NLP A1 Report

1. Hyper Parameters Explored
    a. **batch_size** - This parameter is used to generate a batch of data for each epoch of training process. Increasing the batch_size will like give a better result, because we are using Gradient Descent optimizer, and a bigger batch means more stable loss value. On the contrary, a smaller batch does not have enough samples, and thus when we approximate the gradient using a small mini batch it turns out to be noisy.
    b. **embedding_size** - This is the size of the embedding vector created for each word in the vocabulary. Increasing the embedding_size may give a better accuracy but will slow down the training process due to increased computations.
    c. **skip_window** - This parameter decides how many words to consider left and right from a context word while generating batches during training. Increasing the skip_window may result in better accuracy. However, skip_window should not be made very large, because then will consider words that are far away from the current word and they might not be related in any way.
    d. **num_skips** - This parameter controls the number of samples to draw in a window. This parameter is generally increased when increasing the size of the skip window. Doing so would increase the probability of each word in the context being considered. That would increase the accuracy if both skip_window and num_skips are small to moderate.
    e. **learning_rate** - This parameter controls the size of the updates, made to the parameters of the model during each epoch of training. Keeping this value small may give a better result as a small step in the direction of the neg gradient will eventually lead us to (local) optima.
    f. **max_num_step** - This parameter controls the number of epochs to run for training the model. This is often used in conjunction with the learning rate. As a small learning rate would require more updates of the parameters to reach the optima.

2. Results on the Analogy Task

a. Using Cross Entropy

Batch_size	embedding_size	skip_window	num_skips	Learning rate	max_num_steps	Avg Accuracy
128	128	4	8	1.0	200001	29%
128	128	4	8	0.01	200001	29%
128	128	8	16	0.01	200001	29%
128	*256	4	8	0.01	200001	33.3%
128	*256	4	8	0.001	400001	34.7%
256	*256	4	8	0.0001	1000001	35%
512	128	2	4	0.01	200001	29%

b. Using NCE

Batch_size	embedding_size	skip_window	num_skips	Learning rate	max_num_steps	Avg Accuracy
128	128	4	8	1.0	200001	27.9%
128	128	4	8	0.01	200001	29.5%
128	128	8	16	0.01	200001	29%
128	*256	4	8	0.01	200001	31.6%
128	*256	4	8	0.001	500001	29%
256	*256	4	8	0.0001	1000001	29.7%
512	*1024	8	16	0.1	150001	32.2%

Observations:

- Increasing **num_max_steps** and decreasing the **learning_rate** increases Accuracy.
- Increasing the **size of vector** increases the accuracy but takes longer to train.
- Same happens when increasing the **batch_size**.
- I did not observe any significant changes when trying different values of the **num_skips** and **skip_window.**

NOTE :

- The models marked with * in the tables above, give better accuracy in the word analogy task as compared to the baseline, but I won't be submitting those. Because on analyzing the nearest words for those models, I realised that it barely learned anything, even after extended training.
- The reason for such an observation can be that I increased the **embedding_size,** hence I could not use the pretrained model. So when training from scratch, the model did not learn good word embeddings.

3. Top 20 similar words

a. NCE :

Nearest to first: most, and, was, at, of, to, in, during, s, one, on, is, name, that, which, by, from, nine, for, following,

Nearest to american: british, german, english, french, war, italian, its, december, european, international, united, understood, brought, autres, ancient, states, borges, rucker, union, russian,

Nearest to would: been, not, will, they, who, could, do, that, must, india, said, does, we, but, may, these, so, to, did, which,

b. Cross Entropy

Nearest to first: last, name, following, during, most, original, second, same, end, until, after, best, city, book, before, united, next, main, beginning, title,

Nearest to american: german, british, french, english, italian, its, russian, war, european, understood, international, borges, irish, canadian, united, trade, d, writer, player, terminal,

Nearest to would: not, could, will, been, we, that, said, must, india, they, do, does, did, who, you, families, if, to, should, may,

4. Summary of justification for NCE loss

NCE loss is basically used because when we try to make a language model using softmax we normalize over the entire vocabulary. And this computation is expensive.

In NCE loss method, we train a Logistic regression classifier, which can distinguish between the sample from data or sample from noise distribution. By doing so, we convert the
a multinomial classification problem (as it is the problem of predicting the context word) to a binary classification problem.

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