Visual Similarity from Optimizing Feature and Memory On A Hypersphere

Xinlei Pan Rudrasis Chakraborty Stella X. Yu
UC Berkeley / ICSI
{xinleipan,rudra,stellayu}@berkeley.edu

Abstract

Supervised learning of classification from annotated images develops a latent feature representation that captures semantic visual similarity. We propose an unsupervised metric learning method that develops apparent visual similarity from images alone. Our method maps high-dimensional visual data onto a low-dimensional hypersphere and consolidate such feature representations into a visual memory representation. Optimizing the feature mapping and visual memory on a hypersphere achieves maximal discrimination among instances. Our formulation and solution is not only more principled in theory than closely related unsupervised instance discrimination algorithms, but also better in practice in terms of classification accuracy, convergence rate, and feature transferability. We also show that our learned feature can be very useful for vision-based reinforcement learning tasks to improve sample efficiency.

1. Introduction

Supervised learning with deep neural nets (DNN) has made a significant progress in various computer vision tasks [5, 3]. The features learned often implicitly capture visual similarities among the input data. As supervised learning methods require labeled data and obtaining big labeled data is labor and time intensive, unsupervised learning methods that can learn from unlabeled data have received wide attention recently [14, 1, 4].

Our approach towards unsupervised learning from visual data is inspired by the fact that learned feature representations on deep neural networks for image classification tasks usually implicitly cluster images of the similar classes together. Similar images of different classes can sometimes be mis-classified due to their visual similarity [14]. Intuitively, by mapping images to a low dimensional hypersphere and forcing the distance between features of all images to be as far as possible, the learned feature representation will cluster similar images together while putting distinctive images apart.

Similar ideas have been explored in previous works [14, 1, 2]. In particular, Wu *et al*. [14] explored instance-wise discriminative learning as the approach towards unsupervised feature learning and has achieved superior results on image classification tasks. However, the instance-wise discriminative method does not guarantee that image features will be distributed onto the low dimensional hyper-sphere appropriately. Since the features are on a unit hyper-sphere, euclidean distance used in their method fails to capture similarities between features. In this work, we propose a novel unsupervised learning method based on Riemannian optimization on hypersphere. Our proposed algorithm maps training data onto a unit hyper-sphere and then do the optimization on the hypersphere. In Fig. 1 we show the comparison of the learned feature representation by our proposed approach and Wu *et al*. [14]. Clearly, features learned by our method are distributed appropriately on hyper-sphere and images of the same class are clustered together while the features learned by Wu *et al*. [14] mix together for images of different classes.

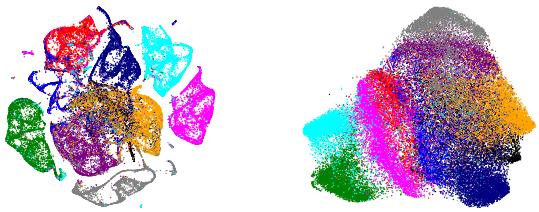

Figure 1: Visualization of learned low dimensional feature representations on CIFAR10 dataset. Left: ours; Right: Wu *et al*. [14]. The visualization is done using t-SNE dimension reduction [9]. Different colors indicate different image classes.

2. Related Works

Unsupervised feature learning has received increasing attention in recent years. Related approaches included generative models such as Auto-Encoders [13], Restricted Bolztmann Machines (RBMs) [12], generative adversarial networks [4] and Variational Auto-Encoders [6]; metric learning models, for example, Wu *et al*. [14] gives a non-parametric instance discrimination based unsupervised learning method.

3. Unsupervised Learning on A Hypersphere

Our model can be formulated as a neural network that takes in images and outputs a low dimensional vector representation. The optimization goal is to maximize the feature difference between different image instances as much as possible. Specifically, our model has two sets of parameters: one is the parametric feature mapping function $\mathbf{f}(x;\theta)$ that takes an image to a point on a unit hypersphere (denoted by $\mathbf{S}^m$), and the other is the non-parametric feature memory bank V that stores the consolidated representation for all the training instances. Let $\{x_i\}_{i=1}^n$ be the set of images. We use $\mathbf{f}_i = \mathbf{f}(x_i;\theta)$ to denote the feature corresponds to image x_i. As each $\mathbf{f}_i$ lies on $\mathbf{S}^m$, $\|\mathbf{f}_i\| = 1$, for all x_i. For a self-contained model derivation, we first review the Riemannian geometry of the hypersphere $\mathbf{S}^m$. **Tangent space:** $T_{\mathbf{p}}\mathbf{S}^m$ at $\mathbf{p} \in \mathbf{S}^m$ is defined by $T_{\mathbf{p}}\mathbf{S}^m = \left\{\mathbf{u} \in \mathbf{R}^{m+1} | \mathbf{u}^t\mathbf{p} = 0\right\}$.**Riemannian metric:** Given $\mathbf{p} \in \mathbf{S}^m$ and $\mathbf{u}, \mathbf{v} \in T_{\mathbf{p}}\mathbf{S}^m$, the Riemannian metric $g_{\mathbf{p}} : T_{\mathbf{p}}\mathbf{S}^m \times T_{\mathbf{p}}\mathbf{S}^m \to \mathbf{R}$ is $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{u}^t\mathbf{v}$ Note that this is the induced metric from the Euclidean metric on the ambient space $\mathbf{R}^{m+1}$. **Geodesic distance:** The geodesic distance on $\mathbf{S}^m$ induced by the above Riemannian metric is $d(\mathbf{p}, \mathbf{q}) = \arccos(\mathbf{p}^t\mathbf{q})$. where $\mathbf{p}, \mathbf{q} \in \mathbf{S}^m$. **Riemannian exponential map:** Given $\mathbf{p} \in \mathbf{S}^m$ and $\mathbf{u} \in T_{\mathbf{p}}\mathbf{S}^m$, the Riemannian exponential map, $\mathsf{Exp}_{\mathbf{p}}(\mathbf{u})$ is: $\mathsf{Exp}_{\mathbf{p}}(\mathbf{u}) = \cos(\|\mathbf{u}\|)\mathbf{p} + \sin(\|\mathbf{u}\|)\frac{\mathbf{u}}{\|\mathbf{u}\|}$ **Riemannian inverse exponential map:** Given $\mathbf{p} \in \mathbf{S}^m$, let $\mathcal{B}_r(\mathbf{p}) = \{\mathbf{q} \in \mathbf{S}^m | d(\mathbf{p}, \mathbf{q}) < r\}$ be the geodesic ball of radius r centered at $\mathbf{p}$. If $r < \pi/2$, then inside $\mathcal{B}_r(\mathbf{p})$, $\mathsf{Exp}_{\mathbf{p}}$ is a diffeomorphism and hence has an inverse. The inverse exponential map is given by: $\mathsf{Exp}_{\mathbf{p}}^{-1}(\mathbf{q}) = \frac{\theta}{\sin(\theta)}(\mathbf{q} - \mathbf{p}\cos(\theta))$, where, $\theta = d(\mathbf{p}, \mathbf{q})$ and $\mathbf{q} \in \mathcal{B}_r(\mathbf{p}) \subset \mathbf{S}^m$. Note that r is the injectivity radius of $\mathbf{S}^m$ at $\mathbf{p}$. With image x mapped to feature $\mathbf{f}(x;\theta)$, the probability of x as an observation of instance class $\mathbf{v}_i$ in memory depends on the distance between x and $\mathbf{v}_i$ and is given by: $P(x;\theta,V) = \frac{\exp(-d^2(\mathbf{f}(x;\theta),\mathbf{v}_i)/T)}{\sum_{j=1}^n \exp(-d^2(\mathbf{f}(x;\theta),\mathbf{v}_j)/T)}$, where, T is the tunable hyperparameter. Observe that the key difference between the formulation of [14] and our is that in our case, the probability depends on the arc cosine distance while the probability in [14] depends on the cosine similarity. Assume that $x_1, \cdots, x_n$ are i.i.d. samples. The joint probability of drawing samples $x_1, \cdots, x_n$ is given by:

$$P(x_1, \ldots, x_n; \theta, V) = \prod_{i=1}^n P(x_i; \theta, V) = \prod_{i=1}^n \frac{\exp(-d^2(\mathbf{f}_i, \mathbf{v}_i)/T)}{\sum_{j=1}^n \exp(-d^2(\mathbf{f}_i, \mathbf{v}_j)/T)}. \quad (1)$$

The optimal feature mapping θ^* and visual memory $V*$ should be obtained by maximizing the above likelihood or equivalently minimizing the negative log likelihood $\ell(\theta, V)$ as given below:

$$\begin{aligned}(\theta^*, V^*) &= \arg\max_{\theta,V} P(x_1, \ldots, x_n; \theta, V) \\ &= \arg\min_{\theta,V} \ell(\theta, V) \\ \ell(\theta, V) &= -\log P(x_1, \ldots, x_n; \theta, V) \\ &= \sum_{i=1}^n \left(d^2(\mathbf{f}_i, \mathbf{v}_i)/T + \log\left(\sum_{j=1}^n \exp(-d^2(\mathbf{f}_i, \mathbf{v}_j)/T) \right) \right).\end{aligned}$$

Minimizing the objective function $\ell(\theta, V)$: Now, we give the steps to optimize the objective function ℓ. We will use SGD to learn θ. As each column of V lies on $\mathbf{S}^m$, we will use Riemannian stochastic gradient descent (SGD) to optimize $\{\mathbf{v}_i\}$. Notice that although we have used Jensen's inequality to lower bound the objective function, we have minimized the original objective function instead of the lower bound. In contrast to the method by [14], our method has the flexibility to learn both the parameters (θ) as well as the memory bank (V). And we will emperically see that this gives better feature learning. By taking gradient of $\ell(\theta, V)$ with respect to $\mathbf{v}_i$, we get:

$$\frac{\partial \ell}{\partial \mathbf{v}_i} = -\mathsf{Log}_{\mathbf{v}_i}(\mathbf{f}_i) + \sum_{j=1}^n \frac{\exp\left(-d^2(\mathbf{f}_j, \mathbf{v}_i)/T\right)}{\sum_{k=1}^n \exp\left(-d^2(\mathbf{f}_k, \mathbf{v}_i)/T\right)} \mathsf{Log}_{\mathbf{v}_i}(\mathbf{f}_j) \quad (2)$$

Now we state and prove some propositions that we will need to show the convergence of the Riemannian SGD.

3.1. k-Nearest Neighbor Classifiers

In order to perform classification at the testing time, we use k nearest neighbor (k-NN) method. Namely, we calculate the similarity between the input instance's feature vector $\mathbf{f}_\theta(x)$ and all other training instances $\{x_i\}_{i=1}^N$'s feature vectors $\{\mathbf{f}_\theta(x_i)\}_{i=1}^N$: $w(\mathbf{f}_\theta(x), \mathbf{f}_\theta(x_i)) = \mathbf{f}_\theta(x)^T\mathbf{f}_\theta(x_i)$. Then we select the top k instances within the N training data points that have the largest $w(\mathbf{f}_\theta(x), \mathbf{f}_\theta(x_i))$, and then calculate the weighted class score, where every class gets a score $S(c) = \sum_{i=1}^k w(\mathbf{f}_\theta(x), \mathbf{f}_\theta(x_i))1(c_i = c)$. Here c_i is the class label for the i-th instance. The class with the maximum score will be assigned as the prediction of the class of the input testing instance.

4. Experiments

We investigate the difference between our method and Wu *et al.*'s work [14] in unsupervised image classification in terms of image classification accuracy, convergence rate and cross dataset generalization ability. We will introduce the experimental set up and the dataset we use in our experiments. In addition to unsupervised image classification

problems, we also evaluate the quality of the learned representation by applying it as an initialization for a reinforcement learning task and also compare with [14]. We investigate whether the learned feature representation can be beneficial for improving sample effeciency for general vision based reinforcement learning tasks.

4.1. Experimental Setup

The purpose of our experiment is to investigate the following questions. Does our method achieve better image classification accuracy than Wu *et al*. [14] within the same number of epochs? Does our method converge faster than Wu *et al*. [14]? Given the model trained, can the learned feature extraction method be generalized to another distinct dataset? Given the feature representation learned, is the feature representation useful for other task other than image classification? To answer the above questions, we designed the following evaluation experiments.

Unsupervised Image Classification. We compare with Wu *et al*. [14] on unsupervised image classification task. Since in [14] they already compared with existing unsupervised learning approaches, we only need to compare with this baseline approach. We consider network architecture of ResNet-50 [5] for learning the low dimensional feature representations. For both methods, the network is randomly initialized. We consider the following datasets for this experiment: the MNIST dataset [8]; the CIFAR-10 [7] dataset; the street view house number (SVHN) dataset [11]. We use a temperature $T = 1.0$ after tuning that hyperparameter. We train the unsupervised feature learning network on these datasets using both methods, and evaluate the accuracy on the held-out testing datasets after every epoch. The best accuracy within the first 400 epochs are compared.

Convergence Evaluation. We evaluate on MNIST, CIFAR-10, SVHN of the convergence rate using our methods and the baseline methods to investigate whether our method performs better than the baseline in terms of convergence. We compare the convergence rate of our model versus Wu *et al*. [14] by comparing the per epoch best accuracy. Specifically, the best accuracy for both methods up till epoch t are compared and plotted.

Transfer Learning. We evaluate whether the learned feature representation can be generalizable between different datasets and evaluated the transfer from MNIST to SVHN dataset, and SVHN to MNIST, as well as MNIST to a new dataset, the FashionMNIST dataset [15]. The final best accuracy among the first 400 epochs is obtained and compared against Wu *et al*. [14]. The convergence of the transfer learning is also evaluated.

Unsupervised Feature Learning for Deep Reinforcement Learning. In this work, we also apply the feature to a reinforcement learning task to help reduce sample complexity in deep reinforcement learning. We focus on training a

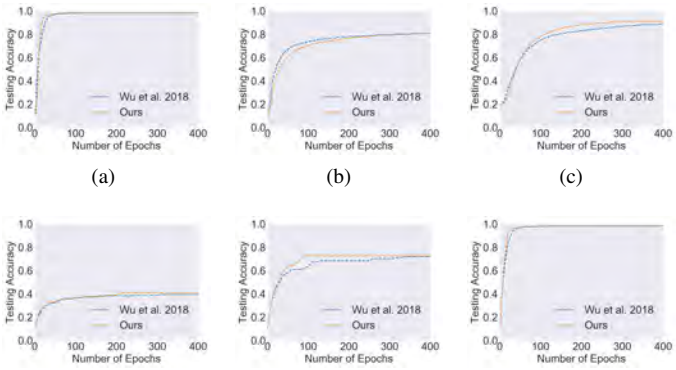

Figure 2: Results on convergence rate. (a) MNIST, (b) CIFAR10, (c) SVHN, (d) MNIST transfer to SVHN, (e) SVHN transfer to MNIST, (f) MNIST transfer to Fashion-MNIST.

deep Q learning policy (DQN) [10]. We first train the feature extractor with visual data collected int he simulation environment. Then we use the feature extractor to serve as the first several layers for the policy network and then train the policy together with the pretrained feature extractor. To show the improvement of sample efficiency, we compare this with the case where the network is randomly initialized. The policy network is composed of the convolutional neural network used in [10] as a feature extractor followed by two fully connected layers of output dimensions 512 and n_a, where n_a is the number of discrete actions in the environment that we evaluate.

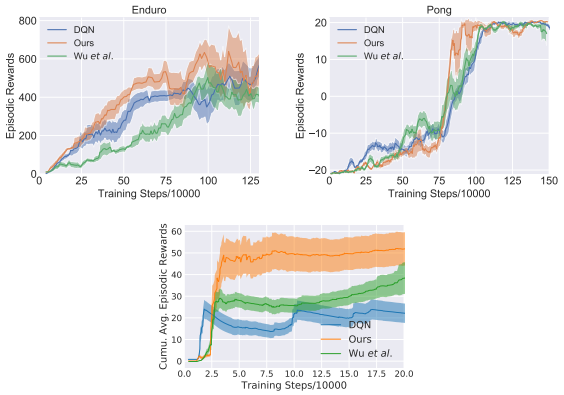

Figure 3: Reinforcement learning task training reward curve for Enduro (first), Pong (second), and TORCS (third) environment.

4.2. Results and Analysis

We show in Table 1 the results of evaluating model's accuracy on MNIST, CIFAR10 and SVHN datasets. It can be seen from the table that our methods outperforms Wu *et al*. [14] on all three datasets, and the gap on SVHN is significant. In Figure 2's first three sub-figures, we show the con-

vergence curve of accuracy for our method and the method from Wu *et al*. [14]. Our method still outperforms Wu *et al*. [14] on MNIST, CIFAR10 and SVHN, with a large gap in SVHN dataset. In Table 2, we show the transfer learning final accuracy for both methods on MNIST, SVHN and the Fashion MNIST dataset. To compare the transfer learning convergence rate, we show in Figure 2 the convergence curve for both methods on all three target datasets. From these results, it's clear that our method outperforms Wu *et al*. [14] in terms of image classification accuracy, convergence rate, as well as transferability.

Table 1: Comparison of image classification accuracy on various image classification datasets.

Method	MNIST	CIFAR10	SVHN
Ours	**98.51%**	**82.53%**	**91.39%**
[14]	97.85%	80.65%	88.84%

Table 2: Transfer Learning Results. Columns correspond to the source dataset, and rows correspond to the target dataset.

Target\Source	MNIST	SVHN
MNIST (Ours)	-	**72.74%**
SVHN (Ours)	**40.63%**	-
FashionMNIST (Ours)	**98.37%**	-
MNIST [14]	-	71.88%
SVHN [14]	39.41%	-
FashionMNIST [14]	98.21%	-

Reinforcement Learnign Results. In terms of improving sample efficiency in DRL, we put in Fig. 3 the illustration that our approach helps to improve the sample efficiency of vision based reinforcement learning tasks, with comparison with Wu *et al*. [14].

5. Conclusion

In this work, we propose a novel manifold hypersphere unsupervised learning method for feature representation learning. We compared with one of the state of the art method in this field and our results show that our method exceeds the performance of [14] in terms of better accuracy, convergence rate and also transferability. Moreover, we show by experiments in reinforcement learning that our learned feature extractor can help to improve sample efficiency in vision based reinforcement learning tasks.

References

[1] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 1

[2] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 1

[3] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 3

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[7] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 3

[8] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 3

[9] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 1

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 3

[11] 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*, volume 2011, page 5, 2011. 3

[12] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Robust boltzmann machines for recognition and denoising. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2264–2271. IEEE, 2012. 1

[13] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 1

[14] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3, 4

[15] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 3