Assignment 1 - Word2Vec Skip-gram model

Prasanna Patil
CSA, IISc,
Banaglore
patilk@iisc.ac.in

Abstract

This document contains the final report of assignment 1. The task was to train a skip-gram based Word2Vec model as described by Mikolov in (Mikolov et al., 2013). This report contains charts for various hyper-parameter configurations of skip-gram architecture. It shows the performance of learned embedding on SimLex-999 (Hill et al., 2014) evaluation benchmark and comparison of word analogy task.

1 Task 1

In this task, the word2vec skip-gram model was implemented using tensorflow. The core logic is implemented by hand and tensorflow is used only for training of the model using gradient descent optimization. The convergence of word2vec is defined in terms of loss on training data.

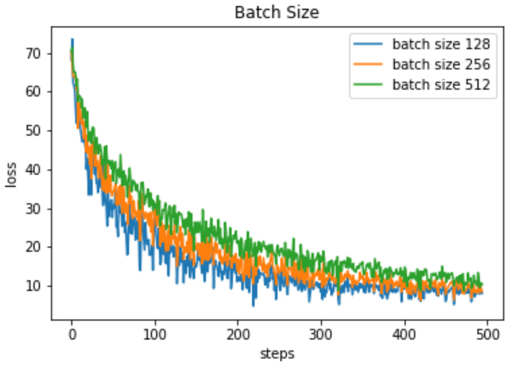
preprocessing of data Following preprocessing procedures were applied to the training data before training the model.

- All the words were converted to lower case and only alphabetic words were considered.
- The default word tokenizer exposed by nltk corpus interface was used to extract words.
- All words appearing less than 5 times in entire corpus were ignored.
- The subsampling and negative sampling probabilities were calculated as described in (Mikolov et al., 2013).

The training data was then converted to pair of words using subsampling method and for given window size. All models were then trained on this preprocessed training data.

1.1 Comparison for different values of hyperparameters

Batch Size Following plot shows the convergence of word2vec architecture for different batch sizes. The batch sizes considered for this task are **128, 256, 512**.



(a) Figure 1

Figure 1: Figure: Training loss for different batch sizes.

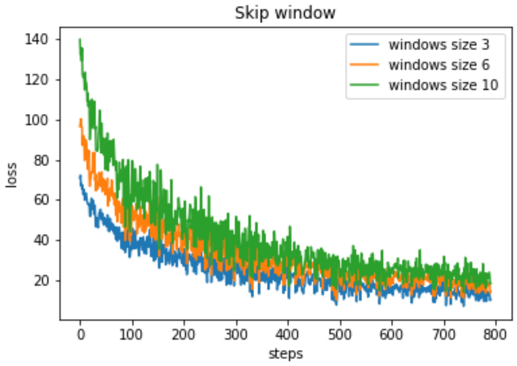
Skip Window Following plot shows the convergence of word2vec architecture for different skip window sizes. The skip window sizes considered for this task are **3, 6, 10**.

Negative Samples Following plot shows the convergence of word2vec architecture for different size of negative samples. The negative samples considered for this task are **8, 12, 20**.

Embedding Dimension Following plot shows the convergence of word2vec architecture for different dimensions of embedding. The embedding dimension considered for this task are **128, 256, 300**.

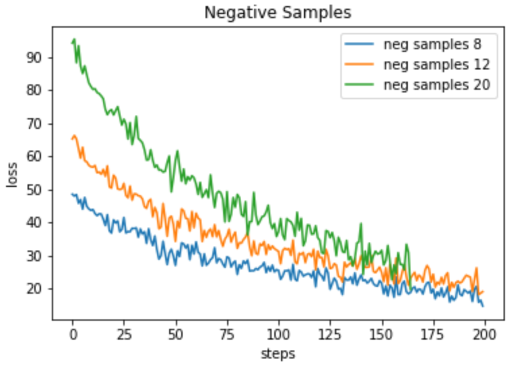
1.1.1 Final model

Based upon above results, three variants were considered for final model training. Above results show the convergence during early training and



(a) Figure 2

Figure 2: Figure: Training loss for different skip window sizes.



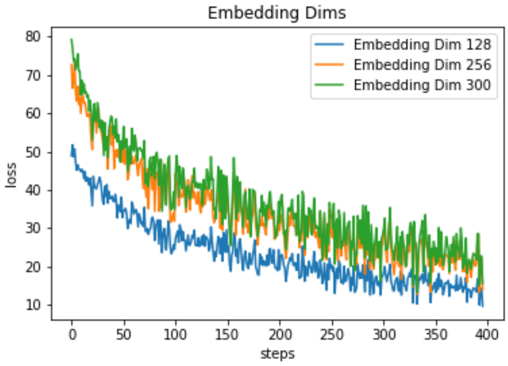
(a) Figure 3

Figure 3: Figure: Training loss for different negative sample sizes.

need not behave same when model is trained for longer time to minimize the convergence loss at the end. Note that the batch size doesn't seem to have much affect on convergence and hence same batch size is used to train final models. The skip window is also kept to 3 for all 3 models, as it leads to early convergence. However, this also may happen due to less training data available. Due to lack of resources skip window was kept to 3 for early convergence of model.

Three models considered for final training are:

- Embedding dims :- 256, Negative samples :- 8, Batch size :- 256
- Embedding dims :- 300, Negative samples :- 8, Batch size :- 256
- Embedding dims :- 300, Negative samples :- 10, Batch size :- 256



(a) Figure 3

Figure 4: Figure: Training loss for different embedding dimensions.

All 3 models were tested using word similarity on simlex-999 evaluation dataset.

1.2 SimLex-999 Evaluation

The results of the final 3 models for SimLex-999 task are shown in table below. The performance metric used for evaluation is Spearman's correlation coefficient which is rank based correlation coefficient. Correlation coefficient was calculated for the cosine similarity between embedding of a pair of words generated by model and the similarity value given by SimLex-999 dataset.

Note that models were trained for 10 epochs only and checkpoint was created for every 2 epochs. The table shows best Spearman correlation coefficient achieve by any of them. The model performing best on this evaluation set was finalized for next task.

Table 1: SimLex-999 performance

Model	Spearman Correlation
NCE-300-256-8	0.113
NCE-256-256-8	0.0841
NCE-300-256-10	0.157

2 Task 2

2.1 Word Analogy task

Performance of the final model on word analogy task was carried out by using question words [1] set. The final performance was evaluated for different

[1] code.google.com/p/word2vec/source/browse/trunk/questions-words.txt

top-k values. That is, we look for top k neighbors of given analogy to find resulting word. The performance of final model for different values of k is shown above. Here, the model is performing quite worse as performance is less than 1%. This could be happening because of smaller dataset size. After preprocessing the corpus, it contains only 10k words which is very small vocabulary size.

Table 2: Word Analogy Quantitative

K-Value	Top-K Accuracy
20	1.53%
10	0.28%
5	0.094%
1	0%

However, upon further investigation, it was found that model is indeed learning analogical reasoning through words. For example, look at following word analogies:

Table 3: Word Analogy Qualitative

Word 1	Word 2	Word 3	Word 4
Man	Engineering	Female	Secretariat
Finance	Bank	Court	Smuggling
Banker	Bank	Professor	Science

Upon giving first 3 words as input, it was found that the 4th word appears in Top-10 neighbors of the vector calculated as $vec(word1) - vec(word2) + vec(word3)$.

2.2 Biases

Upon investigating the neighboring words associated with specific words, following biases were observed (The value encapsulated in parenthesis is cosine similarity between two words):

- The word $income$ is more biased towards word $male(0.20052858)$ than with word $female(0.112244435)$. Similar case was seen for word $finance$, $science$ and $crime$.

- The word $nations$ is also more associated with word $developing(0.25341165)$ than with word $developed(0.1958274)$. Similarly, $nations$ is more close to $western$ than with $asian$.

- The word $money$ is more close to word $poor(0.1521132)$ than with word $rich(0.121930875)$.

- The word $government$ is more close to the word $welfare(0.2324206)$ than with word $development(0.17713012)$.

These biases can be explained because of the training data which is reuters articles. Since reuters is European news agency the words are more aligned the way this news is generated.

3 References

References

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.