CS 4803 / 7643: Deep Learning

Topics:

- Review of supervised learning
- Unsupervised Learning
- Unsupervised transfer learning

Yen-Chang Hsu

Robotics Ph.D. student

Georgia Tech

Outline

- Review of supervised learning
- Unsupervised Learning
- Unsupervised transfer learning

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification,
regression, object detection,
semantic segmentation, image
captioning, etc.

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

→ Cat

Classification

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

DOG, **DOG**, **CAT**

Object Detection

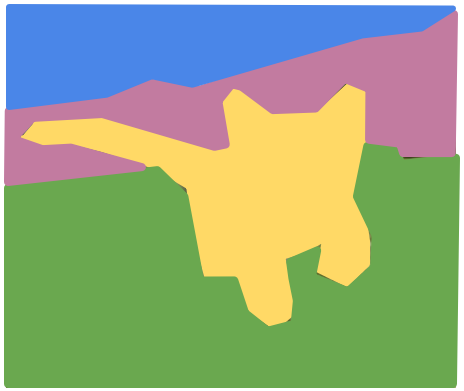
Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.



GRASS, **CAT**, **TREE**, **SKY**

Semantic Segmentation

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

A cat sitting on a suitcase on the floor

Image captioning

Caption generated using neuraltalk2
Image is CC0 Public domain.

Outline

- Review of supervised learning
- Unsupervised Learning
- Unsupervised transfer learning

Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

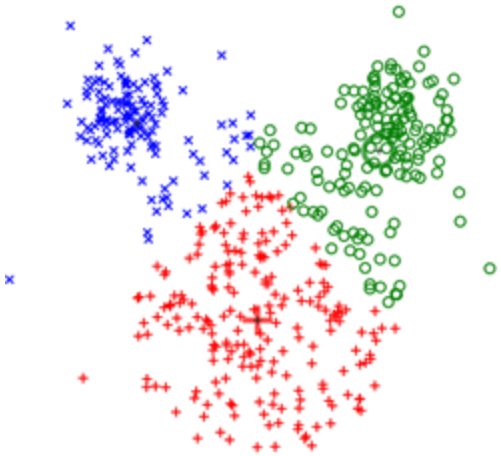
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



K-means clustering

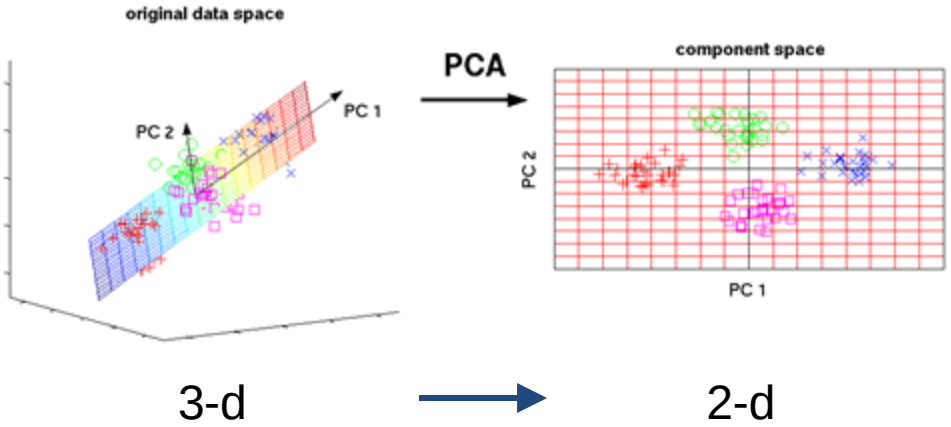
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Principal Component Analysis
(Dimensionality reduction)

Supervised vs Unsupervised Learning

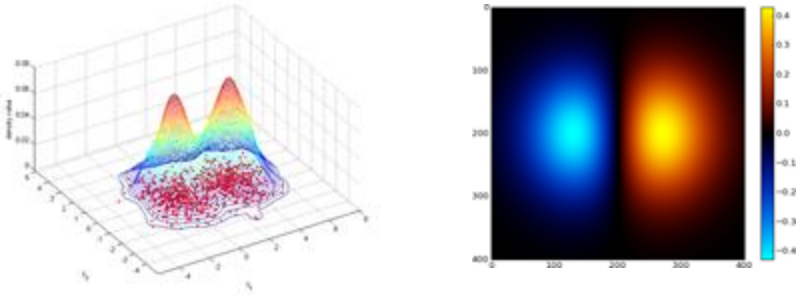
Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

1-d density estimation



2-d density estimation

Tasks

Supervised Learning

x	→	Classification	→ y	Discrete
x	→	Regression	→ y	Continuous

Unsupervised Learning

x	→	Clustering	→ c	Discrete
x	→	Dimensionality Reduction	→ z	Continuous
x	→	Density Estimation	→ p(x)	On simplex

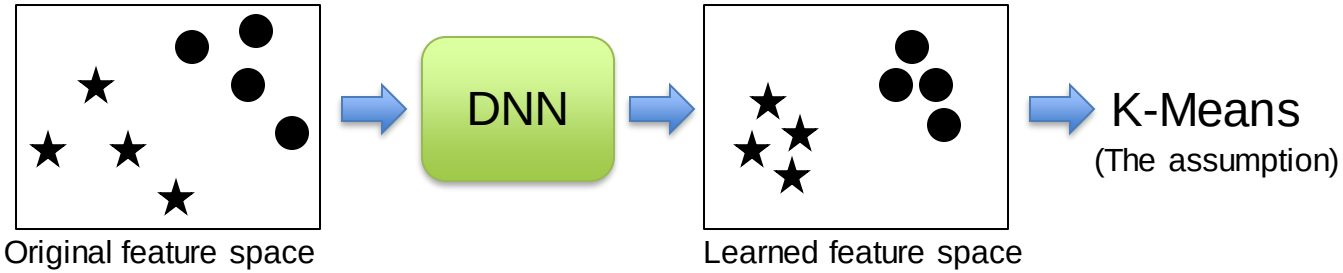
DNN and Unsupervised Learning

- DNN is a mapping function. How to train it without knowing the target?

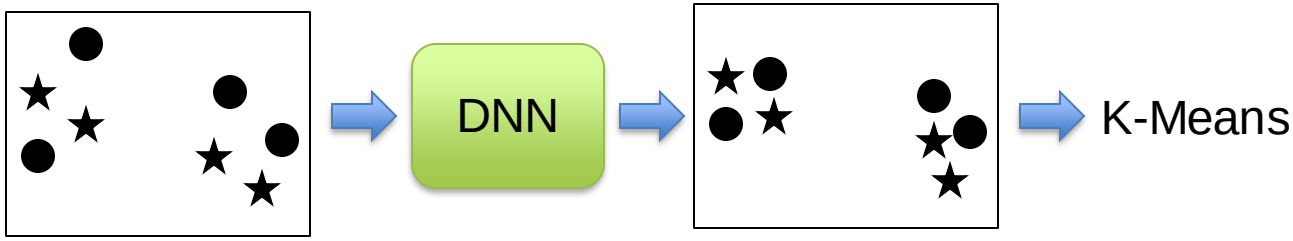
- Idea1
  - Create the objective function based on assumptions instead of labels.

Clustering assumption

- Clustering
  - Assumption
    - High density region forms a cluster while low density region separate clusters which hold a coherent semantic meaning.
  - We hope:

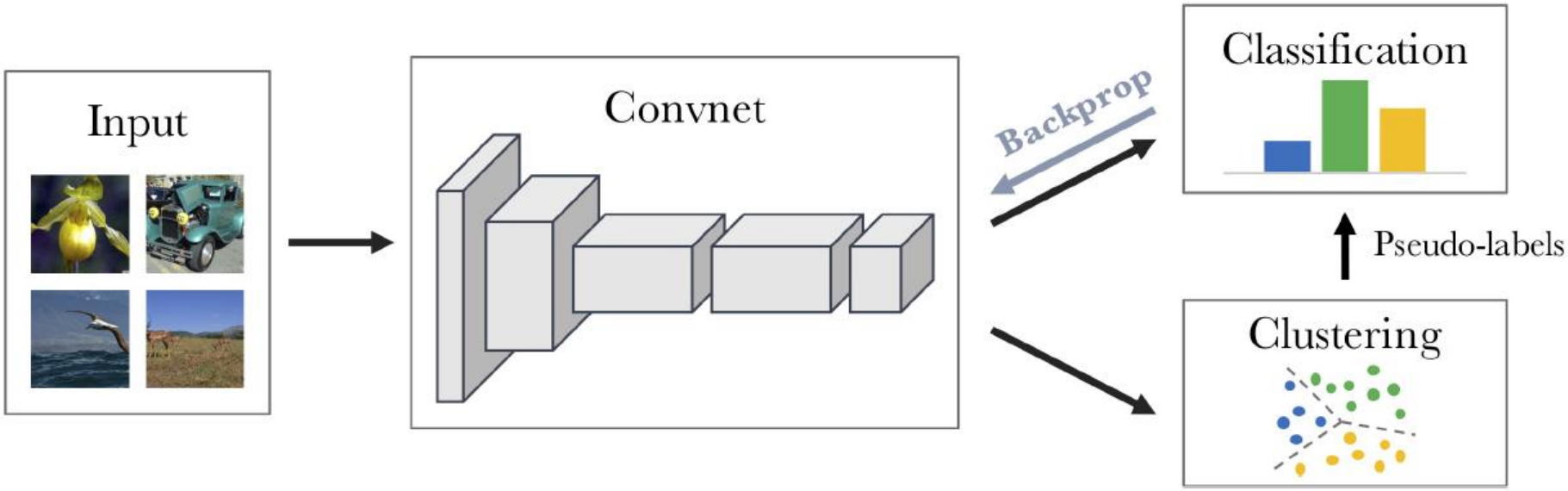


  - Sometimes, this could happen:



Create Pseudo targets

- The clustering assumption leads to good feature learning with a careful engineering to avoid
  - Trivial parameterization
  - Empty cluster



Deep Clustering for Unsupervised Learning of Visual Features

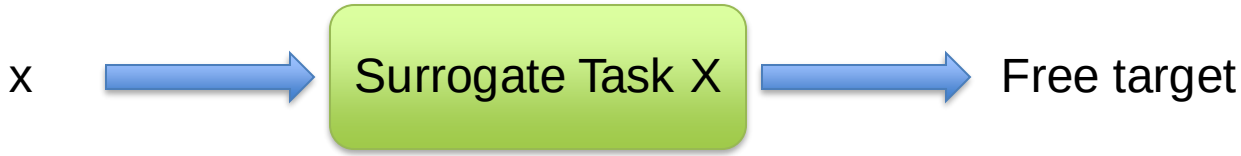
Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze

DNN and Unsupervised Learning

- DNN is a mapping function. How to train it without knowing the target?

- Idea 1
  - Use assumptions to create the objective function
- Idea 2
  - Design a proxy task which has free supervision
    - Self-supervised learning
    - Learning with a surrogate task

Lots of Surrogate Tasks!

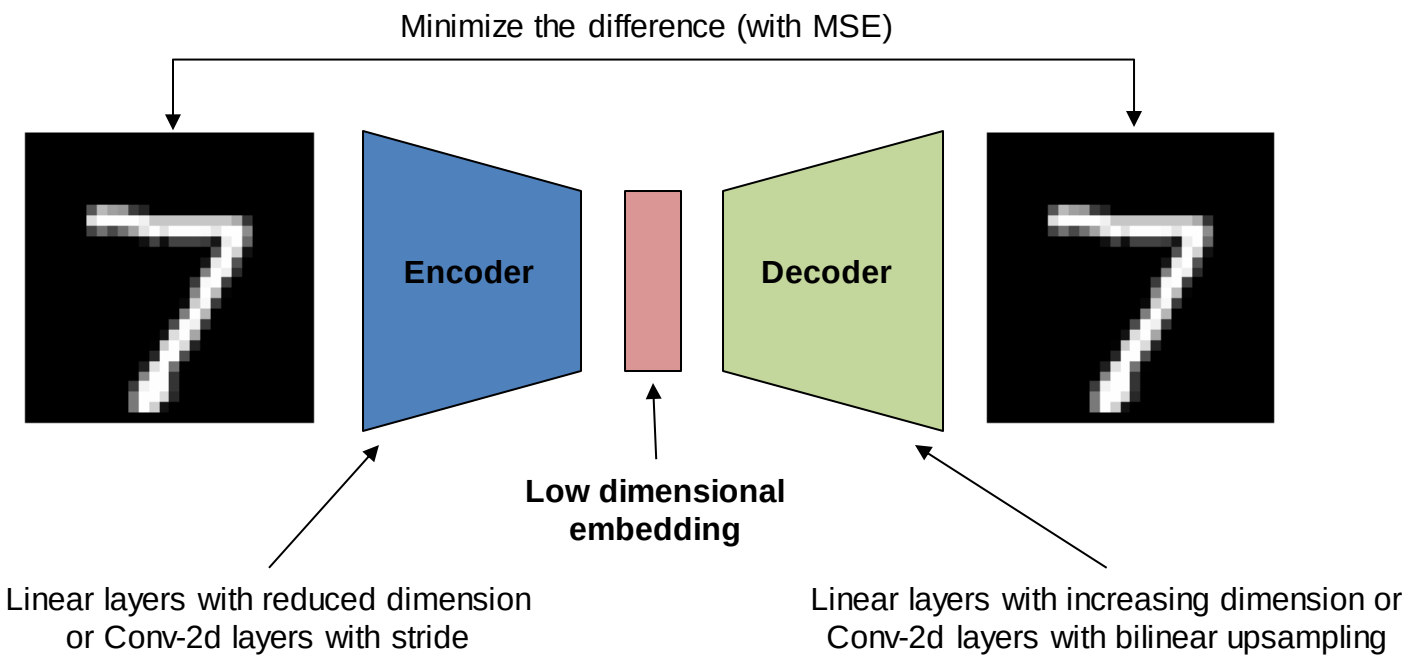


- Reconstruction
- Colorization
- Relative image patch location
- Rotate images, predict if image is rotated or not
- Video: Next frame prediction
- …

We talk about images mostly, but similar idea applies to other modalities as well (ex: Language)

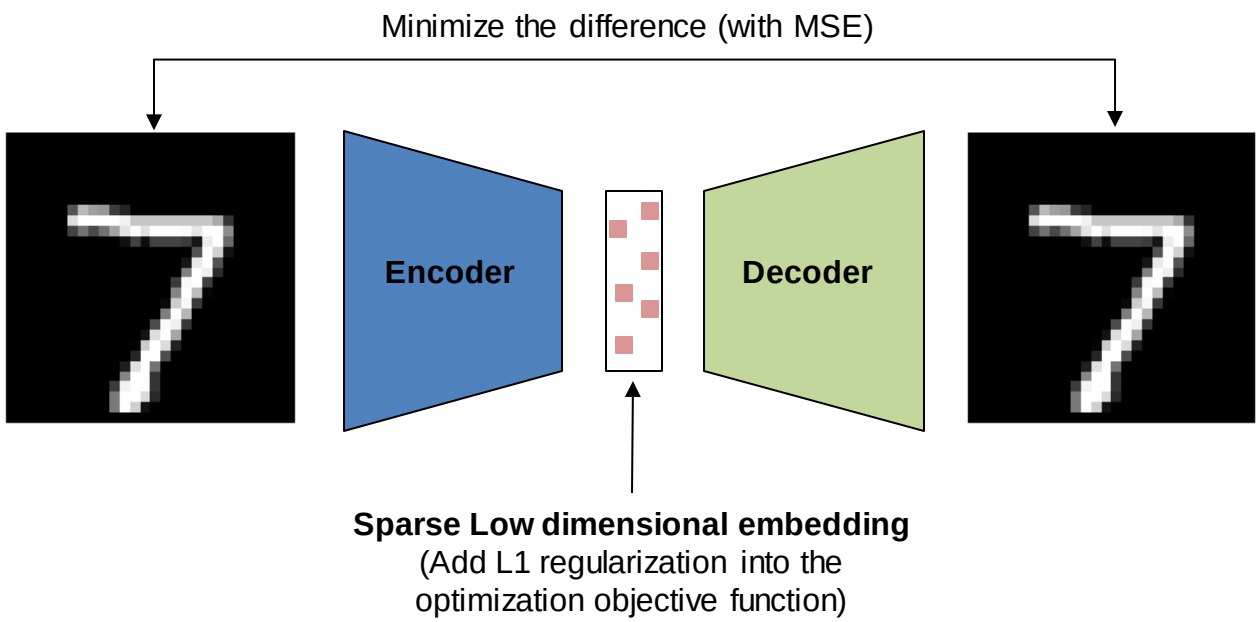
Reconstruction

- Autoencoder
  - The autoencoder performs the same feed-forward and backpropagation computations as the standard feed-forward Artificial Neural Network (ANN).



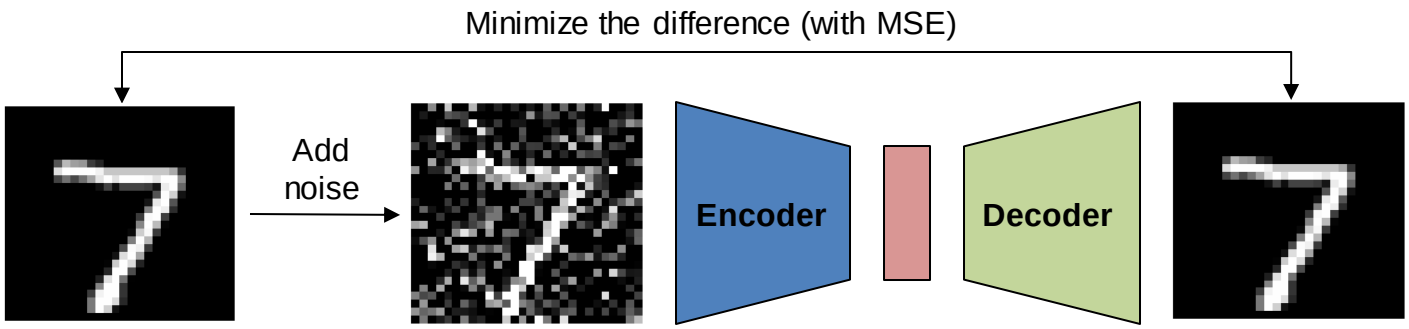
Reconstruction

- Autoencoder + additional assumptions
  - Sparse Autoencoder
    - Assume the embedding is sparse (few non-zero values)



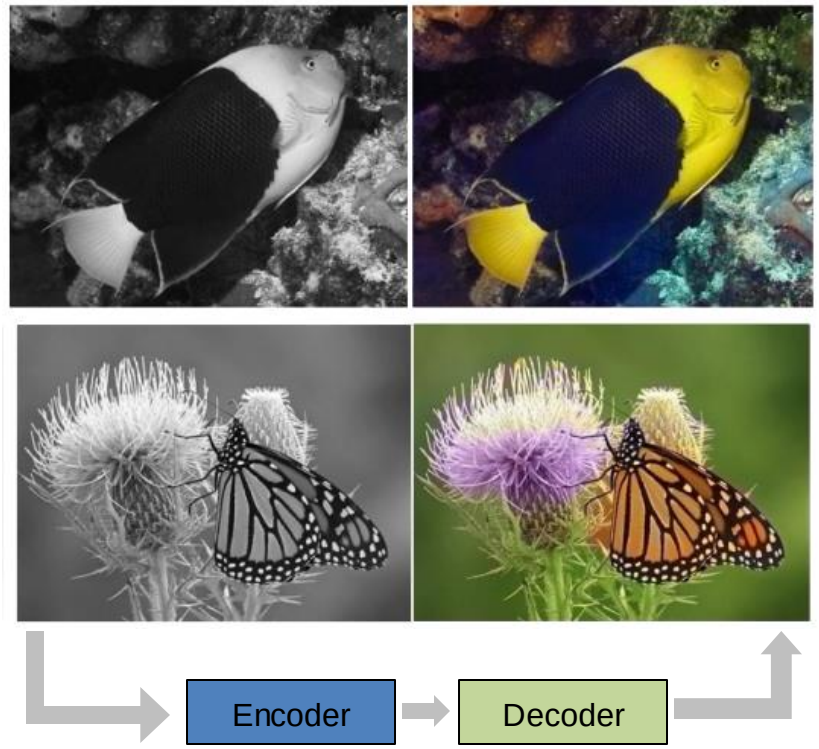
Reconstruction

- Autoencoder + additional assumptions
  - Denoising Autoencoder
    - Assume the embedding should not encode the noise.



Colorization

- Input: Grayscale image
- Output: Color image
- Objective function: MSE

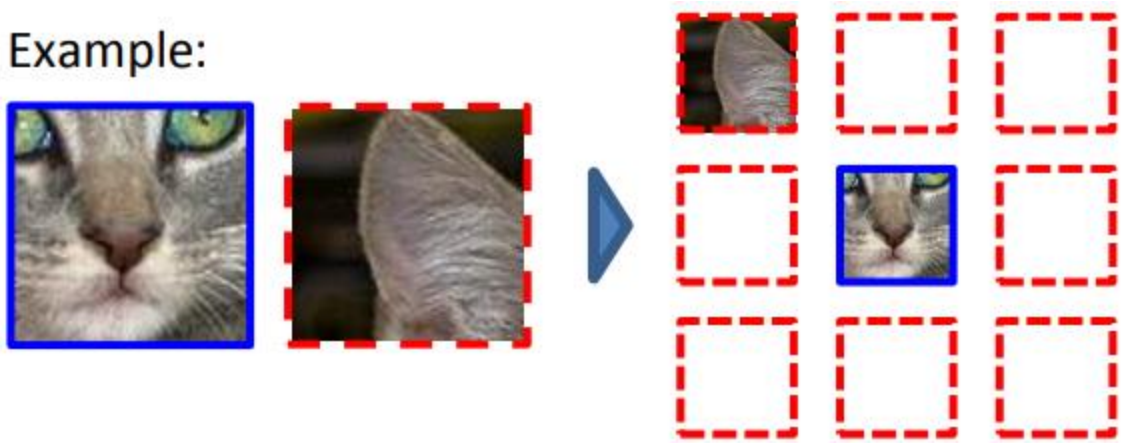


Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei A. Efros

Context Prediction

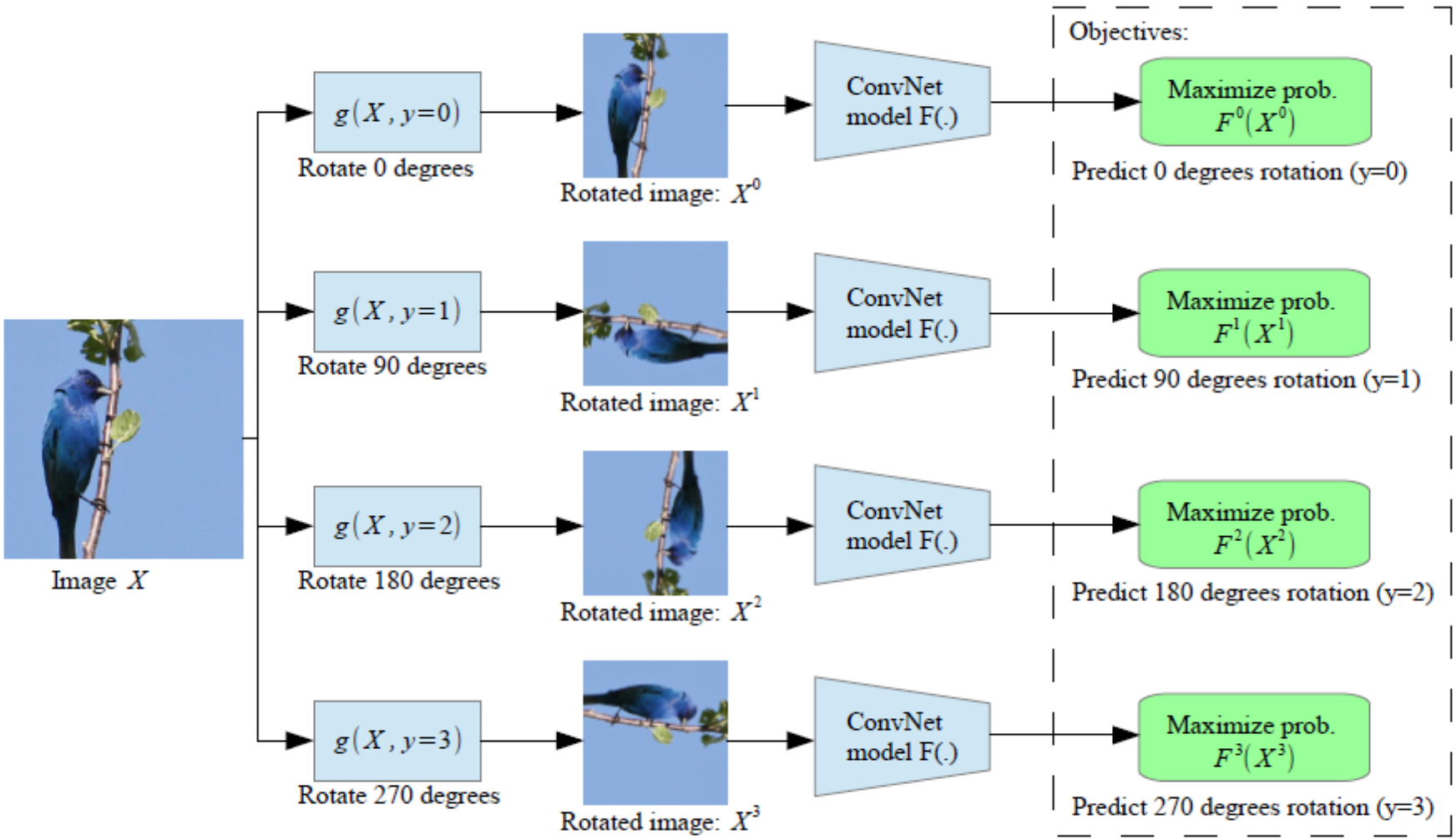
- Input: Two patches from the same image
- Output: One of the 8 locations
- Objective function: Cross-entropy (classification)



Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch, Abhinav Gupta, Alexei A. Efros

Predicting Rotations: RotNet



Unsupervised Representation Learning by Predicting Image Rotations

Spyros Gidaris, Praveer Singh, Nikos Komodakis

Evaluate unsupervised learning

- Steps
  1. Train the model with a surrogate task
  2. Extract the ConvNet (or encoder part)
  3. Transfer to the actual task
     - Use it to initialize the model of another supervised learning task
     - Use it to extract features for learning a separate classifier (ex: SVM)

RotNet: Results

Table 7: **Task & Dataset Generalization: PASCAL VOC 2007 classification and detection results, and PASCAL VOC 2012 segmentation results.** We used the publicly available testing frameworks of Krähenbühl et al. (2015) for classification, of Girshick (2015) for detection, and of Long et al. (2015) for segmentation. For classification, we either fix the features before conv5 (column *fc6-8*) or we fine-tune the whole model (column *all*). For detection we use multi-scale training and single scale testing. All approaches use AlexNet variants and were pre-trained on ImageNet without labels except the ImageNet labels and Random entries. After unsupervised training, we absorb the batch normalization units on the linear layers and we use the weight rescaling technique proposed by Krähenbühl et al. (2015) (which is common among the unsupervised methods). As customary, we report the mean average precision (mAP) on the classification and detection tasks, and the mean intersection over union (mIoU) on the segmentation task.

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

DNN and Unsupervised Learning

- DNN is a mapping function. How to train it without knowing the target?

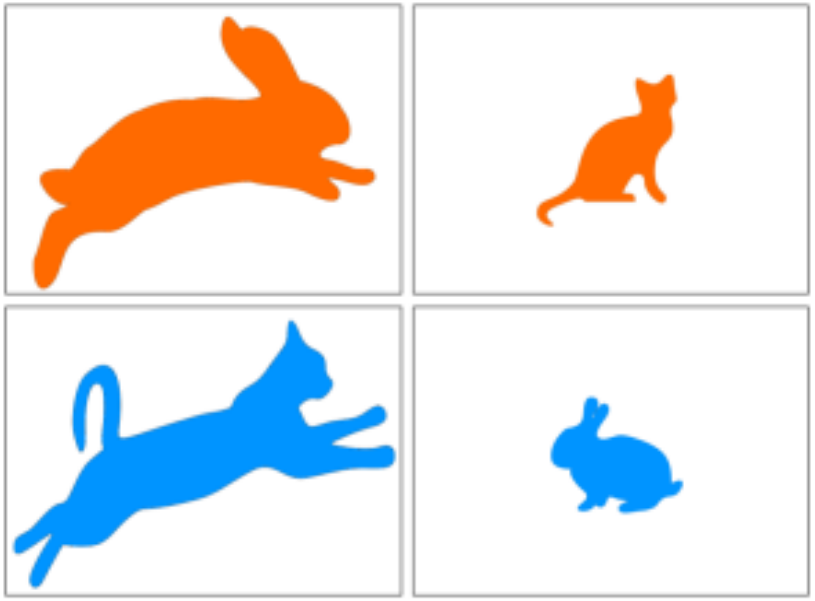
- Idea 1
  - Use assumptions to create the objective function
- Idea 2
  - Design a proxy task which has free supervision
    - Self-supervised learning
    - Learning with a surrogate task
- Idea 3
  - Transfer the supervision from labeled data to unlabeled data

Outline

- Review of supervised learning
- Unsupervised Learning
- Unsupervised transfer learning

Revisit Clustering

- Let's consider grouping the following four images into two clusters:

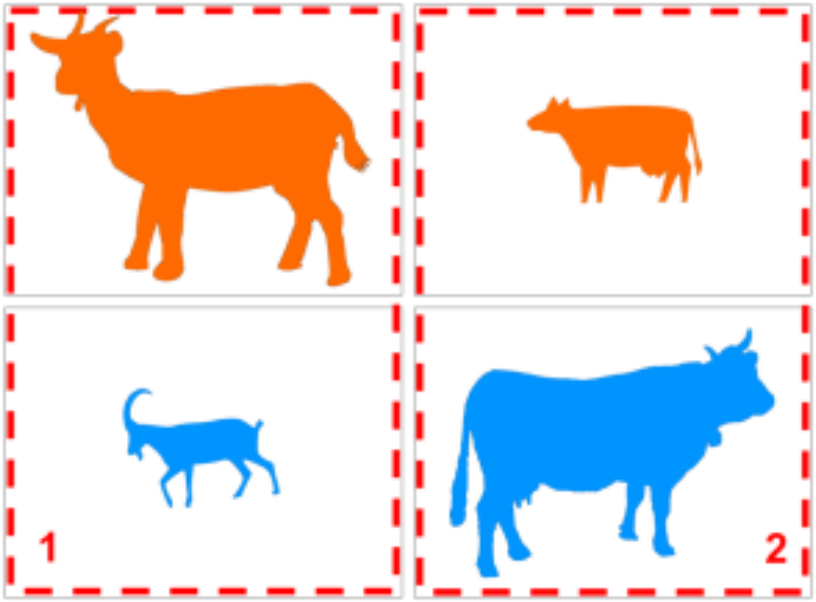


Hint: There are four possible criteria: color, pose, species, and size

(Image source: https://mlatgt.blog/2018/04/29/learning-to-cluster/)

Revisit Clustering

- We usually see examples like this:



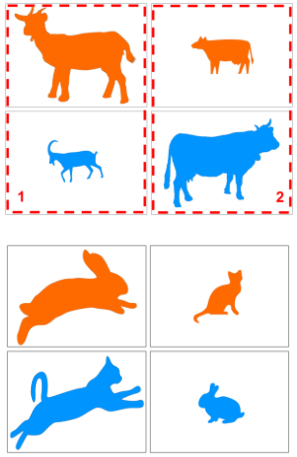
We learn the clustering criteria from examples.

Can machine do a similar thing?

(Image source: https://mlatgt.blog/2018/04/29/learning-to-cluster/)

The transfer learning problem

- Source
  - Labeled dataset
- Target
  - **Unlabeled** data
  - Unseen categories
- The problems



Source

Domain Shift

New Category

Different domain

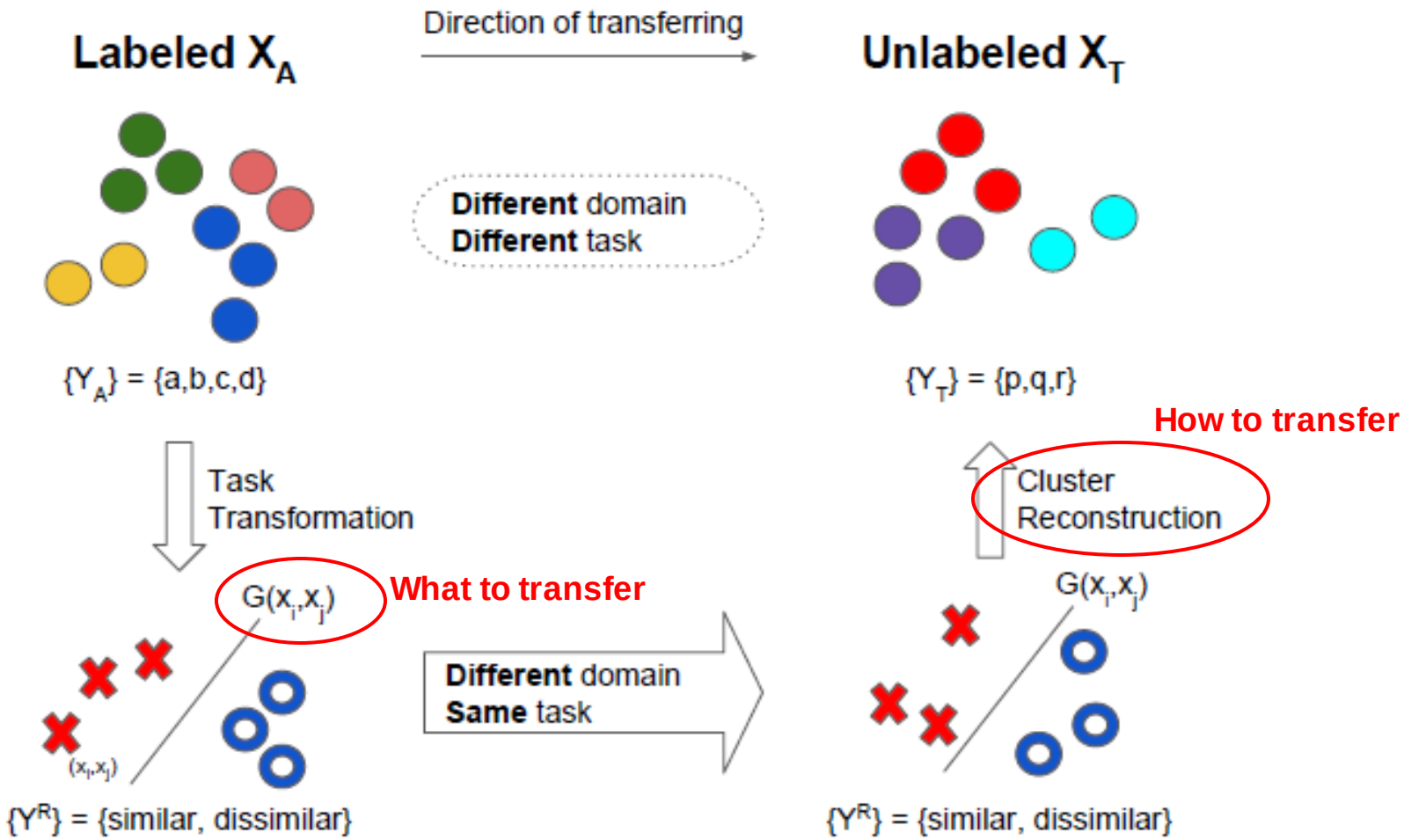
Different task

Target

The transfer learning problem

- Source
  - Labeled dataset
- Target
  - Unlabeled data
  - Unseen categories
- What to transfer
  - A similarity prediction function
    - If two input images are from the same class → Output "Similar"
    - Category-agnostic
- How to transfer
  - Deep clustering objective
    - With pairwise constraints → Constrained clustering

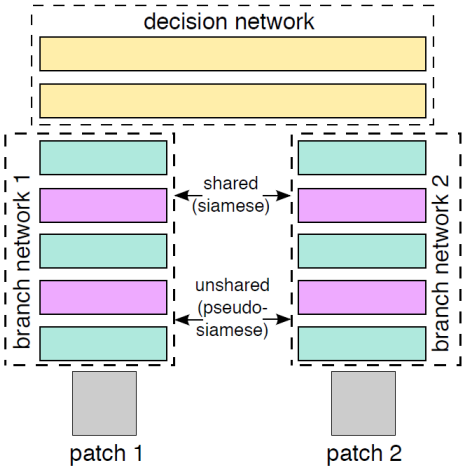
The transferring process



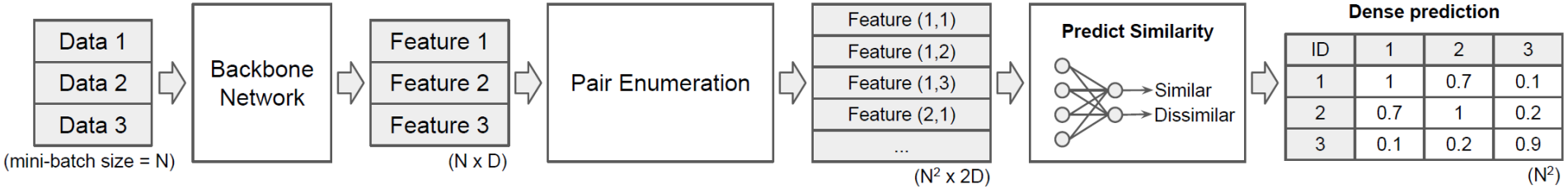
[Hsu et al., ICLR 2018]

Learning a Similarity Function

- Lots of work on use of *Siamese* networks to train a similarity metric
- We use a custom architecture that is similar but much more efficient using a *pair enumeration layer*
- Use cross-entropy to optimize rather than hinge loss

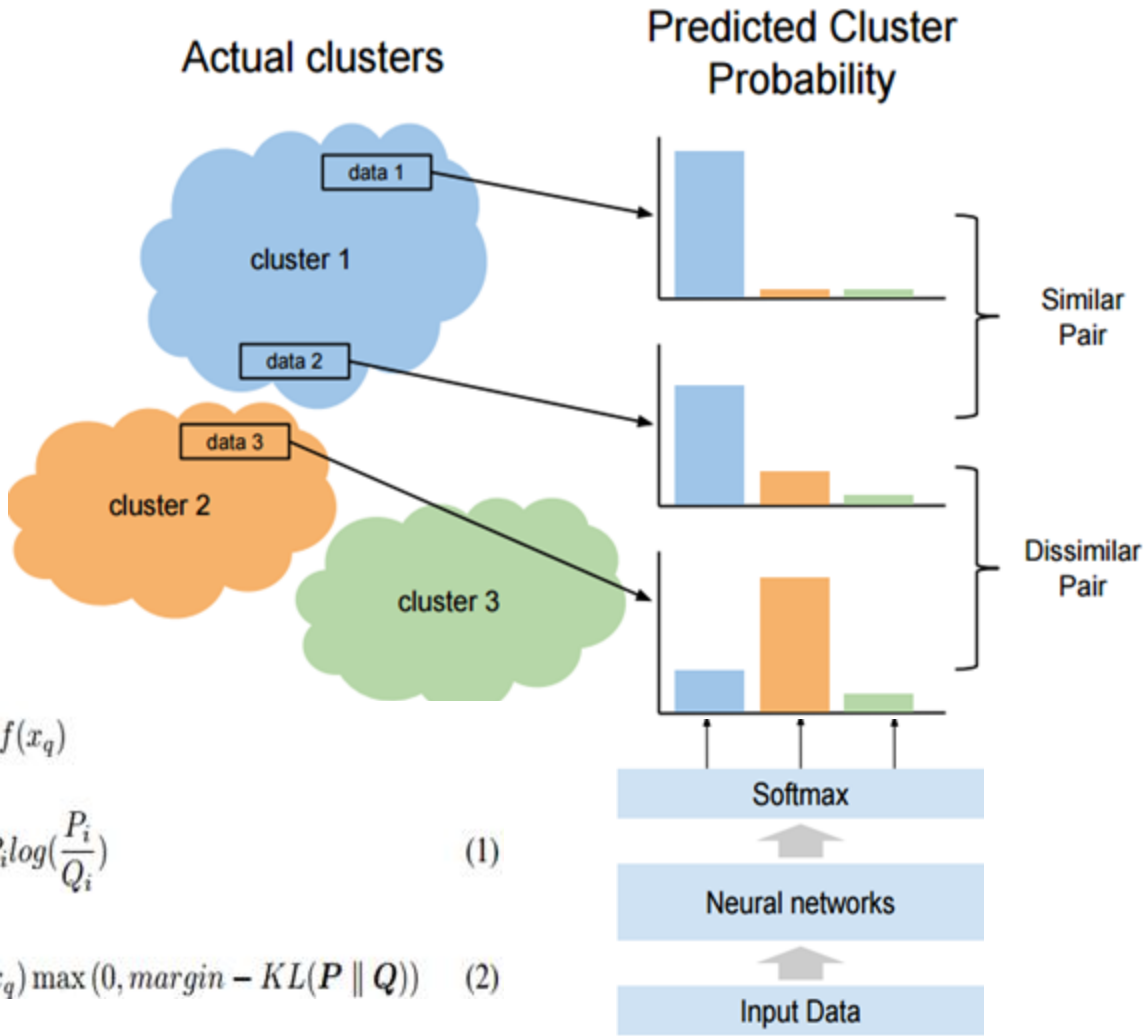


[Zagoruyko & Komodakis, CVPR 2015]



[Hsu et al., ICLR 2018]

The deep clustering objective



$$P = f(x_p), Q = f(x_q)$$

$$KL(P \parallel Q) = \sum_{i=1}^{k} P_i log(\frac{P_i}{Q_i}) \quad (1)$$

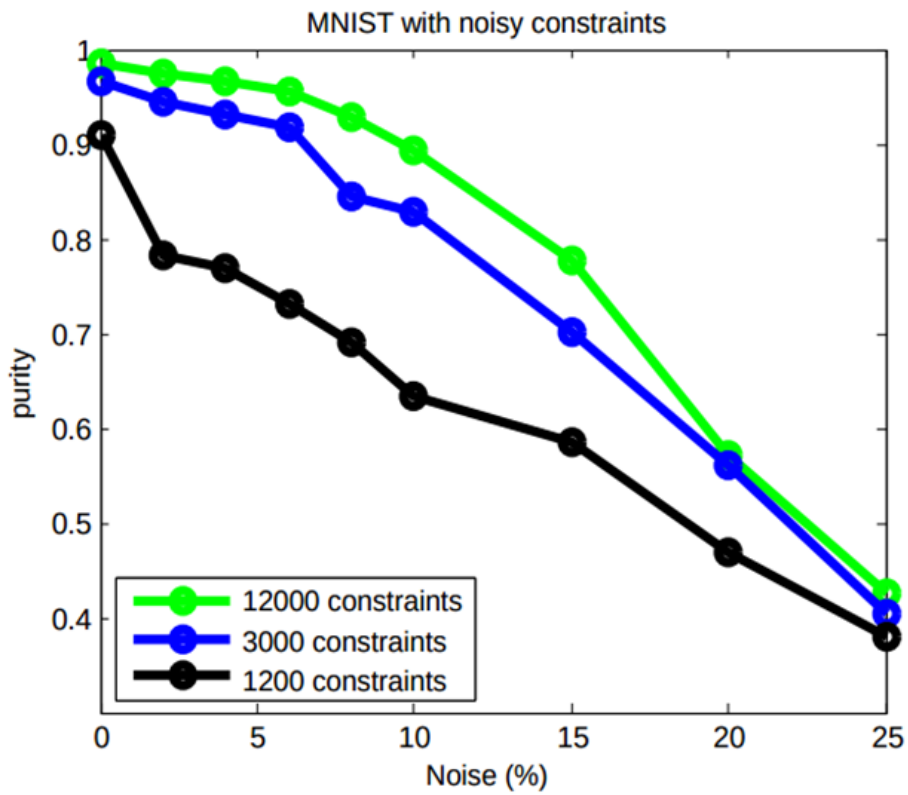
$$loss(P \parallel Q) = I_s(x_p, x_q) KL(P \parallel Q) + I_{ds}(x_p, x_q) \max(0, margin - KL(P \parallel Q)) \quad (2)$$

$$L(P, Q) = loss(P \parallel Q) + loss(Q \parallel P)$$

[Hsu et al., ICLR Workshop Track 2016]

The deep clustering objective

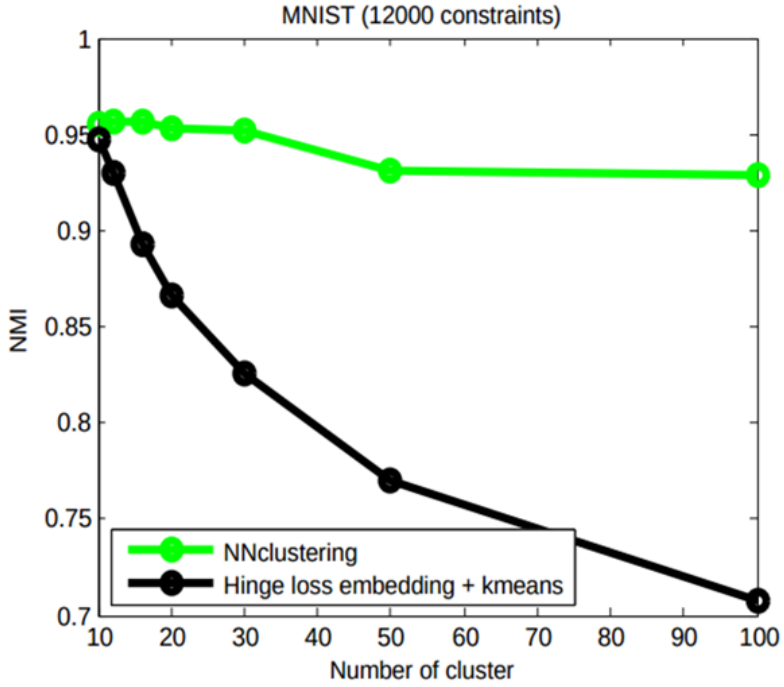
- Robust to noise
  - Increased robustness with number of constraints
  - It is a crucial property since our transferring scheme uses noisy pairwise constraints



[Hsu et al., ICLR Workshop Track 2016]

The deep clustering objective

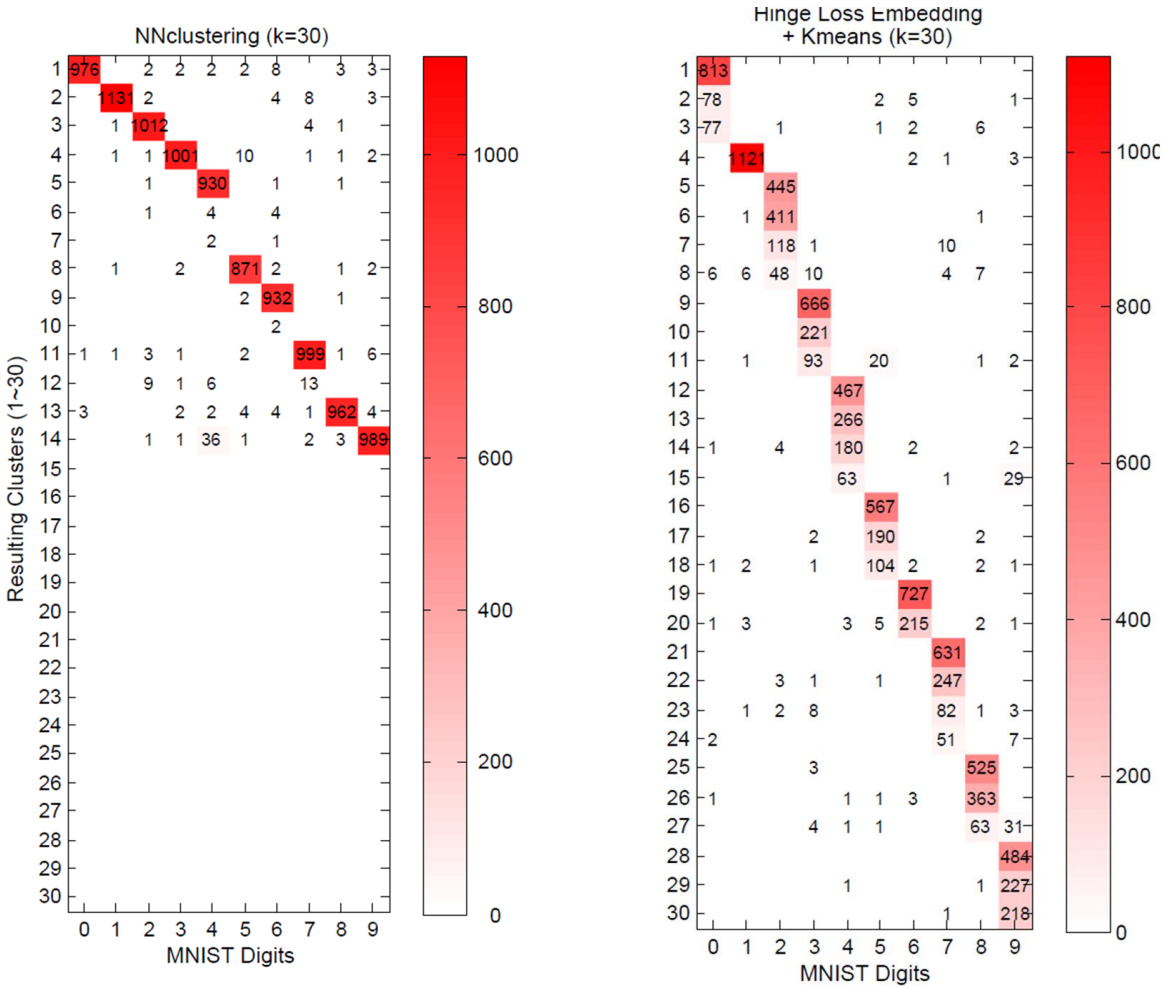
- Robust to an unknown number of cluster
  - Key difficulty for clustering methods is knowing how many clusters there are.
  - In our method, we give it enough number of cluster, then the optimization automatically leads to use only a suitable number of cluster.



[Hsu et al., ICLR Workshop Track 2016]

The deep clustering objective

- Robust to an unknown number of cluster



[Hsu et al., ICLR Workshop Track 2016]

The deep clustering objective

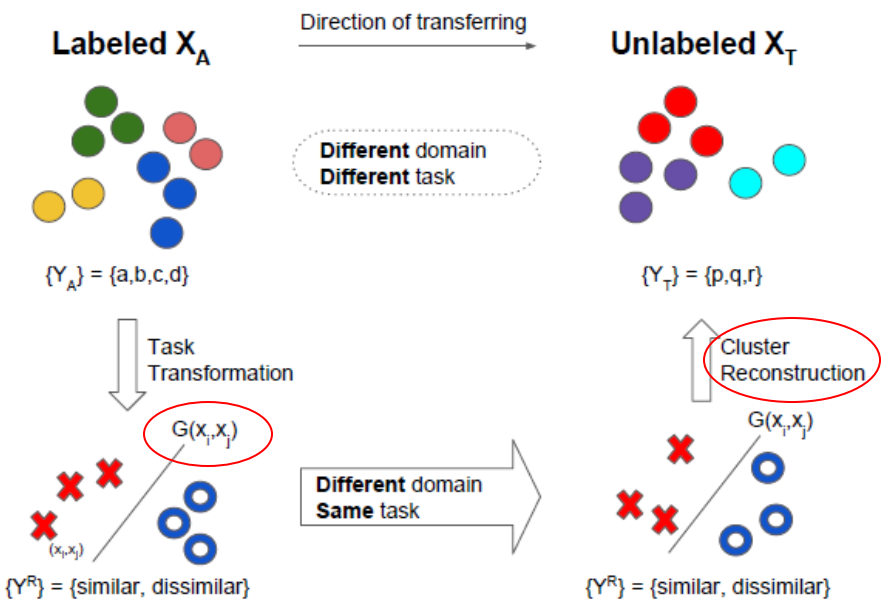
Advantages

- Multilayer neural networks ✔
- Joint feature learning and clustering ✔
- End-to-end training/testing ✔
- Robust to noise ✔
- Robust to #cluster ✔
- Good performance upper bound ✔

Putting it All Together

- Given a similarity metric trained on a dataset, we can *transfer* it to use in our clustering objective

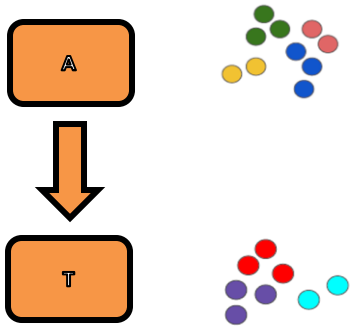
- Very flexible and can be used for multiple problem types:
  - Cross-task learning
  - Cross-domain (domain adaptation)

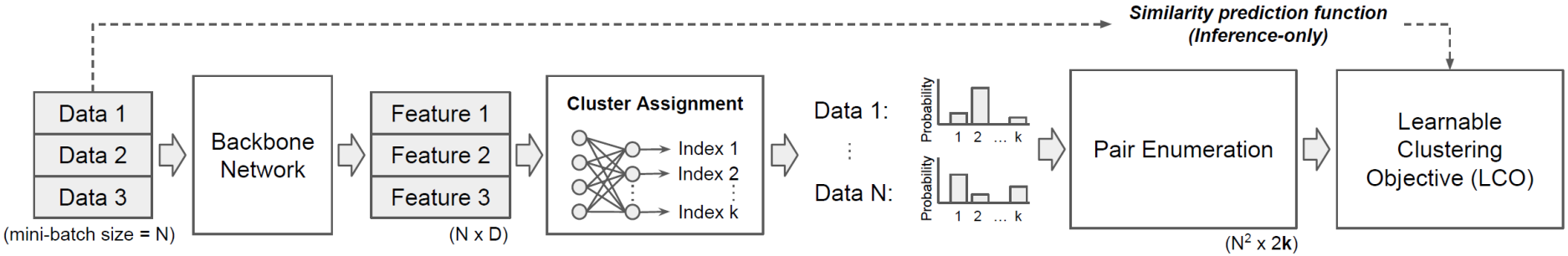


[Hsu et al., ICLR 2018]

Cross-Task Learning

- Auxiliary dataset (A) used for training of similarity, target (T) is to be clustered
- We use the similarity network outputs (inference only) on new unlabeled dataset
  - This is used for optimizing both *features* and *clustering* output on unlabeled data!
- Similarity metric will be noisy (since it is from a different domain)
  - Hence the need for our robust clustering method





[Hsu et al., ICLR 2018]

Results: Cross-Task Learning

- Use Omniglot, since there are many tasks (alphabets) with varying number of categories

Labeled

Unlabeled

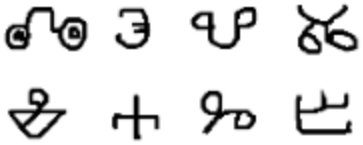


Table 1: Unsupervised cross-task transfer from $Omniglot_{bg}$ to $Omniglot_{eval}$. The performance is averaged across 20 alphabets which have 20 to 47 letters. The ACC and NMI without brackets have the number of clusters equal to ground-truth. The "(100)" means the algorithms use $K = 100$. The characteristics of how each algorithm utilizes the pairwise constraints are marked in the "Constraints in" column, where metric stands for the metric learning of feature representation.

Method	Constraints in Metric	Constraints in Clustering	ACC	ACC (100)	NMI	NMI (100)
K-means			21.7%	18.9%	0.353	0.464
LPNMF			22.2%	16.3%	0.372	0.498
LSC			23.6%	18.0%	0.376	0.500
ITML	o		56.7%	47.2%	0.674	0.727
SKMS	o		-	45.5%	-	0.693
SKKm	o		62.4%	46.9%	0.770	0.781
SKLR	o		66.9%	46.8%	0.791	0.760
CSP		o	62.5%	65.4%	0.812	0.812
MPCK-means	o	o	81.9%	53.9%	0.871	0.816
CCN (Ours)	o	o	**82.4%**	**78.1%**	**0.889**	**0.874**

[Hsu et al., ICLR 2018]

Randomized ImageNet Experiments

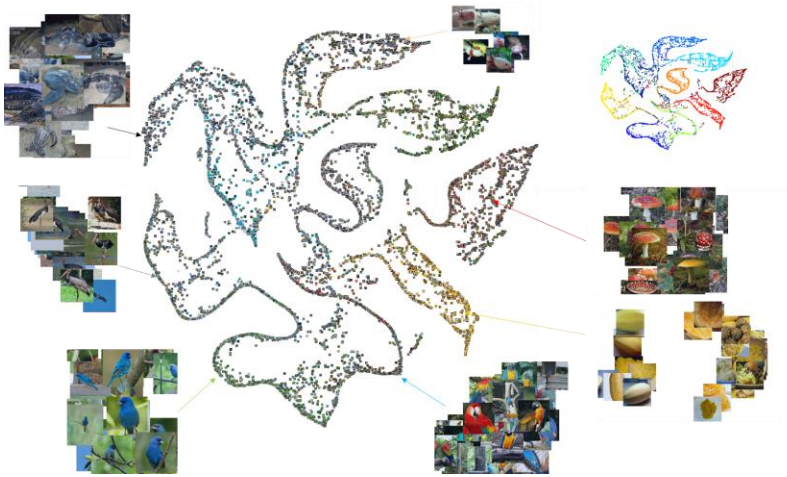
- Train similarity network on 882 classes
- Test on random subsets of 30 classes from remaining 118 classes
- Many clustering methods cannot scale to ImageNet

Table 6: Unsupervised cross-task transfer learning on ImageNet. The values are the average of three random subsets in $ImageNet_{118}$. Each subset has 30 classes. The "ACC" has $K = 30$ while the "ACC (100)" sets $K = 100$. All methods use the features (outputs of average pooling) from Resnet-18 pre-trained with $ImageNet_{882}$ classification.

Method	ACC	ACC(100)	NMI	NMI(100)
K-means	71.9%	34.5%	0.713	0.671
LSC	73.3%	33.5%	0.733	0.655
LPNMF	43.0%	21.8%	0.526	0.500
CCN (ours)	**73.8%**	**65.2%**	**0.750**	**0.715**

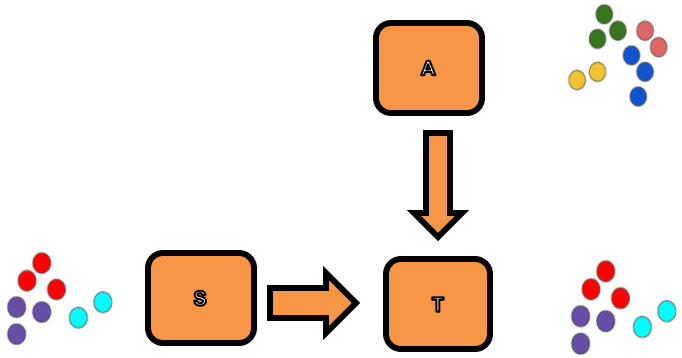
[Hsu et al., ICLR 2018]

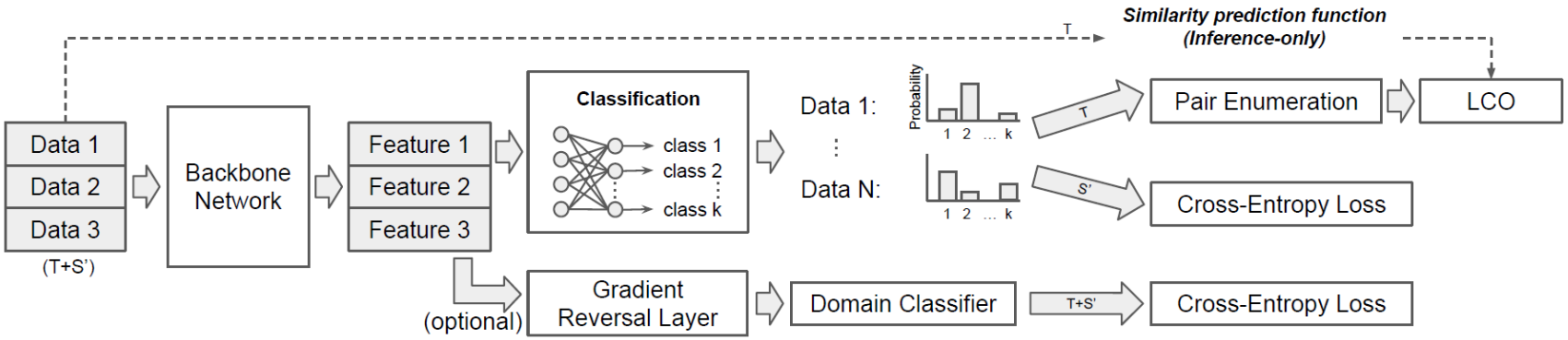
[Hsu et al., Arxiv 2016]



Domain Adaptation

- Our approach can apply to domain adaptation as well!
  - Same object categories across domains
  - Target is unlabeled
- Can use multiple loss functions, add domain discrepancy loss as well





[Hsu et al., ICLR 2018]

Results: Domain Adaptation on Office-31

- For larger image datasets, we use ImageNet as auxiliary dataset for similarity learning
- In LCO condition, LCO and cross-entropy loss (for source) is used
  - In LCO+DANN the domain adaptation loss is used as well
- **Transfer of similarity function is somewhat orthogonal to explicit domain adaptation**

[https://people.eecs.berkeley.edu/~jhoffman/domainadapt]

Table 2: Unsupervised cross-domain transfer (domain adaptation) on the Office-31 dataset. The backbone network used here is Resnet-18 (He et al., 2016) pre-trained with ImageNet.

	A → W	D → W	W → D	A → D	D → A	W → A	Avg
Source-Only	66.8	92.8	96.8	67.1	51.4	53.0	71.3
DANN (Ganin et al., 2016)	73.2	97.0	99.0	69.3	58.0	57.8	75.7
JAN (Long et al., 2017)	74.5	94.1	**99.6**	**75.9**	58.7	59.0	76.9
LCO (ours)	76.7	97.3	98.2	71.2	61.0	60.5	77.5
LCO+DANN	**78.2**	**97.4**	98.6	73.5	**62.8**	**60.6**	**78.5**

A: ImageNet

S': Left of arrow

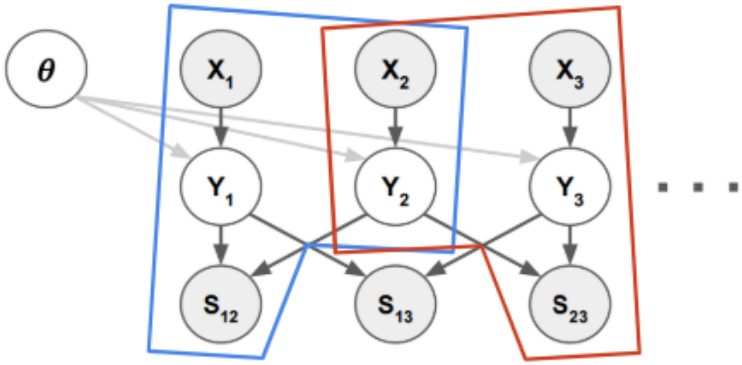
T: Right of arrow

Some Limitations

- Similarity transfer is hard
- Especially true when similarity function is trained on smaller datasets
- The strategy applies to only categorization tasks

Enhanced Results

- Even better constrained clustering loss based on novel probabilistic graphical formulation
- Directly optimize clustering likelihood



$$\mathcal{L}(\theta; \mathbf{X}, \mathbf{Y}, \mathbf{S}) = \mathrm{P}(\mathbf{X}, \mathbf{Y}, \mathbf{S}; \theta) = \mathrm{P}(\mathbf{S}|\mathbf{Y})\mathrm{P}(\mathbf{Y}|\mathbf{X}; \theta)\mathrm{P}(\mathbf{X})$$

$$\begin{aligned}\mathcal{L}(\theta; \mathbf{X}, \mathbf{S}) &\approx \sum_{\mathbf{Y}} \mathrm{P}(\mathbf{S}|\mathbf{Y})\mathrm{P}(\mathbf{Y}|\mathbf{X}; \theta) \\ &\approx \prod_{i,j} \Big(\sum_{Y_i = Y_j} \mathbb{1}[s_{ij} = 1]\mathrm{P}(Y_i|x_i; \theta)\mathrm{P}(Y_j|x_j; \theta) + \\ &\qquad \sum_{Y_i \neq Y_j} \mathbb{1}[s_{ij} = 0]\mathrm{P}(Y_i|x_i; \theta)\mathrm{P}(Y_j|x_j; \theta) \Big).\end{aligned}$$

ImageNet Cross-Task Transfer Learning

Method	ACC	ACC(100)	NMI	NMI(100)
K-means	71.9%	34.5%	0.713	0.671
LSC	73.3%	33.5%	0.733	0.655
LPNMF	43.0%	21.8%	0.526	0.500
CCN-KCL	73.8%	65.2%	0.750	0.715
CCN-CCL	**74.4%**	**71.5%**	**0.762**	**0.765**

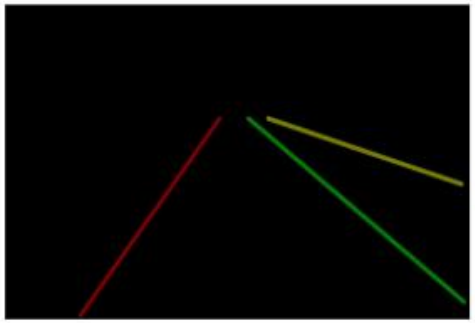
[Hsu et al., ICLR 2019]

Extension

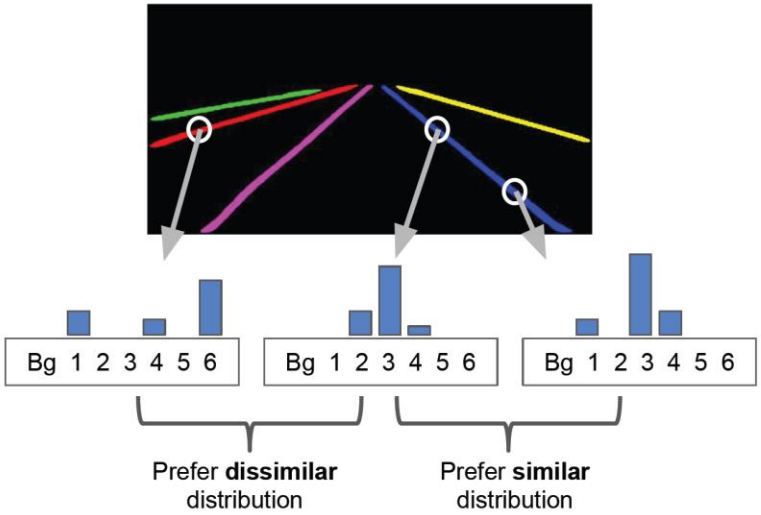
- Apply the deep clustering loss to learn pixel-level clustering in a single forward
  - Won CVPR 2017 Lane detection challenge 2nd place.

Each image is a clustering problem

Learned clustering function
(Neural networks)



Desired clustering result



[Hsu et al., IJCNN 2018]

Video

Summary

- Unsupervised learning with DNN
  - Use assumptions to create the objective function
  - Design a proxy task which has free supervision
    - Self-supervised learning
    - Learning with a surrogate task
  - Transfer the supervision from labeled data to unlabeled data