Course Software Infrastructure and Background

The purpose of this document is to give you the basic instructions to install and use software in this course, for use in the assignments and the project. While a pre-requisite of this course is that you have already used Python and one of PyTorch or Tensorflow, this document focuses on how to use PyTorch and the related packages that you will need. In addition, it has some pointers to background on some of the packages.

There are two choices of environments to use in this course: **Google Colab** (described the Section 1) on a cloud-based computer or using **Anaconda Python and Jupyter Notebook** on your own computer, described in Section 2.

    The choice between environments is important and you may actually be best served, in the long term, using both. Here's why: Google Colab takes away the difficult of doing software installation, and gets you up and running quickly. It also supplies GPU hardware that will run your code faster, and that may be important. However, professional machine learning engineers know how to install and run many different environments and maintain them in a proper working state, because they have to deploy working systems, not only on their own machines, but across the servers they are deployed on. You will also find running the first parts of this assignment easier on a local-to-your-computer notebook - for example, it doesn't disconnect with inactivity which happens with Google Colab. Also, it is good to be comfortable and familliar with plain `Python` without notebooks, as it is easier to run scripts that explore things like hyperparameter with plain Python, which you can only do your local computer.

Section 3 provides some links to refreshers and cheatsheets on Python, NumPy, and Matplotlib.

1   Using Google Colab

With a basic google account you can use Google Colab which provides free and fairly good performance computing without the need to do much set up of your environment. When logged in to your google account, go to https://colab.research.google.com. To learn how to write python code into a Google Colab notebook, read and follow the following links:

1. What is Colab?

2. Near the bottom of *What is Colab?* click on these links:

   - Overview of Colaboratory
   - Guide To Markdown
   - Guide to Local Files, Drive, Sheets and Cloud Storage

Colab also has most of the libraries/frameworks we need for this course already installed, including **torch** and **torchtext** (https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html) and **spaCy** (https://spacy.io/). However, to use the english library of spaCy in Colab, you'll need to import it (every time) with the following code in the notebook (the exclamation mark causes the code to run in the shell containing your code):

```
!python -m spacy download en
```

2 Using Your Own Computer: Anaconda, Virtual Environments, Torch and SpaCy

If you don't have access to Google, as above, or prefer to run everything on your own computer you can use **Anaconda** distribution of Python **3.7**, which comes pre-installed with several scientific computing libraries including `NumPy` and `Matplotlib` and `Jupyter Notebook`. We are using version **3.7** to be consistent with the version of Python used in Google colab. I would note that it is good software engineering practice to know how to work with Anaconda environments and installing the software you use, and would recommend that you do this, perhaps in addition to using Google Colab.

1. Download the latest Python 3.7 version from https://www.anaconda.com/distribution/ for your specific operating system (OS), one of Windows, macOS, or Linux. Choose the "64-Bit Graphical Installer" to do the installation. (It is also fine to choose the "64-Bit Command line installer" if you are familiar with the command line.)

2. Follow the detailed installation instruction steps that are given in https://docs.anaconda.com/anaconda/install/ for each OS. You do not need to install Microsoft Visual Studio Code when prompted, but it is a popular development environment. For Linux, you can skip step 2 (hash check) as it is optional.

2.1 Setting up your Virtual Environment

It is good practice to create a 'virtual environment' which ensures that the Python tools and libraries are the right ones that we specify. You will create a virtual environment, called `ece1786`, using the Anaconda 'conda' command as described in the following steps:

1. Open up a *command line terminal*: To do this on a Windows PC, search for "Command" and open Command Prompt; On Mac and Linux, you should open the "Terminal" application.

2. To create the virtual environment, run the following command in the terminal:

```
conda create -n ece1786 python=3.7 anaconda
```

This process will take several minutes, possibly longer if you have an older computer.

3. To test that the environment works, activate the environment by running:

```
conda activate ece1786 (for Mac/Linux)
       activate ece1786 (for Windows)
```

After this, you should see `(ece1786)` as the command line prompt.

4. To exit from the environment, you can simply close the window, or run:

```
conda deactivate (Mac/Linux)
       deactivate (Windows)
```

Then the `(ece1786)` should disappear as the command line prompt.

2.2 Lauching, Learning and Using Jupyter Notebook

Once you've got the virtual environment working, launch it again in a command/terminal window. Then simply type:

```
jupyter notebook
```

After a few moments, a new web browser will launch, and it will contain a list of files that were in the folder/directory in which you launched `jupyter`. To get started with using Jupyter Notebooks as your development platform, you'll need to read a tutorial, such as this one: https://www.dataquest.io/blog/jupyter-notebook-tutorial/. Once you've gone through this, then you can move on to the next section.

2.3 Installing PyTorch and TorchText and Spacy

1. On a command line, activate the `conda` environment for this course first:
   `conda activate ece1786` (macOS/Linux) or `activate ece1786` (Windows).

2. To download PyTorch go to https://pytorch.org/, which at the bottom of the page will give you a `conda` command to download the correct verson for your specific operating system and hardware.

3. You can test that your system has installed PyTorch successfully by bringing up a Python interpreter (i.e. run `python` in command line while inside your `ece1786` conda environment) and running '`import torch`'.

   In addition to PyTorch, we will be using two additional libraries:

- **torchtext** (https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html): This package consists of both data processing utilities and popular datasets for natural language, and is compatible with PyTorch. We will be using `torchtext` to process the text inputs into numerical inputs for our models.

- **SpaCy** (https://spacy.io/): For 'tokenizing' English words. A text input is a sequence of symbols (letters, spaces, numbers, punctuation, etc.). The process of tokenization separates the text into units (such as words) that have linguistic significance.

To install these two packages use the following commands:

```
conda install -c pytorch torchtext
conda install -c conda-forge spacy
python -m spacy download en
```

3 Reference Material on Python, Numpy and Matplotlib

It is the expectation of this course that you are capable in the use of the Python language, and will have already been exposed to NumPy, and Matplotlib, but these links are included for easy reference:

1. For a concise summary of Python, see: https://learnxinyminutes.com/docs/python3/. Python refresher.

2. See the NumPy and Matplotlib section of the Stanford CS231n course Python Tutorial: https://cs231n.github.io/python-numpy-tutorial/.

3. NumPy Tutorial: https://engineering.ucsb.edu/~shell/che210d/numpy.pdf.

4. NumPy cheatsheet: https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf

5. Matplotlib Tutorial: http://scipy-lectures.org/intro/matplotlib/index.html. Another tutorial that focuses more on the image visualization: https://matplotlib.org/users/image_tutorial.html

6. `Matplotlib` cheatsheet: https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf