NLP Assignment 1 Report

Rutvik Parekh
SBU ID: 112687483

# 1) Hyper Parameters explored:

a) **Batch_size**: The initial batch_size given was 128. I experimented with many different values such as 64, 32, 256 and 1024. One interesting observation I observed with increasing the batch size to 1024 was that it became very slow. As the batch size increases, the training time increases.

b) **Num_skips**: Initially, the value provided in the model was 8. I experimented with it and changed the value to 16 and 32. I did not observe much difference in the accuracy or the training time by changing this parameter or maybe the change wasn't big enough to be visible.

c) **Skip_windows**: I changed this in accordance with the num_skips. If num_skips doubled, I doubled the value of skip_windows. And if num_skips was halved, I halved the value of skip_windows

**d) Learning Rate:** Learning rate controls how the controls how quickly a model is adapted to training. Generally, what I observed was that when the learning rate was small, the loss took time to converge to its minimum value in the given number of epochs. So, if the learning rate is too slow, the model may get stuck and move very slowly towards loss reduction. And if the learning rate is too high, we may end up in a suboptimal solution. Having the right learning rate is important to fit the model to the training data. I started with the initial learning rate of 1 and then increased it to 2 and then 4 and so on. I got the best results when the learning rate was 2

e) Max_num_steps: These are the number of epochs. Generally, I observed that the more number of epochs are better for training the model. But, this also depends on the learning rate. The learning rate has to be in sync with the number of epochs. The initial give number of steps was 200000. I changed it to 100000, 50000, 300000 and 400000

2) Results on analogy task for 5 different configurations of the hyper parameters

## For cross-entropy:

1)

1) Batch-size: 64
2) Num Skips:16
3) Skip windows:8
4) Number of Epochs:200000

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 260
Number of Least Illustrative Guessed Incorrectly: 654
Accuracy of Least Illustrative Guesses: 28.4%
Number of Most Illustrative Guessed Correctly: 271
Number of Most Illustrative Guessed Incorrectly: 643
Accuracy of Most Illustrative Guesses: 29.6%
Overall Accuracy: 29.0%

2)

5) Batch-size: 64
6) Num Skips:8
7) Skip windows:4
8) Number of Epochs:400000

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 318
Number of Least Illustrative Guessed Incorrectly: 596
Accuracy of Least Illustrative Guesses: 34.8%
Number of Most Illustrative Guessed Correctly: 298
Number of Most Illustrative Guessed Incorrectly: 616
Accuracy of Most Illustrative Guesses: 32.6%
Overall Accuracy: 33.7%

3)

9) Batch-size: 1024
10) Num Skips:16
11) Skip windows:8
12) Number of Epochs:100000

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 318
Number of Least Illustrative Guessed Incorrectly: 596
Accuracy of Least Illustrative Guesses: 34.8%
Number of Most Illustrative Guessed Correctly: 298
Number of Most Illustrative Guessed Incorrectly: 616
Accuracy of Most Illustrative Guesses: 32.6%
Overall Accuracy: 33.7%

4)

13) Batch-size: 128
14) Num Skips:16
15) Skip windows:8
16) Number of Epochs:250000

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 321
Number of Least Illustrative Guessed Incorrectly: 593
Accuracy of Least Illustrative Guesses: 35.1%
Number of Most Illustrative Guessed Correctly: 297
Number of Most Illustrative Guessed Incorrectly: 617
Accuracy of Most Illustrative Guesses: 32.5%
Overall Accuracy: 33.8%

5)

17) Batch-size: 256
18) Num Skips:16
19) Skip windows:8
20) Number of Epochs:300000

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 316
Number of Least Illustrative Guessed Incorrectly: 598
Accuracy of Least Illustrative Guesses: 34.6%
Number of Most Illustrative Guessed Correctly: 299
Number of Most Illustrative Guessed Incorrectly: 615
Accuracy of Most Illustrative Guesses: 32.7%
Overall Accuracy: 33.6%

For NCE:

1)

21) Batch-size: 128
22) Num Skips:16
23) Skip windows:8
24) Number of Epochs:300000
25) Learning Rate: 1.0

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 255
Number of Least Illustrative Guessed Incorrectly: 659
Accuracy of Least Illustrative Guesses: 27.9%
Number of Most Illustrative Guessed Correctly: 396
Number of Most Illustrative Guessed Incorrectly: 518
Accuracy of Most Illustrative Guesses: 43.3%
Overall Accuracy: 35.6%

2)

26) Batch-size: 256
27) Num Skips:16
28) Skip windows:8
29) Number of Epochs:300009
30) Learning Rate: 2.0

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 255
Number of Least Illustrative Guessed Incorrectly: 659
Accuracy of Least Illustrative Guesses: 26.9%
Number of Most Illustrative Guessed Correctly: 376
Number of Most Illustrative Guessed Incorrectly: 542
Accuracy of Most Illustrative Guesses: 42.3%
Overall Accuracy: 34.6%

3)

31) Batch-size: 64
32) Num Skips:16
33) Skip windows:8
34) Number of Epochs:50000
35) Learning Rate: 5.0

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 316
Number of Least Illustrative Guessed Incorrectly: 598
Accuracy of Least Illustrative Guesses: 34.1%
Number of Most Illustrative Guessed Correctly: 279
Number of Most Illustrative Guessed Incorrectly: 615
Accuracy of Most Illustrative Guesses: 32.1%
Overall Accuracy: 34.9%

4) Batch-size: 256

36) Num Skips:16
37) Skip windows:8
38) Number of Epochs:100000
**39)** Learning Rate: 10.0

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 300
Number of Least Illustrative Guessed Incorrectly: 582
Accuracy of Least Illustrative Guesses: 34.1%
Number of Most Illustrative Guessed Correctly: 315
Number of Most Illustrative Guessed Incorrectly: 599
Accuracy of Most Illustrative Guesses: 33.7%
Overall Accuracy: 34.1%

5) Batch-size: 64
    40) Num Skips:16
    41) Skip windows:8
    42) Number of Epochs:300000
    43) Learning Rate: 25.0

Number of MaxDiff Questions: 914
Number of Least Illustrative Guessed Correctly: 351
Number of Least Illustrative Guessed Incorrectly: 562
Accuracy of Least Illustrative Guesses: 42.6%
Number of Most Illustrative Guessed Correctly: 299
Number of Most Illustrative Guessed Incorrectly: 615
Accuracy of Most Illustrative Guesses: 32.7%
Overall Accuracy: 35.2%

3) I have attached the code for generating the top 20 similar words for first, American and would on BlackBoard.

# Summary of the justication behind the NCE method loss.

Reference
From:https://datascience.stackexchange.com/questions/13216/intuitive-explanation-of-noise-contrastive-estimation-nce-loss
There are some issues with learning the word vectors using an "standard" neural network. In this way, the word vectors are learned while the network learns to predict the next word given a window of words (the input of the network).

Predicting the next word is like predicting the class. That is, such a network is just a "standard" multinomial (multi-class) classifier. And this network must have as many output neurons as classes there are. When classes are actual words, the number of neurons is, well, huge.

A "standard" neural network is usually trained with a cross-entropy cost function which requires the values of the output neurons to represent probabilities - which means that the output "scores" computed by the network for each class have to be normalized, converted into actual probabilities for each class. This normalization step is achieved by means of the softmax function. Softmax is very costly when applied to a huge output layer.

**The (a) solution**

In order to deal with this issue, that is, the expensive computation of the softmax, Word2Vec uses a technique called noise-contrastive estimation. This technique was introduced by [A] (reformulated by [B]) then used in [C], [D], [E] to learn word embeddings from unlabelled natural language text.

The basic idea is to convert a multinomial classification problem (as it is the problem of predicting the next word) to a binary classification problem. That is, instead of using softmax to estimate a true probability distribution of the output word, a binary logistic regression (binary classification) is used instead.

For each training sample, the enhanced (optimized) classifier is fed a true pair (a center word and another word that appears in its context) and a number of kk randomly corrupted pairs (consisting of the center word and a randomly chosen word from the vocabulary). By learning to distinguish the true pairs from corrupted ones, the classifier will ultimately learn the word vectors.

This is important: instead of predicting the next word (the "standard" training technique), the optimized classifier simply predicts whether a pair of words is good or bad.

Word2Vec slightly customizes the process and calls it negative sampling. In Word2Vec, the words for the negative samples (used for the corrupted pairs) are drawn from a specially designed distribution, which favours less frequent words to be drawn more often