ORIE5270 - Homework 2
Due February 8th, 2019 at 1:25pm

You are not allowed to collaborate with your classmates, but you may freely use the internet to look up any information that helps you to do the tasks! In fact, you are expected to accomplish this assignment via some online materials.

Problem 1 (Git Repos) Go to `https://github.coecis.cornell.edu/` and create a **PUBLIC** Cornell github repository and name it as `ORIE5270`. Create a local repository and add this cornell github repository as its remote repo. We will check your homework via the repo in Cornell github. Your `ORIE5270` repo should contain:

(1) A `README` file describing the repo,
(2) A `HW2` directory which contains your answers to this homework.
(3) In your `HW2` directory, there should be a `solution` file (in a markdown, or txt format). **You should also submit this file to Blackboard**. The `solution` file contains:
   (a) **At the beginning, there should be a `url` telling us where to find your repo**.
   (b) It should describe how to run your program file (e.g., what package should one install), description of each program file, and which plots we should look at.
   (c) See Problem 2 and 3 for additional contents it should contain.

   You should put your program code in separated files (under the `HW2` directory) and provide enough comments in the code so it is readable.

We will also look for a **merge** in your repo. You can achieve this task in whichever way you like. For example, you can create a branch called `Problem` after creating your local repo and the `README` file. You create the `HW2` directory and work out the following problem in the branch `Problem`. Then you merge the `Problem` branch to the `master` branch. (If you do it in this way, make sure to push the `Problem` branch to the remote repo and keep it there.)

Problem 2 (Erdos Number) The Erdos number is the number of "hops" needed to connect the author of a paper with the prolific late mathematician Paul Erdos. An author's Erdos number is 1 if he has co-authored a paper with Erdos, 2 if he has coauthored a paper with someone who has coauthored a paper with Erdos, etc. In this task, we ask you to pull and analyze the data of all authors with Erdos number 1. The data is at the url

`https://files.oakland.edu/users/grossman/enp/Erdos0.html`

which contains the author name, the year he/she first coauthored with Erdos, and the number of time he/she coauthored with Erdos. You can find more information in `https://oakland.edu/enp/thedata/`.

Write a python script `p2.py` so that when we run it, it can finish the following tasks:

(1) Pull and store the data of the author name, the year he/she first coauthored with Erdos, and the number of times he/she co-authored with Erdos in a Pandas DataFrame object with name `df` *directly* from
   `https://files.oakland.edu/users/grossman/enp/Erdos0.html`.
   The word "directly" means you should NOT assume you have manually copy and paste the content of `Erdos0.html` to a local file in the repo. Your data frame `df` should have 3 columns and 511 rows, and has column name `Name`, `Year`, and `Frequency` corresponding to author name, the year of first coauthoring with Erdos and the number of times of coauthoring. Print the first 10 rows of `df` so we know

you are doing it right. (It would be better if you exclude some special characters at the end of the names but we do not require you to do that.)

(2) Print the name who has coauthored with Erdos most often and his/her frequency in a readable format. For example, if John Smith has coauthored with Erdos most often and his frequency is 10, a readable format would be

`John Smith has collaborate with Erdos most Often and his frequency is 10.`

(3) Print the name who last coauthored with Erdos in terms of the year of his/her first joint paper (so if a person coauthored many times with Erdos, we only consider the year of his/her first joint paper with Erdos) and the year in a readable format.

(4) Plot histograms of the `Year` column and the `Frequency` column side by side and save them as a pdf file in your repo. There should be enough guidance on the plot so that we know which histogram corresponds to which column of `df`.

In your solution file, you should write answers for item (2) and (3) above and tell us how to run your code. Also tell us which plots we should look at. You also have to comment on the histograms you have plotted. You are expected to search over online materials to see how to complete the above tasks. **Bonus:** Can you find your instructors' Erdos number?

Problem 3 (Convolutional Neural Networks) For this problem, you will write a python file to build a Convolutional Neural Networks (CNN). To understand CNN conceptually, I suggest you read

`https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`

You might want to go to

`https://adventuresinmachinelearning.com/keras-tutorial-cnn-11-lines`

to see how to implement a CNN.

Write a python script to `p3.py` to finish the following tasks:

(1) Build a convolutional network for classification of 10 classes using `Keras` for RGB images of size 32×32. Your neural net should have layers in the following order:

Convolution Layer $\rightarrow$ MaxPooling Layer $\rightarrow$ Convolution Layer $\rightarrow$ Flattern Layer $\rightarrow$ Dense layer $\rightarrow$Dense Layer.

You should use

`loss=keras.losses.categorical_crossentropy, optimizer=keras.optimizers.Adam(), metrics=['accuracy']`

when compiling the model. Activation function except the last Dense layer should be `relu`. The last Dense Layer should use `softmax`. You have freedom in choosing other arguments. Print a summary with number of parameters using `Keras` build-in functions.

(2) Apply your model to the CIFAR10 small image data (see `https://keras.io/datasets/` for how to import the data). Use 10% of the data set to reduce the computational burden. Your `model.fit` should have arguments

`validation_split=0.33, epochs=10, batch_size=128`.

Plot the history of `val_loss` and `loss` in `model.fit` in the same graph and export it to the repo. See "https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras" for how to achieve the history.

(3) Export a plot for the history of `val_loss` and `loss` when your model in (a) are equipped with a batch normalization layer right after the first convolutional layer.

(4) Export a plot for the history of `val_loss` and `loss` when your model in (a) are equipped with a dropout layer (0.15 as the argument) right after the first convolutional layer.

All your plots should be readable meaning that it is easy to see which line corresponds to which and the plots have appropriate x and y axis labels. You should search for (c) and (d) online to see how to add a batch normalization layer and a dropout layer. In your solution file, you should copy and paste the summary information of your model in (a). You should tell us how to run your 'p3.py' file, which plots we should look at for (b), (c), (d). Comment on the plots on (c) and (d). Do the additional layers in (c) and (d) help you in training? Do you have any other observations?