Implementation of Word2Vec

Abdullah Mitkar

October 9, 2019

abdullah.mitkar@stonybrook.edu 112685069

1 Hyper parameter Exploration

This section elucidates the tuning and exploration of hyper parameter that increases the accuracy of the model. Hyper parameter tuning involves making changes to a list of parameters viz, batch_size, skip_window, num_skips, learning rate, num_samples, max_num_steps etc in the code to achieve a improved accuracy.
Exploring the parameters that lead to better accuracy is done by simply changing the values in the code and building the model again. The parameters tuned and explored includes., batch_size, skip_window, num_skips, learning_rate (for Gradient Descent) and num_samples.
The meaning of each of the parameter is as follows:

- Batch_size: This is the no of items or samples in the batch used for training the model.
- Skip_window: This is the no of target words from a single window to be considered in the left and right directions of the context word.

  Size of window = skip window * 2 (left and right) + 1 (center word)
- Num_skips: This denotes the maximum no of samples to be considered in a batch from a single window.
- Learning rate: This is "change in weight" or "the punishing factor" to the weights to be able to identify gradients (maxima or minima) so as to avoid overfitting. This is also called "regularization" constant.
- Num_samples: Number of negative examples to sample. This is used in Noise Contrastive Estimation loss function.

2 Results of hyper parameter tuning

2.1 Cross Entropy

Following is the result of analogy task for multiple configurations of parameters for Cross Entropy:

batch_size	skip_window	num_skips	learning rate	num_samples	max_num_steps	loss	accuracy
32	2	4	0.6	8	50000	2.23	31.80 %
32	4	4	0.6	48	50000	2.23	34.70 %
32	4	4	1	16	50000	2.23	30.60 %
32	4	4	1.2	48	50000	2.23	28.90 %
64	2	4	1	16	50000	2.68	32.50 %
64	2	4	1	16	200001	2.68	32.80 %
64	4	4	1	16	50000	2.78	32.40 %
64	8	16	1	16	200001	3.55	31.50 %
64	8	16	1	32	50000	3.55	32.40 %
64	8	16	1	48	50000	3.55	33.40 %
64	8	16	0.6	48	200000	3.55	36.40 %
128	4	8	1.2	128	200001	3.94	32.30 %
128	8	8	1	32	200001	4.02	28.90 %
128	8	16	1	48	50000	4.18	32.20 %
256	4	8	1	32	50000	4.57	31.30 %
256	8	8	1	32	200001	1.07	32.40 %

2.1.1 Trends in hyper parameter tuning for Cross Entropy

- Batch size: An increase in batch size increases the accuracy of the model with no impact on the avg loss.
- Skip_window: An increase in skip window increases the accuracy of the model with no impact on the avg loss.
- Num_skips: An increase in the number of skips increases the avg loss value but a substantial increase in the accuracy.
- learning rate: Decrease in the learning rate increases the accuracy.
- num sample: An increase in the number of samples increase the accuracy.
- Max steps: Increase in the max_steps increases accuracy but flattens out(no change) after some value.

2.1.2 The best explored configuration for Cross Entropy

The best configuration appears to be the one with the following parameters having an accuracy of 36.4

- samples: 64
- Skip Window: 8
- Num of skips: 16
- Learning rate: 0.6
- No of steps: 200001

2.2 Noise Contrastive Estimation

Following is the result of analogy task for some configurations of parameters for Noise Contrastive Estimation:

batch_size	skip_window	num_skips	learning rate	num_samples	max_num_steps	loss	accuracy
64	2	4	1	16	200001	0.47	31.70 %
64	8	8	0.6	32	50000	1.09	28.70 %
64	8	16	0.6	32	50000	1.24	33.3 %
64	8	16	1	32	50000	1.05	33.4 %
64	8	16	1	64	50000	1.279	34.2 %
64	8	16	1	32	200001	0.91	34.3 %
64	8	16	0.6	32	100001	1.39	34.6 %
128	4	8	1	64	200001	1.47	30.70 %
128	4	8	1.2	128	200001	3.306	31.70 %
128	8	8	1	32	200001	1.07	28.90 %
128	8	8	1	64	200001	1.07	32.10 %
128	8	8	1	64	200001	1.47	28.90 %
256	8	8	1	32	200001	1.07	33.80 %
512	4	8	1.9	128	200001	3.54	30.90 %

2.2.1 Trends in hyper parameter tuning for Noise Constrastive Estimation

This experimentation is carried out by varying the values of the parameters and keeping other values same and constant.

- Batch size: An increase in batch size increases the accuracy of the model with no impact on the avg loss.
- Skip_window: An increase in skip window increases the accuracy of the model with no impact on the avg loss.

- Num_skips: An increase in the number of skips increases the avg loss value but a substantial increase in the accuracy.
- learning rate: Decrease in the learning rate increases the accuracy.
- num sample: No impact on the accuracy since this is not used calculation of loss function since this is not used.
- Max steps: Increase in the max_steps increases accuracy but flattens out(no change) after some value.

2.2.2 The best configuration for Noise Contrastive Estimation

The best configuration appears to be the one with the following parameters having an accuracy of 34.6

- sample: 64
- Skip Window: 8
- Num of skips: 16
- Learning rate: 0.6
- Number of samples: 32
- No of steps: 100001

3 Words Similar to 'First', 'American', 'Would'

Top 20 similar words according to your NCE and cross entropy model for the words:first, american, would. Please report these in a table. And discuss what kinds of similarities do you notice?

3.1 Cross Entropy

Word	Similar Words
first	leading, ghosted, bodhimandala, dinah, malayans, smearing, deb, yung, secco, tottering, carciofi, heathen, generalissimo, teff, matches, rehired, marvel, collage, clades, bibliotheque
american	kang, hepthalites, intonation, climatological, schizophrenics, rg, rusholme, lutetia, spooky, uncontrollably, fours, cacofonix, paoli, johansen, katan, cholecystectomy, bombed, beats, popery, gn,
would	rotha, shareholder, mammalian, harmonized, colonist, laia, modeling, achelous, anchovies, bothe, overbending, imams, comarques, patten, ecoregions, tiny, brewers, mellitus, flexing, friedrichshain,

There is little similarity between these words for models with high accuracy. With an accuracy of 35 %, there is a 65 % of getting words with little similarity and probably these words fall into those categories.

3.2 Noise Constrastive Estimation

Word	Similar Words
first	vakataka, lexeme, devotes, chow, sweeps, teil, supersoldiers, beausobre, globalists, totalism, gibraltarian, geochemistry, sculpted, meeting, actualization, script, absorptions, tamandua, oblivious, girl,
american	ellipticals, chiefly, undescribed, dagger, smoked, diatessaron, quemada, orphic, perspectives, katahdin, hershkovits, faint, typo, constan, kojeve, appeased, shaybani, techno, sterne, omri,
would	publica, domanick, danubian, fated, hashihito, unglaciated, jambo, superintendents, epd, ithacan, chicano, sul, anschluss, canna, mistress, hatton, wednesdays, amygdalin, legible, safra,

There is little similarity between these words for models with high accuracy. With an accuracy of 35 %, there is a 65 % of getting words with little similarity and probably these words fall into those categories.

4 Justification behind NCE method loss

Q: A summary of the justification behind the NCE method loss. No more than a page. This should be a summary of section 3 (3.1 specifically) in the NCE paper. Explain how the proposed loss function models the probability distribution of a word in its context.
- Noise contrastive estimation stems from the ideology of density reduction to binary classification problem. It reduces the multiclass classification of cross entropy using softmax to a binary classification.
The logistic regression model is fed some ”data” sample from the data distributions and ”noise” samples from noise distribution.
It addresses the problem of cross entropy. How? One of the problems of cross entropy is the normalisation factor that in turn depended on the size of vocabulary. NCE allows us to fit model that are not normalised.
The training data including ”good” data and ”noise” is fetched from the unigram disctrution of the vocabulary. This is so because (a) it is easy to calculate its probability and (b) it does not have a 0 value for any data point.
The sample can be a part of ”data” sample or ”noise” sample. Let us assume that the noise samples are k times more frequent that data samples.
Therefore, probability that this sample is from the data sample given by

$$\boxed{\mathrm{P}^h(D = 1|w, \theta) = \frac{P^h_\theta(w)}{(P^h_\theta(w)+kP_n(w))}}$$

NCE aims at solving the problems created by cross entropy involving the size of the vocabulary.
The derivative of NCE[from paper] involves sum over k noise samples instead of entire vocabulary and hence NCE is independent of the vocabulary size and directly proportional to the noise samples.
Essentially what NCE does is that it calculates scoring function for positive case and negative case. For the positive case, it wants the scoring function to return a high value. For negative case, it wants the scoring function to return a low value.

Intuition behind NCE:

- NCE wants that Score(positive) ¿ Score(Negative)
- NCE wants S(positive) to dominate
- How to achieve this?
- Score(negative) must be small

- 1 - Score(negative) must be large
- Score(Positive) is large
- The scoring function aims at maximising the sum of S(Positive) and S(negative)
- Or minimising the negative of the sum of S(Positive) and S(Negative)

This is how NCE helps is producing a good model based on its scoring function.