Homework NLP

Multilingual word embeddings

Question

Let's show that

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T, \text{ with } U\Sigma V^T = \text{SVD}\left(YX^T\right)$$

Let $W \in O_d(\mathbb{R}$. We first notice that :

$$\begin{aligned}\|WX - Y\|_F^2 &= \|WX\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle \\ &= \|X\|_F^2 + \|Y\|_F^2 - 2\langle WX, Y\rangle\end{aligned}$$

Where $\langle WX, Y\rangle = \text{Tr}\left((WX)^T Y\right) = \text{Tr}\left(W^T Y X^T\right)$. Hence,

$$\begin{aligned}\underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F &= \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F^2 \\ &= \underset{W \in O_d(\mathbb{R})}{\operatorname{argmax}} \text{Tr}\left(W^T Y X^T\right)\end{aligned}$$

Let $U\Sigma V^T$ be the SVD decomposition of YX^T.

$$\begin{aligned}\text{Tr}\left(W^T Y X^T\right) &= \text{Tr}\left(W^T U\Sigma V^T\right) \\ &= \text{Tr}\left(V^T W^T U\Sigma\right)\end{aligned}$$

Let $Z = W^T U\Sigma$. Z is orthogonal as the product of orthogonal matrices. $\text{Tr}\left(Z\Sigma\right) = \sum_{i=1}^n Z_{i,i}\Sigma_{i,i}$. $(\Sigma_{i,i})_i$ are non-negative and Z is orthogonal hence the previous quantity is maximized if $Z_{i,i} = 1$ which means that $Z = I_n$. We then conclude that $W^* = UV^T$.

Sentence classification with BoW

Question

We test the sentence classification task using both the average of word vectors and the idf-weighted-average .

We get the following results :

Dataset	Average	idf
Train	0.4641	0.4904
Dev	0.4078	0.4078

Table 1: BoW Accuracy

We get better results using the idf-weighed average of word vectors on Train set, on dev set we get the same performance, we'll keep the idf-weighed results.

Deep Learning models for classification

0.1 Question

We use the cross-entropy loss function (per Pytorch's documentation) :

$$loss(x, class) = -log\left(\frac{exp(x[class])}{\sum_j exp(x[j])}\right) = -x[class] + log\left(\sum_j exp(x[j])\right)$$

It is good to notice here the we don't need any one-hot coding or a softmax layer on our model as all of this is taken care of by the CrossEntropyLoss in pytorch.

0.2 Question

The model seems to start overfitting quickly as we can see that the Validation error starts going up while Train error keeps going down.

Figure 1: Train and Validation Losses .

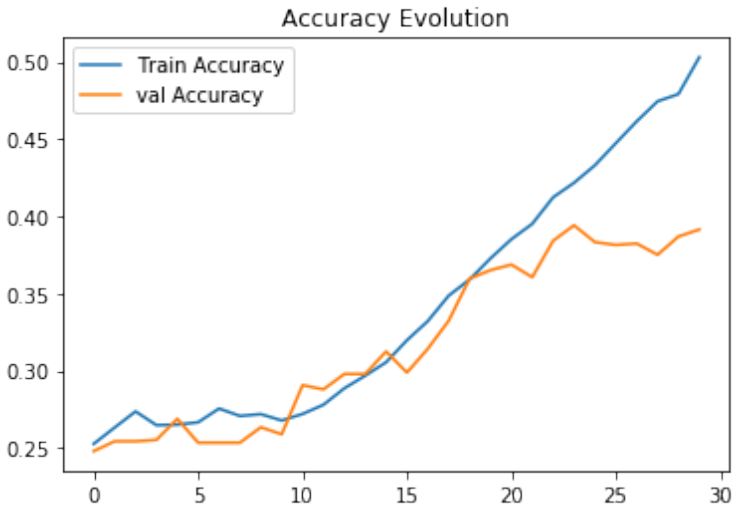

Figure 2: Train and Validation accuracy .

0.3 Innovate

model
we'll Use more or less the same model as before the only change we'll introduce is by using pre-trained word embeddings (we'll use Glove, which is publically available at Glove Project)
One thing worth mentioning is that since we use pretrained embedding we need to recreate our datasets such that the words and indexes match the ones used in the pretrained embedding. We get a better performance on dev set by using this as the accuracy goes from **40.3%** to **45.04%**