Name: Vivek Viswanath
UFID: 98337168.

EEL 6935
Programming Assignment - 2 (Derivations)

Q3.2 **Neural Network Basics**

Ans. The sigmoid function is

$$f(x) = \frac{1}{1+e^{-x}}. \text{ Let } g(x) = 1+e^{-x}$$

Now, $$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g} \times \frac{\partial g}{\partial x}$$

$$= -\frac{1}{(1+e^{-x})^2} \cdot (-e^{-x})$$

$$= \left(\frac{1}{1+e^{-x}}\right)^2 \cdot e^{-x}$$

$$= e^{-x} \cdot f^2(x)$$

$$= \left(\frac{1}{f(x)} - 1\right) f^2(x)$$

$$= f(x)(1-f(x))$$

$$\therefore, \frac{d}{dx}(f(x)) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = f(x)(1-f(x))$$

(Ans.)

⇒ Result implemented in q1-sigmoid.py

Q 3.3 Word2Vec.

1. Softmax Cost And Gradient.

Given a predicted word vector $\hat{r}$, & word prediction is made with softmax function in word2vec :

$$\hat{y}_i = P_r(w_i \mid \hat{r}, u_{w_1 \ldots |V|}) = \frac{\exp(u_{w_i}^T \hat{r})}{\sum_{j=1}^{|V|} \exp(u_{w_j}^T \hat{r})}$$

Now, the question states that we are to assume cross-entropy cost is applied to this prediction. Therefore,

applying cross-entropy cost, we get

$$J(\hat{r}, w) = -\log\left(\frac{e^{w_i^T \hat{r}}}{\sum_{j=1}^{|V|} e^{w_j^T \hat{r}}}\right)$$

$$= -\log e^{w_i^T \hat{r}} + \log \sum_{j=1}^{|V|} e^{w_j^T \hat{r}}$$

$$= -w_i^T \hat{r} + \log \sum_{j=1}^{|V|} e^{w_j^T \hat{r}}$$

Let $z_j = w_j^T \hat{r}$ and $1[j=i]$ which is an indicator function indicating $j=i$ means the function value $= 1$ & 0 otherwise.

Then, $$\frac{\partial J}{\partial z_k} = \frac{e^{z_k}}{\sum_{j=1}^{|V|} e^{z_j}} - 1[k=i] \quad (1)$$

& $$\frac{\partial J}{\partial \hat{r}} = \sum_{k=1}^{|V|} \frac{\partial J}{\partial z_k} \cdot \frac{\partial z_k}{\partial \hat{r}}$$

$$= \sum_{k=1}^{|V|} w_j \left(\frac{e^{z_k}}{\sum_{j=1}^{|V|} e^{z_j}} - 1[k=i] \right) \quad (2)$$

$\therefore$ $$\frac{\partial J}{\partial w_k} = \frac{\partial J}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_k}$$

$$\Rightarrow \boxed{\frac{\partial J}{\partial w_k} = \hat{r} \left(\frac{e^{z_k}}{\sum_{j=1}^{|V|} e^{z_j}} - 1[k=i] \right)}$$

(from (1) & (2))

(Ans).

$\Rightarrow$ Result implemented in q3_word2vec.py

Q 3.3 Word2Vec

2. <u>Negative Sampling Loss</u>.

Let $z_j = \omega_j^T \hat{x}$ & let $\mathbb{1}[j=i]$ mean that the function evaluates to 1 if $j=i$ & 0 otherwise.

For $i \notin K$:

$$\frac{\partial J}{\partial z_i} = -\frac{\partial}{\partial z_i}(\log(\sigma(z_i)))$$

$$= -\frac{\sigma'(z_i)}{\sigma(z_i)} = -\frac{[\sigma(z_i)(1-\sigma(z_i))]}{\sigma(z_i)}$$

$$= \sigma(z_i) - 1.$$

For $i \in K$:

$$\frac{\partial J}{\partial z_i} = - \frac{-\sigma'(z_i)}{\sigma(-z_i)}$$

$$= \frac{\sigma(-z_i)(1-\sigma(-z_i))}{\sigma(-z_i)}$$

$$= 1 - \sigma(-z_i) = \sigma(z_i)$$

We see that this is the prediction error. Then, through chain rule,

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_j} = \left(\sigma(w_j^T \hat{r}) - 1[j=i]\right)\hat{r}$$

&

$$\frac{\partial J}{\partial \hat{r}} = \frac{\partial J}{\partial z_j} \cdot \frac{\partial z_j}{\partial \hat{r}} = \left(\sigma(w_j^T \hat{r}) - 1[j=i]\right)w_j$$

∴, The negative sampling loss is much cheaper to evaluate because we don't need to sum over the whole vocabulary, only $|K|$ samples.

(Ans)

P.T.O.

Q 34. Sentiment Analysis

Please find the implementations in the python files submitted.

3. When we plot the classification accuracy with respect to regularization value, we can see that that regularization improves the accuracy on the development set, but too much regularization introduces a bias that results in worse performance (lower accuracy).

(Ans.)