

Assignment 2

Nikolaos Smyrnioudis

August 26, 2020

1 Written

1.1 a

Since the true distribution y_w is a one hot encoding of the desired outside word, $y_w = 1$ iff $w = o$

1.2 b

Compute the partial derivative of $J_{naive-softmax}(v_c, o, U)$ with respect to v_c . Please write your answer in terms of y , \hat{y} , and U

$$J_{naive_softmax}(v_c, o, U) = -\log(P(O = o|C = c))$$

$$J_{naive_softmax}(v_c, o, U) = -(u_o^T v_c) + \log\left(\sum_{w \in Vocab} \exp(u_w^T v_c)\right)$$

$$\frac{\partial J}{\partial v_c} = -u_o + \frac{1}{\sum_{w \in Vocab} \exp(u_w^T v_c)} * \sum_{w \in Vocab} (\exp(u_w^T v_c) * u_w)$$

$$= -u_o + \sum_{w \in Vocab} \frac{(\exp(u_w^T v_c) * u_w)}{\sum_{w \in Vocab} \exp(u_w^T v_c)}$$

$$= -u_o + \sum_{w \in Vocab} (P(O = w|C = c) * u_w)$$

$$= -U * y + U * \hat{y}$$

$$= U * (\hat{y} - y)$$

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

1.3 c

Compute the partial derivative of $J_{naive-softmax}(v_c, o, U)$ with respect to each of the ‘outside’ word vectors, u_w ’s. There will be two cases: when $w = o$, the true ‘outside’

word vector, and $w \neq o$, for all other words. Please write your answer in terms of y , \hat{y} , and v_c .

$$J_{naive_softmax}(v_c, o, U) = -\log(P(O = o|C = c))$$

$$J_{naive_softmax}(v_c, o, U) = -(u_o^T v_c) + \log\left(\sum_{w \in Vocab} \exp(u_w^T v_c)\right)$$

Case when $w = o$

$$\frac{\partial J}{\partial u_w} = -v_c + \frac{1}{\sum_{w \in Vocab} \exp(u_w^T v_c)} * \exp(u_o^T v_c) * v_c$$

$$\frac{\partial J}{\partial u_w} = v_c * \left(\frac{1}{\sum_{w \in Vocab} \exp(u_w^T v_c)} * \exp(u_o^T v_c) - 1\right)$$

$$\frac{\partial J}{\partial u_w} = v_c * (P(O = o|C = c) - 1)$$

$$\frac{\partial J}{\partial u_w} = v_c * (y^T \hat{y} - 1)$$

Case when $w \neq o$

$$\frac{\partial J}{\partial u_w} = \frac{1}{\sum_{w \in Vocab} \exp(u_w^T v_c)} * \exp(u_w^T v_c) * v_c$$

$$\frac{\partial J}{\partial u_w} = P(O = w|C = c) * v_c$$

$$\frac{\partial J}{\partial u_w} = \hat{y}_w * v_c$$

1.4 d

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

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

$$\frac{d\sigma}{dx} = \frac{e^x}{(e^x + 1)^2}$$

$$\frac{d\sigma}{dx} = \frac{1}{e^x + 1} \frac{e^x}{e^x + 1}$$

$$\frac{d\sigma}{dx} = \sigma(-x)\sigma(x)$$

$$\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$$

1.5 e

$$\begin{aligned}
J_{neg_sample}(v_c, o, U) &= -\log(\sigma(u_o^T v_c)) - \sum_{k=1}^K \log(\sigma(-u_k^T v_c)) \\
\frac{\partial J}{\partial v_c} &= -\frac{1}{\sigma(u_o^T v_c)} * \sigma(u_o^T v_c) \sigma(-u_o^T v_c) * u_o \\
&\quad + \sum_{k=1}^K \frac{1}{\sigma(-u_k^T v_c)} * \sigma(-u_k^T v_c) * \sigma(u_k^T v_c) u_k \\
&= -\sigma(-u_o^T v_c) * u_o + \sum_{k=1}^K \sigma(u_k^T v_c) u_k \\
\frac{\partial J}{\partial u_o} &= -\frac{1}{\sigma(u_o^T v_c)} * \sigma(u_o^T v_c) \sigma(-u_o^T v_c) * v_c \\
&= -\sigma(-u_o^T v_c) * v_c \\
\frac{\partial J}{\partial u_k} &= \sigma(u_k^T v_c) v_c
\end{aligned}$$

1.6 f

$$J_{skip-gram}(u_c, w_{t-m}, \dots, w_{t+m}, U) = \sum_{-m \leq j \leq m; j \neq 0} J(v_c, w_{t+j}, U)$$

1.6.1 i

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

1.6.2 ii

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

1.6.3 iii

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

2 Results

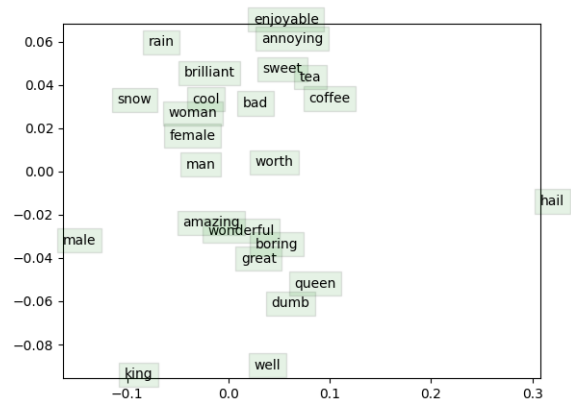


Figure 1: A plot of the resulting word vectors

```
Activities Terminal 25 Aug 10:45
sm@ideapad-sm: ~/stanford/assignment-2/a2$
iter 39590: 9.298053
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: 9.280713
iter 39670: 9.383074
iter 39680: 9.347061
iter 39690: 9.351811
iter 39700: 9.340526
iter 39710: 9.348751
iter 39720: 9.376277
iter 39730: 9.377825
iter 39740: 9.483598
iter 39750: 9.449631
iter 39760: 9.411052
iter 39770: 9.470179
iter 39780: 9.515252
iter 39790: 9.602017
iter 39800: 9.575306
iter 39810: 9.555311
iter 39820: 9.545193
iter 39830: 9.634330
iter 39840: 9.667367
iter 39850: 9.728679
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: 9.717683
iter 39970: 9.776979
iter 39980: 9.813174
iter 39990: 9.854022
iter 40000: 9.812286
smity checks: cost at convergence should be around or below 10
training took 4531 seconds
sm@ideapad-sm: ~/stanford/assignment-2/a2$
```

Figure 2: The time needed to train and output loss