

word2vec Models *

Khanh Nguyen

June 11, 2015

There are two *word2vec* models: skip-gram and continuous bag-of-words (CBOW). They both manage to capture interactions between a centered word and its context words. However, they do it differently, and somehow oppositely. While skip-gram models the distribution of context words given the centered word, CBOW is concerned about predicting the centered word given its context.

In this note, I will make no distinction in notation between a word and its corresponding vector. For example, w denotes both the word and its word vector. I treat distinct word vectors as distinct words.

1 General model

Given a predicted word vector \hat{r} and a target word vector w_t . The probability of the target word conditional on the predicted word is calculated by a softmax function:

$$P(w_t|\hat{r}) = \frac{\exp(w_t^T \hat{r})}{\sum_{w \in W} \exp(w^T \hat{r})}$$

where W is the set of all target word vectors.

Notice that \hat{r} is neither an element of W nor computed from elements of W . As we will see later, elements of W will be called *output vectors* and \hat{r} will be computed from a different set consisting of *input vectors*.

word2vec models' cost functions (for one target word) minimize the negative log-likelihood of the target word vector given its corresponding predicted word:

$$\mathcal{L}(w_t, \hat{r}) = -\log P(w_t|\hat{r}) = \log \left(\sum_{w \in W} \exp(w^T \hat{r}) \right) - w_t^T \hat{r}$$

The gradient with respect to w of $\mathcal{L}(w_t, \hat{r})$ is:

$$g_1(w, w_t, W, \hat{r}) = \frac{\partial}{\partial w} \mathcal{L}(w_t, \hat{r}) = \hat{r} (P(w|\hat{r}) - I\{w = w_t\}) \quad (1)$$

where $I\{.\}$ is the indicator function.

*This material is for self-study only.

The gradient with respect to \hat{r} is:

$$g_2(w_t, W, \hat{r}) = \frac{\partial}{\partial \hat{r}} \mathcal{L}(w_t, \hat{r}) = \sum_{w \in W} [P(w|\hat{r})w] - w_t \quad (2)$$

2 Skip-gram

For an index i and a window size c , skip-gram predicts the context words $\{w_j\}$, ($i - c \leq j \leq i + c, j \neq i$) given the centered word r_i . Hence, in the general model, $w_t = w_j$ and $\hat{r} = r_i$ for this case.

The cost function is derived as follows:

$$\mathcal{L}_{skipgram}(c, i) = \sum_{i-c \leq j \leq i+c, i \neq j} -\log P(w_j | r_i)$$

The gradients of this function are:

$$\frac{\partial}{\partial w} \mathcal{L}_{skipgram}(c, i) = r_i \sum_{i-c \leq j \leq i+c, i \neq j} g_1(w, w_j, W, r_i) \quad (3)$$

$$\frac{\partial}{\partial r_i} \mathcal{L}_{skipgram}(c, i) = \sum_{i-c \leq j \leq i+c, i \neq j} g_2(w_j, W, r_i) \quad (4)$$

3 Continuous bag-of-words

Intuitively, this model reverses the modeling mechanism of skip-gram. CBOW predicts a word given its context. The target word vector is now the output vector of the word at index i , w_i ; the predicted word vector is the sum over all context input vectors:

$$\hat{r} = \sum_{i-c \leq j \leq i+c, i \neq j} r_j$$

The CBOW's cost function is as follows:

$$\mathcal{L}_{CBOW}(c, i) = -\log P(w_i | \hat{r})$$

and the gradients are: student

$$\frac{\partial}{\partial w} \mathcal{L}_{CBOW}(c, i) = g_1(w, w_i, W, \hat{r}) \quad (5)$$

$$\frac{\partial}{\partial r_j} \mathcal{L}_{CBOW}(c, i) = g_2(w_i, W, \hat{r}) \quad (6)$$

for $i - c \leq j \leq i + c, i \neq j$. Otherwise,

$$\frac{\partial}{\partial r_j} \mathcal{L}_{CBOW}(c, i) = 0 \quad (7)$$

4 Negative sampling

Read [word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method](#) (Goldberg and Levy, 2014) for the motivation of negative sampling.