebrew.

4. If you are working in a virtual environment on OSX, you may *potentially* encounter errors with matplotlib. Check here.
5. We know it is not that easy to fill in other's code. You are free to change the code we provide even though it is not inside the missing lines. Keep in mind that you should not change the whole structure of the codes and we will not support the such change on code.
6. Do **NOT** use other functions (such as kNN or dense layer in sklearn, tensorflow or pytorch) in your inplementation. If you report your results by using these function directly, you will get 0 points.
7. More reminders to be updated!

Acknowledgement

We thank Prof. Jonanthan Kao, Zheng Cheng, Tianwei Xing, together with Serena Yeung & Justin Johnson, for permission to use their code written for the ECE239AS: Neural Networks & Deep Learning (Winter 2018). This graduate course is highly recommended and helpful if you are interested in advanced techniques in neural networks.