Engineering the Neural Collapse Geometry of Supervised-Contrastive Loss (Student Abstract)

Jaidev Gill, Vala Vakilian, Christos Thrampoulidis

University of British Columbia
jaidevg5@student.ubc.ca, vaalaa@student.ubc.ca, cthrampo@ece.ubc.ca

Abstract

Supervised-contrastive loss (SCL) is an alternative to cross-entropy (CE) for classification tasks that makes use of similarities in the embedding space to allow for richer representations. Previous works have used trainable prototypes to help improve test accuracy of SCL when training under imbalance. In this work, we propose the use of fixed prototypes to help engineering the feature geometry when training with SCL. We gain further insights by considering a limiting scenario where the number of prototypes far outnumber the original batch size. Through this, we establish a connection to CE loss with a fixed classifier and normalized embeddings. We validate our findings by conducting a series of experiments with deep neural networks on benchmark vision datasets.

1 Introduction

Neural Collapse (NC), formalized by Papyan, Han, and Donoho (2020), is a phenomenon where training a deep-net model beyond zero training error on a balanced dataset using cross-entropy (CE) results in the feature embeddings to collapse to their corresponding class mean and they to converge to form a symmetric ETF geometry. In other words, the class-mean embeddings form an implicit geometry described by vectors of equal norms and angles that are maximally separated. In addition to a number of works further analysing the NC implicit feature geometry when training with CE (Mixon, Parshall, and Pi 2020; Thrampoulidis et al. 2022), a more recent line of studies has focused on NC in the context of supervised-contrastive loss (SCL) (Graf et al. 2021; Zhu et al. 2022; Kini et al. 2023).

Drawing inspiration from unsupervised contrastive learning (Chen et al. 2020), SCL was proposed by Khosla et al. (2021) as a substitute to CE for classification. Specifically, SCL makes use of semantic information by directly contrasting learned features. Graf et al. (2021) was the first to theoretically analyze the feature geometry of SCL, demonstrating that it forms an ETF when data is balanced. However, when the label distribution is imbalanced, the geometry changes is no longer symmetric, potentially hurting test accuracy. To combat this, Zhu et al. (2022) proposed a training framework called balanced contrastive learning (BCL) which improves the SCL generalization test accuracy under imbalances. Their framework uses a class averaging modification to SCL alongside a set of k *trainable* prototypes, representing class centers. In another related work, Cui et al. (2021) introduced PaCo, a supervised contrastive method that also takes advantage of such trainable class centers.

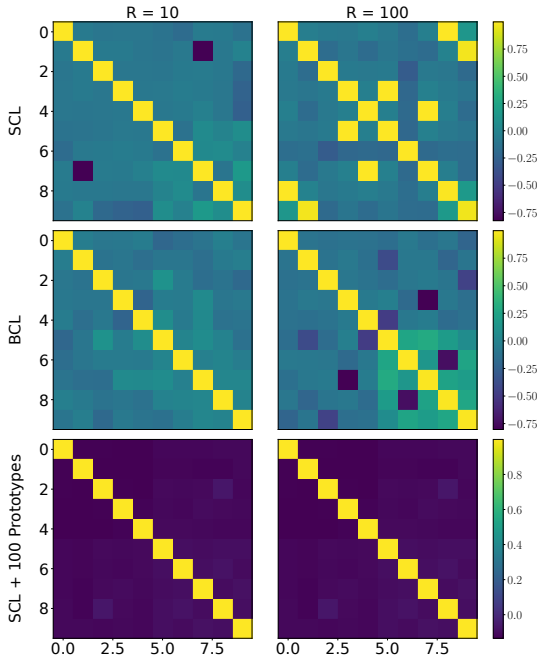

Figure 1: Comparison of Gram matrices $\mathbf{G_M}$ at last epoch (350) trained on STEP imbalanced CIFAR-10 and ResNet-18 with (Top) vanilla SCL ($n_w = 0$) (Middle) Class averaging (BCL) (Zhu et al. 2022) satisfying class representation requirements through batch binding (Kini et al. 2023) (Bottom) SCL with ($n_w = 100$) prototypes.

These works collectively suggest that prototypes can play a crucial role in determining the implicit geometry when training with SCL, as Zhu et al. (2022) claims their frameworks helps achieve an ETF geometry. However, in both cases, prototypes are trainable parameters, optimized alongside various other heuristics and modifications. Thus, it is challenging to ascertain their specific impact on the training process. This raises the question *what is the direct impact*

of prototypes on the SCL geometry when isolated from other modifications? In order to answer this question, this paper investigates the implicit geometry of SCL with *fixed* prototypes. We use experimental results to illustrate how prototypes can help achieve a symmetric feature geometry. In addition, through theoretical analysis, we establish a connection to CE with fixed classifiers when the number of prototypes far outnumber the batch size.

2 Tuning Geometry with Prototypes

Setup. We consider a k-class classification task with training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i\in[N]}$ where $\mathbf{x}_i \in \mathbb{R}^p$ are the N training points with labels $y_i \in [k]$.[1] The SCL loss is optimized over batches $B \subset [N]$, $|B| = n$, belonging to a batch-set $\mathcal{B}$. Concretely, $\mathcal{L} := \sum_{B\in\mathcal{B}} \mathcal{L}_B$, where the loss for each batch B is given below as

$$\mathcal{L}_B := \sum_{i\in B} \frac{1}{n_{B,y_i} - 1} \sum_{\substack{j\in B \\ j\neq i \\ y_j = y_i}} \log\Big(\sum_{\substack{\ell\in B \\ \ell\neq i}} \exp\left(\mathbf{h}_i^\top \mathbf{h}_\ell - \mathbf{h}_i^\top \mathbf{h}_j\right)\Big). \tag{1}$$

Here, $\mathbf{h}_i := \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{x}_i) \in \mathbb{R}^d$ is the last-layer learned feature-embedding for a network parameterized by $\boldsymbol{\theta}$. n_{B,y_i} indicates the number of samples with label y_i in B. As per standard practice (Chen et al. 2020; Khosla et al. 2021), we assume a normalization layer on the output of the network, hence $\|\mathbf{h}_i\| = 1\ \forall i \in [N]$. It is also common to include a scaling of the inner products by a temperature parameter τ (Khosla et al. 2021); since this can be absorbed in the normalization, we drop it above for simplicity.

Methodology. Inspired by the class-complement method of Zhu et al. (2022), the learnable class centers of Cui et al. (2021), and the batch-binding algorithm of Kini et al. (2023), we propose using *fixed* prototypes. These prototypes collectively form a desired reference geometry for the embeddings to learn.

Definition 1 (Prototype)**.** A *prototype* $\mathbf{w}_c \in \mathbb{R}^d$ for class $c \in [k]$ is a fixed vector that represents the desired representation of embeddings $\{\mathbf{h}_i\}_{y_i=c}$ in class c.

Our method optimizes SCL with a new batch $\{\mathbf{h}_i\}_{i\in B} \cup \mathcal{W}$, where $\mathcal{W} := \bigcup_{i=1}^{n_w} \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ and n_w is the number of added prototypes per class. We highlight two key aspects of this strategy. (i) First, as n_w increases, there is *no increase* in the computational complexity of the loss computation, as the number of inner products evaluated increases from ${}^{n^2}/_2$ in vanilla SCL (Eq. (1)) to ${}^{n^2}/_2 + nk$ when prototypes are introduced. This increase is solely due to the presence of k distinct prototypes and remains constant regardless of n_w. (ii) Second, we guarantee that prototypes are fixed and form a suitable, engineered geometry, defined formally in Definition 2 below. In particular, this is in contrast to Cui et al. (2021); Zhu et al. (2022) where prototypes are learned. See Fig. 1 for a comparison of geometries learned using our method.

[1] We denote $[N] := \{1, 2, \ldots, N\}$.

Definition 2 (Prototype Geometry)**.** Given a set of prototypes $\{\mathbf{w}_c\}_{c\in[k]}$ the prototype geometry is characterized by a symmetric matrix $\mathbf{G}_* = \mathbf{W}^\top\mathbf{W}$ where $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_k]$.

Proposition 1. *Let $\hat{n} := k \cdot n_w$ be the total number of prototypes added to the batch, and n be the original batch size. Then in the limit $\hat{n} \gg n$ the batch-wise SCL loss becomes,*

$$\mathcal{L}_B \to -\sum_{i\in B}\left[\log\left(\frac{\exp(\mathbf{w}_{y_i}^\top \mathbf{h}_i)}{\sum_{c\in[k]}\exp(\mathbf{w}_c^\top \mathbf{h}_i)}\right) + \mathbf{w}_{y_i}^\top \mathbf{h}_i\right].$$

When considering the limiting setting of SCL with prototypes, we arrive at Prop. 1. Proof, and further exploration can be found in Gill, Vakilian, and Thrampoulidis (2023).

Remark 1. *This setting is remarkably similar to Yang et al. (2022) that trains CE loss with fixed classifiers forming an ETF geometry. However two key differences emerge: (i) the features and prototypes are normalized, i.e. $\|\mathbf{h}_i\| = 1\ \forall i \in B$, $\|\mathbf{w}_c\| = 1\ \forall c \in [k]$, and (ii) here, there is an additional alignment-promoting regularization induced by the inner product between $\mathbf{h}_i$ and $\mathbf{w}_{y_i}$.*

3 Concluding Remarks

In this work, we have isolated and explored the effects of prototypes on supervised-contrastive loss. In doing so, we have identified a reliable method in tuning the learnt embedding geometry. In addition, a theoretical link to cross-entropy was established. Overall, our discoveries indicate that employing fixed prototypes offers a promising avenue for streamlining framework modifications that typically treat prototypes as trainable parameters without a clear understanding of their direct contribution. Moreover, this opens up an exciting avenue for future research to explore how choosing prototype geometries favoring larger angles for minority classes can positively impact generalization performance.

Acknowledgements

JG and CT gratefully acknowledge the support of an NSERC USRA. The authors also acknowledge use of the Sockeye cluster by UBC Advanced Research Computing. This work is supported by an an NSERC Discovery Grant, NSF Grant CCF-2009030, and by a CRG8-KAUST award.

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709.

Cui, J.; Zhong, Z.; Liu, S.; Yu, B.; and Jia, J. 2021. Parametric Contrastive Learning. arXiv:2107.12028.

Gill, J.; Vakilian, V.; and Thrampoulidis, C. 2023. Engineering the Neural Collapse Geometry of Supervised-Contrastive Loss. arXiv:2310.00893.

Graf, F.; Hofer, C.; Niethammer, M.; and Kwitt, R. 2021. Dissecting Supervised Contrastive Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 3821–3830. PMLR.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2021. Supervised Contrastive Learning. arXiv:2004.11362.

Kini, G. R.; Vakilian, V.; Behnia, T.; Gill, J.; and Thrampoulidis, C. 2023. Supervised-Contrastive Loss Learns Orthogonal Frames and Batching Matters. arXiv:2306.07960.

Mixon, D. G.; Parshall, H.; and Pi, J. 2020. Neural collapse with unconstrained features. *CoRR*, abs/2011.11619.

Papyan, V.; Han, X. Y.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Thrampoulidis, C.; Kini, G. R.; Vakilian, V.; and Behnia, T. 2022. Imbalance Trouble: Revisiting Neural-Collapse Geometry. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 27225–27238. Curran Associates, Inc.

Yang, Y.; Chen, S.; Li, X.; Xie, L.; Lin, Z.; and Tao, D. 2022. Inducing Neural Collapse in Imbalanced Learning: Do We Really Need a Learnable Classifier at the End of Deep Neural Network? arXiv:2203.09081.

Zhu, J.; Wang, Z.; Chen, J.; Chen, Y.-P. P.; and Jiang, Y.-G. 2022. Balanced Contrastive Learning for Long-Tailed Visual Recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6898–6907.