CS224n Winter 2019 Homework 2: word2vec

SUNet ID: 05794739
Name: Luis Perez
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a) The key insight for this equality is that the vector of the true distribution $\boldsymbol{y}$ is one-hot encoded vector with 1 for the true, outside word o and 0 everywhere else. We therefore have:

$$\begin{aligned}
\text{CrossEntropy}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= -\sum_{w \in Vocab} y_w \log(\hat{y}_w) \\
&= -1 \cdot \log(\hat{y}_o) - \sum_{w \in Vocab, w \neq o} 0 \cdot \log(\hat{y}_w) \\
&= -\log(\hat{y}_o) \\
&= -\log P(O = o \mid C = c) = \boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})
\end{aligned}$$

(b) We compute the partial derivate of the cross-entropy loss with respect to v_c.

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{v}_c} &= -\frac{\partial}{\partial \boldsymbol{v}_c}\left[\sum_{w \in Vocab} \boldsymbol{y}_w \log \hat{\boldsymbol{y}}_w\right] && \text{(Results from 1a)} \\
&= -\sum_{w \in Vocab} \boldsymbol{y}_w \frac{\partial}{\partial \boldsymbol{v}_c} \log \frac{\exp \boldsymbol{u}_w^T \boldsymbol{v}_c}{\sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c} && \text{(Definition of } \hat{\boldsymbol{y}}_w\text{)} \\
&= -\sum_{w \in Vocab} \frac{\boldsymbol{y}_w}{\hat{\boldsymbol{y}}_w} \frac{\partial}{\partial \boldsymbol{v}_c} \frac{\exp \boldsymbol{u}_w^T \boldsymbol{v}_c}{\sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c} && \text{(Derivative of } \log x \text{ and Chain rule)} \\
&= -\sum_{w \in Vocab} \frac{\boldsymbol{y}_w}{\hat{\boldsymbol{y}}_w} \left[\frac{\boldsymbol{u}_w \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c) \sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c}{\left(\sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c\right)^2} - \frac{\exp(\boldsymbol{u}_w^T \boldsymbol{v}_c) \sum_{k \in Vocab} \boldsymbol{u}_k \exp \boldsymbol{u}_k^T \boldsymbol{v}_c}{\left(\sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c\right)^2}\right] \\
& && \text{(Quotient Rule of Derivates)} \\
&= -\sum_{w \in Vocab} \frac{\boldsymbol{y}_w}{\hat{\boldsymbol{y}}_{\boldsymbol{w}}} \left[\boldsymbol{u}_w \hat{\boldsymbol{y}}_w - \hat{\boldsymbol{y}}_w \sum_{k \in Vocab} \frac{\boldsymbol{u}_k \exp \boldsymbol{u}_k^T \boldsymbol{v}_c}{\sum_{\ell \in Vocab} \exp \boldsymbol{u}_\ell^T \boldsymbol{v}_c}\right] \\
& && \text{(Simplification using definition of } \hat{\boldsymbol{y}}_w\text{, reindex sum)} \\
&= -\sum_{w \in Vocab} \boldsymbol{y}_w \boldsymbol{u}_{\boldsymbol{w}} + \left(\sum_{w \in Vocab} \boldsymbol{y}_w\right)\left(\sum_{k \in Vocab} \boldsymbol{u}_k \hat{\boldsymbol{y}}_k\right) \\
& && \text{(Defintion of } \hat{\boldsymbol{y}}_k\text{, distribute sum, simplify)} \\
&= \boldsymbol{U}[\hat{\boldsymbol{y}} - \boldsymbol{y}] && \text{(Convert to matrix form.)}
\end{aligned}$$

(c) We compute the partial derivate of the cross-entry loss with respect to $\boldsymbol{u}_w$. We have:

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{u}_w} &= -\frac{\partial}{\partial \boldsymbol{u}_w}\left[\sum_{k \in Vocab} \boldsymbol{y}_k \log \hat{\boldsymbol{y}}_k\right] && \text{(Results from 1a)} \\
&= -\sum_{k \in Vocab} \boldsymbol{y}_k \frac{\partial}{\partial \boldsymbol{u}_w} \log \frac{\exp \boldsymbol{u}_w^T \boldsymbol{v}_c}{\sum_{k \in Vocab} \exp \boldsymbol{u}_k^T \boldsymbol{v}_c} && \text{(Definition of } \hat{\boldsymbol{y}}_k\text{)} \\
&= -\sum_{k \in Vocab} \frac{\boldsymbol{y}_k}{\hat{\boldsymbol{y}}_k} \frac{\partial}{\partial \boldsymbol{u}_w} \frac{\exp \boldsymbol{u}_k^T \boldsymbol{v}_c}{\sum_{\ell \in Vocab} \exp \boldsymbol{u}_\ell^T \boldsymbol{v}_c} && \text{(Derivative of } \log x \text{ and Chain rule)} \\
&= -\frac{\boldsymbol{y}_w}{\hat{\boldsymbol{y}}_w} \frac{\boldsymbol{v}_c \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c) \sum_{\ell \in Vocab} \exp \boldsymbol{u}_\ell^T \boldsymbol{v}_c - \boldsymbol{v}_c \left(\exp \boldsymbol{u}_w^T \boldsymbol{v}_c\right)^2}{\left(\sum_{\ell \in Vocab} \exp \boldsymbol{u}_\ell^T \boldsymbol{v}_c\right)^2} \\
&+ \sum_{k \in Vocab, k \neq w} \frac{\boldsymbol{y}_k}{\hat{\boldsymbol{y}}_k} \frac{\boldsymbol{v}_c \exp\left(\boldsymbol{u}_w^T \boldsymbol{v}_c\right) \exp\left(\boldsymbol{u}_k^T \boldsymbol{v}_c\right)}{\left(\sum_{\ell \in Vocab} \exp \boldsymbol{u}_\ell^T \boldsymbol{v}_c\right)^2} && \text{(Quotient Rule and split into cases)} \\
&= -\boldsymbol{v}_c \boldsymbol{y}_w \left[1 - \hat{\boldsymbol{y}}_w\right] + \boldsymbol{v}_c \sum_{k \in Vocab, k \neq w} \boldsymbol{y}_k \hat{\boldsymbol{y}}_w && \text{(Simplify)} \\
&= [-\boldsymbol{y}_w + \hat{\boldsymbol{y}}_w(\boldsymbol{y}_w + \sum_{k \in Vocab, k \neq w} \boldsymbol{y}_k)]\boldsymbol{v}_c && \text{(Refactoring)} \\
&= [\hat{\boldsymbol{y}}_w - \boldsymbol{y}_w]\boldsymbol{v}_c
\end{aligned}$$

We can, in fact, write the above for the entire matrix $\boldsymbol{U}$ as follows:

$$\frac{\partial J}{\partial \boldsymbol{U}} = \boldsymbol{v}_c[\hat{\boldsymbol{y}} - \boldsymbol{y}]^T$$

(d) We compute the derivative element by element. We have:

$$\begin{aligned}
\frac{d}{d\boldsymbol{x}_i}\sigma(\boldsymbol{x}_i) &= \frac{d}{d\boldsymbol{x}_i}\left[\frac{e^{\boldsymbol{x}_i}}{1 + e^{\boldsymbol{x}_i}}\right] \\
&= \frac{e^{\boldsymbol{x}_i}(1 + e^{\boldsymbol{x}_i}) - e^{\boldsymbol{x}_i} \cdot e^{\boldsymbol{x}_i}}{(1 + e^{\boldsymbol{x}_i})^2} \\
&= \left(\frac{e^{\boldsymbol{x}_i}}{1 + e^{\boldsymbol{x}_i}}\right)\left(1 - \frac{e^{\boldsymbol{x}_i}}{1 + e^{\boldsymbol{x}_i}}\right) \\
&= \sigma(x_i)[1 - \sigma(x_i)]
\end{aligned}$$

As such, we have the vector derivate as:

$$\frac{d}{d\boldsymbol{x}}\sigma(\boldsymbol{x}) = \sigma(\boldsymbol{x}) \circ [\mathbf{1} - \sigma(\boldsymbol{x})]$$

where $\circ$ represents element-wise vector product.

(e) We compute the requested derivatives for the negative sampling loss function. First, with respect to $\boldsymbol{u}_o$.

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{u}_o} &= -\frac{\partial}{\partial \boldsymbol{u}_o} \log \sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c) && \text{(Only the first term depends on } \boldsymbol{u}_o\text{)} \\
&= -\frac{\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)(1-\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c))}{\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)} \boldsymbol{v}_c && \text{(Derivative of } \log x \text{ and results from 1d)} \\
&= -(1-\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)) \boldsymbol{v}_c
\end{aligned}$$

Next, with respect to $\boldsymbol{u}_k$.

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{u}_k} &= -\frac{\partial}{\partial \boldsymbol{u}_k} \log \sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c) && \text{(Only the } k+1\text{-th term depends on } \boldsymbol{u}_k\text{)} \\
&= \frac{\sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)(1-\sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c))}{\sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)} \boldsymbol{v}_c && \text{(Derivative of } \log x \text{ and results from 1d)} \\
&= (1-\sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)) \boldsymbol{v}_c
\end{aligned}$$

And finally, with respect to $\boldsymbol{v}_c$.

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

Computing the naive-softmax requires iterating over the entire vocabulary, which can be extremely large, while the negative sampling loss requires considering only $K+1$ samples.

(f) We now compute the skip-gram loss function. We have:

i

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

ii

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

iii

$$\frac{\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \cdots, w_{t+m}, \boldsymbol{v}_c)}{\partial \boldsymbol{v}_w} = 0 \qquad (w \ne c)$$

Problem 2

(a) In 'word2vec.py'

(b) In 'sgd.py'

(c) Using 'run.py' we get Figure 1. The plot shows the projection of our learned word vectors into two dimensions.

We see a few clusters forming – such as adjectives ("amazing", "wonderful", "great", "boring") – which, not surprisingly, contain not only synonyms but also antonyms. We also see a few other interesting clusters, such as "queen" and "dumb" (surprising, and sexist) as well as "female" and "woman" (not surprising). Lastly, we see "hail" as a single, unclustered word, which is a bit surprising given what we would intuitively expect (to cluster with "snow", "rain" as a weather phenomena).

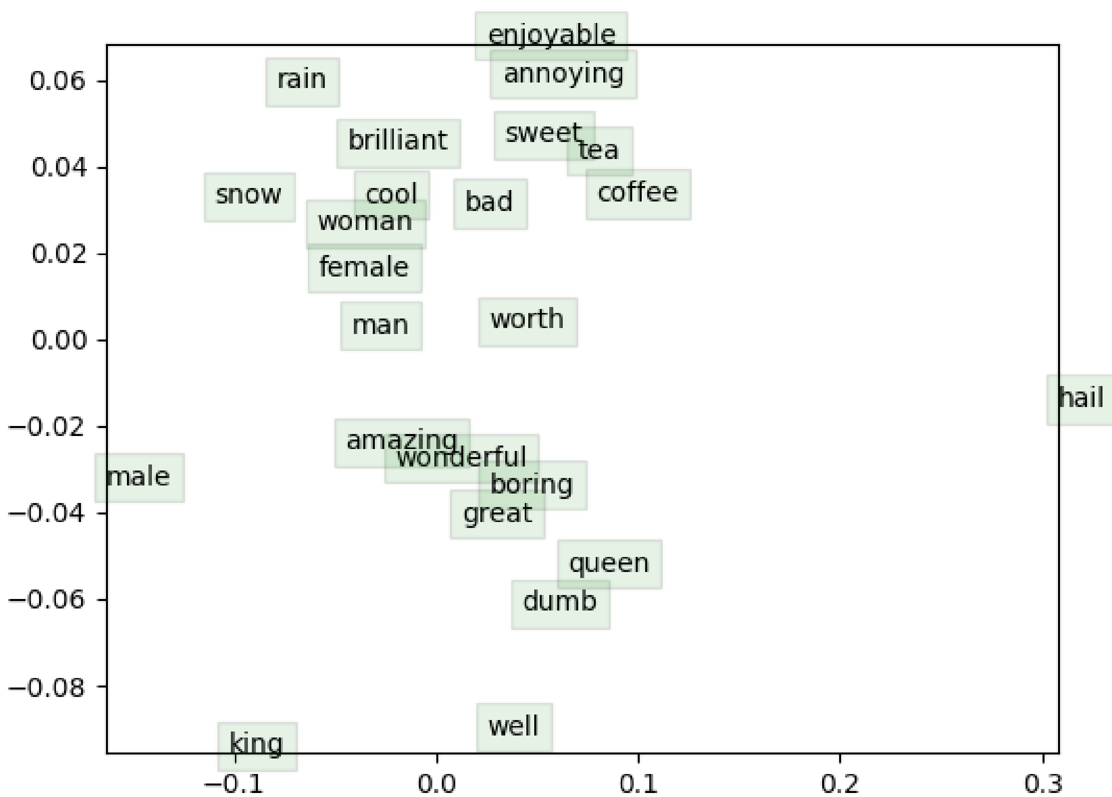

Figure 1: Projection of word2vec embeddings.