NTU MSAI AI6127 Assignment 1 Answer Sheet

Note: For Prof, TA, other NTU students information, all the code here are modified based this example

- **Deep_NLP_Assignment1.pdf** is the assignment requirement published by the professor
- In case the Github can not render the math eqution correctly, please refer to the AnswerSheet
- main_fnn.py is the script to execute the training process, the model hyper parameters are included in this file. this file is modified based on the **main.py**
- model.pt is generated by the main_fnn.py
- embedding_eval.py is the script to generate spearmanr.csv, which contains the Spearman correlation required by the assignment. The empty Spearman correlation in the csv file means that one or both of the word pair did not show in the input data set.

Question 1 Proof:

In the multi-class logistic regression, the entropy loss function is:

$$Loss = -\sum_{k}^{n} y_k \log \hat{y}_k$$

The predicted value is:

$$\hat{y}_k = \frac{e^{o_k}}{\sum_i e^{o_i}}$$

where $o_i = W_i z$ given the z is the input vector to the GLM layer, W is the matrix containing the associated weight vectors. The Derivative of the predicted value is:

$$\frac{\partial \hat{y}_k}{\partial o_i} = \frac{\partial}{\partial o_i}\left(\frac{e^{o_k}}{\sum_i e^{o_i}}\right)$$

$$= \frac{\frac{\partial e^{o_k}}{\partial o_i} \cdot \sum_i e^{o_i} - e^{o_k} \cdot e^{o_i}}{\left(\sum_i e^{o_i}\right)^2}$$

when $i = k$, $\frac{\partial e^{o_k}}{\partial o_i} = e^{o_k}$, when $i \neq k$, $\frac{\partial e^{o_k}}{\partial o_i} = 0$: Thus

$$\frac{\partial \hat{y}_k}{\partial o_i} = \begin{cases} \hat{y}_k(1 - \hat{y}_k), & i = k \\ -\hat{y}_k \hat{y}_i, & i \neq k \end{cases}$$

Combined with Loss function:

$$\begin{aligned} \frac{\partial Loss}{\partial o_i} &= -\sum_k y_k \frac{\partial \log \hat{y}_k}{\partial o_i} \\ &= -\sum_k y_k \frac{1}{\hat{y}_k} \frac{\partial \hat{y}_k}{\partial o_i} \\ &= -y_i(1 - \hat{y}_i) + \sum_{k \neq i} \hat{y}_i y_k \\ &= -y_i + y_i \hat{y}_i + \sum_{k \neq i} \hat{y}_i y_k \\ &= \hat{y}_i(y_i + \sum_{k \neq i} y_i) - y_i \\ &= \hat{y}_i(\sum_k y_k) - y_i \\ &= \hat{y}_i - y_i \end{aligned}$$

Given that $\frac{\partial o_i}{\partial W_i} = z$, we can get

$$\frac{\partial Loss}{\partial W} = \frac{\partial Loss}{\partial o_i}\frac{\partial o_i}{\partial W_i} = (\hat{y_i} - y_i)z$$

Question 2

sub-question (iii ~ vi)

please refer to the main_fnn.py, model.py

sub-question (vii)

Please see the generate.py. I adapted this script to support interactive text generation. The generated.txt is the corresponse generated text file

sub-question (viii)

The most time consuming step in the feed-forward should be the softmax computation and we should implement negetive sampling to address this problem. But in the reality the softmax step cost least time, please refer to the following snapshot

```
-----------------------------------------------------------------------------------------
Epoch 10 start at 2020-02-15 16:36:31
-----------------------------------------------------------------------------------------
| epoch  10 |   200/ 2983 batches | lr 1.25 | ms/batch  2.72 | loss 10.44 | ppl 34076.62
| epoch  10 |   400/ 2983 batches | lr 1.25 | ms/batch  2.73 | loss 10.38 | ppl 32302.25
| epoch  10 |   600/ 2983 batches | lr 1.25 | ms/batch  2.50 | loss 10.38 | ppl 32167.64
| epoch  10 |   800/ 2983 batches | lr 1.25 | ms/batch  3.48 | loss 10.38 | ppl 32349.64
| epoch  10 |  1000/ 2983 batches | lr 1.25 | ms/batch  2.76 | loss 10.38 | ppl 32284.81
| epoch  10 |  1200/ 2983 batches | lr 1.25 | ms/batch  2.39 | loss 10.39 | ppl 32433.37
| epoch  10 |  1400/ 2983 batches | lr 1.25 | ms/batch  2.55 | loss 10.39 | ppl 32409.06
| epoch  10 |  1600/ 2983 batches | lr 1.25 | ms/batch  2.25 | loss 10.39 | ppl 32413.59
| epoch  10 |  1800/ 2983 batches | lr 1.25 | ms/batch  2.25 | loss 10.39 | ppl 32458.51
| epoch  10 |  2000/ 2983 batches | lr 1.25 | ms/batch  2.23 | loss 10.39 | ppl 32500.75
| epoch  10 |  2200/ 2983 batches | lr 1.25 | ms/batch  2.51 | loss 10.38 | ppl 32357.55
| epoch  10 |  2400/ 2983 batches | lr 1.25 | ms/batch  2.48 | loss 10.39 | ppl 32438.70
| epoch  10 |  2600/ 2983 batches | lr 1.25 | ms/batch  2.31 | loss 10.39 | ppl 32481.89
| epoch  10 |  2800/ 2983 batches | lr 1.25 | ms/batch  2.24 | loss 10.39 | ppl 32456.43
Total Encoder Time: 7.951249s, Hidden Layer Time: 10.305751s, Decoder Time: 6.362265s, Softmax Time:3.206784s
-----------------------------------------------------------------------------------------
| end of epoch  10 | time: 100.00s | valid loss 10.39 | valid ppl 32575.82
-----------------------------------------------------------------------------------------
=========================================================================================
| End of training | test loss 10.38 | test ppl 32331.13
=========================================================================================
```

sub-question (ix)

For the code please refer to embedding_eval.py For the result Please refer to spearmanr.csv, please note that the empty spearman for some word pair mean one or both of them did not show in the data/dictionary

Question 3

If the student are familiar with the machine learning, deep learning, python and pytorch, i guess it would take about one day to complete the assignment.

If the student has no idea of the above baisc knowledge, the time cost on the assignment would be much more longer, because it require student to pick up everything, probably would cost around a week (just finish the assignment, may not be understand the concept of the Embedding, FNN, RNN, pytorch etc.)

But From my personal perspective, the assignment should be really challengable and pratical to student. I personally hope the following assignment or project could lead us to build a workable and meaningful model.