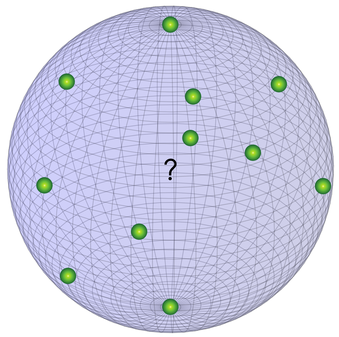
Hyperspherical Prototype Networks

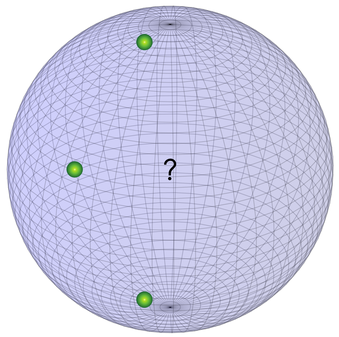
Pascal Mettes, Elise van der Pol, and Cees Snoek
University of Amsterdam

Hyperspheres as output spaces

In hyperspherical output spaces, classes can be described by prototypes that are defined **a priori** with large margin separation and prior knowledge.



How would you position two classes?

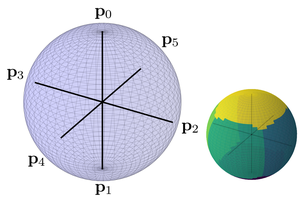


For cat, airplane tiger, what should go where?

Benefits over softmax cross-entropy
- ✔ Enables an embedding of inductive biases prior to learning.
- ✔ Freedom to choose any output dimensionality.
- ✔ Unification with regression using the same loss.

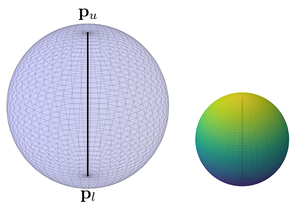
Benefits over other prototype approaches
- ✔ No chicken-egg problem (prototypes and inputs depend on each other).
- ✔ No need for expensive prototype updating.

Proposed approach



Classification:
- Prototypes no longer need to be learned.
- A priori positioning with separation.
- Move training examples to fixed prototypes.

Regression:
- Retain two prototypes on extreme ends.
- Prototypes state lower and upper bound.
- Same loss as for classification.
- Multi-task learning possible in same space.



Classification

Match examples with fixed class prototypes based on their angle:

$$\mathcal{L}_{\text{c}} = \sum_{i=1}^{N} (1 - \cos\theta_{\mathbf{z}_i, \mathbf{p}_{y_i}})^2 = \sum_{i=1}^{N} (1 - \frac{|\mathbf{z}_i \cdot \mathbf{p}_{y_i}|}{||\mathbf{z}_i|| \ ||\mathbf{p}_{y_i}||})^2$$

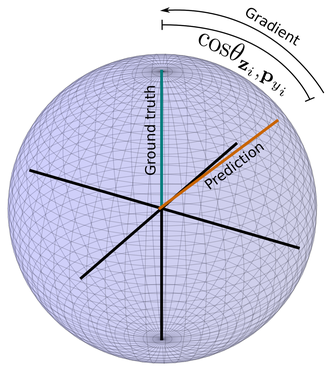
Training examples · Network output · Class prototype · Cosine similarity

Prototypes not updated, backpropagation through examples.

$$\frac{d}{\mathbf{z}_i}(1 - \cos\theta_i)^2 = 2(1 - \cos\theta_i)\left(\frac{\cos\theta_i \cdot \mathbf{z}_i}{||\mathbf{z}_i||^2} - \frac{\mathbf{p}_{y_i}}{||\mathbf{z}_i||||\mathbf{p}_{y_i}||}\right)$$

During inference, simply select class of nearest prototype:

$$c^* = \arg\max_{c \in C} \left(\cos\theta_{f_\phi(\tilde{\mathbf{x}}), \mathbf{p}_c}\right)$$



Prototype construction

Optimally separting arbitrary numbers of points for any dimensionality is an open mathematical problem. We resort to an approximation through optimization with the following objective:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}' \in \mathbb{P}} \left(\max_{(k,l,k \neq l) \in C} \cos\theta_{(\mathbf{p}'_k, \mathbf{p}'_l)}\right)$$

Separating nearest pair per step is inefficient. We opt to separate each prototype from its neighbour:

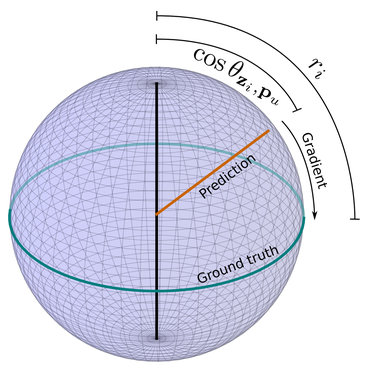
$$\mathcal{L}_{\text{HP}} = \frac{1}{K} \sum_{i=1}^{K} \max_{j \in C} \mathbf{M}_{ij}, \quad \mathbf{M} = \hat{\mathbf{P}}\hat{\mathbf{P}}^T - 2\mathbf{I}, \quad \text{s.t. } \forall_i \ ||\hat{\mathbf{P}}_i|| = 1$$

We also propose an extra ranking loss to preserve privileged class similarities:

$$\mathcal{L}_{\text{PI}} = \frac{1}{|T|} \sum_{(i,j,k) \in T} -\bar{S}_{ijk} \log S_{ijk} - (1 - \bar{S}_{ijk}) \log(1 - S_{ijk})$$

Number of triplets · $[\cos\theta_{\mathbf{w}_i, \mathbf{w}_j} \geq \cos\theta_{\mathbf{w}_i, \mathbf{w}_k}]$ Word embedding similarities · Sigmoid over prototype similarities

Regression and multi-task learning



Maintain two prototypes pointing in opposite directions.
Prototypes denote lower and upper regression bound:

$$\mathcal{L}_{\text{r}} = \sum_{i=1}^{N} (r_i - \cos\theta_{\mathbf{z}_i, \mathbf{p}_u})^2, \qquad r_i = 2 \cdot \frac{y_i - v_l}{v_u - v_l} - 1$$

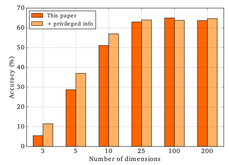
Regression value as interpolation between bounds

Joint regression and classification possible in same space.
One axis for regression, other axes for classification.
No need to balance tasks as they have the same loss.

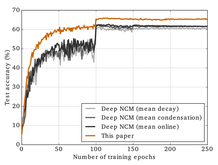
Classification results

Dimensions	CIFAR-100 10	CIFAR-100 25	CIFAR-100 50	CIFAR-100 100	ImageNet-200 25	ImageNet-200 50	ImageNet-200 100	ImageNet-200 200
One-hot	-	-	-	62.1 ± 0.1	-	-	-	33.1 ± 0.6
Word2vec	29.0 ± 0.0	44.5 ± 0.5	54.3 ± 0.1	57.6 ± 0.6	20.7 ± 0.4	27.6 ± 0.3	29.8 ± 0.3	30.0 ± 0.4
This paper	**51.1** ± 0.7	**63.0** ± 0.1	**64.7** ± 0.2	**65.0** ± 0.3	**38.6** ± 0.2	**44.7** ± 0.2	**44.6** ± 0.0	**44.7** ± 0.3

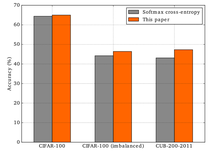
Our approach is preferred over one-hot and word2vec embeddings.
Separation matters especially when output spaces are small.
Above results for ResNet-32, results hold for DenseNet-121.



This paper with privileged info.



This paper vs Deep NCM.



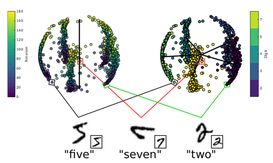
This paper vs cross-entropy.

Regression results

Output space / *Learning rate*	Omniart S^1 1e-2	Omniart S^1 1e-3	Omniart S^2 1e-2	Omniart S^2 1e-3
MSE	210.7 ± 140.1	110.3 ± 0.8	339.9 ± 0.0	109.9 ± 0.5
This paper	**84.4** ± 10.7	**76.3** ± 5.6	**82.9** ± 1.9	**73.2** ± 0.6

Predict creation year for 20th century paintings.
Our approach is stable across learning rates and output spaces.

Multi-task results



Visualization: Rotated MNIST
Classify on xy-plane, regress on z-axis.
Tasks are separated on same sphere.

Evaluation: OmniArt
Classify style, regress creation year.
We do both, without task weighting.

Task weight	0.01	0.10	0.25	0.50	0.90
Creation year (MAE ↓)					
MTL baseline	262.7	344.5	348.5	354.7	352.3
This paper	65.2	64.6	**64.1**	68.3	83.6
Art style (acc ↑)					
MTL baseline	44.6	47.9	49.5	47.2	47.1
This paper	46.6	51.2	**54.5**	52.6	51.4