Template

Summary

Research Objective

Problem Statement(What is the problem to be solved?)

Methods

Evaluation

Conclusion

Note

Reference

Unsupervised Pre-Training of Image Features on Non-Curated Data

Summary

This paper aims at bridging the gap of performance between models trained on curated dataset and ones trained on non-curated dataset. It leverages the combination of classification and clustering. The way it builds is based on RotNet and DeepCluster. The pseudo-labels are generated via hierarchical clustering and the algorithm learns linear classifiers which is applied on the learned representations w.r.t the pseudo-label in each hierarchy and the losses are built upon this regime.

It also uses distributed training where each GPU is responsible for a specific super-class.

One of the questions is whether the images of the clusters overlap with each other. RotNet and DeepCluster should also be checked.

Research Objective(作者的研究目标)

- Bridge the gap between unsupervised learning on the curated dataset and on the raw dataset.
- Capture complementary statistics from large scale of data via combining classification and clustering

Problem Statement(What is the problem to be solved?)

- Convnets on pretrained data perform well but collecting large curated dataset is effort-costing
- Simply discarding labels doesn't undo the effect of collecting curated dataset
- Previous unsupervised learning are trained on curated dataset
- Cluster relying on inter-image similarities are sensitive to data distribution

Methods

General Description

- Automatically generates targets by clustering the features of the entire dataset, under constraints derived from self-supervision.
- propose a hierarchical formulation that is suitable for distributed training.
- clustering

$$\min_{C\in\mathbb{R}^{d\times k}} \sum_{n=1}^{N} \left[\min_{z_n\in\{0,1\}^k \text{ s.t. } z_n^\top \mathbf{1}=1} \|Cz_n - f_\theta(x_n)\|_2^2 \right]$$

- Self-supervised learning to do classification

$$\min_{\theta,V} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, V f_\theta(x_n)),$$

V is linear classifier

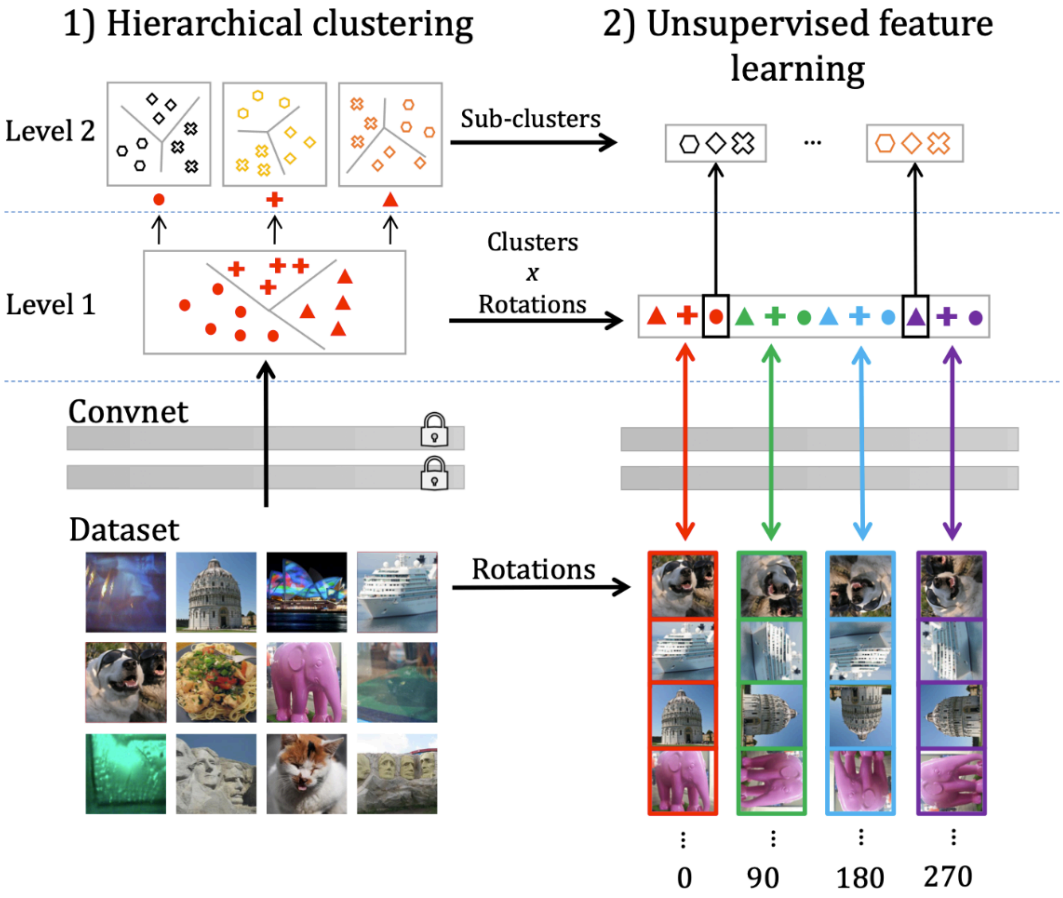
Steps

- Use RotNet to extract useful features.
- Combining clustering and classification

$$\min_{\theta,W} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n \otimes z_n, W f_\theta(x_n)).$$

  Instead of simply summing the above two equations, assuming the two tasks are independent, it **captures more signals** via forming the problem based on the Cartesian product of the target label(y_n) and the class label(z_n)

- Scaling up to large number of targets
  - Since the Cartesian product would be the order of $O(|\mathcal{Y}||\mathcal{Z}|)$(clusters), it is required to design a method to reduce computation cost. Then it comes *hierarchical training*.



- First predict the 4*m superclasses(level 1), 4 types of rotations and m superclasses. Then divide each superclass into subclasses.
- An image that does not belong to the super-class does not belong either to any of its sub-classes.
- Generate the super class labels and sub class labels via hierarchical KMeans
- Learn linear classifier in each hierarchy to classify the representations to the generated pseudo-labels

Evaluation(How to evaluate this method?)

Pretrain on ImageNet

Since derived from RotNet and DeepCluster, compare the ImageNet classification performance with these two methods

Comparing clustering of the method with directly supervising

Conclusion(Strong or weak conclusion)

well-suited for distributed training

With such amount of data, our approach su passes unsupervised methods trained on curated datasets, which validates the potential of unsupervised learning in applications where annotations are scarce or curation is not trivial.

References

1. Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Confer- ence on Computer Vision (ECCV), 2018
2. Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rota- tions. In International Conference on Learning Representa- tions (ICLR), 2018.
3. Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash.
   Boosting self-supervised learning via knowledge transfer. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
4. Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. arXiv preprint arXiv:1901.09005, 2019.
5. Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo.
   Aet vs. aed: Unsupervised representation learning by auto- encoding transformations rather than data. arXiv preprint arXiv:1901.04596, 2019.
6. Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In Proceedings of the European Conference on Computer Vision (ECCV), 2016
7. Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros.
   Context encoders: Feature learning by inpainting. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Unsupervised Learning of Visual Features by Contrasting Cluster Assignments

Summary

This paper aims at improving the models pretrained using unsupervised learning methods, especially contrastive learning, as well as reducing the computation and memory cost, which is achieved by avoid using memory bank or momentum network and adopting the strategy of online computing. The basic idea of this paper is to maintain the consistency between features computed from different data augmentation on the same image, using swapped prediction, i.e. having the model predict the code of one augmentation using the feature of the other augmentation. Code are generated by learnable prototypes[4].

Research Objective

- Previous contrastive learning methods:
    - Contrastive loss
    - Image transformation
- Computing all pairwise loss not practical
    - Relax loss computation
    - Relax task from pairwise contrast to clustering
- Propose:
    - Compute code online
    - Enforce consistency between different views of the same image
    - Predict the code of a view from representation of another view
    - multi-crop strategy to increase the number of views of an image with no computational or memory overhead.

Problem Statement(What is the problem to be solved?)

- Improve the performance of unsupervised pretraining on downstream tasks
- Avoid using memory bank or momentum network which is memory costing

Methods

General Description

- Main idea is to enforce consistency between codes from different augmentations of the same image.
- The codes are computed from **trainable** prototypes.
- Swapped prediction:

$$L(z_t, z_s) = l(z_t, q_s) + l(z_s, q_t)$$

    - l is cross-entropy loss, s and t are two different data augmentations of the same image
    - Taking this loss over all the images and pairs of data augmentations leads to the following loss function for the swapped prediction problem

Steps

- Computing codes online[4]
    - compute codes using the prototypes C such that all the examples in a batch are **equally** partitioned by the prototypes.

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

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

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

  - Solution found in the loss function above is called a **continuous code**. Discretize it called **discrete code** using methods in [4] empirically works better in the whole dataset but in the batch-wise case which is adopted in this paper, continuous code works better.
  - Preserve the continuous code and take the form of a normalized exponential matrix[5]

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

  - Here the procedure of calculating Q is the same as SeLa. The code can be understood as pseudo-labels in SeLa

- Dealing with small batches
  - Augment the training batch with the features calculated in the last batches; Compute the code using the augmented set; training with only the original batch
- Multi-crop: Augmenting views with smaller images
  - Increasing the number of crops or “views” quadratically increases the memory and compute requirements
  - Use two standard resolution crops and sample V additional low resolution crops that cover only small parts of the image.

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

Evaluation

- pretrain on ImageNet
- Evaluate on downstream tasks

Conclusion

Note

Reference

[1] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)

[2] Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. arXiv preprint arXiv:1902.06162 (2019)

[3] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the Conference on Computer Vision and Pattern Recognition

[4] Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and repre- sentation learning. International Conference on Learning Representations (ICLR) (2020)

[5] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems (NeurIPS) (2013)

SELF-LABELLING VIA SIMULTANEOUS CLUSTERING AND REPRESENTATION LEARNING

Summary

This method proposes a self-supervised method that basically first assigns pseudo-label to the data points then train a classifier under the supervision of the pseudo-labels. It claims that using KMeans to assign pseudo-labels is not a general solution and is prone to lead to a degenerated solution - all data points are in one cluster and features are learned to be constant. Therefore, it instead proposes a way that constrains the data points to be equally partitioned by the clusters using EM. The learning objective is consistent between the representation learning step and classification step - Cross Entropy Loss, which is claimed to be superior than DeepCluster.

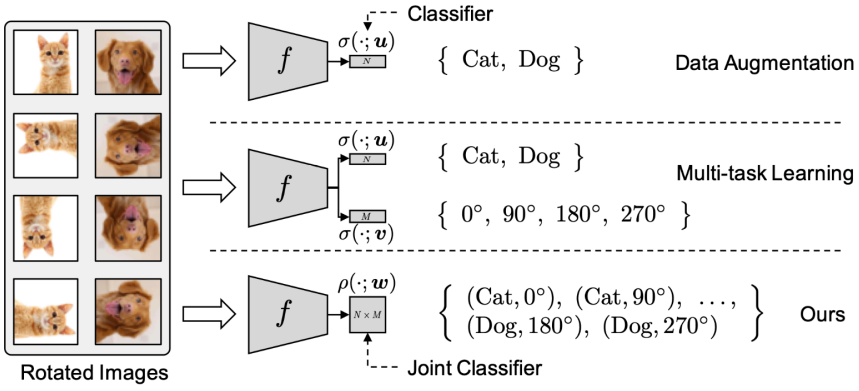
It is also pointed out that the data points can be clustered in many different equally good ways via concatenating with different classification heads. - *This remains a question*

RETHINKING DATA AUGMENTATION: SELF-SUPERVISION AND SELF-DISTILLATION

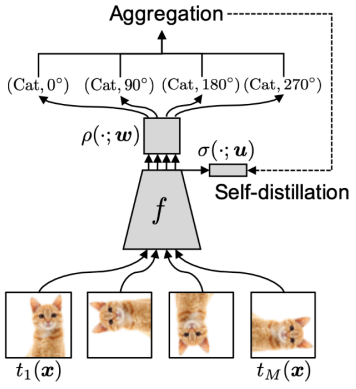
ICML 2020; Hankook Lee, Sung Ju Hwang, Jinwoo Shin

Summary

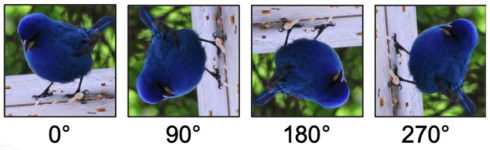
The paper points out that the the invariance to data augmentation on some tasks may hurt performance. The multi-task setting is thus not rationale. Instead, the author proposes to avoid forcing the invariance to the data augmentation by adopting the setting of predicting the joint probability of data augmentation label and self-supervise label. From my understanding, the avoidance of invariance refers to predicting the data label based on the conditional information of data augmentation instead of forcing all the prediction be invariant of augmentation. During the inference time, since the data augmentation is known, for all the possible choices of data augmentation, we can predict the conditional information based on the applied data augmentation and then aggregate among them. The performance can be further boosted by training another classifier that could incorporate the data augmentation info during training termed by **self-distillation**.



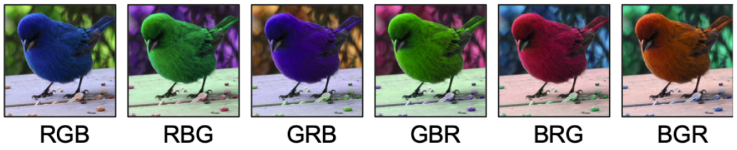
(a) Difference with previous approaches.



(b) Aggregation & self-distillation.



(c) Rotation ($M = 4$).



(d) Color permutation ($M = 6$).

Main equations

- Multi-task(conventional one that may hurt performance on some tasks):
- Joint classifier:

$$\mathcal{L}_{\text{MT}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{u}, \boldsymbol{v}) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}); \boldsymbol{u}), y) + \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}); \boldsymbol{v}), j)$$

$$P_{\text{aggregated}}(i|\boldsymbol{x}) = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{w}_{ij}^{\top} \tilde{\boldsymbol{z}}_j.$$

- Aggregated Inference:

$$P_{\text{aggregated}}(i|\boldsymbol{x}) = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{w}_{ij}^{\top} \tilde{\boldsymbol{z}}_j.$$

- Self-distillation:

$$\begin{aligned}\mathcal{L}_{\text{SDA+SD}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{u}) = \mathcal{L}_{\text{SDA}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{w}) \\ + D_{\text{KL}}(P_{\text{aggregated}}(\cdot|\boldsymbol{x}) \| \sigma(f(\boldsymbol{x}; \boldsymbol{\theta}); \boldsymbol{u})) + \beta \mathcal{L}_{\text{CE}}(\sigma(f(\boldsymbol{x}; \boldsymbol{\theta}); \boldsymbol{u}), y)\end{aligned}$$