Learning Word Embeddings without Context Vectors

Alexey Zobnin
Yandex,
National Research University
Higher School of Economics,
Moscow, Russia
azobnin@hse.ru

Evgenia Elistratova
Moscow State University,
Moscow, Russia
evg3307@yandex.ru

Abstract

Most word embedding algorithms such as word2vec or fastText construct two sort of vectors: for words and for contexts. Naive use of vectors of only one sort leads to poor results. We suggest using indefinite inner product in skip-gram negative sampling algorithm. This allows us to use only one sort of vectors without loss of quality. Our “context-free” `cf` algorithm performs on par with SGNS on word similarity datasets.

1 Introduction

Vector representation of words are widely used in NLP tasks. Two approaches to word embeddings are usually contrasted: implicit (word2vec-like) and explicit (SVD-like). Implicit models are usually faster and consume less memory than their explicit analogues.

Typically, word embedding algorithms produce two matrices both for “words” and “contexts”. Usually, contexts are the words themselves. It is believed that word and context vectors cannot be equated to each other.

In practice, however, only the vectors of one sort are considered. For example, typical solutions to word similarity or analogy problems use only the inner products of word vectors.

We present a modified skip-gram negative sampling algorithm that produces related word and context vectors. One may say that some components of our word and context vectors are equal, while other components have different signs. Another point of view is to say that word and context vectors are completely equal, but the inner product between them is indefinite. This relation was suggested by the properties of explicit SVD embeddings.

2 Preliminaries

We briefly recall the skip-gram negative sampling (SGNS) algorithm implemented in popular programs `word2vec` (Mikolov et al., 2013a,b) and `fastText` (Bojanowski et al., 2017). Given row vectors of a current word w, its context c_0 and negative context samples $c_1, \dots, c_k$, SGNS algorithm computes the loss

$$\mathcal{L} = -\ln \sigma(wc_0^T) - \sum_{j=1}^{k} \ln \sigma(-wc_j^T), \quad (1)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Starting from random initial approximation of word vectors, the algorithm uses stochastic gradient descent (SGD) for optimization. The intuition is the following: the word vector should be similar to the true context vector and dissimilar to random negative contexts. Due to the properties of $\sigma(x)$, the gradient and update formulas look simple.

Let n be the size of the vocabulary, and d be the dimension of embeddings. Let W and C be $n \times d$ matrices of word (respectively, context) vectors written in rows. The SGNS loss (1) depends only on elements of WC^T, and does not depend on W or C separately. The optimal solution of SGNS is not unique: the transformation $W \mapsto WS$, $C \mapsto C\left(S^{-1}\right)^T$ for an invertible $d \times d$-matrix S, gives an equivalent solution. However, such transformation could change the inner products between word vectors dramatically if S is not orthogonal. Nevertheless, SGNS produces a “good” solution without any regularization. This phenomenon is not well understood, and can be seen as “implicit regularization” of SGD.

Levy and Goldberg showed that the implicit matrix $M = WC^T$ tends to the shifted PMI matrix $\mathcal{M}$ when d is sufficiently large (Levy and Goldberg, 2014). Here $\mathcal{M}_{i,j} = \mathrm{PMI}(w_i, w_j) - \log k$,

Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 244–249
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

where

$$\mathrm{PMI}(w_i, w_j) = \log \frac{P(w_i w_j)}{P(w_i)P(w_j)}$$

is the pointwise mutual information of the pair of words w_i, w_j, and k is the amount of negative samples. Thus, SGNS can be considered as an implicit matrix factorization problem.

We recall that a (compact) singular value decomposition (SVD) of a real-valued $m \times n$-matrix M of rank r is a decomposition $M = U\Sigma V^T$, where U and V are respectively $m \times r$ and $n \times r$ matrices with orthogonal columns, and Σ is $r \times r$ diagonal matrix with non-zero singular values on diagonal. We refer to (Golub and Van Loan, 2012) for details. SVD embeddings are obtained from truncated SVD decomposition $M \approx U_d \Sigma_d V_d^T$, where only d top singular values and vectors are kept. The common way is to take $W = U_d\sqrt{\Sigma_d}$ and $C = V_d\sqrt{\Sigma_d}$.

3 Main result

First of all, we make an observation about SVD embeddings. We shall use it further to modify the SGNS algorithm.

3.1 SVD of real symmetric matrices

Proposition. *Let M be a real-valued symmetric $n \times n$-matrix.*

1. *There exists a decomposition $M = WDW^T$, where $D = \mathrm{diag}(\pm 1)$.*
2. *There exists a compact SVD $M = U\Sigma V^T$ such that $V = UD$.*
3. *If all singular values of M are different, then all singular value decompositions of M have this form*[1].

In other words, the corresponding columns of U and V either coincide, or differ in sign.

Proof. All eigenvalues of a real symmetric matrix are real. Moreover, M has an eigendecomposition $M = C\Lambda C^T$, where $\Lambda = \mathrm{diag}(\lambda_i)$ is a diagonal matrix of eigenvalues, and C is an orthogonal matrix. Let r be the rank of M. We may remove columns in C and Λ that correspond to zero eigenvalues and assume that C is $n \times r$-matrix and Λ is $r \times r$-matrix with non-zero eigenvalues on diagonal. Note that non-zero singular values σ_i of M are absolute values of λ_i.

1. Let $\Sigma = \mathrm{diag}(\sigma_i)$ be a diagonal matrix of non-zero singular values. Then $\Lambda = \Sigma D = \sqrt{\Sigma} D \sqrt{\Sigma}$, where
$$D_{ii} = \begin{cases} 1, \text{ if } \lambda_i > 0, \\ -1, \text{ if } \lambda_i < 0. \end{cases}$$
Now take $W = C\sqrt{\Sigma}$.
2. Write M as $C\Sigma DC^T$ and take $U = C$, $V = CD$.
3. If all singular values are different, then SVD is determined uniquely up to a simultaneous change of signs in some columns of U and V. Take the SVD constructed above and note that the relation $V = UD$ is preserved after these transformations. □

We denote by q the amount of -1 in D. Due to the Sylvester's law of inertia, it is uniquely determined by M and equals to the amount of negative eigenvalues of M.

3.2 Negative eigenvalues of word relation matrices

Consider an $n \times n$-matrix $M = (f(w_i, w_j))$ describing the relation between the words w_i and w_j. For example, $f(w_i, w_j)$ may be the shifted PPMI, as suggested in (Levy and Goldberg, 2014):

$$f(w_i, w_j) = \max\left(0, \mathrm{PMI}(w_i, w_j) - \log k\right).$$

As a rule, the relation f is symmetric, and hence M is symmetric too.

Let's look at the SVD embeddings obtained from this matrix. Let $M = U\Sigma V^T$ be an SVD, and $M \approx M_d = U_d \Sigma_d V_d^T$ be its truncated approximation. Then symmetric SVD embeddings are $W = U_d\sqrt{\Sigma_d}$ and $C = V_d\sqrt{\Sigma_d}$. In the following, we will naturally assume that top d singular values of M are different and non-zero: all real cases are just like that. By the proposition we have that some columns of W and C coincide, while the others differ in sign. As a consequence, we obtain that for such SVD embeddings word and context vectors are equally good in applied problems, because inner products $(w_i w_j)$ and $(c_i c_j)$ are the same. We would like to construct implicit SGNS-like embeddings with similar properties.

[1] This is also valid for a more general case when equal singular values of M correspond to the eigenvalues of the same sign.

The amount of negative eigenvalues of M measures the deviation from the positive definiteness in some sense. To estimate it, we construct shifted PPMI matrices for Wikipedia corpora in three different languages (English, French and Russian). Each corpus contains 1M articles. We measure the amount of negative eigenvalues among top by magnitude eigenvalues. The results for $k = 1$ (pure PPMI matrix, no shift) and for $k = 5$ are presented in Tables 1 and 2, respectively. We see that the rate of negative eigenvalues is about 11–13% for $k = 1$ and about 7–8% for $k = 5$. Surprisingly, this rate actually does not depend on the language.

corpus \ dimension	**100**	**200**	**300**
English Wikipedia	15	27	39
French Wikipedia	10	22	34
Russian Wikipedia	11	22	31

Table 1: Amount of negative eigenvalues in the truncated SVD of PPMI matrix ($k = 1$, i. e., no shift).

corpus \ dimension	**100**	**200**	**300**
English Wikipedia	9	16	24
French Wikipedia	7	15	22
Russian Wikipedia	7	14	20

Table 2: Amount of negative eigenvalues in the truncated SVD of shifted PMI matrix with $k = 5$.

3.3 Context-free SGNS algorithm

Recall that SGNS loss depends on the inner products of words and contexts. Let's fix the matrix

$$D = \mathrm{diag}(\underbrace{-1, \ldots, -1}_{q}, \underbrace{1, 1, \ldots, 1}_{p=d-q})$$

specifying indefinite inner product in the embedding space. The amount of minus ones in this matrix, q, will be a hyperparameter of our algorithm. Next, we equate word and context vectors to each other. This corresponds to the implicit factorization WDW^T instead of WC^T.

Thus, we replace the initial SGNS loss (1) with

$$\mathcal{L}_q = -\ln \sigma(wDw_0^T) - \sum_{j=1}^{k} \ln \sigma(-wDw_j^T).$$

The solution of this new optimization problem is also not unique. Replacing W with WS for any matrix S that preserves D, i. e., such that $SDS^T = D$, yields another solution. Nevertheless, the solution set in our case is "smaller" in some sense than in the pure SGNS case, where S may be any invertible matrix.

Since we got rid of context vectors, we call this algorithm *context-free SGNS*[2] and denote it `cf`.

4 Experimental results

4.1 Training setup

We train word embeddings on the English Wikipedia dump. We preprocess this dump using gensim.corpora.wikicorpus package[3] and take a subsample of 100K articles. Our corpus consists of approximately 175M words with $n \approx 312000$. We use fastText[4] skipgram mode with the default values of parameters and without ngrams (`-maxn 0`) as a vanilla SGNS implementation. We implement our model by modifying the C++ implementation of fastText[5]. We learn `cf` vectors of dimension $d = 100$ and $100 + q$ for $q = 0, 5, 10, 15, 20, 25$. Vectors of dimension $100 + q$ were projected to 100 "positive" components.

4.2 Datasets

Datasets for word similarity evaluation consist of pairs of words rated by humans. We use the following well-known English similarity datasets: MEN-3k (Bruni et al., 2014), MTurk-287 (Halawi et al., 2012), RW-STANFORD (Luong et al., 2013), SimLex-999 (Hill et al., 2015), SimVerb-3500 (Gerz et al., 2016), VERB-143 (Baker et al., 2014), and WS-353 (Finkelstein et al., 2002) splitted into similarity and relatedness parts (Agirre et al., 2009). We use the evaluation code (Faruqui and Dyer, 2014) for computing Spearman rank correlation ρ between the similarity scores and the annotated ratings.

4.3 Results

Table 3 shows the results for word vectors of dimension 100 with q negative components in D. The best results are written in bold, and the best `cf` results are underlined. We see that $q = 0$, i. e., the case of positive semidefinite approximation, has the worst performance. The better perfor-

[2] Not to be confused with context-free grammars.

[3] https://radimrehurek.com/gensim/corpora/wikicorpus.html.

[4] https://fasttext.cc/.

[5] https://github.com/jen1995/fastText/tree/sgns-loss.

Dataset	SGNS	$q = 0$	$q = 5$	$q = 10$	$q = 15$	$q = 20$	$q = 25$
MEN-TR-3k	**.7314**	.6790	.7234	.7185	.7165	.7124	.7092
MTurk-287	.6668	.6335	.6574	.6604	.6615	.6609	**.6725**
RW-STANFORD	.3801	.2862	.4094	.4202	**.4223**	.4151	.4090
SIMLEX-999	**.3323**	.2520	.3141	.3194	.3126	.3098	.3226
SimVerb-3500	**.2022**	.1317	.1890	.1948	.1975	.1942	.1965
VERB-143	.3000	.2995	.3323	.3304	.3498	**.3834**	.3595
WS-353-REL	**.6648**	.6142	.6372	.6594	.6389	.6306	.6296
WS-353-SIM	**.7612**	.7079	.7508	.7538	.7525	.7506	.7332
average	.4380	.3700	.4346	**.4384**	.4382	.4345	.4340

Table 3: Word similarity for $d = 100$.

Dataset	SGNS	$q = 0$	$q = 5$	$q = 10$	$q = 15$	$q = 20$	$q = 25$
MEN-TR-3k	.7314	.6790	.7333	.7348	.7351	**.7371**	.7370
MTurk-287	.6668	.6335	.6612	.6697	.6719	**.6746**	.6645
RW-STANFORD	.3801	.2862	.4015	**.4091**	.4069	.4022	.3999
SIMLEX-999	**.3323**	.2520	.3138	.3127	.3199	.3166	.3235
SimVerb-3500	**.2022**	.1317	.1926	.1917	.1894	.1905	.1914
VERB-143	.3000	.2995	**.3362**	.3148	.3141	.3285	.3315
WS-353-REL	.6648	.6142	.6673	.6714	**.6765**	.6718	.6741
WS-353-SIM	.7612	.7079	**.7627**	.7476	.7577	.7554	.7480
average	.4380	.3700	.4382	**.4394**	**.4394**	.4392	**.4394**

Table 4: Word similarity for $p = 100$ (only “positive” components of $p + q$-dimensional `cf` vectors were taken).

mance of `cf` is achieved at $q = 10$ or $q = 15$. This argees with empirical results of Subsection 3.2. In general, `cf` is either on par with SGNS, or slightly loses. In the second experiment we learn `cf` vectors of higher dimension $100 + q$ and projected them to 100 “positive” components. These results are shown in Table 4. Starting from $q = 5$, they are slightly better than SGNS.

5 Related work

There were several attempts to establish a connection between word and context vectors. In (Li et al., 2017) a PMI matrix is approximated by a positive semidefinite matrix of the form WW^T. Dependencies between word and context vectors for word2vec SGNS model were studied in (Mimno and Thompson, 2017).

In (Allen et al., 2018) it is suggested that word and context vectors should be conjugated in the complex space. Our model can be reformulated in terms of complex vectors too in the case $p = q$, but we prefer to stay in reals. One of the problems with complex embedding is that $\sigma(z)$ is not holomorphic in the whole complex plane, and should be replaced with $\sigma(|z|)$ or $\sigma(\operatorname{Re} z)$. Complex-valued embeddings are also discussed in (Trouillon et al., 2016), where a complex decomposition like $M = \operatorname{Re} U\Lambda\bar{U}^T$ is suggested.

In (Assylbekov and Takhanov, 2019) it was conjectured that context vectors are reflections of word vectors in half the dimensions. Our result is similar, but we suggest using lower amount of reflections. Perhaps this difference is due to the fact that we consider shifted PPMI matrices, while they consider pure PMI matrices.

Embeddings in hyperbolic space with Minkowski metric was suggested in (Leimeister and Wilson, 2018). In (Soleimani and Matwin, 2018) there is an erroneous statement that a thresholded PMI matrix, as well as any symmetric matrix, has SVD decomposition $U\Sigma U^T$: in fact it is true only for positive semidefinite matrices.

Non-uniqueness of SGNS solutions was addressed in (Fonarev et al., 2017) and (Mu et al., 2018), resulting in new models. Implicit regularization of SGD in neural networks and in matrix factorization problems was studied in (Gunasekar et al., 2017; Neyshabur et al., 2017; Ma et al., 2018), but SGNS loss was not considered directly in these works.

6 Conclusion

We proposed cf, an alternative to SGNS algorithm that do not use context vectors. Instead, indefinite inner product between word vectors is used. Our algorithm shows similar results compared to SGNS.

The phenomenon of implicit regularization of SGNS, as well as the problem of finding the linguistic interpretation of "negative" components of word vectors in our algorithm, deserve further investigation.

Acknowledgements

The authors thank Anna Potapenko for fruitful discussions and anonymous referees for valuable comments.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: NAACL-2009*, pages 19–27. ACL.

Carl Allen, Ivana Balažević, and Timothy Hospedales. 2018. What the vec? Towards probabilistically grounded embeddings. *arXiv:1805.12164*.

Zhenisbek Assylbekov and Rustem Takhanov. 2019. Context vectors are reflections of word vectors in half the dimensions. *arXiv:1902.09859*.

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 278–289.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors. org. In *Proceedings of 52nd Annual Meeting of ACL: System Demonstrations*, pages 19–24.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Alexander Fonarev, Oleksii Grinchuk, Gleb Gusev, Pavel Serdyukov, and Ivan Oseledets. 2017. Riemannian optimization for skip-gram negative sampling. In *Proceedings of the 55th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 2028–2036.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182.

Gene H Golub and Charles F Van Loan. 2012. *Matrix computations*, volume 3. JHU press.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. 2017. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Matthias Leimeister and Benjamin J Wilson. 2018. Skip-gram word embeddings in hyperbolic space. *arXiv:1809.01498*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Shaohua Li, Jun Zhu, and Chunyan Miao. 2017. Psdvec: A toolbox for incremental and scalable word embedding. *Neurocomputing*, 237:405–409.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the 17th Conference on Computational Natural Language Learning*, pages 104–113.

Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. 2018. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878.

Cun Mu, Guang Yang, and Zheng Yan. 2018. Revisiting skip-gram negative sampling model with rectification. *arXiv:1804.00306*.

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. 2017. Geometry of optimization and implicit regularization in deep learning. *arXiv:1705.03071*.

Behrouz Haji Soleimani and Stan Matwin. 2018. Spectral word embedding with negative sampling. In *32nd AAAI Conference on Artificial Intelligence*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.