Sentimental Classification Over Products Reviews

Author: Aqeel Labash
Supervisor: Tambet Matiisen
Institute of Computer Science
University of Tartu

26 January 2016

Abstract This project train Multilayer perceptron Module on reviews and try to predict the rate for other reviews depending on the previous learning.

Introduction

Currently, there are many surveys, opinions, reviews for products, services and variety of things. it's a tiresome job to distinguish between negative and positive reviews.Usually, we are concerned with specific type of reviews like negative reviews to enhance the review subject. A better approach is to predict more levels than two (negative , positive). That will produce more options. Consequently, a level closer to our needs can be selected.
To train our module consisting of bag of words and neural networks reviews with rank submitted by users were used to accomplish this task.

Background

For the task of sentimental analysis there is four main categories [1] :

- Keyword Spotting : classifying based on presence of unambiguous words such as happy , sad , afraid.[2]
- Lexical affinity:detects words effect and assigns arbitrary words a probable "affinity" to particular emotions.[3]
- statistical methods: depends on latent semantic analysis, support vector machines, "bag of words" and Semantic Orientation and other methods from machine learning.[4]
- concept-level techniques : depends on knowledge representation like ontologies and semantic networks[5]

Arabic Language

Below is a list of some characteristics of Arabic language that make it difficult to process it efficiently:

- **Stop words**: As any language, Arabic language has stop words. The problem occur when users write stop words directly connected to a words.
- **Accents:**in Arabic language there are many accents, probably around 22.And, people write as they speak.
- **Stemming:** Each word in Arabic language has a root. Some roots has more than 30 derivative. Linking derivatives with the root is non-trivial and correctness cannot be guaranteed.

Dataset

Collecting Data

The source of the data is souq.com.Firstly, the crawler start crawling over pages to get links and extract items id. After some analysis on the website Architecture and the items id. The reviews were obtained through web service in the website. After that, 1000 thread (response time for each request was around 300 ms to 500 ms) were used to get the data for items from id=0 to 10000000. Consequently, reviews were divided on pages (5 reviews per page). At the end I obtained 465k reviews.

Cleaning Data

To clean the data following measures were taken;

- Remove non-Arabic Reviews
- Remove non identified Unicodes
- Remove Stop Words
- Remove Empty Reviews

After cleaning the data, I end up with 194k review in Arabic language ready to be used in our Module. It was divided into two sets i.e. training set %20 and testing set %80. Previously, the data get shuffled before assigning the reviews to sets

Method

The library Keras in Python was used to achieve this target. First step was creating simple bag of words. Second step was using Multilayer Perceptron as suggested by project documentation (reuters example) with one layer.
The next sections will explain the parameters selection.

Bag of Words

For this task, I used Keras Library for pre-processing text.

Choosing Number of Words The Dictionary size was more than 50k words so because of insufficient memory I tried to get the highest number of words that would work with my memory which was 1900. Those 1900 words were selected depending on the most appeared words in the text(all reviews texts).And, now each review can be represented with a vector of 1900.

Representation

for the bag of words there is many ways to represent the words of the reviews :

- count : repetition of word in review
- freq : count/length of review
- binary: 1 if word exist in review, 0 if otherwise.
- tfidf: $\frac{\text{TF(Term Frequency)}}{\text{IDF (Inverse Document Frequency)}}$
  TF: $log(freq)$
  IDF: $log(\frac{\text{Number of Reviews}}{\text{Reviews has the word}})$[6]

To decide which representation is better; here is a comparison between them. Comparing training accuracy with test accuracy for different representations.

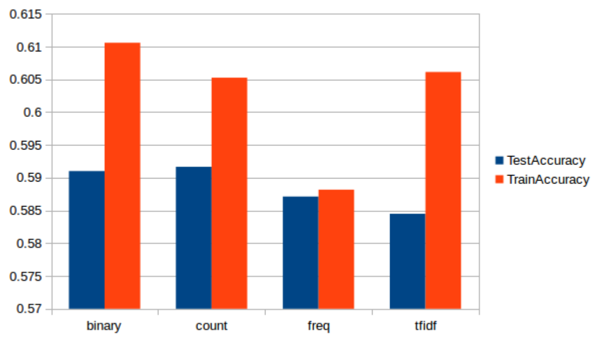

Figure 1: Test and train accuracy for different types of representations

Figure 1 represent that count got the best test accuracy.

Optimizer

For optimization function we have SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax. So I tried all of them and compared the test and train accuracy as shown in the following figure

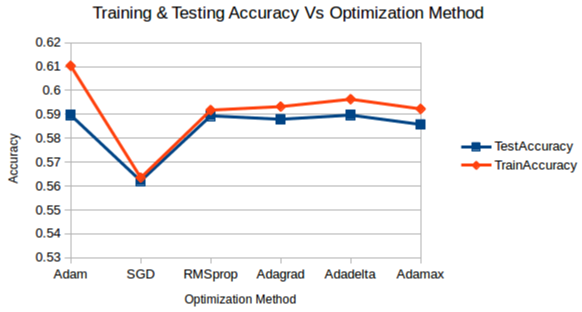

Figure 2: Test and train accuracy for optimizations

We can see that the test accuracy for Adam and Adadelta are so close. But since I've been focusing on test accuracy I'll continue that way by choosing Adadelta for optimization.

Activation

For activation, these functions was tested : Softmax, Softplus, Relu, Tanh, Sigmoid, Hard_sigmoid, Linear.Acurracy comparsion among them in Figure 3

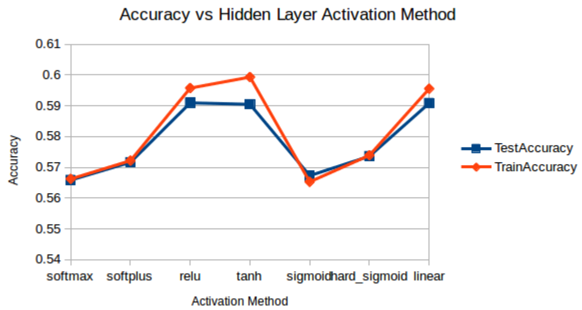

Figure 3: Test and train accuracy for hidden layer activation method

Figure 3 shows that relu got the highest test accuracy.

Dropout

For dropout I tried values from 0.1-0.6

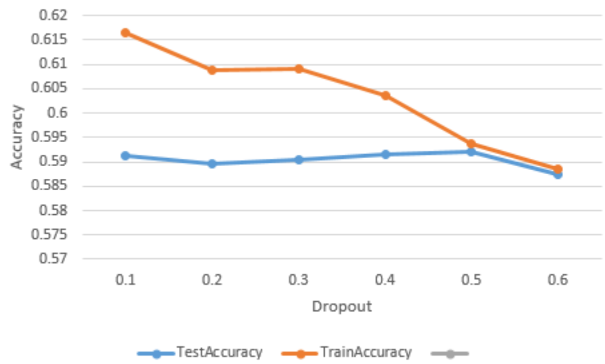

Figure 4: Test and train accuracy for different dropout values

The highest test accuracy was on 0.5 with very slight difference from 0.1.So I went with 0.5.

Layer Size

For layer size the values : 1,2,4,8,16,32,64,128,256,512 been tested on our module and the result as in Figure 5.

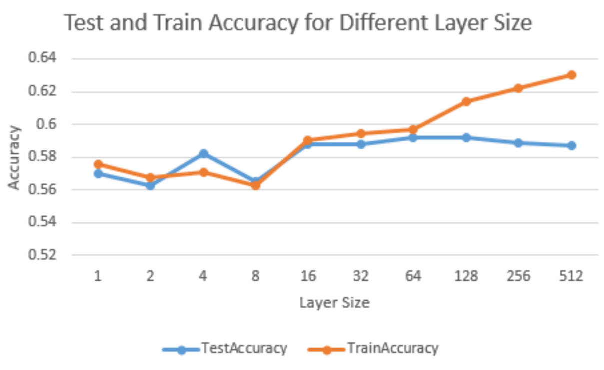

Figure 5: Test and train accuracy for different layer size

Maybe it's hard to see that 128 gave slightly better results so it seemed the layer size to choose.

Batch Size

For batch size the values : 1,2,4,8,16,32,64,128 were tested and the results in figure 6

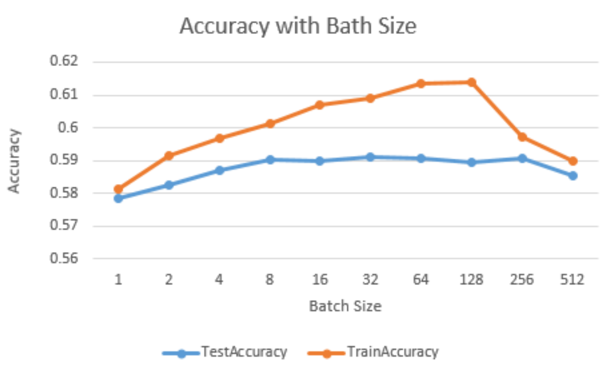

Figure 6: Test and train accuracy for different batch sizes

We can notice from figure 6 that we have a slight curve when the batch size is 32. Which to be the best choice for this step.

EPOCH

To choose Epoch and avoid over fitting the values: 1,5,10,20,35 been checked.There was no need to continue testing values after 35 because even the training Accuracy kept raising the test accuracy kept decreasing as in figure 7

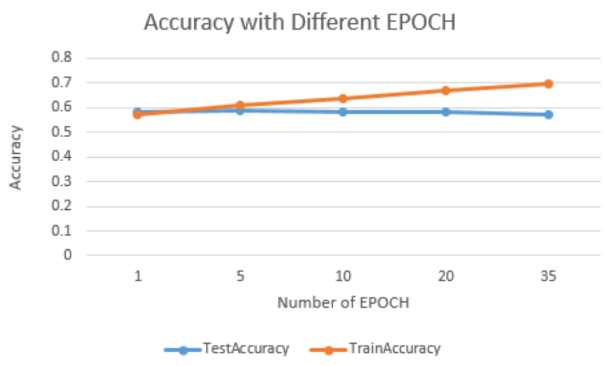

Figure 7: Test and train accuracy for different number of EPOCH

Figure 7 reveals that EPOCH 5 givs us the best accuracy without over fitting.

Results

By the end of project the final parameters and functions used :

- Top 1900 words as dictionary.
- Batch size : 32.
- EPOCH : 5.
- Word representation in bag of words : count.
- Optimization function : Adadelta.
- Nodes in hidden layer :128.
- Dropout :0.5
- Activation function for hidden layer : relu.
- Activation function for output: softmax.
- Loss function : categorical_crossentropy.

The outcome of using those parameters and functions lead to test accuracy %59 and %61 as train accuracy.

Discussion

In general the model performed well and got some how acceptable results considering the amount of data used. Choosing the training set has an effect around (0.02 - 0.1) on the test accuracy.Right now training set is selected randomly.So choosing training set with some certain rules will increase the overall accuracy.

Future work

Actually, there are many things we can do and compare the results with what we have.

- Remove diacritics can minimize the dictionary size and give a better estimating for the words we should use.
- Stem the words, in Arabic language each word has a root and some root words has more than 30-40 leaf.

- Apply a new method specifically this method nlp.standford
- Choose the training set with certain rules.

Conclusion

Utilizing the 192K reviews with simple bag of words, multilayer Perceptron as the module for neural network,optimizing the functions and parameters the accuracy for testing was %59, which is some how acceptable. However, the accuracy could be improved if future work implemented.

References

[1] Cambria, Erik; Schuller, Björn; Xia, Yunqing; Havasi, Catherine (2013). "New Avenues in Opinion Mining and Sentiment Analysis". IEEE Intelligent Systems 28 (2): 15–21. doi:10.1109/MIS.2013.30

[2] Ortony, Andrew; Clore, G; Collins, A (1988). The Cognitive Structure of Emotions (PDF). Cambridge Univ. Press.

[3] Stevenson, Ryan; Mikels, Joseph; James, Thomas (2007). "Characterization of the Affective Norms for English Words by Discrete Emotional Categories" (PDF). Behavior Research Methods 39 (4): 1020–1024.

[4] Kim, S. M.; Hovy, E. H. (2006). "Identifying and Analyzing Judgment Opinions." (PDF). Proceedings of the Human Language Technology / North American Association of Computational Linguistics conference (HLT-NAACL 2006). New York, NY.

[5] Cambria, Erik; Hussain, Amir (2012). Sentic Computing: Techniques, Tools, and Applications (PDF). Springer.

[6] https://en.wikipedia.org/wiki/Tf–idf