**CS 590 NLP**

**HW6**

**MLPs**

**Due 04/06 11:59 pm**

**Overall Goal:**

In this homework you will incorporate word2vec embeddings with simple neural networks for classification.

|  |
| --- |
| **Useful python packages:** |
| For word2vec, you may find the gensim package useful: <https://radimrehurek.com/gensim/models/word2vec.html>  Note that gensim has built in pretrained word2vec embeddings so you should choose 1 or experiment with multiple in this homework. (<https://radimrehurek.com/gensim/models/word2vec.html#pretrained-models)>  Gensim also has nice built-in functions for working with word2vec data. |
| For the Multi-Layer Perceptron, there is a sklearn package:  <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>  You may also implement this manually, but this would require also implementing the loss algorithm and stochastic gradient descent for training, so I would recommend against this. |

|  |
| --- |
| **Dataset** |
| The dataset that will be used is the toxic comment dataset (same as HW3-4). This consists of toxic comments, so be cautioned when viewing the dataset.  You can find the training and test sets here: <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>  Note for the word2vec tasks you will only need to worry about the toxic label (toxic=1 or 0).  The training set (**train.csv)** will be used for training the LM, while the test set (**test.csv**) and labels **(test\_labels.csv)** will be used for testing and analyzing your models.  Take time to understand how the training set and test set are laid out. |

|  |
| --- |
| **Final Report** |
| Just like in the previous homework, you will be documenting observations and results in a final report. This is the best way to show your thought process throughout the homework, so it is best to update it as you go along and then refine at the end, rather than try and write all at the end. |

|  |
| --- |
| **MLP Tasks** |
| These tasks are similar to the NB tasks from HW4, however, you are now working with word2vec embeddings as the input and not count vectorizer or tf-idf.  You will create 2 required functions for MLPs (you may create more functions for your own use, but you need to at least create these two as specified):  **train\_MLP\_model(path\_to\_train\_file, num\_layers = 2)**  This method trains a multi-layer perceptron model on some training data and returns that trained model. The training texts should be represented by word2vec embeddings. You may use any pretrained word2vec embeddings you choose. Recall that the input size will affect how much of the input text is able to be sent in to the model. The MLP slides had some possible solutions, so you may choose any of these (but you should always note your decisions). **The format for the train file should follow the same format as the training data file!**  **test\_MLP\_model(path\_to\_test\_file, MLP\_model)**  This method tests a trained MLP model on some test file and outputs a test file in the same format as the input test file but with 2 columns added: 1) probability of that text being toxic, 2) class prediction (toxic, not toxic). **The format for the input file should follow that of the test file.**  Once these functions are implemented use them to accomplish the following tasks.   1. Train a 2 layer MLP model on the entire train set. 2. Test the trained model on the test set and produce predictions for all test texts. 3. Repeat 1 and 2 for a 1 layer MLP and a 3 layer MLP. Compare and contrast the overall accuracies for the models. 4. Write the analysis and comparison observations in a report document to be handed in alongside code. Note that even if changing something causes the accuracy to drop, this should be reported along with your reasoning of the drop/improvement in accuracy. This report is how you can demonstrate that you are thinking through the problem and give you practice for your final project.   **Please take time to understand how sklearn represents multiple layers! If you are getting the same results with more or less layers you make be making an implementational mistake!!!** |

|  |
| --- |
| **Additional Tips/Guidance** |
| 1. The accuracy you achieve will not determine your final grade for this homework. Rather your thoughtfulness in approaching and analyzing your models. This means that lack in analysis will receive lower scores as indicated in the grading scale. **However, not addressing errors in implementation will cause loss of points. (e.g. if you get the same results for different layers, you probably are making a mistake!)** 2. Note that preprocessing the text is up to you. This assignment is not specifically evaluating preprocessing like the first assignment, but here you have a chance to practice any preprocessing for your final project and future assignments. (**You should be doing some preprocessing, or else you’ll be purposely setting your ML model up to fail.**) 3. Get familiar with the gensim and sklearn libraries on your own on small sets of data. Just copying and pasting the code without fully understanding it will end up being detrimental as you may not be able to accurately give proper analysis and observations of results (especially for the MLP model where you are expected to make improvements/changes).    1. Do not just copy and past the sklearn class code. You may use portions from the sklearn classes, but too much unused code from classes will result in lost points due to poorly written code. Experiment with the code and only import what you need. 4. DO NOT SHARE CODE. I have linked sklearn libraries which will be useful for this assignment. You may also come to me to discuss/figure things out. 5. Start the assignment early. You will have 2 weeks to complete this assignment, which is plenty of time if you start early any familiarize yourself with the libraries. **IF YOU WAIT UNTIL THE LAST MINUTE TO START, YOU WILL BE LESS LIKELY TO DO WELL.** I won’t be granting any additional extensions so each late assignment with follow the late grade policy. 6. Ask questions/approach like a researcher. Think like this is your chance to explore NLP models and analyze their effectiveness. |

**Grading**

Assignment will be graded as follows:

|  |  |
| --- | --- |
| **Description** | **Points** |
| Code Runs | 5 |
| MLP Implementation/Tasks | 20 |
| Report analysis and observations | 15 |
| Documentation (Comments, functions, etc) | 10 |
| **Total:** | **50** |