

Author: Benjamin Smidt
Created: October 3rd, 2022
Last Updated: October 5th, 2022

CS 224N A2: Word Vectors

Note to the reader. This is my work for assignment one of Stanford's course CS 224N: Natural Language Processing with Deep Learning. You can find the lecture Winter 2021 lectures series on YouTube [here](#). This document is meant to be used as a reference, explanation, and resource for the assignment, not necessarily a comprehensive overview of Word Vectors. If there's a typo or a correction needs to be made, feel free to email me at benjamin.smidt@utexas.edu so I can fix it. Thank you! I hope you find this document helpful :).

Contents

1	Written Understanding	2
1.1	Problem A	2
1.2	Problem B	3
1.3	Problem C	4
1.4	Problem D	6
1.5	Problem E	7
1.6	Problem F	7
1.7	Problem G	8
	1.7.1 (i)	9
	1.7.2 (ii)	11
	1.7.3 (iii)	12
1.8	Problem H	12

1 Written Understanding

1.1 Problem A

Instructions

Prove that the naive-softmax loss is the same as the cross-entropy loss between \mathbf{y} and $\hat{\mathbf{y}}$, i.e. (note that $\mathbf{y}, \hat{\mathbf{y}}$ are vectors and \hat{y}_o is a scalar).

Solution

To start with, our naive-softmax loss is defined as

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

where

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

Let's define our variables. $\hat{\mathbf{y}}$ is our score vector. Note that the numerator is a vector of length V while the denominator is a scalar (this notation is a bit abusive but I think it actually makes things clearer). Thus, each index in the vector can be interpreted as the probability that the corresponding word (using the the index and one hot vector) is the center word.

$$\hat{\mathbf{y}} = \frac{\exp(\mathbf{u}^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)}$$

\mathbf{y} is the one hot vector of the true center word. Then

$$- \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = -\mathbf{y}_1 \log(\hat{\mathbf{y}}_1) + \dots + -\mathbf{y}_o \log(\hat{\mathbf{y}}_o) + \dots + -\mathbf{y}_w \log(\hat{\mathbf{y}}_w)$$

Where the index o indicates the index containing the only 1 within $\hat{\mathbf{y}}$ (since it is a one-hot vector).

$$- \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = -(0) \log(\hat{\mathbf{y}}_1) + \dots + -(1) \log(\hat{\mathbf{y}}_o) + \dots + -(0) \log(\hat{\mathbf{y}}_w)$$

$$- \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = -\log(\hat{\mathbf{y}}_o)$$

$$- \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = -\log\left(\frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)}\right)$$

$$\begin{aligned}
& - \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = -\log P(O = o | C = c) \\
& - \sum_{w \in \text{Vocab}} \mathbf{y}_w \log(\hat{\mathbf{y}}_w) = \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})
\end{aligned}$$

I know the instructions said one line but I was going for clarity here. Obviously, I could not define the variables (leave it to you to figure out what they mean) and just write some one liner that connects the dots. However, I wanted to make this as clear to understand as possible. Hopefully this leaves no room for ambiguity.

1.2 Problem B

Instructions

Compute the partial derivative of $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$ with respect to \mathbf{v}_c . Please write your answer in terms of \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{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 \mathbf{v}_c a more desirable vector (namely, a vector closer to outside word vectors in its window)?

Solution

We start with our definition of $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$.

$$\begin{aligned}
\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U}) &= -\log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} &= \frac{\partial}{\partial \mathbf{v}_c} -\log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} &= \frac{\partial}{\partial \mathbf{v}_c} -\mathbf{u}_o^\top \mathbf{v}_c + \frac{\partial}{\partial \mathbf{v}_c} \log \sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)
\end{aligned}$$

Everything up to this point should be easy to follow if you remember calculus (although see the first lecture where he goes through these exact steps in detail if you are confused). I then do a quick change of variables to ensure I know what I'm taking my partial derivative with respect to.

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = -\mathbf{u}_o + \frac{1}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \frac{\partial}{\partial \mathbf{v}_c} \sum_{j \in \text{Vocab}} \exp(\mathbf{u}_j^\top \mathbf{v}_c)$$

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = -\mathbf{u}_o + \frac{\sum_{j \in \text{Vocab}} \mathbf{u}_j^\top \exp(\mathbf{u}_j^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)}$$

From here I hope you can see that if you remove the \mathbf{u}_j^\top form the second sum term you're left with $\hat{\mathbf{y}}$. Simplifying with this in mind we get the following.

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = -\mathbf{U}^\top \mathbf{y} + \mathbf{U}^\top \hat{\mathbf{y}}_w$$

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = \mathbf{U}^\top (\hat{\mathbf{y}} - \mathbf{y})$$

1. The gradient is zero when $\hat{\mathbf{y}} = \mathbf{y}$. Obviously if our predicted and correct vectors are equivalent then our accuracy is perfect and there's no update that could improve the loss.
2. Because we're doing gradient descent (as opposed to ascent), the update adds some portion ($\propto \alpha$) of $\mathbf{U}^\top \mathbf{y}$ and subtracts some portion of $\mathbf{U}^\top \hat{\mathbf{y}}$. This makes intuitive sense because adding the correct vector \mathbf{u}_o^\top makes it more like that vector (which is what we want) and subtracting by $\mathbf{U}^\top \hat{\mathbf{y}}$, the weighted average of our incorrect vectors that are producing the loss, makes it less like those vectors (again, what we want).

1.3 Problem C

Instructions

Compute the partial derivatives of $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$ with respect to each of the 'outside' word vectors, \mathbf{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 \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{v}_c . In this subpart, you may use specific elements within these terms as well (such as $\mathbf{y}_1, \mathbf{y}_2, \dots$). Note that \mathbf{u}_w is a vector while $\mathbf{y}_1, \mathbf{y}_2, \dots$ are scalars.

Solution

Case 1: $\mathbf{u}_w = \mathbf{u}_o$

We start with the case that $\mathbf{u}_w = \mathbf{u}_o$. That is, the gradient with respect to the correct embedding vector.

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

$$\begin{aligned}
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= \frac{\partial}{\partial \mathbf{u}_o} - \log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= \frac{\partial}{\partial \mathbf{u}_o} - \log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= \frac{\partial}{\partial \mathbf{u}_o} - \mathbf{u}_o^\top \mathbf{v}_c + \frac{\partial}{\partial \mathbf{u}_o} \log \sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c) \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= -\mathbf{v}_c + \frac{1}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \frac{\partial}{\partial \mathbf{u}_o} \sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c) \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= -\mathbf{v}_c + \frac{\mathbf{v}_c \exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= -\mathbf{v}_c + \mathbf{v}_c (\hat{\mathbf{y}} \cdot \mathbf{y}) \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} &= \mathbf{v}_c (\hat{\mathbf{y}}_0 - 1)
\end{aligned}$$

Note that $\hat{\mathbf{y}} \cdot \mathbf{y}$ is the dot product of the two vectors.

Case 2: $\mathbf{u}_w = \mathbf{u}_o$

Now we move onto the case that $\mathbf{u}_w \neq \mathbf{u}_o$. That is, the gradient with respect to any vector \mathbf{u}_w that isn't \mathbf{u}_o . I'll use the notation \mathbf{u}_j to indicate a specific \mathbf{u}_w and (hopefully) prevent any confusion.

$$\begin{aligned}
\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U}) &= -\log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} &= \frac{\partial}{\partial \mathbf{u}_j} - \log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} &= \frac{\partial}{\partial \mathbf{u}_j} - \log \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} &= \frac{\partial}{\partial \mathbf{u}_j} - \mathbf{u}_o^\top \mathbf{v}_c + \frac{\partial}{\partial \mathbf{u}_j} \log \sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c) \\
\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} &= \frac{1}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)} \frac{\partial}{\partial \mathbf{u}_j} \sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)
\end{aligned}$$

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} = \frac{\mathbf{v}_c \exp(\mathbf{u}_j^\top \mathbf{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\mathbf{u}_w^\top \mathbf{v}_c)}$$

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} = \mathbf{v}_c \hat{\mathbf{y}}_j$$

1.4 Problem D

Instructions

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

Solution

We already know

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} = \mathbf{v}_c \hat{\mathbf{y}}_j \quad \text{and} \quad \frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_0} = \mathbf{v}_c (\hat{\mathbf{y}}_0 - 1)$$

Since \mathbf{y} is a one hot vector, then

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_j} = \mathbf{v}_c (\hat{\mathbf{y}} - \mathbf{y})$$

With that we can write

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{U}} = \frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_1}, \dots, \frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_V}$$

where V is the index of the last word vector in \mathbf{U} . It follows then that

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{U}} = \mathbf{v}_c \hat{\mathbf{y}}_1, \mathbf{v}_c \hat{\mathbf{y}}_2, \dots, \mathbf{v}_c \hat{\mathbf{y}}_V$$

$$\frac{\partial \mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{U}} = \mathbf{v}_c (\hat{\mathbf{y}} - \mathbf{y})^\top$$

where we use the outer product to get the proper shape for \mathbf{U} .

1.5 Problem E

Instructions

The ReLU (Rectified Linear Unit) activation function is given by the Equation:

$$f(x) = \max(0, x) \quad (1)$$

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

We can break ReLU into two cases.

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x > 0 \end{cases}$$

Then the derivation becomes trivial.

$$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

1.6 Problem F

Instructions

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

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

$$\begin{aligned}
\sigma'(x) &= \frac{e^x}{(e^x + 1)^2} \\
\sigma'(x) &= \frac{e^x}{e^x + 1} \frac{1}{e^x + 1} \\
\sigma'(x) &= \frac{1}{1 + e^{-x}} \frac{1}{e^x + 1} \\
\sigma'(x) &= \sigma(x) \frac{1}{e^x + 1} \\
\sigma'(x) &= \sigma(x) \frac{e^x + 1 - e^x}{e^x + 1} \\
\sigma'(x) &= \sigma(x) \left(\frac{e^x + 1}{e^x + 1} - \frac{e^x}{e^x + 1} \right) \\
\sigma'(x) &= \sigma(x) \left(1 - \frac{1}{1 + e^{-x}} \right) \\
\sigma'(x) &= \sigma(x) (1 - \sigma(x))
\end{aligned}$$

1.7 Problem G

Instructions

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

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

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

1.7.1 (i)

Instructions

Please repeat parts (b) and (c), computing the partial derivatives of $\mathbf{J}_{\text{neg-sample}}$ with respect to \mathbf{v}_c , with respect to \mathbf{u}_o , and with respect to the s^{th} negative sample \mathbf{u}_{w_s} . Please write your answers in terms of the vectors \mathbf{v}_c , \mathbf{u}_o , and \mathbf{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.

Solution

(b) Compute the partial derivative of $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$ with respect to \mathbf{v}_c . Please write your answer in terms of \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{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 \mathbf{v}_c a more desirable vector (namely, a vector closer to outside word vectors in its window)?

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

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

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

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

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = -\frac{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)}{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)} [1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)] \mathbf{u}_o - \sum_{s=1}^K \frac{\sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)}{(\sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c))} [1 - \sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)] (-\mathbf{u}_{w_s})$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K [1 - \sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K \left[1 - \frac{1}{1 + \exp(-(-\mathbf{u}_{w_s}^\top \mathbf{v}_c))} \right] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K \left[\frac{1 + \exp(\mathbf{u}_{w_s}^\top \mathbf{v}_c)}{1 + \exp(\mathbf{u}_{w_s}^\top \mathbf{v}_c)} - \frac{1}{1 + \exp(\mathbf{u}_{w_s}^\top \mathbf{v}_c)} \right] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K \left[\frac{\exp(\mathbf{u}_{w_s}^\top \mathbf{v}_c)}{1 + \exp(\mathbf{u}_{w_s}^\top \mathbf{v}_c)} \right] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K \left[\frac{1}{1 + \exp(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)} \right] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{u}_o + \sum_{s=1}^K \sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c) \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{v}_c} = -[1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)] \mathbf{u}_o + \sum_{s=1}^K \sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c) \mathbf{u}_{w_s}$$

You can see that the gradient is zero when $[1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)] \mathbf{u}_o = \sum_{s=1}^K \sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c) \mathbf{u}_{w_s}$. For our first term, if $\mathbf{u}_o^\top \mathbf{v}_c$ is small (dissimilar) then $\sigma(\mathbf{u}_o^\top \mathbf{v}_c)$ evaluates to a large number. Thus, in our gradient descent update, we'll be adding $\mathbf{C} \mathbf{u}_o$ ($\mathbf{C} = 1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)$) to \mathbf{v}_c . Intuitively this makes sense because adding $\mathbf{C} \mathbf{u}_o$ to \mathbf{v}_c will make \mathbf{v}_c more like \mathbf{u}_o , which is what we want.

If $\mathbf{u}_o^\top \mathbf{v}_c$ is large (similar) then $\sigma(\mathbf{u}_o^\top \mathbf{v}_c)$ evaluates to a small number and our gradient update for the first time adds approximately zero. Again, this is intuitive because if the vectors are similar already then we don't need to change \mathbf{v}_c much.

For our second term, a similar line of reasoning can be invoked to see that we're subtracting out $\mathbf{C} \mathbf{u}_{w_s}$ from \mathbf{v}_c if $\sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c)$ is large and doing nothing if $\sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c)$ is small. Again, this makes sense because we want \mathbf{v}_c to be less like the vectors that aren't \mathbf{u}_o and don't need to change \mathbf{v}_c if that's already the case.

(c) Compute the partial derivatives of $\mathbf{J}_{\text{naive-softmax}}(\mathbf{v}_c, o, \mathbf{U})$ with respect to each of the 'outside' word vectors, \mathbf{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 \mathbf{y} , $\hat{\mathbf{y}}$, and \mathbf{v}_c . In this subpart, you may use specific elements within these terms as well (such as $\mathbf{y}_1, \mathbf{y}_2, \dots$). Note that \mathbf{u}_w is a

vector while $\mathbf{y}_1, \mathbf{y}_2, \dots$ are scalars.

Case 1: $\frac{\partial}{\partial \mathbf{u}_o}$

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

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} = -\frac{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)}{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)} [1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)] \mathbf{v}_c$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_o} = [\sigma(\mathbf{u}_o^\top \mathbf{v}_c) - 1] \mathbf{v}_c$$

Case 2: $\frac{\partial}{\partial \mathbf{u}_{w_s}}$

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

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_{w_s}} = -\frac{\sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)}{\sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)} [1 - \sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)] (-\mathbf{u}_{w_s})$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_{w_s}} = [1 - \sigma(-\mathbf{u}_{w_s}^\top \mathbf{v}_c)] \mathbf{u}_{w_s}$$

$$\frac{\partial \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U})}{\partial \mathbf{u}_{w_s}} = \sigma(\mathbf{u}_{w_s}^\top \mathbf{v}_c) \mathbf{u}_{w_s}$$

See part (b) of this problem for that last jump, it's the exact same as before.

1.7.2 (ii)

Instructions

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 $\mathbf{U}_{o, \{w_1, \dots, w_K\}} = [\mathbf{u}_o, -\mathbf{u}_{w_1}, \dots, -\mathbf{u}_{w_K}]$, a matrix with the outside vectors stacked as columns, and $\mathbf{1}$, a $(K+1) \times 1$ vector of 1's.

Solution

Since we compute the sigmoid function so much we could reuse the following:

$$\sigma(\mathbf{U}_o^\top \mathbf{v}_c) - \mathbf{1}$$

1.7.3 (iii)

Instructions

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

Caveat: So far we have looked at re-using quantities and approximating softmax with sampling for faster gradient descent. Do note that some of these optimizations might not be necessary on modern GPUs and are, to some extent, artifacts of the limited compute resources available at the time when these algorithms were developed.

Solution

This loss function is much more efficient because, provided we have enough negative samples, the negative sampling we use will approximate the gradient update we would've gotten for the entire vocabulary but with literally a fraction of the vocabulary used. Instead of iterating through the entire vocabulary we can just iterate through negative samples which can be orders of magnitude smaller in size.

1.8 Problem H

Instructions

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, \dots, w_K and their outside vectors as $\mathbf{u}_{w_1}, \dots, \mathbf{u}_{w_K}$. 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 $o \notin \{w_1, \dots, w_K\}$. For a center word c and an outside word o , the negative sampling loss function is given by:

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

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

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

Solution