doc2vec(paragraph vector)

September 17, 2016

1 Introduction

Distributed memory(DM) version of doc2vec using hierarchical softmax. The document and word vectors are averaged here to calculate the hidden layer h.

Hierarchical softmax uses a binary tree model, all V words are leafs of the tree. The unique path to the target word is used to estimate the probability of the word. Assume that $n(w, j)$ is the j-th node on the unique path from the root to word w, and that $L(w)$ is the length of this path, then $n(w, 1)$ represents the root node and $n(w, L(w))$ the leaf node w. For any inner node n, $ch(n)$ denotes an arbitrary fixed child of n, for example always the left child node. $[\![x]\!]$ is 1 if x is true and -1 otherwise. $v_{w_{t-k,j}}$ are used to calculate the hidden layer and are also called input vectors of words, $v^{'}_{n(w_{t,j},l)}$ are used to calculate the probability of a word in hierarchical softmax, they are also called output vectors of words.

J is the number of documents, N_j the number of words in document j, h is the hidden layer.C_p is the number of word vectors that contribute to the hidden layer (2*k (context words)) + 1 (for the document vector).

2 Math

$$L = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{N_j - 2k}\sum_{t=k}^{N_j-k} log\, p(w_{t,j}|w_{t-k,j}, ..., w_{t+k,j}, d_j) \tag{1}$$

$$= \frac{1}{J}\sum_{j=1}^{J}\frac{1}{N_j - 2k}\sum_{t=k}^{N_j-k}\sum_{l=1}^{L(w_{t,j})-1} log\, \sigma([\![n(w_{t,j}, l+1) = ch(n(w_{t,j}, l))]\!] \cdot {v^{'}_{n(w_{t,j},l)}}^{T} h) \tag{2}$$

h in case of averaging of the word and document vectors, $v_{d_j} \equiv \theta_j$

$$= \frac{1}{J}\sum_{j=1}^{J}\frac{1}{N_j - 2k}\sum_{t=k}^{N_j-k}\sum_{l=1}^{L(w_{t,j})-1} log\, \sigma([\![n(w_{t,j}, l{+}1) = ch(n(w_{t,j}, l))]\!] \cdot {v^{'}_{n(w_{t,j},l)}}^{T} \frac{1}{C_p}\,(v_{w_{t-k,j}}{+}...{+}v_{w_{t+k,j}}{+}v_{d_j})) \tag{3}$$

$$\frac{\partial L}{\partial v_{d_j}} = \frac{1}{J}\frac{1}{N_j - 2k}\sum_{t=k}^{N_j-k}\sum_{l=1}^{L(w_{t,j})-1} \big(1{-}\sigma([\![n(w_{t,j}, l{+}1) = ch(n(w_{t,j}, l))]\!] \cdot {v^{'}_{n(w_{t,j},l)}}^{T} \frac{1}{C_p}\,(v_{w_{t-k,j}}{+}...{+}v_{w_{t+k,j}}{+}v_{d_j}))\big) \cdot [\![n(w_{t,j}, l+1) = ch(n(w_{t,j}, l))]\!] v^{'}_{n(w_{t,j},l)} \frac{1}{C_p}$$

$$= \frac{1}{J} \frac{1}{N_j - 2k} \sum_{t=k}^{N_j - k} \sum_{l=1}^{L(w_{t,j})-1} \left(y_{t,l} - \sigma({v'_{n(w_{t,j},l)}}^T \frac{1}{C_p} (v_{w_{t-k,j}} + ... + v_{w_{t+k,j}} + v_{d_j}))\right) \cdot v'_{n(w_{t,j},l)} \frac{1}{C_p} \quad (4)$$

,where $y_{t,l}$ is 1 if $[\![n(w_{t,j}, l+1) = ch(n(w_{t,j}, l))]\!] = 1$ and 0 otherwise.

$$= \frac{1}{J} \frac{1}{N_j - 2k} \sum_{t=k}^{N_j - k} \sum_{l=1}^{L(w_{t,j})-1} \left(y_{t,l} - \frac{1}{1 + \exp\left(-{v'_{n(w_{t,j},l)}}^T \frac{1}{C_p} (v_{w_{t-k,j}} + ... + v_{w_{t+k,j}} + v_{d_j})\right)}\right) \cdot v'_{n(w_{t,j},l)} \frac{1}{C_p} \quad (5)$$

Update of v_{d_j} by gradient ascent:

$$v_{d_j}^{\tau+1} = v_{d_j}^{\tau} +$$

$$\eta \cdot \left(\frac{1}{J} \frac{1}{N_j - 2k} \sum_{t=k}^{N_j - k} \sum_{l=1}^{L(w_{t,j})-1} \left(y_{t,l} - \frac{1}{1 + \exp\left(-{v'_{n(w_{t,j},l)}}^T \frac{1}{C_p} (v_{w_{t-k,j}} + ... + v_{w_{t+k,j}} + v_{d_j}^{(\tau)})\right)}\right) \cdot v'_{n(w_{t,j},l)} \frac{1}{C_p}\right)$$

,where η is the learning rates and τ describes the time points