Project 2 Deep Learning
Natural Language Processing

John Levy
johnlevy125@gmail.com

1 Monolingual (English) word embeddings

See code for the implementation.

2 Multilingual (English-French) word embeddings

Let's prove that :

$$W^* = \underset{W \in O_d(\mathbb{R})}{\arg\min} ||WX - Y||_F^2 = UV^T \text{ with } U\Sigma V^T = \text{SVD}(YX^T)$$

$$\begin{aligned}
\underset{W \in O_d(\mathbb{R})}{\arg\min} ||WX - Y||_F^2 &= \underset{W \in O_d(\mathbb{R})}{\arg\min} Tr((WX - Y)^T (WX - Y)) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\min} Tr((WX)^T WX)) + Tr(Y^T Y) - 2Tr(X^T W^T Y) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\min} -2Tr(X^T W^T Y) \qquad \text{because it is the only term that depends on W} \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} Tr(X^T W^T Y) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} Tr(W^T Y X^T) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} Tr(W^T U\Sigma V^T) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} Tr(V^T W^T U\Sigma) \qquad V^T W^T U \text{ is orthogonal as a product of orthogonal matrices}
\end{aligned} \tag{1}$$

Let's consider $Z = V^T W^T U$, we can denote

$$Tr(V^T W^T U\Sigma) = Tr(Z\Sigma) = \sum_{i=1}^{n} \sigma_{i,i} z_{i,i}$$

And as Σ is diagonal with all of its diagonal coefficients positive, and as Z is orthogonal, that means $\forall i,\ z_{i,i} \leq 1$, we can deduce that $\sum_{i=1}^{n} \sigma_{i,i} z_{i,i}$ is maximal when $Z = V^T W^T U = I_d$.

Hence, as $V^T W^T U = I_d$, we can deduce that $\boxed{W^* = UV^T}$.

3 Sentence classification with BoV

In the following table, we can see that the result are usually better without weighted average, but for both, the train and accuracy are close, and it seems that this logistic classifier don't do well.

	Average Word Vectors	Weighted Average Word Vectors
Train Accuracy	0.4624	0.4331
Dev Accuracy	0.4187	0.4042

Table 1: *Accuracy with Logistic Regression*

To have better result, I plot the evolution of the accuracy for different regulators parameters, the results are presented in the following graph.

Figure 1: *Evolution of the Accuracy for different penalization*

Now we are trying to implement an another model, I chose a RandomForest Classifier. This time, it seems that weighted average performs better than without IDF.

	Average Word Vectors	Weighted Average Word Vectors
Train Accuracy	0.3875	0.40625
Dev Accuracy	0.3396	0.3469

Table 2: *Accuracy with Random Forest*

4 Deep Learning models for classification

Question

The loss that I used is the categorical cross entropy loss given by :

$$\boxed{L(y,\hat{y}) = -(1/N)\sum_{i=0}^{N}\sum_{j=0}^{M}(y_{ij} * \log(\hat{y}_{ij}))}$$

Where N is the size of the batch, M the number of class, $\hat{y}$ the output of our Neural Network (i.e our prediction) and y the truth label. Here M = 5 because we have 5 class. This kind of loss is very useful when we work on multi-class classification task.

Question

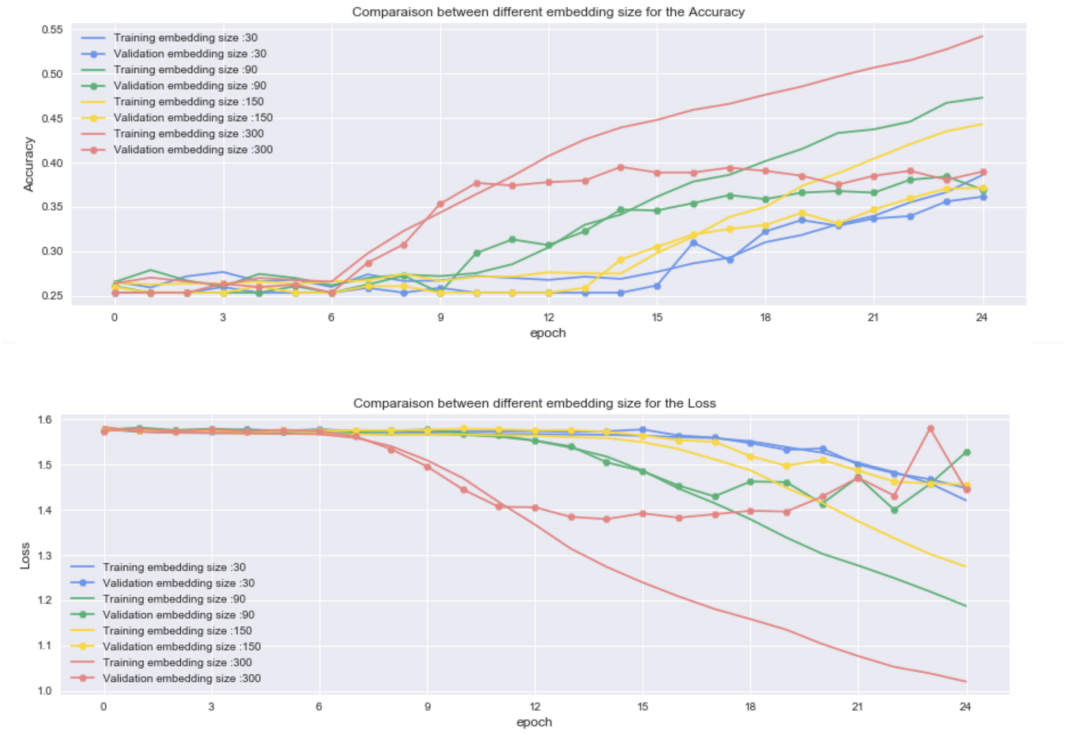

Figure 2: *Evolution of the Accuracy (graph 1) and Loss (graph 2) in the Training and Dev Set for different embedding size.*

As the embedding layer is initialized randomly, I tried several size for the embedding layer in order to improve the results, each size of embedding is represented in a specific color.
We can see a big difference between the accuracy and loss of the training set (represented with lines), and the accuracy and loss of the dev set (represented with lines that have a little rectangle). The model is clearly overfitting as the results on the training set are quite good, but poor on the dev set.

Question

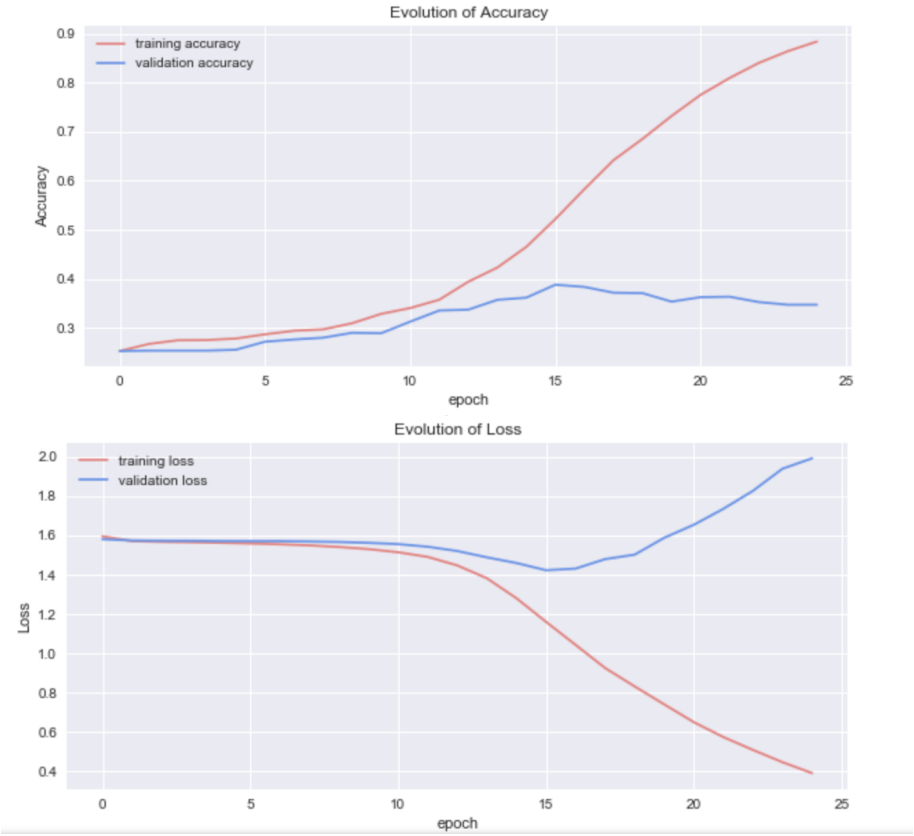

Figure 3: *Evolution of the Accuracy (graph 1) and Loss (graph 2) in the Training and Dev Set.*

The model that I have implemented is composed by :
- 1 Embedding layer with the embedding given by our function word2vec.
-1 Conv1D with 50 filters and a kernel size of 2 and a relu activation function.
-1 Max Pooling1D with a pool size of 5
-1 Conv1D with 50 filters and a kernel size of 3 and a relu activation function.
-1 Max Pooling1D with a pool size of 5
-1 Bidirectional LSTM with 128 hidden units, and a dropout of 0.5.
-1 Fully connected layer with 5 outputs units that corresponds to our 5 class with a softmax activation function.

My loss is still a cross entropy but this time I chose an Adam optimizer with a learning rate of 10^{-4}.
The main motivation to use convolutions is that a sliding window passes over texts and allows to identify important features. The max pooling will keep only the most important one, and by using a Bidirectional LSTM we go through the sentence by starting at the the end and also by starting at the beginning. Hence, we "analyze" the sentence by taking into consideration all its meaning.
We can see in the above graphs that the model quickly overfit after 15 epochs as the training accuracy improves but the dev accuracy does not improve.
The best accuracy that we have with this model is around 40%.