CSE40201 - Natural Language Processing
Assignment 2
Understanding word2vec

Student name: Nguyen Minh Duc
Student ID: 20202026

1 Written Assignment

Problem 1. (6 points) Prove that the naive-softmax loss (Equation 2) is the same as the cross-entropy loss between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, i.e. (note that $\boldsymbol{y}, \hat{\boldsymbol{y}}$ are vectors and $\hat{\boldsymbol{y}}_o$ is a scalar):

$$-\sum_{w \in \text{Vocab}} \boldsymbol{y}_w \log\left(\hat{\boldsymbol{y}}_w\right) = -\log\left(\hat{\boldsymbol{y}}_o\right).$$

Solution.

$$-\sum_{w \in \text{Vocab}} \boldsymbol{y}_w \log\left(\hat{\boldsymbol{y}}_w\right) = -\boldsymbol{y}_o \log\left(\hat{\boldsymbol{y}}_o\right) - \sum_{\substack{w \in \text{Vocab} \\ w \neq o}} \boldsymbol{y}_w \log\left(\hat{\boldsymbol{y}}_w\right) = -1 \times \log\left(\hat{\boldsymbol{y}}_o\right) - 0 = -\log\left(\hat{\boldsymbol{y}}_o\right).$$

Problem 2. (10 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 $\boldsymbol{U}$. Additionally, answer the following two questions with one sentence each: (1) When is the gradient zero? (2) Why does subtracting this gradient, in the general case when it is nonzero, make $\boldsymbol{v}_c$ a more desirable vector (namely, a vector closer to outside word vectors in its window)?

Solution.

Note that

$$\begin{aligned}
\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) &= -\log\left(\frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}\right) \\
&= \log\left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)\right) - \log\left(\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)\right) \\
&= \log\left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)\right) - \boldsymbol{u}_o^\top \boldsymbol{v}_c. \qquad (1)
\end{aligned}$$

Taking derivatives of both sides with respect to $\boldsymbol{v}_c$, we obtain

$$
\begin{aligned}
\frac{\partial}{\partial \boldsymbol{v}_c} J_{\text{naive-softmax}} &= \frac{\partial}{\partial \boldsymbol{v}_c} \left(\log \left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \right) - \boldsymbol{u}_o^\top \boldsymbol{v}_c \right) \\
&= \frac{\partial}{\partial \boldsymbol{v}_c} \log \left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \right) - \frac{\partial}{\partial \boldsymbol{v}_c} \boldsymbol{u}_o^\top \boldsymbol{v}_c \\
&= \frac{\frac{\partial}{\partial \boldsymbol{v}_c} \left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \right)}{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} - \boldsymbol{u}_o \\
&= \frac{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \boldsymbol{u}_w}{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} - 1 \boldsymbol{u}_o \\
&= \left(\sum_{w \in \text{Vocab}} \frac{\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \boldsymbol{u}_w}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} \right) - \boldsymbol{y}_o \boldsymbol{u}_o - \boldsymbol{0} \\
&= \left(\sum_{w \in \text{Vocab}} P(O = w \mid C = c) \boldsymbol{u}_w \right) - \boldsymbol{y}_o \boldsymbol{u}_o - \sum_{\substack{w \in \text{Vocab} \\ w \neq o}} \boldsymbol{y}_w \boldsymbol{u}_w \\
&= \left(\sum_{w \in \text{Vocab}} \hat{\boldsymbol{y}}_w \boldsymbol{u}_w \right) - \left(\sum_{w \in \text{Vocab}} \boldsymbol{y}_w \boldsymbol{u}_w \right) \\
&= \sum_{w \in \text{Vocab}} (\hat{\boldsymbol{y}}_w - \boldsymbol{y}_w) \, \boldsymbol{u}_w
\end{aligned}
$$

This is a linear combination of vectors $\boldsymbol{u}_w$ in U scaled by the components of $\hat{\boldsymbol{y}} - \boldsymbol{y}$. Therefore, this can be rewritten as a form of matrix-vector multiplication, i.e.

$$
\frac{\partial}{\partial \boldsymbol{v}_c} J_{\text{naive-softmax}} = \boldsymbol{U} \left(\hat{\boldsymbol{y}} - \boldsymbol{y} \right).
$$

1. **Q**: When is the gradient zero?

   A: The gradient zero will become zero if and only if $\boldsymbol{U} \left(\hat{\boldsymbol{y}} - \boldsymbol{y} \right) = 0$, which means either $\hat{\boldsymbol{y}} = \boldsymbol{y}$ or non-trivial null space of $\boldsymbol{U}$ contains $\hat{\boldsymbol{y}} - \boldsymbol{y}$ (However, the latter case rarely happens).

2. **Q**: Why does subtracting this gradient, in the general case when it is nonzero, make $\boldsymbol{v}_c$ a more desirable vector (namely, a vector closer to outside word vectors in its window)?

   A: By subtracting this gradient, the function moves closer to the local minima, which makes $\boldsymbol{v}_c$ a more desirable vector as the predicted distribution becomes more similar to the true distribution.

Problem 3. (10 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 $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, ...). Note that $\boldsymbol{u}_w$ is a vector while $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ are scalars.

Solution.

Using the equation (1) proven previously, we have

$$
\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = \log \left(\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \right) - \boldsymbol{u}_o^\top \boldsymbol{v}_c.
$$

Taking derivatives of both sides with respect to $\boldsymbol{u}_w$, we obtain

$$\frac{\partial}{\partial \boldsymbol{u}_w} J_{\text{naive-softmax}} = \frac{\partial}{\partial \boldsymbol{u}_w} \log \left(\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c) \right) - \frac{\partial}{\partial \boldsymbol{u}_w} \boldsymbol{u}_o^\top \boldsymbol{v}_c$$

When $w = o$:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}_o} J_{\text{naive-softmax}} &= \frac{\partial}{\partial \boldsymbol{u}_o} \log \left(\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c) \right) - \frac{\partial}{\partial \boldsymbol{u}_o} \boldsymbol{u}_o^\top \boldsymbol{v}_c \\
&= \frac{\frac{\partial}{\partial \boldsymbol{u}_o} \left(\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c) \right)}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} - \boldsymbol{v}_c \\
&= \frac{\frac{\partial}{\partial \boldsymbol{u}_o} \left(\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c) \right)}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} - \boldsymbol{v}_c \\
&= \frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c) \boldsymbol{v}_c}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} - \boldsymbol{v}_c \\
&= \hat{\boldsymbol{y}}_o \boldsymbol{v}_c - \boldsymbol{v}_c \\
&= \left(\hat{\boldsymbol{y}}_o - 1 \right) \boldsymbol{v}_c
\end{aligned}$$

When $w \neq o$:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}_w} J_{\text{naive-softmax}} &= \frac{\partial}{\partial \boldsymbol{u}_w} \log \left(\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c) \right) - \frac{\partial}{\partial \boldsymbol{u}_w} \boldsymbol{u}_o^\top \boldsymbol{v}_c \\
&= \frac{\frac{\partial}{\partial \boldsymbol{u}_w} \left(\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c) \right)}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} - 0 \\
&= \frac{\frac{\partial}{\partial \boldsymbol{u}_w} \left(\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \right)}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} \\
&= \frac{\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c) \boldsymbol{v}_c}{\sum_{k \in \text{Vocab}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_c)} \\
&= \hat{\boldsymbol{y}}_w \boldsymbol{v}_c
\end{aligned}$$

Problem 4. (2 points) Write down the partial derivative of $\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to $\boldsymbol{U}$. Please break down your answer in terms of $\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_1}$, $\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_2}$, $\cdots$, $\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_{|\text{Vocab}|}}$. The solution should be one or two lines long.

Solution.

Using the previous result from problem 3, we can easily obtain the following:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{U}} J_{\text{naive-softmax}} &= \left[\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_1}, \frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_2}, \cdots, \frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_{|\text{Vocab}|}} \right] \\
&= \left[\hat{\boldsymbol{y}}_1 \boldsymbol{v}_c,\ \hat{\boldsymbol{y}}_2 \boldsymbol{v}_c,\ \ldots,\ \hat{\boldsymbol{y}}_{o-1} \boldsymbol{v}_c,\ \left(\hat{\boldsymbol{y}}_o - 1 \right) \boldsymbol{v}_c,\ \hat{\boldsymbol{y}}_{o+1} \boldsymbol{v}_c,\ \ldots,\ \hat{\boldsymbol{y}}_{|\text{Vocab}|} \boldsymbol{v}_c \right].
\end{aligned}$$

Problem 5. (4 points) The ReLU (Rectified Linear Unit) activation function is given by:

$$f(x) = \max(0, x).$$

Please compute the derivative of $f(x)$ with respect to x, where x is a scalar. You may ignore the case that the derivative is not defined at 0.

Solution.

Since the case when $x = 0$ is ignored, there are two cases:

When $x > 0$:

$$\begin{aligned} f(x) &= \max(0, x) \\ &= x \\ \implies \frac{d}{dx} f(x) &= \frac{d}{dx} x \\ &= 1. \end{aligned}$$

When $x < 0$:

$$\begin{aligned} f(x) &= \max(0, x) \\ &= 0 \\ \implies \frac{d}{dx} f(x) &= \frac{d}{dx} 0 \\ &= 0. \end{aligned}$$

Problem 6. (6 points) The sigmoid function is given by:

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

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.

Taking the natural log of both sides, we obtain

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

Taking the derivative with respect to x:

$$\begin{aligned} \frac{d}{dx} \ln \sigma(x) &= \frac{d}{dx} (x - \ln (e^x + 1)) \\ \frac{\sigma'(x)}{\sigma(x)} &= 1 - \frac{e^x}{e^x + 1} \\ &= 1 - \sigma(x) \\ \implies \sigma'(x) &= \sigma(x) (1 - \sigma(x)) \end{aligned}$$

Problem 7. (12 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 $\boldsymbol{u}_{w_1}, \boldsymbol{u}_{w_2}, \ldots, \boldsymbol{u}_{w_K}$. For this question, assume that the K negative samples are distinct. In other words, $i \neq j$ implies $w_i \neq w_j$ for $i, j \in \{1, \ldots, K\}$. 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:

$$\boldsymbol{J}_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \tag{2}$$

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

(i) Please repeat problems 2 and 3, 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 the s^{th} negative sample $\boldsymbol{u}_{w_s}$. Please write your answers in terms of the vectors $\boldsymbol{v}_c$, $\boldsymbol{u}_o$, and $\boldsymbol{u}_{w_s}$, where $s \in [1, K]$. **Note:** You should be able to use your solution to problem 6 to help compute the necessary gradients here.

(ii) In the lecture, we learned that an efficient implementation of backpropagation leverages the reuse of previously-computed partial derivatives. Which quantity could you reuse between the three partial derivatives to minimize duplicate computation? Write your answer in terms of $\boldsymbol{U}_{o,\{w_1,\ldots,w_K\}} = \left[\boldsymbol{u}_o, -\boldsymbol{u}_{w_1}, \ldots, -\boldsymbol{u}_{w_K}\right]$, a matrix with the outside vectors stacked as columns, and $\mathbf{1}$, a $(K+1) \times 1$ vector of 1's.

(iii) Describe with one sentence why this loss function is much more efficient to compute than the naive-softmax loss.

Solution.

(i) Taking the derivative with respect to $\boldsymbol{v}_c$ from Equation (2), we obtain

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{v}_c} J_{\text{neg-sample}} &= \frac{\partial}{\partial \boldsymbol{v}_c} \left(-\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \right) \\
&= -\frac{\partial}{\partial \boldsymbol{v}_c} \log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \frac{\partial}{\partial \boldsymbol{v}_c} \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \\
&= \left(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{u}_o - \sum_{s=1}^{K} \frac{\partial}{\partial \boldsymbol{v}_c} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \\
&= \left(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{u}_o - \sum_{s=1}^{K} \left(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c) - 1\right) u_{w_s}.
\end{aligned}$$

Taking the derivative with respect to $\boldsymbol{u}_o$ from Equation (2), we have

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}_o} J_{\text{neg-sample}} &= \frac{\partial}{\partial \boldsymbol{v}_c} \left(-\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \right) \\
&= -\frac{\partial}{\partial \boldsymbol{u}_o} \log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \frac{\partial}{\partial \boldsymbol{u}_o} \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \\
&= \left(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c - 0 \\
&= \left(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c.
\end{aligned}$$

Taking the derivative with respect to the s^{th} negative sample $\boldsymbol{u}_{w_s}$ from Equation (2),

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}_{w_s}} J_{\text{neg-sample}} &= \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \left(-\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{i=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \right) \\
&= -\frac{\partial}{\partial \boldsymbol{u}_{w_s}} \log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \sum_{i=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \\
&= 0 - \left(\frac{\partial}{\partial \boldsymbol{u}_{w_s}} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) + \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \sum_{\substack{i=1 \\ i \neq s}}^{K} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \right) \\
&= -\left(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c - 0 \\
&= -\left(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c.
\end{aligned}$$

(ii) Note that all of the terms in the above derivatives are of the form $\left(\sigma\left(\boldsymbol{u}_i^\top \boldsymbol{v}_c\right) - 1\right) \boldsymbol{v}'$. Consider $\sigma\left(\boldsymbol{U}_{o,\{w_1,\ldots,w_K\}}^\top \boldsymbol{v}_c\right) - \mathbf{1}$:

$$\begin{aligned}
\sigma\left(\boldsymbol{U}_{o,\{w_1,\ldots,w_K\}}^\top \boldsymbol{v}_c\right) - \mathbf{1} &= \sigma\left(\begin{bmatrix} \boldsymbol{u}_o^\top \\ -\boldsymbol{u}_{w_1}^\top \\ \vdots \\ -\boldsymbol{u}_{w_k}^\top \end{bmatrix} \boldsymbol{v}_c \right) - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \sigma\left(\boldsymbol{u}_o^\top \boldsymbol{v}_c\right) \\ \sigma\left(-\boldsymbol{u}_{w_1}^\top \boldsymbol{v}_c\right) \\ \vdots \\ \sigma\left(-\boldsymbol{u}_{w_k}^\top \boldsymbol{v}_c\right) \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \sigma\left(\boldsymbol{u}_o^\top \boldsymbol{v}_c\right) - 1 \\ \sigma\left(-\boldsymbol{u}_{w_1}^\top \boldsymbol{v}_c\right) - 1 \\ \vdots \\ \sigma\left(-\boldsymbol{u}_{w_k}^\top \boldsymbol{v}_c\right) - 1 \end{bmatrix}
\end{aligned}$$

All components in the above vector are present in all three derivatives. Thus, we can use $\sigma\left(\boldsymbol{U}_{o,\{w_1,\ldots,w_K\}}^\top \boldsymbol{v}_c\right) - \mathbf{1}$ for backpropagation.

(iii) In the naive-softmax loss, we have to calculate the softmax function, which means having to iterate all words in the vocabulary, but in the negative sampling loss, we only need to iterate through K random outside words and one center word, making the computation much more efficient.

Problem 8. (4 points) 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 $\boldsymbol{u}_{w_1}, \ldots, \boldsymbol{u}_{w_K}$. 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:

$$\boldsymbol{J}_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \tag{3}$$

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 $\boldsymbol{u}_{w_s}$. Please write your answers in terms of the vectors $\boldsymbol{v}_c$ and $\boldsymbol{u}_{w_s}$, where $s \in [1, K]$. Hint: break up the sum in the loss function into two sums: a sum over all sampled words equal to w_s and a sum over all sampled words not equal to w_s. Notation-wise, you may write 'equal' and 'not equal' conditions below the summation symbols, such as in Equation 4.

Solution.

Taking the derivative with respect to the s^{th} negative sample $\boldsymbol{u}_{w_s}$ from Equation (3),

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{u}_{w_s}} J_{\text{neg-sample}} &= \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \left(-\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{i=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \right) \\
&= \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \left(-\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{\substack{1 \leq i \leq K \\ w_i = w_s}} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) - \sum_{\substack{1 \leq i \leq K \\ w_i \neq w_s}} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \right) \\
&= -\frac{\partial}{\partial \boldsymbol{u}_{w_s}} \log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \sum_{\substack{1 \leq i \leq K \\ w_i = w_s}} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) - \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \sum_{\substack{1 \leq i \leq K \\ w_i \neq w_s}} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) \\
&= 0 - \sum_{\substack{1 \leq i \leq K \\ w_i = w_s}} \frac{\partial}{\partial \boldsymbol{u}_{w_s}} \log(\sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_c)) - 0 \\
&= -\sum_{\substack{1 \leq i \leq K \\ w_i = w_s}} \left(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c
\end{aligned}$$

This could also be rewritten as

$$\frac{\partial}{\partial \boldsymbol{u}_{w_s}} J_{\text{neg-sample}} = -\text{count}_K(w_s) \left(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c) - 1\right) \boldsymbol{v}_c,$$

where $\text{count}_K(w_s)$ is the number of words equal to w_s in the negative sampling set.

Problem 9. (6 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:

$$\boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U}) = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U}) \tag{4}$$

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

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

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

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

Solution.

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

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

(iii)
$$\frac{\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{v}_w} = 0, \text{ when } w \neq c$$

2 Coding assignment

Problem 1. (40 points) After 40,000 iterations, the script will finish and a visualization for your word vectors will appear. It will also be saved as `word_vectors.png` in your project directory. **Include the plot in your homework write-up.** In at most three sentences, briefly explain what you see in the plot. This may include but is not limited to, observations on clusters and words that you expect to cluster but do not.

Solution.

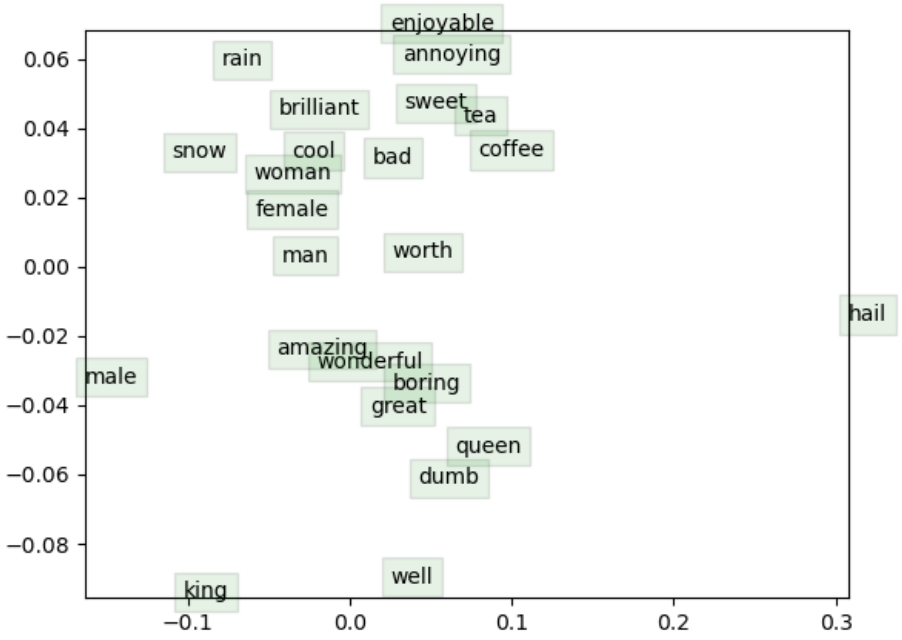

Figure 1: Visualization for word vectors after training the Skip-gram model

Let's begin with clusters, as one could notice, there are clusters formed by words with similar syntactic, for instance, "amazing", "wonderful", "boring", and "great" all cluster since they are adjectives with similar syntactic positions in a sentence; "woman", "female", and "man" is another example; weather-related words like "rain", "snow", and drink-related words such as "tea", "coffee" are relatively close to each other. Secondly, let's discuss some potential outliers, "hail", "male", and "king" stand out as they are not near any known clusters; also there are some words that seem to appear in the wrong cluster, for example "brilliant", and "bad" should join with those adjectives mentioned above; as we discussed in the lecture, ("male", "king"), and ("female, "queen") should be somewhat close to each other, but they are far away in our vector space. Lastly, let's talk about some weird clusters in our visualization, for some reason, "queen" and "dumb" appear to have great similarities; "man" is closer to "female" than "male".