April 4, 2017

Daniel Solomon
ID: 206335283

Gal Ron
ID: 304856529

Advanced Methods in NLP - Assignment 1: Word Vectors

Question 1(a)

We remember that:

$$softmax(\mathbf{x})_i = \frac{e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j}}$$

We note that:

$$softmax(\mathbf{x}+c)_i = \frac{e^{(\mathbf{x}_i+c)}}{\sum_j e^{(\mathbf{x}_j+c)}} = \frac{e^c e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j} e^c} = \frac{e^c e^{\mathbf{x}_i}}{e^c \sum_j e^{\mathbf{x}_j}} = \frac{e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j}} = softmax(\mathbf{x})_i$$

And so we get:

$$softmax(\mathbf{x}) = softmax(\mathbf{x}+c)$$

We can use this property to optimize *softmax* by choosing $c = -max_i x_i$.

Question 1(c)

The sigmoid function is: $\sigma(x) = \frac{1}{1+e^{-x}}$

Let's consider: $f(x) = \frac{1}{\sigma(x)} = 1 + e^{-x}$.

On the one hand, $\frac{d}{dx}f(x) = \frac{d}{dx}\sigma(x)^{-1} = -\frac{d}{dx}\sigma(x)\cdot\sigma(x)^{-2}$

On the other hand, $\frac{d}{dx}f(x) = \frac{d}{dx}\left(1+e^{-x}\right) = -e^{-x} = 1 - f(x) = 1 - \frac{1}{\sigma(x)} = \frac{\sigma(x)-1}{\sigma(x)}$

So we get that:

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

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

So the gradients of the sigmoid function can be written as a function of the function value:

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

Question 2(a)

$$\hat{\boldsymbol{y}}_{\boldsymbol{o}} = p(o \mid c) = \frac{\exp\left(\boldsymbol{u}_{\boldsymbol{o}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right)}{\sum_{w=1}^{W} \exp\left(\boldsymbol{u}_{\boldsymbol{w}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right)}$$

$$J_{softmax-CE}(o, v_c, U) = CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i} y_i \cdot \log\left(\hat{y}_i\right)$$

$$\frac{\partial J}{\partial \boldsymbol{v}_c} = -\boldsymbol{u}_{\boldsymbol{i}} + \sum_{w=1}^{W} \hat{y}_w \cdot \boldsymbol{u}_{\boldsymbol{w}}$$

Question 2(b)

$$\frac{\partial J}{\partial \boldsymbol{u}_w} = \begin{cases} \left(\hat{y}_w - 1\right)\boldsymbol{v}_{\boldsymbol{c}}, w = o \\ \hat{y}_w \boldsymbol{v}_{\boldsymbol{c}} \quad , otherwise \end{cases}$$

Question 2(c)

The sigmoid function is: $\sigma(x) = \frac{1}{1+e^{-x}}$

$$J_{neg-sample}(o, v_c, U) = -\log\left(\sigma\left(\boldsymbol{u}_{\boldsymbol{o}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right)\right) - \sum_{k=1}^{K} \log\left(\sigma\left(-\boldsymbol{u}_{\boldsymbol{k}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right)\right)$$

$$\frac{\partial J}{\partial \boldsymbol{v}_c} = \left(\sigma\left(\boldsymbol{u}_{\boldsymbol{o}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right) - 1\right)\boldsymbol{u}_o - \sum_{k=1}^{K} \left(\sigma\left(-\boldsymbol{u}_{\boldsymbol{k}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right) - 1\right)\boldsymbol{u}_{\boldsymbol{k}}$$

$$\frac{\partial J}{\partial \boldsymbol{u}_o} = \left(\sigma\left(\boldsymbol{u}_{\boldsymbol{o}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right) - 1\right)\boldsymbol{v}_{\boldsymbol{c}}$$

$$for\ k \in [K]: \frac{\partial J}{\partial \boldsymbol{u}_k} = -\left(\sigma\left(-\boldsymbol{u}_{\boldsymbol{k}}^{\top}\boldsymbol{v}_{\boldsymbol{c}}\right) - 1\right)\boldsymbol{v}_{\boldsymbol{c}}$$

Question 2(d)

If we use $F(o, \boldsymbol{v}_c)$ for $J_{softmax-CE}(o, v_c, \ldots)$ or $J_{neg-sample}(o, v_c, \ldots)$, the cost for a context centered around c is:

$$J = \sum_{-m \le j \le m, j \ne 0} F\left(\boldsymbol{w}_{\boldsymbol{c+j}}, \boldsymbol{v}_{\boldsymbol{c}}\right)$$

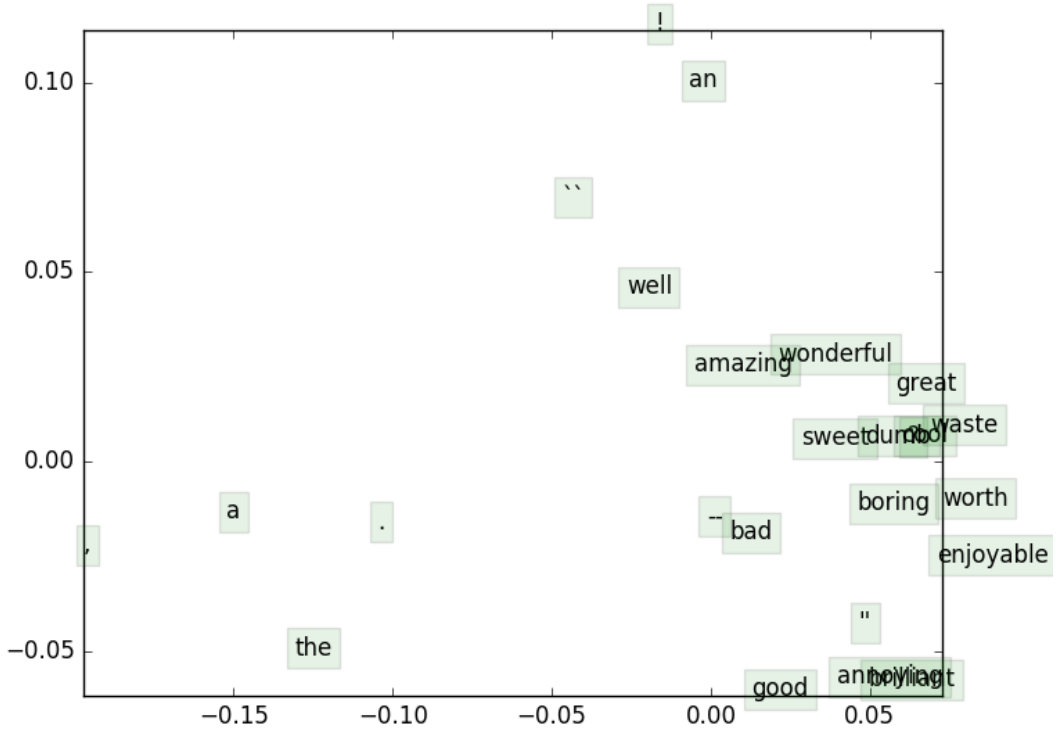
We denote the set of all output vectors $\boldsymbol{u}_{\boldsymbol{k}}$ as $\boldsymbol{U}$.

$$\frac{\partial J}{\partial \boldsymbol{U}} = \sum_{-m \le j \le m, j \ne 0} \frac{\partial F\left(\boldsymbol{w}_{\boldsymbol{c+j}}, \boldsymbol{v}_{\boldsymbol{c}}\right)}{\partial \boldsymbol{U}}$$

$$\frac{\partial J}{\partial \boldsymbol{v_c}} = \sum_{-m \le j \le m, j \neq 0} \frac{\partial F(\boldsymbol{w_{c+j}}, \boldsymbol{v_c})}{\partial \boldsymbol{v_c}}$$

and $\frac{\partial J}{\partial v_j} = 0$ if $j \neq c$.

Question 2(h)



KNN:

Words related to "the":
['comedy\\/thriller', 'a', 'if', 'that', 'or', 'the', '.', 'is', 'derek', 'bolt', 'decide']

Words related to "unique":
['unique', '1979', 'dares', 'realized', 'succumb', 'chabrolian', 'ba', 'puns', 'regardless', 'imaginative', 'lunar']

Words related to "superb":
['best', 'pool', 'industry', 'mine', 'gold', 'ghoulish', 'zingers', 'superb', 'moppets', 'roussillon', 'transporter']

Words related to "comedy":
['cute', 'longest', 'comedy', 'sensation', 'fast', 'cleaving', 'mature', 'first-timer', 'singing', 'observation', 'often-funny']

Words related to "surprisingly":
['hundred', 'surprisingly', 'philandering', 'either', 'protective', '20-car', 'unusually', 'bollywood', 'dogs', 'thinking', 'soderbergh']

- Notice that we implemented knn without full sorting (using argpartition for performance) therefore the related words list order is not by closeness.
  We can see that a lot of syntactic related words in the related words list and sometimes we see even semantic related words (‘comedy’ – ‘often funny’, ‘superb’ – ‘best’ and so on).
- We can see that in the image there are some close words with similar meaning such as ‘wonderful’ and ‘great’, and close words with similar connotation such as ‘bad’, ‘boring’ and ‘worth’ (negative connotation). Still we can observe that words which are not semantically related such as ‘waste’ and ‘great’ are still close in the image, but they are indeed syntactically close (both can be interpreted as adjectives).
  So, bottom line word2vec words embedding shows syntactic and semantic closeness for some words, but not for all of them (might perform better with larger corpus).

[4]