Deep Learning
Natural Language Processing Project

Amine KHELDOUNI

January 10^{th}, 2019

Monolingual embeddings

We successfully managed to compute the nearest neighbors of any word of our dataset. Below is an example of word vectors outputs :

First word	Second word	Score
cat	dog	67.17%
dog	pet	68.42%
dogs	cats	70.74%
Paris	France	70.59%
Germany	Berlin	70.6%

TABLE 1 – Embeddings score between two input words

Word	Most similar words
cat	cats, kitty, kitten, Cat, dog
dog	dogs, puppy, Dog, canine, pup
dogs	Dogs, puppies, cats, canine
Paris	Parisian, France, Versailles, Lyon, Bordeaux
Germany	Austria, Bavaria, Berlin, Munich

TABLE 2 – Most similar words embeddings to initial input word

On the other hand, we perform the same operations on sentences using a bag-of-words vectors class :

Type	First sentence	Second sentence	Score
mean	"1 man singing and 1 man playing a saxophone in a concert."	"10 people venture out to go crosscountry skiing."	62.24%
idf	"1 man singing and 1 man playing a saxophone in a concert."	"10 people venture out to go crosscountry skiing."	62.24%

TABLE 3 – Sentence encoding score between two example sentences

Type	Sentence	Two nearest sentences
mean	"1 smiling african american boy ."	"an african american man smiling ." "a little african american boy and girl looking up ."
idf	"1 smiling african american boy ."	"an african american man smiling ." "a little african american boy and girl looking up ."

TABLE 4 – Most similar sentence encoding for an input sentence (in mean or weighted average)

Multilingual word embeddings

Question 1

In this section, we aim to find a mapping W that will map a source word space to a target word space. This can be expressed as the following optimization problem :

$$\underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F$$

Given an orthogonal W, we can decompose the loss function as follows :

$$\begin{aligned} \|WX - Y\|_F^2 &= < WX - Y, WX - Y >_F \\ &= \|X\|_F^2 + \|Y\|_F^2 - 2 < WX, Y >_F \quad (\|WX\| = \|X\| \text{ since } W \in \mathcal{O}_d(\mathbb{R})) \end{aligned}$$

Therefore, minimizing $\|WX - Y\|_F$ is equivalent to maximizing $< WX, Y >_F$.

On the other hand, we operate an SVD decomposition of YX^T ($U\Sigma V^T = SVD(YX^T)$). Then

$$\begin{aligned} < WX - Y >_F &= < W, YX^T >_F \qquad \text{(by orthogonality of } W) \\ &= < W, U\Sigma V^T >_F \\ &= < U^T W V, \Sigma >_F \\ &= \operatorname{Tr}(V^T W U \Sigma) \qquad (\Sigma \text{ is diagonal}) \end{aligned}$$

The matrix $A = V^T W U$ is orthogonal (product of orthogonal matrices). Therefore the diagonal elements of A are bounded in absolute value by 1.

Hence, considering the fact that Σ have non-negative diagonal elements, the maximization of $Tr(A\Sigma)$ implies $A^* = \mathcal{I}_d$. Then, isolating W^* we finally get the formula :

$$W^* = UV^T$$

Sentence Classification with BoV

Question 1

The logistic regression model performs similarly whether we consider the average of word vectors or the weighted-average. The score on the training dataset is only of 49.81% (49.33% for weighted-average), and for the validation set 43.5%.
These results are not satisfying and are genuinely improvable. Therefore, we tried to perform a Support Vector Classification to improve our performances, however the accuracy did not improve that much.

Deep Learning models for classification

Question 1

Since we have a multi-class classification problem, we used for this experiment a categorical cross-entropy loss, which formula is expressed as follows :

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{1}_{\{y_i=k\}} \log\left(p(\hat{y}_i = k)\right)$$

Question 2

Below is our accuracy and loss curves for the Deep classifier over 5 epochs.

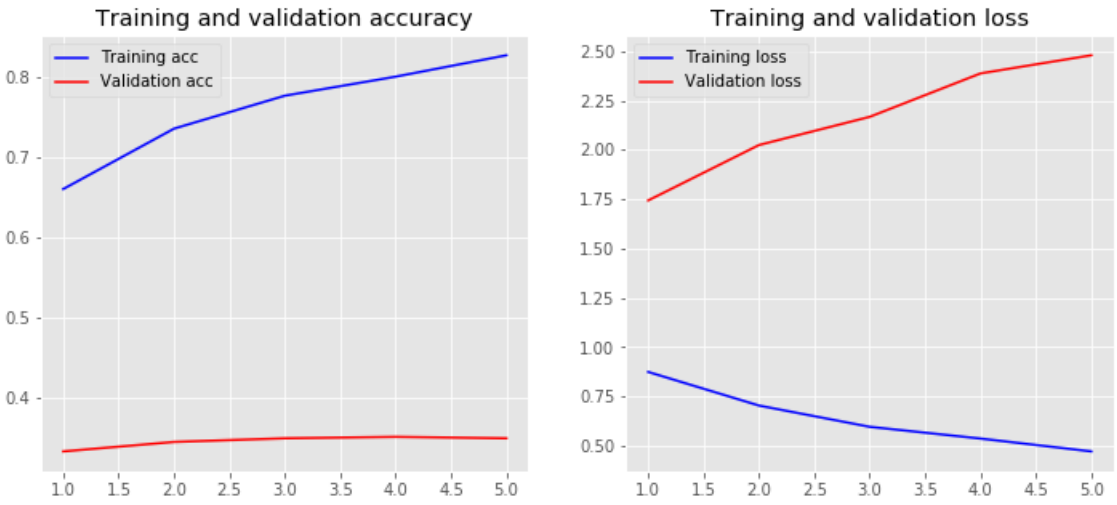

FIGURE 1 – Accuracy and Loss curves for the basic 5-class of words classifier

We notice that the validation accuracy fails to increase as the dev loss keeps increasing, whilst the training loss decreases and maximizes its accuracy (overfitting). This may explain why we did not choose to go further in the episodes, because we are overfitting the training set. Finally, our accuracy holds a value of 91.7% in the training and 34.97% on the test set.

Question 3

Our motivation for seeking a more adapted and working DL model is to increase the accuracy of the validation set. Therefore, the model will succeed more in generalization and predicting unknown or unseen sequences.

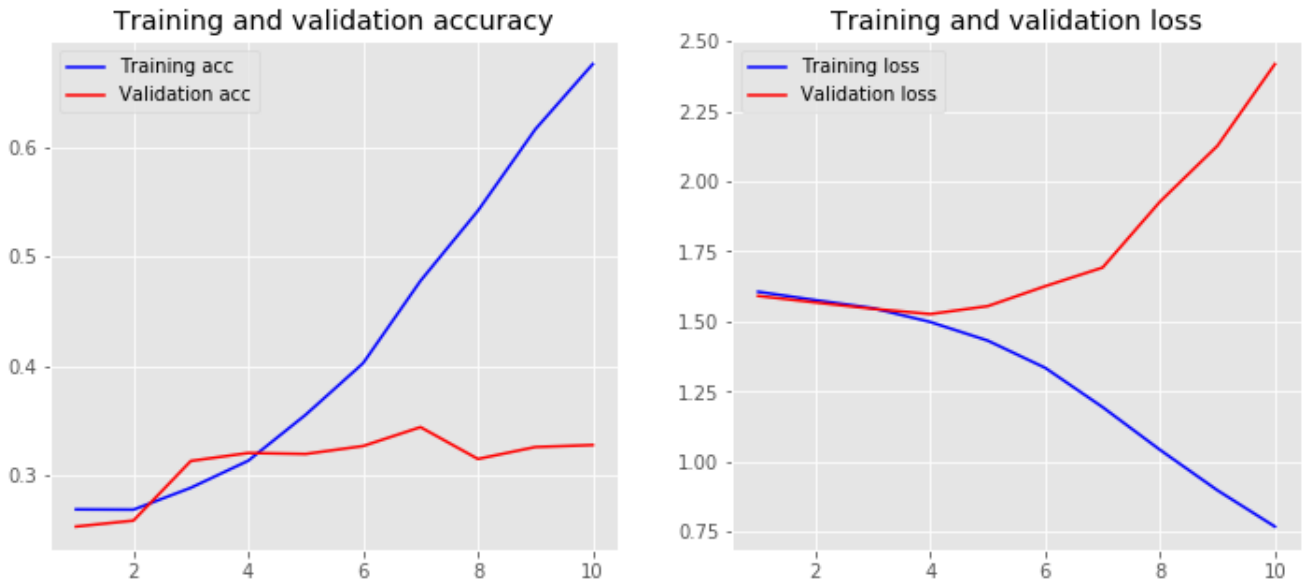

FIGURE 2 – Accuracy and Loss curves for another classification model

We notice that the training loss decreases and the model still overfits as the validation loss keeps increasing after 4 epochs.

We also noticed that it is not much the number of optimization steps which matters to us, but the complexity of our neural network which adapts at first to general data but overfits the training dataset very fast over the epochs.

We end up with a train accuracy of 77% and a test accuracy of 32.8%. Thus, we couldn't improve the model further without overfitting the training set.