Revisiting Representation Degeneration Problem in Language Modeling

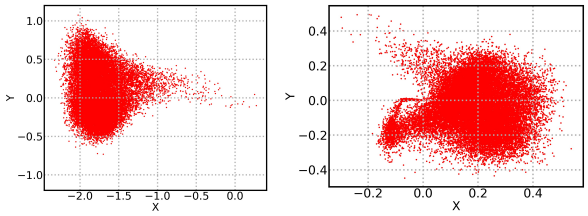
Zhong Zhang[1], **Chongming Gao**[1], **Cong Xu**[1], **Rui Miao**[2], **Qinli Yang**[1], **Junming Shao**[1]*
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Guizhou University, Guiyang, China
{zhongzhang,congxu}@std.uestc.edu.cn,
chongming.gao@gmail.com,
miaorui93@163.com,
{qinli.yang,junmshao}@uestc.edu.cn

Abstract

Weight tying is now a common setting in many language generation tasks such as language modeling and machine translation. However, a recent study reveals that there is a potential flaw in weight tying. They find that the learned word embeddings are likely to degenerate and lie in a narrow cone when training a language model. They call it the *representation degeneration problem* and propose a cosine regularization to solve it. Nevertheless, we prove that the cosine regularization is insufficient to solve the problem, as the degeneration is still likely to happen under certain conditions. In this paper, we revisit the representation degeneration problem and theoretically analyze the limitations of the previously proposed solution. Afterward, we propose an alternative regularization method called Laplacian regularization to tackle the problem. Experiments on language modeling demonstrate the effectiveness of the proposed Laplacian regularization.

1 Introduction

Language modeling is a fundamental task in natural language processing, applications include machine translation (Bahdanau et al., 2015; Vaswani et al., 2017), image captioning (Vinyals et al., 2015; Xu et al., 2015) and speech recognition (Yu and Deng, 2016), to name a few. In the era of deep learning, a general model architecture usually contains a word embedding layer as input, multiple layers to encode word context as a fixed-size hidden state, and a softmax layer to transform the hidden-state into a categorical distribution of the next word (Merity et al., 2018; Yang et al., 2018; Gong et al., 2018; Wang et al., 2019; Gao et al., 2019). While in practice, the parameters of the embedding layer and the softmax layer are usually shared, which is called weight tying (Inan et al., 2017; Press and Wolf, 2017).



(a) Vanilla (b) CosReg

Figure 1: Illustration of the degeneration phenomenon from (Gao et al., 2019). (a). Word embeddings trained from vanilla Transformer. (b). Word embeddings trained with cosine regularization.

Despite the improvements from weight tying, a recent work (Gao et al., 2019) discovers that, with weight tying, the learned word embeddings are positively correlated and spread in a narrow cone as visualized in Figure 1(a). A similar phenomenon is observed in Gong et al. (2018). Thus, the semantic expressiveness of word embeddings is limited. They call it the *representation degeneration problem*. To tackle the problem, the authors propose a cosine regularization that minimizes the cosine similarities between any two word embeddings to enlarge the aperture of the cone. They show that it improves the language modeling performance and eases the degeneration as visualized in Figure 1(b).

However, we argue that the cosine regularization might not be the best choice for solving this problem, and the reasons are: i) The cosine regularization minimizes similarities between any two word embeddings without considering whether they are semantically close or not. But we wish two words with similar semantics stay close in the embedding space. ii) Although the cosine regularization improves language generation performance, it does not fundamentally solve the representation degeneration problem. We prove that the degeneration still exists when there exists a certain regularization

* Corresponding author http://dm.uestc.edu.cn

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 518–527
November 16 - 20, 2020.

structure. Finally, we analyze the general condition of degeneration and show that there still are many low-frequency words that meet the condition and thus degenerate. Therefore, we argue that the degeneration is still likely to happen even with cosine regularization.

Motivated by these issues, we propose an alternative Laplacian regularization to tackle the representation degeneration problem. As the distributional hypothesis (Harris, 1954) states: two words that occur in similar contexts tend to have similar meanings. The general idea of Laplacian regularization is to minimize the squared Euclidean distance between two word embeddings when they have large context similarity. In contrast to cosine regularization, Laplacian regularization prevents minimizing all similarities of word pairs indiscriminately. Although the Laplacian regularization does not theoretically solve the degeneration problem either, we empirically demonstrate that it achieves better performance in most cases of language modeling experiments, and word embeddings are less likely to degenerate.

In summary, the main contributions of our work are listed as follows.

- We revisit the representation degeneration problem and theoretically analyze the limitations of the previously proposed cosine regularization solution.
- We propose an alternative Laplacian regularization to tackle the representation degeneration problem. We show that it eases the degeneration to an extent comparing with cosine regularization.
- We conduct experiments on language modeling task to demonstrate the effectiveness of our method.

2 Representation Degeneration Problem

In this section, we introduce the notations and review the representation degeneration problem.

Given a vocabulary of words (indices) $\mathcal{V} = \{1, ..., N\}$, and a text corpus represented as a sequence of words $\boldsymbol{y} = (y_1, ..., y_M)$, where $y_i \in \mathcal{V}$. The joint probability of sequence $\boldsymbol{y}$ is factorized into a product of conditional probabilities using the chain rule.

$$P(Y = \boldsymbol{y}) = \prod_{t=1}^{M} P(Y_t = y_t | Y_{<t} = \boldsymbol{y}_{<t}), \quad (1)$$

where $\boldsymbol{y}_{<t}$ denotes the first $t-1$ words in $\boldsymbol{y}$. Current neural language models encode variable-length context as a fixed-size hidden state denoted as $\boldsymbol{h}_i$. The conditional probability is calculated by the softmax function, and the model is trained by minimizing the negative log-likelihood loss as follows.

$$L_{NLL} = -\frac{1}{M} \sum_{i=1}^{M} \log \frac{\exp(\boldsymbol{w}_{y_i}^T \boldsymbol{h}_i)}{\sum_{l=1}^{N} \exp(\boldsymbol{w}_l^T \boldsymbol{h}_i)}, \quad (2)$$

where $\boldsymbol{w}$ is the parameter of the softmax layer. When using weight tying, $\boldsymbol{w}_l$ is the embedding for the l-th word.

Next, we investigate the optimization process of word embeddings. We follow the analysis in Gao et al. (2019) and only focus on the extreme case of a non-appeared word $\boldsymbol{w}_N$ in the following analysis, since the analysis can be extended to the case of rarely appeared words by applying Theorem 3 in Gao et al. (2019). Assume $y_i \neq N$ for all i, which means the N-th word with embedding $\boldsymbol{w}_N$ does not appear in the corpus. Under the log-likelihood maximization objective and fixing all other parameters, we write the objective function for optimizing variable $\boldsymbol{w}_N$ as follows.

$$\min_{\boldsymbol{w}_N} \frac{1}{M} \sum_{i=1}^{M} \log(\exp(\boldsymbol{w}_N^T \boldsymbol{h}_i) + G_i), \quad (3)$$

where $G_i = \sum_{l=1}^{N-1} \exp(\boldsymbol{w}_l^T \boldsymbol{h}_i)$ and can be considered as a constant. Let $\boldsymbol{v}$ be a uniformly negative direction of $\boldsymbol{h}_i$, i.e., $\boldsymbol{v}^T \boldsymbol{h}_i < 0$ for all i. It is easy to see that the optimal solution of Eq. (3) can be achieved by setting $\boldsymbol{w}_N^* = \lim_{k \to \infty} k \cdot \boldsymbol{v}$ and the minimum objective value is bounded by $\frac{1}{M} \sum_{i=1}^{M} \log(G_i)$. The authors prove that such a uniformly negative direction $\boldsymbol{v}$ exists if and only if the convex hull of the hidden states does not contain the origin. They discuss that the condition is very likely to hold, especially when layer normalization is applied. We further observe that the condition holds almost for sure in actual language modeling, even without layer normalization.

From the above analysis, we have an intuition for the representation degeneration problem. We can see that the embedding $\boldsymbol{w}_N$ can be optimized along any uniformly negative direction to infinity. As the set of uniformly negative direction is convex, $\boldsymbol{w}_N$ is likely to lie in a convex cone and move to infinity during optimization. This conclusion also applies to the case of rarely appeared words to a large extent (Gao et al., 2019). As most words in

natural language are low-frequency words according to Zipf's law, the learned word embeddings tend to degenerate and lie in a narrow cone, which limits the model's semantic expressiveness. Notably, Gong et al. (2018) also show that the learned word embeddings overly encode word frequency information rather than semantic information, which implicitly supports the existence of the degeneration problem.

3 Solutions to The Problem

In this section, we first introduce the solution proposed in Gao et al. (2019). Then we theoretically analyze the limitations of the previously proposed method. Finally, we propose an alternative regularization to tackle the problem.

3.1 Cosine Regularization

As word embeddings tend to lie in a narrow cone, a straightforward solution is to enlarge the aperture of the cone, which is defined as the maximum angle between any two boundaries of the cone. However, for the ease of optimization, Gao et al. (2019) proposes to minimize the cosine similarities between any two word embeddings. The overall loss is the typical negative log-likelihood loss plus the regularization term as follows.

$$L = L_{NLL} + \gamma \frac{1}{N^2} \sum_{i}^{N} \sum_{j \neq i}^{N} \hat{\boldsymbol{w}}_i^T \hat{\boldsymbol{w}}_j, \qquad (4)$$

where $\hat{\boldsymbol{w}} = \boldsymbol{w}/||\boldsymbol{w}||$ is the normalized direction of $\boldsymbol{w}$, and $\gamma > 0$ is a hyperparameter.

The cosine regularization minimizes the similarities of all word pairs indiscriminately, which might not be a good idea, especially when two words are semantically close and correlated. More importantly, this regularization technique is theoretically insufficient to solve the representation degeneration problem. We will show that in the following analysis.

Following the previous study, we write the objective function with cosine regularization term w.r.t. a non-appeared word $\boldsymbol{w}_N$ as follows.

$$\min_{\boldsymbol{w}_N} \frac{1}{M} \sum_{i=1}^{M} \log(\exp(\boldsymbol{w}_N^T \boldsymbol{h}_i) + G_i) + \hat{\boldsymbol{w}}_N^T \hat{\boldsymbol{w}}_C, \qquad (5)$$

where $\hat{\boldsymbol{w}}_C = \frac{2\gamma}{N^2} \sum_{j=1}^{N-1} \hat{\boldsymbol{w}}_j$ and can be considered as a constant. As the cosine regularization term is a function of $\boldsymbol{w}_N$, setting $\boldsymbol{w}_N = \lim_{k\to\infty} k \cdot \boldsymbol{v}$ may not achieve the optimal solution of Eq. (5), which prevents word embeddings from lying in the cone. However, we find that the degeneration still exists in certain cases. To show that, we first define the uniformly negative direction cone as follows.

Definition 1. *Let $\mathcal{C}$ denote the uniformly negative direction cone of hidden states, i.e., $\mathcal{C} = \{\boldsymbol{v} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\} | \boldsymbol{v}^T \boldsymbol{h}_i < 0, \forall i = 1, ..., M\}$.*

Note that $\mathcal{C}$ is a set of vectors, we use $-\mathcal{C}$ to denote the set of the negative vectors for convenience. Since the cosine regularization term is the projection length of vector $\hat{\boldsymbol{w}}_C$ in direction of unit vector $\hat{\boldsymbol{w}}_N$, the objective value depends on $\hat{\boldsymbol{w}}_C$. The following theorem states that the degeneration exists when $\hat{\boldsymbol{w}}_C$ lies in certain directions.

Theorem 1. *If the uniformly negative direction cone $\mathcal{C}$ is not empty, and $\hat{\boldsymbol{w}}_C$ is in $-\mathcal{C}$, then the optimal solution of Eq. (5) can be achieved by setting $\boldsymbol{w}_N^* = \lim_{k\to\infty} k \cdot \boldsymbol{v}^*, \exists \boldsymbol{v}^* \in \mathcal{C}$. The minimum objective value is $\frac{1}{M} \sum_{i=1}^{M} \log(G_i) - ||\hat{\boldsymbol{w}}_C||$.*

Proof. Since $\hat{\boldsymbol{w}}_C$ is in $-\mathcal{C}$, it is easy to check that there exists a uniformly negative direction vector $\boldsymbol{v}^*$ that is in $\mathcal{C}$ and has the opposite direction of $\hat{\boldsymbol{w}}_C$. Note that the two terms in Eq. (5) have bounded minimum values $\frac{1}{M} \sum_{i=1}^{M} \log(G_i)$ and $-||\hat{\boldsymbol{w}}_C||$, which can be both simultaneously achieved by setting $\boldsymbol{w}_N^* = \lim_{k\to\infty} k \cdot \boldsymbol{v}^*$. □

We argue that the condition in Theorem 1 is likely to happen in language modeling. Under the log-likelihood maximization objective, each appeared word embedding $\boldsymbol{w}_{y_i}$ tends to be optimized to maximize the correlation between it and its hidden state $\boldsymbol{h}_i$. Note that $\hat{\boldsymbol{w}}_C$ represents the average direction of all appeared words. Therefore, $\hat{\boldsymbol{w}}_C$ is likely to negatively correlate with a uniformly negative direction $\boldsymbol{v}$ and lie in $-\mathcal{C}$. From Theorem 1, we can see that the degeneration still exists as long as $\hat{\boldsymbol{w}}_C$ has an opposite direction of $\mathcal{C}$. Nevertheless, this condition still seems strong. We will give a general condition under which the degradation exists. We first provide a lemma as follows.

Lemma 1. *Let $\boldsymbol{w}_N^*$ be the optimal solution of Eq. (5). If $\boldsymbol{w}_N^*$ is in $\mathcal{C}$, then $||\boldsymbol{w}_N^*|| = \infty$ and the minimum objective value is $\frac{1}{M} \sum_{i=1}^{M} \log(G_i) + \hat{\boldsymbol{w}^*}_N^T \hat{\boldsymbol{w}}_C$.*

Proof. We prove the lemma by contradiction. Suppose there is an optimal solution $\boldsymbol{w}_N$ with a finite

length that is in $\mathcal{C}$. Let $\boldsymbol{w}_N^* = \lim_{k\to\infty} k \cdot \boldsymbol{w}_N$ and $L(\cdot)$ denote the objective function of Eq. (5). Since $\hat{\boldsymbol{w}}_N^T \hat{\boldsymbol{w}}_C = \hat{\boldsymbol{w}}^{*T}_N \hat{\boldsymbol{w}}_C$, it is easy to check that the objective value $L(\boldsymbol{w}_N) > L(\boldsymbol{w}_N^*) = \frac{1}{M}\sum_{i=1}^{M} \log(G_i) + \hat{\boldsymbol{w}}^{*T}_N \hat{\boldsymbol{w}}_C$, which raises the contradiction. □

Denote $Z_i = \sum_{l=1}^{N} \exp(\boldsymbol{w}_l^T \boldsymbol{h}_i)$. Based on Lemma 1, we give the following theorem.

Theorem 2. *If the uniformly negative direction cone $\mathcal{C}$ is not empty, and $\mathbb{E}(\frac{G}{Z}) < \exp\Big(-\frac{4\gamma(N-1)}{N^2}\Big)$, then the optimal solution of Eq. (5) is in $\mathcal{C}$.*

Proof. Suppose there are two cases of optimal solution: $\boldsymbol{w}_N^{*'} \in \mathcal{C}$ and $\boldsymbol{w}_N^{*''} \notin \mathcal{C}$. From Lemma 1, we have $||\boldsymbol{w}_N^{*'}|| = \infty$, and $L(\boldsymbol{w}_N^{*'})$ is upper bounded. We compare the maximum value of $L(\boldsymbol{w}_N^{*'})$ and the minimum value of $L(\boldsymbol{w}_N^{*''})$.

$$
\begin{aligned}
& L(\boldsymbol{w}_N^{*'})_{\max} - L(\boldsymbol{w}_N^{*''})_{\min} \\
=& \frac{1}{M}\sum_{i=1}^{M} \log(G_i) + ||\hat{\boldsymbol{w}}_C|| - \\
& \Big(\frac{1}{M}\sum_{i=1}^{M} \log(\exp(\boldsymbol{w}_N^{*''T}\boldsymbol{h}_i) + G_i) - ||\hat{\boldsymbol{w}}_C||\Big) \\
=& \frac{1}{M}\sum_{i=1}^{M} \log\frac{G_i}{Z_i} + 2||\hat{\boldsymbol{w}}_C||. \qquad (6)
\end{aligned}
$$

Note that $0 \le ||\hat{\boldsymbol{w}}_C|| \le \frac{2\gamma(N-1)}{N^2}$. By letting Eq. (6) < 0, we have

$$
\frac{1}{M}\sum_{i=1}^{M} \log\frac{G_i}{Z_i} < -\frac{4\gamma(N-1)}{N^2}. \qquad (7)
$$

We write Eq. (7) as expectation form and apply Jensen's inequality.

$$
\begin{aligned}
\mathbb{E}\big(\log\frac{G}{Z}\big) &< -\frac{4\gamma(N-1)}{N^2} \qquad (8) \\
\log\mathbb{E}\big(\frac{G}{Z}\big) &< -\frac{4\gamma(N-1)}{N^2} \qquad (9) \\
\mathbb{E}\big(\frac{G}{Z}\big) &< \exp\Big(-\frac{4\gamma(N-1)}{N^2}\Big). \qquad (10)
\end{aligned}
$$

Eq. (10) gives the condition of $L(\boldsymbol{w}_N^{*'})$ is constantly smaller than $L(\boldsymbol{w}_N^{*''})$, under which the optimal solution is in $\mathcal{C}$. □

Note that the vocabulary size N is usually large in language modeling, e.g., 10000 for Penn Treebank data set and over 30000 for WikiText-2 data set. Suppose $\gamma = 1$, the right side of the inequality has a value of 0.9996 and 0.9999, respectively. It makes the inequality very likely to hold in practice, especially for low-frequency words, and we will empirically demonstrate it in the experiment. Based on Theorem 2 and Lemma 1, we argue that the cosine regularization is insufficient to solve the representation degeneration problem.

3.2 Laplacian Regularization

The distributional hypothesis (Harris, 1954) is a common assumption in various NLP tasks, which states that two words that occur in similar contexts tend to have similar meanings. We borrow this idea and propose an alternative Laplacian regularization technique. The overall objective is as follows.

$$
\begin{aligned}
L &= L_{NLL} + \lambda\frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} ||\boldsymbol{w}_i - \boldsymbol{w}_j||^2 s_{ij} \\
&= L_{NLL} + \lambda\frac{1}{N^2}\mathrm{Tr}(\boldsymbol{W}^T\boldsymbol{L}\boldsymbol{W}), \qquad (11)
\end{aligned}
$$

where $\lambda > 0$ is a hyperparameter, and s_{ij} is a similarity weight that measures the context similarity between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$ is called graph Laplacian matrix. $\boldsymbol{D}$ is a diagonal matrix whose entries are column or row sums of $\boldsymbol{S}$. s_{ij} can be calculated by any similarity function, for example, cosine similarity is used in this study.

$$
s_{ij} = \frac{\boldsymbol{h}_i^T \boldsymbol{h}_j}{||\boldsymbol{h}_i|| \cdot ||\boldsymbol{h}_j||}. \qquad (12)
$$

Note that we detach $\boldsymbol{h}$ from the computational graph to cut off the back propagation gradient flow in implementation.

However, computing the Laplacian regularization term with full vocabulary words is computationally expensive. Another issue is that computing s_{ij} needs to sample appropriate contexts for word $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. To address these issues, we compute the Laplacian regularization term in a stochastic mini-batch way. Specifically, let $\boldsymbol{H} \in \mathbb{R}^{B\times T\times D}$ be the hidden state matrix before the softmax layer, where B is the batch size and T is the sequence length in one batch. We only compute words that are predicted by these $B \times T$ hidden states and use the corresponding hidden states as contexts to calculate s_{ij}. Here we use this simple way to calculate s_{ij} only for the ease of implementation. Though, one could design a sophisticated strategy to incorporate extra knowledge by selecting word pairs and manipulating similarity weights.

By contrast, Laplacian regularization minimizes the similarities of word pairs discriminately. It makes word embeddings with similar contexts closer in Euclidean space, which better captures the semantic correlation of words. More importantly, we show that it is less affected by the representation degeneration problem. We first write the objective function with Laplacian regularization term w.r.t. a non-appeared word $\boldsymbol{w}_N$ as follows.

$$\min_{\boldsymbol{w}_N} \frac{1}{M}\sum_{i=1}^{M}\log(\exp(\boldsymbol{w}_N^T\boldsymbol{h}_i)+G_i) + \frac{\lambda}{N^2}\sum_{j=1}^{N}||\boldsymbol{w}_N-\boldsymbol{w}_j||^2 s_j. \quad (13)$$

Theorem 3. *Let $\boldsymbol{w}_N^*$ be the optimal solution of Eq. (13) and assume $s_j > 0$. For any $\boldsymbol{w}_N^*$, $||\boldsymbol{w}_N^*|| < \infty$.*

Proof. We prove the theorem by contradiction. Suppose $\boldsymbol{w}_N$ is an optimal solution with $||\boldsymbol{w}_N|| = \infty$. It is easy to check that $\sum_{j=1}^{N}||\boldsymbol{w}_N-\boldsymbol{w}_j||^2 s_j = \infty$. Because the first term in Eq. (13) has bounded minimum value, the overall objective value is infinite. However, the objective function exists finite values, which raises the contradiction. □

Note that when using cosine similarity to calculate s_{ij}, it does not guarantee positive weights. However, we observe that in actual language modeling experiments, it is nearly impossible to have $\boldsymbol{h}_i^T\boldsymbol{h}_j \leq 0$, which further suggests the existence of $\mathcal{C}$. From the above theorem, we can see that the optimal solution $\boldsymbol{w}_N^*$ cannot go along with any direction to infinity. However, it is difficult to give a quantitative analysis of whether the optimal solution will lie in $\mathcal{C}$ or not. We only give a qualitative analysis here. We first write the derivative of Eq. (13) w.r.t. $\boldsymbol{w}_N$ as follows.

$$\frac{\partial L}{\partial \boldsymbol{w}_N} = \frac{1}{M}\sum_{i=1}^{M}\frac{\exp(\boldsymbol{w}_N^T\boldsymbol{h}_i)\cdot\boldsymbol{h}_i}{\exp(\boldsymbol{w}_N^T\boldsymbol{h}_i)+G_i} + \frac{2\lambda}{N^2}\sum_{j=1}^{N}(\boldsymbol{w}_N-\boldsymbol{w}_j)s_j. \quad (14)$$

Qualitatively, the gradient direction involves three directions: $\boldsymbol{h}_i$, $\boldsymbol{w}_N$ and $-\boldsymbol{w}_j$. Suppose that $\boldsymbol{h}_i$ dominates the gradient direction, when applying gradient descent, the optimal solution is likely to fall into the uniformly negative direction cone $\mathcal{C}$. However, as $\boldsymbol{w}_j$ is an appeared word, it is likely to positively correlated with $\boldsymbol{h}_i$ under the log-likelihood maximization objective. Therefore, $-\boldsymbol{w}_j$ could have the opposite direction of $\boldsymbol{h}_i$ and serve as a counterbalance to ease the degeneration effect. As for $\boldsymbol{w}_N$, it can be considered as a regularization to prevent having too large parameters. We empirically demonstrate the effectiveness of our method in the following experiments.

4 Experiments

In this section, we conduct experiments on language modeling task to demonstrate the effectiveness of our method.

4.1 Language Modeling

We conduct language modeling experiment on two widely used data sets of Penn Treebank (PTB) (Mikolov et al., 2010) and WikiText-2 (WT2) (Merity et al., 2017). We use two recent works as our baselines: the AWD-LSTM model[1] (Merity et al., 2018) and the AWD-LSTM-MoS model[2] (Yang et al., 2018), which achieved the state-of-the-art performance. Also, we compare with the cosine regularization technique (Gao et al., 2019), as we are all targeting the same representation degeneration problem.

For experimental settings, we faithfully follow all the settings[3] in AWD-LSTM and AWD-LSTM-MoS. There are no extra hyperparameters in our method except for λ. We set it to 0.01 and 0.001 for PTB and WT2, respectively. For cosine regularization, we set γ to 1 as described in its paper.

It is worth noting that the baseline papers' results are based on an older Pytorch 0.4.1 version, we find that the Pytorch version has a large impact on the language modeling performance since Pytorch $0.4.1$ and > 1.0 have significant differences in implementation. On PTB data set, we can get a better 57.39/54.94 perplexity comparing with 58.34/56.18 by simply switching to a newer Pytorch without other changes. We must point out that building a new model upon the latest codebase, but still borrowing the numbers directly from the baseline paper could be misleading and result in unfair comparison. To this end, all experiments including the baselines are conducted under the

[1] https://github.com/salesforce/awd-lstm-lm

[2] https://github.com/zihangdai/mos

[3] The parameter settings are slightly different between the papers and the Github code. We use the Github configurations since they are consistent with the latest released code.

Data set	Model	#Param	Val.	Test
PTB	(Merity et al., 2018) - AWD-LSTM w.o. finetune	24M	61.49	59.14
	(Gao et al., 2019) - AWD-LSTM-CosReg w.o. finetune	24M	**61.29**	**58.94**
	Ours - AWD-LSTM-LapReg w.o. finetune	24M	61.38	59.07
	(Merity et al., 2018) - AWD-LSTM w.t. finetune	24M	59.54	57.27
	(Gao et al., 2019) - AWD-LSTM-CosReg w.t. finetune	24M	59.48	57.18
	Ours - AWD-LSTM-LapReg w.t. finetune	24M	**58.71**	**56.44**
	(Yang et al., 2018) - AWD-LSTM-MoS w.o. finetune	22M	58.34	56.18
	(Gao et al., 2019) - AWD-LSTM-MoS-CosReg w.o. finetune	22M	58.26	56.18
	Ours - AWD-LSTM-MoS-LapReg w.o. finetune	22M	**57.92**	**55.92**
	(Yang et al., 2018) - AWD-LSTM-MoS w.t. finetune	22M	56.83	54.64
	(Gao et al., 2019) - AWD-LSTM-MoS-CosReg w.t. finetune	22M	56.94	54.73
	Ours - AWD-LSTM-MoS-LapReg w.t. finetune	22M	**56.41**	**54.38**

Table 1: Perplexity on validation and test sets on Penn Treebank. Smaller the perplexity, better the result.

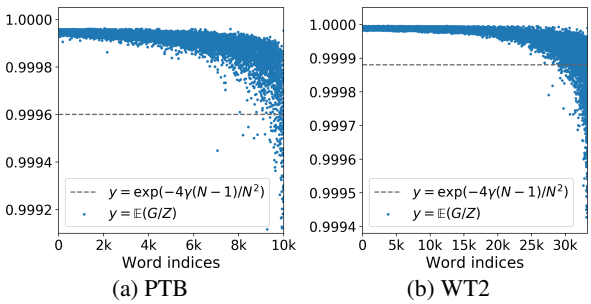
same environment of Pytorch 0.4.1 to make a fair comparison.

The language modeling results on PTB and WT2 data sets are presented in Table 1 and Table 2, respectively. Our method generally outperforms baseline methods with and without finetune. On PTB data set, our method improves the AWD-LSTM and AWD-LSTM-MoS baselines by up to 0.83/0.83 and 0.42/0.26 in terms of valid/test perplexity, respectively. On WT2 data set, our method improves the AWD-LSTM and AWD-LSTM-MoS baselines by up to 0.46/0.04 and 0.47/0.62 in terms of valid/test perplexity, respectively. When compared with cosine regularization, our method equipped with AWD-LSTM is sometimes underperformed. But our method consistently outperforms cosine regularization equipped with AWD-LSTM-MoS by up to 0.53/0.35 and 0.71/0.48 in terms of valid/test perplexity on PTB and WT2 data sets, respectively. Note that we do not change any configuration in baselines but only add regularization terms to the loss function. Thus, the improvements purely come from the regularization, which suggests that they ease the degeneration to an extent. By comparison, Laplacian regularization is generally better than cosine regularization.

To see how the regularization strength λ affects the language modeling performance, we run AWD-LSTM-MoS-LapReg on the large data set WT2 with λ tuned in the order of magnitude $\{1.0, 0.1, 0.01, 0.001, 0.0001\}$. The test perplexities are *non-convergence*, 62.60, 62.88, 62.83, 63.02, respectively. We can see that the perplexity fluctuates in an acceptable range and achieves the best at $\lambda = 0.1$.

4.2 Empirical Study for Theorem 2

We empirically examine whether the condition in Theorem 2 holds in actual language modeling. We calculate $\mathbb{E}\left(\frac{G}{Z}\right)$ from the trained AWD-LSTM-CosReg model on the PTB and WT2 data sets, respectively. As we can see from Figure 2, many low-frequency words' $\mathbb{E}\left(\frac{G}{Z}\right)$ are smaller than $\exp\left(-\frac{4\gamma(N-1)}{N^2}\right)$, especially for the data set with large vocabulary size, which shows that the condition in Theorem 2 is likely to hold in practice. It suggests that the degeneration still exists even with the cosine regularization, which is insufficient to solve the problem.



(a) PTB (b) WT2

Figure 2: $\mathbb{E}\left(\frac{G}{Z}\right)$ on PTB and WT2 data sets, respectively. The word indices are sorted by their frequencies in descending order.

Data set	Model	#Param	Val.	Test
WT2	(Merity et al., 2018) - AWD-LSTM w.o. finetune	33M	68.57	65.39
	(Gao et al., 2019) - AWD-LSTM-CosReg w.o. finetune	33M	68.24	65.54
	Ours - AWD-LSTM-LapReg w.o. finetune	33M	**68.11**	**65.35**
	(Merity et al., 2018) - AWD-LSTM w.t. finetune	33M	67.33	64.30
	(Gao et al., 2019) - AWD-LSTM-CosReg w.t. finetune	33M	**66.75**	**64.13**
	Ours - AWD-LSTM-LapReg w.t. finetune	33M	66.94	64.46
	(Yang et al., 2018) - AWD-LSTM-MoS w.o. finetune	35M	65.92	63.45
	(Gao et al., 2019) - AWD-LSTM-MoS-CosReg w.o. finetune	35M	66.16	63.31
	Ours - AWD-LSTM-MoS-LapReg w.o. finetune	35M	**65.45**	**62.83**
	(Yang et al., 2018) - AWD-LSTM-MoS w.t. finetune	35M	64.31	61.75
	(Gao et al., 2019) - AWD-LSTM-MoS-CosReg w.t. finetune	35M	64.08	61.48
	Ours - AWD-LSTM-MoS-LapReg w.t. finetune	35M	**63.80**	**61.28**

Table 2: Perplexity on validation and test sets on WikiText-2. Smaller the perplexity, better the result.

4.3 Visualization of Word Embeddings

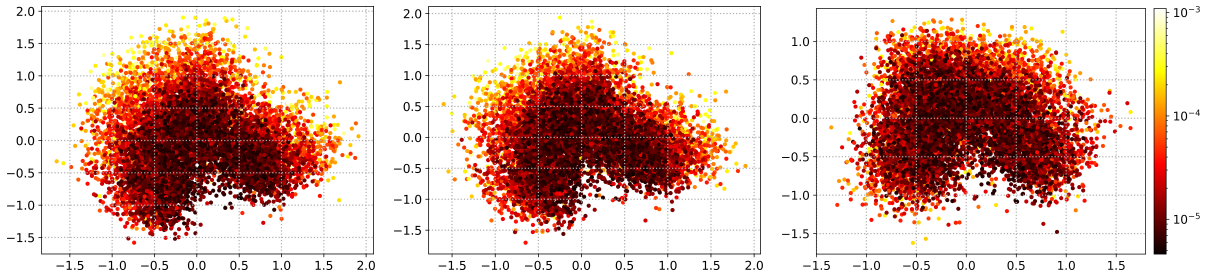
To empirically investigate the effect of regularization techniques on word embeddings, we extract word embeddings trained on PTB data set and project them into 2-dimensional space for visualization. As shown in Figure 3(a), the word embeddings are clustered by their frequencies rather than semantics. The low-frequency words tend to cluster in a local region, which suggests that word embeddings lie in a narrow cone in the embedding space and the degeneration happens. However, when regularization techniques are applied, the learned word embeddings are more uniformly distributed around the origin and the degeneration effect is eased. As we can see from Figure 3(b) and Figure 3(c), the low/high-frequency word embeddings are better mixed, while the Laplacian regularization looks better than others.

5 Discussion and Future Work

From the above study, we analyze the limitations of the cosine regularization and empirically demonstrate the effectiveness of our proposed Laplacian regularization method. However, there is also an issue in it. To this end, we make further discussion in this section. Hopefully, it will provide some inspirations for later researches.

There is one question that must be asked: Does the Laplacian regularization completely solve the representation degeneration problem? Unfortunately, we cannot give a definite positive answer. From the above empirical studies, we have evidence that the Laplacian regularization can ease the degeneration to an extent. However, there is also a failure case, the model cannot converge when the value of λ is set too large. Because if λ is sufficiently large, the regularization term will dominate the objective value and all word embeddings will be optimized to huddle together. The premise of this failure case is that the similarity weights s_{ij} are all positive. Interestingly, we observe that almost similarity weights are positive, even though they are calculated by the cosine function, which further suggests that there may exist some intrinsic mechanism that causes the degeneration phenomenon. We will leave it to future study. Despite this issue, the Laplacian regularization is also a general framework to incorporate the external knowledge of word pair relations like semantic knowledge graph and synonymy/antonymy, which might bring benefits in certain applications.

In addition, we find that the representation degeneration problem is highly related to the softmax bottleneck problem (Yang et al., 2018). As a matter of fact, we consider they are two sides of the same problem. The softmax bottleneck states that a language model's output log-probability matrix should be high-rank for natural language. But the rank is limited by the embedding dimension D and thus the expressiveness of a model is compromised. The softmax bottleneck problem roots in an insufficient embedding dimension D. However,



(a) Vanilla (b) CosReg (c) LapReg

Figure 3: Visualization of word embeddings on PTB data set using PCA. Points are dyed by log-normalized frequencies, the lower the darker.

what really matters is the rank of the embedding matrix rather than the dimension D. As for the representation degeneration problem, it reveals that for a $N \times D$ word embedding matrix, the rank could be smaller than D since word embeddings are correlated and lie in a narrow cone. Thus, the true crux here is about the spectral density distribution of the embedding matrix. There is also evidence (Mu and Viswanath, 2018) that an embedding matrix with more uniformly distributed singular values better improves downstream task performance. Thus, we suggest two lines of researches to enhance the expressiveness of a language model. The first is to learn better expressive word embeddings (Gao et al., 2019; Gong et al., 2018; Wang et al., 2019). The second is to design better expressive output/activation functions (Yang et al., 2018; Ganea et al., 2019; Yang et al., 2019; Kanai et al., 2018; Takase et al., 2018). Nonetheless, we want to clarify that only focusing on the embedding/output layers is far more insufficient for language modeling, since it is the middle layers that provide the major non-linearity which matters most for the expressiveness. Exploring new architectures like the BERT (Devlin et al., 2019) and the Transformer-XL (Dai et al., 2019) is also essential for the future study.

6 Related Work

For neural language modeling, Merity et al. (2018) build an important baseline named AWD-LSTM which applies various regularization techniques to train LSTM. Melis et al. (2018) also achieve similar results with highly regularized LSTMs. Built on AWD-LSTM, Yang et al. (2018) propose the AWD-LSTM-MoS model that achieves significantly lower perplexities by addressing the softmax bottleneck. Gong et al. (2018) find that word embeddings in language modeling are biased towards word frequency and propose an adversarial training scheme to address the problem. Similarly, Wang et al. (2019) introduce an adversarial noise to the embedding layer while training language models. Recently, another promising trend of language model that is built upon the self-attention mechanism like the Transformer-XL (Dai et al., 2019) rapidly emerges.

Gao et al. (2019) first point out the representation degeneration problem in training neural language models when applying the weight tying technique. A similar phenomenon can also be observed in Gong et al. (2018), though it does not explicitly target the degeneration problem. Furthermore, Ethayarajh (2019) observes that the contextualized representations are also anisotropic and lie in a narrow cone in all non-input layers. Recently, Wang et al. (2020) propose a new method that explicitly controls the singular value distribution to tackle the representation degeneration problem. We also consider that the softmax bottleneck problem (Yang et al., 2018) is highly related to the representation degeneration problem. There are a series of works (Ganea et al., 2019; Yang et al., 2019; Kanai et al., 2018; Takase et al., 2018) that follow this line of research.

The Laplacian regularization has been widely used in various fields like semi-supervised learning (Belkin and Niyogi, 2004), face recognition (Cai et al., 2007), graph embedding (Yu et al., 2020), and metric learning (Hoi et al., 2010), to name a few. However, to the best of our knowledge, it has not been applied for regularizing the word embedding matrix yet. We are probably the first to propose the Laplacian regularization on word embeddings.

7 Conclusion

In this paper, we study the *representation degeneration problem* that is first pointed out by Gao et al. (2019). We theoretically analyze the limitations of the previously proposed solution. Afterward, we propose an alternative Laplacian regularization method to tackle the problem. Experiments on language modeling demonstrate the effectiveness of our method. In the future study, we will try to further investigate this problem from the perspective of spectral density of embedding matrix.

Acknowlegments

This work is supported by the National Natural Science Foundation of China (61976044), Fundamental Research Funds for the Central Universities (ZYGX2019Z014), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062), National key research and development program (2016YFB0502300), The Belt and Road Fund on Water and Sustainability of the State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering [grant number 2019], State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin (IWHR-SKL-201911).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Mikhail Belkin and Partha Niyogi. 2004. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1-3):209–239.

Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang. 2007. Learning a spatially smooth subspace for face recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 55–65.

Octavian Ganea, Sylvain Gelly, Gary Bécigneul, and Aliaksei Severyn. 2019. Breaking the softmax bottleneck via learnable monotonic pointwise nonlinearities. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2073–2082.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *Proceedings of the 7th International Conference on Learning Representations*.

ChengYue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. FRAGE: frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, pages 1334–1345.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Steven CH Hoi, Wei Liu, and Shih-Fu Chang. 2010. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(3):1–26.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *Proceedings of the 5th International Conference on Learning Representations*.

Sekitoshi Kanai, Yasuhiro Fujiwara, Yuki Yamanaka, and Shuichi Adachi. 2018. Sigsoftmax: Reanalysis of the softmax bottleneck. In *Advances in Neural Information Processing Systems*, pages 284–294.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proceedings of the 6th International Conference on Learning Representations*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *Proceedings of the 6th International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *11th Annual Conference of The International Speech Communication Association*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the 6th International Conference on Learning Representations*.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Sho Takase, Jun Suzuki, and Masaaki Nagata. 2018. Direct output connection for a high-rank language model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4599–4609.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Dilin Wang, ChengYue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6555–6565.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *Proceedings of the 8th International Conference on Learning Representations*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2048–2057.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *Proceedings of the 6th International Conference on Learning Representations*.

Zhilin Yang, Thang Luong, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Mixtape: Breaking the softmax bottleneck efficiently. In *Advances in Neural Information Processing Systems*, pages 15922–15930.

Dong Yu and Li Deng. 2016. *automatic speech recognition*. Springer.

Zhongjing Yu, Zhong Zhang, Haoran Chen, and Junming Shao. 2020. Structured subspace embedding on attributed networks. *Information Sciences*, 512:726–740.