NLP Project

Lahlali Kenza

January 12, 2019

1 Multilingual word embeddings

Using the orthogonality and the properties of the trace, prove that, for X and Y two matrices:

$W^* = argmin_{W \in O_d(R)} ||WX - Y||_f = UV^T$ with $U\Sigma V^T = SVD(YX^T)$
We will exploit below the orthogonality of W, the SVD decomposition as well $Trace(A^T) = Trace(A)$

Indeed,
$||WX - Y||^2 = Trace((WX - Y)^T(WX - Y))$
$= Trace(X^TW^TWX - (WX)^TY - Y^T(WX) + Y^TY)$
$= Trace(-(WX)^TY - Y^T(WX)) + Trace(X^TX) + Trace(Y^TY)$
$= -2 * Trace(Y^TWX) + Trace(X^TX) + Trace(Y^TY)$
$= -2 * Trace(U^TW^T) + Trace(X^TX) + Trace(Y^TY)$
$= -2 * Trace(\Sigma V^TW^TU) + Trace(X^TX) + Trace(Y^TY)$

Finally, we get: $W^* = argmax_{W \in O_d(R)} Trace(\Sigma V^TW^TU)$

Therefore, $W^* = argmax_{W \in O_d(R)} \sum_i \Sigma_{ii} W'_{ii}$ with $W' = V^TW^TU$
Nonetheless, W' is orthogonal and $_{ii}$ are non negative, the maximum is reached for W' equal to I, so $I = W' = V^TW^TU$,

Therefore, $W = UV^T$.

2 Sentence classification with BoV

What is your training and dev errors using either the average of words or the weighted-average? Here are my results:

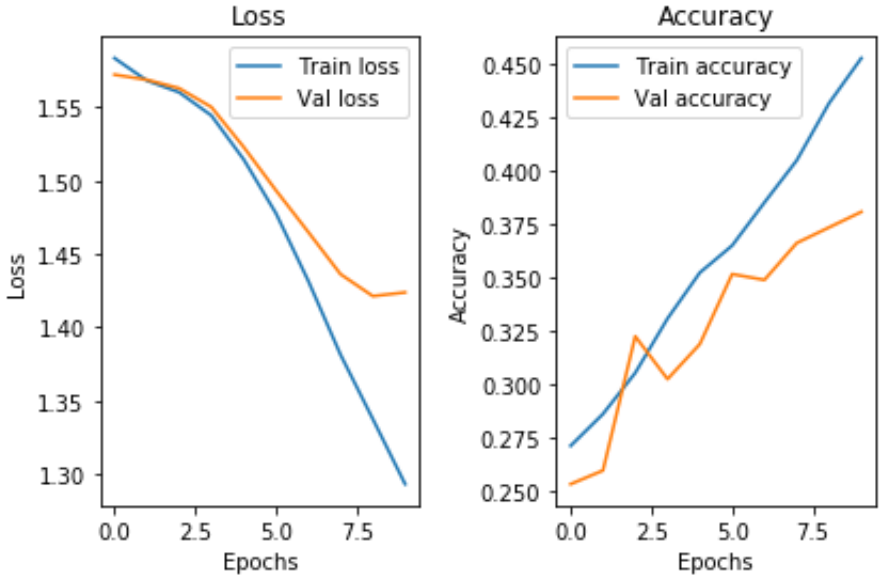
```
-------Without idf--------
accuracy score for train is  0.46769662921348315
accuracy score for dev is 0.4250681198910082
-------With idf--------
accuracy score for train is  0.4621956928838951
accuracy score for dev is 0.4268846503178928
```

3 Deep Learning models for classification

Which loss did you use?
I used *the Categorical Crossentropy loss* because we are dealing with a multi classification problem. The neural network tries to approach p and gives q. The cross entropy represents the difference between the output q with respect to the true value q.

The loss is given for a set of input X by :
$H = -\sum_x p(x)log(q(x))$

Plot the evolution of the train/dev results with respect to the number of epochs



What are your motivations?
The first idea is to remove the Embedding Layer and use a pretrained embedding using the Amazon Crawl. Indeed, the data wasn't sufficient enough and an overfit is very likely that's why I decided to train the model on a huge quantity of data. Secondly, I used BiLSTM because according to some papers, it enables the network to learn fast. Then, I added a dropout layer to reduce overfit. I get an accuracy of 0.4342.