Deep Learning for Natural Language Processing

Ayman Chaouki

January 12, 2019

1 Monolingual embeddings

See nlp_project.ipynb

2 Multilingual word embeddings

Let's prove that $W^* = \underset{W \in O_d(\mathbb{R})}{argmin} ||WX - Y||_F = UV^T$ with $U\Sigma V^T = SVD(YX^T)$

We have

$$\begin{aligned} ||WX - Y||^2 &= ||WX||^2 - 2 < WX, Y > + ||Y||^2 \\ &= < WX, WX > -2 < WX, Y > + ||Y||^2 \\ &= < W^TWX, X > -2 < WX, Y > + ||Y||^2 \end{aligned}$$

Since $W \in O_d(\mathbb{R})$ we have $W^TW = I_d$ and thus

$$\begin{aligned} ||WX - Y||^2 &= < X, X > -2 < WX, Y > + ||Y||^2 \\ &= ||X||^2 - 2 < WX, Y > + ||Y||^2 \end{aligned}$$

We deduce that $W^* = \underset{W \in O_d(\mathcal{R})}{argmin} ||WX - Y||_F = \underset{W \in O_d(\mathcal{R})}{argmax} < WX, Y >$ Now let's write the singular value decomposition of Y^TX:

We have $U\Sigma V^T = SVD(YX^T)$ with $U, V \in O_d(\mathcal{R})$ and Σ is diagonal:

$$\begin{aligned} < WX, Y > &= < W, YX^T > \\ &= < W, U\Sigma V^T > \\ &= < U^TWV, \Sigma > \\ &= < Z, \Sigma > \end{aligned}$$

here $Z = U^TWV \in O_d(\mathbb{R})$, therefore $ZZ^T = I_d \Rightarrow \sum_{i=1}^{d} Z_{ii}^2 = 1 \Rightarrow \forall i \;\; Z_{ii}^2 \leq 1$

Wa have

$$\begin{aligned} < WX, Y > &= < Z, \Sigma > \\ &= \sum_{i=1}^{d} Z_{ii}\Sigma_{ii} \end{aligned}$$

And since $\forall i \ \ \Sigma_{ii} \geq 0$ because of the SVD, and maximizing $< WX, Y >$ with respect to W is equivalent to taking $Z_{ii} = 1 \forall i$, and we deduce $W^* = UV^T$

3 Sentence classification with BoV

With average word vectors: $accuracy = 0.4205$
With idf weighted average word vectors: $accuracy = 0.4278$

4 Deep Learning models for classification

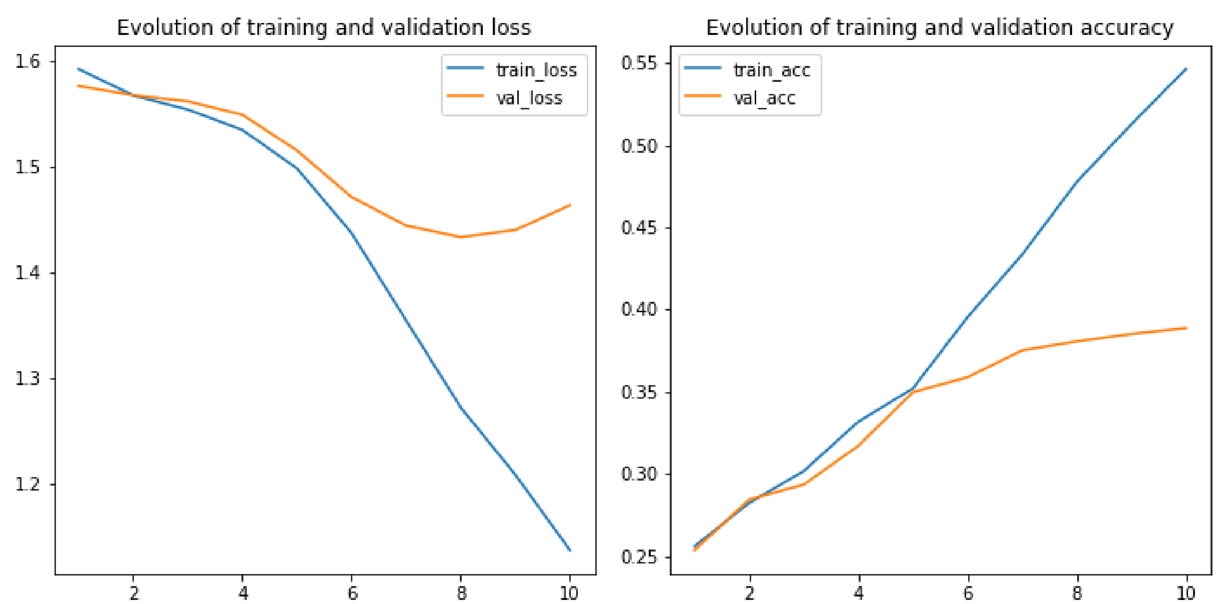
4.1 Choice of the loss function

This is a multiclass-classification problem, it is therefore natural to use the categorical cross-entropy as a loss function. Its mathematical formulation is as follows:

$$H(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{5} \hat{y}_i^{(j)} log \left(y_i^{(j)} \right)$$

Here $\hat{y}$ and y are respectively the one-hot encoded predicted and true vectors where $y_i^{(j)}$ denotes for example the value of the true one-hot encoded vector in sample i and class j.
N is the total number of samples

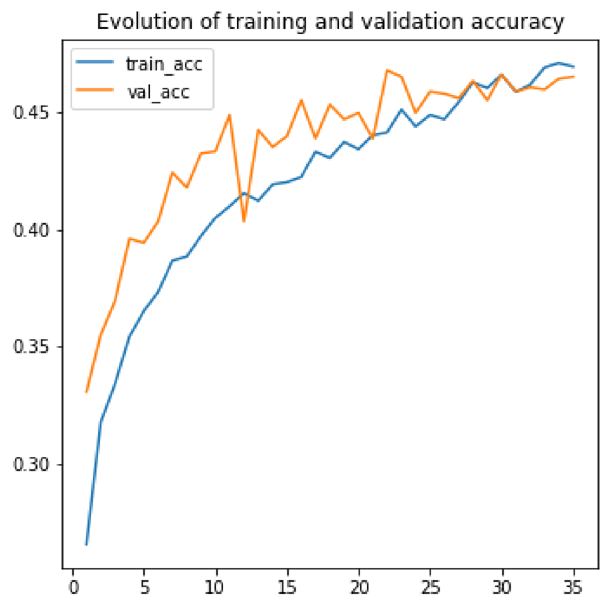
4.2 Evolution of train/dev



This model clearly overfits the data. This is surely due to training the embedding layer which contains $6 * 10^6$ parameters on a small dataset, even with the high dropouts rates (0.7 in all my dropouts), the model still highly overfits the data.

4.3 New model

Instead of using a trainable embedding layer, we set it to non trainable mode and feed it with a good representative embedding matrix. Such an embedding matrix can be $300-$Dimensional word embedding vectors that we loaded in the first question.
We still add some dropouts to avoid overfitting, and we also add a dense layer before the dense output layer.



And indeed this model does not overfit the data as the orevious one did.