Deep Learning for Natural Language Processing

Juan LONDONO

January 11th 2019

1 Monolingual embeddings (/6)

See Code.

2 Multilingual word embeddings (/4)

Question 1

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

$$W* = \underset{W}{\text{argmin}} \|WX - Y\|_F = UV^T \text{ with } U\Sigma V^T = SVD(YX^T)$$

Let us develop the Frobenius norm $\|WX - Y\|_F$:

$$\|WX - Y\|_F = (tr((WX - Y)^T(WX - Y)))^{1/2}$$

Therefore we only need to minimize $tr((WX - Y)^T(WX - Y))$

$$tr((WX - Y)^T(WX - Y)) = tr((X^TW^T - Y^T)(WX - Y))$$

$$\iff = tr(X^TX - X^TW^TY - Y^TWX + Y^TY)$$

$$\iff = tr(X^TX - X^TW^TY - (X^TW^TY)^T + Y^TY)$$

As $tr(X^TX)$ and $tr(Y^TY)$ are constants, we need to minimize:

$$tr(-X^TW^TY - (X^TW^TY)^T) = -tr(X^TW^TY) - tr(X^TW^TY)^T$$

$$\iff = -2 * tr(X^TW^TY)$$

We therefore need to maximize $tr(Y^TWX)$.

We know that $tr(AB) = tr(BA)$, hence:

$$tr(Y^TWX) = tr(XY^TW) = tr((YX^T)^TW) = tr((U\Sigma V^T)^TW) = tr(V\Sigma U^TW) = tr(U^TWV\Sigma)$$

Let us write $U^TWV = M$.

We know that $tr(M\Sigma) = \Sigma_i(M_{i,i}\Sigma_{i,i}) \leq \Sigma_i(\Sigma_{i,i})$, since $M_{i,i} \leq 1$ for all i, as M is orthogonal.

Therefore $tr(M\Sigma)$ is maximized when each $M_{i,i} = 1$.

$$\iff M = Id \iff W = UV^T$$

.

3 Sentence classification with BoV (/4)

Question 1

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

	Training Accuracy	Dev Accuracy
Word vectors average	0.4993561980568887	0.43272727272727274
Weighted average	0.497600374575676	0.4090909090909091

As we may see, our model tends to overfit since the training accuracy is 6 to 9 points higher than the dev accuracy. We also notice that the word vectors model performs 3 points better than the weighted average model in the dev set. This is the one that we chose to predict the test results.

4 Deep Learning models for classification (/6)

Question 1

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

For the loss of the classifier, I used the categorical cross entropy. This loss is well adapted to our model since is is useful for multiclass classification problems. It returns the cross-entropy between an approximating distribution and a true distribution.

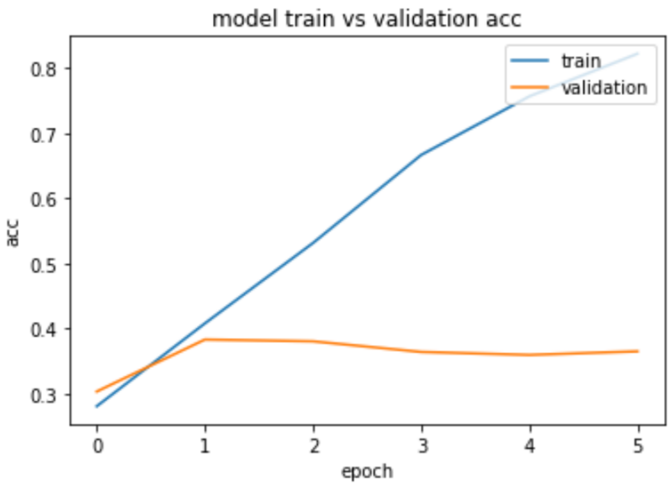
This function computes

$$H(p, q) = -\Sigma_x p(x) log(q(x))$$

Where *p=true dist* and *q=coding dist.*

Question 2

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



As we may see, our model overfits very quickly. The maximum accuracy for the validation set is obtained in 1 or 2 epochs and then decreases, while the train accuracy continues to increase steadily.

Question 3

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

I decided to use another encoder. I used the Tokenizer function to preprocess the text.
Each input line would therefore have a size (,vocab size) instead of (,52). This higher dimension might lead to enhanced prediction.

I used a classical two-layer dense NN, which improved the previous results by 4 points.
Any attempt to perform a Conv1D or an LSTM lead to a very high computational time, and no extra performance.