Deep learning for sentiment analysis

Amine Biad, Ghiles Sidi Said, Mohamed Ali Darghouth and Walid Belrhalmia

I.Introduction: Why sentiment analysis?

Sentiment analysis task have attracted more and more attention during the last years. From an economical point of view and for a business manager for instance ,it is critical to use any information to improve the company services and products. For instance it is interesting to identify the mood of a customer based on his reviews and feedback to improve the business experience for other customers, fix what went wrong and improve what is working well. On the other hand identifying the mood of a person in real time based on its tweets and comments in facebook would be interesting to display the right ad content in the right moment. It doesn"t make sense to display an add for holidays in south Africa for a person who just twitted about his Christmas eve. To do so, there is a need to build and deploy real time models for sentiment analysis based on textual inputs.

Illustration 1: Google search using Sentiment analysis as keywords from 2005 to 2017

Sentiment analysis is challenging due to the fact that humans language is too rich. On the other hand, words can have different meanings in different context. Besides, it is hard to identify when words are used in a sarcastic way. One other barriers for this task is that emotions and opinions are expressed in very different ways depending on the culture and the language.

The work is organized as follows: we will present our objectives, then we will present the methods we will be using for the classification tasks and finally we will study the impact of the several parameters on the models performances .

II. Data set description

We used two datasets for this project.

III. Goals

The goal from this project is to:

1. Evaluer les performances des diffetents modèles
2. Study the impact of the parameters on the quality of the results
3. Try this model on amazon reviews dataset
4.Explaining the results .

2.Models

2.1. Convolution neural networks for text classification

The model we will be using is inspired from Mr kim’s work. The idea behind this paper is to represent the input data , texts for our case , into another space representation then to apply on this space representation multiple convolution filters, with different parameters in order to catch different features from the text. The outputs of the filters are then fed into a maxpooling layer. Depending on the number of classes, the output of the maxpooling layer is then fed into a suitable output layer for the problem.

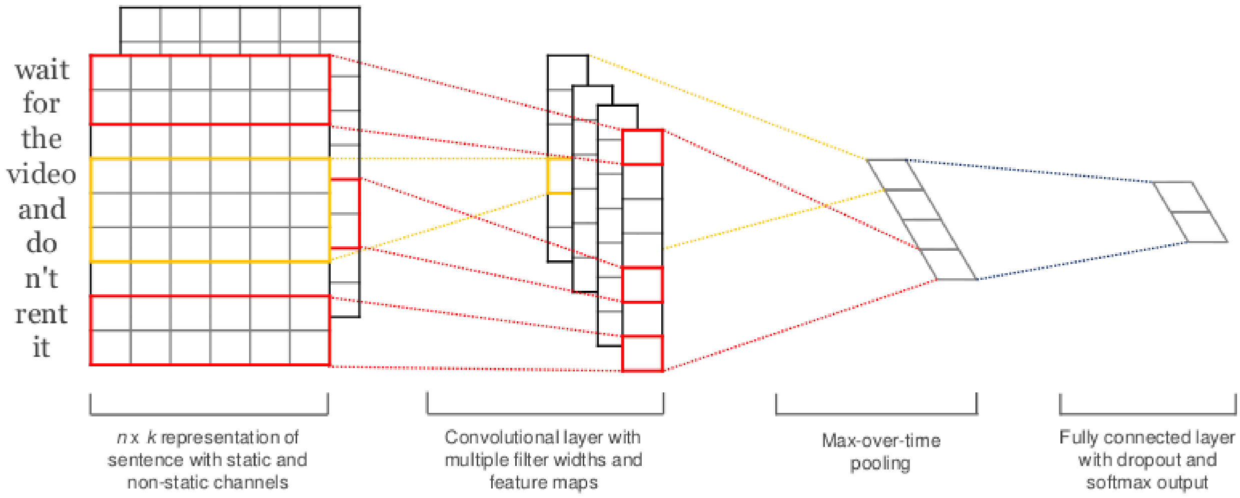

Illustration 2: Convolution neural network for text classification

We decided to use this model since convolution will be usefull to highlight the ngrams .

2.2 .Recurrent neural networks

Text is by defintion a sequence of words. In order to catch this sequence structure and take it into consideration, we suggest to use a simple recurrent neural network.We will refer to recurrent neural networks with RNN in the following.

2.3. Simple Long short term memory

Long short term memories (LSTM) are a special kind of recurrent neural networks which tries to solve the vanishing gradient problem faced by the RNNs. We will use these models on our task to check if there are differences.

2.4. Bidirectional Long short term memory

Bidirectional Long short term memory is a combination of two LSTMs where one is fed with data in the order of appearances of words in the sentence, the other one is fed with words in the reverse order. This combination

2.5. Bidirectional LSTM

We used also a bidirectional LSTM to check the differences .A bidrectional LSTM is based on two LSTMS. One is fed with data in the order of appearances of words in the sentence, the other one is fed with words in the reversed order. It is interesting to test this kind of neural networks since the discriminative information can be on the last layers of the neural network.

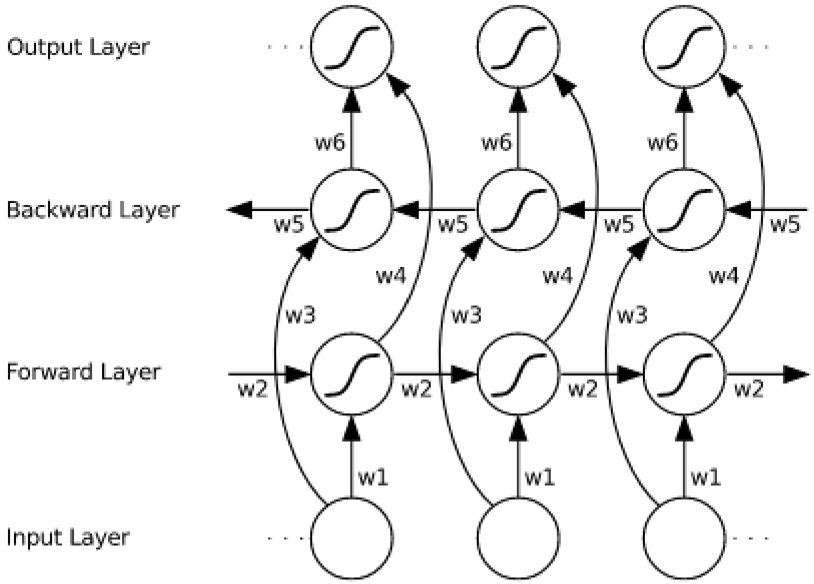

Illustration 3: A simple representation of Bidirectional LSTM

3.First experiments:
3.1.Data description
We trained the previously described models on the first dataset. The measured score for evaluation is the accuracy.
3.2. Experiments:

For the CNN we used 3 convolution filters of size 128 and kernel size 3,4 and 5. we used also with these filters respectively the functions relu, sigmoid and tanh.

The implementation was done using Keras 2.0.8 running on top of Tensor Flow and using Python 3. For the hardware we used an Nvidia Geforce 1050 GTX.
To accelerate the training process, we used 500.000 tweets and 7 epochs. We fixed also a maximum length for each tweet of 75 words. We used as input for each network an embedding matrix of depth 30 and a batch of 30 tweets. Each model also had a drop out layer of probability 0.2.
Even though we tried to accelerate the training processing, it was particularly slow for the sequential models. For the vanilla LSTM for instance, it took us 3 hours to finish the training phase.
Since this is a binary classification problem, we chose the accuracy as metric to evaluate the performance our models.

The following table summarizes the outcome of our work so far:

Model	Accuracy
CNN	0.7924
RNN	0.788
LSTM	0.7572
Bidirectional LSTM	0.7369

What is interesting to note is that the bidirectional LSTM is slightly worse than the LSTM for this application from an accuracy perspective.

4.Conclusion

This work is far from over. We will have to reach what we defined in the objectives.

5.References:

https://www.linkedin.com/pulse/importance-sentiment-analysis-social-media-christine-day/
https://medium.com/@shiyan/materials-to-understand-lstm-34387d6454c1
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
http://www.cs.toronto.edu/~graves/preprint.pdf