1 ASSIGNMENT 2

(a) (3 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):

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

Your answer should be one line. You may describe your answer in words.

> As we know that $\boldsymbol{y}$ is a one hot vector, so
>
> $$-\sum_{w \in Vocab} \boldsymbol{y}_w \log(\hat{\boldsymbol{y}}_w) = -1 \times \boldsymbol{y}_o \log(\hat{\boldsymbol{y}}_o) = -\log(\hat{\boldsymbol{y}}_o)$$

(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 $\boldsymbol{U}$. 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)?

- **Note**: Your final answers for the partial derivative should follow the shape convention: the partial derivative of any function $f(x)$ with respect to x should have the **same shape** as x.[4]
- Please provide your answers for the partial derivative in vectorized form. For example, when we ask you to write your answers in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{U}$, you may not refer to specific elements of these terms in your final answer (such as $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\ldots$).

> (1)
>
> $$\begin{aligned}
J_{naive-softmax}(\boldsymbol{v}_c, o, \boldsymbol{U}) &= -\log P(o|c) \\
&= -\log \frac{exp(u_o^T v_c)}{\sum_{w \in Vocab} exp(u_w^T v_c)} \\
&= \log \sum exp(u_w^T v_c) - u_o^T v_c \\
& Therefore \\
\frac{\partial J}{\partial v_c} &= \frac{1}{\sum exp(u_w^T v_c)} \frac{\partial \sum exp(u_w^T v_c)}{\partial v_c} - u_o \\
&= \frac{1}{\sum exp(u_w^T v_c)} \sum exp(u_w^T v_c) u_w - u_o \\
&= \sum P(w|c) u_w - u_o \\
& When\ \partial J\ is\ 0,\ we\ have \\
u_o &= \sum P(w|c) u_w
\end{aligned}$$
>
> (2)
>
> That is because when we are doing this gradient descent, P(o|c) tends to go up, which means outside words are closer to the center word

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

$$
\begin{aligned}
\frac{\partial J}{\partial u_w} &= \frac{1}{\sum exp(u_w^T v_c)} \frac{\partial \sum exp(u_w^T v_c)}{\partial u_w} - \frac{\partial u_o^T v_c}{\partial u_w} \\
&= \frac{1}{\sum exp(u_w^T v_c)} exp(u_w^T v_c) v_c - \frac{\partial u_o^T v_c}{\partial u_w} \\
&= P(w|c) v_c - \frac{\partial u_o^T v_c}{\partial u_w} \\
When\ w = o,\ then\ y_o = 1,\ & \frac{\partial u_o^T v_c}{\partial u_w} = v_c \\
\frac{\partial J}{\partial u_o} &= (\hat{y}_o - 1) v_c = (\hat{y}_o - y_o) v_c \\
When\ w \neq o,\ then\ y_w = 0,\ & \frac{\partial u_o^T v_c}{\partial u_w} = 0 \\
\frac{\partial J}{\partial u_w} &= (\hat{y}_w - 0) v_c = (\hat{y}_w - y_w) v_c
\end{aligned}
$$

(d) (1 point) 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.

$$
\begin{aligned}
\frac{\partial J}{\partial u_i} &= (\hat{y}_i - y_i) v_c \\
i &= 1, 2, \ldots, |Vocab|
\end{aligned}
$$

(e) (2 points) The ReLU (Rectified Linear Unit) activation function is given by Equation 4:

$$
f(x) = \max(0, x) \tag{4}
$$

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.[5]

$$
\begin{aligned}
&When\ x < 0 \\
&\frac{df}{dx} = 0 \\
&When\ x > 0 \\
&\frac{df}{dx} = 1
\end{aligned}
$$

(f) (3 points) The sigmoid function is given by Equation 5:

$$
\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{5}
$$

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 $\sigma(x)$.

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

$w_i \neq w_j$ 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:

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

for a sample $w_1, \ldots w_K$, where $\sigma(\cdot)$ is the sigmoid function.[7]

(i) 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 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 part (f) to help compute the necessary gradients here.

(ii) In lecture, we learned that an efficient implementation of backpropagation leverages the re-use 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 $\boldsymbol{U}_{o,\{w_1,\ldots,w_K\}} = [\boldsymbol{u}_o, -\boldsymbol{u}_{w_1}, \ldots, -\boldsymbol{u}_{w_K}]$, a matrix with the outside vectors stacked as columns, and $\mathbf{1}$, a $(K+1) \times 1$ vector of 1's. [8]

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

(i)

$$
\begin{aligned}
\frac{\partial J}{\partial v_c} &= -\frac{1}{\sigma(u_o^T v_c)} \sigma(u_o^T v_c)(1-\sigma(u_o^T v_c)) u_o - \sum_{s=1}^{K} \frac{1}{\sigma(-u_{w_s}^T v_c)} \sigma(-u_{w_s}^T v_c)(1-\sigma(-u_{w_s}^T v_c))(-u_{w_s}) \\
&= -(1-\sigma(u_o^T v_c)) u_o + \sum_{s=1}^{K} (1-\sigma(-u_{w_s}^T v_c)) u_{w_s} \\
\frac{\partial J}{\partial u_o} &= -(1-\sigma(u_o^T v_c)) v_c \\
\frac{\partial J}{\partial u_{w_s}} &= (1-\sigma(-u_{w_s}^T v_c)) v_c
\end{aligned}
$$

(ii)

$$
(1-\sigma(u_w^T v_c))
$$

(iii)

This loss function does not calculate all the losses from every word in vocabulary.

And it also eliminates the use of logarithm

(h) (2 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:

$$\boldsymbol{J}_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \tag{7}$$

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 8.

$$\frac{\partial J}{\partial u_{w_s}} = \sum_{i \wedge (w_s = w_i)}^{K} (1 - \sigma(-u_{w_s}^T v_c)) v_c$$

(i) (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:

$$\boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U}) = \sum_{\substack{-m \le j \le m \\ j \neq 0}} \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U}) \tag{8}$$

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.

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

(i)

$$\sum_{-m \le j \le m, j \neq 0} \frac{\partial J(v_c, wt + j, v)}{\partial U}$$

(ii)

$$\sum_{-m \le j \le m,\ j \neq 0} (1 - \sigma(u_{w_j}^T v_{w_0})) u_{w_0} + \sum_{s=1}^{K} (1 - \sigma(-u_{w_s}^T v_{w_0})) u_{w_s}$$

(iii)

$$0$$