TP2- Deep learning

Idriss MGHABBAR

December 2018

1 Monolingual embeddings

On notebook.

2 Multilingual word embeddings

$$\begin{aligned} argmin_W \|WX - Y\|^2 &= argmin_W (WX - Y) \cdot (WX - Y) \\ &= argmin_W \|X\|^2 + \|Y\|^2 - 2Y \cdot WX \\ &= argmax_W Y \cdot WX \\ &= argmax_W tr(X^T W^T Y) \\ &= argmax_W tr(W^T Y X^T) \\ &= argmax_W W \cdot YX^T \\ &= argmax_W W \cdot USV^T \\ &= argmax_W U^T WV \cdot S \end{aligned} \tag{1}$$

Where $\cdot$ means the frobenius dot product between two matrices. $U^T WV$ is a product of three orthogonal matrices, it is hence orthogonal. The last dot product is then maximized for:

$$U^T WV = I \tag{2}$$

$$W = UV^T \tag{3}$$

3 Sentence classification with BoV

When using the mean-BoV, we get :

- **Training error** : 0.387
- **Validation erorr** : 0.421

When using the weighted average, we get:

- **Training error** : 0.385
- **Validation erorr** : 0.434

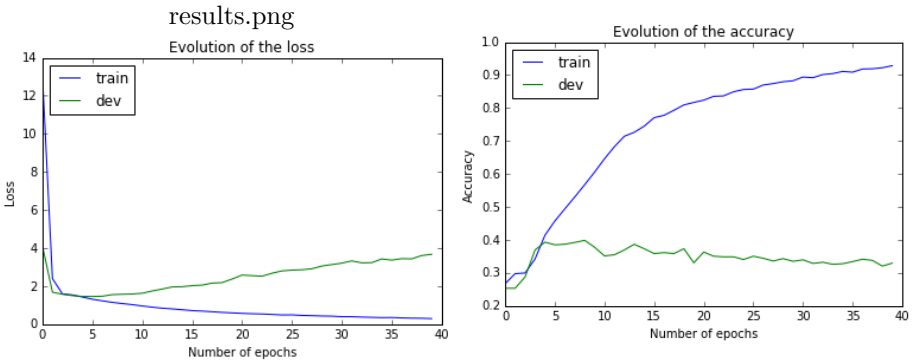
4 Deep Learning models for classification

- **The loss used** is categorical cross-entropy :

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log(\hat{y}_{ij}) \tag{4}$$

  Where : i indexes the samples, j indexes the classes, y_{ij} is the true label and $\hat{y}_{ij}$ is the predicted j-th class probability .

- **The loss/accuracy plots** for the train/dev sets :



(a) Loss results

(b) Accuracy results

Figure 1: A figure with two subfigures

- **The new model** : For the preprocessing, we use Keras tokenizer instead of one hot encoding. We will use this architecture:
  - A trainable embedding layer initialized using pretrained embeddings (wiki.en.vec).
  - Conv1D layer using LeakyReLU activation followed by MaxPooling1D. The parameters used are: 128 filters, kernel size = 5 and pool size = 3. We also use l2 regularization on the weights (kernel regularizer).
  - Conv1D layer using LeakyReLU activation followed by GlobalAVerage-Pooling.Same values used but with higher regularization.
  - We use a dropout of p=0.5 to reduce overfitting.
  - Dense layer with softmax activation.

This architecture was motivated by the fact that conv1D enables to catch dependency patterns present in sentences.