CS57700 Homework 1

Due date: Wednesday March 4, 2020 11:59pm

This homework will involve conceptual exercises and coding assignments. Instructions below detail how to turn in the conceptual part on Gradescope and codes via turnin.

1 Coding Assignment

In this section you will implement logistic regression and multi-layer neural network architecture for a multi-class classification problem. You must implement both of the classifiers from scratch in `Python 3`, no built in functions from libraries like `sklearn` can be used. However, you can use libraries like `pandas, numpy` etc for pre-processing and implementation purpose.

1.1 Problem

Identification of political framing is an interesting and important problem in Natural Language Processing. In simple words, framing means the perspective from which an event is described. For example, the mass immigration issue may be described from an economic perspective (job unavailability for the natives, increase in house rent and so on) or from a humanitarian perspective (low quality of life of the immigrants, lower wage and so on). Identifying the framing effect in text is an important problem as framing is often related to the ideological stance of the user. In this assignment you will identify frames in the Tweets of US politicians.

1.2 Dataset Description

You are provided with 1230 Tweets from US politicians where each Tweet is annotated with one of the 17 frames identified by political scientists. The detailed description of the frames can be found in Table 2 of the paper ”Leveraging Behavioral and Social Information for Weakly Supervised Collective Classification of Political Discourse on Twitter”[1]. The training set can be found in `train.csv`. You are given the test set of 820 Tweets in the file `test.csv`. Note that, the gold labels are not provided for the test set. The description of each column of the `csv` files are given below.

- `tweet_id`: Unique identifier for each tweet.
- `issue`: The issue being discussed in the tweet.
- `text`: Text of the tweet.
- `author`: The political identity (Democrat/Republican) of the author of the tweet.
- `label`: Framing label of the tweet (One of the 17 frames). This field is kept empty in `test.csv`.

You have to build classifiers using the training set and predict the labels of the test set. **You must not use the test set while training the models.** You are provided with a file `main.py` where you will find the starter codes. Feel free to define any additional functions needed in this file.

[1] https://www.aclweb.org/anthology/P17-1069.pdf

Hint: Politicians from different parties often use different frames while discussing the same issue to establish their stances. So, knowing the issue being discussed and the political affiliation of the politician who is talking may be a good indicator of the frame present in the text.

1.3 Task 1: Logistic Regression (20 Points)

Build a Logistic Regression classifier to identify the frames present in the Tweets. Fill in the function `LR()` in `main.py`. This function should learn a Logistic Regression classifier using the training set and predict the frames in the Tweets contained in `test.csv`. It will output a csv file named `test_lg.csv` with the predicted frames in the tweets in `test.csv`. `test_lg.csv` must have all the columns in the given order - `tweet_id, issue, text, author, label`. Make sure you preserve the mapping `tweet_id : text : label`, as `tweet_ids` are the identifiers which will be used to match your predicted labels with gold labels. (Hint: Follow the starter code provided)

1.4 Task 2: Multi-layer Neural Network (20 Points)

Build a Multi-layer Neural Network classifier to identify the frames present in the Tweets. Fill in the function `NN()` in `main.py`. This function should learn a Multi-layer Neural Network classifier using the training set and predict the frames in the Tweets contained in `test.csv`. It will output a csv file named `test_nn.csv` with the predicted frames in the tweets in `test.csv`. `test_nn.csv` must have all the columns in the given order - `tweet_id, issue, text, author, label`. Make sure you preserve the mapping `tweet_id : text : label`, as `tweet_ids` are the identifiers which will be used to match your predicted labels with gold labels. (Hint: Follow the starter code provided)

1.5 Bonus (5 Points at Most)

You must do cross validation and can do any kind of feature engineering to implement the tasks and improve the performance. The top 5 students will be given bonus points which can be supplemented to use any point loss only in the homeworks. The top scorer will get 5 points, 2nd top scorer will get 4, 3rd will get 3 and so on. The evaluation criteria is Macro F1 score of the prediction task on the test set.

2 Conceptual Questions (20 Points)

Now answer the following questions based on your implementation of Logistic Regression and Neural Network classifiers.

1. Write down the logistic function and loss function you used in Logistic Regression and derive the gradient. (4 Points)

2. Describe the architecture of the neural network you implemented using a computation graph **(Must be drawn using any software. Handwritten/drawn ones will be discarded from grading)**. Your answer should include the activation functions used in each layer (2 Points), the number of layers and number of neurons in each layer (2 Points). State the loss function used in your neural network implementation (2 Points). Derive the partial derivative associated with each node in the computation graph. (4 Points)

3. State one hyper-parameter you tuned in case of Logistic Regression and one in case of Neural Network. How did tuning these hyper-parameters change the result? Explain with learning curves. You may have tuned more than one hyper-parameter for each model. In this section explain only one from each model. (4 Points)

4. Explain in three sentences. What is framing? What can this analysis be used for in real-world situations? (2 Points)

2.1 Submission Instructions:

2.1.1 Conceptual Part

Upload your answers to the conceptual questions as a **typed pdf** in gradescope: `https://www.gradescope.com/courses/61435`

- For your pdf file, use the naming convention `username_hw#.pdf`. For example, your TA with username *roy98* would name his pdf file for HW1 as `roy98_hw1.pdf`.
- To make grading easier, please start a new page in your pdf file for each question. Hint: use a `\newpage` command in LaTeX after every question ends. For example, for HW1, use a `\newpage` command after each of the questions 1-4.
- After uploading to gradescope, mark each page to identify which question is answered on the page. (Gradescope will facilitate this.)
- Follow the above convention and instruction for future homeworks as well.

2.1.2 Coding Part

You need to submit your codes via Turnin. Log into `data.cs.purdue.edu` (physically go to the lab or use ssh remotely) and follow these steps:

- Place only the files `main.py, test_lr.csv, test_nn.csv` in a folder named `username_hw#`. For example, your TA with username *roy98* would name his folder for HW1 as `roy98_hw1`. **This naming convention is important. If the folder is not named correctly, there's no way to identify whose submission is that. Hence, may result in no grading.**
- Change directory to outside of `username_hw#` folder (run `cd ..` from inside `username_hw#` folder)
- Execute the following command to turnin your code: `turnin -c cs577 -p hw1 username_hw#`
- To overwrite an old submission, simply execute this command again.
- To verify the contents of your submission, execute this command: `turnin -v -c cs577 -p hw1`.
  Do not forget the `-v` option, else your submission will be overwritten with an empty submission.