Deep Learning: Deep Learning for Natural Language Processing

Due on January 11, 2019

Acher Clément

Question 1

Let's show that

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T, \text{ with } U\Sigma V^T = \text{SVD}\left(YX^T\right)$$

Let $W \in O_d(\mathbb{R}$. We first notice that :

$$\begin{aligned} \|WX - Y\|_F^2 &= \|WX\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle \\ &= \|X\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle \qquad (W \in O_d(\mathbb{R})) \end{aligned}$$

Where $\langle WX, Y\rangle = \text{Tr}\left((WX)^T Y\right) = \text{Tr}\left(W^T Y X^T\right)$. Hence,

$$\begin{aligned} \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F &= \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F^2 \\ &= \underset{W \in O_d(\mathbb{R})}{\operatorname{argmax}} \text{Tr}\left(W^T Y X^T\right) \end{aligned}$$

Let $U\Sigma V^T$ the SVD decomposition of YX^T.

$$\begin{aligned} \text{Tr}\left(W^T Y X^T\right) &= \text{Tr}\left(W^T U \Sigma V^T\right) \\ &= \text{Tr}\left(V^T W^T U \Sigma\right) \end{aligned}$$

Let $Z = W^T U \Sigma$. Z is orthogonal as the product of orthogonal matrices. $\text{Tr}\left(Z\Sigma\right) = \sum_{i=1}^n Z_{i,i}\Sigma_{i,i}$. $(\Sigma_{i,i})_i$ are non-negative and Z is orthogonal hence the previous quantity is maximized if $Z_{i,i} = 1$ which means that $Z = I_n$. We then conclude that $W^* = UV^T$.

Question 2

We test the sentence classification task using bot the average of word vectors and the weighted-average based on the idf.
We get the following results :

Dataset	Average	IDF
Train	0.4992	0.4993
Dev	0.4405	0.4233

We get slightly better results using the average of word vectors.

A classifier based on SVM was also trained and achieves similar results. Random forest was also tested, but even after tweaking the hyperparameters, it was still overfitting the train dataset.

Question 3

We use the categorical cross-entropy loss function :

$$-\sum_{o=1}^{N}\sum_{c=1}^{5} y_{o,c}\log(p_{o,c})$$

Where :

- N is the number of observations

Question 3 continued on next page. . .

- y is the true distribution : $y_{o,c} = 1$ if observation o is of class c.
- p is the predicted distribution - i.e. the output of the model. $p_{o,c}$ is the probability of observation o to be from class c according to the model.

Question 4

We train a classifier based on a the LSTM model. Without using pretrained embedding vectors, the model is quite far from beating the logistic regression. Also note that it overfits very quickly : to avoid this the model was just trained on 2 epochs. See Figure 1. After 2 epochs, the dev loss starts to increase.

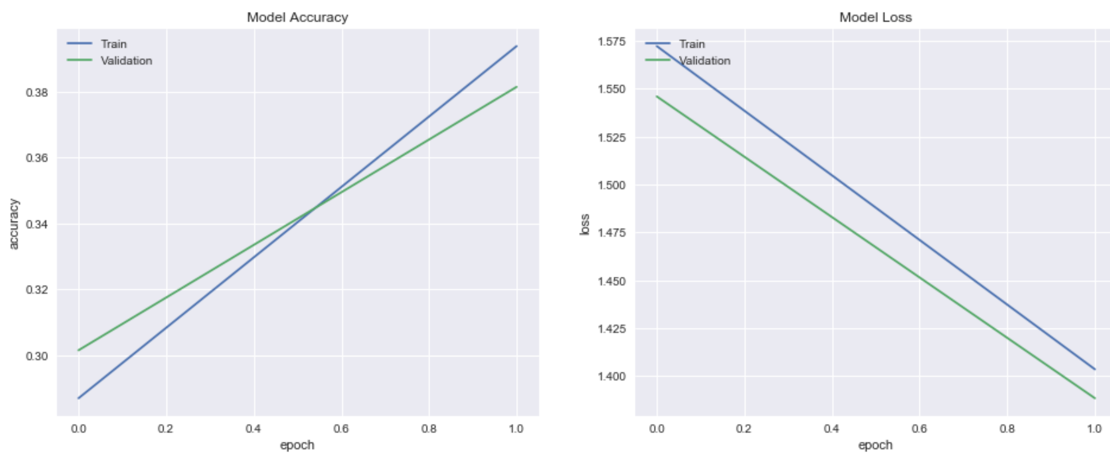

Figure 1: Loss and accuracy for the LSTM model

Question 5

Given the poor results that we get from simply using a LSTM, we decide to use the pretrained Fasttext vectors as embeddings. We still use LSTM cells, but we put a 1D convolutionnal layer before, which gives a convolutionnal LSTM network. Another nice advantage of this solution is that with the pretrained embeddings and the maxpool layer, training is much faster.

See Figure 2 for the accuracy and loss plots. This models also tends to overfit : we stop the training after 6 epochs. This models gets better results than the vanilla LSTM model, but does not perform better than the logistic regression on the validation dataset.

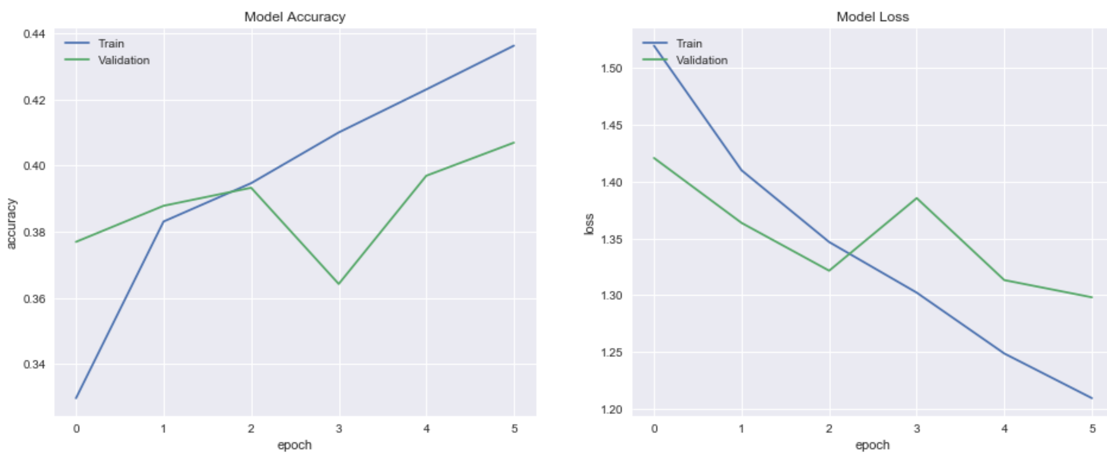

Figure 2: Loss and accuracy for the CNN-LSTM model