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 \(x \rightarrow y\)

**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 $x \rightarrow y$

**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 \( x \rightarrow y \)

**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 \(x \rightarrow y\)

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

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
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)

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
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.

Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-d density estimation

2-d density estimation

2-d density images left and right are CC0 public domain.

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Tasks

Supervised Learning

\[ x \xrightarrow{\text{Classification}} y \text{ Discrete} \]

\[ x \xrightarrow{\text{Regression}} y \text{ Continuous} \]

Unsupervised Learning

\[ x \xrightarrow{\text{Clustering}} c \text{ Discrete} \]

\[ x \xrightarrow{\text{Dimensionality Reduction}} z \text{ Continuous} \]

\[ x \xrightarrow{\text{Density Estimation}} p(x) \text{ On simplex} \]
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:

![Original feature space](image1)
![Learned feature space](image2)

- Sometimes, this could happen:

![Original feature space](image3)
![Learned feature space](image4)
Create Pseudo targets

• The clustering assumption leads to good feature learning with a careful engineering to avoid
  – Trivial parameterization
  – Empty cluster
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).

![Diagram of autoencoder]
Reconstruction

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

Minimize the difference (with MSE)

Sparse Low dimensional embedding
(Add L1 regularization into the optimization objective function)
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

(C) Dhruv Batra & Zsolt Kira
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.

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

(C) Dhruv Batra & Zsolt Kira
Outline

• Review of supervised learning
• Unsupervised Learning
• Unsupervised transfer learning
Revisit Clustering

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

![Images of animals](https://mlatgt.blog/2018/04/29/learning-to-cluster/)

**Hint:** There are four possible criteria: color, pose, species, and size.
Revisit Clustering

• We usually see examples like this:

We learn the clustering criteria from examples.
Can machine do a similar thing?
The transfer learning problem

- **Source**
  - Labeled dataset
- **Target**
  - *Unlabeled* data
  - Unseen categories
- **The problems**
  - Domain Shift
    - Different domain
  - New Category
    - 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

Labeled $X_A$

{\{Y_A\} = \{a,b,c,d\}}

Direction of transferring

Unlabeled $X_T$

{\{Y_T\} = \{p,q,r\}}

Task Transformation

$G(x_i,x_j)$

{\{Y^R\} = \{similar, dissimilar\}}

How to transfer

Cluster Reconstruction

What to transfer

Different domain

Different task

Same task

Different domain

(C) Dhruv Batra & Zsolt Kira

[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

[Hsu et al., ICLR 2018]
[Zagoruyko & Komodakis, CVPR 2015]
The deep clustering objective

\[ P = f(x_p), Q = f(x_q) \]

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

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

\[ L(P, Q) = loss(P \parallel Q) + loss(Q \parallel P) \]
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

![MNIST with noisy constraints graph](image)
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.
The deep clustering objective

- Robust to an unknown number of cluster
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

[Diagram of data processing steps]

[Hsu et al., ICLR 2018]
Results: Cross-Task Learning

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

![Labeled and Unlabeled examples](image)

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.

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

[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.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
<th>ACC(100)</th>
<th>NMI</th>
<th>NMI(100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>71.9%</td>
<td>34.5%</td>
<td>0.713</td>
<td>0.671</td>
</tr>
<tr>
<td>LSC</td>
<td>73.3%</td>
<td>33.5%</td>
<td>0.733</td>
<td>0.655</td>
</tr>
<tr>
<td>LPNMF</td>
<td>43.0%</td>
<td>21.8%</td>
<td>0.526</td>
<td>0.500</td>
</tr>
<tr>
<td>CCN (ours)</td>
<td><strong>73.8%</strong></td>
<td><strong>65.2%</strong></td>
<td><strong>0.750</strong></td>
<td><strong>0.715</strong></td>
</tr>
</tbody>
</table>

[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.**

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.

<table>
<thead>
<tr>
<th></th>
<th>A $\rightarrow$ W</th>
<th>D $\rightarrow$ W</th>
<th>W $\rightarrow$ D</th>
<th>A $\rightarrow$ D</th>
<th>D $\rightarrow$ A</th>
<th>W $\rightarrow$ A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-Only</td>
<td>66.8</td>
<td>92.8</td>
<td>96.8</td>
<td>67.1</td>
<td>51.4</td>
<td>53.0</td>
<td>71.3</td>
</tr>
<tr>
<td>DANN (Ganin et al., 2016)</td>
<td>73.2</td>
<td>97.0</td>
<td>99.0</td>
<td>69.3</td>
<td>58.0</td>
<td>57.8</td>
<td>75.7</td>
</tr>
<tr>
<td>JAN (Long et al., 2017)</td>
<td>74.5</td>
<td>94.1</td>
<td>99.6</td>
<td>75.9</td>
<td>58.7</td>
<td>59.0</td>
<td>76.9</td>
</tr>
<tr>
<td>LCO (ours)</td>
<td>76.7</td>
<td>97.3</td>
<td>98.2</td>
<td>71.2</td>
<td>61.0</td>
<td>60.5</td>
<td>77.5</td>
</tr>
<tr>
<td>LCO+DANN</td>
<td><strong>78.2</strong></td>
<td><strong>97.4</strong></td>
<td><strong>98.6</strong></td>
<td><strong>73.5</strong></td>
<td><strong>62.8</strong></td>
<td><strong>60.6</strong></td>
<td><strong>78.5</strong></td>
</tr>
</tbody>
</table>

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

\[
\mathcal{L}(\theta; X, Y, S) = P(X, Y, S; \theta) = P(S|Y)P(Y|X; \theta)P(X)
\]

\[
\mathcal{L}(\theta; X, S) \approx \sum_Y P(S|Y)P(Y|X; \theta)
\]

\[
\approx \prod_{i,j} \left( \sum_{Y_i=Y_j} 1[s_{ij}=1]P(Y_i|x_i; \theta)P(Y_j|x_j; \theta) + \sum_{Y_i\neq Y_j} 1[s_{ij}=0]P(Y_i|x_i; \theta)P(Y_j|x_j; \theta) \right).
\]

### ImageNet Cross-Task Transfer Learning

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
<th>ACC(100)</th>
<th>NMI</th>
<th>NMI(100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>71.9%</td>
<td>34.5%</td>
<td>0.713</td>
<td>0.671</td>
</tr>
<tr>
<td>LSC</td>
<td>73.3%</td>
<td>33.5%</td>
<td>0.733</td>
<td>0.655</td>
</tr>
<tr>
<td>LPNMF</td>
<td>43.0%</td>
<td>21.8%</td>
<td>0.526</td>
<td>0.500</td>
</tr>
<tr>
<td>CCN-KCL</td>
<td>73.8%</td>
<td>65.2%</td>
<td>0.750</td>
<td>0.715</td>
</tr>
<tr>
<td><strong>CCN-CCL</strong></td>
<td><strong>74.4%</strong></td>
<td><strong>71.5%</strong></td>
<td><strong>0.762</strong></td>
<td><strong>0.765</strong></td>
</tr>
</tbody>
</table>
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

Desired clustering result

 Learned clustering function (Neural networks)
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