1 Written: Understanding word2vec (26 points)

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.

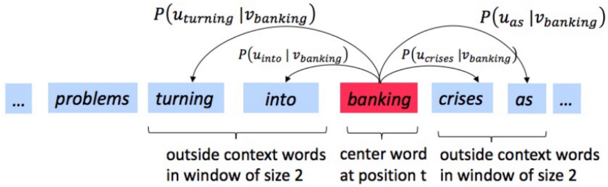

Figure 1: The word2vec skip-gram prediction model with window size 2

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

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

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:

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

We can view this loss 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).

[1] Assume that every word in our vocabulary is matched to an integer number k. Bolded lowercase letters represent vectors. $\boldsymbol{u}_k$ is both the k^{th} column of $\boldsymbol{U}$ and the '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 distribution q is $-\sum_i p_i \log(q_i)$.

(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 : one-hot vector
$\hat{y}$: $P(O|C=c)$ (prob.)

sol) y is a one-hot vector with a 1 for the true outside word o,

$$\therefore -\sum_{w=1}^{V} y_w \log(\hat{y}_w) = -\left(y_1 \log(\hat{y}_1) + \cdots + y_o \log(\hat{y}_o) + \cdots + y_V \log(\hat{y}_V)\right)$$
$$= -y_o \log(\hat{y}_o) = -\log(\hat{y}_o)$$

(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 $\boldsymbol{U}$. Note that in this course, we expect your final answers to follow the shape convention.[3] 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 $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{U}$ in your final answer (such as $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\dots$).

$$\frac{\partial J(v_c, o, U)}{\partial v_c} = \frac{\partial}{\partial v_c}\left(-\log\left(\frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}\right)\right)$$

$$= -\frac{\partial}{\partial v_c} \log(\exp(u_o^T v_c)) + \frac{\partial}{\partial v_c} \log\left(\sum_{w=1}^{V} \exp(u_w^T v_c)\right)$$

① : $\frac{\partial}{\partial v_c} \log(\exp(u_o^T v_c)) = \frac{\partial}{\partial v_c}(u_o^T v_c) = u_o = Uy$

② : $\frac{\partial}{\partial v_c} \log\left(\sum_{w=1}^{V} \exp(u_w^T v_c)\right) = \frac{1}{\sum_{w=1}^{V} \exp(u_w^T v_c)} \times \sum_{x=1}^{V} \exp(u_x^T v_c) \times u_x$

$$= \sum_{x=1}^{V} \frac{\exp(u_x^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)} u_x = \sum_{x=1}^{V} P(x|c)\, u_x = \sum_{x=1}^{V} \hat{y}_x u_x = U\hat{y}$$

$$\therefore \frac{\partial J(v_c, o, U)}{\partial v_c} = -① + ② = -Uy + U\hat{y} = U(\hat{y} - y)$$

(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 your answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{v}_c$. In this subpart, you may use specific elements within these terms as well, such as $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots)$.

i)
$$\frac{\partial J}{\partial u_w} = \frac{\partial}{\partial u_w}\left(-\log\left(\frac{\exp(u_o^T v_c)}{\sum_{x=1}^{V}\exp(u_x^T v_c)}\right)\right)$$

$$= -\frac{\partial}{\partial u_w}\log(\exp(u_o^T v_c)) + \frac{\partial}{\partial u_w}\log\left(\sum_{x=1}^{V}\exp(u_x^T v_c)\right)$$

$$= -\frac{\partial}{\partial u_w}(u_o^T v_c) + \frac{1}{\sum_{x=1}^{V}\exp(u_x^T v_c)}\frac{\partial}{\partial u_w}\left(\sum_{x=1}^{V}\exp(u_x^T v_c)\right)$$

ii) if) $w = o$

$$\frac{\partial J}{\partial u_o} = -v_c + \frac{1}{\sum_{w=1}^{V}\exp(u_w^T v_c)} \times \exp(u_o^T v_c) \times v_c$$

$$= -v_c + \hat{y}_o v_c = \underline{v_c(\hat{y}_o - 1)}$$

iii) if) $w \neq o$

$$\frac{\partial J}{\partial u_w} = 0 + \frac{\exp(u_w^T v_c)\, v_c}{\sum_{x=1}^{V}\exp(u_w^T v_c)} = \underline{\hat{y}_w v_c}$$

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

$$\frac{\partial J}{\partial U} = \frac{\partial J}{\partial u_1} + \frac{\partial J}{\partial u_2} + \cdots + \frac{\partial J}{\partial u_{|Vocab|}}$$

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

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

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 $\sigma(x)$.

$$\frac{d}{dx}\sigma(x) = \frac{d}{dx}(1+e^{-x})^{-1} = -1 \times (1+e^{-x})^{-2} \times -e^{-x}$$

$$= \sigma(x)\left(\frac{e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(1 - \frac{1}{1+e^{-x}}\right)$$

$$= \underline{\sigma(x)(1-\sigma(x))}$$

(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 $\boldsymbol{u}_1, \dots, \boldsymbol{u}_K$. 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:

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

for a sample $w_1, \dots w_K$, where $\sigma(\cdot)$ is the sigmoid function.[4]

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 (e) to help compute the necessary gradients here.

(i) $\dfrac{\partial J_{neg\_sample}}{\partial v_c}$,

$$\frac{\partial J}{\partial v_c} = \frac{\partial}{\partial v_c}\left(-\log(\sigma(u_o^T v_c))\right) - \frac{\partial}{\partial v_c}\left(\sum_{k=1}^{K} \log(\sigma(-u_k^T v_c))\right)$$

$$= -\frac{\sigma(u_o^T v_c)(1-\sigma(u_o^T v_c))}{\sigma(u_o^T v_c)} u_o - \sum_{k=1}^{K} \frac{\sigma(-u_k^T v_c)(1-\sigma(-u_k^T v_c))}{\sigma(-u_k^T v_c)} \times (-u_k)$$

$$= -u_o(1-\sigma(u_o^T v_c)) + \sum_{k=1}^{K} u_k(1-\sigma(-u_k^T v_c))$$

(ii) $\dfrac{\partial J}{\partial u_o}$

$$= \frac{\partial}{\partial u_o}\left(-\log(\sigma(u_o^T v_c))\right) - \frac{\partial}{\partial u_o}\left(\sum_{k=1}^{K} \log(\sigma(-u_k^T v_c))\right)$$

$$= -\frac{\sigma(u_o^T v_c)(1-\sigma(u_o^T v_c))}{\sigma(u_o^T v_c)} v_c - 0 \ (\because o \notin \{w_1, w_2, \dots w_K\})$$

$$= -v_c(1-\sigma(u_o^T v_c))$$

(iii) $\dfrac{\partial J}{\partial u_k}$

$$= \frac{\partial}{\partial u_k}\left(-\log(\sigma(u_o^T v_c))\right) - \frac{\partial}{\partial u_k}\left(\sum_{x=1}^{K} \log(\sigma(-u_x^T v_c))\right)$$

$$= -\sum_{x=1}^{K} \frac{\sigma(-u_x^T v_c)(1-\sigma(-u_x^T v_c))}{\sigma(-u_x^T v_c)} \frac{\partial(-u_x^T v_c)}{\partial u_k} = \sum_{x=1}^{K} (1-\sigma(-u_x^T v_c)) \frac{\partial(u_x^T v_c)}{\partial u_k}$$

$$= \underbrace{(1-\sigma(-u_k^T v_c))v_c}$$

(iv) why negative sampling is more efficient to compute than the naive-softmax loss

→ naive-softmax loss는 softmax를 구하기 위해 전체 corpus 에서 $u_w^T v_c$를 연산해줘야 하기에 연산 비용이 많다, 하지만 negative sampling은 K개의 negative sample과 outside word 1개의 연산만 진행하기에 연산 비용이 적다.

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

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

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

Compute the partial derivative of $\boldsymbol{J}_{\text{neg-sample}}$ with respect to a negative sample $\boldsymbol{u}_k$. Please write your answers in terms of the vectors $\boldsymbol{v}_c$ and $\boldsymbol{u}_k$, 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.

sol) $\frac{\partial}{\partial u_k} J_{\text{neg-sample}}$

$$= \frac{\partial}{\partial u_k}\left(-\log(\sigma(u_o^T v_c))\right) - \frac{\partial}{\partial u_k}\left(\sum_{x=1}^{k} \log(\sigma(-u_x^T v_c))\right)$$

$$= 0 - \frac{\partial}{\partial u_k}\left\{ n \log(\sigma(-u_k^T v_c)) + \sum_{x=1}^{m} \log(\sigma(-u_x^T v_c)) \right\}$$

이때 n은 $\{w_1, w_2, \ldots, w_k\}$ 중 u_k와 같은 값을 가지는 개수, m은 $\{w_1, w_2, \ldots w_k\}$을 u_k와 같은 값을 제거한 집합의 개수 $(n+m=K)$

$$= -n \frac{\sigma(-u_k^T v_c)(1-\sigma(-u_k^T v_c))}{\sigma(-u_k^T v_c)} \cdot -v_c = \underline{n v_c (1-\sigma(-u_k^T v_c))}$$

(h) (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:

$$\boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U}) = \sum_{\substack{-m \le j \le m \\ j \ne 0}} \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U}) \tag{7}$$

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 $w \ne c$

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) $\dfrac{\partial J_{s\text{-}g}(v_c, w_{t-m}, \ldots, w_{t+m}, U)}{\partial U} = \dfrac{\partial}{\partial U} \left\{ \sum_{\substack{-m \le j \le m \\ j \ne 0}} J(v_c, w_{t+j}, U) \right\}$

(ii) $\dfrac{\partial J_{s\text{-}g}}{\partial v_c} = \dfrac{\partial}{\partial v_c} \left\{ \sum_{\substack{-m \le j \le m \\ j \ne 0}} J(v_c, w_{t+j}, U) \right\}$

(iii) $\dfrac{\partial J_{s\text{-}g}}{\partial v_w} = \dfrac{\partial}{\partial v_w} \left\{ \sum_{\substack{-m \le j \le m \\ j \ne 0}} J(v_c, w_{t+j}, U) \right\}$, $w \ne c$