Deep Learning for Natural Language Processing
Homework 1

Mohamed Kerroumi

- **Question 1**:

$$
\begin{aligned}
W^* &= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} Tr\Big((X^\top W^\top - Y^\top)(WX - Y)\Big) \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} Tr\Big(X^\top W^\top WX - X^\top W^\top Y - Y^\top WX + Y^\top Y\Big) \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} Tr\Big(X^\top X + Y^\top Y\Big) \quad - \quad 2Tr\Big(X^\top W^\top Y\Big) \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} -2Tr\Big(X^\top W^\top Y\Big) \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmax}} Tr\Big(Y^\top WX\Big) \\
&= \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmax}} \langle W, YX^\top \rangle
\end{aligned}
$$

Let $U, V \in \mathcal{O}_d(\mathbb{R})$ and $\Sigma \in \mathbb{R}^{d,d}$ a diagonal matrix with positive values (i.e $\Sigma_{ii} > 0$) such that:

$$U\Sigma V^\top = SVD(YX^\top)$$

So :

$$W^* = \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmax}} \langle W, U\Sigma V^\top \rangle = \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmax}} \langle U^\top WV, \Sigma \rangle$$

W, U, V are orthogonal so $P = U^\top WV$ is orthogonal and we have : $\langle P, \Sigma \rangle = \sum_{i=1}^n P_{ii}\Sigma_{ii}$
We know that if P is orthogonal , $\forall i, j \in 1, 2, ..., n, \quad |P_{i,j}| \leq 1$ So

$$\langle P, \Sigma \rangle \leq \sum_{i=1}^n \Sigma_{ii}$$

With equality in the case $P^* = \mathbb{I}_d$, hence

$$W^* = \underset{W \in \mathcal{O}_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^\top$$

- **Question 2**:
The training and validation accuracies of the best model are reported in the table below:

Model	Average	IDF Weighted-average
Training Accuracy	43.05 %	47 %
Dev Accuracy	38.41 %	39.87 %

Table 1: Models Comparison

- **Question 3**:
  For the loss, I used the categorical cross entropy, the expression of this loss is:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}\mathbb{1}_{y_i \in C_k}\log P_{model}[y_i \in C_k]$$

Where N: number of observations.
C: number of classes in this case C = 5.
y_i : the true label.
P_{model} : The probability predicted by our model.

- **Question 4**:
  the evolution of train/dev results w.r.t the number of epochs.

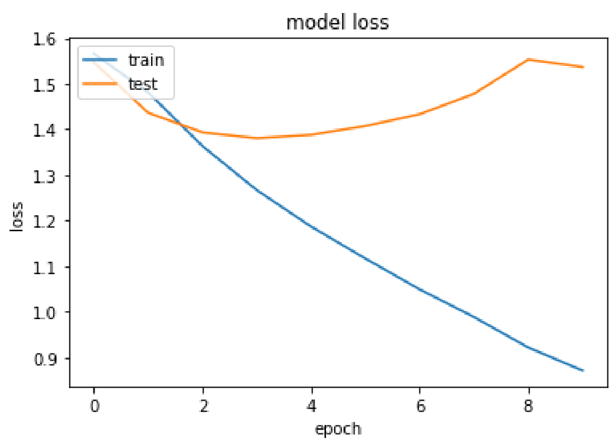

Figure 1: the evolution of train/dev loss w.r.t the number of epochs.

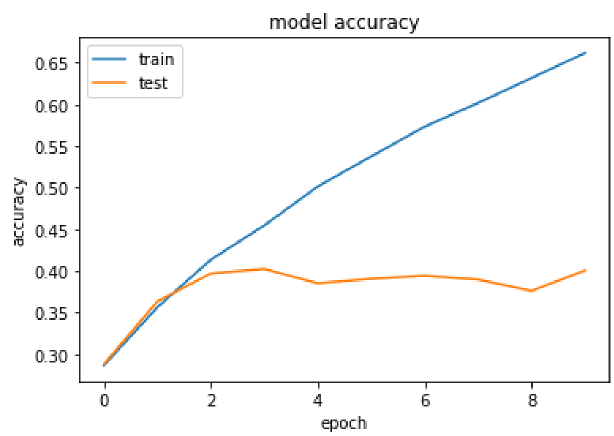

Figure 2: the evolution of train/dev accuracy w.r.t the number of epochs.

- **Question 5**: I modified slightly the previous architecture, I added a 1D CNN followed by a Maxpooling layer, and a Bidirectional LSTM, I added Dropout in some layers to prevent overfitting. The validation accuracy slightly outperformes the previous architecture.