Deep Learning for Natural Language Processing Project 2

BOURIAL, Sarah
sarah.bourial@student-cs.fr

VIAL, Jennifer
jennifer.vial@student-cs.fr

January 2019

Abstract

In this assignment we will cover monolingual word and sentence embeddings, multilingual word embeddings and sentence classification with Bag-of-Vectors (BoV) and LSTMs.

1 Multilingual word embeddings

Using the orthogonality and the properties of the trace, prove that, for X and Y two matrices:

$$\mathrm{W}^* = \underset{W \in O_d(R)}{argmin} \parallel WX - Y \parallel_F^2 = UV^T \ ; \ with\ UV^T = SVD(YX^T)$$

Proof. Start by expanding $\parallel WX - Y \parallel_F^2$:

$$\parallel WX - Y \parallel_F^2 = tr(WXY)(X^T W^T Y^T)$$
$$\parallel WX - Y \parallel_F^2 = tr(WX)(X^T W^T) - tr(WXY^T) - tr(YX^T W^T) + tr(YY^T)$$
$$\parallel WX - Y \parallel_F^2 = \parallel WX \parallel_F^2 + \parallel Y \parallel_F^2 - 2tr(YX^T W^T)$$

Thus, minimizing $\parallel WX - Y \parallel_F^2$ means trying to maximise $tr(YX^T W^T)$. Let $U\Sigma V^T$ be the singular value decomposition of YX^T. Then,

$$tr(YX^T W^T) = tr(U\Sigma V^T W^T) = tr(\Sigma V^T W^T U) \tag{1}$$

Set Z as $Z = V^T W^T U$. Since Z is the product of orthogonal matrices, Z is orthogonal. We thus seek to maximize $tr(\Sigma Z)$ by choosing the right W. Σ is diagonal so we get $tr(\Sigma Z) = \sum_{i=1}^{n} \Sigma_{ii} Z_{ii}$. The Σ_{ii} are non-negative, Z is orthogonal for all choices of W thereby the maximum is achieved by choosing W such that all of $Z_{ii} = 1$. This means $Z = I$. Therefore an optimal W is $W = UV^T$.

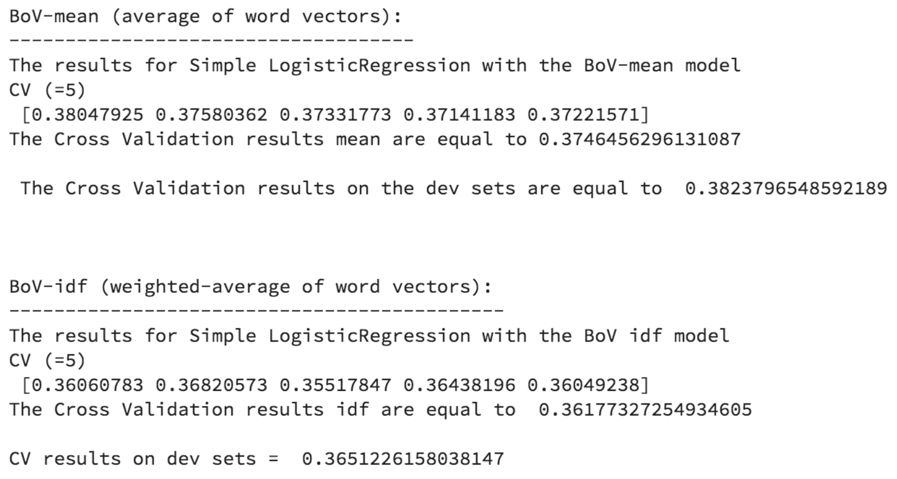
Q.E.D.

2 Sentence classification with BoV

Question: What is your training and dev errors using either the average of word vectors or the weighted-average?

2.1 Logistic Regression

We perform Simple Logistic Regression on the dev set using two methods: (i) average word of vectors (BoV-mean) and (ii) weighted average of word vectors (BoV-idf).



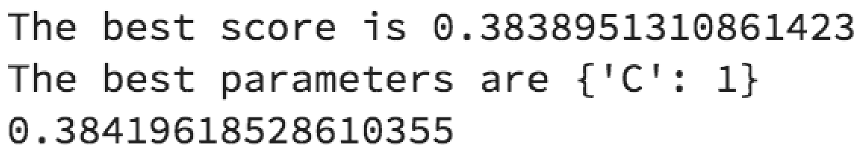
```
BoV-mean (average of word vectors):
------------------------------------
The results for Simple LogisticRegression with the BoV-mean model
CV (=5)
 [0.38047925 0.37580362 0.37331773 0.37141183 0.37221571]
The Cross Validation results mean are equal to 0.3746456296131087

 The Cross Validation results on the dev sets are equal to  0.3823796548592189


BoV-idf (weighted-average of word vectors):
--------------------------------------------
The results for Simple LogisticRegression with the BoV idf model
CV (=5)
 [0.36060783 0.36820573 0.35517847 0.36438196 0.36049238]
The Cross Validation results idf are equal to  0.36177327254934605

CV results on dev sets =  0.3651226158038147
```

Figure 1 : Performance of the logistic regression using BoV-mean and BoV-idf

We evaluate the models by their cross-validation accuracy. We can thus see that the average word vectors method (BoV-mean) performs best.

2.2 Parameters Tuning

To improve the accuracy of the model, we have decided to tune its L2 parameters using GridSearch and StratifiedKFold. We get the following results:



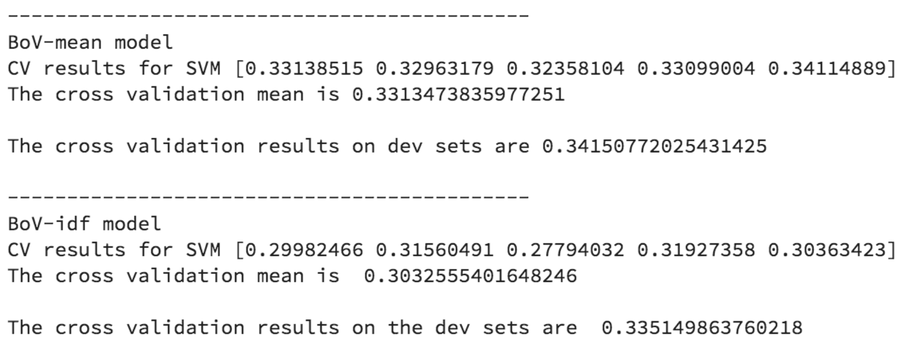
```
The best score is 0.3838951310861423
The best parameters are {'C': 1}
0.38419618528610355
```

Figure 2 : Parameters tuning of the logistic regression using BoV-mean and BoV-idf

We notice a slight improvement in the accuracy score of 0.38 (up by 0.02) compared to the previous non-tuned model.

2.3 Other Model

Further, we chose to apply a Support Vector Machine (SVM) approach. We used this classifiers with both BoV-mean and BoV-idf. The results are as follows:



```
-------------------------------------------
BoV-mean model
CV results for SVM [0.33138515 0.32963179 0.32358104 0.33099004 0.34114889]
The cross validation mean is 0.3313473835977251

The cross validation results on dev sets are 0.34150772025431425

-------------------------------------------
BoV-idf model
CV results for SVM [0.29982466 0.31560491 0.27794032 0.31927358 0.30363423]
The cross validation mean is  0.3032555401648246

The cross validation results on the dev sets are  0.335149863760218
```

Figure 3 : Performance of SVM classifier using BoV-mean and BoV-idf

Surprisingly, this more advanced classifier presents less satisfying results than the previous Logistic Regression approach with parameters tuning.

3 Deep Learning models for classification

3.1 Which loss did you use? Write the mathematical expression of the loss you used for the 5-class classication.

The network model we decide to build focuses on the RMSprop gradient descent optimization algorithm with "categorical cross entropy" as loss function. This loss function is suitable here as we are facing a multi-class classification problem whereby every one of our sentences can only belong to a single category.

The categorical cross-entropy loss is also known as the negative log-likelihood. It is a popular loss function for categorization problems and measures the similarity between two probability distributions, typically the true labels and the predicted labels. It is given by $L = -\sum y * log(y_{prediction})$ where y is the probability distribution of true labels (typically a one-hot vector) and $y_{prediction}$ is the probability distribution of the predicted labels, often coming from a softmax.

In NLP, this loss function is based on the fact that any directly decodable coding scheme for coding a message to identify one value x_i out of a set of possibilities X can be seen as representing an implicit probability distribution $q(x_i) = 2^{-l_i}$ over X, where l_i the length of the code for x_i in bits.

In this sense, cross entropy can be interpreted as the expected message-length per datum when a wrong distribution Q is assumed while the data actually follows a distribution P. That is why the expectation is taken over the probability distribution P and not Q, as follows:

$$H_(p, q) = E_p[l_i] = E_p[log\frac{1}{q(x_i)}] = \sum_{x_i} p(x_i) log\frac{1}{q(x_i)} = -\sum_{x} p(x) log q(x) \tag{2}$$

3.2 Plot the evolution of train/dev results w.r.t the number of epochs.

```
Best Dev accuracy: 0.39963669404996516
```

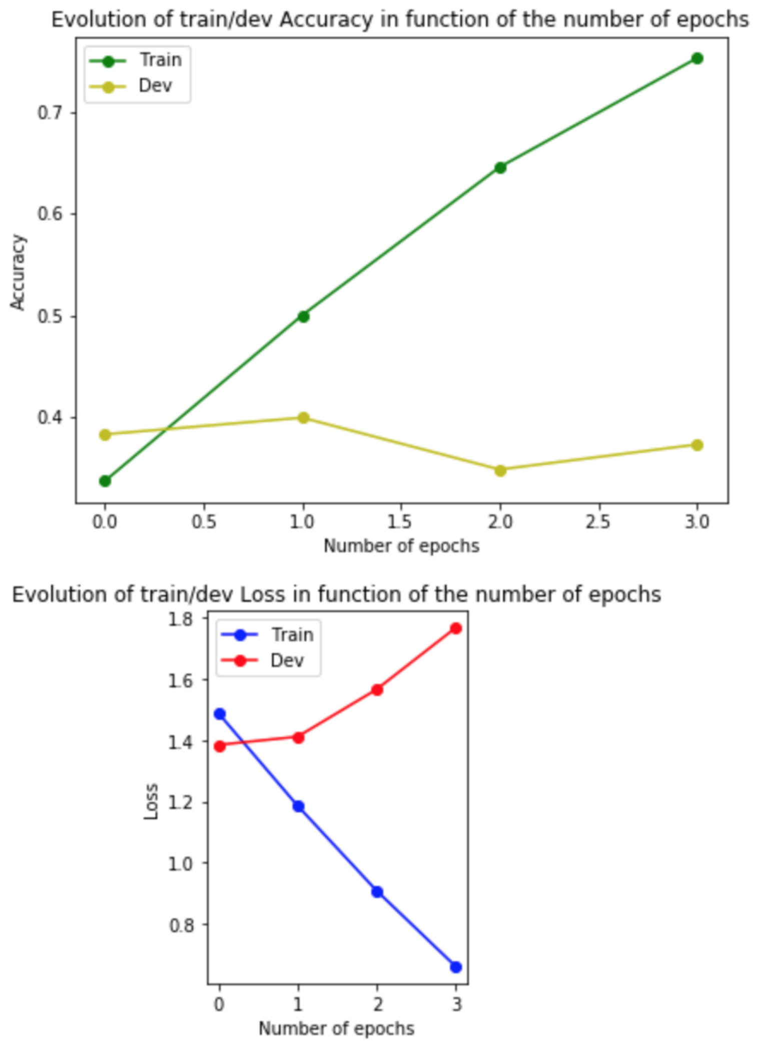
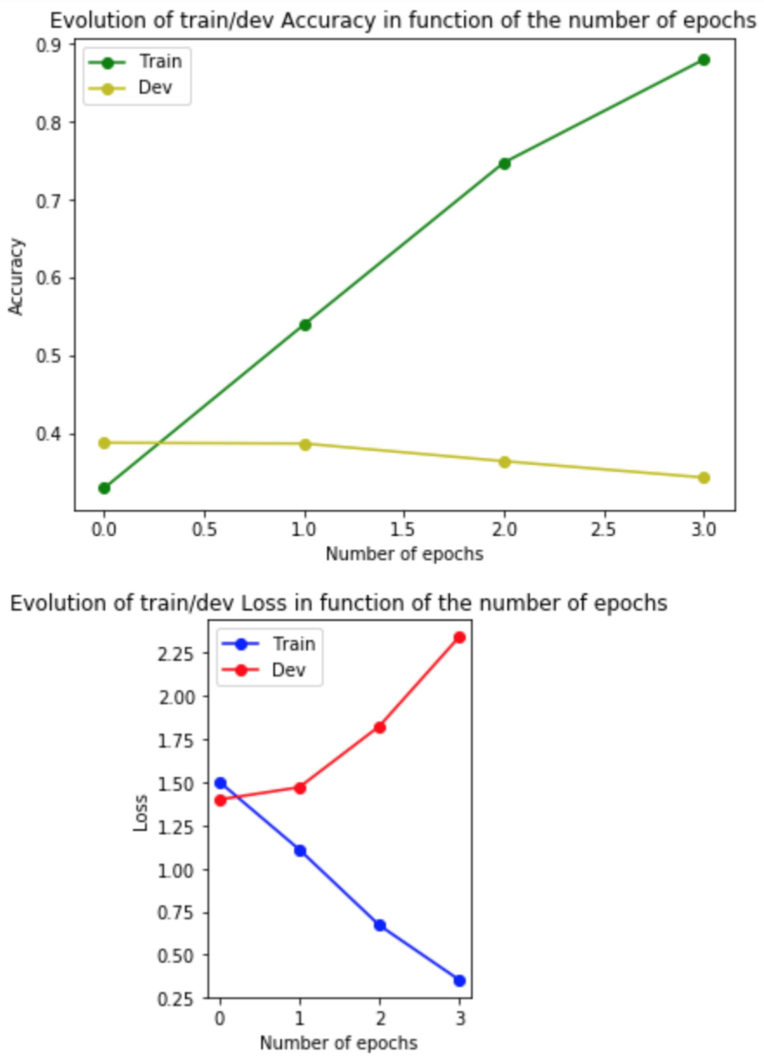


Figure 4: Evolution of train/dev results w.r.t the number of epochs

Observation: As the number of epoch increases, the train accuracy increases likewise while the loss decreases sharply. As for the dev test, the accuracy starts increasing and then decreases from number of epoch = 2 before increasing slowly again. The loss is on a sharp increase as the number of epochs increases.
The best accuracy given this model is 39.96%.

3.3 Be creative: use another encoder. Make it work! What are your motivations for using this other model?

The second network model we decide to build focuses this time on the Adam gradient descent optimization algorithm. Indeed, optimisers play a very crucial role to increasing the accuracy of the model and thus we decide to modifiy this parameter of our model to see the subsequent evolution in terms of accuracy.



```
Best Dev accuracy: 0.38873751136684936
```

Figure 5: Evolution of train/dev results w.r.t the number of epochs

Observation: Just like before, as the number of epoch increases the train accuracy increases likewise while the loss decreases sharply. As for the dev test, the accuracy decreases throughout the experiment. The loss is on a sharp increase as the number of epochs increases. The best accuracy given this model is 38.87%, which is not an improvement on our previous score achieved with an RMSprop optimizer. We conclude that the Adam optimizer is not suitable for our SST data.