Joel Mbouwe
jores-jores.mbouwe-nanmou@student.ecp.fr

Lab session # 1
DEEP LEARNING 2019

02/27/20

1 Multilingual word embeddings

The goal here is to find :

$$\begin{aligned}
W^* &= \underset{W \in O_d(\mathbb{R})}{argmin} \|WX - Y\|_F \\
&= \underset{W \in O_d(\mathbb{R})}{argmin} \|WX - Y\|_F^2 \\
&= \underset{W \in O_d(\mathbb{R})}{argmin} \|WX\|_F^2 - 2\langle WX, Y\rangle + \|Y\|_F^2 \qquad \text{with } \langle A, B\rangle = tr(AB^T) \\
&= \underset{W \in O_d(\mathbb{R})}{argmax} \langle WX, Y\rangle \qquad \text{since } W \in O_d : \|WX\|_F^2 = \|X\|_F^2 \text{ is independent of } W \\
&= \underset{W \in O_d(\mathbb{R})}{argmax} \frac{1}{\sqrt{d}} \langle WX, Y\rangle
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\tfrac{1}{\sqrt{d}}\langle WX, Y\rangle &= \langle W, YX^T\rangle \\
&\leq \frac{1}{\sqrt{d}} \|W\|_F \|YX^T\|_F \qquad \text{by the theorem of Cauchy-Schwartz} \\
&= \|YX^T\|_F \\
&= \|\Sigma\|_F \qquad \text{where } \Sigma \text{ is such that } U\Sigma V^T = YX^T \text{ with } U, V \in O_d(\mathbb{R}) \\
&\leq tr(\Sigma) \qquad \text{since } \Sigma \text{ is diagonal with all its diagonal terms been positives}
\end{aligned}$$

Which means that $tr(\Sigma)$ is a majorant of $\frac{1}{\sqrt{d}}\langle WX, Y\rangle$ for all W $\in$O$_d(\mathbb{R})$
Furthermore such majorant is reached for $W = UV^T$. Indeed :

$$\begin{aligned}
\langle UV^T, YX^T\rangle = \langle UV^T, U\Sigma V^T\rangle &= \langle U, U\Sigma V^T V\rangle \\
&= \langle U, U\Sigma\rangle \\
&= \langle I_d, U^T U\Sigma\rangle \\
&= tr(\Sigma)
\end{aligned}$$

Finally we obtain $W^* = UV^T$

2 Sentence classification with BoW

I trained a a logistic regression and i made a gridsearchCV in order to find the optimal value of the penalization parameter. My logistic regression was slightly better when considering the mean of the words of the sentences without weights. I also implemented a Randomforest classifier. It outputs an accuracy of 0.404.

Accuracy without IDF	0.410
Accuracy with IDF	0.406

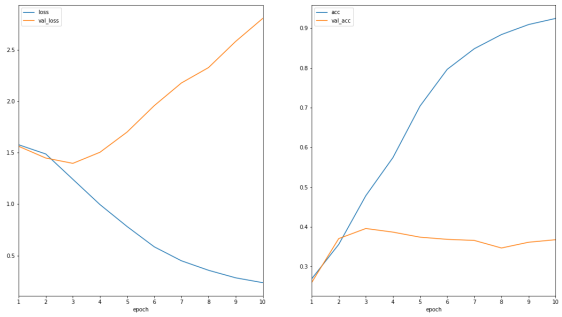
3 Deep Learning models for classification

- For the classification, i use the categorical crossentropy loss:

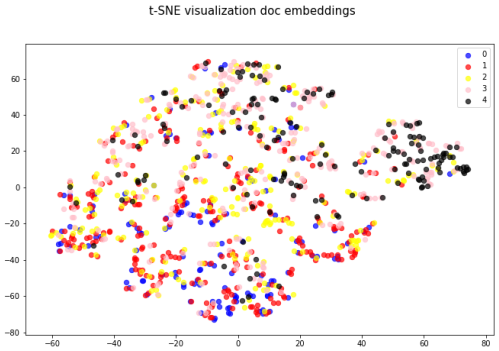
$$Loss = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{5} y_{ij}\log\left(p_{ij}\right)$$

  Where i indexes the samples and j indexes the 5 classes, and y is the sample label ($y_{ij} = 1$ if sample i belongs to class j, otherwise 0) and p_{ij} are the the predicted probabilities of sample i belonging to class j with $\sum_j p_{ij} = 1$ for each i.

- The best accuracy i get with the initial framework was around 0.39. Figures below show the results of the model and a T-SNE of the inner representation of the validation waves .



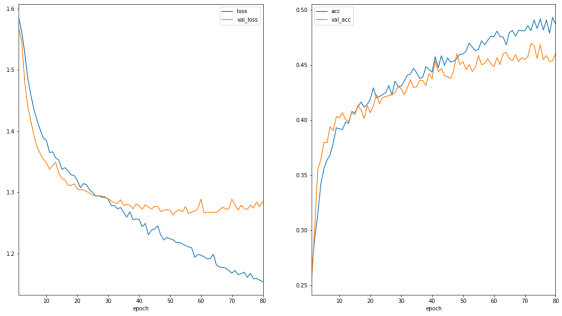
(a) Evolution of train/test loss/accuracy over the epochs



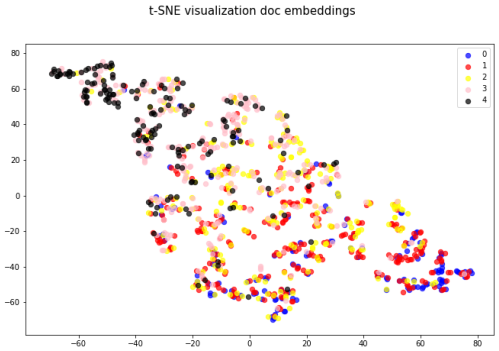
(b) T-SNE of an inner representation of sentences

- My inovation consists in using the embeddings of the word2vec and training a lstm framework with those embeddings. Indeed, the main problem with the previous approach was related to the few amount of data which did not allow the model to learn good representations of words and then performing not quite well on the classification task.

  In order to tackle this problem, i decided to use the word2vec embeddings as the initialization of the embedding layer. I first try to keep training the embedding layer but it impacts negatively the accuracy. I then decided to freeze the embedding layer and only train the following layers. I also increase the dropout rate from 0.1 to 0.3. The figure below shows the new results.



(a) Evolution of train/test loss/accuracy over the epochs



(b) T-SNE of an inner representation of sentences