Review：Unsupervised Learning of Visual Features by Contrasting Cluster Assignments

SwAV proposes a new self-supervised learning paradigm to learn feature representation by comparing clustering assignments of different views, avoiding direct feature comparison.

Main idea:

- comparing clustering assignments of different views instead of directly comparing sample features
- Online clustering mechanism is introduced to dynamically learn prototype vectors.
- Sinkhorn-Knopp algorithm is used to optimize clustering assignment.
- Multi-crop strategy is proposed to significantly improve performance.

The training process of SwAV consists of the following steps:

1. Data augmentation: Generate two different augmented views x_t, x_s for each image
2. Feature extraction: Get features z_t = f(x_t), z_s = f(x_s) through encoder f
3. Prototype mapping: Match features with a set of prototype vectors {c1,...,cK} to get scores
4. Encoding calculation:
    - Use Sinkhorn-Knopp algorithm to calculate soft clustering assignment to get encoding q_t, q_s
    - Use softmax to calculate probability distribution to get p_t, p_s
5. Cross prediction: Minimize cross prediction loss L(z_t, z_s) = l(z_t, q_s) + l(z_s, q_t) Where l(z,q) measures the degree of fit of feature z to the prediction of encoding q

Formula derivation and description:

L(zt, zs) = ℓ(zt, qs) + ℓ(zs, qt)

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

The core loss function in SwAV.

zt and zs: the feature representations of two different data-augmented views of the same image

qt and qs: the "encodings" (cluster assignment probabilities) corresponding to these two features.

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}, \quad \text{where} \quad \mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)}. \qquad (2)$$

q_s^(k): the soft clustering assignment obtained by the Sinkhorn-Knopp algorithm

p_t^(k): the probability distribution of the similarity between feature z_t and prototype c_k after softmax

τ: a temperature parameter used to adjust the smoothness of the distribution

$$-\frac{1}{N}\sum_{n=1}^{N}\sum_{s,t\sim\mathcal{T}}\left[\frac{1}{\tau}\mathbf{z}_{nt}^{\top}\mathbf{C}\mathbf{q}_{ns}+\frac{1}{\tau}\mathbf{z}_{ns}^{\top}\mathbf{C}\mathbf{q}_{nt}-\log\sum_{k=1}^{K}\exp\left(\frac{\mathbf{z}_{nt}^{\top}\mathbf{c}_k}{\tau}\right)-\log\sum_{k=1}^{K}\exp\left(\frac{\mathbf{z}_{ns}^{\top}\mathbf{c}_k}{\tau}\right)\right].$$

Average the entire batch of N samples and all possible augmentation pairs (s, t)

$$\max_{\mathbf{Q}\in\mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^{\top}\mathbf{C}^{\top}\mathbf{Z}\right)+\varepsilon H(\mathbf{Q}), \tag{3}$$

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) generally leads to a trivial solution where all samples collapse into an unique representation and are all assigned uniformely to all prototypes. Hence, in practice we keep ε low. Asano *et al.* [2] enforce

The term Tr(Q^T C^T Z) maximizes the similarity between feature Z and prototype C.

$$\mathcal{Q}=\left\{\mathbf{Q}\in\mathbb{R}_{+}^{K\times B}\mid \mathbf{Q}\mathbf{1}_B=\frac{1}{K}\mathbf{1}_K,\mathbf{Q}^{\top}\mathbf{1}_K=\frac{1}{B}\mathbf{1}_B\right\}, \tag{4}$$

where 1K denotes the vector of ones in dimension K. These constraints enforce that on average each prototype is selected at least B K times in the batch.

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

Q* is the optimal solution to the optimization problem (3), that is, the optimal soft clustering assignment matrix, which is a K×B matrix that represents the soft assignment probability of B samples to K prototypes. u and v are renormalized vectors with dimensions K and B respectively

u = ones(K)/K

v = ones(B)/B

exp(C^T Z/ε) calculates the similarity between the feature and the prototype

Diag(u) and Diag(v) ensure that the equilibrium constraint is met

Use the Sinkhorn-Knopp algorithm to iteratively calculate the renormalized vectors u and v

Only 3 iterations are needed to get good results

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

z_t1, z_t2 are two full-resolution views

z_t3 to z_t(V+2) are V low-resolution views

1_{v≠i} means not comparing with itself

The above formula is equal to:

Comparison between full resolution views+Comparison between low resolution view and

full resolution view

$$I(z_{t1}, q_{t2}) + I(z_{t2}, q_{t1}) + \sum_{v=3}^{V+2} [I(z_{tv}, q_{t1}) + I(z_{tv}, q_{t2})]$$