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

NIPS, 2016
Presenter: Beilun Wang

Outline

Outline

Motivation

Motivation:

- Advanced Neural Network (NN) needs regularization, which is key to prevent overfitting and improve generalization of the learned classifier.
- No neural network representations to form clusters.
- Not that related to term “Parsimonious Representations”?

Problem Setting:

- Input: Training set
- Target: Regularized Deep Neural Net considering different clusters (e.g., sample clustering, spatial clustering, channel co-clustering).
- In this talk, I’ll focus on sample clustering.

Outline

Previous Solutions

- Batch Normalization : imposing constrains in the mini-batch
- Dropout : prevent co-adaption
- K-means clustering

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - Notations
   - Problem setting
   - Problem formulation
3. Summary

Contributions

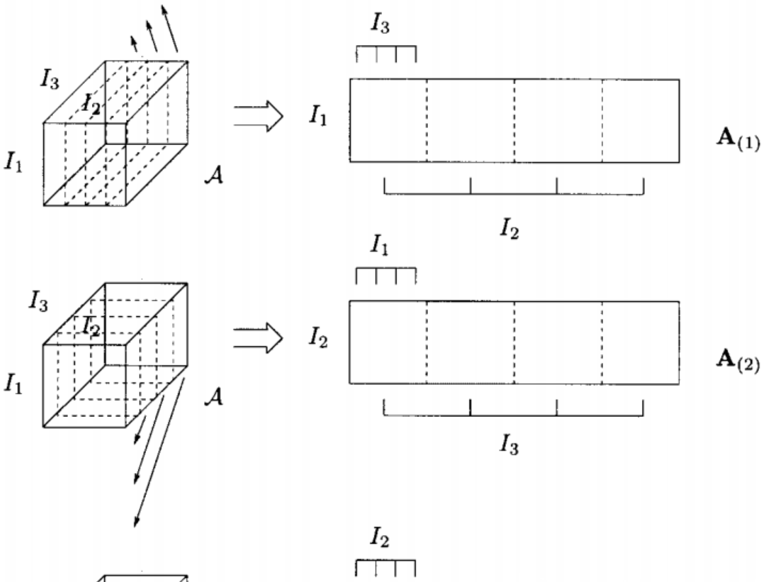
- a new type of regularization that encourages the network representations to form clusters
- This benefits unsupervised learning and zero-shot learning.
- Certain equations in this paper is problematic.

Outline

Notations

- $[K]$: $\{1, 2, \ldots, K\}$.
- $\backslash$: The sets substraction.
- $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$: An n-mode vectors of a D-order tensor.
- $T^{\{I_n\} \times \{I_j | j \in [D] \backslash n\}}$: the N-node matrix unfolding.

The N-node matrix unfolding



The N-node matrix unfolding

A whiteboard example.

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - Notations
   - Problem setting
   - Problem formulation
3. Summary

Problem setting

We assume the representation of one layer within a neural network to be a $4-D$ tensor $\mathbf{Y} \in \mathbb{R}^{N \times C \times H \times W}$.

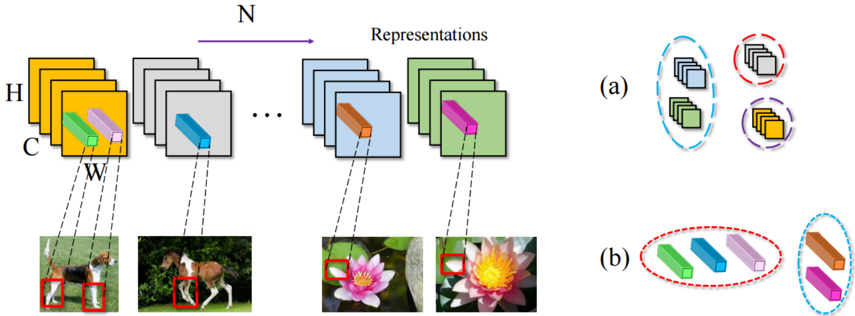
- N: the number of samples within a mini-batch
- C: the number of hidden units in this layer
- H: the height of the output of this layer
- W: the width of the output of this layer

For example, $H = W = 1$ when this layer is a fully connected layer.

Calculate H and W

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$

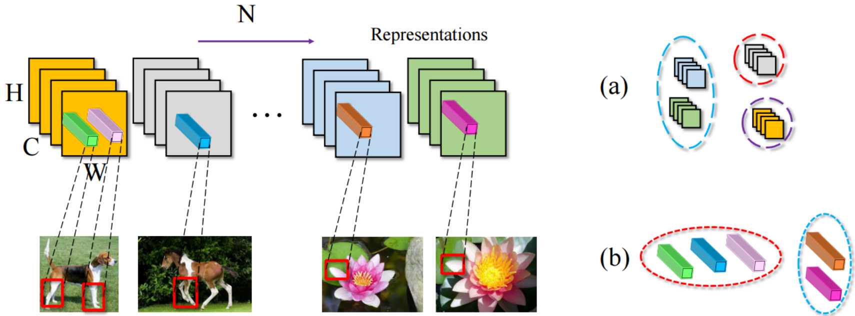
Problem setting



Problem setting: Clustering in different layers

- Bottom layer representations may focus on low-level visual cues, such as color and edges.
- Top layer features may focus on high-level attributes which have a more semantic meaning.
- See the examples in the figure.

Problem setting



Outline

Key insight: Regularized formulation

To use the clusters in a certain layer, this paper choose the following formulation:

$$\arg\min \mathcal{L} + \mathcal{R} \tag{1}$$

Where $\mathcal{L}$ is the loss function and $\mathcal{R}$ is a regularizer push the clustering structure in a certain layer.

The problem left is the formulation of $\mathcal{R}$.

Key insight: Sample Clustering Regularizer

Suppose $\mathbf{Y} \in \mathbb{R}^{N \times C \times H \times W}$, the matrix unfolding of $\mathbf{Y}$ by the sample dimension is $T^{\{N\} \times \{H,W,C\}}(\mathbf{Y}) \in \mathbb{R}^{N \times HWC}$. Then the regularizer formulate as follows:

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

Where μ is a matrix size $K \times HWC$ encoding all the centers with K the total number of clusters. $z_n \in [K]$ means which cluster the n-th sample belongs to.

Clearly, if the n-th sample belongs to a wrong cluster, the value of this regularizer becomes large.

Key insight: How to optimize

- In each layer you want to add this sample clustering regularization, you implement a smoothed k-means algorithm
- After you get fixed μ, you update weights by backpropogation.

Let $T^{\{N\}\times\{H,W,C\}}(\mathbf{Y}) = \mathbf{X}$. Then the gradient of regularizer equals to:

$$\frac{\partial \mathcal{R}}{\partial \mathbf{X}_n} = \frac{1}{NHWC}(\mathbf{X}_n - \mu_n) \tag{3}$$

Different from the paper.

Experiment Results

The result beats the state-of-art baselines in CIFAR 10 and CIFAR 100.

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	$\mathbf{81.05} \pm 0.41$	63.60 ± 0.55	$\mathbf{50.50} \pm 0.38$
Spatial-Clustering	90.50 ± 0.05	$\mathbf{81.02} \pm 0.12$	64.38 ± 0.38	$\mathbf{50.18} \pm 0.49$
Channel Co-Clustering	89.26 ± 0.25	$\mathbf{80.65} \pm 0.23$	63.42 ± 1.34	$\mathbf{49.80} \pm 0.25$

Summary

- This paper propose a regularized loss function for the deep neural nets, which enforce the clustering in the NN.
- Some prolems left:
    - Some experiment results don't achieve the state-of-art.
    - Certain equation in the paper is hard to understand.