

1 Understanding word2vec

Let's have a quick refresher on the word2vec algorithm. 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 that word o is an 'outside' word for c , i.e., the probability that o falls within the contextual window of c .

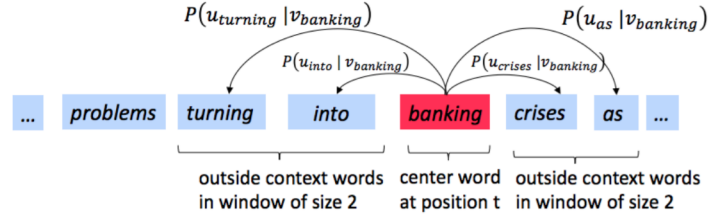


Figure 1: The word2vec skip-gram prediction model with window size 2

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

$$P(O = o|C = c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in \text{Vocab}} \exp(u_w^T v_c)} \quad (1)$$

Here, \mathbf{u}_o is the 'outside' vector representing outside word o , and \mathbf{v}_c is the 'center' vector representing center word c . To contain these parameters, we have two matrices, \mathbf{U} and \mathbf{V} . The columns of \mathbf{U} are all the 'outside' vectors \mathbf{u}_w . The columns of \mathbf{V} are all of the 'center' vectors \mathbf{v}_w . Both \mathbf{U} and \mathbf{V} contain a vector for every $w \in \text{Vocabulary}$. Recall from lectures that, for a single pair of words c and o , the loss is given by:

$$\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U}) = -\log P(O = o|C = c) \quad (2)$$

We can view this loss as the cross-entropy between the true distribution \mathbf{y} and the predicted distribution $\hat{\mathbf{y}}$. Here, both \mathbf{y} and $\hat{\mathbf{y}}$ are vectors with length equal to the number of words in the vocabulary. Furthermore, the k^{th} 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 \mathbf{y} is a one-hot vector with a 1 for the true outside word o , and 0 everywhere else. The predicted distribution $\hat{\mathbf{y}}$ is the probability distribution $P(O|C = c)$ given by our model in equation (1).

- (a) Show that the naive-softmax loss given in Equation (2) is the same as the cross-entropy loss between \mathbf{y} and $\hat{\mathbf{y}}$; i.e., show that

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

Answer :

The true empirical distribution \mathbf{y} is a one-hot vector with a 1 for the true outside word o , and 0

everywhere else. So we can write :

$$y_w = \mathbb{1}_{\{w=o\}}$$

and then,

$$-\sum_w y_w \log(\hat{y}) = -y_o \log(\hat{y}_o) = -\log(\hat{y}_o)$$

- (b) Compute the partial derivative of $\mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U})$ with respect to \mathbf{v}_c . Please write your answer in terms of \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{U} . Note that in this course, we expect your final answers to follow the shape convention. This means that the partial derivative of any function $f(\mathbf{x})$ with respect to \mathbf{x} should have the same shape as \mathbf{x} . For this subpart, please present your answer in vectorized form. In particular, you may not refer to specific elements of \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{U} in your final answer (such as \mathbf{y}_1 , \mathbf{y}_2 , \dots).

Answer :

$$\begin{aligned} \frac{\partial}{\partial v_c} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial v_c} [-u_o^T v_c + \log(\sum_w \exp(u_w^T v_c))] \\ &= -u_o + \sum_x \frac{\exp(u_x^T v_c)}{\sum_w \exp(u_w^T v_c)} u_x \\ &= -u_o + \sum_x P(O = x | C = c) u_x \\ &= -\mathbf{y}^T \mathbf{U}^T + \hat{\mathbf{y}}^T \mathbf{U}^T \\ &= \mathbf{U}^T (\hat{\mathbf{y}} - \mathbf{y})^T \end{aligned}$$

- (c) Compute the partial derivatives of $\mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U})$ with respect to each of the ‘outside’ word vectors, \mathbf{u}_w ’s. There will be two cases: when $w = o$, the true ‘outside’ word vector, and $w \neq o$, for all other words. Please write your answer in terms of \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{v}_c . In this subpart, you may use specific elements within these terms as well, such as $(\mathbf{y}_1, \mathbf{y}_2, \dots)$.

Answer :

$$\frac{\partial}{\partial u_w} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) = \frac{\partial}{\partial u_w} [-u_o^T v_c + \log(\sum_w \exp(u_w^T v_c))]$$

if $w = o$,

$$\begin{aligned} \frac{\partial}{\partial u_w} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) &= -v_c + \frac{\exp(u_o^T v_c)}{\sum_w \exp(u_w^T v_c)} v_c \\ &= -v_c + P(O = o | C = c) v_c \\ &= v_c (P(O = o | C = c) v_c - 1) \end{aligned}$$

if $w \neq o$,

$$\begin{aligned}\frac{\partial}{\partial u_w} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) &= 0 + \frac{\exp(u_w^T v_c)}{\sum_w u_w^T v_c} v_c \\ &= 0 + P(O = w / C = c) v_c \\ &= (P(O = w / C = c) v_c\end{aligned}$$

so for any w ,

$$\frac{\partial}{\partial u_w} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) = (\hat{y}_w - y_w) v_c$$

- (d) Compute the partial derivative of $\mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U})$ with respect to \mathbf{U} . Please write your answer in terms of $\frac{\partial}{\partial u_1} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U})$, $\frac{\partial}{\partial u_2} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U})$, ..., $\frac{\partial}{\partial u_{|V_{ocab}|}} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U})$. The solution should be one or two lines long.

Answer :

$\frac{\partial}{\partial \mathbf{U}} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U})$ must be of the shape of \mathbf{U} :

$$\frac{\partial}{\partial \mathbf{U}} \mathbf{J}_{naive-softmax}(\mathbf{v}_c, o, \mathbf{U}) = \left(\frac{\partial}{\partial u_1} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U}) \quad \frac{\partial}{\partial u_2} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U}) \quad \dots \quad \frac{\partial}{\partial u_{|V_{ocab}|}} \mathbf{J}(\mathbf{v}_c, o, \mathbf{U}) \right)$$

- (e) The sigmoid function is given by Equation 4:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \quad (4)$$

Please compute the derivative of $\sigma(x)$ with respect to x , where x is a scalar. Hint: you may want to write your answer in terms of $\sigma(x)$.

Answer :

$$\begin{aligned}\sigma'(x) &= \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \frac{-1 + 1 + e^{-x}}{(1 + e^{-x})(1 + e^{-x})} \\ &= \sigma(x) \frac{-1 + 1 + e^{-x}}{1 + e^{-x}} \\ &= \sigma(x)(1 - \sigma(x))\end{aligned}$$

- (f) Now we shall consider the Negative Sampling loss, which is an alternative to the Naive Softmax loss. Assume that K negative samples (words) are drawn from the vocabulary. For simplicity of notation we shall refer to them as w_1, w_2, \dots, w_K and their outside vectors as $\mathbf{u}_1, \dots, \mathbf{u}_K$. For this question,

assume that the K negative samples are distinct. In other words, $i \neq j$ implies $w_i \neq w_j$ for $i, j \in \{1, \dots, K\}$. Note that $o \neq w_1, \dots, w_K$. For a center word c and an outside word o , the negative sampling loss function is given by:

$$\mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) = -\log(\sigma(u_o^T v_c)) - \sum_k \log(\sigma(-u_k^T v_c)) \quad (5)$$

for a sample w_1, \dots, w_K where $\sigma'(\cdot)$ is the sigmoid function.

Please repeat parts (b) and (c), computing the partial derivatives of $\mathbf{J}_{neg-sample}$ with respect to \mathbf{v}_c , with respect to \mathbf{u}_o , and with respect to a negative sample \mathbf{u}_k . Please write your answers in terms of the vectors \mathbf{u}_o , \mathbf{v}_c , and \mathbf{u}_k , where $k \in [1, K]$. After you've done this, describe with one sentence why this loss function is much more efficient to compute than the naive-softmax loss. Note, you should be able to use your solution to part (e) to help compute the necessary gradients here.

Answer :

$$\begin{aligned} \frac{\partial}{\partial v_c} \mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial v_c} [-\log(\sigma(u_o^T v_c)) - \sum_{k=1}^K \log(\sigma(-u_k^T v_c))] \\ &= \frac{-u_o \sigma(u_o^T v_c) [1 - \sigma(u_o^T v_c)]}{\sigma(u_o^T v_c)} - \sum_{k=1}^K \frac{-u_k \sigma(-u_k^T v_c) [1 - \sigma(-u_k^T v_c)]}{\sigma(-u_k^T v_c)} \\ &= -u_o [1 - \sigma(u_o^T v_c)] + \sum_{k=1}^K u_k [1 - \sigma(-u_k^T v_c)] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial u_o} \mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial u_o} [-\log(\sigma(u_o^T v_c)) - \sum_{k=1}^K \log(\sigma(-u_k^T v_c))] \quad (k \neq o) \\ &= \frac{-v_c \sigma(u_o^T v_c) [1 - \sigma(u_o^T v_c)]}{\sigma(u_o^T v_c)} - 0 \\ &= -v_c [1 - \sigma(u_o^T v_c)] \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial u_k} \mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial u_k} [-\log(\sigma(u_o^T v_c)) - \sum_{i=1}^K \log(\sigma(-u_i^T v_c))] \quad (k \neq o) \\ &= 0 + \frac{v_c \sigma(-u_k^T v_c) [1 - \sigma(-u_k^T v_c)]}{\sigma(-u_k^T v_c)} \\ &= v_c [1 - \sigma(-u_k^T v_c)] \end{aligned}$$

This loss function is much more efficient to compute than the naive-softmax loss because we don't need to go through the all vocabulary.

- (g) Now we will repeat the previous exercise, but without the assumption that the K sampled words are distinct. Assume that K negative samples (words) are drawn from the vocabulary. For simplicity of notation we shall refer to them as w_1, w_2, \dots, w_K and their outside vectors as $\mathbf{u}_1, \dots, \mathbf{u}_K$. In this question, you may not assume that the words are distinct. In other words, $w_i = w_j$ may be true when $i \neq j$ is true. Note that $o \notin \{w_1, \dots, w_K\}$. For a center word c and an outside word o , the negative sampling loss function is given by:

$$\mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) = -\log(\sigma(u_o^T v_c)) - \sum_k \log(\sigma(-u_k^T v_c)) \quad (6)$$

Compute the partial derivative of $\mathbf{J}_{neg-sample}$ with respect to a negative sample \mathbf{u}_k . Please write your answers in terms of the vectors \mathbf{v}_c and \mathbf{u}_k , where $k \in [1, K]$. Hint: break up the sum in the loss function into two sums: a sum over all sampled words equal to \mathbf{u}_k and a sum over all sampled words not equal to \mathbf{u}_k

Answer :

$$\begin{aligned}\frac{\partial}{\partial v_c} \mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial v_c} [-\log(\sigma(u_o^T v_c)) - \sum_{k=1}^K \log(\sigma(-u_k^T v_c))] \\ &= \frac{-u_o \sigma(u_o^T v_c) [1 - \sigma(u_o^T v_c)]}{\sigma(u_o^T v_c)} - \sum_k \frac{-u_k \sigma(-u_k^T v_c) [1 - \sigma(-u_k^T v_c)]}{\sigma(-u_k^T v_c)} \\ &= -u_o [1 - \sigma(u_o^T v_c)] + \sum_k u_k [1 - \sigma(-u_k^T v_c)]\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial u_k} \mathbf{J}_{neg-sample}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial u_k} [-\log(\sigma(u_o^T v_c)) - \sum_{i=1}^K \log(\sigma(-u_i^T v_c))] \\ &= 0 + \frac{\partial}{\partial u_k} [-\sum_{j=l}^K \log(\sigma(-u_j^T v_c)) - \sum_{i=1}^l \log(\sigma(-u_i^T v_c))] \\ &\quad \text{(where } l \text{ is the number of time we have drawn } u_k\text{)} \\ &= 0 + l \frac{v_c \sigma(-u_k^T v_c) [1 - \sigma(-u_k^T v_c)]}{\sigma(-u_k^T v_c)} \\ &= l \cdot v_c [1 - \sigma(-u_k^T v_c)]\end{aligned}$$

- (h) Suppose the center word is $c = w_t$ and the context window is $[w_{t-m}, \dots, w_{t-1}, w_t, w_{t+1}, \dots, w_{t+m}]$, where m is the context window size. Recall that for the skip-gram version of word2vec, the total loss for the context window is:

$$\mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \mathbf{J}(\mathbf{v}_c, w_{t+j}, \mathbf{U}) \quad (7)$$

Here, $\mathbf{J}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$ represents an arbitrary loss term for the center word $c = w_t$ and outside word w_{t+j} . $\mathbf{J}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$ could be $\mathbf{J}_{naive-softmax}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$ or $\mathbf{J}_{neg-sample}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$, depending on your implementation.

Write down three partial derivatives:

- (i) $\partial \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{U}$
- (ii) $\partial \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_c$
- (iii) $\partial \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_w$ when $w \neq c$

Write your answers in terms of $\mathbf{J}(\mathbf{v}_c, w_{t+j}) / \partial \mathbf{U}$ and $\mathbf{J}(\mathbf{v}_c, w_{t+j}) / \partial \mathbf{v}_c$. This is very simple each solution should be one line.

Answer :

$$\begin{aligned}
(i) \quad & \frac{\partial}{\partial \mathbf{U}} \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial}{\partial \mathbf{U}} \mathbf{J}(\mathbf{v}_c, w_{t+j} \mathbf{U}) \\
(ii) \quad & \frac{\partial}{\partial v_c} \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial}{\partial v_c} \mathbf{J}(\mathbf{v}_c, w_{t+j} \mathbf{U}) \\
(iii) \quad & \frac{\partial}{\partial v_w} \mathbf{J}_{skip-gram}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial}{\partial v_w} \mathbf{J}(\mathbf{v}_c, w_{t+j} \mathbf{U}) = 0 \quad (w \neq c)
\end{aligned}$$