1 Understanding word2vec

The key insight behind word2vec is that 'a word is known by the company it keeps'. Concretely, suppose we have a 'center' word c and a contextual window surrounding c. We shall refer to words that lie in this contextual window as 'outside words'. For example, in Figure 1 we see that the center word c is 'banking'. Since the context window size is 2, the outside words are 'turning', 'into', 'crises', and 'as'.
The goal of the skip-gram word2vec algorithm is to accurately learn the probability distribution $P(O|C)$. Given a specific word o and a specific word c, we want to calculate $P(O = o|C = c)$, which is the probability of the word o is an 'outside' word for c, i.e., the probability that o falls within the contextual windows of c.

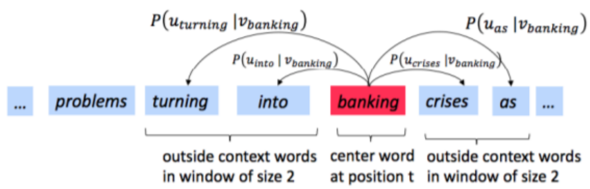

Figure 1: The word2vec skip-gram prediction model with windows size 2

In word2vec, the conditional probability distribution is given by taking vector dot-products and applying the softmax function:

$$P(O = o|C = c) = \frac{exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum\limits_{w \in Vocab} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \tag{1}$$

Here, $\boldsymbol{u}_o$ is the 'outside' vector representing outside word o, and v_c is the center 'center' vector representing center word c. To contain these parameters, we have two matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$. The columns of $\boldsymbol{U}$ are all the 'outside' vectors u_w. The columns of $\boldsymbol{V}$ are all the 'center' vectors v_w. Both $\boldsymbol{U}$ and $\boldsymbol{V}$ contain a vector for every $w \in Vocabulary$.[1]
Recall that, for a single pair of words c and o, the loss is given by:

$$\boldsymbol{J}_{naive-softmax}(\boldsymbol{v_c}, \boldsymbol{o}, \boldsymbol{U}) = -\log P(O = o|C = c). \tag{2}$$

Another way to view this loss is the cross-entropy[2] between the true distribution $\boldsymbol{y}$ and the predicted distribution $\hat{\boldsymbol{y}}$. Here, both $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the vectors with length equal to the number of words in the vocabulary. Furthermore, the k^{th}

[1] Assume that every word in our vocabulary is matched to an integer number k. $\boldsymbol{u}_k$ is both the k^{th} column of $\boldsymbol{U}$ and 'outside' word vector for the word indexed by k. $\boldsymbol{v}_k$ is both the k_{th} column of $\boldsymbol{V}$ and the 'center' word vector for the word indexed by k. In order to simplify notation we shall interchangeably use k to refer to the word and the index-of-the-word.

[2] The Cross Entropy Loss between the true (discrete) probability distribution p and another probability distribution q is $-\Sigma_i p_i \log(q_i)$.

entry in these vectors indicates the conditional probability of the k^{th} word being an 'outside word' for the given c. The true empirical distribution $\boldsymbol{y}$ is a one-hot vector with a 1 for the true outside word o, and 0 everywhere else. The predicted distribution $\hat{\boldsymbol{y}}$ is the probability distribution $P(O|C=c)$ given by our model in the Equation 1.

(a) Naive softmax loss in the Equation 2 is the same as the cross-entropy loss between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$; i.e.,

$$-\sum_{w \in Vocab} y_w \log \hat{y}_w = -log(\hat{y}_o) \tag{3}$$

Reason: For two words c and o, $y_w = 1$ when $w = o$, $y_w = 0$ otherwise.

(b) Partial derivative of $\boldsymbol{J}_{naive-softmax}(\boldsymbol{v_c}, \boldsymbol{o}, \boldsymbol{U}) = J$ with respect to (v_c). [3]

Let us rewrite J as:

$$\begin{aligned} J &= -\log \hat{y}_o = -\log(\frac{exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum\limits_{w \in Vocab} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}) \\ &= -[\log(exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \log(\sum_{w \in Vocab} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c))] \\ &= -\boldsymbol{u}_o^\top \boldsymbol{v}_c + \log(\sum_{w \in Vocab} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)) \end{aligned} \tag{4}$$

Now, let us find $\frac{\partial J}{\partial \boldsymbol{v_c}}$. [4]

$$\frac{\partial J}{\partial \boldsymbol{v_c}} = -\boldsymbol{u}_o + \frac{\sum\limits_{w \in Vocab} exp(\boldsymbol{u}_w \boldsymbol{v}_c).\boldsymbol{u}_w}{\sum\limits_{w \in Vocab} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \tag{5}$$

Re-arranging,

$$\frac{\partial J}{\partial \boldsymbol{v_c}} = -\boldsymbol{u}_o + \sum_{w \in Vocab} (\frac{exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\sum\limits_{x \in Vocab} exp(\boldsymbol{u}_x^\top \boldsymbol{v}_c)} \boldsymbol{u}_w) \tag{6}$$

From Equation 1,

[3] For more details visit: https://stats.stackexchange.com/questions/253244/gradients-for-skipgram-word2vec

[4] If you are confused about why $\frac{\partial \boldsymbol{u}_o^\top \boldsymbol{v}_c}{\partial \boldsymbol{v_c}} = \boldsymbol{u}_o$, please read about numerator layout notation and denominator layout notation. You may start here: https://www.comp.nus.edu.sg/ cs5240/lecture/matrix-differentiation.pdf

$$\frac{\partial J}{\partial \boldsymbol{v_c}} = -\boldsymbol{u}_o + \sum_{w \in Vocab} (\hat{y_w} \boldsymbol{u}_w) \tag{7}$$

Writing in the matrix form,

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{v_c}} &= -Uy + U\hat{y} \\ &= U[\hat{y} - y] \end{aligned} \tag{8}$$