Deep Learning for Natural Language Processing

Ngoc-Tam LE

1 Monolingual embeddings

See notebook.

2 Multilingual word embeddings

Question The objective function in $W \in O_n(\mathbb{R})$ to minimize is :

$$\begin{aligned}||WX - Y||_F^2 &= \|WX\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle_F \\ &= \|WX\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle_F \\ &= \|X\|_F^2 + \|Y\|_F^2 - 2\langle W, YX^T\rangle_F\end{aligned}$$

Thus, the optimization problem can be rewritten as :

$$max\ \langle W, YX^T\rangle_F,\ s.t.\ W \in O_n(\mathbb{R})$$

Let $U\Sigma V^T$ be the SVD decomposition of YX^T. Then $\langle W, YX^T\rangle_F = \langle W, U^T\rangle_F = \langle U^T WV, \Sigma\rangle_F$. Σ is positive and diagonal, so the scalar product is maximal for $U^T WV = Id$ by Cauchy-Schwarz inequality (for an orthogonal matrix $A, \langle A, \Sigma\rangle_F = \sum_i \Sigma_i A_i < \sum_i \Sigma_i$ with equality for $A = Id$). Hence, $W^* = UV^T$.

3 Sentence classification with BoW

After some tuning to find an optimal regularization parameter, I get the following scores :

	train	dev
no idf	0.450	0.401
idf	0.455	0.402

Table 1: Scores for the Logistic regression model

We have slightly better results for the idf-weighted vectors, but not significantly more than with a simple mean.

4 Deep Learning models for classification

Loss We have a classification problem on a data set $\{(x_i, y_i)\}_{1 \leq i \leq N}$ with $\forall i \leq N, y_i \in 1, ..., K$ ($K = 5$ for our data set). Thus, I used the categorical cross-entropy. It's expression is the following :

$$\mathcal{L}(\Theta, x, y) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{1}_{\{y_i=k\}} log(p_k(x_i))$$

Where $(p_1(x_i), ..., p_K(x_i))$ denotes the vector output of the model for an input x_i.

Train and dev results We plot the accuracy and loss results :

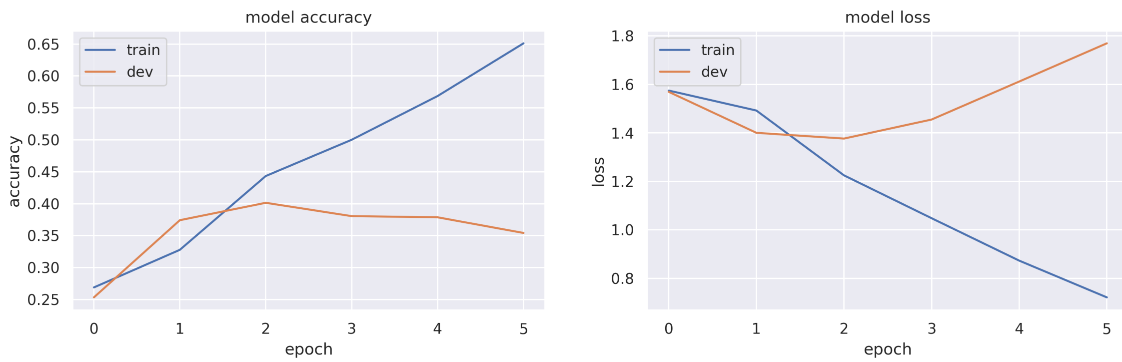

Figure 1: Accuracy and loss evolution on the train an dev set

New architecture using Conv1D and BiLSTM Embeddings represent the main part of the model parameters. I decided to use pre-trained embeddings so that the training become less expensive. Then, I wanted to exploit sequential correlations in both directions, so I chosed an architecture composed with a BiLSTM layer and a Conv1D layer. I also added some dropout to avoid overfitting. The output of the Conv1D is put into a Global Max Pooling and also an Averaged pooling layer to add more information. For some runs, this architecture yields better results.