DSGA1011: Assignment #1

Haonan Tian
ht1151@nyu.edu

New York University — October 10, 2018

Introduction

This is a report to a natural language processing program which performs sentiment analysis based on movie reviews.

The movie reviews applied in this program are all from IMDB dataset which consisting 50000 movies. Half of these movies are used as testing set and the other half of the movies are split into a training set with 20000 movies and a validation set with 5000 movies.

The program has two versions. The basic version established a binary classifier to classify the sentiment scores into positive or negative. The advanced version established a model which tries to classify movie reviews according to their labeled sentimental scores from 0 to 10.

For both versions, standard data preprocessing techniques are applied to clean the input data. All input movie reviews are tokenized with punctuations removed. A vocabulary is established which contains specified number of most frequent tokens. Then two lookup tables are set up so that the tokenized datasets can be converted into datasets represented in indices which are ready to serve as input to machine learning models. Finally, a bag-of-words model is set up to convert input tokens by the embedding layers and then the results are fed into a logistic regression model to perform classification.

The advanced version of this program is designed to predict the rating of movie reviews with scores range from 0 to 10. The files are loaded according to their scores and the label sets are established correspondingly. The model used to perform the prediction is a two-layer neural network which applies Relu and Softmax activation functions.

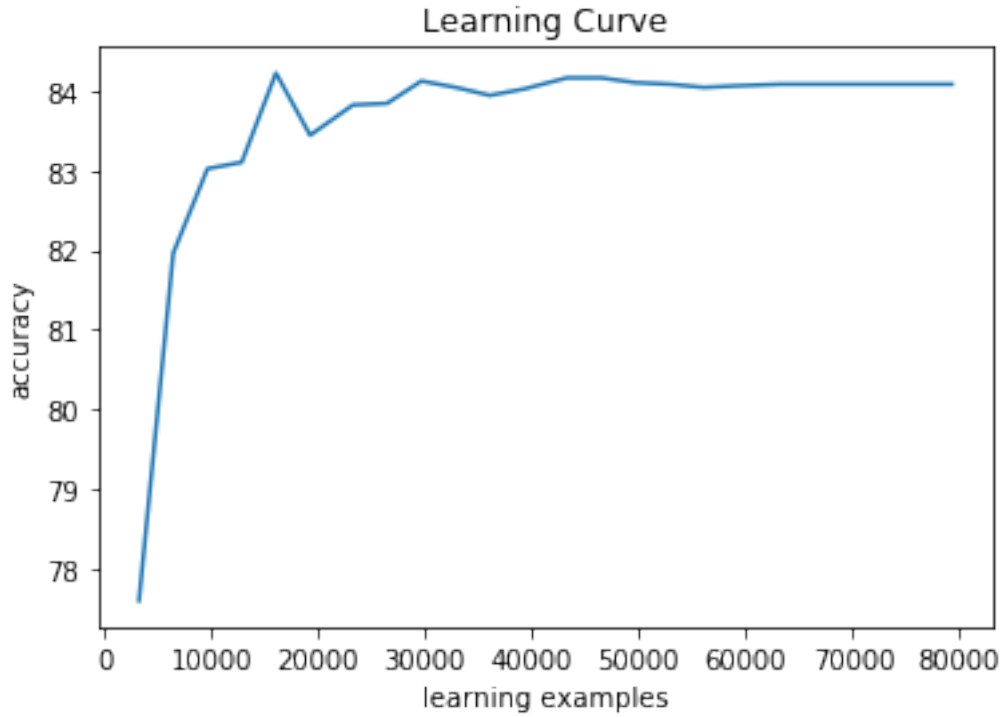
In order to detect the influence of different choices of hyper-parameters to the general accuracy of the model, different tokenization schemes are applied. In this program, there are two versions of tokenization which are named as basic tokenization and advanced tokenization. The basic tokenization applies SpaCy package to tokenize the movie reviews with only punctuations removed while the advanced tokenization applies NLTK to have punctuations and stop words removed from tokens. The advanced tokenization also applied word stemming to unify the format of words.

Examples about the correct classifications and incorrect classifications are recorded in the file three correct reviews.txt and three incorrect reviews.txt. As can be seen from the above tables , the best performance of the model occurs when the hyper-parameters are tuned to the following settings:

1.Vocabulary Size: 20k
2.Number of Grams: 1
3.Learning Rate: 0.01 with annealing
4.Embedding Dimensions: 100
5.Tokenization scheme: Advanced

With these settings of hyper-parameters, the accuracy of model used on the test set is 0.86536

The learning curve on the validation set by using the above parameters are shown as follows:



all code can be found at https://github.com/haoNTT

Results

The hyper-parameters tested in this programs are:

1. Different tokenization schemes: basic tokenization and advanced tokenization
2. N-gram tokenizations: N ranges from 1 to 4
3. Vocabulary sizes
4. Embedding dimensions
5. Learning rate: different choices of values or weather the annealing is applied
6. Optimizer: Adam optimizer or SGD optimizer

1 Results

Table 1: Basic 1-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	No	0.8356
15k	200	ADAM	0.01	No	0.828
15k	100	SGD	0.01	No	0.6248
15k	200	SGD	0.01	No	0.6642
20k	100	ADAM	0.01	No	0.8314
20k	200	ADAM	0.01	No	0.8288
15k	100	ADAM	0.01	Yes	0.8476
15k	100	ADAM	0.03	Yes	0.8432
15k	100	ADAM	0.1	Yes	0.8354

Table 2: Basic 2-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	No	0.817
15k	200	ADAM	0.01	No	0.8222
15k	300	ADAM	0.01	No	0.819
15k	100	SGD	0.01	No	0.506
20k	100	ADAM	0.01	No	0.8356
20k	200	ADAM	0.01	No	0.8228
20k	100	SGD	0.01	No	0.5642
20k	100	ADAM	0.01	Yes	0.8412
20k	100	ADAM	0.03	Yes	0.8329

Table 3: Basic 3-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	No	0.817
15k	200	ADAM	0.01	No	0.8222
15k	100	SGD	0.01	No	0.521
15k	200	SGD	0.01	No	0.554
20k	100	ADAM	0.01	No	0.8286
20k	200	ADAM	0.01	No	0.8228
20k	100	ADAM	0.01	Yes	0.839
20k	100	ADAM	0.03	Yes	0.8298

Table 4: Basic 4-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	Yes	0.7222
15k	100	ADAM	0.01	No	0.728
15k	100	SGD	0.01	Yes	0.5016
15k	100	SGD	0.01	No	0.4998
20k	100	ADAM	0.01	Yes	0.7252
20k	100	ADAM	0.01	No	0.7268
20k	100	ADAM	0.03	Yes	0.7374

Table 5: Advanced 1-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	Yes	0.8422
20k	100	ADAM	0.01	No	0.7918
20k	100	ADAM	0.01	Yes	0.843
20k	200	ADAM	0.01	Yes	0.843
20k	100	ADAM	0.1	Yes	0.8222
20k	200	SGD	0.01	Yes	0.5208

Table 6: Advanced 2-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	Yes	0.8168
20k	100	ADAM	0.01	No	0.8152
20k	100	ADAM	0.01	Yes	0.8246
20k	200	ADAM	0.01	Yes	0.8188
20k	100	ADAM	0.1	Yes	0.812
20k	200	SGD	0.01	Yes	0.5128

Table 7: Advanced 3-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
15k	100	ADAM	0.01	Yes	0.7631
20k	100	ADAM	0.01	No	0.7346
20k	100	ADAM	0.01	Yes	0.786
20k	200	ADAM	0.01	Yes	0.7722
20k	200	SGD	0.01	Yes	0.514

Table 8: Advanced 4-gram Tokenization Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
20k	100	ADAM	0.01	Yes	0.742
20k	300	ADAM	0.01	Yes	0.7512

Table 9: 1-gram Tokenization for Score Rating Result Table

Vocabulary Size	Embedding Dimensions	Optimizer	Learning Rate	Learning Annealing	Accuracy
30k	100	ADAM	0.01	Yes	0.4028
30k	200	ADAM	0.01	Yes	0.3960