Deep Learning for Natural Language Processing

—

Project

TAs: Louis Martin, Edouard Grave
Student: Antoine Moulin

Abstract

In this assignment we will cover monolingual word and sentence embeddings, multilingual word embeddings and sentence classification with Bag-of-Vectors (BoW) and LSTMs. An iPython notebook for the full project is attached: `nlp_project.ipynb`. It contains instructions on what you must code. Please refer to the "Send your answers" part for a description of what we expect from you in terms of deliverable.

1 Monolingual embeddings

In `nlp_project.ipynb` you are asked to write functions for computing the nearest neighbors of any word, without using an external package. You will build two classes for word vectors and bag-of-words vectors, such that you get the desirable outputs (see code).

2 Multilingual word embeddings

The goal is to find a mapping W that will map a source word space (e.g. French) to a target word space (e.g. English), such that the mapped source words are close to their translations in the target space. For this, we need a dictionary of "anchor points". Here, we will use the identical character strings in both languages. We can show that the solution of $\underset{W \in O_d(\mathbb{R})}{\arg\min}$ has a closed form.

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

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

Solution.

$$\begin{aligned}
W^* &= \underset{W \in O_d(\mathbb{R})}{\arg\min} ||WX - Y||_F^2 \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\min} ||WX||_F^2 + ||Y||_F^2 - 2\left\langle WX, Y\right\rangle_F \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\min} ||X||_F^2 + ||Y||_F^2 - 2\left\langle WX, Y\right\rangle_F \qquad (W \text{ orthogonal}) \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} \left\langle WX, Y\right\rangle_F \\
&= \underset{W \in O_d(\mathbb{R})}{\arg\max} \left\langle W, YX^\top\right\rangle_F
\end{aligned}$$

Let us write the SVD of $YX^\top$: $YX^\top = U\Sigma V^\top$ with U, V orthogonal matrices and Σ a diagonal matrix with nonnegative entries. We have:

$$W^* = \underset{W \in O_d(\mathbb{R})}{\arg\max} \left\langle U^\top WV, \Sigma\right\rangle_F$$

Denoting $S = U^\top WV$, which is an orthogonal matrix as well, we can write

$$\begin{aligned} W^* &= \underset{S \in O_d(\mathbb{R})}{\arg\max} \langle S, \Sigma \rangle_F \\ &\leq ||\Sigma||_F \qquad \text{(Cauchy-Schwarz, } S \text{ orthogonal)} \end{aligned}$$

And the maximum is reached with $S^* = I_d$. Hence, $U^\top W^* V = I_d$ i.e.

$$\boxed{W^* = UV^\top}$$

□

In `nlp_project.ipynb` you are asked to create X and Y using the identical character strings in each language, compute W and output target nearest neighbors of source words in the shared space.

3 Sentence classification with BoW

In this section and the following, we give you the train, dev and test sets of the Stanford Treebank fine-grained sentiment analysis task. It consists of input sentences that you have to classify into 5 classes. For the test set, we only provide you with the input samples, not the ground-truth labels. You will have to produce your predictions using your best model, and send it to us. We will evaluate ourselves the quality of your predictions.

You are asked to use scikit-learn to learn a logistic regression on top of bag-of-words embeddings on the SST task.

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

Solution. For the average of word vectors, the best performance is obtained with $C = 10$. The train accuracy is 0.466 and the dev accuracy is 0.417.

For the weighted-average, the best performance is obtained with $C = 1$. The train accuracy is 0.482 and the dev accuracy is 0.42. Hence, the results are better we using the IDF. □

4 Deep Learning models for Classification

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

Solution. For this model, I used the categorical cross entropy loss. For K classes, this loss is given by:

$$\mathcal{L}(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log(\hat{y}_i)$$

where y is the ground truth and $\hat{y}$ the prediction. □

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

Solution. The results are shown in figure 1. □

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

Solution. Besides using a LSTM, we also use a 1D CNN formed by one convolutional layer and one max pooling layer. This enables the network to learn some invariant features for each class and thus yields better performance. □

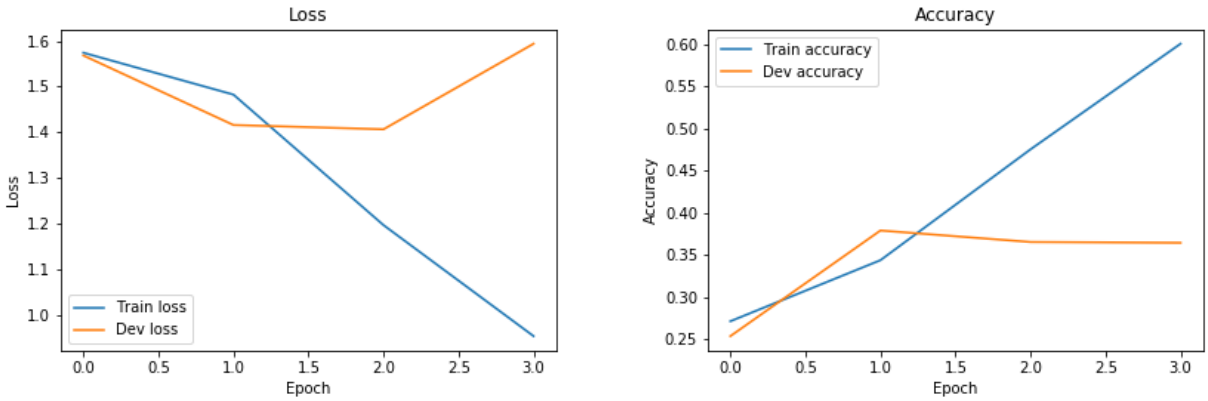

Figure 1: Loss and accuracy w.r.t. the number of epochs.

5 Sending your answers

You should create a `.zip` file, named `nlp_familyname_firstname.zip` and send it to `vincent.lepetit@enpc.fr`. The `.zip` file should contain:

- `answers.pdf` (with your answers to the questions above)
- `nlp_project.ipynb`
- `logreg_bow_y_test_sst.txt` and `XXX_bow_y_test_sst.txt`
- `logreg_lstm_y_test_sst.txt` and `XXX_XXX_y_test_sst.txt`

Please consider that having the same format for all the students save TAs a lot of time. We will consider penalties for submissions that do not follow these simple rules. Thanks!

For any questions please send an email to `louismartin@fb.com`.