

CS59000: Machine Learning for Natural Language Processing

HOMEWORK 1

Due: Oct 05, 2017 on Thursday

1 Programming: Sentiment Analysis

In this part, you are required to write programs for sentiment analysis for the given data set.

1.1 Models

You are required to implement two models: **Multi-Layer Perceptron** (MLP, or three-layer neural network) https://en.wikipedia.org/wiki/Multilayer_perceptron and **Perceptron** <https://en.wikipedia.org/wiki/Perceptron>.

The instructor wishes the students to have a good understanding on neural networks, so you are expected to calculate the gradients of the loss functions by hand and implement them. Therefore, in this homework only `numpy` and `scipy` are allowed as external python packages.

1.2 Dataset

You will use the data set prepared by us for this homework. “*sentences.txt*” as the sentences and “*labels.txt*” as the corresponding polarity labels, while “1” for positive and “0” for negative. The order of these two files are corresponding. You need to split the data set for training, validation and testing by yourself. Typically, the proportion respectively are 60%, 20% and 20%. There will be a different reserved testing data set (which we intentionally take out from the data set) used for “real” testing during the grading, named as “*test_sentences.txt*” and “*test_labels.txt*”. And we will rank your model performance among the students taking this class. Those who rank in top 5 will have 10 bonus points, and those who rank in top 10 to top 6 will have 5 bonus points. The model performance is measured by looking at the F1 score, which is the harmonious mean of precision and recall. During grading, we will look at the higher score between your MLP and Perceptron to determine the final performance of your models.

1.3 Features

You are required to extract features by yourselves. Features can be but not limited to *bag of words*: *uni-gram*, *bi-gram*, *tri-gram*, etc.. You can even design new features by yourself. But remember, more features mean more model parameters, which imply slower speed of your program, and possible over-fitting. Your program **must** finish running within a reasonable time (Say around 3 minutes).

1.4 Hyper Parameter Tuning

You can exploit your model by exploring different hyper parameters. For example, the learning rate, the numbers of hidden nodes in the MLP, etc.. Also you can try different gradient methods.

1.5 Tips

For Perceptron, a convenient way for parameter update is to use a list or a dictionary in python. For reference of the gradients calculation, you can refer to this website: http://deeplearning.stanford.edu/wiki/index.php/Backpropagation_Algorithm.

1.6 Starter Code

The file “hw1.py” has been provided as the starter code for your implementation. If you are not comfortable with python class, you can use functions to replace it. The evaluation function has been written for your convenience.

1.7 Submission

Please follow the instructions given in HW0 for submission, including codes and report.