Prototypical Contrastive Learning of Unsupervised Representations

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, Steven C.H. Hoi
Salesforce Research
Preprint, Under review

Introduction

Instance-wise contrastive learning representation:

- Positive pair: pull closer
- Negative pair: push apart

Address the fundamental limitations of instance-wise contrastive learning

- Semantic structure of data is not encoded by learned representation
- Negative examples are pushed far away regardless their similarity

Solution: assign several prototypes of different granularity

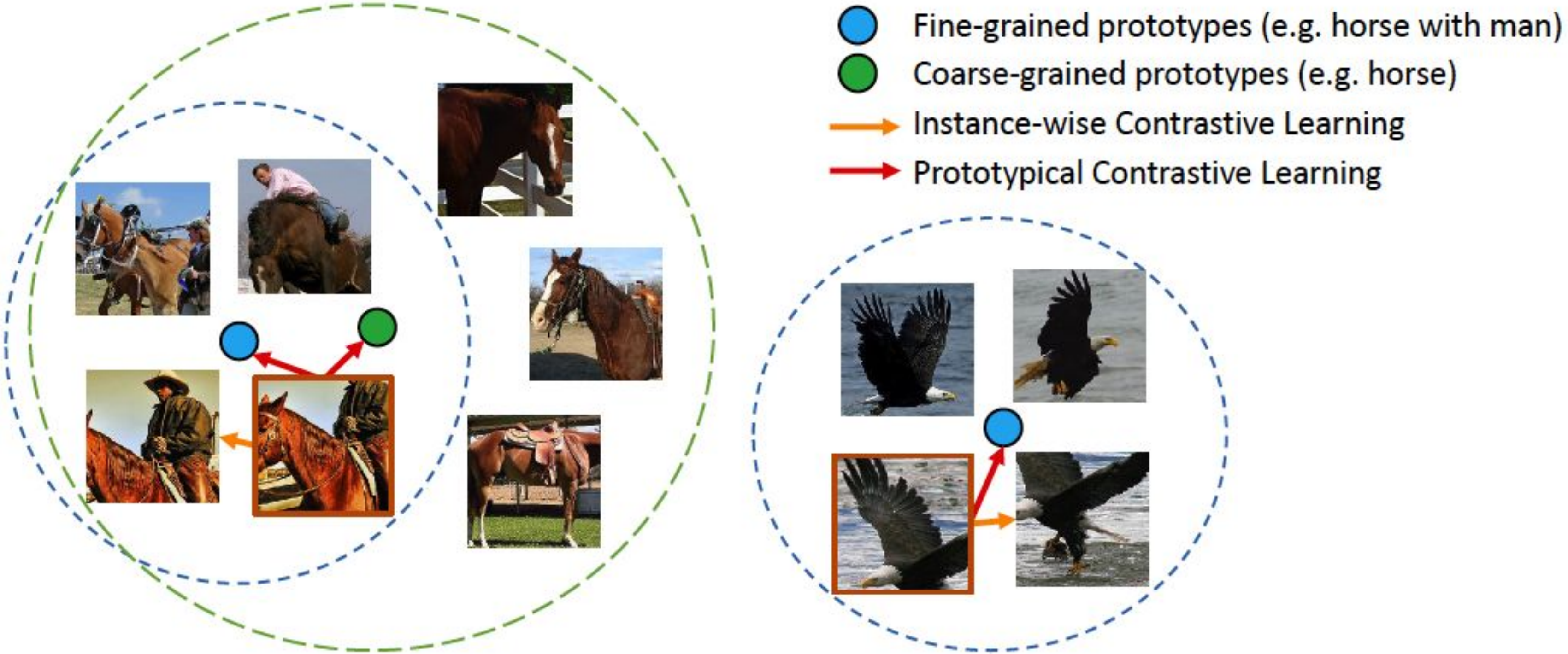

Figure 1: Illustration of Prototypical Contrastive Learning. Each instance is assigned to multiple prototypes with different granularity. PCL learns an embedding space which encodes the semantic structure of data.

Instance-wise Contrastive Learning

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{n} -\log \frac{\exp(v_i \cdot v'_i/\tau)}{\sum_{j=0}^{r} \exp(v_i \cdot v'_j/\tau)},$$

Where v’_i is positive embedding, v’_j is negative embedding

T is temperature hyper-parameters

Prototype contrastive learning

Optimization

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} -\log \frac{\exp(v_i \cdot c_s/\phi_s)}{\sum_{j=1}^{k} \exp(v_i \cdot c_j/\phi_j)},$$

Cluster concentration estimation

$$\phi = \frac{\sum_{z=1}^{Z} \|v'_z - c\|_2}{Z \log(Z + \alpha)},$$

ProtoNCE

$$\mathcal{L}_{\text{ProtoNCE}} = \sum_{i=1}^{n} -\left(\log \frac{\exp(v_i \cdot v'_i/\tau)}{\sum_{j=0}^{r} \exp(v_i \cdot v'_j/\tau)} + \frac{1}{M} \sum_{m=1}^{M} \log \frac{\exp(v_i \cdot c_s^m/\phi_s^m)}{\sum_{j=0}^{r} \exp(v_i \cdot c_j^m/\phi_j^m)} \right)$$

Algorithm 1: Prototypical Contrastive Learning.

Input: encoder f_θ, training dataset X, number of clusters $K = \{k_m\}_{m=1}^{M}$

$\theta' = \theta$ // initialize momentum encoder as the encoder

while not MaxEpoch **do**

  /* E-step */

  $V' = f_{\theta'}(X)$ // get momentum features for all training data

  for $m = 1$ **to** M **do**

    $C^m = k\text{-means}(V', k_m)$ // cluster V' into k_m clusters, return prototypes

    $\phi_m = \text{Concentration}(C^m, V')$ // estimate the distribution concentration around each prototype with Equation 12

  end

  /* M-step */

  for x **in** Dataloader(X) **do** // load a minibatch x

    $v = f_\theta(x), v' = f'_\theta(x)$ // forward pass through encoder and momentum encodeer

    $\mathcal{L}_{\text{ProtoNCE}}(v, v', \{C^m\}_{m=1}^{M}, \{\phi_m\}_{m=1}^{M})$ // calculate loss with Equation 11

    $\theta = \text{SGD}(\mathcal{L}_{\text{ProtoNCE}}, \theta)$ // update encoder parameters

    $\theta' = 0.999 * \theta' + 0.001 * \theta$ // update momentum encoder

  end

end

Result: Low-shot image classification

Method	architecture	VOC07					Places205				
		k=1	k=2	k=4	k=8	k=16	k=1	k=2	k=4	k=8	k=16
Random	ResNet-50	8.0	8.2	8.2	8.2	8.5	0.7	0.7	0.7	0.7	0.7
Supervised		55.6	65.0	73.9	79.4	81.7	15.5	21.0	26.7	31.9	35.9
Jigsaw [24, 36]	ResNet-50	26.5	31.1	40.0	46.7	51.8	4.6	6.4	9.4	12.9	17.4
MoCo [3]		31.2	40.5	50.6	58.9	65.6	9.1	13.2	17.7	23.3	28.4
PCL (ours)		**40.9**	**52.7**	**61.4**	**68.1**	**73.7**	**11.4**	**15.7**	**20.3**	**25.0**	**29.5**
SimCLR [8]	ResNet-50-MLP	35.2	42.9	53.7	60.5	67.0	9.9	14.1	19.3	23.8	28.5
PCL (ours)		**47.1**	**54.7**	**64.1**	**70.9**	**76.5**	**12.1**	**17.2**	**21.6**	**27.0**	**31.0**

Table 1: **Low-shot image classification** on both VOC07 and Places205 datasets using linear SVMs trained on fixed representations. All methods were pretrained on ImageNet-1M dataset (except for Jigsaw [24, 36] trained on ImageNet-14M). We vary the number of labeled examples k and report the mAP (for VOC) and accuracy (for Places) across 5 runs. Results for Jigsaw were taken from [36]. We use the released pretrained model for MoCo, and re-implement SimCLR. MoCo, SimCLR, and PCL are trained for the same number of epochs (200 epochs).