SkipGram
Final Project Report

Gladys Roch / Aurélien Houdbert

February 2021

1 Introduction

The subject of word similarity is difficult but very interesting. Many recent papers such as [2] who reached a Spearman Correlation score of 0.37 on Simlex-999 in 2013 using a skipgram model or again [1] who managed to score a 0.56 Spearman Correlation using neural machine translation in 2014.

In this first assignment we implemented a skip-gram model with negative sampling. The main idea is to learn two different representations of words : as word and as context. If we denote x_w a word seen as a word and y_c a word seen as context and the model variable I where $I = 1$ means that *x and y is a valid pair of word and context.*

$$p(I = 1|x, y, \theta) = \frac{1}{1 + e^{-x_w * y_c}}$$

2 Derivation

Let's denote W and C the two matrices of shape (N, k) where N is the size of the vocabulary and k is the size of the embedding. Each line of W and C respectively represent the embedding of a word in the vocabulary seen as word and context respectively. To find optimal parameters for W and C we will try to optimize the following expression where $ns(X)$ denotes the negative samples of X :

$$\underset{W,C}{\operatorname{argmin}} - \sum_{X,Y} \left[log\left(\frac{1}{1 + e^{-X_W Y_C}}\right) + \sum_{Z \in ns(X)} log\left(\frac{1}{1 + e^{X_W Z_C}}\right) \right]$$

For each X word and Y context we can describe the loss as follow :

$$Loss(X, Y) = -log\left(\frac{1}{1 + e^{-X_W Y_C}}\right) - \sum_{Z \in ns(X)} log\left(\frac{1}{1 + e^{X_W Z_C}}\right)$$

We will perform stochastic gradient descent to update W and C parameters. Given a word X, a context Y and negative samples Z for X we need to compute the partial derivatives according to X, Y and Z.

$$\begin{aligned} \frac{\partial Loss(X, Y)}{\partial X} &= -\frac{Y}{1 + e^{XY}} + \sum_{Z \in ns(X)} \frac{Z}{1 + e^{-XZ}} \\ \frac{\partial Loss(X, Y)}{\partial Y} &= -\frac{X}{1 + e^{XY}} \\ \frac{\partial Loss(X, Y)}{\partial Z} &= \frac{X}{1 + e^{-XZ}} \end{aligned} \tag{1}$$

We will the update W and C parameters using stochastic gradient descent. let's denote α the learning rate :

$$\begin{aligned}
X_w^{(t+1)} &= X_w^{(t)} - \alpha \frac{\partial Loss(X,Y)}{\partial X} \\
Y_c^{(t+1)} &= Y_c^{(t)} - \alpha \frac{\partial Loss(X,Y)}{\partial Y} \\
Z_c^{(t+1)} &= Z_c^{(t)} - \alpha \frac{\partial Loss(X,Y)}{\partial Z}
\end{aligned} \tag{2}$$

3 Results

3.1 Loss

Using a train set of 5000 sentences and the following set of hyperparameters :
- nEmbed=200,
- negativeRate=2,
- winSize = 5,
- minCount = 5,
- learning_rate = 0.05,
we get a loss of **1.77** after 3 epochs.

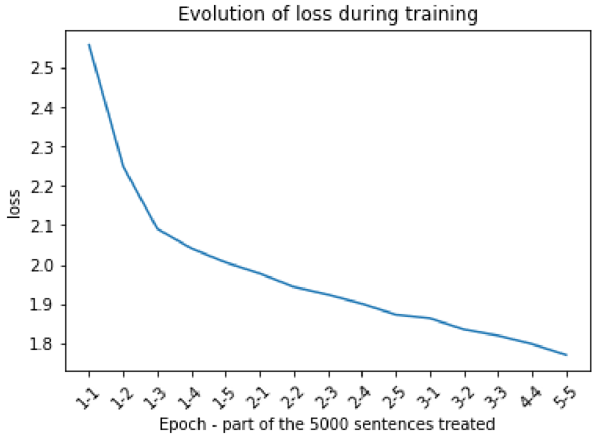

Figure 1 – Evolution of the loss during training with hyperparameters nEmbed=200, negative sampling of 2, window size of 5, minCount=5, learning rate of 0.05.

To evaluate the performance of our model we compute the Spearman correlation with the simlex dataset.

For this model we get a correlation of **0.072** using the cosine similarity and the Spearman correlation. Contrary to the Pearson correleation, the Spearman correlation does not assume that both datasets are normally distributed. Like other correlation coefficients, this one varies between -1 and +1 with 0 implying no correlation.

3.2 Words Similarity

The following table prensent some similarity values for common english adjectives :

Similarity Results		
Word 1	Word 2	Similarity
old	new	0.181
fast	slow	0.126
big	small	0.061
president	Obama	0.500
president	Bush	0.632
president	Romney	0.100
USA	Boston	0.241
France	Paris	0.122
USA	Paris	0.021

We can see that adjectives that can be replaced by each other in a sentence whilst keeping the sentence coherent, have a high similarity (old/new). Words naturally more associated we each other have higher similarity : President/ Obama is a closer pair than President/Romney as Romney was never president of the US. The pair County/City give the same results.

4 Additional Experiments

We ran several small experiments to identify suitable hyper-parameters. Because of computing time, we did not really perform a grid search but we studied several parameters independently. For each parameter we trained on 3 epochs with a learning rate of 0.05 on a corpus of 5000 sentences.

We first decided to study the impact of the representation of each word vector on the correlation coefficient between our embedding prediction and SimLex data. We tried 3 different representation. The first one using only W embedding, the second one using only C embedding and the third one taking the average embedding between W and C. In all three cases we used a negative sample size of 5, a window size of 5, and embedding size of dimension 100 and we trained for 3 epochs on 5000 sentences with a learning rate of 0.05.

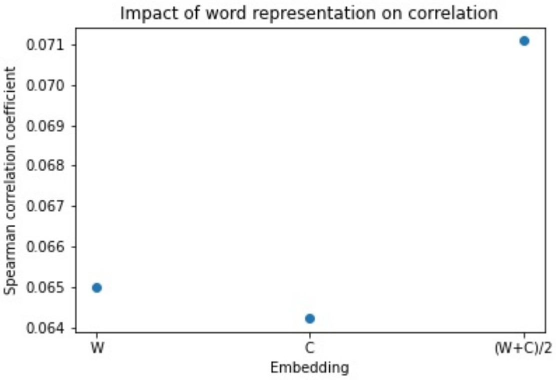

Figure 2 – Impact of embedding on correlation. **W** is the word embedding matrix and **C** is he context embedding matrix.

From this figure, the best embedding representation seems to be the average of W and C that gave the highest Spearman's Correlation coefficient. Therefore we will use this result for our next experiment.

The second experiment we ran was to study the impact of negative sample size on the correlation coefficient between our embedding prediction and SimLex data. We used the same parameters as the previous experiment for negative saple size of 2, 5 and 10. We also used the best embedding representation according to the previous experiment which is the average between W and C.

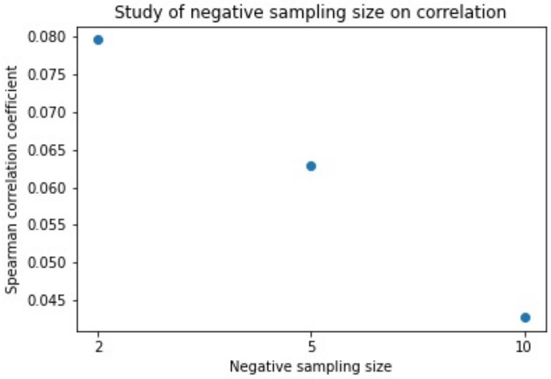

Figure 3 – Impact of negative sample size on correlation.

From this figure, in the condition of training mentioned above, the best Spearman's correlation coefficient was achieve using a negative rate of 2. It would be interesting to see

Références

[1] Felix Hill et al. "Embedding Word Similarity with Neural Machine Translation." In : (2014). url : https://arxiv.org/abs/1412.6448.

[2] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space." In : (2013). url : https://arxiv.org/abs/1301.3781.