

CS 224n Assignment #2

Yoshihiro Kumazawa

May 30, 2020

1 Written

(a)

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

(b)

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

(c)

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

(d)

$$\frac{d}{d\mathbf{x}} \sigma(\mathbf{x}) = \frac{d}{d\mathbf{x}} \frac{1}{1 + e^{-\mathbf{x}}} = \frac{e^{-\mathbf{x}}}{(1 + e^{-\mathbf{x}})^2} = \sigma(\mathbf{x})(1 - \sigma(\mathbf{x})).$$

(e)

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{v}_c} \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) &= \frac{\partial}{\partial \mathbf{v}_c} \left(-\log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \log(\sigma(-\mathbf{u}_k^\top \mathbf{v}_c)) \right) \\
&= -\frac{\partial}{\partial \mathbf{v}_c} \log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) - \sum_{k=1}^K \frac{\partial}{\partial \mathbf{v}_c} \log(1 - \sigma(\mathbf{u}_k^\top \mathbf{v}_c)) \\
&= -\frac{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)(1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c))}{\sigma(\mathbf{u}_o^\top \mathbf{v}_c)} \mathbf{u}_o + \sum_{k=1}^K \frac{(1 - \sigma(\mathbf{u}_k^\top \mathbf{v}_c))\sigma(\mathbf{u}_k^\top \mathbf{v}_c)}{1 - \sigma(\mathbf{u}_k^\top \mathbf{v}_c)} \mathbf{u}_k \\
&= -(1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)) \mathbf{u}_o + \sum_{k=1}^K \sigma(\mathbf{u}_k^\top \mathbf{v}_c) \mathbf{u}_k. \\
\frac{\partial}{\partial \mathbf{u}_o} \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) &= -\frac{\partial}{\partial \mathbf{u}_o} \log(\sigma(\mathbf{u}_o^\top \mathbf{v}_c)) = -(1 - \sigma(\mathbf{u}_o^\top \mathbf{v}_c)) \mathbf{v}_c. \\
\frac{\partial}{\partial \mathbf{u}_k} \mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) &= -\frac{\partial}{\partial \mathbf{u}_k} \sum_{k=1}^K \log(1 - \sigma(\mathbf{u}_k^\top \mathbf{v}_c)) = \sigma(\mathbf{u}_k^\top \mathbf{v}_c) \mathbf{v}_c.
\end{aligned}$$

These are computationally less expensive than the naive-softmax loss because its summation ranges over only K numbers, which is usually much smaller than $|\text{Vocab}|$.

$$\begin{aligned}
\text{(f) (i)} \quad \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{U} &= \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \partial \mathbf{J}(\mathbf{v}_c, w_{t+j}, \mathbf{U}) / \partial \mathbf{U}. \\
\text{(ii)} \quad \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_c &= \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \partial \mathbf{J}(\mathbf{v}_c, w_{t+j}, \mathbf{U}) / \partial \mathbf{v}_c. \\
\text{(iii)} \quad \partial \mathbf{J}_{\text{skip-gram}}(\mathbf{v}_c, w_{t-m}, \dots, w_{t+m}, \mathbf{U}) / \partial \mathbf{v}_w &= 0.
\end{aligned}$$

2 Coding

(a) See `word2vec.py`.

(b) See `sgd.py`.

(c) Figure 1 shows the visualized word vectors. Some similar words like 'woman' and 'female' are put close but other similar words like 'man' and 'male' are far apart. Thus our word embedding does not retain the parallelogram property that well-trained word embedding methods are expected to have.

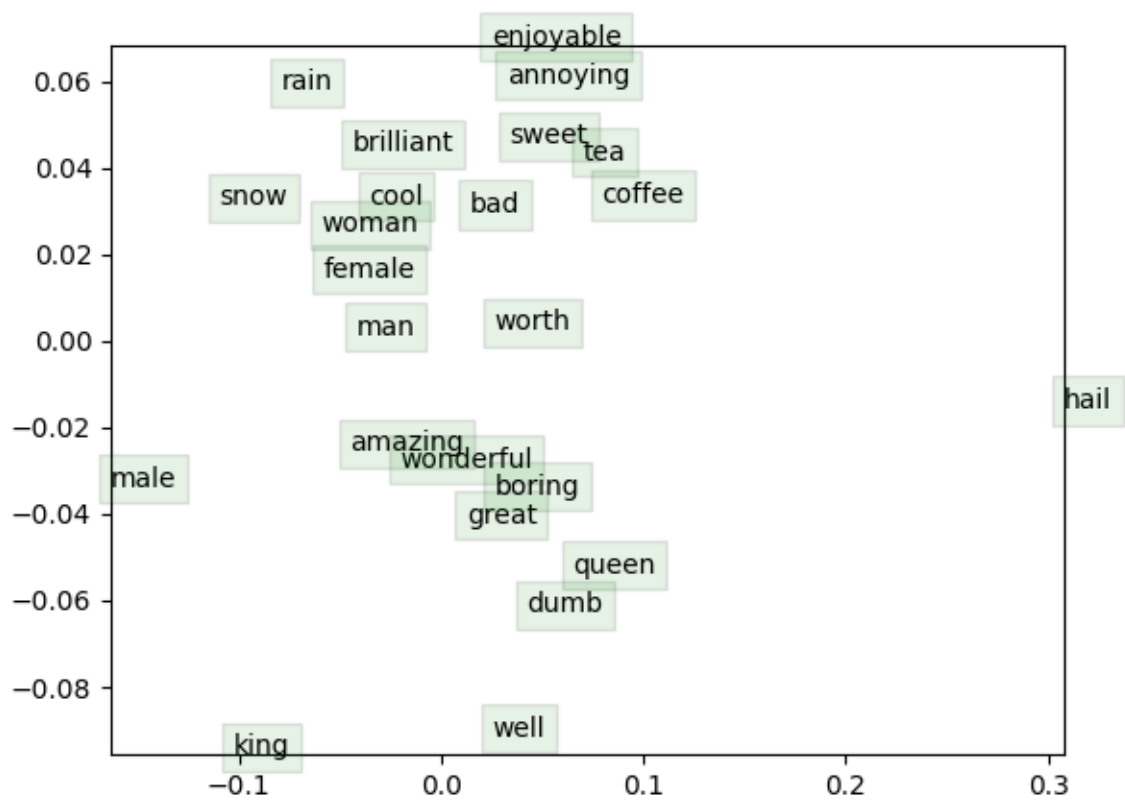


Figure 1: Visualized word vectors