Perceptual Image Similarity for Unsupervised Representation Learning

Berkan Hiziroglu
Brown University

Advisor: Stephen Bach

May 2020

Abstract

Neural networks have the potential to learn useful features from raw data and successfully use them for a variety of downstream tasks. The performance of a machine learning model relies upon the quality of the learned representations. This work focuses on improving the performance of unsupervised representation learning models by adding image generation as an extra task and using perceptual image similarity as a loss function. Our experiments show that our method works on different neural network architectures and on different datasets. Our method achieves state of the art representation learning performance for AlexNet on CIFAR-10, CIFAR-100, SVHN and yields significant improvement on ImageNet.

1 Introduction

Learning from data without any labels is an important paradigm in machine learning because it is scalable and low-cost. Unsupervised learning algorithms learn the underlying distribution of the data without having access to labels and the learned representations can be used for different applications.

Image representation learning models learn embeddings for images. These embeddings can be used for a variety of downstream tasks. It is important to develop unsupervised learning models because there are real world problems that do not have labeled data to train a model on. Learned representations are generalizable and well suited for being used on different applications because an unsupervised model is primarily interested in learning the underlying distribution of the input data, as opposed to learning to fit on the data for a specific task as in supervised learning.

Training unsupervised models is hard mainly because there are no labels to train on. The lack of labels make the learning signal weaker and inaccurate. There are statistical models that can learn from data without labels such as K-means but these methods lack generalizability and do not work as well as neural networks on learning the underlying distribution.

Most of the work in the unsupervised learning literature create a pretext task that can be constructed from data alone such as clustering, inpainting and generation. Occasionally, there is a manual algorithm that prevents the neural network from mapping all data points to the same class or cluster. Better pretext tasks lead to better performance so the literature is focused on finding better pretext tasks. There is no definite logic in the reason why some pretext tasks yield a better representation. The only way to find whether a pretext task is superior to another is to train an unsupervised model on it and see the results. Our method is a novel and general approach that is suitable to any pretext task.

Our novel approach consists of adding a generator to an unsupervised learning model and training it along with the model using a perceptual similarity metric. Our method can be used on different architectures and on different datasets. We show that our method achieves state of the art representation learning performance on CIFAR-10, CIFAR-100 and SVHN datasets.

Summary of Contributions:

- Using perceptual similarity metric for image generation from a learned representation
- Improving the performance of multiple unsupervised learning techniques by adding image generation as an extra learning task

2 Related Work

All neural networks inherently do representation learning. In the supervised domain, the network learns a good representation of the data and a classifier for the task at the same time. In the unsupervised domain, representation learning is done as an exclusive task and the performance of the model is determined by how well it does in a downstream task using the learned representations.

Unsupervised learning tasks in the literature include inpainting [10], colorization [14], generation [3] and transformation prediction [4]. Recent state of the art work in the literature focus on self labeling and clustering. Self labeling models use a network to cluster the data and then use the cluster assignments as labels. [1] follows a similar approach, they apply K-means on learned representations to generate cluster assignments and train the model using these assignments. They repeat this process after every epoch. Naively following this approach would make the network map every data point to the same cluster. To prevent this, the authors manually move data points from crowded clusters to empty clusters. [12] also uses a neural network to map data points to cluster and then run a linear program to convert the cluster assignments to labels. The linear program is used to prevent the network from mapping every data point to the same cluster.

3 Methods

3.1 Self Labeling

In the supervised learning setting, neural networks are trained on a task that requires data with labels. The parameters of the network are learned by optimization with respect to a loss function that uses the predictions of the model and the ground truth labels. Since there are no ground truth labels in the unsupervised setting, there needs to be a task that can be constructed from data. In this project, we are focused on self labeling and clustering as pretext tasks.

Self labeling methods use a neural network to label the data and use these labels to train the network. It is an iterative process, after every epoch or every few epochs, the network labels all data points again and use the new labels to train. Self labeling methods usually learn the parameters of the network by optimization with respect to an energy function, as opposed to a loss function. Iteratively updating the labels and optimizing with respect to an energy function is guaranteed to continuously improve the quality of the labels. This does not mean that the process is able to converge to a globally-optimum point. Please see [8] and [2] for details.

We follow the work of [1] and [12] and treat the data labels as cluster assignments. In [1], parameters of the network are learned by classification and new cluster are assigned by applying K-means on learned representations after every epoch. [12] also uses classification to learn the parameters of the network. After every epoch, they use the network to create a label matrix. Then, they run a linear program on top of that matrix with a constraint to put equal number of data points in each cluster and use the output of the linear program as new labels. We tried our method on both these work and improved their performance.

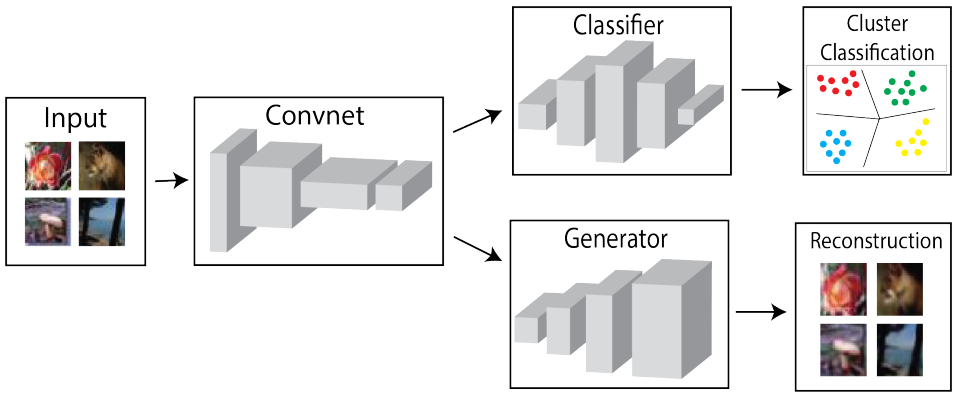

Figure 1: Illustration of our method. Features learned by the convolutional part are used for classification and generation. Classifier assigns images to clusters and cluster assignments are used as pseudo-labels to train the network. Generator reconstructs the input images from the learned representation. Figure is inspired from Caron et al. [1]

3.2 Reconstruction

In our method, learned representations are used both for the pretext task and input reconstruction. Exploring data labels and learning the parameters of a neural network simultaneously is a method that is used in the literature and our contribution is a simple generator that helps in learning better representations. As shown in Figure 1, the generator takes the output of the last convolutional layer and reconstructs the input image. L2 and perceptual similarity distances are used to train the generator. We experimented training the generator with different loss functions and the addition of L2 and perceptual similarity gave the best results. Figure 2 shows the quality of the reconstructions with different loss functions.

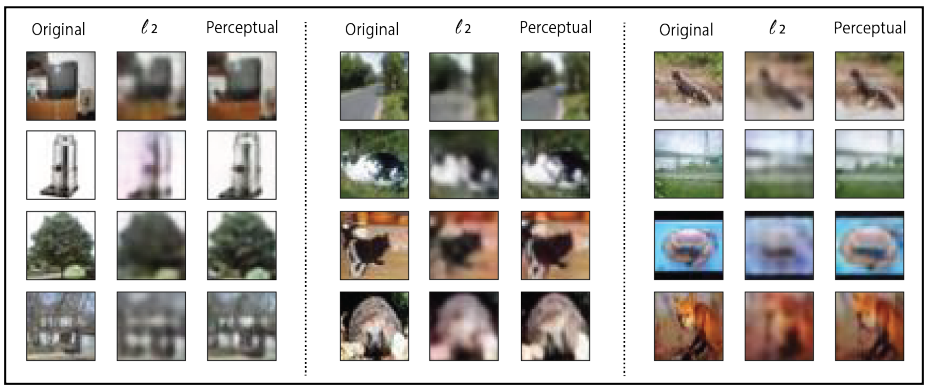

Figure 2: Reconstructions of the generator when trained with L_2 and perceptual similarity [15] as the loss function.

4 Experiments

4.1 Datasets

We have used CIFAR [6], ImageNet ILSVRC-12 [11] and Street View House Numbers [9] datasets in our experiments. These datasets are all standard benchmarks for evaluation in representation learning.

Method	Dataset		
	CIFAR-10	CIFAR-100	SVHN
Classifier/Feature	Linear Classifier / conv5		
Supervised	91.8	71.0	96.1
Counting	50.9	18.2	63.4
DeepCluster	77.9	41.9	92.0
Instance	70.1	39.4	89.3
AND	77.6	47.9	93.7
SL	83.4	57.4	94.5
Our method	**83.6**	**58.5**	**94.6**
Classifier/Feature	Weighted kNN / FC		
Supervised	91.9	69.7	96.5
Counting	41.7	15.9	43.4
DeepCluster	62.3	22.7	84.9
Instance	60.3	32.7	79.8
AND	74.8	41.5	90.9
SL	77.6	44.2	92.8
Our method	**77.9**	**44.5**	**92.9**

Table 1: Linear classification and Nearest Neighbour evaluation on CIFAR and SVHN using AlexNet. Results of previous methods are taken from [12].

4.2 Approach

A good representation should capture a distribution similar to the distribution of the raw data. We use the learned representations as an input to a supervised model to perform evaluation. We test the quality of the learned representations using linear probes following [5]. After the unsupervised training is finished, a linear classifier is trained on top of the pre-trained network. The pre-trained network's parameters are not optimized during this step and only the parameters of the linear classifier are trained. We use AlexNet for the neural network architecture [7].

Although linear probing is the standard evaluation method for unsupervised representation learning, some work in the literature add non-linearity to the evaluation process to increase the performance and get results close to the supervised upper-bound. For non-linear probing experiments, we apply our method on AET model [13] and report the results on Table 3.

4.3 Results

Table 1 shows the results on CIFAR-10, CIFAR-100 and SVHN datasets. Our method gets a higher score than all previous methods. SL [12] was the previous state of the art model in unsupervised representation learning and applying our method to their model achieves a new state of the art.

Table 2 shows the ImageNet experiments. We did not include all previous methods and we did not use the largest model available in SL [12]. These experiments show that our method is able to increase the quality of the learned representations on a large-scale dataset like ImageNet as well.

Method	c3	c4	c5
Inpainting	21.0	19.8	15.5
BiGAN	31.0	29.9	28.0
Instance retrieval	31.8	34.1	35.6
RotNet	38.7	38.2	36.5
AND	35.9	39.7	37.9
SL*	40.1	42.1	38.8
Our method	**40.5**	**42.7**	**39.1**

Table 2: Linear classification evaluation on ImageNet using AlexNet. Linear probes are inserted after the third, fourth and fifth convolutional layers. * smaller version of SL [12] is used. See Experiments section for details. Results of previous methods are taken from [12]

Method	Error rate
Roto-Scat + SVM	17.7
ExamplarCNN	15.7
DCGAN	17.2
Scattering	15.3
RotNet + FC	10.94
RotNet + conv	8.84
AET-affine + FC	9.77
AET-project + FC	**9.41**
Our method	**9.01**

Table 3: Non-linear classification evaluation on CIFAR-10 using AlexNet. These experiments use a non-linear activation functions in linear probe evaluation. The generator is added to AET-project + FC setup. Results of previous methods are taken from [13]

5 Conclusion and Future Work

We build on the current state-of-the-art work in unsupervised representation learning and experiment with a different architecture and set-up to further extend the strength of the learned representations. Adding a generator and doing two pretext tasks from a single representation has the potential to make the model learn better features from raw data. The next steps in this work include extending the generator architecture to force the representation to include more information about the data and using different perceptual similarity metrics.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. *CoRR*, abs/1807.05520, 2018.

[2] Yann Le Cun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. December 2005.

[3] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *CoRR*, abs/1907.02544, 2019.

[4] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[5] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. *CoRR*, abs/1904.11567, 2019.

[6] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[8] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. *A tutorial on energy-based learning*. MIT Press, 2006.

[9] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[10] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[12] Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020.

[13] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. AET vs. AED: unsupervised representation learning by auto-encoding transformations rather than data. *CoRR*, abs/1901.04596, 2019.

[14] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016.

[15] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018.