1.a) $\text{softmax}(x_i + c) = \frac{e^{x_i + c}}{\sum_i e^{x_i + c}} = \frac{e^c \cdot e^{x_i}}{e^c(\sum_i e^{x_i})} = \frac{e^{x_i}}{\sum_i e^{x_i}} = \text{softmax}(x_i)$

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

$$\frac{d[\sigma(x)]}{dx} = -1 \cdot \frac{(e^{-x})}{(1+e^{-x})^2} = \frac{e^{-x} \cdot 1}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) = \sigma(x) \cdot [1 - \sigma(x)]$$

c) i. $\hat{y}_i = P(i|c) = \frac{e^{u_i^T \cdot v_c}}{\sum_w e^{u_w^T \cdot v_c}}$ $\qquad \hat{Y}^T = [P(1|c) \;\; P(2|c) \cdots \underbrace{P(o|c)}_{\text{"}\hat{y}_o\text{"}} \cdots P(W|c)]$

From chain rules, we get $\frac{\partial J_{CE}}{\partial v_c} = -\sum_i y_i \cdot \frac{\partial \log \hat{y}_i}{\partial v_c} = -\sum_i y_i \frac{1}{\hat{y}_i} \cdot \left[\frac{\partial \hat{y}_i}{\partial z_w} \cdot \frac{\partial z_w}{\partial v_c}\right]$ *

$\{z_w = u_w^T \cdot v_c\}$

Because Y is a one-hot vector, and the derivatives of softmax have upper bound, Clearly, from *

$$\frac{\partial J_{CE}}{\partial v_c} = -y_o \cdot \frac{1}{\hat{y}_o} \cdot \frac{\partial \hat{y}_o}{\partial z_o} \cdot \frac{\partial z_o}{\partial v_c} + 0$$

$$= -y_o \cdot \frac{1}{\hat{y}_o} \cdot \hat{y}_o \cdot (1-\hat{y}_o) \cdot u_o^T$$

$$= \boxed{-(1-\hat{y}_o) \cdot u_o^T} \rightarrow (\hat{y}_o - 1) \cdot u_o^T$$

o is the one number's Index of the one hot vector Y
But o can consider as one of w

$$\frac{\partial \hat{y}_{w_o}}{\partial z_{w_o}} = \frac{d \frac{e^{z_{w_o}}}{e^{z_{w_o}} + b}}{d e^{z_{w_o}}} \qquad \boxed{b = \sum_{w \neq w_o} e^{z_w}}$$

from 1.b)

$$= \hat{y}_{w_o} [1 - \hat{y}_{w_o}]$$

ii. The same logic, but different last term,

$$\frac{\partial J_{CE}}{\partial u_w} = -y_o \cdot \frac{1}{\hat{y}_o} \cdot \frac{\partial \hat{y}_o}{\partial z_o} \cdot \frac{\partial z_o}{\partial u_w} + 0$$

$$= \boxed{-(1-\hat{y}_o) \cdot v_c^T}$$

$\Longrightarrow$ For $\forall w \in \{1, 2 \cdots W\}$, $\frac{\partial J_{CE}}{\partial u_w}$ is the same

$$\frac{\partial J_{CE}}{\partial U} = \begin{bmatrix} \leftarrow \frac{\partial J_{CE}}{\partial u_1} \rightarrow \\ \vdots \\ \leftarrow \frac{\partial J_{CE}}{\partial u_w} \rightarrow \\ \vdots \\ \leftarrow \frac{\partial J_{CE}}{\partial u_W} \rightarrow \end{bmatrix} = \begin{bmatrix} \leftarrow (\hat{y}_o - 1) \cdot v_c^T \rightarrow \\ \vdots \\ \vdots \\ \vdots \\ \leftarrow (\hat{y}_o - 1) \cdot v_c^T \rightarrow \end{bmatrix}$$

~~1.d~~

1.C) iii. For a given word o, $J_{neg}(o, v_c, U) = -\log(\hat{y}_o(u_o^T \cdot v_c)) - \sum_k \log(\hat{y}_k(-u_k v_c))$

$$\boxed{\begin{aligned}\hat{y}_o(u_o^T \cdot v_c) &= \sigma(u_o^T \cdot v_c)\\ \hat{y}_k(-u_k^T v_c) &= \sigma(-u_k^T v_c)\end{aligned}}$$

8 $$\frac{\partial \sum_k \log(\hat{y}_k(-u_k^T \cdot v_c))}{\partial v_c} = \sum_k \frac{1}{\hat{y}_k} \cdot \underbrace{\hat{y}_k \cdot (1-\hat{y}_k)}_{\text{from b.}} \cdot (-u_k)^T$$

$$= \sum_k (1-\hat{y}_k)(-u_k^T)$$

11 Therefore, 一次 center word $$\frac{\partial J_{neg}}{\partial v_c} = (\hat{y}_o - 1)u_o^T - \sum_k (\hat{y}_k - 1) \cdot u_k^T$$

$$= [\sigma(u_o^T \cdot v_c) - 1] \cdot u_o^T - \sum_k [\sigma(-u_k^T \cdot v_c) - 1] u_k^T \quad (\forall o, k \quad o \neq k)$$

from b&i.

$\Rightarrow$ (dot合并) !

18 $$\frac{\partial J_{neg}}{\partial u_o} = (\hat{y}_o - 1) v_c^T + 0 \leftarrow \left\{ \forall o, o \neq k, \ \frac{\partial(-u_k^T \cdot v_c)}{\partial u_o} = 0. \right.$$

$$\frac{\partial J_{neg}}{\partial u_w} = 0 \qquad \left\{ \forall\, w \neq o \text{ and } w \neq k, \ \frac{\partial(\pm u_o^T \cdot v_c)}{\partial u_w} = \frac{\partial(-u_k^T \cdot v_c)}{\partial u_w} = 0 \right.$$

25 $$\frac{\partial J_{neg}}{\partial u_k} = 0 - (\hat{y}_k - 1) \cdot v_c^T \qquad \left\{ \forall k, k \neq o, \ \frac{\partial(-u_o^T \cdot v_c)}{\partial u_k} = 0. \right.$$

When implement, 1 can be substituted by y_i which is not equal 0

1.C).iv. Needless to say, we can treat $F(W_{c+j}, V_c)$ as follows:

$$F(W_{c+j}, V_c, Neg(W_{c+j})) = -\log\left(\sigma\left[\left(U^{(\text{obs})}_{w_{c+j}}\right)^T \cdot V_c\right]\right) - \sum_{k} \log\left[\sigma(-U_k^T \cdot V_c)\right]$$

$$k \in Neg(W_{c+j})$$

For one center word (V_c), we can have multiples training samples, which is defined by m.

The same, $Neg(w_{c+j})$ is defined previously, as a hyper-parameters. the number of it is treated.

Therefore, from the answer before, we can derivate the Cross Entropy Loss for one center word, which is used to update V_k, U_k respectively.

$$\frac{d \sum_{j\neq 0}^{[m,m]} F(W_{c+j}, V_c, Neg(W_{c+j}))}{d V_c} = \sum_{j\neq 0}^{[-m,m]} \left\{ \left[\sigma(U_{w_{c+j}}^T \cdot V_c) - 1\right] \cdot U_{w_{c+j}}^T - \sum_k \left[\sigma(-U_k V_c) - 1\right] U_k^T \right\}$$

The summation of 2m vectors (grad).

$W \times$ length of word vector

$$\frac{d \sum_{j\neq 0}^{[-m,m]} \cdot F[W_{c+j}, V_c, Neg(W_{c+j})]}{d U_w}$$

shape ⇒ [ ]

$$= \begin{cases} \sum_{j\neq 0}^{[-m,m]} (\hat{y}_{c+j} - 1) \cdot V_c^T & w \in \{W_{c+j}\}, \ \forall j \in [-m,-1] \cup [1,m] \ \& \ j \in N \\ \sum_{j\neq 0}^{[-m,m]} \cdot [-(\hat{y}_k - 1)] & w \in \{Neg(W_{c+j})\} \ \& \ w \notin W_{c+j}, \ \forall j \in [-m,-1] \cup [1,m] \ \& \ j \in N \\ \text{It can separate, and update separately, rather than update in a whole.} & W \in \{W_{c+j}\} \cap \{Neg(W_{c+j})\} \\ 0 & \text{else} \end{cases}$$

CS584: Natural Language Processing - HW2 Name: YANG, Wei

Report

Hyperparameters:

- Embedding dimension = 10
- Context size = 5
- GD Step size = 0.3
- Anneal rate = 0.5 for every 20000 iterations
- Initialized Input – WV with uniform: (-0.5, 0.5)/ Embedding dimension

Sample info

- # words = 19539

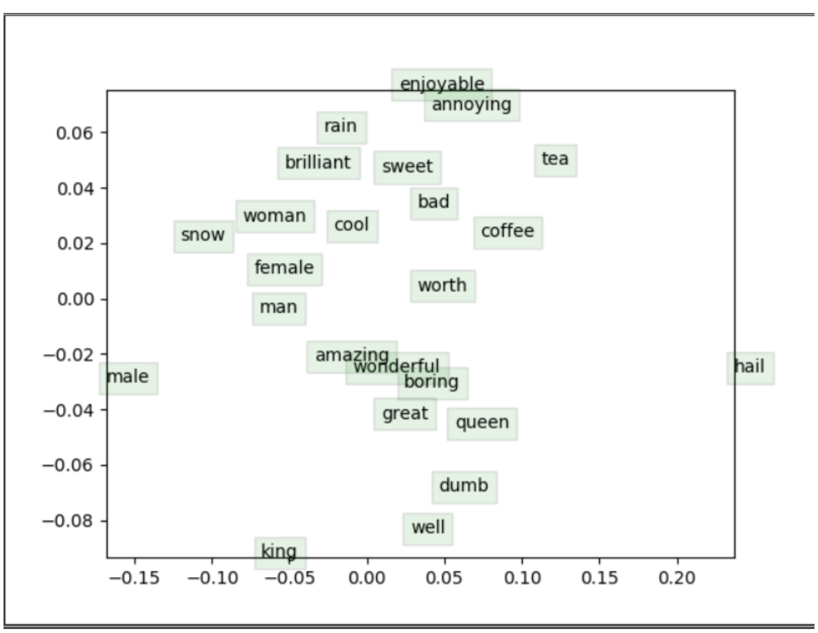
Training time: (40000 iterations)

- 11288 seconds

Future Improvement:

- Preprocessing: Stop words should be deleted
- High frequency word pairs could be treated as one single word
- Increase the embedding dimension
- Stop the random seeds
- Regularization: Unknown
- Learning rate: unknown

The graph tells us the word vectors do capture the features between different words. For example, the vector of "female-male" is similar to "king-queen". Although the "man-woman" vector doesn't show the same pattern, it may be trigger by dimension reduction, which loss information. Or maybe they will show similarity on a higher dimension graph.



The output of KNN show us the most similar 5 words (for the neat format), the self is excluded. From this result, we can hardly find some common underlying characteristics between them apparently. As the 2D graph above, the similar vectors may have stayed together, but it's not distinguished to separate different concepts. However, after we put those words in the original sentences, we may observe that they share some common sentence structures.

I only use the input-W vector to generate similar words below, which doesn't contain all the information of words. Also, with the more training data, the accuracy of the distribution representation of those words may increase.

```
great      : kenneth          |fun-for-fun     |juan            |toast           |pianist        |
cool       : fast-edit        |intelligence    |right           |determined      |schmaltzy      |
brilliant  : undermines       |shared          |stephen         |ugly-duckling   |dignified      |
wonderful  : inclination      |gag             |unforced        |disappointing   |repulsive      |
well       : labored          |entirety        |countenance     |operational     |salacious      |
amazing    : cultist          |madness         |him             |overwhelmingly  |people         |
worth      : devolves         |creatively      |dilbert         |fiddle          |dating         |
sweet      : ambiguous        |delusional      |helps           |on-screen       |wrap           |
enjoyable  : whiney           |flamboyant      |impressed       |reef            |spit           |
boring     : six              |road            |xerox           |symbols         |definition     |
bad        : bang-up          |any             |skeleton        |dragon          |frozen         |
dumb       : overinflated     |bonanza         |notably         |rice            |foreboding     |
annoying   : haynes           |titled          |portrayed       |call            |impressions    |
female     : illuminates      |relevant        |clear           |dogs            |hand           |
male       : chaplin          |scripted        |hype            |racism          |josh           |
queen      : unqualified      |wanes           |chew            |cornball        |aggrieved      |
king       : lector           |composition     |xtc             |changed         |submarine      |
man        : subjects         |manifesto       |inexorably      |craven          |below          |
woman      : wears            |zen             |impersonal      |paulette        |autobiographical|
rain       : bartlett         |cagney          |diversions      |perilously      |oftentimes     |
snow       : compensate       |snatch          |savvy           |free            |bluffs         |
hail       : skullduggery     |riveted         |fish            |scum            |delves         |
coffee     : sinner           |demographic     |developments    |nonconformity   |amazingly      |
tea        : bringing         |homiletic       |scarlet         |banging         |detached       |
```

I uploaded the model's parameters training with 40000 iterations.

BTW, for HW1, the result is not converged, which may result from the regularization term, which is too large (I set with 0.02 before). If the framework of coding is given, it will be more time-saving. Thanks for kindly giving opportunity on the first HW, I didn't expect the difficulties and time spent on assignment for the first time.