cs224n 2019 Assignment 1

Zhou Xiaorui - jarvan000@gmail.com

March 20, 2019

Notice: The answer is done by myself, and I'm not a Stanford student.

(a) only the o'th position of $\boldsymbol{y}$ is not zero, so the sum can be reduced to only one part:

$$-\sum_{w\in\text{Vocab}} y_w \log(\hat{y}_w) = -y_o \log(\hat{y}_o) = -\log(\hat{y}_o) \tag{1}$$

(b) since

$$\begin{aligned}\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) &= -\log\frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum_{w\in\text{Vocab}}\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \\ &= -\boldsymbol{u}_o^\top \boldsymbol{v}_c + \log\sum_{w\in\text{Vocab}}\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)\end{aligned} \tag{2}$$

so we have

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{v}_c} &= -\boldsymbol{u}_o + \frac{\sum_{x\in\text{Vocab}}\exp(\boldsymbol{u}_x^\top \boldsymbol{v}_c)\boldsymbol{u}_x}{\sum_{w\in\text{Vocab}}\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)} \\ &= -\boldsymbol{u}_o + \sum_{x\in\text{Vocab}}\frac{\exp(\boldsymbol{u}_x^\top \boldsymbol{v}_c)}{\sum_{w\in\text{Vocab}}\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}\boldsymbol{u}_x \\ &= -\boldsymbol{u}_o + \sum_{x\in\text{Vocab}}\hat{y}_x\boldsymbol{u}_x \\ &= \boldsymbol{U}(\hat{y} - y)\end{aligned}$$

According to the result, we need to compute the matrix multiplication over the whole vocabulary, which may contain millions of words, and that is time-consuming and unnecessary.

(c) when $\boldsymbol{u}_w = \boldsymbol{u}_o$

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_w} &= \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_o} \\ &= -\boldsymbol{v}_c + \frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum_{w\in\text{Vocab}}\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}\boldsymbol{v}_c \\ &= -\boldsymbol{v}_c + \hat{y}_o\boldsymbol{v}_c\end{aligned}$$

when $\boldsymbol{u}_w \neq \boldsymbol{u}_o$

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_w} &= \frac{\exp(\boldsymbol{u}_w^\top \boldsymbol{v}_c)}{\sum_{x\in \text{Vocab}} \exp(\boldsymbol{u}_x^\top \boldsymbol{v}_c)} \boldsymbol{v}_c \\ &= \hat{y}_w \boldsymbol{v}_c\end{aligned}$$

so from above two equation, we can get

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{U}} = \boldsymbol{v}_c(\hat{y} - y)^\top$$

(d) very basic partial derivative computation.

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

(e) basic partial derivative computation, using chain rule.

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{v}_c} &= -\frac{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))\boldsymbol{u}_o}{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)} - \sum_{k=1}^{K} \frac{(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c))(1-(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))(-\boldsymbol{u}_k)}{\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)} \\ &= -(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))\boldsymbol{u}_o - \sum_{k=1}^{K}(1-(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))(-\boldsymbol{u}_k) \\ &= -(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))\boldsymbol{u}_o + \sum_{k=1}^{K}(1-(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))\boldsymbol{u}_k\end{aligned}$$

for $\boldsymbol{u}_o$ and $\boldsymbol{u}_w$, we apply the same technique.

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_o} &= -\frac{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))\boldsymbol{v}_c}{\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)} \\ &= -(1-\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c))\boldsymbol{v}_c\end{aligned}$$

$$\begin{aligned}\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}_k} &= -\frac{(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c))(1-(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))(-\boldsymbol{v}_c)}{\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)} \\ &= (1-(\sigma(-\boldsymbol{u}_k^\top \boldsymbol{v}_c)))(-\boldsymbol{v}_c)\end{aligned}$$

using Negative Sampling loss, we only need to compute K+1 parameter, which is much more efficient than if we need to compute over the whole vocabulary.

(f) just add the loss of every context word, we can get the answer.

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

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

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

Useful reference: http://www.amendgit.com/post/cs224n/cs224n-assignment-1/