Weifeng Jin

UID: 1769685
LING 570 HW #11
Date: 12/13/2017

Q1

1a Logistic Function:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where:
e = the natural logarithm base
x_0 = the x-value of the sigmoid's midpoint
L = the curve's maximum value, and
k = the steepness of the curve

1b Standard Logistic Function:

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

1c Derivative of Standard Logistic Function:

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

Q2

2a Definition of Sigmoid Function:
A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.

2b Relationship between Sigmoid and Logistic Function:
The standard logistic function is a kind of sigmoid function.

Q3

3a Tanh Function:

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

3b Relationship between Sigmoid and Tanh Function:
The Tanh function is a kind of sigmoid function.

Q4

4a Softmax Function:

$$\sigma : \mathbb{R}^K \to [0,1]^K$$

$$\sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

4b Calculation:

$$\sigma([1,2,3])_1 = \frac{e}{e+e^2+e^3}$$

$$\sigma([1,2,3])_2 = \frac{e^2}{e+e^2+e^3}$$

$$\sigma([1,2,3])_3 = \frac{e^3}{e+e^2+e^3}$$

$$\sigma([1,2,3]) = [\frac{e}{e+e^2+e^3}, \frac{e^2}{e+e^2+e^3}, \frac{e^3}{e+e^2+e^3} = [0.09, 0.24, 0.67]$$

Q5

5a $A \times B = \begin{bmatrix} 1\times 2 + 2\times 1 + 3\times 3 \\ 4\times 2 + 5\times 1 + 6\times 3 \\ 7\times 2 + 8\times 1 + 9\times 3 \end{bmatrix} = \begin{bmatrix} 13 \\ 31 \\ 49 \end{bmatrix}$

$B \times A =$ undefined

the transpose of A $A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

the transpose of B $B^T = \begin{bmatrix} 2 & 1 & 3 \end{bmatrix}$

The dimensions of B are 3×1
The dimensions of the transpose of B are 1×3

5b $A \times B = \begin{bmatrix} 1\times 2 + 2\times 1 & 1\times 0 + 2\times 2 \\ 3\times 2 + 4\times 1 & 3\times 0 + 4\times 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 8 \end{bmatrix}$

$B \times A = \begin{bmatrix} 2\times 1 + 0\times 3 & 2\times 2 + 0\times 4 \\ 1\times 1 + 2\times 3 & 1\times 2 + 2\times 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 7 & 10 \end{bmatrix}$

Matrix multiplication is NOT commutative.

Q6

6a For the fake task, the input is a word w_1, the output is a probability distribution describing the probability of choosing w_2 by picking a word at random nearby w_1

6b There are 2 layers. Input layer has 100K neurons, the hidden layer has 50 neurons, the output layer has 100K neurons.

6c There are 2 big matrices in the network. The first matrix has dimensions $100K \times 50$ and the second has $50 \times 100K$. There are $300 \times 100K$ weights to be estimated during training.

6d Because we need to train the neural network and obtain the weights for some specific task. Although we would not use the network model for our test, we need the weights for our word vectors.

6e From the blog, x "is a one-hot vector representing the input word" and y "is also a one-hot vector representing the output word".

6f One-hot is a vector representation of the word. In this example, it has 100K components and we place "1" in the position corresponding to the word and 0s elsewhere. It is used in the input layer. Because only one component of the vector is 1, all else are 0 and there is only one hot component in the vector.

6f Softmax makes it possible for the output vector to be a probability distribution.

Q7

7a Vector dimensionality, training epochs of the model, number of training words.

7b From paper #2, Negative Sampling is defined by the objective:

$$\log \sigma({v'_{w_o}}^T v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} [\log \sigma({v'_{w_i}}^T v_{w_I})]$$

which is used to replace every $\log P(w_O|w_I)$ term in the Skip-gram objective. Thus the task is to distinguish the target word w_O from draws from the noise distribution $P_n(w)$ using logistic regres- sion, where there are k negative samples for each data sample.

The benefit of negative sampling is that it uses only samples and does not need the numerical probabilities of the noise distribution.

7c According to paper #2, there is an imbalance between the rare and frequent words. And these frequent words provide less information value than the rare words. Thus, to counter such imbalance, we have to use subsampling: each word w_i in the training set is discarded with probability computed by the formula:

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

where $f(w_i)$ is the frequency of word w_i and t is a chosen threshold.

7d According to paper # 2, to learn vector representation for phrases, we first find words that appear frequently together, and infrequently in other contexts. And we form phrases based on the unigram and bigram counts, using

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)}$$

δ is a discounting coefficient and prevents too many phrase forming. The bigrams with score above the chosen threshold would be chosen to form phrases.

Running 3 passes of the training data, the maximum length of a phrase that can be found is 4.

We use the Skip-Gram to compare the similarity between the given phrase and other words/phrase just as comparison between words.