ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Hierarchical priors for Hyperspherical Prototypical Networks

Samuele Fonio[1], Lorenzo Paletto[1], Mattia Cerrato[3], Dino Ienco[2], Roberto Esposito[1]

1- University of Turin - Computer Science Department
Corso Svizzera, 185, Turin - Italy

2- INRAE, UMR TETIS
Rue Jean François Breton, 500, Montpellier - France

3- Johannes Gutenberg-Universität Mainz
Saarstrasse 21, Mainz, Germany

Abstract. In this paper, we explore the usage of hierarchical priors to improve learning in contexts where the number of available examples is extremely low. Specifically, we consider a Prototype Learning setting where deep neural networks are used to embed data in hyperspherical geometries. In this scenario, we propose an innovative way to learn the prototypes by combining class separation and hierarchical information. In addition, we introduce a contrastive loss function capable of balancing the exploitation of prototypes through a prototype pruning mechanism. We compare the proposed method with state-of-the-art approaches on two public datasets.

1 Introduction

Prototype Learning (PL) [1] has shown interesting results in contexts where data is scarce [2, 3]. In this context, it is common to use a cosine measure to estimate the similarity between examples and prototypes. Also, by projecting the embeddings onto the hypersphere, one enables the usage of fixed *hyperspherical* prototypes [4]. Operating in this way, the embedding space can be evenly partitioned and each partition can be assigned to a single class prototypes. As a consequence, the distance between class prototypes is maximized and it is not needed to learn the prototypes during training.

On the other hand, the use of hierarchical knowledge as a prior for learning the prototypes has been widely studied in the PL literature [5, 4, 6, 7], and it has been shown to be rewarding in terms of accuracy. Indeed, breaking the uniformity using a class taxonomy may improve the performances by allowing to place the prototypes according to the natural distribution of the examples.

While we experimentally observed that the use of hierarchical information is beneficial, we also noticed that prototypes of nearby concepts (in the hierarchy) risk being positioned too close to each other, making it difficult to correctly distinguish between the corresponding classes.

This work has been partially supported by the Spoke 1 "FutureHPC & BigData" of ICSC - Centro Nazionale di Ricerca in High-Performance-Computing, Big Data and Quantum Computing, funded by European Union - NextGenerationEU.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Fig. 1: Cost matrices for CIFAR100 and CUB2011 classes. Each entrance represents the length of the shortest path in the tree to reach one class starting from the other. Yellow is the maximum distance (e.g. 10 in a 5 deep hierarchy) and purple is the lower (always 0 for elements in the diagonal). CIFAR100's hierarchy has five levels, CUB2011 three.

Our contributions are as follows: in Section 2 we introduce a method to weaken the hierarchical information as a way to overcome the aforementioned problem; in Section 3, we introduce a contrastive loss along with a prototype pruning mechanism to improve learning the embeddings; in Section 4 we consider a baseline approach and two state-of-the-art methods and compare them with our proposed approach on two publicly available datasets.

2 Prototypes Learning

Prototypes are the anchor points that determine the prediction. While more general definitions are available, here we consider the setting where classes are determined by proximity to a single fixed prototype. Specifically, closeness is evaluated using cosine similarity in an embedding space learned by a neural network.

In [4], the authors introduce hyperspherical prototypes, i.e., they propose a method to learn equally spaced prototypes $\pi_i, i = 1, \ldots, K$ on the n-dimensional hypersphere[1] by minimizing the *uniform loss* $\mathcal{L}_U$:

$$\mathcal{L}_U(P) = \frac{1}{K}\sum_{i=1}^{K} \max_{j \in \mathcal{C}} M_{ij}, \quad M = PP^t - 2I, \quad \text{s.t. } \forall i \, \|P_i\| = 1, \tag{1}$$

where $\mathcal{C}$ is the set of classes, $P \in \mathbb{R}^{K \times n}$ denotes the current set of hyperspherical prototypes, I denotes the identity matrix and M denotes a matrix containing the pairwise cosine similarities between prototypes.

When a hierarchical structure is available over the classes, learning the points uniformly on the hypersphere is not usually the best choice. Let us then assume to have a tree-shaped hierarchy over the classes, and let us represent this information by means of a cost matrix H, such that H_{ij} is the distance on the hierarchy between class i and class j (see Fig. 1). In [5], this information is

[1] It is worth noting that positioning points uniformly on an n-dimensional hypersphere cannot be done exactly in general.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

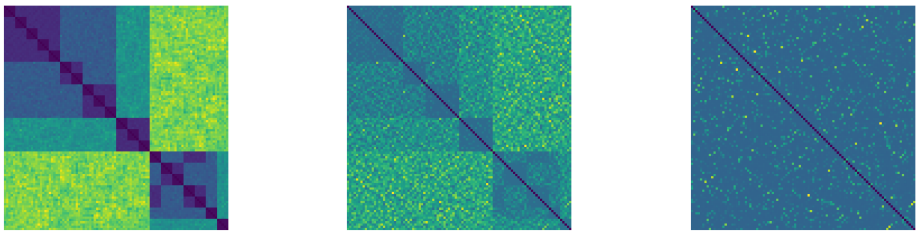

Fig. 2: Cosine distance matrices computed among CIFAR100 prototypes positioned with the methods presented (darker colors are used for more similar prototypes). On the left prototypes learnt minimizing $\mathcal{L}_D$. On the right, prototypes learnt minimizing $\mathcal{L}_U$. In the center prototypes learnt by the combination of the two (our approach).

exploited by learning the prototype so to minimize the *distortion loss* $\mathcal{L}_D$:

$$\mathcal{L}_D(\boldsymbol{\pi}) = \frac{1}{K(K-1)} \min_{z \in \mathbb{R}^+} \sum_{k,l \in \mathcal{C}} \left(\frac{z \cdot d(\pi_k, \pi_l) - H_{k,l}}{H_{k,l}} \right)^2, \tag{2}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ and π_k (π_j) is the embedding of the k-th (j-th) prototype.

We claim that oftentimes, the hierarchical information may impact negatively on the performances because it positions some of the prototypes too close to each other. Here we propose to merge the previous methods in a two steps solution. We start by learning the prototypes minimizing $\mathcal{L}_D$ until convergence, then we update them by minimizing $\mathcal{L}_U$. The result of the two-step procedure is a set of prototypes that takes advantage of the hierarchical prior in a weakened fashion since it also spreads them out on the hypersphere.

Fig. 2 shows the effect of our strategy. The three panels report matrices showing with colors the cosine distances between π_i and π_j, for all $i, j \in \{1 \ldots K\}$. The left panel refers to the prototypes learned by minimizing $\mathcal{L}_D$ (*full hierarchy* in the following); the right panel reports about the prototypes learned by minimizing $\mathcal{L}_U$ (*no hierarchy*); the central panel shows the results of our approach (*weak hierarchy*). As it should be apparent, minimization of the distortion loss tends to position prototypes close to each other, with the risk of making it difficult to distinguish between similar classes. Minimizing the uniform loss spreads prototypes out, completely disregarding the hierarchical information. Our approach finds a compromise between these two extremes.

3 Learning Embeddings via Contrastive Learning

Mettes et al. [4] proposes a non-contrastive learning approach learning the embeddings in a prototype learning setting. Their approach is based on the minimization of the similarity between the examples of a given class and the corresponding prototype embedding.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Method		**Number of examples per class**				
Loss	Hierarchy	10	15	20	25	30
$\text{cHPS}_{\frac{1}{K}}$	full	$\underline{16.84 \pm 0.55}$	$\underline{\mathbf{20.96 \pm 0.60}}$	$\underline{\mathbf{25.00 \pm 0.47}}$	$\underline{\mathbf{28.47 \pm 0.99}}$	$\underline{\mathbf{33.12 \pm 0.51}}$
	weak	16.46 ± 0.75	19.95+0.60	24.59 ± 0.41	27.27 ± 1.51	31.89 ± 0.77
	no	15.88 ± 0.64	18.21 ± 0.63	22.64 ± 0.76	25.56 ± 0.34	29.58 ± 0.63
cHPS_0	full	14.50 ± 0.81	18.71 ± 0.54	23.33 ± 0.48	25.70 ± 1.47	30.06 ± 0.91
	weak	$\underline{16.49 \pm 0.57}$	$\underline{19.29 \pm 0.64}$	$\underline{24.26 \pm 0.39}$	$\underline{26.54 \pm 0.68}$	$\underline{31.54 \pm 0.84}$
	no	15.67 ± 0.35	18.15 ± 0.52	22.13 ± 0.44	25.12 ± 0.84	28.89 ± 0.80
HPS	full	11.02 ± 0.54	13.57 ± 0.50	16.04 ± 0.38	18.49 ± 1.01	21.25 ± 0.64
	weak	$\underline{15.39 \pm 0.74}$	$\underline{18.64 \pm 0.21}$	$\underline{21.67 \pm 0.54}$	$\underline{24.93 \pm 0.81}$	$\underline{27.44 \pm 0.76}$
	no	14.45 ± 0.80	17.78 ± 0.50	20.23 ± 0.59	22.96 ± 0.76	24.75 ± 0.67
HBL	full	16.89 ± 0.63	$\underline{20.44 \pm 0.63}$	$\underline{23.52 \pm 0.20}$	$\underline{26.25 \pm 0.41}$	$\underline{28.35 \pm 0.75}$
	weak	$\underline{\mathbf{17.13 \pm 0.54}}$	20.08 ± 0.45	23.48 ± 1.03	25.41 ± 0.53	27.36 ± 0.25
	no	16.07 ± 0.28	19.78 ± 0.51	21.95 ± 0.55	23.93 ± 0.43	25.83 ± 0.44
CE	-	13.95 ± 0.55	18.02 ± 0.36	21.23 ± 0.46	23.46 ± 0.73	27.24 ± 0.47

Table 1: Percentage of test accuracy for CIFAR100. For each dataset, we underline the best result for each method, and show in bold the best overall method.

To improve this method, and following the insights in [8], we propose the adoption of a contrastive loss function. In our proposal, the contrastive loss allows pruning the contribution of the prototypes based on their likelihood of representing the correct class for the current example. Given a training set $\{x_i; y_i\}_{i=1}^N$, where $y_i \in \mathcal{C}$ denotes the class label, let $f_\theta(x_i)$ denote the n-dimensional output of a network $f_\theta(\cdot)$. The proposed contrastive loss function is defined as:

$$\mathcal{L}_C(\theta;\gamma) = \frac{1}{N}\sum_{i=0}^{N} -\log \frac{\exp(s^i_{k,k})}{\sum_{j\in\mathcal{C}\setminus\{k\}} \exp(s^i_{k,j})\mathbb{1}_{\{p^i_{k,j}>\gamma\}} + \exp(s^i_{k,k})}, \qquad (3)$$

where $s^i_{k,j} = 1 + \cos(f_\theta(x^i_k), \pi_j)$ is a modified version of the cosine similarity between the embedding $f_\theta(x^i_k)$ and the prototype π_j, x^i_k is the i-th observation with true class k and $p^i_{k,j} = \frac{exp(s^i_{k,j})}{\sum_{l\in\mathcal{C}} exp(s^i_{k,l})}$ is its probability of belonging to class j. $\mathcal{L}_C(\theta;\gamma)$ depends on a threshold γ allowing the user to control the pruning of prototypes that are less likely to provide useful contributions.

In the following, we will refer with cHPS_γ to learning hyperspherical embedding via minimization of contrastive loss $\mathcal{L}_C$ with parameter γ. We note that cHPS_0 minimizes $\mathcal{L}_C(\theta;0)$, which reverts to a standard contrastive loss that does not prune any prototype.

4 Experiments and Results

We evaluated the effectiveness of our proposed approaches on two datasets: CIFAR100 [9] and CUB2011 [10].

The two datasets' hierarchies were taken respectively from [5] and [11]. To simulate the few data scenario we downsampled the original datasets (by extracting at random a subset with m examples per class), trained a model and evalu-

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Method		**Number of examples per class**				
Loss	Hierarchy	10	15	20	25	30
$\text{cHPS}_{\frac{1}{K}}$	full	$\mathbf{12.21 \pm 0.56}$	$\mathbf{15.16 \pm 0.85}$	$\mathbf{17.10 \pm 0.54}$	$\mathbf{18.86 \pm 0.37}$	$\mathbf{19.47 \pm 0.63}$
	weak	11.82 ± 0.21	14.93 ± 0.42	16.91 ± 0.50	18.09 ± 0.58	18.95 ± 0.61
	no	9.75 ± 0.35	12.37 ± 0.41	14.47 ± 0.59	15.60 ± 0.39	16.71 ± 0.55
cHPS_0	full	9.39 ± 0.44	12.31 ± 0.45	14.44 ± 0.58	15.91 ± 0.40	16.56 ± 0.53
	weak	11.27 ± 0.44	14.04 ± 0.44	16.39 ± 0.80	17.96 ± 0.37	18.83 ± 0.53
	no	10.62 ± 0.49	13.74 ± 0.54	15.76 ± 0.32	17.45 ± 0.29	18.47 ± 0.26
HPS	full	8.09 ± 0.70	10.49 ± 0.58	12.59 ± 0.66	13.72 ± 0.44	14.45 ± 0.31
	weak	8.93 ± 0.17	11.80 ± 0.57	13.68 ± 0.51	15.35 ± 0.21	16.56 ± 0.56
	no	5.81 ± 0.47	9.49 ± 0.46	12.08 ± 0.83	13.81 ± 0.62	15.04 ± 0.46
HBL	full	12.01 ± 0.38	14.57 ± 0.60	16.04 ± 0.46	17.19 ± 0.65	18.26 ± 0.20
	weak	12.06 ± 0.39	14.47 ± 0.79	15.94 ± 0.55	16.52 ± 0.71	17.25 ± 0.50
	no	11.01 ± 0.35	12.86 ± 0.63	14.31 ± 0.35	15.12 ± 0.33	16.06 ± 0.52
CE	-	9.84 ± 0.77	13.68 ± 0.51	16.12 ± 0.48	18.14 ± 0.30	19.11 ± 0.66

Table 2: Percentage of test accuracy for CUB2011. For each dataset, we underline the best result for each method, and show in bold the best overall method.

ated it on a separated and fixed test set. We repeated for 5 times and present the average over the 5 runs. We experimented varying m in $\{10, 15, 20, 25, 30\}$.

For our methods we trained a ResNet-18 network [12] from scratch, used embedding dimension $n = 64$, and normalized the outputs to operate on the hypersphere S^{63}. All other hyperparameters have been set following [5] with the exception of using a batch size of 32 instead of 128. We tested our method with $\gamma = 0$ (cHPS_0) and $\gamma = 1/K$ ($\text{cHPS}_{\frac{1}{K}}$, i.e., a prototype is included in the loss only if its likelihood of corresponding to the correct class was better than random chance).

For what concerns the prototypes, we experimented with three settings: *full hierarchy* prototypes are obtained by minimizing the distortion loss $\mathcal{L}_D$ using the same hyperparameters used in [5]; *no hierarchy* prototypes were trained minimizing the uniform loss $\mathcal{L}_U$ using the same hyperparameters used in [4]; *weak hierarchical* prototypes (our proposal) were trained starting from the *full hierarchy* prototypes and then minimizing the uniform loss. As regards the uniform part, the learning rate has been set to 5e-3 for CIFAR100 and 5e-2 for CUB2011. All hyperparameters (with the exception of those set as in the referred works) have been optimized on a separate validation set.

As competitors, we use HPS [4] and HBL [6], which were tested with their original experimental setting. As a baseline, we also include results on a model trained with the standard cross entropy classification loss. Table 1 and Table 2 show that hierarchical priors improve the performances, as the methods using the prototypes that embed hierarchical information always outperform the uniform ones. For cHPS_0 and HPS, our weak hierarchical prototypes provide the best results. Interestingly, HBL performs better with the full hierarchy prototypes, probably because the hyperbolic geometry allows for a better exploitation of the hierarchical representations [13]. Our proposed $\text{cHPS}_{\frac{1}{K}}$ has the best performances in almost every experiment, rewarding the pruning of the prototypes.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

On the other hand, weak hierarchical prototypes do not seem to provide benefits with this method. Investigating why full hierarchical prototypes benefits both HBL and cHPS$_{\frac{1}{K}}$ is an interesting direction for future research.

5 Conclusions

In this paper we introduce a way to weaken the hierarchial information, a contrastive loss for hyperspherical embeddings, and a prototype pruning mechanism. We provide experimental evidences that validates these contributions. Results show that: *i)* a weak use of the hierarchical information improves results in hyperspherical geometries when no prototype pruning is performed; *ii)* the contrastive loss function combined with prototype pruning compares favorably with state-of-the-art methods.

Experimental evidence also shows that combining prototype pruning and weakening the hierarchical information is not always beneficial. We believe this to be an interesting observation, explaining why this is the case is not trivial and worthy of further research.

References

[1] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE CVPR*, pages 3474–3482, 2018.

[2] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[3] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[4] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019.

[5] Loic Landrieu and Vivien Sainte Fare Garnot. Leveraging class hierarchies with metric-guided prototype learning. In *British Machine Vision Conference (BMVC)*, 2021.

[6] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *arXiv preprint arXiv:2106.14472*, 2021.

[7] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF CVPR*, pages 12506–12515, 2020.

[8] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF CVPR*, pages 2495–2504, 2021.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[11] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your" flamingo" is my" bird": fine-grained, or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11476–11485, 2021.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR*, pages 770–778, 2016.

[13] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *Graph Drawing: 19th International Symposium, GD 2011, Eindhoven, The Netherlands, September 21-23, 2011, Revised Selected Papers 19*, pages 355–366. Springer, 2012.