# Assignment 2 for CS224n

## (a)

Show that naive-softmax loss is the same as the cross-entropy loss between $y$ and $\hat{y}$, i.e. show that:

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

Sol: Because $\boldsymbol{y}$ is a one-hot vector with a 1 for the true outside word o, and 0 everywhere else. Thus the LHS is essentially $-(0log(\hat{y}_1) + \ldots + 1log(\hat{y}_o) + \ldots + 0log(\hat{y}_{|V|}) = -log(\hat{y}_o)))$.

## (b)

Compute the partial derivative $J_{naive-softmax}(uc, o, U) = -logP(O = o|C = c)$ w.r.t $v_c$.

Sol: It is defined that $-logP(O = o|C = c) = -log\frac{e^{u_o^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}}$ , thus we can compute that

$$\begin{aligned}
\frac{\partial J}{\partial v_c} &= -u_o + \frac{1}{\sum_{w \in Vocab} e^{u_w^T v_c}} \sum_{w \in Vocab} (e^{u_w^T v_c} u_w) \\
&= -u_o + \sum_{w \in Vocab} (\frac{e^{u_w^T v_c}}{\sum_{w \in Vocab} e^{u_w^T v_c}} u_w) \\
&= -u_o + \sum_{w \in Vocab} (P(u_w|v_c)u_w) \\
&= -u_o + \sum_{w \in Vocab} (\hat{y}_w u_w)
\end{aligned}$$

## (c)

Compute the partial derivative $J_{naive-softmax}(uc, o, U) = -logP(O = o|C = c)$ w.r.t $u_w$.

Sol:

**Case 1**: $w = o$. We have

$$\begin{aligned}
\frac{\partial J}{\partial u_{w=o}} &= -v_c + \frac{1}{\sum_{w \in Vocab} e^{u_w^T v_c}} (e^{u_o^T v_c} v_c) \\
&= v_c(\hat{y}_o - 1)
\end{aligned}$$

**Case 2**: $w \neq o$. We have

$$\begin{aligned}
\frac{\partial J}{\partial u_{w \neq o}} &= \frac{1}{\sum_{w \in Vocab} e^{u_w^T v_c}} (e^{u_{w \neq o}^T v_c} v_c) \\
&= v_c \hat{y}_{w \neq o}
\end{aligned}$$

## (d)

Compute the partial derivative $J_{naive-softmax}(uc, o, U) = -logP(O = o|C = c)$ w.r.t U.

Sol: It is straightforward to show that $\frac{\partial J}{\partial U} = [\frac{\partial J}{\partial u_1}, \ldots, \frac{\partial J}{\partial u_{|Vocab|}}]_{k \times |Vocab|}$, where k is the dimension of word vector.

## (e)

Compute the derivative of sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ w.r.t x, where x is a scalar.

Sol: It is straightforward to show that $\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{1+e^{-x}} = \sigma(x)(1-\sigma(x))$.

## (f)

Now consider Negative Sampling Loss that $J_{neg-sample}(v_c, o, U) = -log(\sigma(u_o^T v_c)) - \sum_{k=1}^{K} log(\sigma(-u_k^T v_c))$. Compute the partial derivative of it w.r.t $v_c, u_o, u_k$.

Sol: It can be showed that

$$\begin{aligned}
\frac{\partial J}{\partial v_c} &= -\frac{1}{\sigma(u_o^T v_c)}\sigma'(u_o^T v_c)u_o - \sum_k \frac{\sigma'(-u_k^T v_c)}{\sigma(-u_k^T v_c)}(-u_k) \\
&= (\sigma(u_o^T v_c) - 1)u_o + \sum_k (1 - \sigma(-u_k^T v_c))u_k \\
\frac{\partial J}{\partial u_o} &= (\sigma(u_o^T v_c) - 1)v_c \\
\frac{\partial J}{\partial u_k} &= (1 - \sigma(-u_k^T v_c)v_c
\end{aligned}$$

It can be seen that with Negative Sampling, it is much more efficient to compute the gradient, since we only need K samples (O(K)) while the naive softmax needs the whole vocab (O(|Vocab|)).

## (g)

Now consider Negative Sampling Loss that $J_{neg-sample}(v_c, o, U) = -log(\sigma(u_o^T v_c)) - \sum_{k=1}^{K} log(\sigma(-u_k^T v_c))$. Compute the partial derivative of it w.r.t $u_k$ without assumption that the K negative samples are distinct.

Sol:

$$\frac{\partial J}{\partial u_k} = \sum_{j=k} (1 - \sigma(-u_j^T v_c)v_c$$

## (h)

Compute the following three partial derivatives.

(i) $\partial J(v_c, w_{t-m}, \dots, w_{t+m}, U)/\partial U$.

Sol: $\partial J_{skip-gram}(v_c, w_{t-m}, \dots, w_{t+m}, U)/\partial U$ = $\sum -m \le j \le m, j \ne 0 J_{skip-gram}(v_c, w_j, U)/\partial U$.

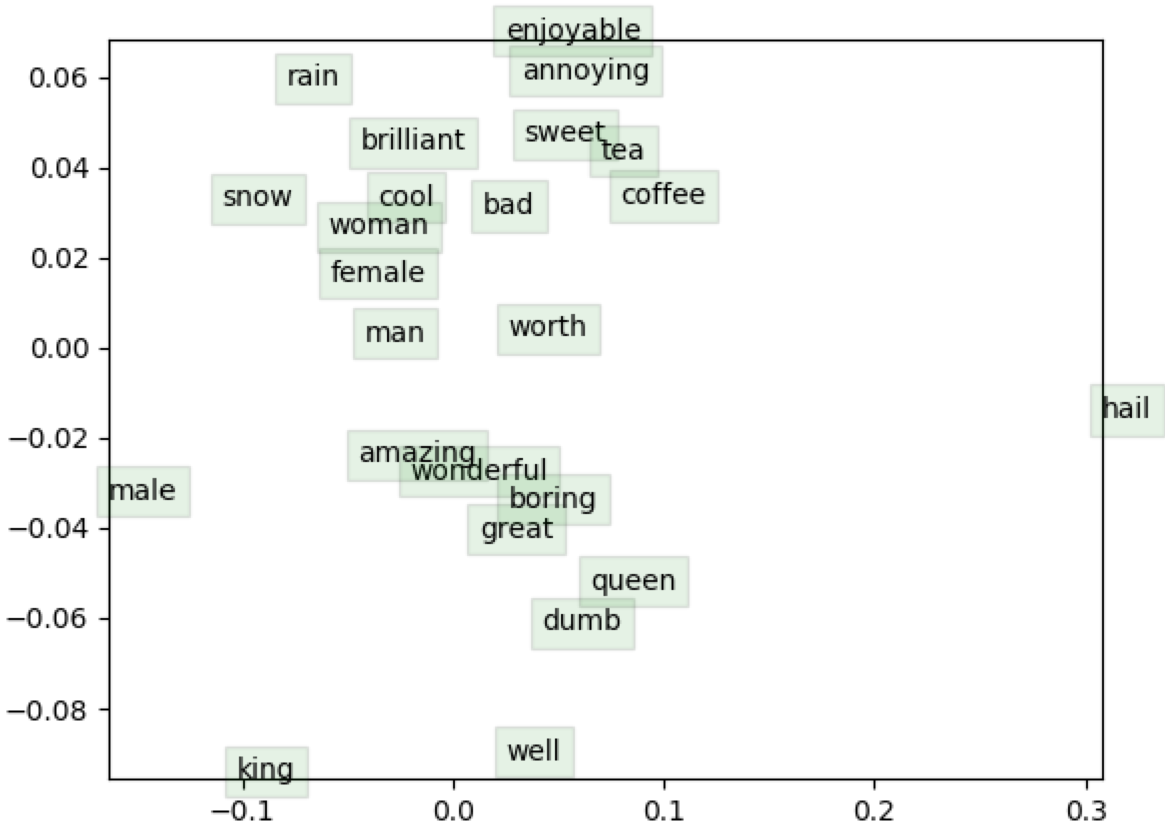
(ii)$\partial J(v_c, w_{t-m}, \dots, w_{t+m}, U)/\partial v_c$.

Sol: $\partial J(v_c, w_{t-m}, \dots, w_{t+m}, U)/\partial v_c$ = $\sum -m \le j \le m, j \ne 0 J_{skip-gram}(v_c, w_j, U)/\partial v_c$.

(iii) $\partial J(v_c, w_{t-m}, \dots, w_{t+m}, U)/\partial v_w$ when $w \ne c$.

Sol: 0.

## Code result



enjoyable
annoying
rain
brilliant
sweet
tea
snow
cool
bad
coffee
woman
female
man
worth
hail
amazing
wonderful
male
boring
great
queen
dumb
well
king
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.1
0.0
0.1
0.2
0.3