Learning Deep Parsimonious Representations

Renjie Liao[1], Alexander Schwing[2], Richard S. Zemel[1,3], Raquel Urtasun[1]

[1]University of Toronto [2]University of Illinois at Urbana-Champaign [3]Canadian Institute for Advanced Research

email: {rjliao, zemel, urtasun}@cs.toronto.edu, aschwing@illinois.edu

Introduction & Related Work

Good representations should at least be ***Parsimonious***, ***Interpretable*** and ***Generalizable***. Many existing methods try to achieve some of these goals. Methods like weight decay put regularizer on weights while Dropout, Denoise AutoEncoder, Contractive AutoEncoder, DeCov directly regularize the hidden representations. Here we exploit clustering loss $\mathcal{R}$ as a regularizer which leads to the overall objective $\mathcal{L} + \lambda\mathcal{R}$, where $\mathcal{L}$ is the conventional loss function, like cross-entropy.

Clustering Regularization

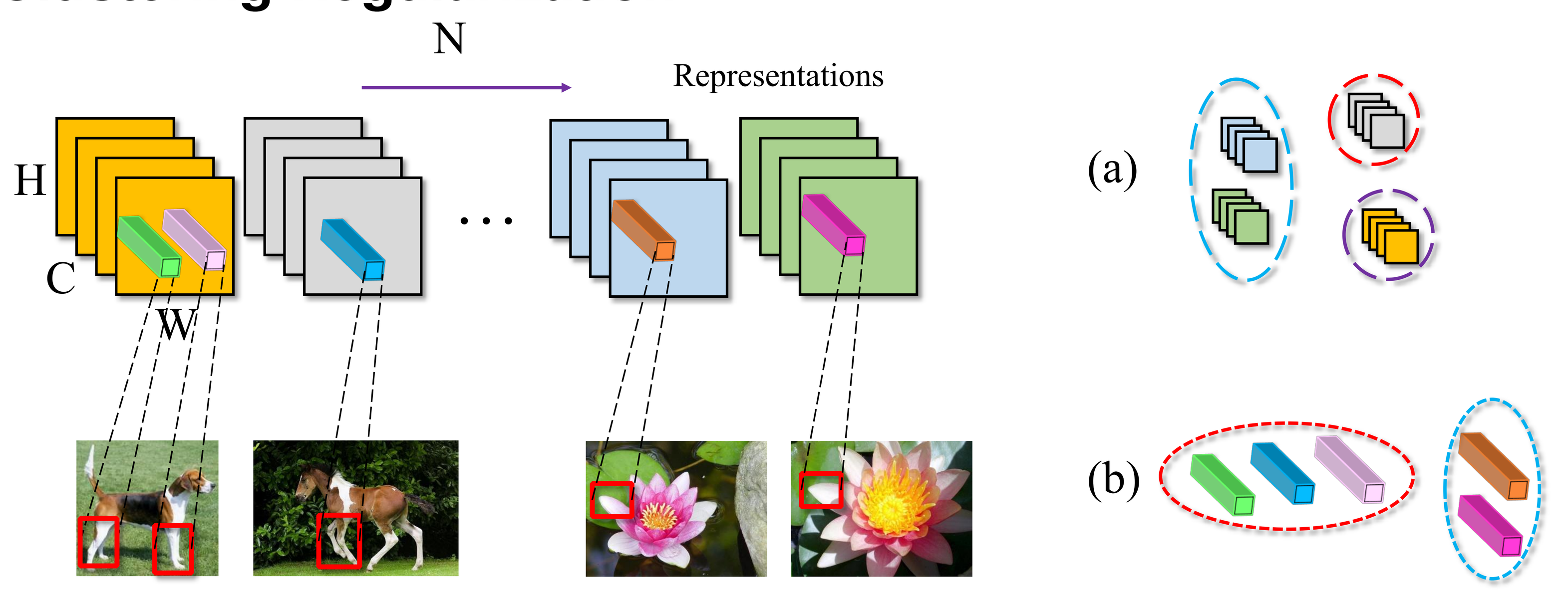

Figure 1: (A) Sample clustering and (B) spatial clustering. Samples, pixels, and channels are visualized as multi-channel maps, cubes, and maps in depth respectively. The receptive fields in the input image are denoted as red boxes.

Assuming the representation of one layer in a neural network is a 4-D tensor $\mathbf{Y} \in \mathbb{R}^{N \times C \times H \times W}$, we show three different types of clustering as below,

Sample Clustering

It captures global patterns shared among different samples,

$$\mathcal{R}_{sample}(\mathbf{Y}, \mu) = \frac{1}{2NCHW} \sum_{n=1}^{N} \left\| T^{\{N\} \times \{H,W,C\}}(\mathbf{Y})_n - \mu_{z_n} \right\|^2. \quad (1)$$

Spatial Clustering

It captures local patterns shared among different spatial positions,

$$\mathcal{R}_{spatial}(\mathbf{Y}, \mu) = \frac{1}{2NCHW} \sum_{i=1}^{NHW} \| T^{\{N,H,W\} \times \{C\}}(\mathbf{Y})_i - \mu_{z_i} \|^2. \quad (2)$$

Channel Co-Clustering

It captures global patterns shared among different samples and within each sample,

$$\mathcal{R}_{channel}(\mathbf{Y}, \mu) = \frac{1}{2NCHW} \sum_{i=1}^{NC} \| T^{\{N,C\} \times \{H,W\}}(\mathbf{Y})_i - \mu_{z_i} \|^2. \quad (3)$$

Learning Parsimonious Representations

Initialization: Maximum training iteration R, batch size B, smooth weight α, set of clustering layers $\mathcal{S}$ and set of cluster centers $\{\mu_k^0 | k \in [K]\}$, update period M
For iteration $t = 1, 2, ..., R$:
    For layer $l = 1, 2, ..., L$:
        Compute the output representation of layer l as x.
        If $l \in \mathcal{S}$:
            Assigning cluster $z_n = \operatorname{argmin}_k \|\mathbf{X}_n - \mu_k^{t-1}\|^2, \forall n \in [B]$.
            Compute cluster center $\hat{\mu}_k = \frac{1}{|\mathcal{N}_k|} \sum_{n \in \mathcal{N}_k} \mathbf{X}_n$, where $\mathcal{N}_k = [B] \bigcap \{n | z_n = k\}$.
            Smooth cluster center $\mu_k^t = \alpha\hat{\mu}_k + (1 - \alpha)\mu_k^{t-1}$
        End
    End
    Compute the gradients with cluster centers μ_k^t fixed.
    Update weights.
    Update drifted cluster centers using Kmeans++ every M iterations.
End

Experiments

Auto-Encoder

Measurement	Train	Test
AE	2.69 ± 0.12	3.61 ± 0.13
AE + Sample-Clustering	2.73 ± 0.01	3.50 ± 0.01

Table 1: Autoencoder Experiments on MNIST. We report the average of mean reconstruction error over 4 trials and the corresponding standard deviation.

Image Classification

Dataset	CIFAR10 Train	CIFAR10 Test	CIFAR100 Train	CIFAR100 Test
Caffe	94.87 ± 0.14	76.32 ± 0.17	68.01 ± 0.64	46.21 ± 0.34
Weight Decay	95.34 ± 0.27	76.79 ± 0.31	69.32 ± 0.51	46.93 ± 0.42
DeCov	88.78 ± 0.23	79.72 ± 0.14	77.92	40.34
Dropout	99.10 ± 0.17	77.45 ± 0.21	60.77 ± 0.47	48.70 ± 0.38
Sample-Clustering	89.93 ± 0.19	**81.05** ± 0.41	63.60 ± 0.55	**50.50** ± 0.38
Spatial-Clustering	90.50 ± 0.05	**81.02** ± 0.12	64.38 ± 0.38	**50.18** ± 0.49
Channel Co-Clustering	89.26 ± 0.25	**80.65** ± 0.23	63.42 ± 1.34	**49.80** ± 0.25

Table 2: CIFAR10 and CIFAR 100 results. For DeCov, no standard deviation is provided for the CIFAR100 results. All our approaches outperform the baselines.

Visualization on CIFAR10

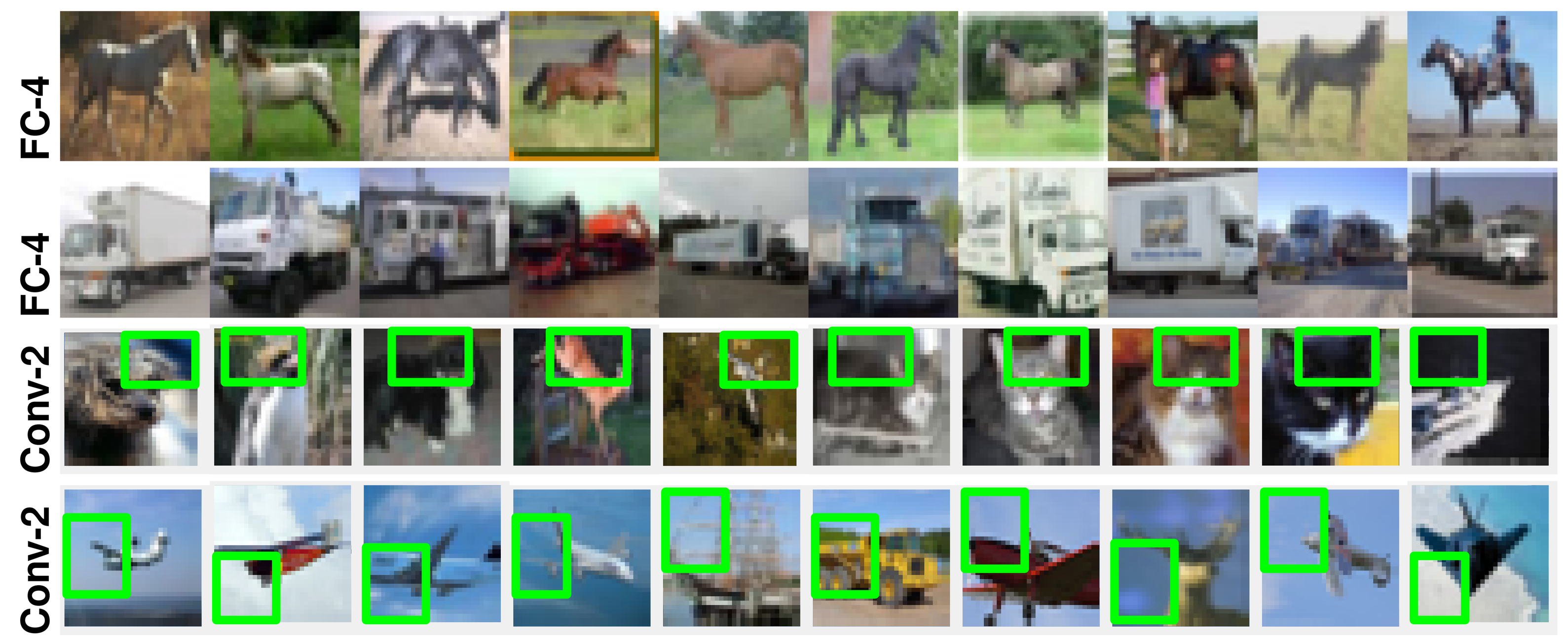

Figure 2: Visualization of clusterings on CIFAR10 dataset. Rows 1, 2 each show examples belonging to a single sample-cluster; rows 3, 4 show regions clustered via spatial clustering. Receptive fields are truncated to fit images.

Evaluation of Clustering

Method	Baseline	Sample-Clustering
NMI	0.4122 ± 0.0012	0.4914 ± 0.0011

Table 3: Normalized mutual information of sample clustering on CIFAR100.

Fine-Grained Classification

Method	Train	Test
DeCAF	-	58.75
Sample-Clustering	100.0	**61.77**
Spatial-Clustering	100.0	61.67
Channel Co-Clustering	100.0	61.49

Table 4: Classification accuracy on CUB-200-2011.

Visualization on CUB

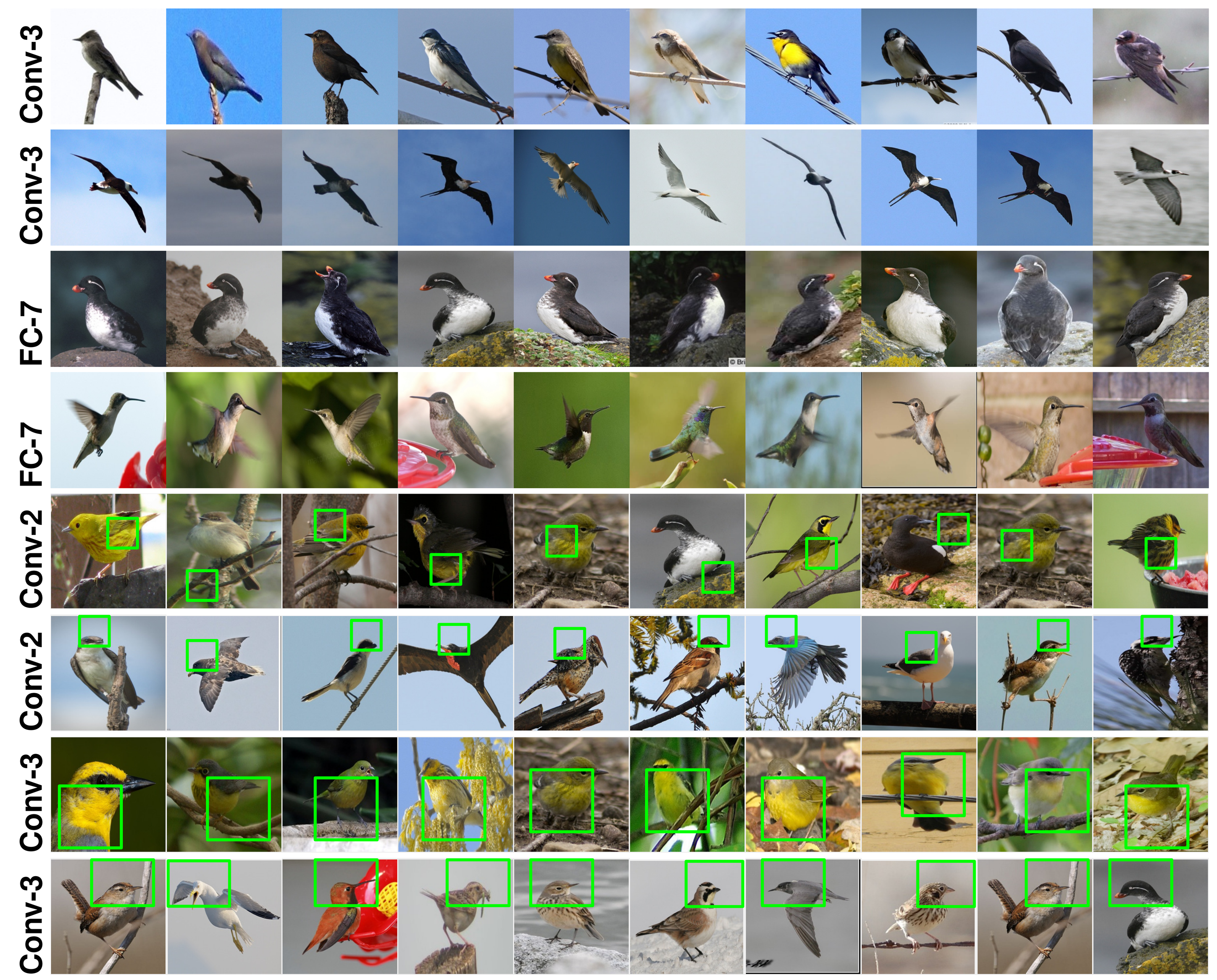

Figure 3: Visualization of sample and spatial clustering on CUB-200-2011 dataset. Row 1-4 and 5-8 show sample and spatial clusters respectively. Receptive fields are truncated to fit images.

Zero-Shot Learning

Based on the learned representations, we perform zero-shot learning via solving the following unregularized structure SVM,

$$\min_{W} \quad \frac{1}{N} \sum_{n=1}^{N} \max_{y \in \mathcal{Y}} \left\{ 0, \Delta(y_n, y) + x_n^\top W \left[\phi(y) - \phi(y_n)\right] \right\}. \quad (4)$$

The results are listed as below,

Method	Top1 Accuracy
ALE	26.9
SJE	40.3
Sample-Clustering	**46.1**

Table 5: Zero-shot learning on CUB-200-2011.

Future Work

- Back-propagate the gradient through unrolled steps of K-means
- Exploit soft cluster assignments
- Apply to semi-supervised tasks

Code is available at https://github.com/lrjconan/deep_parsimonious.