

# cs224n a2 written

## 1. Written: Understanding word2vec (26 points)

(a) (3 points) Show that the naive-softmax loss given in Equation (2) is the same as the cross-entropy loss between  $y$  and  $\hat{y}$ ; i.e., show that

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

Your answer should be one line.

**Solution:**

$$-\sum_{w \in \text{Vocab}} y_w \log(\hat{y}_w) = -\left( \sum_{\substack{w \in \text{Vocab} \\ w \neq o}} y_w \log(\hat{y}_w) + y_o \log(\hat{y}_o) \right) = -\log(\hat{y}_o)$$

(b) (5 points) Compute the partial derivative of  $J_{\text{naive-softmax}}(v_c, o, U)$  with respect to  $V_c$ . Please write your answer in terms of  $y$ ,  $\hat{y}$  and  $U$ . Note that in this course, we expect your final answers to follow the shape convention. This means that the partial derivative of any function  $f(x)$  with respect to  $x$  should have the same shape as  $x$ . For this subpart, please present your answer in vectorized form. In particular, you may not refer to specific elements of  $y$ ,  $\hat{y}$  and  $U$  in your final answer (such as  $y_1$ ,  $y_2$ , ...).

**Solution:**

$$\begin{aligned} \frac{\partial J}{\partial v_c} &= \frac{\partial -\log(O=o|C=c)}{\partial v_c} \\ &= -\frac{\partial \log \frac{\exp(u_o^T v_c)}{\sum_{w \in \text{Vocab}, w \neq o} \exp(u_w^T v_c)}}{\partial v_c} \\ &= -\frac{\partial \log \exp(u_o^T v_c)}{\partial v_c} + \frac{\partial \log \sum_{w=1}^V \exp(u_w^T v_c)}{\partial v_c} \\ &= -u_o + \frac{1}{\sum_{w=1}^V \exp(u_w^T v_c)} \frac{\partial \sum_{x=1}^V \exp(u_x^T v_c)}{\partial v_c} \\ &= -u_o + \frac{1}{\sum_{w=1}^V \exp(u_w^T v_c)} \sum_{x=1}^V \exp(u_x^T v_c) u_x \\ &= -u_o + \sum_{x=1}^V \frac{\exp(u_x^T v_c) u_x}{\sum_{w=1}^V \exp(u_w^T v_c)} \\ &= -u_o + \sum_{x=1}^V P(x|c) u_x \\ &= U(-y + \hat{y}) \end{aligned}$$

(c) (5 points) Compute the partial derivatives of  $J_{\text{naive-softmax}}(v_c, o, U)$  with respect to each of the 'outside' word vectors  $u_w$ '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  $y$ ,  $\hat{y}$  and  $v_c$ . In this subpart, you may use specific elements within these terms as well, such as ( $y_1$ ,  $y_2$ , ...).

**Solution:**

$$\begin{aligned} \frac{\partial J}{\partial u_w} &= \frac{\partial -\log(O=o|C=c)}{\partial v_c} \\ &= -\frac{\partial \log \frac{\exp(u_o^T v_c)}{\sum_{w \in \text{Vocab}, w \neq o} \exp(u_w^T v_c)}}{\partial u_w} \\ &= -\frac{\partial \log \exp(u_o^T v_c)}{\partial u_w} + \frac{\partial \log \sum_{w=1}^V \exp(u_w^T v_c)}{\partial u_w} \end{aligned}$$

when  $u_w = u_o$ :

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{u}_w} &= -v_c + \frac{1}{\sum_{w \in V_{ocab}} \exp(\mathbf{u}_o^T \mathbf{v}_c)} \sum_{w \in V_{ocab}} \exp(\mathbf{u}_o^T \mathbf{v}_c) v_c \\
&= -v_c + v_c P(O|C) \\
&= v_c (P(O|C) - 1) \\
&= v_c (\hat{y} - 1)
\end{aligned}$$

when  $\mathbf{u}_w \neq \mathbf{u}_o$ :

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{u}_w} &= 0 + \frac{\partial \log \sum_{w \in V_{ocab}, w \neq 0} \exp(\mathbf{u}_w^T \mathbf{v}_c)}{\partial \mathbf{u}_w} \\
&= \frac{\sum_{x \in V_{ocab}, x \neq 0} \exp(\mathbf{u}_x^T \mathbf{v}_c) v_c}{\sum_{w \in V_{ocab}, w \neq 0} \exp(\mathbf{u}_w^T \mathbf{v}_c)} \\
&= v_c P(O|C) \\
&= v_c \hat{y}
\end{aligned}$$

In the elements  $y_1, \dots, y_o, \dots, y_{|V_{ocab}|}$ , the value of  $y_o$  is 1 and other turns to be 0. So  $\frac{\partial J}{\partial \mathbf{u}_w}$  can be merged according to two situation above:

$$\frac{\partial J}{\partial \mathbf{u}_w} = v_c (\hat{y} - y)$$

(d) (1 point) Compute the partial derivative of  $J_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$  with respect to  $\mathbf{U}$ . Please write your answer in terms of  $\frac{\partial J(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_1}, \frac{\partial J(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_2}, \dots, \frac{\partial J(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_{|V_{ocab}|}}$ . The solution should be one or two lines long.

**Solution:**

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_o, \dots, \mathbf{u}_{|V_{ocab}|}]$$

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{U}} &= \left[ \frac{\partial J}{\partial \mathbf{u}_1}, \dots, \frac{\partial J}{\partial \mathbf{u}_o}, \dots, \frac{\partial J}{\partial \mathbf{u}_n} \right] \\
&= [v_c \hat{y}_1, \dots, v_c (\hat{y}_o - 1), \dots, v_c \hat{y}_{|V_{ocab}|}]
\end{aligned}$$

(e) (3 Points) 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  $\sigma(x)$ .

**Solution:**

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

(f) (4 points) 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  $o \notin \{w_1, \dots, w_K\}$ . For a center word  $c$  and an outside word  $o$ , the negative sampling loss function is given by:

$$\mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) = -\log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \log(\sigma(-\mathbf{u}_k^\top \mathbf{v}_c)) \quad (5)$$

for a sample  $w_1, w_2, \dots, w_K$ , where  $\sigma(\cdot)$  is the sigmoid function.

Please repeat parts (b) and (c), computing the partial derivatives of  $\mathbf{J}_{\text{neg-sample}}$  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.

**Solution:**

(i) the partial derivatives of  $\mathbf{J}_{\text{neg-sample}}$  with respect to  $\mathbf{v}_c$ :

$$\begin{aligned}
\frac{\partial J}{\partial v_c} &= -\frac{\partial \log(\sigma(u_o^T v_c))}{\partial v_c} - \frac{\partial \sum_{k=1}^K \log(\sigma(-u_k^T v_c))}{\partial v_c} \\
&= -\frac{1}{\sigma(u_o^T v_c)} \sigma(u_o^T v_c)(1 - \sigma(u_o^T v_c))u_o + \sum_{k=1}^K \frac{1}{\sigma(-u_k^T v_c)} \sigma(-u_k^T v_c)(1 - \sigma(-u_k^T v_c))u_k \\
&= -(1 - \sigma(u_o^T v_c))u_o + \sum_{k=1}^K (1 - \sigma(-u_k^T v_c))u_k \\
&= (\sigma(u_o^T v_c) - 1)u_o + \sum_{k=1}^K (1 - \sigma(-u_k^T v_c))u_k
\end{aligned}$$

the partial derivatives of  $J_{\text{neg-sample}}$  with respect to  $u_o$  :

$$\begin{aligned}
\frac{\partial J}{\partial u_o} &= -\frac{\partial \log(\sigma(u_o^T v_c))}{\partial u_o} - \frac{\partial \sum_{k=1}^K \log(\sigma(-u_k^T v_c))}{\partial u_o} \\
&= -\frac{1}{\sigma(u_o^T v_c)} \sigma(u_o^T v_c)(1 - \sigma(u_o^T v_c))v_c \\
&= (\sigma(u_o^T v_c) - 1)v_c
\end{aligned}$$

the partial derivatives of  $J_{\text{neg-sample}}$  with respect to  $u_k$  :

$$\begin{aligned}
\frac{\partial J}{\partial u_k} &= -\frac{\partial \log(\sigma(u_o^T v_c))}{\partial u_k} - \frac{\partial \sum_{k=1}^K \log(\sigma(-u_k^T v_c))}{\partial u_k} \\
&= \sum_{k=1}^K \frac{1}{\sigma(-u_k^T v_c)} \sigma(-u_k^T v_c)(1 - \sigma(-u_k^T v_c))v_c \\
&= \sum_{k=1}^K (1 - \sigma(-u_k^T v_c))v_c
\end{aligned}$$

(ii) Because this loss function iterate through  $K$  negative samples instead of iterating through all words in the corpus when calculate gradient.

(g) (2 point) 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  $i \neq j$ . 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:

$$\mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) = -\log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \log(\sigma(-\mathbf{u}_k^\top \mathbf{v}_c)) \quad (6)$$

for a sample  $w_1, w_2, \dots, w_K$ , where  $\sigma(\cdot)$  is the sigmoid function.

Compute the partial derivative of  $J_{\text{neg-sample}}$  with respect to a **negative sample**  $u_k$ . Please write your answers in terms of the vectors  $v_c$  and  $u_k$ , 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  $u_k$  and a sum over all sampled words not equal to  $u_k$ .

**Solution:**

$$\begin{aligned}
\frac{\partial J}{\partial u_k} &= -\frac{\partial \log(\sigma(u_o^T v_c))}{\partial u_k} - \frac{\partial \sum_{k=1}^K \log(\sigma(-u_k^T v_c))}{\partial u_k} \\
&= \sum_{k'=k} \frac{1}{\sigma(-u_{k'}^T v_c)} \sigma(-u_{k'}^T v_c)(1 - \sigma(-u_{k'}^T v_c))v_c \\
&= \sum_{k'=k} (1 - \sigma(-u_{k'}^T v_c))v_c
\end{aligned}$$

(h) (3 points) 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:

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

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  $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$  or  $\mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, w_{t+j}, \mathbf{U})$ , depending on your implementation.

Write down three partial derivatives

$$(i) \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{U}$$

$$(ii) \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_c$$

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

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

**Once you're done :** Given that you computed the derivatives of  $J(\mathbf{v}_c, w_{t+j}, \mathbf{U})$  with respect to all the model parameters  $\mathbf{U}$  and  $\mathbf{V}$  in parts (a) to (c), you have now computed the derivatives of the full loss function  $J_{\text{skip-gram}}$  with respect to all parameters. You’ re ready to implement word2vec!

**Solution:**

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

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

$$(iii) \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_w = 0$$