# ADVANCED METHODS IN NLP
# EXERCISE #1 SOLUTION

Uri Avron [uriavron@gmail.com] [308046994]
Ofri Kleinfeld [ofrik@mail.tau.ac.il] [302893680]
Ido Calman [calman.ido@gmail.com] [308353499]

March 27, 2018

# 1 Basics

**(a)** Indeed, for every dimension $i$:

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

which implies that $softmax(\mathbf{x}+c) = softmax(\mathbf{x})$. □

**(c)** Let us compute the gradient:

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

# 2 Word2vec

**(a)** Note that $y$ is an one-hot vector and therefore $\mathrm{CE}(y, \hat{y})$ is dependant only in $\hat{y}_o = p(o|c)$, so we get:

$$\mathrm{CE}(y,\hat{y}) = -\sum_o y_o \cdot \log(\hat{y}_o) = -\log(\hat{y}_o) = -u_o^T v_c + \log\left(\sum_{w=1}^{W} \exp(u_w^T v_c)\right)$$

Now we derive:

$$\begin{aligned}
\frac{\partial \mathrm{J}_{\mathrm{softmaxCE}}}{\partial v_c} &= \frac{\partial \left[-u_o^T v_c + \log\left(\sum_{w=1}^{W} \exp(u_w^T v_c)\right)\right]}{\partial v_c} \\
&= -u_o + \left(\sum_{w=1}^{W} \exp(u_w^T v_c)\right)^{-1} \sum_{w=1}^{W} \exp(u_w^T v_c) \cdot u_w \\
&= -u_o + \sum_{w=1}^{W} \underbrace{\frac{\exp(u_w^T v_c)}{\sum_{j=1}^{W} \exp(u_w^T v_c)}}_{=p(w|c)} \cdot u_w \\
&= -u_o + \sum_{w=1}^{W} p(w|c) \cdot u_w
\end{aligned}$$

**(b)** As in (a):

$$\begin{aligned}\frac{\partial \mathrm{J}_{\mathrm{softmaxCE}}}{\partial u_o} &= -v_c + \left(\sum_{w=1}^{W} \exp(u_w^T v_c)\right)^{-1} \exp(u_o^T v_c) \cdot v_c \\ &= -v_c + \underbrace{\frac{\exp(u_o^T v_c)}{\sum_{j=1}^{W} \exp(u_w^T v_c)}}_{=p(o|c)} \cdot v_c \\ &= -v_c \left(1 - p(o|c)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathrm{J}_{\mathrm{softmaxCE}}}{\partial u_{w\neq o}} &= \left(\sum_{j=1}^{W} \exp(u_j^T v_c)\right)^{-1} \exp(u_w^T v_c) \cdot v_c \\ &= -v_c \cdot p(w|c)\end{aligned}$$

**(c)** Using 1(c) and the fact that:

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

we get:

$$\begin{aligned}\frac{\partial \mathrm{J}_{\mathrm{negsample}}}{\partial v_c} &= \frac{\partial\left(-\log\sigma(u_o^T v_c) - \sum_{k=1}^{K} \log\sigma(-u_k^T v_c)\right)}{\partial v_c} \\ &= -\sigma^{-1}(u_o^T v_c) \cdot \frac{\partial\sigma(u_o^T v_c)}{\partial v_c} - \sum_{k=1}^{K} \sigma^{-1}(-u_k^T v_c) \cdot \frac{\partial\sigma(-u_k^T v_c)}{\partial v_c} \\ &= -\sigma^{-1}(u_o^T v_c) \cdot \sigma(u_o^T v_c) \cdot \sigma(-u_o^T v_c) \cdot u_o \\ &- \sum_{k=1}^{K} \sigma^{-1}(-u_k^T v_c) \cdot \sigma(-u_k^T v_c) \cdot \sigma(u_k^T v_c) \cdot (-u_k) \\ &= -\sigma(-u_o^T v_c) \cdot u_o + \sum_{k=1}^{K} \sigma(u_k^T v_c) \cdot u_k\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathrm{J}_{\mathrm{negsample}}}{\partial u_o} &= \frac{\partial \left( -\log \sigma(u_o^T v_c) - \sum_{k=1}^{K} \log \sigma(-u_k^T v_c) \right)}{\partial u_o} \\
&= -\sigma^{-1}(u_o^T v_c) \cdot \frac{\partial \sigma(u_o^T v_c)}{\partial u_o} \\
&= \sigma^{-1}(u_o^T v_c)\sigma(u_o^T v_c)\sigma(-u_o^T v_c) \cdot v_c \\
&= -\sigma(-u_o^T v_c) \cdot v_c \\
&= -\left(1 - \sigma(u_o^T v_c)\right) \cdot v_c \\
&= \left(\sigma(u_o^T v_c) - 1\right) \cdot v_c
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathrm{J}_{\mathrm{negsample}}}{\partial u_{k(\neq o)}} &= \frac{\partial \left( -\log \sigma(u_o^T v_c) - \sum_{k=1}^{K} \log \sigma(-u_k^T v_c) \right)}{\partial u_k} \\
&= -\sigma^{-1}(-u_k^T v_c) \cdot \frac{\partial \sigma(-u_k^T v_c)}{\partial u_k} \\
&= -\sigma^{-1}(-u_k^T v_c)\sigma(-u_k^T v_c)\sigma(u_k^T v_c) \cdot (-v_c) \\
&= \sigma(u_k^T v_c) \cdot v_c
\end{aligned}$$

**(d)** Let $F \in \{\mathrm{J}_{\mathrm{softmaxCE}}, \mathrm{J}_{\mathrm{negsample}}\}$ so we actually found the gradients in previous sections:

$$\frac{\partial \mathrm{J}_{\mathrm{skipgram}}(w_c, v_c)}{\partial U} = \sum_{-m \le j \ne 0 \le m} \frac{\partial F(w_{c+j}, v_c)}{\partial U}$$

$$\frac{\partial \mathrm{J}_{\mathrm{skipgram}}(w_c, v_c)}{\partial v_c} = \sum_{-m \le j \ne 0 \le m} \frac{\partial F(w_{c+j}, v_c)}{\partial v_c}$$

$$\frac{\partial \mathrm{J}_{\mathrm{skipgram}}(w_c, v_c)}{\partial v_{j \ne c}} = 0$$

**(e)** Below are the results of our test. Even though the graph in 2 dimensions is not very explanatory, We do note that two very **similar** words like "amazing" and "wonderful" are very close when measuring by the cosine similarity. Moreover, the **article** words such as "a", "the", "an" and signs like dot and comma appear relatively far from the adjectives. These observations give us a good hint about the job that was done in Word2Vec, which is to represent words with an expressive low dimensional vector.

**List of related words** (Nearest Neighbors):
"the": ['the', 'if', 'that', 'comedy\\/thriller', 'or', 'a', '.', 'is', 'derek', 'bolt', 'decide']
"unique": ['unique', '1979', 'puns', 'ba', 'realized', 'succumb', 'chabrolian', 'dares', 'regardless', 'imaginative', 'lunar']
"superb": ['superb', 'mine', 'gold', 'zingers', 'moppets', 'roussillon', 'best',

'ghoulish', 'industry', 'pool', 'transporter']
"comedy": ['comedy', 'sensation', 'observation', 'fast', 'first-timer', 'singing', 'cleaving', 'longest', 'cute', 'mature', 'often-funny']
"surprisingly": ['surprisingly', 'either', 'hundred', '20-car', 'philandering', 'unusually', 'protective', 'dogs', 'bollywood', 'thinking', 'soderbergh']