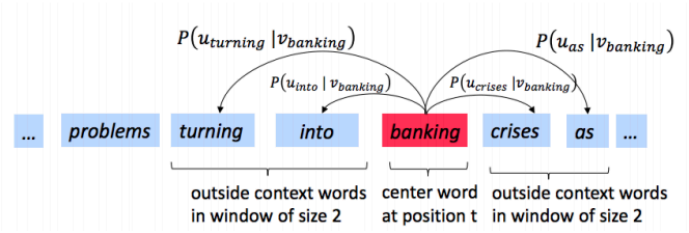
CS 224n Assignment #2: word2vec (43 Points)

1 Written: Understanding word2vec (23 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.



V: input vector
U: output vector
o: context word의 index
c: center word의 index
w: 단어의 vocabulary

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:

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

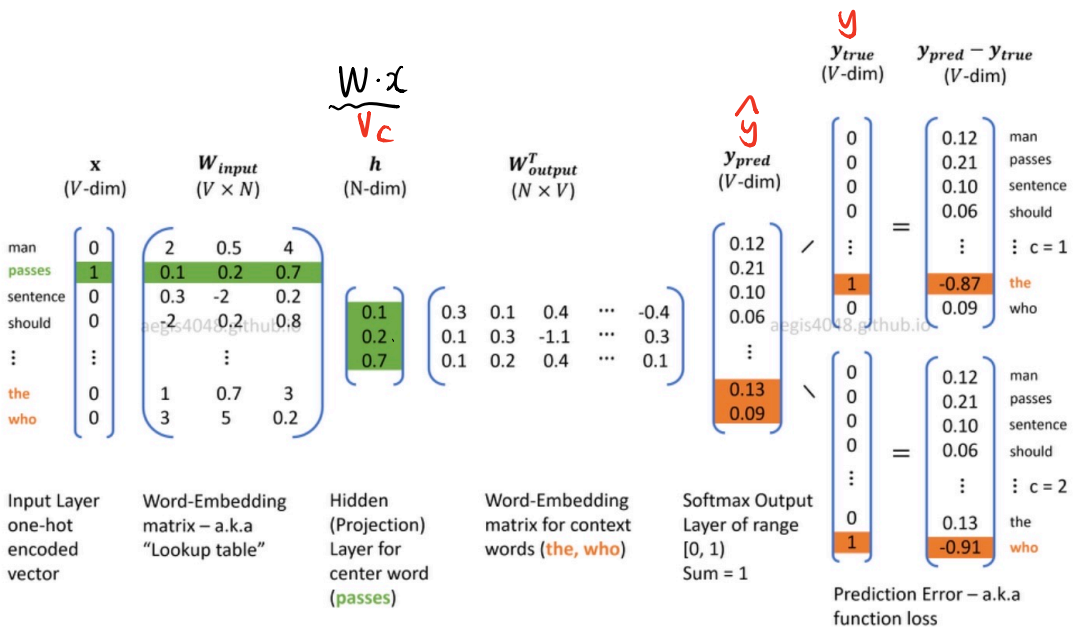
Here, $\boldsymbol{u}_o$ is the 'outside' vector representing outside word o, and $\boldsymbol{v}_c$ is the 'center' vector representing center word c. To contain these parameters, we have two matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$. The columns of $\boldsymbol{U}$ are all the 'outside' vectors $\boldsymbol{u}_w$. 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).

(one-hot vector)
실제 outside word인 index만 1, 나머지는 0


$W \cdot x$
v_c
$\hat{y}$
y
$\mathbf{x}$ (V-dim)
W_{input} ($V \times N$)
h (N-dim)
W^T_{output} ($N \times V$)
y_{pred} (V-dim)
y_{true} (V-dim)
$y_{pred} - y_{true}$ (V-dim)
man
passes
sentence
should
the
who
c = 1
c = 2
Input Layer one-hot encoded vector
Word-Embedding matrix – a.k.a "Lookup table"
Hidden (Projection) Layer for center word (passes)
Word-Embedding matrix for context words (the, who)
Softmax Output Layer of range [0, 1) Sum = 1
Prediction Error – a.k.a function loss

(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 : 실제 확률분포 , $\hat{y}$: 모델에서 구한 확률분포 (둘다 단어개수와 동일한 길이)

- V: input vector
- U: output vector
- o: outside word의 index
- c: center word의 index
- w: 단어의 w번째 단어

$$\hat{y}_o = P(u_o|v_c) = \frac{e^{u_o^T v_c}}{\sum_{w \in vocab} e^{u_w^T v_c}}$$

y는 one-hot vector로 실제 outside vector인 o 에서만 1이고, 나머지는 0

$$-\sum_{w \in Vocab} y_w \log(\hat{y}_w) = -(\underbrace{y_1}_{0}\log(\hat{y}_1) + \underbrace{y_2}_{0}\log(\hat{y}_2) + \cdots + \underbrace{y_o}_{1}\log(\hat{y}_o) + \cdots + \underbrace{y_{|V|}}_{0}\log(\hat{y}_{|V|}))$$

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

(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.

$$\frac{\partial J_{naive-softmax}}{\partial v_c} = \frac{\partial}{\partial v_c}[-\log(\hat{y}_o)]$$

$$= \frac{\partial}{\partial v_c}\left[-\log\left(\frac{e^{u_o^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}}\right)\right]$$

$$= -\frac{\partial}{\partial v_c}\left[\log(e^{u_o^T v_c}) - \log(\sum_{w \in Vocab} e^{u_w^T v_c})\right]$$

$$= -\frac{\partial}{\partial v_c}\left[u_o^T v_c\right] + \frac{\partial}{\partial v_c}\left[\log(\sum_{w \in Vocab} e^{u_w^T v_c})\right]$$

Softmax에서 log 취한건 log랑 e 때려 ...

$a^T \cdot x$ 는 (a: vector) x에 대해 미분해서 a

$$= -(u_o) + \frac{\sum_{x \in Vocab} u_x \cdot e^{u_x^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}}$$

인덱스는 다르게

$$= -u_o + \sum_{x \in Vocab} \frac{e^{u_w^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}} \cdot u_x$$

$$= -u_o + \sum_{x \in Vocab} P(u_x \mid v_c) \cdot u_x$$

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

$\hat{y}_o = P(u_o \mid v_c)$

$$= -u_o + \sum_{x \in Vocab} \hat{y}_x \cdot u_x \Rightarrow (y, \hat{y}, U)\text{로만 표현됨}$$

(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: ① when $w = o$, the true 'outside' word vector, and ② $w \neq o$, for all other words. Please write you answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{v}_c$.

① W번째 단어가 실제 outside vector 인 경우

② W번째 단어가 실제 outside vector가 아닐 경우

① $\frac{\partial J_{\text{naive-softmax}}}{\partial u_{w=o}} = \frac{\partial}{\partial u_{w=o}} [-\log(\hat{y}_o)]$

$$= \frac{\partial}{\partial u_{w=o}} \left[-\log \left(\frac{e^{u_o^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}} \right) \right]$$

$$= -\frac{\partial}{\partial u_{w=o}} [u_o^T v_c] + \frac{\partial}{\partial u_{w=o}} \left[\log \left(\sum_{w \in Vocab} e^{u_w^T v_c} \right) \right]$$

$$= -v_c + \left(\frac{\frac{\partial}{\partial u_{w=o}} \left(\sum_{w \in Vocab} e^{u_w^T v_c} \right) \cdot v_c}{\sum_{w \in Vocab} e^{u_w^T v_c}} \right)$$

$e^{u_o^T v_c}$ (미분한 뒤 w=o 인것만 남는데까지)

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

② $\frac{\partial J_{\text{naive-softmax}}}{\partial u_{w \neq o}} = \frac{\partial}{\partial u_{w \neq o}} [-\log(\hat{y}_o)]$

$$= \frac{\partial}{\partial u_{w \neq o}} \left[-\log \left(\frac{e^{u_o^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}} \right) \right]$$

$$= -\frac{\partial}{\partial u_{w \neq o}} [u_o^T v_c] + \frac{\partial}{\partial u_{w \neq o}} \left[\log \left(\sum_{w \in Vocab} e^{u_w^T v_c} \right) \right]$$

$$= 0 + \frac{e^{u_{w \neq o}^T v_c} \, v_c}{\sum_{w \in Vocab} e^{u_w^T v_c}}$$

$\hat{y}_{w \neq o}$

$$= v_c \cdot \hat{y}_{w \neq o}$$

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

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

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

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

$$= \left[-(1+e^{-x})^{-2}\right]\left[-e^{-x}\right]$$

$$= \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{(1+e^{-x})}\left(\frac{e^{-x}}{1+e^{-x}}\right)$$

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

$\sigma(x)$ $\sigma(x)$

$$= \sigma(x)(1-\sigma(x))$$

(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.

CASE 1 : Center word에 대해서 미분

$$\frac{\partial J_{\text{neg-sample}}}{\partial v_c} = \frac{\partial}{\partial v_c}\left[-\log(\sigma(u_o^\top v_c)) - \sum_{k=1}^{K}\log(\sigma(-u_k^\top v_c))\right]$$

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

위의 sigmoid 미분 이용

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

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

CASE 2 : outside word에 대해서 미분

$$\frac{\partial J_{\text{negative sample}}}{\partial u_o} = \frac{\partial}{\partial u_o}\left[-\log(\sigma(u_o^\top v_c)) - \sum_{k=1}^{K}\log(\sigma(-u_k^\top v_c))\right]$$

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

u_o가 없으므로 0이다

$$= \frac{\partial}{\partial u_o}\left[-\log(\sigma(u_o^\top v_c))\right]$$

$$= -\left[\frac{V_c \cdot \sigma(u_o^T v_c)(1-\sigma(u_o^T v_c))}{\sigma(u_o^T v_c)}\right] = -v_c(1-\sigma(u_o^T v_c))$$

CASE 3 : K negative samples의 경우 만

$$\frac{\partial J_{neg\text{-}sample}}{\partial u_k} = \frac{\partial}{\partial u_k}\left[-\log(\sigma(u_o^T v_c)) - \sum_{k=1}^{K}\log(\sigma(-u_k^T v_c))\right]$$

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

$0 =$

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

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

(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:

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

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 $\boldsymbol{J}_{\text{skip-gram}}$ with respect to all parameters. You're ready to implement `word2vec`!*

① $\partial J_{\text{skip-gram}}(v_c, w_{t-m}, \cdots, w_{t+m}, U)/\partial U$

$$= \sum_{-m \le j \le m,\ j \neq 0} \partial J_{\text{skip-gram}}(v_c, w_{t+j}, U)/\partial U$$

② $\partial J_{\text{skip-gram}}(v_c, w_{t-m}, \cdots, w_{t+m}, U)/\partial v_c$

$$= \sum_{-m \le j \le m,\ j \neq 0} \partial J_{\text{skip-gram}}(v_c, w_{t+j}, U)/\partial v_c$$

③ $\partial J_{\text{skip-gram}}(v_c, w_{t-m}, \cdots, w_{t+m}, U)/\partial v_w \quad (w \neq c)$

$$= 0$$