Visual Representation Learning

CS 6384 Computer Vision
Professor Yu Xiang
The University of Texas at Dallas
Learning Visual Representations

Neural Network → Feature representation → Classification, Clustering, Segmentation, Detection, Image captioning, Etc.
Generative Models

• Autoencoder

• Variational Autoencoder (VAE)
Generative Models

• Bi-directional GAN

ADVERSARIAL FEATURE LEARNING. Donahue et al., ICLR, 2017
Discriminative Models (Supervised Learning)

Input image → Convolutional layer → ReLU layer → Pooling layer → Fully connected layer → Output vector → Classification

Feature representation
Supervised Representation Learning

• Train neural networks for image classification
• Use internal features in the network as feature representations
• Applications

Image retrieval

Supervised Representation Learning

Image clustering


Supervised Representation Learning

• Training with classification loss functions
  • E.g., cross-entropy loss

• Can we have better loss functions for representation learning?

• Deep metric learning
  • Learning distance metrics with neural networks
Distance metrics

- L1 distance
  \[ D(x, y) = \sum_{i=1}^{N} |x_i - y_i| \]

- L2 distance
  \[ D(x, y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \]

- Cosine distance
  \[ D(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|} \]

Cosine similarity
Deep Metric Learning

\[ D(x_1, x_2) = D(f(x_1), f(x_2)) \]

L2 distance
\[ D(x_1, x_2) = \| f(x_1) - f(x_2) \|_2 \]

Learning the distance metric is equivalent to learning the feature representation.
Contrastive Loss

- Use positive pairs and negative pairs

\[ x_1 \xrightarrow{\text{Neural Network}} f(x_1) \]

Positive pair \( f(x_1) f(x_2) \) should be close

\[ D(x_1, x_2) \text{ small} \]

\[ x_2 \xrightarrow{\text{Neural Network}} f(x_2) \]

Negative pair \( f(x_1) f(x_2) \) should be far

\[ D(x_1, x_2) \text{ large} \]

Learning a Similarity Metric Discriminatively, with Application to Face Verification. Chopra et al., CVPR, 2005.
Contrastive Loss

- Training data \( \{(x_i, x_j, y_{ij})\} \)

\[ y_{ij} = \begin{cases} 1 & \text{if positive pair} \\ 0 & \text{if negative pair} \end{cases} \]

\[ J = \frac{1}{m} \sum_{(i,j)}^{m/2} y_{i,j} D_{i,j}^2 + (1 - y_{i,j}) [\alpha - D_{i,j}]^2_+ \]

- \( m \): number of images in a batch

\([x]_+ = \max(0, x)\)

Learning a Similarity Metric Discriminatively, with Application to Face Verification. Chopra et al., CVPR, 2005.
Contrastive Loss

\[ J = \frac{1}{m} \sum_{(i,j)}^{m/2} y_{i,j} D_{i,j}^2 + (1 - y_{i,j}) [\alpha - D_{i,j}]^2_+ \]

\[ \frac{\partial J}{\partial D_{i,j}} = \frac{2}{m} \left( y_{i,j} D_{i,j} - (1 - y_{i,j})[\alpha - D_{i,j}]_+ \right) \]

\[ D_{i,j} = \| f(x_i) - f(x_j) \|_2 \]

\[ \frac{\partial D_{i,j}}{\partial f(x_i)} = \frac{f(x_i) - f(x_j)}{\| f(x_i) - f(x_j) \|} \]

\[ \mathbf{X}_i \rightarrow \text{Neural Network} \rightarrow f(x_i) \]

\[ \frac{\partial J}{\partial f(x_i)} \]
Triplet Loss

• Use a triplet (anchor, positive, negative)

\[ J = \frac{3}{2m} \sum_{i}^{m/3} [D_{ia,ip}^2 - D_{ia,in}^2 + \alpha]_+ \]

\[ D_{ia,ip} = \| f(x_i^a) - f(x_i^p) \| \quad \quad D_{ia,in} = \| f(x_i^a) - f(x_i^n) \| \]

Lifted Structured Loss

• Consider all positive pairs and negative pairs in a mini-batch

\[
J = \frac{1}{2|\hat{P}|} \sum_{(i,j) \in \hat{P}} \max(0, J_{i,j})^2
\]

\[
J_{i,j} = \max \left( \max_{(i,k) \in \hat{N}} \alpha - D_{i,k}, \max_{(j,l) \in \hat{N}} \alpha - D_{j,l} \right) + D_{i,j}
\]

Hard negative
Distance for the negative pair
Distance for the positive pair

Relaxed loss
\[
\tilde{J}_{i,j} = \log \left( \sum_{(i,k) \in \hat{N}} \exp\{\alpha - D_{i,k}\} + \sum_{(j,l) \in \hat{N}} \exp\{\alpha - D_{j,l}\} \right) + D_{i,j}
\]

Multi-class N-pair Loss

- Use a positive pair and N-1 negative ones and

\[
\mathcal{L}_{N\text{-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log \left( 1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^T f(\mathbf{x}_i^-) - f(\mathbf{x})^T f(\mathbf{x}^+)) \right)
\]

- Softmax for multi-class classification

InfoNCE (Noise Contrastive Estimation) Loss

• Similar to multi-class N-pair Loss

\[ \mathcal{L}_q = - \log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)} \]

Query q

Positive k+ (K+1)-way softmax classification

Negatives ki Motivated from identifying targets from noisy data
Supervised Representation Learning

• Use class labels to specify positive pairs and negative pairs

• Loss functions
  • Contrastive loss
  • Triplet loss
  • Lifted structured loss
  • N-pair loss
  • InfoNCE

• Consider more relationships in a mini-batch is better
Unsupervised/Self-supervised Representation Learning

• Pretext tasks
  • Tasks designed for feature learning
  • Not the final tasks

• Positive pairs from different views of the same image

Learning Representations by Maximizing Mutual Information Across Views. Bachman et al., NeurIPS, 2019
Unsupervised/Self-supervised Representation Learning

- Pretext task: context prediction

Unsupervised/Self-supervised Representation Learning

• Pretext task: rotation prediction
Unsupervised/Self-supervised Representation Learning

• Pretext task: colorization

Unsupervised/Self-supervised Representation Learning

• Pretext task: inpainting

Unsupervised/Self-supervised Representation Learning

• Pretext task: clustering

Deep Clustering for Unsupervised Learning of Visual Features. Caron et al., ECCV, 2018
SimCLR

• A simple framework for contrastive learning of visual representations

\[
\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}
\]

SimCLR

- Transformations

SimCLR

• After training, keep the encoder network $h_i = f(\tilde{x}_i) = \text{ResNet}(\tilde{x}_i)$

• Linear evaluation protocol for classification
  • A linear classifier is trained on top of the frozen base network

**SimCLR**


2x, 4x: more channels in ResNet

![ImageNet top-1 accuracy chart](image)

<table>
<thead>
<tr>
<th>1st transformation</th>
<th>Crop</th>
<th>Cutout</th>
<th>Color</th>
<th>Sobel</th>
<th>Noise</th>
<th>Blur</th>
<th>Rotate</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crop</td>
<td>33.1</td>
<td>33.9</td>
<td>56.3</td>
<td>46.0</td>
<td>39.9</td>
<td>35.0</td>
<td>30.2</td>
<td>39.2</td>
</tr>
<tr>
<td>Cutout</td>
<td>32.2</td>
<td>25.6</td>
<td>33.9</td>
<td>40.0</td>
<td>26.5</td>
<td>25.2</td>
<td>22.4</td>
<td>29.4</td>
</tr>
<tr>
<td>Color</td>
<td>55.8</td>
<td>35.5</td>
<td>18.8</td>
<td>21.0</td>
<td>11.4</td>
<td>16.5</td>
<td>20.8</td>
<td>25.7</td>
</tr>
<tr>
<td>Sobel</td>
<td>46.2</td>
<td>40.6</td>
<td>20.9</td>
<td>4.0</td>
<td>9.3</td>
<td>6.2</td>
<td>4.2</td>
<td>18.8</td>
</tr>
<tr>
<td>Noise</td>
<td>38.8</td>
<td>25.8</td>
<td>7.5</td>
<td>7.6</td>
<td>9.8</td>
<td>9.8</td>
<td>9.6</td>
<td>15.5</td>
</tr>
<tr>
<td>Blur</td>
<td>35.1</td>
<td>25.2</td>
<td>16.6</td>
<td>5.8</td>
<td>9.7</td>
<td>2.6</td>
<td>6.7</td>
<td>14.5</td>
</tr>
<tr>
<td>Rotate</td>
<td>30.0</td>
<td>22.5</td>
<td>20.7</td>
<td>4.3</td>
<td>9.7</td>
<td>6.5</td>
<td>2.6</td>
<td>13.8</td>
</tr>
</tbody>
</table>

![2nd transformation chart](image)
SimCLR

https://github.com/google-research/simclr
Summary: Visual Representation Learning

• Generative models
  • Autoencoder
  • VAE
  • GAN

• Discriminative models
  • Supervised learning
    • Training with image classification
    • Deep metric learning
  • Unsupervised/self-supervised learning
    • Use pretext tasks
    • Metric learning loss functions
Further Reading

• Improved Deep Metric Learning with Multi-class N-pair Loss Objective, 2016 [https://papers.nips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf](https://papers.nips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf)