Word Embedding

L.J. Brown

September 23, 2017

We outline our first attempt at a method to map all unique words found in a corpus to vectors in a continuous vector space. The hope is that some relationships between words found in the corpus will be preserved through this mapping and will manifest as characteristics of the vector space. The method involves building a co-occurrence matrix from the corpus that represents how frequently word pairs occur together. We then search for word vectors with the soft constraint that given a word vector pair, their inner product will yield a value close to the two values in the co-occurrence matrix associated with those two words. We do this using Stochastic Gradient Descent. The method we outline draws heavily on the implementations by "Word2vec" and "GloVe".

We first map all n unique words found in the corpus, $W = \{w_1, \ldots, w_n\}$, to integers, $1, \ldots, n$.

$$n \equiv \text{Number of unique words in corpus}$$

$$\text{for } w_i \, \epsilon \, W \mid w_i \to i$$

These mappings will correspond to the rows and columns in the co-occurrence matrix $\boldsymbol{Y}_{n \times n}$.

$$\boldsymbol{Y}_{n \times n} = \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \ldots & c_n \end{array} \\ \begin{pmatrix} y_{11} & y_{12} & \ldots & y_{1n} \\ y_{21} & y_{22} & \ldots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \ldots & y_{nn} \end{pmatrix} \end{array}$$

where w_i corresponds to r_i and c_i

Next we build the matrix $\boldsymbol{Y}_{n \times n}$. For each of the m sentences in the corpus, $S = \{s_1, \ldots, s_m\}$, with lengths $L = \{l_1, \ldots, l_m\}$, we find all $\binom{l_i}{2}$ word pairs, $P = \{p_{1s_i}, \ldots, p_{\binom{l_i}{2}s_i}\}$, and define a distance function, $f(p_{is_i})$, that returns a value one greater than the number of words separating the words in the word pair p_{is_i}. Then,

$$y_{rc} = y_{cr} = \sum_{i=1}^{m} \sum_{j=1}^{\binom{l_i}{2}s_i} \begin{cases} \log\left(\frac{1}{f(p_{js_i})}\right), \text{ if } r \text{ and } c \text{ correspond to the integer mappings of the words found in the wo} \\ 0, \textit{otherwise} \end{cases}$$

Notice $y_{i,j} = y_{j,i}$ in the matrix $\boldsymbol{Y}$ making it symmetric about the diagonal.

Next we search for a set of n word vectors, $w_1, \ldots, w_n$, of dimensionality d that best satisfy the condition that the inner product between any two word vectors is a value close to the two elements in matrix $\boldsymbol{Y}$ corresponding to the two words intersection. We exclude the diagonal elements from meeting this condition which correspond to the intersection of one word with itself. To achieve this we define a set of $n^2 - n$ soft constraints, $C_{12}, C_{13}, \cdots, C_{n(n-1)}$, as

$$C_{ij} \equiv \vec{w}_i^T \cdot \vec{w}_j :\Leftrightarrow y_{ij}, \text{ where } i \neq j.$$

We define an error function, ε, between two word vectors, $\vec{w}_i$ and $\vec{w}_j$, as

$$\varepsilon(i,j) = \vec{w}_i^T \cdot \vec{w}_j - y_{ij}$$

where y_{ij} is an element in the Matrix $\boldsymbol{Y}$

We define an objective function, J, to minimize using **Stochastic Gradient Descent** as

$$J = \frac{1}{2\binom{n}{2}} \sum_{r=1}^{n} \sum_{c=r+1}^{n} \begin{cases} \varepsilon(r,c)^2, \, if \, r \neq c \\ 0, \, otherwise \end{cases}$$

We define the objective function for an individual word vector, $\vec{w}_i$, as

$$J_{\vec{w}_i} = \frac{1}{2} \sum_{j=1}^{n} \begin{cases} \varepsilon(i,j)^2, \, if \, i \neq j \\ 0, \, otherwise \end{cases}$$

$$\frac{\partial J_{\vec{w}_i}}{\partial \vec{w}_i} = \sum_{j=1}^{n} \begin{cases} \vec{w}_i \odot \left(\vec{w}_i^T \cdot \vec{w}_j - y_{ij} \right), \, if \, i \neq j \\ 0, \, otherwise \end{cases}$$

$m \equiv$ Number of training iterations
$\eta \equiv$ Learning rate

For $i = 1, 2, \ldots, m$, do:

$$\vec{w}_i := \vec{w}_i - \eta \frac{\partial J_{\vec{w}_i}}{\partial \vec{w}_i}$$