Tobig's 13기 [illegible]

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.

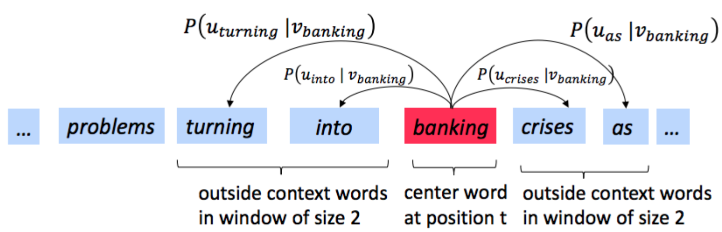

Figure 1: The word2vec skip-gram prediction model with window size 2

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]

$\underset{\text{outside}}{U} =$ [ ] u_w $\quad$ $\underset{\text{center}}{V} =$ [ ] v_w

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).

(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

[1] Assume that every word in our vocabulary is matched to an integer number k. $\boldsymbol{u}_k$ 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 $-\sum_i p_i \log(q_i)$.

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

Your answer should be one line.

(a) $-\sum_{w \in Vocab} y_w \log(\hat{y}_w) = -y_1 \log(\hat{y}_1) - y_2 \log(\hat{y}_2) - \cdots - y_o \log(\hat{y}_o) - \cdots - y_w \log(\hat{y}_w)$

$= 0 \cdot \log\sim - 0 \cdot \log\sim \cdots - 1 \cdot \log(\hat{y}_o) - \cdots - 0 \cdot \log\sim = -\log(\hat{y}_o)$

(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 $\boldsymbol{U}$.

(b) $J = -\log \dfrac{\exp(u_o^T v_c)}{\sum_{w \in Vocab} \exp(u_w^T v_c)}$

$\dfrac{\partial J}{\partial v_c} = \dfrac{\partial}{\partial v_c}\left(-\log \dfrac{\exp(u_o^T v_c)}{\sum_{w \in Vocab} \exp(u_w^T v_c)}\right) = \dfrac{\partial}{\partial v_c}\left(-u_o^T v_c + \log \sum \exp(u_w^T v_c)\right)$

$= -u_o^T + \dfrac{\exp(u_w^T v_c) \cdot u_w^T}{\sum \exp(u_w^T v_c)} = -u_o^T + P(w|c) \cdot u_w^T \qquad u_o = U \cdot y \rightarrow P(w|c) \cdot u_w^T = U \cdot \hat{y}^T$

$\therefore \dfrac{\partial J}{\partial v_c} = U^T(\hat{y} - y)^T$

(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 $\boldsymbol{v}_c$.

(c) $\dfrac{\partial J}{\partial u_o} = \dfrac{\partial}{\partial u_o}\left(-\log \dfrac{\exp(u_o^T v_c)}{\sum \exp(u_w^T v_c)}\right) = -v_c + \dfrac{\partial}{\partial u_o}\left(\log \sum \exp(u_w^T \cdot v_c)\right)$

$= -v_c + \dfrac{\exp(u_w^T \cdot v_c) \cdot v_c}{\sum \exp(u_w^T \cdot v_c)} = -v_c + P(w|c) \cdot v_c$

$\therefore \dfrac{\partial J}{\partial u_o} = \begin{cases} (\hat{y} - y) v_c & (w = o) \\ \hat{y}_w \cdot v_c & (w \neq o) \end{cases}$

(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.

$\dfrac{\partial \sigma(x)}{\partial x} = \dfrac{\partial}{\partial x}\left(\dfrac{1}{1+e^{-x}}\right) = \dfrac{-(-e^{-x})}{(1+e^{-x})^2} = \dfrac{1}{(1+e^{-x})} \cdot \dfrac{1+e^{-x}-1}{(1+e^{-x})} = \sigma(x)(1-\sigma(x))$

(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 $w_1, w_2, \dots, w_K$ and their outside vectors as $\boldsymbol{u}_1, \dots, \boldsymbol{u}_K$. 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.[3]

Please repeat parts (b) and (c), computing the partial derivatives of $\boldsymbol{J}_{\text{neg-sample}}$ i) with respect to $\boldsymbol{v}_c$, ii) with respect to $\boldsymbol{u}_o$, and iii) 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, iv) 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.

i) $$\frac{\partial J}{\partial V_c} = \frac{\partial}{\partial V_c}\left(-\log(\sigma(U_o^T \cdot V_c)) - \sum_{k=1}^{K} \log(\sigma(-U_k^T \cdot V_c))\right) = -\frac{\sigma(U_o^T \cdot V_c)(1-\sigma(U_o^T \cdot V_c)) \cdot U_o^T}{\sigma(U_o^T \cdot V_c)} + \sum_{k=1}^{K} \frac{\sigma(-U_k^T \cdot V_c)(1-\sigma(-U_k^T \cdot V_c)) \cdot U_k^T}{\sigma(-U_k^T \cdot V_c)}$$

$$= -(1-\sigma(U_o^T \cdot V_c)) \cdot U_o^T + \sum_{k=1}^{K}(1-\sigma(-U_k^T \cdot V_c)) \cdot U_k^T$$

ii) $$\frac{\partial J}{\partial U_o} = \frac{\partial}{\partial U_o}\left(-\log(\sigma(U_o^T \cdot V_c)) - \underbrace{\sum_{k=1}^{K} \log(\sigma(-U_k^T \cdot V_c))}_{U_o \text{ 포함 X} \rightarrow \text{ 연산에 영향 X}}\right) = -\frac{\sigma(U_o^T \cdot V_c)(1-\sigma(U_o^T \cdot V_c)) \cdot V_c}{\sigma(U_o^T \cdot V_c)} = -(1-\sigma(U_o^T \cdot V_c)) \cdot V_c$$

iii) $$\frac{\partial J}{\partial U_k} = \frac{\partial}{\partial U_k}\left(\underbrace{-\log(\sigma(U_o^T \cdot V_c))}_{U_k \text{ 포함 X} \rightarrow \text{ 연산에 영향 X}} - \sum_{k=1}^{K} \log(\sigma(-U_k^T \cdot V_c))\right) = \sum_{k=1}^{K} \frac{\sigma(-U_k^T \cdot V_c)(1-\sigma(-U_k^T \cdot V_c)) V_c}{\sigma(-U_k^T \cdot V_c)} = \sum_{k=1}^{K}(1-\sigma(-U_k^T \cdot V_c)) \cdot V_c$$

iv) Naive Softmax Loss에 비해 Negative Sampling Loss의 연산량이 더 적기 때문에 more efficient!

(f) (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:

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

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

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

(iii) $\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \dots w_{t+m}, \boldsymbol{U}) / \partial \boldsymbol{v}_w$ 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`!*

(i) $$\frac{\partial J}{\partial U} = \sum_{\substack{-m \le j \le m \\ j \ne 0}} \frac{\partial J(V_c, W_{t+j}, U)}{\partial U}$$

(ii) $$\frac{\partial J}{\partial V_c} = \sum_{\substack{-m \le j \le m \\ j \ne 0}} \frac{\partial J(V_c, W_{t+j}, U)}{\partial V_c}$$

(iii) center word가 아니면 파라미터 업데이트 되지 않음

$$\therefore \frac{\partial J}{\partial V_w} = 0 \quad (w \neq c)$$