Programming Assignment 5: Convolutional Neural Networks

CS 4670 Spring 2018

April 30, 2018

Key Information

Assigned:	Monday, April 30th (Code is on CMS)
Due:	Wednesday, May 9th
Files to Submit:	`student.py`, `readme.txt`

This project can be done **individually or in pairs**.

Overview

In this assignment, you will explore convolutional neural networks in the context of image classification. You will train networks for two different (but related) classification problems, using the concept of transfer learning. Additionally, you will implement a technique that helps us understand what the network has learned, and you will explore the potential shortcomings of these networks by generating *adversarial examples.*

Initial setup

Note: this step needs to be executed only the first time you are setting up the project on your machine

Before starting up your VM, make sure that it is allocated at least 3000 MB. You can change this in VirtualBox by right-clicking on the VM name and navigating to Settings ⇒ System ⇒ Base memory. If you still run into memory errors during the project, allocate more memory and restart the VM.

Download the files for this project from CMS and put them in a single folder on your VM. There should be three project files on CMS: `PA5.ipynb`, `student.py`, and `setup.sh`.

In a terminal window inside your VM, navigate to the folder containing these files. Then, run `bash setup.sh`. This will execute a setup script that will set up the environment that you need for this project. During the setup, the script might ask you if you would like to install Microsoft VSCode – you can safely say no to that. **Besides VSCode, when the script asks you for a confirmation of any kind, be sure to enter yes!** Sometimes the default option for these confirmation prompts is no instead of yes; please read these prompts and enter yes instead of just hitting the enter key.

Once the setup script completes, run the command `source ~/.bashrc` in the terminal.

In summary, here are the four steps for initial setup; make sure you do all of them!

1. Allocate enough memory (3000 MB should be sufficient) to your VM before starting it
2. Download the project files from CMS and put them in one folder on your VM
3. Run `bash setup.sh` within the project folder; say yes to every confirmation except the one about installing VSCode
4. Run `source ~/.bashrc`

If you have any questions about the initial setup, please check Piazza to see if someone else has run into the same issue. If not, feel free to post a new question!

Anaconda environment

The initial setup downloads and installs a virtual environment manager for Python called Anaconda. Specifically, a virtual Python environment called `pa5` is set up, and all the required packages for this project are installed within it. We can activate this virtual environment now to get access to all of the installed packages by running `source activate pa5` in the terminal. Once the environment has been activated, your terminal prompt should now display:

```
(pa5) >
```

Before running anything for this project, please make sure that you are within the `pa5` environment – this is what we will be using for grading! If you start a new terminal, or come back to your VM in a new session, you will need to reactivate the environment using `source activate pa5` before starting any work on your project.

Opening the assignment

Here is the workflow you should use any time you are working on this project:

1. In a terminal window, activate the `pa5` environment as described above.

2. Run `jupyter notebook` in the directory containing the project files. This should open up a GUI in your browser.

3. Click on `PA5.ipynb`, which will open up the Jupyter Notebook that we will be using for this project. This file contains detailed explanations about the project and the various TODOs. If you haven't used a Jupyter Notebook before, or if you'd like a refresher, check out this quick video tutorial!

4. All the code that you will write should go into `student.py` – this is the file that will be submitted and graded! The cells in the notebook simply import your code and run it, and you should not need to change any of the code in the notebook itself. Moreover, every cell that imports your code will automatically reimport your latest `student.py` code, so you can just save changes to `student.py` and re-run any cell.

5. Note that the notebook will lose its variables when it's restarted. Whenever you start the Jupyter Notebook server afresh, we recommend re-running all the cells up till the one you are currently working on. This will ensure that all the variables up till that point are re-loaded into the notebook.

Testing and Debugging

In the notebook

We have added cells within `PA5.ipynb` that help you visualize your results so far – these should give you a good idea of whether you're on the right track or not! You can also print variables in the notebook cells in case you'd like to look at them for debugging purposes.

Automated tests

Some number of automated tests will be released early next week with instructions for use.

Deliverables

You should submit two files:

1. `student.py`, which should contain all of your work filled in for each of the TODOs. Note that there are a few short questions after Parts 1, 2 and 3 – be sure to fill the answers out in `student.py` in the spaces provided!

2. `readme.txt`, where you should document any known problems in your solution, or anything else you would like us to know. Additionally, please leave feedback on this particular assignment as a whole! It is a new assignment, so we would love to hear your thoughts on it and how we could make it better next time around.