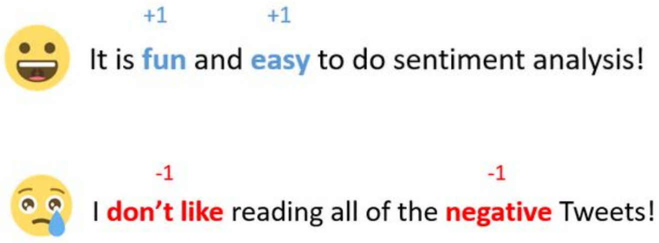
ASSIGNMENT 4

Subject : Sentiment Analysis With Deep Learning
Handed Out : 18.04.2018
Due date : 09.05.2018

Please submit your solution (code and a README file) by 17:00 pm on the due date. Please describe your code in detail in README file.

Introduction

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. An example of sentiment analysis is given below:


+1
+1
It is fun and easy to do sentiment analysis!
-1
-1
I don't like reading all of the negative Tweets!

In this assignment, you will implement a multilayer perceptron (MLP) in Tensorflow. An example structure is given in Figure 1.

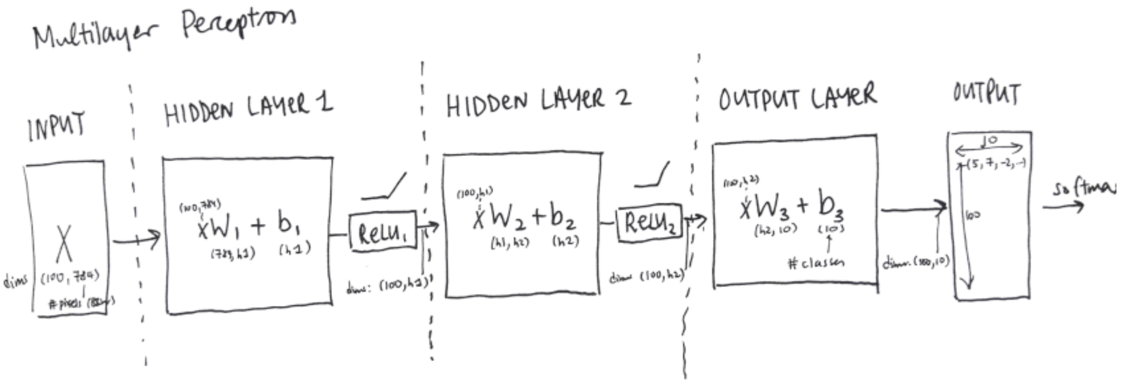


Figure 1: An example structure of Multi Layer Perceptron

Problem Definition

In this assignment, you will do sentiment analysis on a given set of sentences. Your model will learn to classify the given sentences whether they are positive or negative. First, you will train your model on a training set that involves annotated positive and negative sentences, then you will test your model on unannotated sentences which will produce the positive or negative labels for each sentence.

What is Tensorflow?

Tensorflow is an open source machine learning library. It has been developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization for the purposes of conducting machine learning and deep neural networks research.

What is Compositional Semantics?

Compositional semantics studies the meaning of a phrase or a sentence when different units are combined together in order to build larger units. Semantics deals with the meaning of different units such as 'hot' and 'dog' separately in a phrase, whereas compositional semantics deals with the larger unit, that is 'hot dog' as a phrase. Several methods have been proposed to induce the meaning of a phrase. One of the distributional methods is to define the compositional meaning of a phrase in terms of the vectors of smaller units in that phrase. For example, the meaning of a sentence can be expressed in terms of the vectors of each word in that sentence. Several functions have been proposed to capture compositionality, such as addition, multiplication, or weighted addition of the word vectors that the sentence involves.

Multilayer Perceptron

In this assignment, you will implement Multilayer Perceptron with two hidden layers. Input layer is given a 200 dimensional vector that is summation of the word vectors in the sentence. Output layer is composed of two outputs (we have positive and negative sentences). Number of neurons in the first and second hidden layers are 100. You will use ReLU as the activation function in hidden layers and for the output layer, softmax cross entropy will be used. You can change the learning rate hyperparameter (learning_rate=0.001). As the optimization method, you will use Gradient Descent.

Experiments with your architecture - Bonus

You can try different parameters or architectures to obtain a higher accuracy for this task. For example, you can change the learning rate, training epoch or the number of neurons in the hidden layers, or even change the number of hidden layers in the neural network architecture.

You can also try different compositional semantics methods from the literature. For example, you can take the weighted average of the word vectors in a sentence or you can use the summation of the vectors of the words in the sentence by filtering out the stop words.

The rest is left to your imagination. The student who will obtain the highest accuracy on our test set (which will be different than the given test set), will be rewarded with a bonus point. The amount of bonus points will be decided later on.

Dataset

You will be given three input files: positive sentences, negative sentences [1] and pre-trained word vectors file.

To create input of the MLP, you will read the vectors.txt and extract word vectors. In each line a word is separated with colon **(:)** from vectors and each vector is separated with space. You will read positive and negative files (you do not need to convert words to lowercase.

You will remove punctuation and create your label based on two classes. After reading dataset, you will shuffle data and first % 75 examples (the percentage is taken from the command line) will be used as train data and the rest of them will be used as test set. You will print out the accuracy at the end of the code. The accuracy of the model will be calculated by the proportion of the correct labels produced by the model to the total number of reviews.

Notes

- Do not miss the submission deadline.
- Compile your code on *dev.cs.hacettepe.edu.tr* before submitting your work to make sure it compiles without any problems on our server.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate, very similar assignments or code from Internet are going to be considered as cheating.
- You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza. You cannot share algorithms or source code. All work must be individual! Assignments will be checked for similarity, and there will be serious consequences if plagiarism is detected.
- You need to implement **Python** (Python 3). Please submit your source codes and README file in the following submission format.
- You will be graded not only for the output, but also readability, comment lines and README.md.
- I will run your programs from the command line as following. Any other command line format will not be accepted!
  Python

```
python3 assignment4.py positive.txt negative.txt vectors.txt 75
```

```
→ <studend id>
    → code.zip
    → README.md
```

[1] https://github.com/mertkahyaoglu/twitter-sentiment-analysis