Philippe Ganshof
philippe.ganshof@hotmail.com
ENS Paris-Saclay, Master 2, MVA

Deep Learning for Natural Language Processing

Question 2.1:

claim $\underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} ||\mathbf{WX} - \mathbf{Y}||_F = \mathbf{UV}^\top, \quad \mathbf{U\Sigma V}^\top = SVD(\mathbf{YX}^\top)$

Proof $||\mathbf{WX} - \mathbf{Y}||_F = \text{Tr}((\mathbf{WX} - \mathbf{Y})^\top(\mathbf{WX} - \mathbf{Y})) = \text{Tr}(\mathbf{X}^\top\mathbf{W}^\top\mathbf{WX} - \mathbf{X}^\top\mathbf{WY} - \mathbf{Y}^\top\mathbf{WX} - \mathbf{Y}^\top\mathbf{Y}) \underset{W \in O_d(\mathbb{R})}{=}$
$\text{Tr}(\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y} - 2\mathbf{W}^\top\mathbf{YX}^\top) = \text{Tr}(\mathbf{X}^\top\mathbf{X}) - \text{Tr}(\mathbf{Y}^\top\mathbf{Y}) - 2\text{Tr}(\mathbf{W}^\top\mathbf{YX}^\top)$

Since the first two terms do not depend on W, the problem is equivalent to:

$$\underset{W \in O_d(\mathbb{R})}{\operatorname{argmax}} \text{Tr}(\mathbf{W}^\top\mathbf{YX}^\top)$$

Using the Singular Value Decomposition, we get:

$$\text{Tr}(\mathbf{W}^\top\mathbf{YX}^\top) = \text{Tr}(\mathbf{W}^\top\mathbf{U\Sigma V}^\top) = \text{Tr}(\mathbf{W}^\top\mathbf{UV}^\top\mathbf{\Sigma})$$

$\mathbf{O} = \mathbf{W}^\top\mathbf{UV}^\top$ is the matrix product of orthogonal matrices and thus is also orthogonal. Moreover, $\mathbf{\Sigma}$ has positive entries and so we deduce that:

$$\text{Tr}(\mathbf{O\Sigma}) \leqslant \text{Tr}(\mathbf{\Sigma}), \quad \forall\, W \in O_d(\mathbb{R})$$

And if we choose $\mathbf{W} = \mathbf{UV}^\top$, $\mathbf{W}^\top\mathbf{UV}^\top = \text{Id}$ and thus $\text{Tr}(\mathbf{O\Sigma}) = \text{Tr}(\mathbf{\Sigma})$ which concludes that:

$$\underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} ||\mathbf{WX} - \mathbf{Y}||_F = \mathbf{UV}^\top$$

□

Question 3.1:

We plotted the training/dev accuracy with different regularization parameters for both embeddings (with and without IDF weights) as shown in figure 1. We obtain a better optimal accuracy using the IDF embeddings as expected, an improvement of $\sim 2\%$ compared to the embeddings with uniform weights. The optimal hyperparameter using IDF weights is $c = 0.55$ and $c = 1.1$ using uniform weights.

Question 4.1:

We used the categorical Cross-entropy loss. Below is given the mathematical expression of the loss:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where,

- M = Number of classes (in our case, M=5).

- $y_{o,c}$ = Binary indicator ($\in \{0, 1\}$), $y_{o,c} = 1$ if the observation o corresponds to the class label c, $y_{o,c} = 0$ otherwise.

- $p_{o,c}$ = Predicted probability observation o of the class label c (by the model).

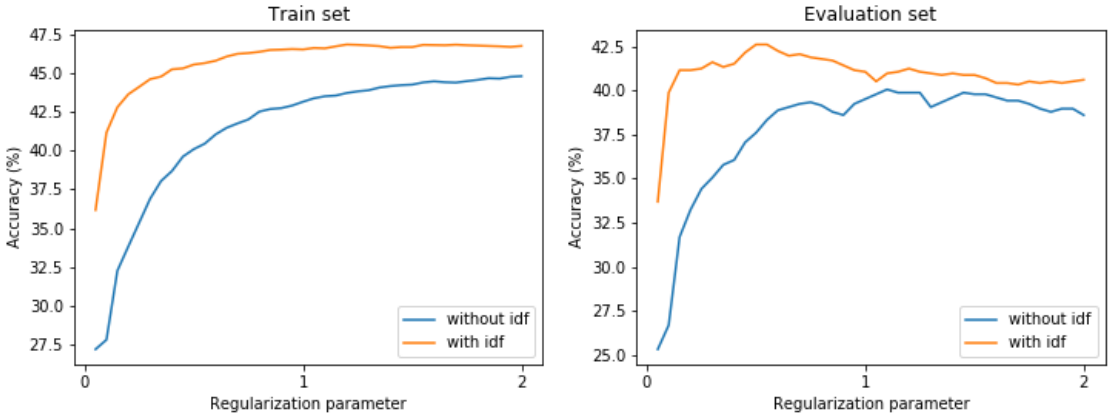

Figure 1: **IDF weights vs Uniform weights.**

Figure 2: **Train vs Evaluation performance.**

Question 4.2:

We plotted the evolution of the train/dev accuracy and the train/dev Cross-entropy loss with the number of epochs as shown in Figure 1. We see that after 13 epochs, the gap between the validation and the train performance increases rapidly which indicates overfitting. Thus, stopping the training after 12 epochs is recommanded in order to avoid overfitting.

Question 4.3:

We made two major modifications in order to increase the performance of our model. First, we used a bidirectional LSTM to provide additional context. We also removed all common words such as *a* or *and* which do not contain much information useful for our task. This process lowers the dimension of our feature space and offers the following advantages for improving the accuracy :

- Less misleading data

- Removes redundant features and noise

- Complexity reduced and so the risk of overfitting as well

- Get more information about the sentences when padding