Assignment 2 [written]

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.

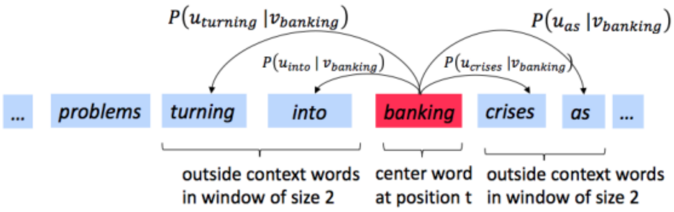

Figure 1: The word2vec skip-gram prediction model with window size 2

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

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

Here, $\boldsymbol{u}_o$ is the 'outside' vector representing outside word o, and $\boldsymbol{v}_c$ is the '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 $\boldsymbol{u}_w$. The columns of $\boldsymbol{V}$ are all of the 'center' vectors $\boldsymbol{v}_w$. Both $\boldsymbol{U}$ and $\boldsymbol{V}$ contain a vector for every $w \in$ Vocabulary.[1]

Recall from lectures that, for a single pair of words c and o, the loss is given by:

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

Another way to view this loss is as 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 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 $\boldsymbol{y}$ 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 equation (1).

Variables notation

$\boldsymbol{U}$, matrix of shape (vocab_size, embedding_dim) , all the 'outside' vectors.

$\boldsymbol{V}$, matrix of shape (vocab_size, embedding_dim) , all the 'center' vectors.

$\boldsymbol{y}$, vector of shape (vocab_size, 1), the true empirical distribution $\boldsymbol{y}$ is a one-hot vector with 1 for the true outside word o, and 0 for the others.

$\hat{\boldsymbol{y}}$, vector of shape (vocab_size, 1), the predicted distribution $\hat{\boldsymbol{y}}$ is the probability distribution $P(O \mid C = c)$ given by our model .

Formula to be used

$$\frac{\partial x^\top}{\partial x} = I$$
$$\frac{\partial A x^\top}{\partial x} = A^\top$$

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

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

Your answer should be one line.

$$y_w = \begin{cases} 1, & w = o \\ 0, & w \neq o \end{cases}$$

$$-\sum_{w=1}^{V} y_w log(\hat{y_w}) = -y_o log(\hat{y_o}) = -log(\hat{y_o})$$

(b) (5 points) Compute the partial derivative of $\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to $\boldsymbol{v}_c$. Please write your answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{U}$.

$$\begin{aligned}
&\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{v}_c} \\
&= -\frac{\partial log(P(O=o|C=c))}{\partial \boldsymbol{v}_c} \\
&= -\frac{\partial log(exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c))}{\partial \boldsymbol{v}_c} + \frac{\partial log(\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c))}{\partial \boldsymbol{v}_c} \\
&= -\boldsymbol{u}_0 + \sum_{w=1}^{V} \frac{exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \boldsymbol{u}_w \\
&= -\boldsymbol{u}_0 + \sum_{w=1}^{V} P(O=w|C=c) \boldsymbol{u}_w \\
&= \boldsymbol{U}^\top (\hat{\boldsymbol{y}} - \boldsymbol{y})
\end{aligned}$$

(c) (5 points) Compute the partial derivatives of $\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to each of the 'outside' word vectors, $\boldsymbol{u}_w$'s. There will be two cases: when $w = o$, the true 'outside' word vector, and $w \neq o$, for all other words. Please write you answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{v}_c$.

$$\begin{aligned}
&\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_w} \\
&= -\frac{\partial log(exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c))}{\partial \boldsymbol{u}_w} + \frac{\partial log(\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c))}{\partial \boldsymbol{u}_w} \\
w = o: & \\
origin &= -\boldsymbol{v}_c + \frac{1}{\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \frac{\partial \sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\partial \boldsymbol{u}_o} \\
&= -\boldsymbol{v}_c + \frac{exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \frac{\partial (\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\partial \boldsymbol{u}_o} \\
&= -\boldsymbol{v}_c + P(O=o|C=c)\boldsymbol{v}_c \\
&= (P(O=o|C=c) - 1)\boldsymbol{v}_c \\
w \neq o: & \\
origin &= \frac{exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\sum_{w=1}^{V} exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \frac{\partial (\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\partial \boldsymbol{u}_w} \\
&= P(O=o|C=c)\boldsymbol{v}_c \\
in\ summary: & \\
&\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_w} \\
&= (\hat{\boldsymbol{y}} - \boldsymbol{y})^\top \times \boldsymbol{v}_c
\end{aligned}$$

(d) (3 Points) The sigmoid function is given by Equation 4:

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

Please compute the derivative of $\boldsymbol{\sigma}(x)$ with respect to $\boldsymbol{x}$, where $\boldsymbol{x}$ is a vector.

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

(e) (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 $w_1, w_2, \ldots, w_K$ and their outside vectors as $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$. Note that $o \notin \{w_1, \ldots, w_K\}$. For a center word c and an outside word o, the negative sampling loss function is given by:

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

for a sample $w_1, \ldots w_K$, where $\sigma(\cdot)$ is the sigmoid function.[3]

Please repeat parts (b) and (c), computing the partial derivatives of $\boldsymbol{J}_{\text{neg-sample}}$ with respect to $\boldsymbol{v}_c$, with respect to $\boldsymbol{u}_o$, and with respect to a negative sample $\boldsymbol{u}_k$. Please write your answers in terms of the vectors $\boldsymbol{u}_o$, $\boldsymbol{v}_c$, and $\boldsymbol{u}_k$, 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 (d) to help compute the necessary gradients here.

$$\begin{aligned}
&\frac{\partial J_{neg-sample}(\boldsymbol{v}_c, o, U)}{\partial \boldsymbol{v}_c} \\
&= \frac{\partial(-log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{k=1}^{K} log(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))}{\partial \boldsymbol{v}_c} \\
&= -\frac{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))}{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)} \frac{\partial(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\partial \boldsymbol{v}_c} - \sum_{k=1}^{K} \frac{\partial log(\sigma(\boldsymbol{u}_k^\top \boldsymbol{v}_c))}{\partial \boldsymbol{v}_c} \\
&= (\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1)\boldsymbol{u}_o + \sum_{k=1}^{K}(1-\sigma(\boldsymbol{u}_k^\top \boldsymbol{v}_c))\boldsymbol{u}_k \\
&\frac{\partial J_{neg-sample}(\boldsymbol{v}_c, o, U)}{\partial \boldsymbol{u}_o} = (\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1)\boldsymbol{v}_c \\
&\frac{\partial J_{neg-sample}(\boldsymbol{v}_c, o, U)}{\partial \boldsymbol{u}_k} = (1-\sigma(\boldsymbol{u}_k^\top \boldsymbol{v}_c))\boldsymbol{v}_c
\end{aligned}$$

(f) (3 points) Suppose the center word is $c = w_t$ and the context window is $[w_{t-m}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots, w_{t+m}]$, where m is the context window size. Recall that for the skip-gram version of `word2vec`, the total loss for the context window is:

$$\boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U}) = \sum_{\substack{-m \le j \le m \\ j \ne 0}} \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U}) \tag{6}$$

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

Write down three partial derivatives:

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

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

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

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

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

$$(i)\frac{\partial J_{skip-gram}(\boldsymbol{v}_c, w_{t-m}, \ldots, w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{U}} = \sum_{-m\leq j\leq m, j\neq 0} \frac{\partial J(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U})}{\partial \boldsymbol{U}}$$

$$(ii)\frac{\partial J_{skip-gram}(\boldsymbol{v}_c, w_{t-m}, \ldots, w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{v}_c} = \sum_{-m\leq j\leq m, j\neq 0} \frac{\partial J(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U})}{\partial \boldsymbol{v}_c}$$

$$(iii)\frac{\partial J_{skip-gram}(\boldsymbol{v}_c, w_{t-m}, \ldots, w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{v}_w} = 0$$

(b)(c) another solution

$$
\begin{aligned}
&forward\ calculation:\\
&\quad x_o = u_o^\top v_c\\
&\quad t_o = exp(x_o)\\
&\quad s_o = \sum_{w\in vocab} exp(x_w)\\
&\quad \hat{y_o} = \frac{t_o}{s_o}\\
&\quad J = -log(\hat{y_o})\\
&backward\ propagation:\\
&\quad \frac{\partial t_o}{\partial x_o} = exp(x_o)\\
&\quad \frac{\partial s_o}{\partial x_o} = exp(x_o)\\
&\quad \frac{\partial \hat{y_o}}{\partial x_o} = \frac{exp(x_o)s_o - exp^2(x_o)}{s_o^2} = \hat{y_o}(1-\hat{y_o})\\
&\quad \frac{\partial \hat{y_o}}{\partial x_w} = \frac{-exp(x_o)exp(x_w)}{s_o^2} = -\hat{y_o}\hat{y_w}\\
&\quad \frac{\partial J}{\partial x_o} = \frac{\partial J}{\partial \hat{y_o}}\frac{\partial \hat{y_o}}{\partial x_o} = -\frac{1}{\hat{y_o}}\hat{y_o}(1-\hat{y_o}) = \hat{y_o} - 1\\
&\quad \frac{\partial J}{\partial x_w} = \frac{\partial J}{\partial \hat{y_o}}\frac{\partial \hat{y_o}}{\partial x_w} = -\frac{1}{\hat{y_o}}(-\hat{y_o}\hat{y_w}) = \hat{y_w}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{v}_c} &= \begin{bmatrix} \frac{\partial J}{\partial x_1}\frac{\partial x_1}{\partial v_c} \\ \cdots \\ \frac{\partial J}{\partial x_o}\frac{\partial x_o}{\partial v_c} \\ \cdots \\ \frac{\partial J}{\partial x_n}\frac{\partial x_n}{\partial v_c} \end{bmatrix} = \begin{bmatrix} \hat{y_1}u_1 \\ \cdots \\ (\hat{y_o}-1)u_o \\ \cdots \\ \hat{y_n}u_n \end{bmatrix} = \boldsymbol{U}^\top(\hat{\boldsymbol{y}} - \boldsymbol{y})\\
\frac{\partial J}{\partial \boldsymbol{u}_w} &= \begin{bmatrix} \frac{\partial J}{\partial x_1}\frac{\partial x_1}{\partial u_1} & \cdots & \frac{\partial J}{\partial x_o}\frac{\partial x_o}{\partial u_o} & \cdots & \frac{\partial J}{\partial x_n}\frac{\partial x_n}{\partial u_n} \end{bmatrix}\\
&= \begin{bmatrix} \hat{y_1}\boldsymbol{v}_c & \ldots & (\hat{y_o}-1)\boldsymbol{v}_c & \ldots & \hat{y_n}\boldsymbol{v}_c \end{bmatrix}\\
&= (\hat{\boldsymbol{y}} - \boldsymbol{y})^\top \times \boldsymbol{v}_c
\end{aligned}
$$