Assignment-2 | NLP

Q:-1 (a) :- given → $\text{softmax}(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ — (1)

R.H.S. = $\text{softmax}(x+c)$

$$= \frac{e^{(x_i+c)}}{\sum_j e^{(x_i+c)}} \quad (\text{from } -1)$$

$$= \frac{e^c \cdot e^{x_i}}{\sum_j e^{x_j} \cdot e^c}$$

$$= \frac{e^c \cdot e^{x_i}}{e^c \sum_j e^{x_j}} \quad (\because c = \text{constant})$$

$$= \frac{e^{x_i}}{\sum_j e^{x_j}}$$

$= \text{softmax}(x)$ = L.H.S.

Hence, it is proved that softmax is invariant to constant offset in the input.

Q:- 1(c) :- $\sigma(x) = \frac{1}{1+e^{-x}}$ — (1)

$\Rightarrow \sigma'(x) = \frac{\partial}{\partial x}\left(\frac{1}{1+e^{-x}}\right)$

$= \frac{(-1)}{(1+e^{-x})^2} \cdot \frac{\partial}{\partial(x)}(1+e^{-x})$

$= \left(\frac{-1}{(1+e^{-x})^2}\right) \cdot (0 - e^{-x})$

$= \frac{e^{-x}}{(1+e^{-x})^2}$

$= \frac{1}{e^x} \cdot \frac{e^{2x}}{(e^x+1)^2}$

$= \frac{e^x \cdot e^x}{e^x \cdot (e^x+1)^2}$

$= \left(\frac{e^x}{e^x+1}\right)\left(\frac{1}{e^x+1}\right)$

$= \left(\frac{1}{1+e^{-x}}\right)\left(1 - 1 + \frac{1}{e^x+1}\right)$

$$= \left(\frac{1}{1+e^{x}}\right)\left(1 - \left(\frac{e^{x}+1-1}{e^{x}+1}\right)\right)$$

$$= \left(\frac{1}{1+e^{x}}\right)\left(1 - \left(\frac{1}{1+e^{-x}}\right)\right)$$

$$= \boxed{\sigma(x)\,(1-\sigma(x))} \quad (\because \text{from } -(1))$$

Hence, ~~it~~ is proved!
given

Q-2(a): $\dfrac{\partial}{\partial v_c} J_{CE}(o, v_c, U) = ?$

» $J_{CE} = -\sum_i y_i \log(\hat{y}_i)$ — (1)

- It is given that y_i is one-hot encoded vector and value for $\hat{y}$ is also given,

Hence,

$$J_{CE} = -\log\left\{\frac{\exp(u_o^T v_c)}{\sum_w \exp(u_w^T v_c)}\right\}$$

$$= -\left[\log(\exp(u_o^T v_c)) - \log\left\{\sum_w \exp(u_w^T v_c)\right\}\right]$$

$$= -u_o^T v_c + \log \sum_w \exp(u_w^T v_c) \quad — (2)$$

$\therefore$ From — (2)

$$\frac{\partial}{\partial v_c} J_{CE} = \left(\frac{1}{\sum_w \exp(u_w^T v_c)} \right) \left(\exp(u_w^T v_c) \right) \cdot u_w^T - u_w^T$$

$$= \left(\frac{\exp(u_w^T v_c)}{\sum_w \exp(u_w^T v_c)} \right) u_w^T - u_w^T$$

$$= \sum_{j=1}^{W} u_j^T \left\{ \frac{\exp(u_j^T v_c)}{\sum_{i=1}^{W} \exp(u_i^T v_c)} - 1 \right\}$$

$$\boxed{\frac{\partial J_{CE}}{\partial v_c} = \sum_{j=1}^{W} u_j^T \left(P(u_j | v_c) - 1 \right)} \quad \text{— (3)}$$

$\therefore$ — (3) is the required solution

Q:- 2(b) :- $\frac{\partial}{\partial u_w} J_{CE} = ?$

- Same as q-2(a), we can derive,

$$J_{CE} = -u_w^T v_c + \log \sum_w \exp(u_o^T v_c) \quad \text{---(1)}$$

→ Now,

$$\frac{\partial J_{CE}}{\partial u_w} = -v_c + \left\{ \frac{1}{\sum_w \exp(u_w^T v_c)} \right\} \left(\exp(u_o^T v_c) \right) (v_c)$$

$$\boxed{\frac{\partial J_{CE}}{\partial u_w} = v_c \left\{ \frac{\exp(u_o^T v_c)}{\sum_{j=1}^{W} \exp(u_j^T v_c)} - 1 \right\}} \quad \text{---(4)}$$

Which is required derivation :

$$\boxed{\frac{\partial J_{CE}}{\partial u_w} = v_c \left\{ P(u_o | v_c) - 1 \right\}}$$

#1 #2

Q:- 2 (c) :- $\frac{\partial}{\partial v_c} J_{neg\text{-}sample} = ?$ | $\frac{\partial}{\partial u_w} J_{neg\text{-}sample} = ?$

⇒ Given,

$$J_{neg\text{-}sample} = -\log\left(\sigma(u_o^T v_c)\right) - \sum_{k=1}^{K} \log\left(\sigma(-u_k^T v_c)\right) \quad \text{---(1)}$$

→ $\therefore \frac{\partial}{\partial v_c} J_{neg\text{-}sample} = \frac{(-1)}{\sigma(u_o^T v_c)} \left\{\sigma(u_o^T v_c) \cdot (1 - \sigma(u_o^T v_c))\right\} \cdot u_o^T$

$$- \sum_{k=1}^{K} \left(\frac{1}{\sigma(-u_k^T v_c)}\right) \cdot \left(\sigma(-u_k^T v_c)\right) \cdot \left(1 - \sigma(-u_k^T v_c)\right) \cdot \cdot (-u_k^T)$$

$$= \left[u_o^T \cdot \left(\sigma(u_o^T v) - 1\right) - \sum_{k=1}^{K} u_k^T \left(\sigma(-u_k^T v_c) - 1\right)\right] \quad \text{---(2)}$$

which is sequised solution #1

⇒ $\therefore \frac{\partial}{\partial u_w} J_{neg\text{-}sample} = \left(\frac{-1}{\sigma(u_o^T v_c)}\right) \left\{\sigma(u_o^T v_c) \cdot (1 - \sigma(u_w^T v_c))\right\} \cdot v_c^T$

$= 0$

$(\because o \notin \{1, \ldots K\}$, hence 2nd-part derivative w.r.t. $u_w = 0)$

$$= \left[v_c^T \left(\sigma \left(u_w^T v_c \right) - 1 \right) \right] \quad \text{---(3)}$$

Which is required **Solution #2**

Q1-2(d):- For skip-gram, the cost for a context centered around c is:

$$J_{skip\text{-}gram} = \sum_{-m \le j \le m, j \neq 0} F(w_{c+j}, v_c) \quad \text{---(1)}$$

derivatives would be,

$$\therefore \frac{\partial J_{skip\text{-}gram}}{\partial w} = \left[\sum_{-m \le j \le m, j \neq 0} \frac{\partial F}{\partial w_{c+j}} \right] \quad \text{---(2)}$$

+

$$\frac{\partial J_{skip\text{-}gram}}{\partial v_c} = \left[\sum_{-m \le j \le m, j \neq 0} \frac{\partial F}{\partial v_c} \right] \quad \text{---(3)}$$

as required!

Nearest Neighbors:

Nearest Words to the word **great**:
great
juan
ringside
toast
kenneth

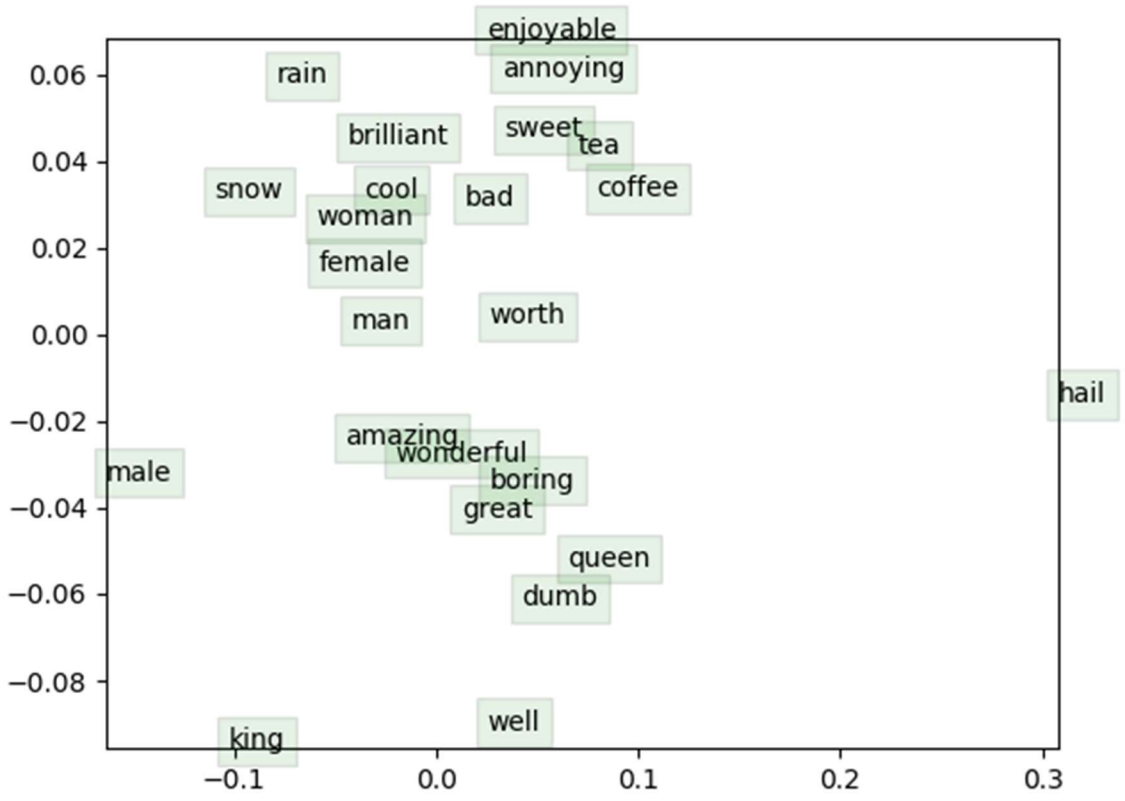
Nearest Words to the word **brilliant**:
stephen
brilliant
liberating
undermines
ugly-duckling

Nearest Words to the word **female**:
female
illuminates
dancing
fear-inducing
skateboarding

Nearest Words to the word **king**:
instinct
king
xtc
composition
lector

Nearest Words to the word **snow**:
snow
presents
savvy
compensate
squandering

Vectors plot:



Analysis:

- By comparing results from kNN and word vectors plot, we can see the drastic difference in word similarities resulted from both.
- By looking at 5 nearest words for the given word, we can see that kNN is only able to give same word as the closest similar word to given word itself. Other 4 words are not even close to being similar. The reason is the way kNN calculates similarity. In this case kNN only depends on cosine distance to calculate similarity, which doesn't take context of the word into account. It only tries to calculate distance based on simple dot product.
- On other hand, word vectors are designed to capture the context of the word given its neighboring words. As we can see in the plot, similar words are grouped together. For example: (Woman, female), (amazing, wonderful, great), (tea, coffee) etc.
- Only advantage of kNN is its speed. Because, word vectors heavily relay on many calculations to capture proper context of the word, word vectors are slow to calculate.