Report for Homework 2: Sentiment Analysis

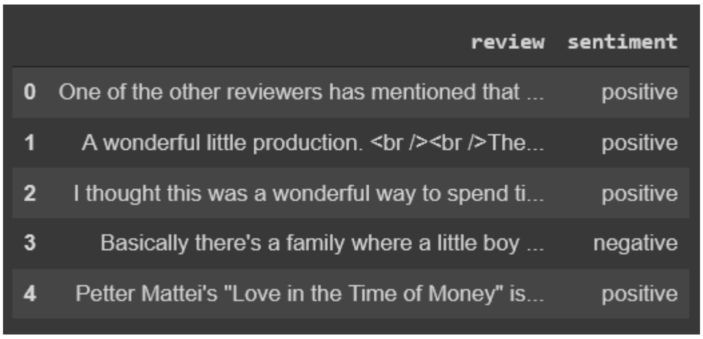
Giovanni Giliberto
Department of Economics and Business, University of Catania
Knowledge Discovery Course
giovanni.giliberto18@gmail.com

5th of June, 2022

1 Dataset Description

The dataset that has been used for this analysis is the "IMDB Dataset" and it can be found at the following link: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews The database contains 50.000 reviews of different films coming from IMDB website. It is a 50.000 x 2 dimensions database, one column contains the text of each single review while the other column contains the label for that particular review. Lables can be either "positive" or "negative" meaning that the review expresses a positive opinion about the film or a negative ones. Labels will then be converted to 0 or 1 such that:

- **Positive: 1**
- **Negative: 0**



	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Figure 1: Dataset preview

The goal of this work is to build a RNN capable of forecasting the sentiment (label) of a given review.

2 Data Pre-Processing

In order for the dataset to be usable for the above-stated goal it is necessary to compute some pre-processing. First of all, after downloading and unzipping the whole database, it has been transformed to a numpy dataframe in order to separate the two columns and work with them separately. The reviews' column has been separated and assigned to a new dataframe. At the end of each review it has been added a symbol, "/n", in order to create an anchor point (more on that later). The whole list of objects (reviews) is then joined together in order to have a single text file. All the text is then made lowercase and, after removing the punctuation, the text is divided again in single reviews using the symbol "/n" at the end of the review as a "split point". After this procedure, it is necessary to build the vocabulary containing each unique word which can be found inside all the reviews. Then each word is substituted with a number so the whole vocabulary is "tokenized", and each review is then clipped in order to set each of them at the same length. Those are crucial steps in order to transform the data into tensors.

3 Splitting

All the fifty thousand reviews are divided into three sub-set: Train set, validation set and test set. The initial configuration established 80 percent of observations for the train set and the remaining 20 percent equally divided between validation and test. Nonetheless I obtained higher results splitting the dataset in this way:

- **Train set= 70 percent**
- **Validation set= 20 percent**
- **Test set= 10 percent**

Training set: 70%
Validation set: 20%
Test set: 10%

Figure 2: Splitting

4 Model Description

The model is basically a Recurrent Neural Network. This specific RNN used for sentiment analysis is the so-called ”Long short term memory” network which is explicitly designed to ”remember” information for long periods. The LSTM model can select which information is needed (and retained) and which one has to be discarded. This workflow is made by means of cell states which contain specific information needed in order to decide which one has to be retained and which one has to be discarded.

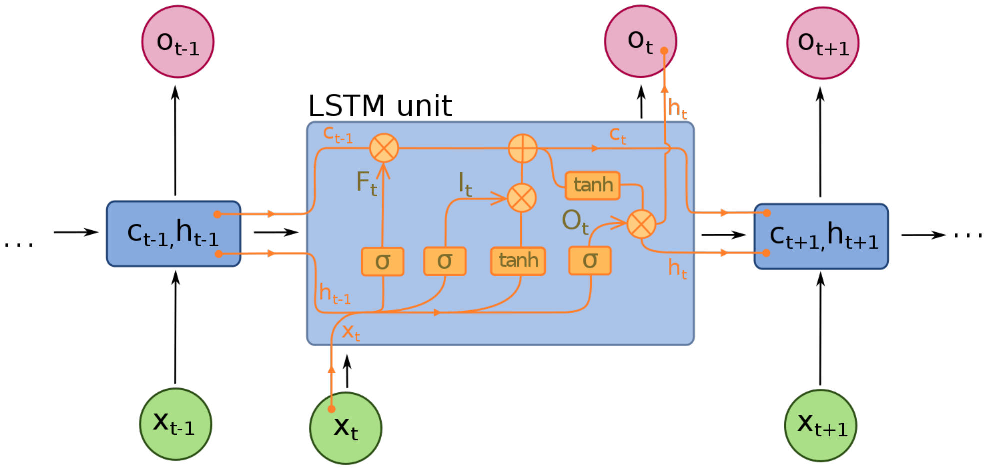


Figure 3: Structure of a LSTM network model

5 Training procedure and Confusion Matrix

Some problems arose during the training procedure, in the sense that the model’s performance where really low, not even 50 percent accuracy in the test set. In order to improve the model’s predicting ability the embed size has been increased but without any tangible improvements unfortunately. Subsequently the batch size has been increased to 128 and all the letters which were uppercase has been transformed into lowercase. After a 20 epochs training the final results are:

The confusion matrix shows the following results for each class:

- **Train loss:** 61 percent

- **Train accuracy:** 66 percent
- **Validation loss:** 60 percent
- **Validation accuracy:** 67 percent
- **Test loss:** 60 percent
- **Test accuracy:** 67 percent

The confusion matrix shows that the model is able to better predict positive reviews compared to negative ones. In fact, the model accuracy for label 1 (positive reviews) is 65 percent while for label 0 (negative reviews) it drops to 57 percent.

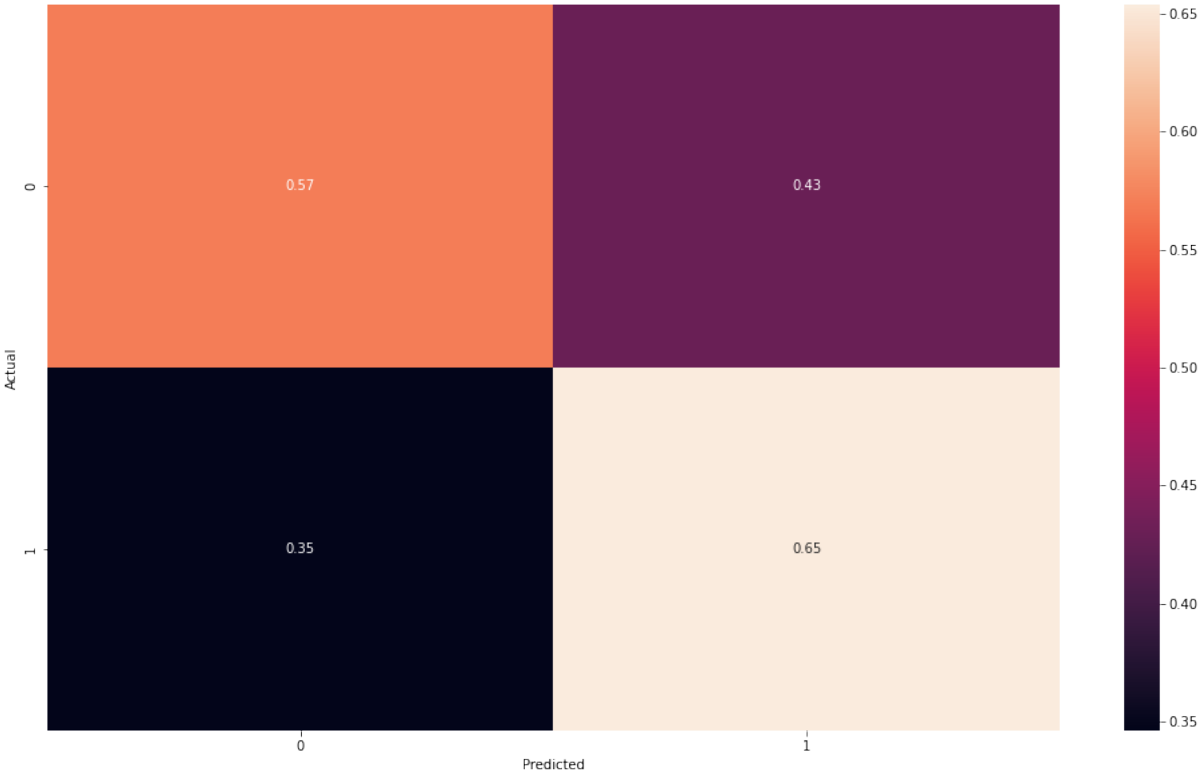


Figure 4: Confusion Matrix