

CS 224n Assignment 2: word2vec

Chrysa Dikonimaki

August 2020

1 Written: Understanding word2vec

1.a

Since y_w is 1 only for one word (the word that we represent as o):

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

1.b

$$\hat{y}_o = P(O = o | C = c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in Vocab} \exp(u_w^T v_c)}$$

so :

$$\begin{aligned} J_{naive-softmax} &= -\log(\hat{y}_o) \\ &= -\log \frac{\exp(u_o^T v_c)}{\sum_{w \in Vocab} \exp(u_w^T v_c)} \\ &= -u_o^T v_c + \log \sum_{w \in Vocab} \exp(u_w^T v_c) \end{aligned}$$

Thus:

$$\begin{aligned} \frac{\partial J_{naive-softmax}}{\partial v_c} &= -u_o^T + \frac{\sum_{w \in Vocab} \exp(u_w^T v_c) * u_w^T}{\sum_{w \in Vocab} \exp(u_w^T v_c)} \\ &= -u_o + \sum_{w \in Vocab} P(O = w | C = c) * u_w \\ &= -u_o + U[P(O = w_1 | C = c), P(O = w_2 | C = c), \dots] \\ &= -U * y + U * \hat{y} \\ &= U(\hat{y} - y) \end{aligned}$$

1.c

1st case: $w=o$

$$\begin{aligned}\frac{\partial J_{naive-softmax}}{\partial u_o} &= -v_c + \frac{\exp(u_o^T v_c) * v_c}{\sum_{w \in V_{ocab}} \exp(u_w^T v_c)} \\ &= -v_c + y^T \hat{y} v_c \\ &= v_c (y^T \hat{y} - 1)\end{aligned}$$

2nd case: $w \neq o$

$$\begin{aligned}\frac{\partial J_{naive-softmax}}{\partial u_w} &= -0 + \frac{\exp(u_w^T v_c) * v_c}{\sum_{w \in V_{ocab}} \exp(u_w^T v_c)} \\ &= \hat{y}_w v_c\end{aligned}$$

1.d

$$\begin{aligned}(\sigma(x))' &= \left(\frac{1}{1+e^{-x}}\right)' \\ &= \frac{1}{(1+e^{-x})^2} (1+e^{-x})' \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{e^{-x}}{(1+e^{-x})} \frac{1}{(1+e^{-x})} \\ &= \sigma(-x) * \sigma(x) \\ &= \frac{e^{-x} + 1 - 1}{(1+e^{-x})} \frac{1}{(1+e^{-x})} \\ &= \left(1 - \frac{1}{(1+e^{-x})}\right) \frac{1}{(1+e^{-x})} \\ &= (1 - \sigma(x))\sigma(x)\end{aligned}$$

1.e

$$\begin{aligned}
\frac{\partial J_{neg-sample}}{\partial v_c} &= -\frac{\sigma(-u_o^T v_c)\sigma(u_o^T v_c) * u_o^T}{\sigma(u_o^T v_c)} - \sum_{k=1}^K \frac{\sigma(u_k^T v_c)\sigma(-u_k^T v_c)(-u_k^T)}{\sigma(-u_k^T v_c)} \\
&= -\sigma(-u_o^T v_c) * u_o^T - \sum_{k=1}^K \sigma(u_k^T v_c)(-u_k^T) \\
&= -\sigma(-u_o^T v_c) * u_o^T + \sum_{k=1}^K \sigma(u_k^T v_c)u_k^T \\
&= -\sigma(-u_o^T v_c) * u_o + \sum_{k=1}^K \sigma(u_k^T v_c)u_k
\end{aligned}$$

$$\begin{aligned}
\frac{\partial J_{neg-sample}}{\partial u_o} &= -\frac{\sigma(-u_o^T v_c)\sigma(u_o^T v_c) * v_c}{\sigma(u_o^T v_c)} - 0 \\
&= -\sigma(-u_o^T v_c)v_c
\end{aligned}$$

$$\begin{aligned}
\frac{\partial J_{neg-sample}}{\partial u_k} &= -0 - \frac{\sigma(u_k^T v_c)\sigma(-u_k^T v_c)(-v_c)}{\sigma(-u_k^T v_c)} \\
&= \sigma(u_k^T v_c)v_c
\end{aligned}$$

1.f

$$\frac{\partial J_{skip-gram}}{\partial U} = \sum_{-m \leq j \leq m, j \neq 0} \frac{\partial J}{\partial U}$$

$$\frac{\partial J_{skip-gram}}{\partial v_c} = \sum_{-m \leq j \leq m, j \neq 0} \frac{\partial J}{\partial v_c}$$

$$\frac{\partial J_{skip-gram}}{\partial v_c} = 0$$

2 Coding: Implementing word2vec

```
iter 39600: 9.269859
iter 39610: 9.288015
iter 39620: 9.279245
iter 39630: 9.248681
iter 39640: 9.243792
iter 39650: 9.262415
iter 39660: 9.286713
iter 39670: 9.303074
iter 39680: 9.347061
iter 39690: 9.351811
iter 39700: 9.346526
iter 39710: 9.348751
iter 39720: 9.376277
iter 39730: 9.377825
iter 39740: 9.403598
iter 39750: 9.449631
iter 39760: 9.411052
iter 39770: 9.478179
iter 39780: 9.515252
iter 39790: 9.602017
iter 39800: 9.575306
iter 39810: 9.535311
iter 39820: 9.545193
iter 39830: 9.634330
iter 39840: 9.667367
iter 39850: 9.726079
iter 39860: 9.773566
iter 39870: 9.726360
iter 39880: 9.816947
iter 39890: 9.796976
iter 39900: 9.756038
iter 39910: 9.709420
iter 39920: 9.708453
iter 39930: 9.681717
iter 39940: 9.611494
iter 39950: 9.629689
iter 39960: 9.717083
iter 39970: 9.776979
iter 39980: 9.813174
iter 39990: 9.854022
iter 40000: 9.812206
sanity check: cost at convergence should be around or below 10
training took 4288 seconds
```

