

PAO Report for NLP : sentiment analysis

Robin Niel

October 8, 2018

Contents

I	SVM and Neural Network comparaison for sentiment analysis	3
1	Motivation	3
2	Experiences	3
3	Results	3
3.1	SVMs hyperparameters	3
3.2	Second and third datasets comparaison	4
II	Experiments	6
4	Influence of dense layers and LSTM	6
5	Negation propagation	6

Part I

SVM and Neural Network comparaison for sentiment analysis

1 Motivation

We want to compare the two main methods for sentiment analysis to make sure that the neural network I have developed is better at sentiment analysis than SVMs.

2 Experiences

To begin with, I started looking for specialised kernels for NLP. Sadly I couldn't find any ressources online about it, the people were recommending using the linear kernel or the rbf one so that is the two I chose.

First we are gonna choose three models to test and choose from :

- SVM with linear kernel
- SVM with rbf kernel
- Neural network

The choosen neural network will be the best architecture I found during my speciality internship. It is summarised in the following array :

Layer name	Number of neurons	Activation function	Dropout	Filter size
Input layer	50	RELU	0.5	NA
Convolutional layer	64	RELU	0.5	10
LSTM layer	50		0.5	NA
Dense layer	50	RELU	0.5	NA
Dense layer	25	RELU	0.5	NA
Dense layer	2	Softmax	0	NA

Concerning the two SVMs, we need to find good hyperparameters in order to have a honest comparaison. To do so, I will under-sample the 2nd and the 3rd dataset in order to have shorter training time. So for the linear kernel, the only hyperparameter will be the C and for the rbf kernel, the C and the γ .

After finding the best hyperparameters, I will train the different models on the whole 2nd and 3rd dataset and then I will compare them on the same test set. After that I will test the three model on the first datasets in order to have a better idea about the generalization capacity of the different models.

3 Results

3.1 SVMs hyperparameters

For the sub-sampling I took 2000 sentences. I found out that the best C for both SVMs was 1 and that the best γ was 0.1.

3.2 Second and third datasets comparaison

After training the three models, here are the results I obtained for the three models :

Model	Training time (mn)	Test set (accuracy)	IMDB (movie) dataset	Amazon dataset	Yelp dataset	Mean
Linear kernel	23,35	0,7017	0,734	0,681	0,699	0,7046
RBF kernel	85,30	0,5961	0,72	0,714	0,7	0,7113
Neural network	1,69	0,8044	0,841	0,816	0,786	0,8123

Following is the neural network train accuracy and loss.

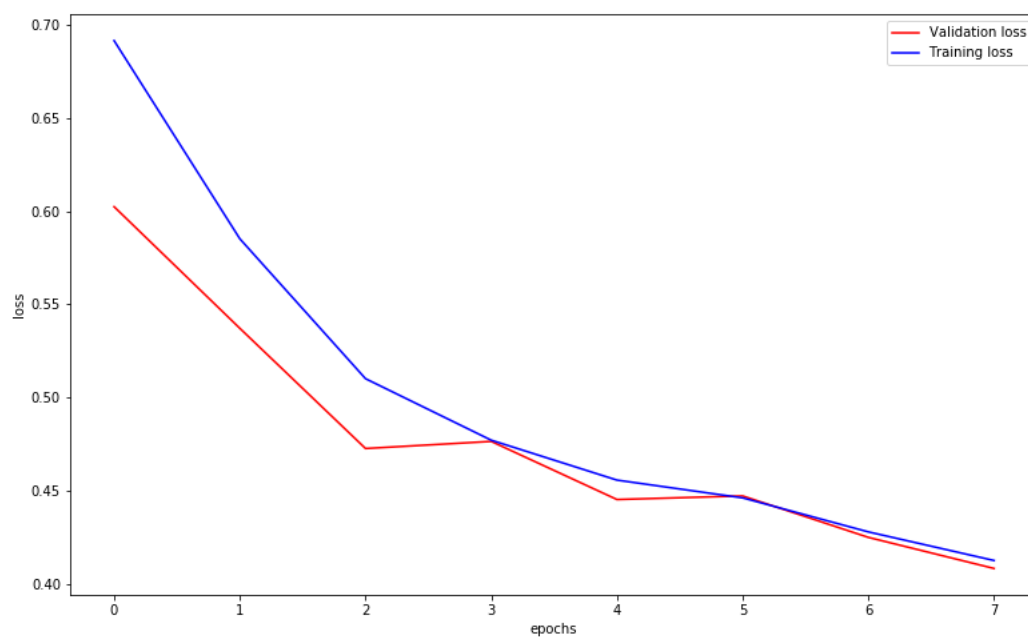
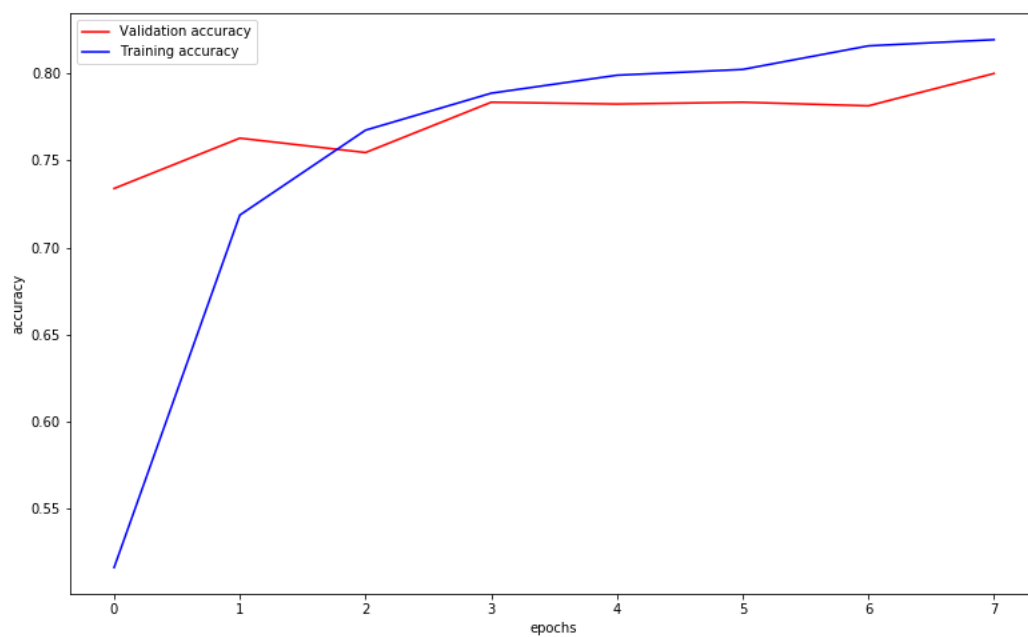


Figure 1: Training accuracy and loss

The first thing that we can notice is that the linear kernel performs better in validation but on a unknown dataset, the rbf kernel tends to perform a bit better. We can also notice that the neural network is performing way better than the two SVMs. We can therefore conclude that the neural network architecture is better than SVMs when it comes to sentiment analysis.

Part II

Experiments

4 Influence of dense layers and LSTM

In this par, we want to explore the influence of the different hyperparameters mainly the number of neurons in the dense layers or the number of LSTM. To do that we follow the same protocol as previously. Here are the results obtained :

Model	Test set (accuracy)	IMDB (movie) dataset	Amazon dataset	Yelp dataset	Mean
Dense {50,50}	0.7966	0.838	0.786	0.793	0.805
Dense {250,250}	0.7995	0.839	0.803	0.792	0.811
Dense {500,500}	0.8024	0.831	0.757	0.776	0.788
Dense {1024,1024}	0.8123	0.823	0.786	0.799	0.802
Dense {100,25}	0.8028	0.825	0.79	0.766	0.793
Dense {500,250}	0.8065	0.841	0.808	0.787	0.811
Dense {1000,500}	0.7986	0.828	0.773	0.767	0.789
Dense {250,250}, LSTM{150}	0.8053	0.808	0.765	0.775	0.783
Dense {250,250}, LSTM{100}	0.8115	0.815	0.757	0.781	0.784
Dense{250,250}, LSTM{50}	0.8094	0.819	0.785	0.79	0.798

For the number of neurons in the dense layers, we can say that the best configuration is the 500,250 or the 250 250.

For the LSTM part, we can observe a decrease in accuracy as the number of LSTM goes higher than 50. The optimal number of LSTM seems to be 50.

5 Negation propagation