Deep Learning for Natural Language Processing

Armand Margerin

February 2020

1 Multilingual Word Embeddings

We have

$$\begin{aligned}\|WX - Y\|_F &= tr((WX - Y)^T(WX - Y)) \\ &= tr(X^TW^TWX - X^TW^TY - Y^TWX + Y^TY) \\ &= tr(X^TW^TWX) - 2tr(X^TW^TY) + tr(Y^TY) \\ &= tr(X^TX) - 2tr(X^TW^TY) + tr(Y^TY)\end{aligned}$$

Hence $argmin_{W \in O_d(R)}(\|WX - Y\|_F) = argmax_{W \in O_d(R)}(tr(X^TW^TY))$

We use the singular value decomposition of YX^t:

$SVD(YX^T) = U\Sigma V^T$

$tr(X^TW^TY) = tr(W^TYX^T) = tr(W^TU\Sigma V^T) = tr(V^TW^TU\Sigma)$

The matrix V^TW^TU is orthogonal since U and V are unitary matrices:

$$U^TWVV^TW^TU = U^TWW^TU = U^TU = I$$

Hence, $tr(V^TW^TU\Sigma) \leq tr(\Sigma)$ and we get the maximum value for $V^TW^TU = I$, i.e $W = UV^T$.

2 Sentence Classification with BoW

Average of word vectors:
Train accuracy: 46,0%
Dev accuracy: 39,4%

Weighted average of word vectors:
Train accuracy: 46,5%
Dev accuracy: 39,2%

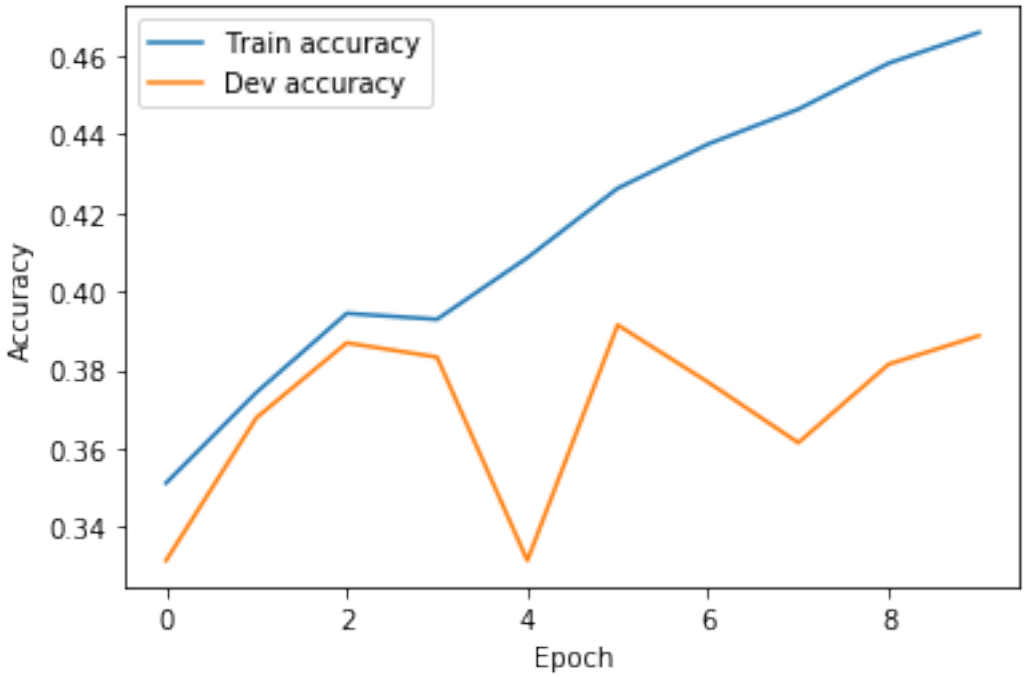
3 Deep Learning models for classification

I chose to use the categorical-crossentropy loss because it is a multi-class classification problem. It is defined by:

$$l(m,n) = \sum_{i=1}^{5} m_i log(n_i)$$

where m is the true class and n the predicted class

Evolution of train/dev results:



Other encoder:

I tried using the same encoder with pre-trained vectors from Word2vec, but it resulted in a poor performance. One possible reason for that is the dimension of the embeddings (300): with this dimension I also had bads results in the original model.