SWAV

Wednesday, 15. September 2021 10:47 PM

- Clustering method
- Uses transformations and contrastive loss
- Compute online
- Enforce consistency between different views of same image
- Swapping Assignments between multiple Views of the same image (SwAV)
- Clustering based
- Works with small and large batch sizes (no memory bank or momentum encoder)
- Uses many views (instead of just two) of the same image
- Multi-crop also improves other methods (SimCLR, Deep Cluster,…)

Mainly :

1) Scalable online cluster loss (+2% ImageNet)
2) Multi-crop (+2-4% ImageNet for SSL)
3) Combine these to reach +4.2% in total
4) First method w/o finetuning or pretraining

- Compute code for image and augmented versions
- Setup swapped prediction problem with loss :

$$L(\mathbf{z}_t, \mathbf{z}_s) \quad = \quad \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t),$$

- Z : features, q : codes

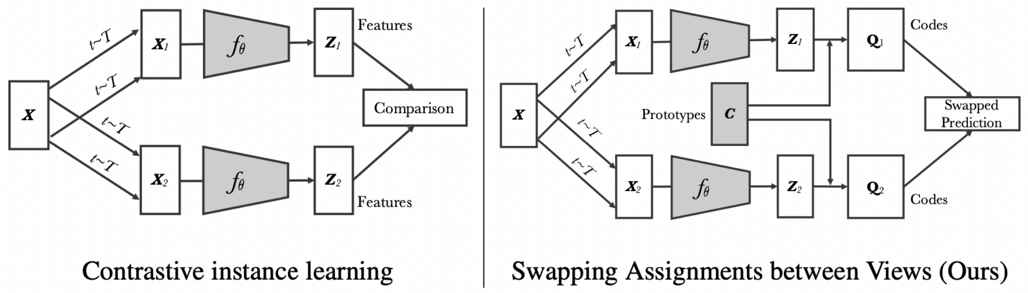

Figure 1: **Contrastive instance learning (left) *vs.* SwAV (right).** In contrastive learning methods applied to instance classification, the features from different transformations of the same images are compared directly to each other. In SwAV, we first obtain "codes" by assigning features to prototype vectors. We then solve a "swapped" prediction problem wherein the codes obtained from one data augmented view are predicted using the other view. Thus, SwAV does not directly compare image features. Prototype vectors are learned along with the ConvNet parameters by backpropragation.

- Only image features in the batch
- Map feature vectors to prototypes with Q :

$$\max_{\mathbf{Q}\in\mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q}),$$

=> maximize similarity between features and prototypes

where H is the entropy function, $H(\mathbf{Q}) = -\sum_{ij} \mathbf{Q}_{ij} \log \mathbf{Q}_{ij}$ and ε is a parameter that controls the smoothness of the mapping. We observe that a strong entropy regularization (*i.e.* using a high ε)

- => high entropy leads to a sort of mode collapse (uniform assignment to all protoypes
  => entropy low
- Inspired by Asanp et al. : contrain Q to belong to the transportation polytope (but with the batch)
- $$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K\times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\},$$
- => continious codes perform better here (rounding is more aggressive
  => fast convergence but lower accuracy)

We thus preserve the soft code $\mathbf{Q}^*$ instead of rounding it. These soft codes $\mathbf{Q}^*$ are the solution of Prob. (3) over the set $\mathcal{Q}$ and takes the form of a normalized exponential matrix [13]:

-

$$\mathbf{Q}^* = \mathrm{Diag}(\mathbf{u}) \exp\left(\frac{\mathbf{C}^\top \mathbf{Z}}{\varepsilon}\right) \mathrm{Diag}(\mathbf{v}), \tag{5}$$

- U and v (renorm vectors) are computed with Sinkhorn Knopp algorithm

-

As noted in prior works [10, 44], comparing random crops of an image plays a central role by capturing information in terms of relations between parts of a scene or an object. Unfortunately, increasing the number of crops or "views" quadratically increases the memory and compute requirements. We propose a `multi-crop` strategy where we use two standard resolution crops and sample V additional low resolution crops that cover only small parts of the image. Using low resolution images ensures only a small increase in the compute cost. Specifically, we generalize the loss of Eq (1):

$$L(\mathbf{z}_{t_1}, \mathbf{z}_{t_2}, \dots, \mathbf{z}_{t_{V+2}}) = \sum_{i \in \{1,2\}} \sum_{v=1}^{V+2} \mathbf{1}_{v \neq i} \ell(\mathbf{z}_{t_v}, \mathbf{q}_{t_i}). \tag{6}$$

- => only using full resolution crops

Results :

Method	Arch.	Param.	Top1
Supervised	R50	24	76.5
Colorization [65]	R50	24	39.6
Jigsaw [46]	R50	24	45.7
NPID [58]	R50	24	54.0
BigBiGAN [15]	R50	24	56.6
LA [68]	R50	24	58.8
NPID++ [44]	R50	24	59.0
MoCo [24]	R50	24	60.6
SeLa [2]	R50	24	61.5
PIRL [44]	R50	24	63.6
CPC v2 [28]	R50	24	63.8
PCL [37]	R50	24	65.9
SimCLR [10]	R50	24	70.0
MoCov2 [11]	R50	24	71.1
SwAV	R50	24	**75.3**

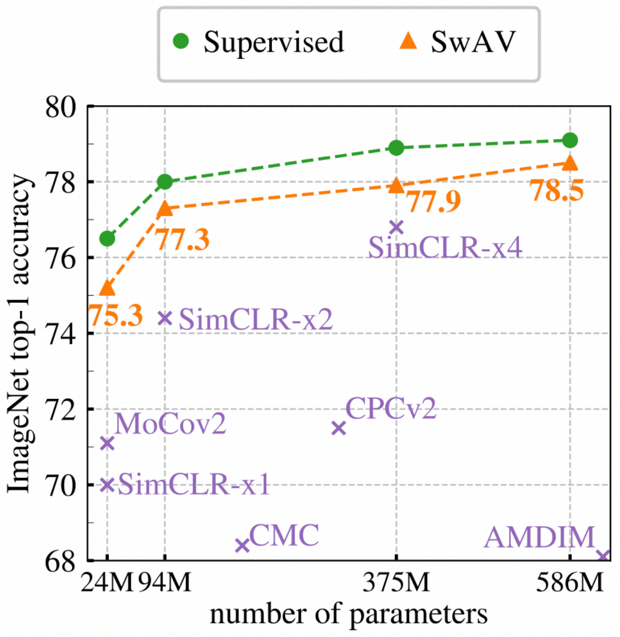

Figure 2: **Linear classification on ImageNet.** Top-1 accuracy for linear models trained on frozen features from different self-supervised methods. **(left)** Performance with a standard ResNet-50. **(right)** Performance as we multiply the width of a ResNet-50 by a factor ×2, ×4, and ×5.