REPORT (SbuID: 112073671)

Code: Submitted in zip file

Implementation and experiment details:

My Models:

For Cross Entropy:

Model File Name: In models_links.txt

Configuration:

batch_size = 212
embedding_size = 128
skip_window = 2
num_skips = 4
num_sampled = 128
Max_num_steps = 200001

For NCE :

Model File Name: In models_links.txt

Configuration:

batch_size = 212
embedding_size = 128
skip_window = 2
num_skips = 4
num_sampled = 128
Max_num_steps = 200001

Prediction File:

Using Cross Entropy: word_analogy_test_predictions_cross_entropy.txt
Using NCE: word_analogy_test_predictions_nce.txt

Report:

1. Hyper parameters:
   Batch_size: the number of instances in one batch
   embedding_size = dimension of embedding vector
   skip_window = how many words to consider left and right from a context word
   (window_size = skip_windows*2+1)
   num_skips = the number of samples you want to draw in a window
   num_sampled = negative sampling (k)
   Max_num_steps = training iterations

2. Analogy Results for below 5 experimented configs:

a. Config1:
   batch_size = 128
   embedding_size = 128
   skip_window = 2
   num_skips = 4
   num_sampled = 64
   Cross Entropy Loss that I got:
   NCE Loss that I got:

b. Config2 - Increasing batch_size:
   batch_size = Tested with 128, 212, 264
   embedding_size = 128
   skip_window = 2
   num_skips = 4
   num_sampled = 64
   Batch sizes 128, 264 and higher gave me accuracy around 32.8% and for 264, I got around 33.2% in NCE. Too higher batch sizes are reducing accuracy.

c. Config3 - Increasing num_samples (k) and batch size:
   batch_size = 264
   embedding_size = 128
   skip_window = 2
   num_skips = 4
   num_sampled = 128
   I did not observe much change by changing num_sampled in both NCE and Cross Entropy.

d. Config4 - Increasing min_num_steps:
   batch_size = 128
   embedding_size = 128
   skip_window = 2
   num_skips = 4
   num_sampled = 64
   Max_num_steps = 400002 (doubled)
   This config was taking more time comparatively to the normal ones considering the all extra the extra iterations. It didn’t felt much useful for me as both Cross ENtropy and NCE losses are converging at earlier steps.

e. Config5 - Increasing skip_window
   batch_size = 128
   embedding_size = 128
   skip_window = 8
   num_skips = 16

num_sampled = 64
#Explained below

Optimal Config that I choose based on above experiments:
Cross entropy and NCE losses mentioned in each of above cases are final converged ones.
Higher skip_window -> big window_size tell us more about the sentences.
Smaller skip_window -> small window_size tells us about similarity of words.
Considering the tasks in word_analogies are relation oriented, I'm thinking a smaller skip_window will be relevant.
Larger batch size in cross entropy and NCE gave less accuracy % (around 2% diff)
So, I choose optimal parameters as mentioned in section 2 in this report for both cases.

3. Top 20 similar words for {'first', 'american', 'would'}:
   a.

Word	Cross Entropy Model	NCE Model
first	last, following, name, most, during, second, original, same, end, until, after, best, at, book, before, next, city, main, beginning, title	all, had, they, but, this, some, their, not, has, other, at, which, its, he, were, also, most, or, his, have
american	german, british, french, english, italian, its, war, russian, european, of, eu, international, other, irish, canadian, borges, united, trade, d, barzani	german, british, french, english, italian, war, russian, european, eu, international, irish, canadian, borges, united, trade, d, barzani, player, comedian, writer
would	not, that, could, will, been, we, said, must, might, they, do, does, to, who, did, you, seems, if, should, but	could, will, we, said, must, might, do, does, who, did, you, seems, if, should, may, so, even, only, these, i

   b. Similarities noticed in above 2:
   All though, there is not much diff in accuracy % in Cross entropy(CE) and NCE, looking at above results,below are observations wrt different scenarios. For the 1st word, few of the results didn't make sense to me. CE included opposites of each of the given words. NCE results are much similar words to the given word than CE.

4. NCE loss justification:

Formula that I used is - (LOG(s1) + SUM(LOG(1-s2)))

Where,

s1 = σ(ucT*uo + bo - log(k*Pr_wo))

s2 = σ(ucT*ux + bx - log(k*Pr_wx))

bo is bias vector wrt uo

bx is bias vector wrt ux

uc is from inputs/embedding for context words

uo, ux are embedding wrt labels and samples respectively

Pr_wo and Pr_wx are probabilities wrt labels and samples respectively

I'm calculating Pr_wo by doing embedding lookup of labels in given unigram_prob and similarly Pr_wx by doing embedding lookup of samples in unigram_prob.

While computing s1 and s2 (above), I'm reshaping (dimensions) tensors accordingly and I'm computing reduced sum over columns for summation in s2.

In word2vec's cross entropy, probabilities of output vectors are normalized using softmax function. Applying this to huge output layer in word2vec's neural net is costly.

NCE solves this huge computational complexity problem by considering multi-classification to binary classification. To get probability of an outer word, NCE replaces above costly computation with a binary logistic regression. NCE does negative sampling i.e., it takes negative words as samples and classifies if a word fits or not wrt to a given(center) word.

In above formula used, the sum calculated in s2 is over k noise samples instead of considering entire vocabulary. Irrespective of size of vocabulary, this gives us linear training time in the number of noise samples. If k is increased, this estimate approaches gradient of the normalized model, allowing us to trade off computation cost against estimation accuracy.

References:

http://www.aclweb.org/anthology/W16-2922

https://www.cs.toronto.edu/~amnih/papers/wordreps.pdf

https://datascience.stackexchange.com/questions/13216/intuitive-explanation-of-noise-contrastive-estimation-nce-loss