Word2vec & Sentiment Analysis

Pingxuan Huang
15307130283

December 27, 2017

Contents

1 Abstract

To analysis sentence, sometimes we need to transform words into vectors, so that we could apply mathematical algorithm on them. In this experiment we will use Word2vec model to achieve this transformation. After getting the presentation of each word, we will use them to do the sentiment analysis.

2 Introduction

The meaning of a word could be figured out from words around it, so a word could be represented as a vector with the help of the contexts in which the word is. In this experiment, we will use skip-grams(SGD) model to extract contextual information.

In this experiment, each word will be represented as tow vectors (one for '*contexs word*' and one for '*center word*',these two vectors will be similar if the train set is very large!). In the Skip-grams model, we will go through all words in the training set and use them as center word, then the model will predict the nearest 2*m words(contexts with window length as m).To be specific, the model will multiply the *output vectors* of every words (negative-sampling model will use some of them) by the *center vector* of the center word, then it will use the Softmax function to calculate the probability of each word. Because the context of the center word is known, the prediction may be wrong, and we will use the difference between the prediction and the reality to modify *word vectors*. Details of training process could be found in following section.

3 Implementation

We use SGD(Stochastic gradient descent) to train the model: based on the loss-function, we can calculate the gradient of each vector and modify them by taking a step in the negative gradient direction.

mathematical details

The answer of question *a,b,c,d* are shown on 1a1b.jpg,1c,jpg,1d.jpg

Coding

Details could be found in *word2vec.py*. There are some points I want to mention: first, to improve the efficiency as much as possible, I have tried my best to transform loop-operation into matrix- operation, but there left one loop in the end of function *negSamplingCostAndGradient*(), it is because one word may be sampled several times in one window; second, the *broadcasting* function of numpy make the transposition of one dimensional vector useless; third, in order to prevent the data becomes too large(or too small) to deal with, we need to do normalization on each row first.

4 Experiment

To test the vectors we have trained, we can use them to do the *Sentimental Analysis*. This task will train a softmax regressor with SGD. The codes for training have been given, and we need to add a function to calculate the mean of all *word vectors* in a sentence as the sentence feature(the coding details could be find in *softmaxreg.py*). To deal with overfitting, we need to add a regularization item to the training process. And we need to try different parameters to search for the best one, details as following:

basic experiment

Both the *word vectors* and the choice of regularization parameter will influence the result. To be specific, both the dimension of the word vectors and the window length used in training process will make difference on the representation of the same word.

In this part, I trained many word vectors sets with different window lengths and different dimensions. After choosing one of them, I did two serious of experiments: in the first serious I will use large-scale regularization parameters(in practice, I set them as $1e-5$,$1e-4$...1,10); in the second serious I will test parameters around the best parameters of the first serious. Result could be found in section *Result & Analysis*.

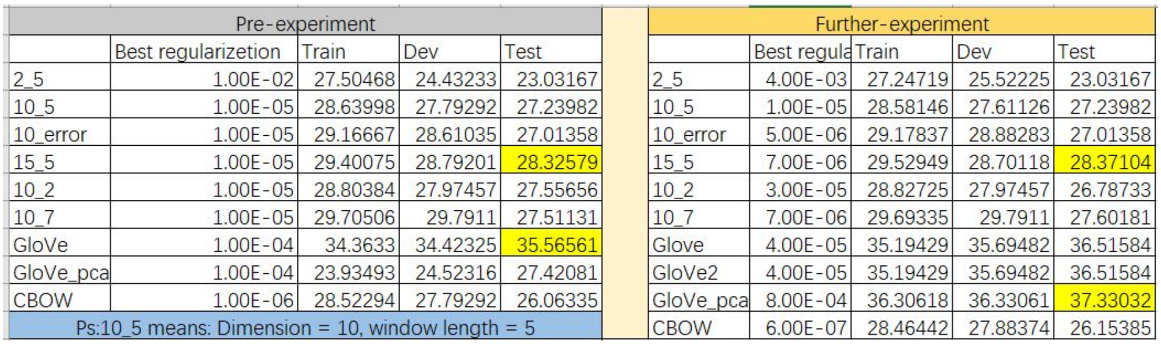
further experiment

Except experiments with *skip-grams* model, I also tried other models:

First, I trained a set of word vectors with the CBOW model (coding details related to cbow model could be found in *word2vec.py*; mathematical details can be found in CBOW.jpg). Second, I downloaded the word vectors of GloVe model[1](trained with Wikipedia) and used the 50-d data to do the experiment; third, I used PCA to reduce the dimension of 100-d GolVe word vectors into 50-d and I used it to do the experiment again. Result and analysis could be found in section *Result & Analysis*.

5 Result & Analysis

Result of all experiments is as follow:



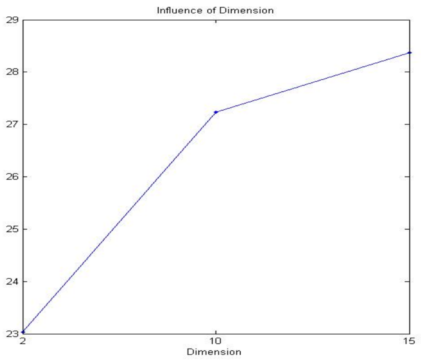
Pre-experiment				
	Best regularizetion	Train	Dev	Test
2_5	1.00E-02	27.50468	24.43233	23.03167
10_5	1.00E-05	28.63998	27.79292	27.23982
10_error	1.00E-05	29.16667	28.61035	27.01358
15_5	1.00E-05	29.40075	28.79201	28.32579
10_2	1.00E-05	28.80384	27.97457	27.55656
10_7	1.00E-05	29.70506	29.7911	27.51131
GloVe	1.00E-04	34.3633	34.42325	35.56561
GloVe_pca	1.00E-04	23.93493	24.52316	27.42081
CBOW	1.00E-06	28.52294	27.79292	26.06335
Ps:10_5 means: Dimension = 10, window length = 5				

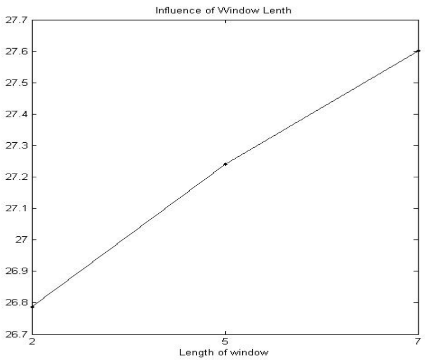
Further-experiment				
	Best regula	Train	Dev	Test
2_5	4.00E-03	27.24719	25.52225	23.03167
10_5	1.00E-05	28.58146	27.61126	27.23982
10_error	5.00E-06	29.17837	28.88283	27.01358
15_5	7.00E-06	29.52949	28.70118	28.37104
10_2	3.00E-05	28.82725	27.97457	26.78733
10_7	7.00E-06	29.69335	29.7911	27.60181
Glove	4.00E-05	35.19429	35.69482	36.51584
GloVe2	4.00E-05	35.19429	35.69482	36.51584
GloVe_pca	8.00E-04	36.30618	36.33061	37.33032
CBOW	6.00E-07	28.46442	27.88374	26.15385

Detailed information could be found in *data/result_of_all.xlsx*

Result of basic experiment

I will analysis the influence of dimensions and window length there. Because the 'further experiments' based on the experiment conducted on large-scale regularization parameters, the data of further experiments are more meaningful, and I will only analysis these data.





It is obvious that as the dimension of vectors and the window length increase, the result becomes better. Contexts can represent the information of center word, therefore as the window length increase, more information can be extracted. However we could speculate that this phenomenon can not continue

[1] https://nlp.stanford.edu/projects/glove/

after window length increases into a certain number because most related information is near the word itself. The same as window length, in other words, as the dimension increases, more information could be represented, but the Calculating cost will also increase significantly.

Result of further experiment

CBOW

The result of CBOW model is worse then skip-grams. CBOW uses contexts word to predict center word. In the training process, this model will see all contexts words the same, it means all contexts words will change same gradient. This operation is time-saving but coarse and may introduce inaccuracy into model.

GloVe

Generally speaking, the work what GloVe does is to record co-occurrence of contexts and words, and reducing the dimension of the co-occurrence matrix.

It is obvious the result of GloVe is much better than skip-grams model, there are two reasons: first, GloVe will take all contexts into consideration but SGD will use only some of them, which means GloVe will extract more information(global information); second, the word vectors of GloVe is 50-d, and in previous part, we have found the result will be better as the dimension increase.

PCA on 100-d GloVe

The result of this experiment is the best and there are also two reasons:first, 100 dimension vectors will record the most information; second, PCA will remain the most important information of a word and it can also denoise.