COMP 6714
Project 2
Word Embeddings Report

Hao Chen
z5102446

Abstract

This project is to generate the word embeddings for ADJ(adjectives), and find synonyms relationship for them. The similarity between two word can be measured by the cosine similarity measure.

1. Introduction

The key idea in this program is training the ship-gram model using the given corpus. The key part of word embeddings is the training data. In the implementation, mini-batches are generated on demand, and used by the model to update the word vectors.

2. Process tokenization

The first part we need to do is the pre-process part. Basically I did 2 things, which are punctuation remove and space replacement and case folding. Since word tokenization in English always has ‘space’ character and punctuation to separates sentences, doing punctuation remove is to reducing the training set for word embeddings. Case folding means all words are changed into lowercase during the preprocessing. Then I used the stop words to reduce some stop words in the original documents.

3. Parameters

Batch_size: 64
Skip_window: 3
Num_samples:4
Vocabulary_size: 5000 commonest words
Learning_rate: 0.002
Number_of_negative_sample: 64
Embedding_dimensions: 200-dimensions

Number_of_iterations: 100001
Loss function: sampled_softmax_loss
Optimization Method: AdamOptimizer

4. About how to generate batch the aim of

As for the parameter skip_window = 3 and num_samples = 4 is used in this project, this means each centre word randomly choose 4 words from 6 context words to forecast it.

5. About how to select the hyperparameters

About how to select the hyerparameters, I think this is the key to get the good performance when computing the k adjective words. In this case we use AdamOptimizer with a small learning rate. The most important parameter should be learning

Learning rate	Average hit
0.01	2.56
0.02	2.78
0.03	2.64

rate, number of negative samples, and vocabulary size. So following the specification of the demo, we start to test the code with 0.001.

Though the performance is not good, it still can be seen that

Vocabulary size	Number of negative samples	Average hits
15000	128	2.32
15000	64	2.66
15000	96	2.20

0.02 is the winner among all three numbers.

Then we can choose vocabulary size and number of negative sample.

So we choose 64 of the vocabulary size.

Vocabulary size	Number of negative samples	Average hits
15000	64	2.10
10000	64	2.35
5000	64	2.77

Finally, the hyperparameters can be chosen as follows:

Batch_size: 64
Skip_window: 3
Num_samples: 4
Vocabulary_size: 5000 most common words
Learning_rate: 0.002
Number_of_negative_sample: 64

5. Conclusion

Training Word Embeddings with NLP process is quite helpful to this project. There are a lot of synonym relationship can be maintained to show the final results.

6. Part of the result

Writing Processed Data file(Success)

Training model with 10 iterations
Initializing the model
Nearest to our: window, related, statement, entries, leads, sustain, orders, risen, blocking, surge,

Nearest to much: margaret, humphreys, concluded, storage, ranging, wells, seeks, realistic, stocks, almost,

Writing Embedding file(Success)

Reading Trained Model
Reading Trained Model(Success)
Average Hits on Dev Set is = 2.750000