Deep Learning
Mini project 1: Deep learning for NLP

OUMESSAOUD Mohamed

MVA
January 2020

2. Multilingual word embeddings

Question :

$$\begin{aligned} W^* &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F \\ &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F^2 \\ &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmin}} \|X\|_F^2 + \|Y\|_F^2 - 2 < WX, Y > \\ &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmax}} < W, YX^T > \\ &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmax}} < W, U\Sigma V^T > \\ &= \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmax}} < U^T W V, \Sigma > \end{aligned} \tag{1}$$

We used the trace trick (i.e. $< A, B^T >=< A^T, B >$ for any given matrices with appropriate dimensions), and that $U\Sigma V^T = SVD(YX^T)$.
The quantity $U^T WV$ is an orthonormal matrix (as it is a product of orthonormal matrices) and thus the expression is maximised when $U^T WV = I_d$ where I_d the identity matrix. Hence,

$$W^* = \underset{W \in \mathbf{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T \tag{2}$$

3. Sentence classification with BoW

Question :

- Logistic regression classifier :

  We used logistic regression classifier; the obtained training and dev errors using either the average of word vectors or the weighted-average :

	words' average	weighted-average
train set	42.64%	46.98%
dev set	39.14%	40.50%

- another classifier : We Try to improve performance with **Random Forest Classifier**.This improves train accuracy; however, this does not improve performance on dev set.

	words' average	weighted-average
train set	99.85%	99.80%
dev set	35.78%	35.60%

4. Deep learning models for classification

Question :

We use the "sparse_categorical_crossentropy". It is defined as categorical crossentropy with integer targets; "categorical crossentropy" itself is defined as follows : for a given set of inputs X, with associated probabilities $p(x) = (p(x)_i)_{i\in 0,..,4}$ and predicted probabilities : $q(x) = (q(x)_i)_{i\in 0,..,4}$:

$$Loss = -\sum_{x\in X} p(x)\log\big(q(x)\big) \tag{3}$$

Question :

plotting the train and test "accuracy" and "loss" vs number of epochs :

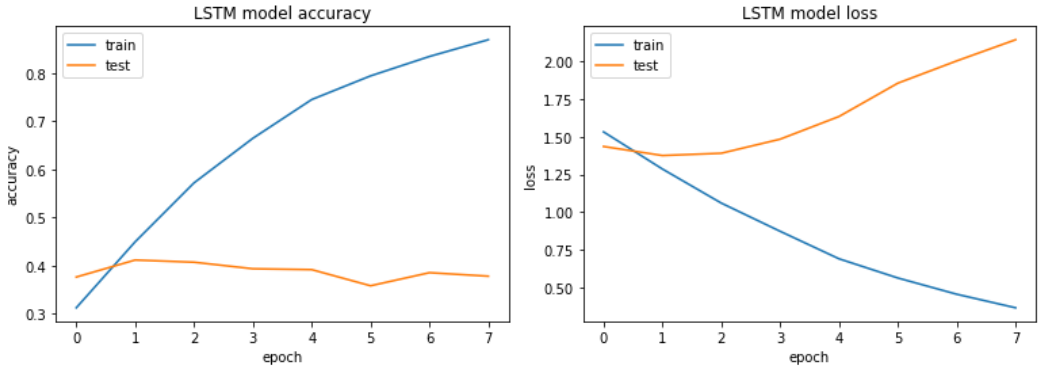

Figure 1: LSTM model accuracy (left) and loss (right) vs n_epochs

We observe that our model gives results that are close to those given by the previous classifiers. Also, we can see that the model is overfitting.

Question : Innovate

The model I have proposed is composed of the following :

- A pretrained embedding matrix based on "crawl-300d-2M.vec" fasttext word embedding; this helps improve model's accuracy.
- A bidirectional LSTM which helps preserve information from both directions (past and future) and thus can help improve performance.
- On Conv1D layer + MaxPooling1D layer; this helps recognize patterns in data.
- L2 regularization in the fully connected layer; this helps prevent the model from overfitting.

  The new model improves validation accuracy up to 43,60% instead of 37,33%.