WERank: Rank Degradation Prevention for Self-Supervised Learning via Weight Regularization

Ali Saheb Pasand[1*], **Reza Moravej**[1*], **Mahdi Biparva**[1†], **Ali Ghodsi**[2]
Huawei Noah's Ark Lab[1], University of Waterloo[2]
∗ Done during internship † correspondence to mahdi.biparva@huawei.com

Abstract

A common phenomenon in self-supervised learning is dimensional collapse (also known as rank degeneration), where the learned embeddings are mapped to a low dimensional subspace of the embedding space. Despite employing mechanisms to prevent dimensional collapse, previous self-supervised approaches have not succeeded in completely alleviating the problem. We propose WERank, a new regularizer on the weight parameters of the neural network encoder to prevent rank degeneration. Our regularization term can be applied on top of any existing self-supervised method without significant computational cost. We provide empirical and mathematical evidence to demonstrate the effectiveness of WERank in avoiding dimensional collapse.

1 Introduction

The goal of Self-Supervised Learning (SSL) methods is to learn useful representations of data without relying on human annotations. Recent advances in SSL have shown that it is possible to learn self-supervised representations that are competitive with supervised labels in a variety of settings including visual and graph domains [1, 8, 5, 3, 14, 6, 4]. SSL methods enforce the model to learn similar representations for semantically similar inputs. However, simply enforcing similarity between similar points will result to the model trivially learning to output a single embedding vector for every input. This phenomena, namely complete collapse, is undesirable since it will provide no gradients for learning and the representations offer no information for the downstream task. Complete collapse is commonly avoided using contrastive losses, regularization techniques, or architectural tricks [4, 8, 5, 3, 14, 6]. Though complete collapse can easily be prevented, it is still common for SSL learned representations to map to a low dimensional subspace of the representation space. Avoiding this kind of dimensional/partial collapse has remained a challenging problem across different SSL approaches [2, 7]. Dimensional collapse is linked with strong correlations between axes, which results in relatively uninformative embeddings [10]. In simple terms, it is desirable to take full advantage of the embedding space to represent more information. Learning dimensionally collapsed, or rank-deficient representations has shown to be a bottleneck in SSL models achieving high performance on downstream tasks [2, 7].

In contrastive methods which employ positive and negative pairs in the loss function, it seems intuitive that the repulsive effect of negative samples prevents rank degradation and encourages full use of all dimensions. Contrary to this intuition, contrastive methods still suffer from dimensional collapse, particularly in the presence of strong augmentations or an over-parameterized encoder [11]. Distillation methods such as BYOL [8], SimSiam [5] employ architectural tricks inspired by knowledge distillation [9] to prevent collapse. While the dynamics of the alignment of eigenspaces between the predictor and its input correlation matrix plays a role in preventing collapse, distillation methods have no explicit mechanism to avert dimensional collapse and are thus prone to rank

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

degradation [13]. Finally, information maximization methods add an explicit regularization term to the loss function to ensure that redundancy is minimized in the embedding space. However, prominent information maximization methods still suffer from dimensional collapse in practice [7].

While the above SSL approaches aim to alleviate rank degradation in the final embedding space, they fail to address it in earlier layers of the neural network. For deep networks, enforcing the regularization term on the output of the final layer does not necessarily prevent rank degradation at earlier layers. The implicit regularization present in deep networks causes dimensional collapse across layers of the network [11] [1]. Thus, the low rank solution found in an early layer would propagate to deeper layers. We propose WERank, a new **W**eight r**E**gularization term which serves as a complementary **Rank** degradation prevention mechanism on top of any SSL method. WERank prevents dimensional collapse throughout the network rather than the final layer. Unlike previous information maximization approaches employing variance/covariance/cross-covariance terms, WERank is directly computed on the weights of the network and is computationally more efficient.

2 WERank Regularization

Given a set of N input data points D, denote $X \in \mathbb{R}^{N \times K}$ as the corresponding data feature matrix, where K is the feature dimension of each data point. The embeddings $Z \in \mathbb{R}^{N \times M}$, are obtained via a network f_θ, namely $Z = f_\theta(X)$. Given a set of augmentations T, two distorted views $t_i(x)$ and $t_j(x)$ are obtained for an input $x \in D$ by applying transformations $t_i, t_j \in T$. The model f_θ is pre-trained to learn useful node representations by enforcing the embeddings of distorted views $t_i(x)$ and $t_j(x)$ to be similar. In general, SSL approaches minimize a loss function of form:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim D, t_i, t_j \sim T} sim(f_\theta(t_i(x)), f_\theta(t_j(x)))$$

where sim is a similarity function such as cosine or euclidean similarity. The representations learned by the trained model f_θ can be used on the downstream task defined for the particular domain, such as image/node/graph classification or prediction. Collapse is often prevented by introducing a regularization term to the loss (such as the variance and covariance terms in VICRreg). Alternatively, contrastive loss functions prevent collapse by pushing dissimilar pairs apart and distillation methods leave the loss unmodified but prevent collapse using architectural tricks.

2.1 Rank Degradation Prevention by Weight Regularization

We aspire WERank from the feature decorrelation losses in information maximization methods. Such methods prevent collapse by enforcing decorrelation between the axes of the embedding vectors in Z via an explicit regularization term in the loss function. Covariance decorrelation makes all the components of embeddings in Z linearly independent from each other, encouraging different dimensions to represent different semantic content. The foundational issue, however, is that the regularization term is applied too late in the process. We aim to derive a computationally efficient method which prevent dimensional collapse at different layers of the network.

Consider a neural network with L trainable weight matrices $W_1, W_2, ..., W_L$. We can write the output of the l_{th} linear layer with input $X^{(l)}$ as:

$$X^{(l+1)} = \sigma(X^{(l)} W_l) = \sigma(H^{(l)} W_l)$$

Given an embedding dimension M, and N data points, we can defined the covariance matrix $C \in \mathbb{R}^{M \times M}$ as:

$$C = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{z}_i - \overline{\boldsymbol{z}})(\boldsymbol{z}_i - \overline{\boldsymbol{z}})^T$$

where $\overline{\boldsymbol{z}} = \sum_{i=1}^{N} \boldsymbol{z}_i / N$. Similar to information maximization methods, dimensional collapse can be alleviated by enforcing feature decorrelation in the embedding space. A straight forward way to achieve feature decorrelation is by introducing the Forbenious norm between the covariance matrix of the embedding vector and the identity as a regularizer to the loss function:

$$||C - I_{(M \times M)}||_F$$

[1] refer to section 3 and supplementary materials (section 4.2) for supporting results

However, our objective is to apply the regularizer to earlier layers of the network rather than to the final output. Thus, we aim to compute the regularization term on the weights of the network $W_1, W_2, ..., W_L$ instead of the final output while achieving the same impact as the above regularizer at every layer.

Proposition: Denote λ_i as the i_{th} eigenvalue of the covariance matrix C, and $W \in \mathbb{R}^{d_{in} \times d_{out}}$ a single layer model, where without loss of generality the input dimension is larger than the output dimension. Then, $||C - I_{(d_{out} \times d_{out})}||_F$ is minimized if $W^T W$ is as close as possible to the diagonal matrix with its i_{th} diagonal element $d_i = \frac{1}{\lambda_i^2(H)}$.

Proof: We can write the Forbenious norm between the covariance matrix of output and identity as a function of the eigenvalues of the covariance matrix:

$$||C - I_{(d_{out} \times d_{out})}||_F = \sqrt{tr\bigg((C-I)(C-I)^T\bigg)} = \sqrt{tr(CC^T) - 2tr(C) + I}$$

$$= \sqrt{\sum_{i=1}^{d_{out}} (\lambda_i^2(C) - 2\lambda_i(C) + 1)} = \sqrt{\sum_{i=1}^{d_{out}} (\lambda_i(C) - 1)^2}$$

The i_{th} eigenvalue of the covariance matrix can be bounded as follows:

$$\lambda_i(C) = \lambda_i(W^T H^T H W) \leq \lambda_i(W^T H^T H)\lambda_{d_{out}}(W) \leq \lambda_i^2(H)\lambda_{d_{out}}^2(W)$$

Where $\lambda_{d_{out}}(W)$ and $\lambda_1(W)$ are the largest and smallest non-zero eigenvalues of W respectively. Thus, from the above we get:

$$||C - I_{(d_{out} \times d_{out})}||_F = \sqrt{\sum_{i=1}^{d_{out}} (\lambda_i(C) - 1)^2} \leq \sqrt{\sum_{i=1}^{d_{out}} (\lambda_i^2(H)\lambda_{d_{out}}^2(W) - 1)^2}$$

Since the input H to the network is not controllable, the only way for controlling the eigenvalues of covariance is regularizing the weights during training. A perfect whitening is only possible if all eigenvalues values of the matrix H are the same and we regularize $W^T W$ to be as close as possible to a diagonal matrix with diagonal elements $\frac{1}{\lambda_i^2(H)}$. However, even in the case of having the same eigenvalues, calculating this regularization term requires performing Eigenvalue Decomposition for each batch and each layer during all epochs which makes it computationally intractable. Additionally, if we perform the training in batch format, the optimization will likely be unstable since the eigenvalues will be different for each batch. We resort to making the identity as the target matrix for $W^T W$. We will show that this regularization term makes all $\lambda_i^2 W$'s as close as possible to 1, and as a result, will prevent the degradation of the rank of input matrix. We thus propose the following regularization term applied to every layer of the network:

$$\mathcal{L}_{reg} = \sum_{l=1}^{L} \alpha_l \Big|\Big| W_l^T W_l - I \Big|\Big|_F$$

where α_i controls the intensity of regularization for different layers. If at any layer l, the output dimension d_{out} is larger than d_{in}, we can replace the regularization term with $\Big|\Big| W_l W_l^T - I, \Big|\Big|_F$. Notice the regularization term whitens the matrix $W_l^T W_l$ or $W_l W_l^T$ with the largest rank $max(d_{in}, d_{out})$.

WERank can be serve as a complimentary rank degeneration prevention mechanism on top of any SSL method by simply being added to the original SSL loss term:

$$\mathcal{L} = \mathcal{L}_{\mathcal{SSL}} + \mathcal{L}_{reg}$$

3 WERank is Effective in Preventing Dimensional Collapse

To empirically support the effectiveness of WERank, we test the regularizer on SSL models under the two identified causes of collapse [11]; namely (i) implicit regularization caused by over-parameterization and (ii) strong augmentation (relevant results can be found in Appendix 4.1).

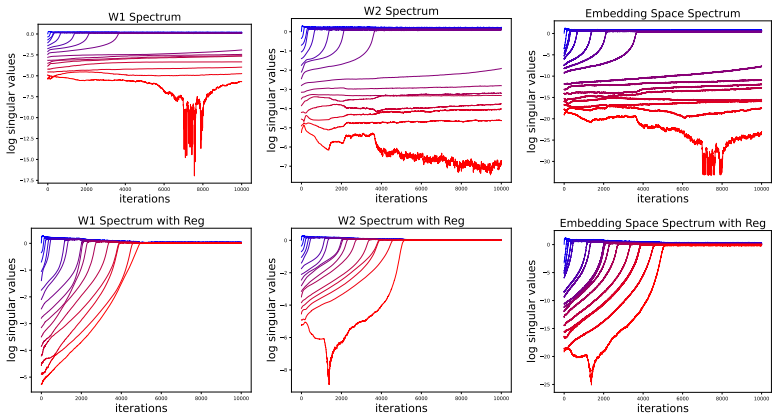

Figure 1: The singular values of the weight matrices and the embedding space covariance matrix during training (top) VICReg with no regularization (button) VICReg with the WERank regularizer. The augmentation magnitude (k) is set to 0.1

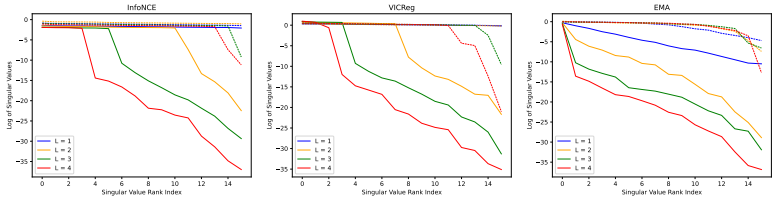

Figure 2: Embedding space singular values after training the models for 5000 epochs. L denotes the number of layers, dashed lines depict the model with WERank regularization. Training each model + WERank for 10000 epochs would result in all singular values converging to 1, while models without WERank face rank degradation.

Though previous work [11] identified the above causes of collapse for contrastive methods, we show that similar conditions can cause collapse in other SSL methods. [2] Following the same experimentsal setup, we sample 1000 points from a 16 dimensional isotropic Gaussian with covariance matrix $\sum_{i,j}(x_i - x_j)(x_i, x_j)^T/N = I$. Two views of the sample x_i are then generated from an additive Gaussian with covariance matrix $\sum_i (x_i^{'} - x_i)(x_i^{'} - x_i)^T/N = \textit{block-diagonal}(0, k * I)$, where the block is 8×8. We consider two models with InfoNCE [4] and VICReg [3] loss functions, as well as one model trained under the student-teacher Exponential Moving Average (EMA) algorithm (similar to BYOL [8]). For each model, we train an identical model with WERank added to the loss with a coefficient of 0.1. The models are trained for 1000 epochs in the full batch regime. We apply basic stochastic gradient descent without momentum or weight decay.

Due to the implicit regularization caused by over-parameterization, the smallest group of singular values [3] grow significantly slower throughout training [11]. We study the impact of WERank on the simplest over-parameterized setting by having a two-layer linear MLP with no bias. We denote the weight matrices of the network as $W_1, W_2 \in \mathbb{R}^{16\times16}$. Figure 1 depicts the evolution of the singular values of the weight and embedding space covariance matrices of VICReg with and without WERank. The whitening loss applied by VICReg on the output of the final layer does not prevent dimensional collapse at earlier layers. However, WERank helps with preventing collapse resulted from the implicit regularization present in deep models by explicitly pushing the singular values up at every layer. In Figure 2, we plot the singular values of the embedding space of different models with and without WERank regularization applied. Evidently, WERank is effective in preventing rank degradation in deeper networks, where the impact caused by implicit regularization aggravates.

[2] It is possible to verify that the theoretical justification provided by Jing et al. [11] can be extended to other SSL methods.

[3] We note that pushing the eigenvalues to 1 has the same impact as pushing the singular values to 1. We visualize singular values for consistency with previous work by Jing et al.[11]

References

[1] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, NeurIPS'19, 2019.

[2] R. Balestriero and Y. LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In *Advances in Neural Information Processing Systems*, NeurIPS'22, 2022.

[3] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2022.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. PMLR, 2020.

[5] X. Chen and K. He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 2021.

[6] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[7] Q. Garrido, R. Balestriero, L. Najman, and Y. Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023.

[8] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, 2020.

[9] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

[10] T. Hua, W. Wang, Z. Xue, Y. Wang, S. Ren, and H. Zhao. On feature decorrelation in self-supervised learning. *CoRR*, 2021.

[11] L. Jing, P. Vincent, Y. LeCun, and Y. Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

[13] Y. Tian, X. Chen, and S. Ganguli. Understanding self-supervised learning dynamics without contrastive pairs, 2021.

[14] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. *CoRR*, 2021.

4 Appendix

4.1 WERank and Augmentation Strength

In networks with limited capacity, strong augmentation along feature dimensions is a secondary cause for dimensional collapse [11]. To test the impact of the magnitude of augmentation k, we choose a simple linear network with weights $W_1 \in \mathbb{R}^{16\times16}$. In figure 3, we find that our method is most impactful when the augmentation is weak. However, the regularization term becomes less effective as the magnitude of augmentation increases. From an analytical lens, collapse in this case happens due to the dynamics of the time derivative of the weight matrix W being determined by the augmentation distribution covariance matrix, refer to section 4.2 in [11]. This is expected, since extreme augmentation limits the amount of common information between the distorted views which can be used by the model for learning. Thus, the model will inevitably collapse in the presence of strong augmentation. We note that the results for VICReg and VICReg + WERank aren't much different because WERank is simply enforcing the same variance/covariance terms as VICReg if only applied to a single layer.

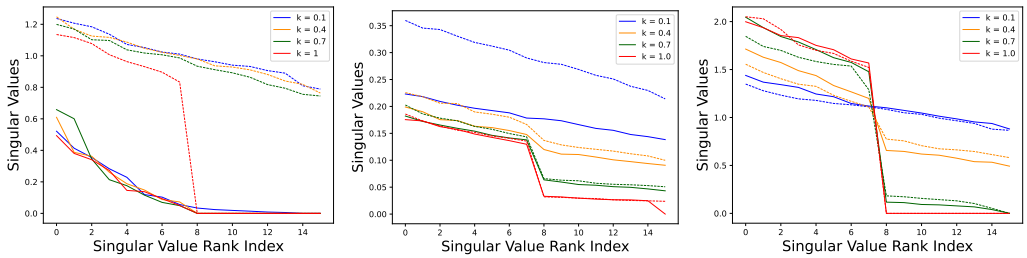

Figure 3: Weight matrix singular value spectrum with different augmentation amplitudes k, measured at the end of training. Solid lines depict the model with no regularizer and dotted lines depict model + WERank. (Left) EMA model (middle) InfoNCE model (right) VICReg model.

4.2 Additional Results on Implicit Regularization

In addition to the VICReg model, we provide plots depicting the evolution of singular values in InfoNCE and EMA models. All models are trained for 10000 epochs in the full batch regime.

We apply basic stochastic gradient descent without momentum or weight decay when training the VICReg and InfoNCE models. However, we find that applying the same optimizer on the EMA model results in the singular values remaining constant throughout training. Thus, we apply the AdamW [12] optimizer with a learning rate of 0.01 and weight decay 0.0003. [4]

The WERank coefficient is set to 0.1 for all models. The variance, invariance, covariance coefficients for VICReg are set to 10, 10 and 1 respectively. The EMA model is implemented in a similar fashion to BYOL [8], where the teacher network is updated with an exponential moving average with a factor of 0.995.

The first eight singular values converge to one in VICReg and InfoNCE models. However, the EMA model has trouble pushing the first eight singular values higher. We suspect this is due to the lack of a mechanism to encourage high rank representations in the EMA model.

[4] In general, the traning dynamics of the EMA model is highly sensitive to the choice of the optimizer. However, changing the hyperparameters of the AdamW optimizer does not result in the singular values converging to 1.

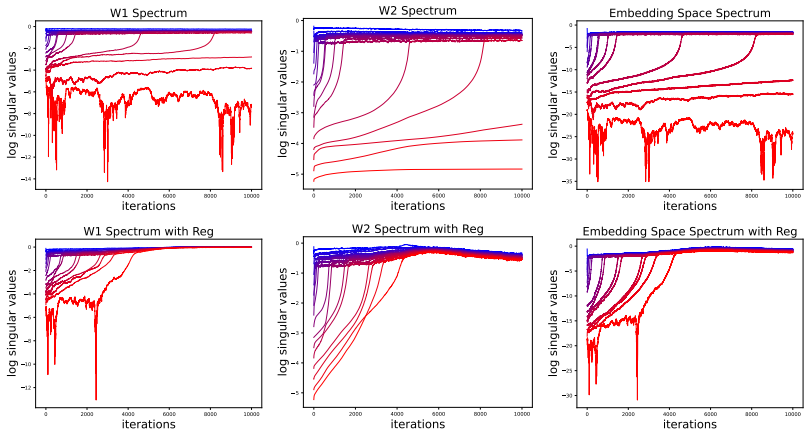

Figure 4: The singular values of the weight matrices and the embedding space covariance matrix during training (top) InfoNCE model with no regularization (button) InfoNCE model with the WERank regularizer. The augmentation magnitude (k) is set to 0.1

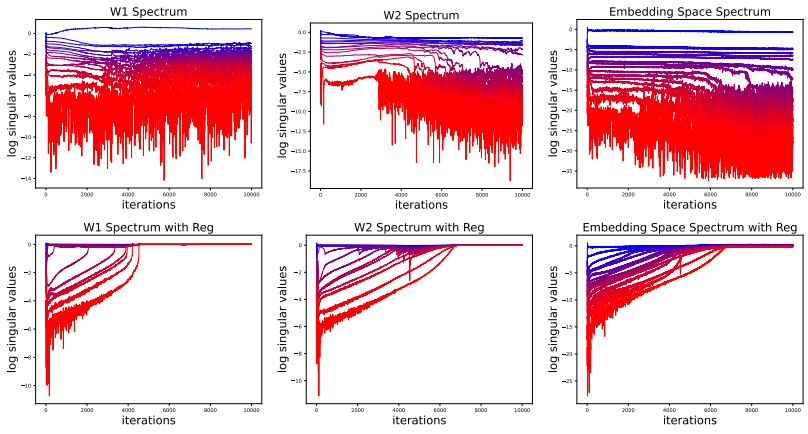

Figure 5: The singular values of the weight matrices and the embedding space covariance matrix during training (top) EMA model with no regularization (button) EMA model with the WERank regularizer. The augmentation magnitude (k) is set to 0.1