Deep Learning for Natural Language Processing -Project

WANG Kewei

2

$$W = U\begin{pmatrix}\Sigma \\ 0\end{pmatrix}V^T$$

$$\|WX - Y\|_F = \left\| U\begin{pmatrix}\Sigma \\ 0\end{pmatrix}V^T X - Y \right\| = \left\| \begin{pmatrix}\Sigma \\ 0\end{pmatrix}V^T X - U^T Y \right\| = \left\| \begin{pmatrix}\Sigma \\ 0\end{pmatrix}V^T X - \left[U_N, U_{M-N}\right]^T Y \right\|$$

$$= \left\| \begin{pmatrix}\Sigma V^T X - U_N^T Y \\ -U_{M-N}^T Y\end{pmatrix} \right\| = \left\| \Sigma V^T X - U_N^T Y \right\| + \left\| U_{M-N}^T Y \right\| \geqslant \left\| U_{M-N}^T Y \right\|$$

$$When \left\| \Sigma V^T X - U_N^T Y \right\| = 0, \; the\ expression\ will\ be\ minimum:$$

$$\Sigma V^T X - U_N^T Y = 0$$

$$\Sigma V^T X = U_N^T Y$$

$$YX^T = U\Sigma V^T$$

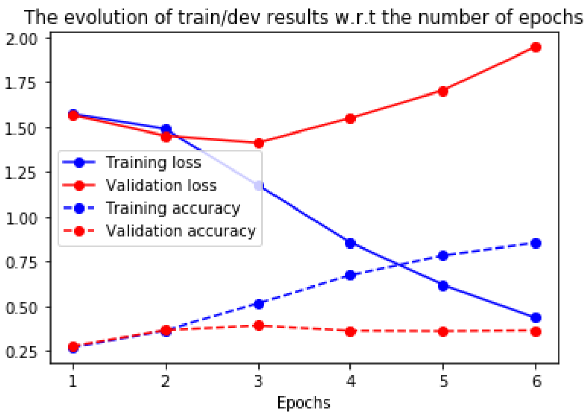
$$SVD\left(YX^T\right) = U\Sigma V^T$$

4(1)

The loss I use is Categorical cross-entropy. The mathematical expression is:

$$L_i = -\sum_j t_{i,j} log(p_{i,j})$$

(2)



(3)

The Tokenizer enables me to use the word2vec, and this bedding function proves to perform better.