Cooperative Learning of Energy-based Model and Latent Variable Model via MCMC

Teaching

Ying Nian Wu

Department of Statistics
University of California, Los Angeles

UC Irvine, Oct 25, 2018
A Tale of Two Nets

Bottom-up ConvNet
- **energy**
- signal
  - (a) Descriptor Net
- Energy-based Model

Top-down ConvNet
- **latent variables**
- signal
  - (b) Generator Net
- Latent Variable Model
Plan

1 **Background**: Supervised and unsupervised learning; convolutional neural networks

2 **Energy-based Model**: Descriptor network (Xie*, Lu*, Zhu, Wu, ICML, 2016)

3 **Latent Variable Model**: Generator network (Han*, Lu*, Zhu, Wu, AAAI, 2017)

4 **Cooperative Learning**: CoopNets (Xie, Lu, Gao, Wu, AAAI, 2018; Xie, Lu, Gao, Zhu, Wu, TPAMI, accepted.)
Modes of learning

- Supervised learning: classification and regression
- Reinforcement learning: policy and value networks based on rewards
- **Unsupervised learning:**
  - **Generative models** and density estimation
    - **Latent variable models**, factor analysis
    - **Energy-based models**, exponential family models
  - Embedding and auto-encoding
Deep learning

- Convolutional neural network (ConvNet or CNN)
- Recurrent neural network (RNN)
- Models with multi-layer latent variables
Supervised learning

<table>
<thead>
<tr>
<th>obs</th>
<th>input</th>
<th>hidden</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$X_1^\top$</td>
<td>$h_1^\top$</td>
<td>$y_1$</td>
</tr>
<tr>
<td>2</td>
<td>$X_2^\top$</td>
<td>$h_2^\top$</td>
<td>$y_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$n$</td>
<td>$X_n^\top$</td>
<td>$h_n^\top$</td>
<td>$y_n$</td>
</tr>
</tbody>
</table>
Supervised learning

output: \( Y_i \)
hidden: \( h_i \)
input: \( X_i \)
Generative Models
Ying Nian Wu
Background
Energy-based model
Latent variable model
Cooperative learning
Divergence triangle

ConvNet

Input layer  Hidden layer  Output layer

Input 1
Input 2
Input 3
Input 4

generative
top-down deconvolutional image synthesis
discriminative
bottom-up convolutional feature extraction

input RGB image
feature maps  sub-sampled  feature maps
1st layer  2nd layer

Class 1
Class 2
Class C
Filtering
Generative Models

Ying Nian Wu

Background

Energy-based model

Latent variable model

Cooperative learning

Divergence triangle

ConvNet

[Le Cun et al. 1998; Krizhevsky et al. 2012]
Coordinate-wise non-linearity

$h(r) = \max(r, 0)$

Figure: Rectified Linear Unit (ReLU).
$L$-layer network

$$X \rightarrow h^{(1)} \rightarrow ... h^{(l-1)} \rightarrow h^{(l)} \rightarrow ... \rightarrow h^{(L)} \rightarrow \hat{Y},$$
Generative Models
Ying Nian Wu
Background
Energy-based model
Latent variable model
Cooperative learning
Divergence triangle

ConvNet

\[ h^{(l)} = f_l(W_l h^{(l-1)} + b_l), \]

where \( l = 1, \ldots, L \), \( h^{(0)} = X \), \( h^{(L+1)} = \hat{Y} \), and \( \theta = (W_l, b_l, l = 1, \ldots, L + 1) \). \( f_l \) is element-wise non-linearity.
Back-propagation

\[ \frac{\partial h^{(l)}}{\partial h^{(l-1)}} = f'_l(W_l h^{(l-1)} + b_l)W_l \]

End-to-end training
ConvNet

\[ h^{(l)} = f_l(W_l h^{(l-1)} + b_l), \]

Encompasses the following:

- Generalized linear model (GLM), e.g., logistic regression
- Linear spline: \( \sum_k \beta_k \max(0, x - b_k) \)
- CART/MARS: recursive partitioning, hinge functions
Unsupervised learning

<table>
<thead>
<tr>
<th>obs</th>
<th>input</th>
<th>hidden</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$X_1^\top$</td>
<td>$h_1^\top$</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>$X_2^\top$</td>
<td>$h_2^\top$</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>$X_n^\top$</td>
<td>$h_n^\top$</td>
<td>?</td>
</tr>
</tbody>
</table>

Diagram: Unsupervised learning model with input layer, hidden layer, and output layer.
Recall supervised learning

output: \( Y_i \)

hidden: \( h_i \)

input: \( X_i \)
Unsupervised learning

Energy-based model

\[
\begin{align*}
\text{energy} & : f(X_i; \theta) \\
\uparrow & \\
\text{features} & : h_i \\
\uparrow & \\
\text{input} & : X_i
\end{align*}
\]

Latent variable model

\[
\begin{align*}
\text{prior} & : p(h) \\
\downarrow & \\
\text{hidden} & : h_i \\
\downarrow & \\
\text{input} & : X_i
\end{align*}
\]
Unsupervised learning

Auto-encoder: encoder and decoder

\[
\text{code : } h_i \\
\uparrow \downarrow \\
\text{input : } X_i
\]

Embedding: relative relationship

\[
\leftarrow h_i \rightarrow \\
\leftarrow X_i \rightarrow
\]

Multi-dimensional scaling
Local linear embedding [Roweis and Saul 2000]
ConvNet

**Bottom-up ConvNet**
- **energy**
  - \( f(X; \theta) \)

**Top-down ConvNet**
- **hidden variables** \( h \)
  - \( X = g(h; \alpha) \)
Energy-based model: descriptor net

[Xie et al. 2016]

\[ X \rightarrow h^{(1)} \rightarrow ... \rightarrow h^{(L)} \rightarrow f(X; \theta) \]

\[ p(X; \theta) = \frac{1}{Z(\theta)} \exp \left[ f(X; \theta) \right] p_0(X). \]

\( p_0(X) \) is the reference distribution such as Gaussian white noise

\[ p_0(X) = \frac{1}{(2\pi s^2)^D/2} \exp \left[ -\frac{\|X\|^2}{2s^2} \right] \]

Can be derived from discriminative ConvNet

Energy function:

\[ E(X; \theta) = \|X\|^2/2s^2 - f(X; \theta) \]
Relationship with discriminative net

[Xie et al. 2016]

\[ X \rightarrow h^{(1)} \rightarrow \ldots \rightarrow h^{(L)} \rightarrow f(X; \theta_k) \]

\[ p(X; \theta_k) = \frac{1}{Z(\theta_k)} \exp [f(X; \theta_k)] p_0(X). \]

\[ \Pr(k|X) = \frac{\exp(f(X; \theta_k) + b_k)}{\sum_{k=0}^{K} \exp(f(X; \theta_k) + b_k)}, \] (1)
Maximum likelihood

\[ X_i \sim P_{\text{data}}, i = 1, ..., n. \]

\[ p(X; \theta) = \frac{1}{Z(\theta)} \exp \left[ -\mathcal{E}(X; \theta) \right] \]

Log-likelihood:

\[ L_p(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p(X_i; \theta) \]

minimize \( KL(P_{\text{data}}|p_{\theta}) \)

Maximum likelihood learning:

\[ -L'_p(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \mathcal{E}(X_i; \theta) - \mathbb{E}_{\theta} \left[ \frac{\partial}{\partial \theta} \mathcal{E}(X; \theta) \right] \]
Energy-based model: descriptor net

Langevin revision:

\[ X_{\tau+1} = X_{\tau} - \frac{\delta^2}{2} \frac{\partial}{\partial X} \mathcal{E}(X_{\tau}; \theta) + N(0, \delta^2 I_D) \]

Density shifting:

\[ -L'_p(\theta) \approx \frac{\partial}{\partial \theta} \left[ \frac{1}{n} \sum_{i=1}^{n} \mathcal{E}(X_i; \theta) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \mathcal{E}(\tilde{X}_i; \theta) \right] \]

Adversarial interpretation:

\[ V = \frac{1}{n} \sum_{i=1}^{n} \mathcal{E}(X_i; \theta) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \mathcal{E}(\tilde{X}_i; \theta) \]
Back-propagation

\[ \frac{\partial f(X; \theta)}{\partial \theta} \text{ for updating } \theta \]
\[ \frac{\partial f(X; \theta)}{\partial X} \text{ for sampling } X \]

The two derivatives share the same chain rule:

\[ \frac{\partial h^{(l)}}{\partial h^{(l-1)}} = f'_l (W_l h^{(l-1)} + b_l) W_l \]
Analysis by synthesis

D1: Dreaming
D2: Make the dreaming more realistic
Texture
Texture
Object

[Lu et al. 2016]
Multi-grid

Multi-grid

<table>
<thead>
<tr>
<th></th>
<th>Real images</th>
<th>DCGAN</th>
<th>Multi-grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception score</td>
<td>11.237</td>
<td>6.581</td>
<td>6.565</td>
</tr>
</tbody>
</table>
Multi-grid

Figure: Learning the multi-grid models from the LSUN bedroom dataset. Left: random samples of training examples. Right: synthesized examples generated by the learned models.
Learning features
Learning features

**Table:** Classification error of L2-SVM trained on the features learned from SVHN.

<table>
<thead>
<tr>
<th>Test error rate with # of labeled images</th>
<th>1,000</th>
<th>2,000</th>
<th>4,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent CD</td>
<td>45.74</td>
<td>39.47</td>
<td>34.18</td>
</tr>
<tr>
<td>One-step CD</td>
<td>44.38</td>
<td>35.87</td>
<td>30.45</td>
</tr>
<tr>
<td>Wasserstein GAN</td>
<td>43.15</td>
<td>38.00</td>
<td>32.56</td>
</tr>
<tr>
<td>Deep directed generative models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCGAN</td>
<td>44.99</td>
<td>34.26</td>
<td>27.44</td>
</tr>
<tr>
<td>Single-grid CD</td>
<td>38.59</td>
<td>32.51</td>
<td>29.37</td>
</tr>
<tr>
<td>Multi-grid CD</td>
<td>30.23</td>
<td>26.54</td>
<td>22.83</td>
</tr>
</tbody>
</table>
**Table:** Classification error of CNN classifier trained on the features of three grids learned from SVHN.

<table>
<thead>
<tr>
<th>Test error rate with # of labeled images</th>
<th>1,000</th>
<th>2,000</th>
<th>4,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGN</td>
<td>36.02</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Virtual adversarial</td>
<td>24.63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Auxiliary deep generative model</td>
<td>22.86</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Supervised CNN with the same structure</td>
<td>39.04</td>
<td>22.26</td>
<td>15.24</td>
</tr>
<tr>
<td>Multi-grid CD + CNN classifier</td>
<td><strong>19.73</strong></td>
<td><strong>15.86</strong></td>
<td><strong>12.71</strong></td>
</tr>
</tbody>
</table>
Learning prior

Generative Models
Ying Nian Wu

Background
Energy-based model
Latent variable model
Cooperative learning
Divergence triangle
## Learning prior

<table>
<thead>
<tr>
<th></th>
<th>Mask</th>
<th>PCD</th>
<th>CD1</th>
<th>SCD</th>
<th>CE</th>
<th>MCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mask</td>
<td>0.056</td>
<td>0.081</td>
<td>0.066</td>
<td>0.045</td>
<td>0.042</td>
<td></td>
</tr>
<tr>
<td>Doodle</td>
<td>0.055</td>
<td>0.078</td>
<td>0.055</td>
<td>0.050</td>
<td>0.045</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>0.069</td>
<td>0.084</td>
<td>0.054</td>
<td>0.060</td>
<td>0.036</td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mask</td>
<td>12.81</td>
<td>12.66</td>
<td>15.97</td>
<td>17.37</td>
<td>16.42</td>
<td></td>
</tr>
<tr>
<td>Doodle</td>
<td>12.92</td>
<td>12.68</td>
<td>14.79</td>
<td>15.40</td>
<td>16.98</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>14.93</td>
<td>15.00</td>
<td>15.36</td>
<td>17.04</td>
<td>19.34</td>
<td></td>
</tr>
</tbody>
</table>
Linear latent variable models

Top-down from hidden variables (factors, sources, causes, code)

\[ X_i = W h_i + \epsilon_i, \quad i = 1, \ldots, n. \]

- Loading/connection weights: 
  \[ x_{ij} = \sum_{k=1}^{d} w_{jk} h_{ik} \]
- Basis vectors: 
  \[ X_i = \sum_{k=1}^{d} W_k h_{ik}. \]
- Matrix factorization: 
  \[ (X_1, \ldots, X_n) = W (h_1, \ldots, h_n) \]
- Distributed representation, embedding, disentangle
Factor analysis

\[ h_i \sim N(0, I_d) \]
\[ X_i = W h_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2 I_p) \]
\[ d < p, \quad i = 1, \ldots, n. \]

Decathlon \( p = 10 \), \( h_i = \) (strength, speed, endurance), \( d = 3 \)
Dimension reduction, principal component analysis
Disentangle, independent causes
Generalizing \( h_i \sim p(h) \)
Factor analysis

$$h_i \sim N(0, I_d)$$

$$X_i = Wh_i + \epsilon_i, \ \epsilon_i \sim N(0, \sigma^2 I_p)$$

(1) Generalize $$h_i \sim p(h)$$: ICA, SCA, NMF, RBM, DAE ...

(2) Generalize to non-linear mapping: $$X_i = g(h_i; W)$$
Make it deep

Generative Models
Ying Nian Wu
Background
Energy-based model
Latent variable model
Cooperative learning
Divergence triangle

Input layer
Hidden layer
Input 1
Input 2
Input 3
Input 4

generative
top-down deconvolutional image synthesis
discriminative
bottom-up convolutional feature extraction

input RGB image
feature maps
1st layer
sub-sampled
2nd layer
sub-sampled
more layers
Generator with known factors

[Dosovitskiy et al., 2016]

\[(h_i, X_i) : X_i = g(h_i; \alpha) \text{ supervised}\]

\[h \to h^{(L)} \to \ldots \to h^{(1)} \to X\]
Latent variable model: generator network

\[ h \sim N(0, I_d) \]
\[ X = g(h; \alpha) + \epsilon \]
\[ h^{(l-1)} = g_l(W_l h^{(l)} + b_l) \]
\[ h^{(L+1)} = h; \quad X = h^{(0)} \]

[Goodfellow et al. 2014; Kingma and Welling 2013]
Alternating back-propagation

[Han et al. 2017]

\[ L = \sum_{i=1}^{n} \left\| X_i - g(h_i; \alpha) \right\|^2 \]

**Inference:** \( h_i \leftarrow h_i + \gamma \frac{\partial L_i}{\partial h_i} \)

**Learning:** \( \alpha \leftarrow \alpha + \gamma \frac{\partial L}{\partial \alpha} \)
Alternating back-propagation

\[
h^{(l-1)} = g_l(W_l h^{(l)} + b_l),
\]
\[
h^{(L)} = h; \quad X = h^{(0)} = g(h; W)
\]
\[
\partial g(h; \alpha)/\partial \alpha; \partial g(h; \alpha)/\partial h
\]
\[
\partial h^{(l-1)}/\partial h^{(l)} \text{ shared computation}
\]
Alternating back-propagation

Joint distribution:

\[
\log p(h, X; \alpha) = \log [p(h)p(X|h; \alpha)] \\
= -\frac{1}{2\sigma^2} \|X - g(h; \alpha)\|^2 - \frac{1}{2} \|h\|^2 + \text{const.}
\]

Inference: \( h \sim p(h|X; \alpha) \) via Langevin dynamics

\[
h_{\tau+1} = h_{\tau} + \frac{\delta^2}{2} \frac{\partial}{\partial h} \log p(h_{\tau} | X, \alpha_t) + N(0, \delta^2 I_d)
\]

Learning with \( \{(h_i, X_i), i = 1, \ldots, n\} \)

\[
\alpha_{t+1} = \alpha_t + \gamma_t \frac{\partial}{\partial \alpha} \sum_{i=1}^{n} \|X_i - g(h_i; \alpha_t)\|^2
\]
Generator network

[Han et al. 2017]
$h = (h_1, h_2)$
Generator network

\[ d = 100 \]
Incomplete data

\[ \|X_i - g(h_i; \alpha)\|^2: \text{sum over visible pixels} \]

<table>
<thead>
<tr>
<th>experiment</th>
<th>P.5</th>
<th>P.7</th>
<th>P.9</th>
<th>M20</th>
<th>M30</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>0.0571</td>
<td>0.0662</td>
<td>0.0771</td>
<td>0.0773</td>
<td>0.1035</td>
</tr>
</tbody>
</table>

[Han et al. 2017]
Non-linear dimension reduction

Reconstruction error on testing examples

<table>
<thead>
<tr>
<th>experiment</th>
<th>$d = 20$</th>
<th>$d = 60$</th>
<th>$d = 100$</th>
<th>$d = 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABP</td>
<td>0.0810</td>
<td>0.0617</td>
<td>0.0549</td>
<td>0.0523</td>
</tr>
<tr>
<td>PCA</td>
<td>0.1038</td>
<td>0.0820</td>
<td>0.0722</td>
<td>0.0621</td>
</tr>
</tbody>
</table>
Shared representation

**Figure:** Face rotation results on testing images. First column: face image under standard pose ($0^\circ$). Second to fifth column: each pair shows the rotated face by our method (left) and the ground truth target (right).
Deformable generator

[Xing, et al., 2018]

Figure: Two generator networks: one appearance generator and one geometric generator.
Training images

Figure: Example training images from CelebA. The training set contains 10000 images from CelebA, and they are cropped to $64 \times 64$ pixels by the OpenFace. These faces have different colors, illuminations, identities, viewing angles, shapes, and expressions.
Appearance latent vector

**Figure:** Each dimension of the appearance latent vector encodes appearance information such as color, illumination and gender.
**Figure:** Each dimension of the geometric latent vector encodes fundamental geometric information such as shape and viewing angle. In the first line, the shape of the face changes from fat to thin from left to the right. In the second line, the pose of the face varies from left to right.
Transferring

Figure: The first row shows the 7 unseen faces from CelebA. The second row shows the generated faces by transferring and recombining the first row’s 2nd-7th faces’ geometric vectors with the first row’s 1th face’s appearance vector. The third row shows the generated faces by transferring and recombining the first row’s 2nd-7th faces’ appearance vectors with the first row’s 1st face’s geometric vector.
Cooperative learning

[Xie et al. 2017]

Bottom-up ConvNet

\[ \text{energy} \]

\[ \uparrow \]

signal

(a) Descriptor Net (teacher)

Top-down ConvNet

\[ \text{latent variables} \]

\[ \downarrow \]

signal

(b) Generator Net (student)

Student writes initial draft
Teacher revises
Student learns from revision
Teacher learns from outside review
Cooperative learning

D1 needs to dream hard, but generator is a better dreamer
D2 learns without thinking
G1 needs to think hard, but descriptor does not need to think. G2 learns only if latent factors known.
Cooperative learning

Generator → initial draft; Descriptor → revised draft
Generator reconstructs the revised, knowing latent factors
Descriptor shifts from revised towards observed
Generator shifts from initial towards revised
Jump-starting each other’s Langevin
Cooperative learning

Generator

G2 updating

G1 Langevin

generated latent factors

initial synthesized examples

Descriptor

D1 Langevin

D2 updating

revised synthesized examples

observed examples
Cooperative learning

\[
\hat{h}_i \sim \mathcal{N}(0, I_d) \\
\hat{X}_i = g(\hat{h}_i; \alpha) + \epsilon_i
\]

Key: \( \hat{h}_i \) known, no need for inference
MCMC teaching

\[ KL(p^{(t+1)} \| p_{\theta(t)}) \leq KL(q_{\alpha(t)} \| p_{\theta(t)}) \]

Descriptor:

\[ \min_{\theta} \left[ KL(P_{\text{data}} \| p_{\theta}) - KL(M_{\theta} q_{\alpha} \| p_{\theta}) \right], \]

Generator:

\[ \min_{\alpha} KL(M_{\theta} q_{\alpha t} \| q_{\alpha}). \]
MCMC teaching

MCMC: $P(t)$ \[\begin{array}{c}
\uparrow \\
\downarrow \\
\end{array}\] Markov transition \[\begin{array}{c}
\to \\
\downarrow \\
\end{array}\] $P(t+1)$

Generator: $\alpha(t)$ \[\begin{array}{c}
\uparrow \\
\downarrow \\
\end{array}\] Parameter updating \[\begin{array}{c}
\to \\
\downarrow \\
\end{array}\] $\alpha(t+1)$
Texture
Figure: Images generated by CoopNets learned from 10 Imagenet scene categories. The training set consists of 1100 images randomly sampled from each category.
Interpolation
Synthesis quality

Figure: Left: Average softmax class probability on single Imagenet category versus the number of training images. Middle: Top 5 classification error. Below: Average pairwise structural similarity.
**Synthesis quality**

**Table:** Inception scores of different methods on learning from 10 Imagenet scene categories. $n$ is the number of training images randomly sampled from each category.

<table>
<thead>
<tr>
<th>Method</th>
<th>$n = 50$</th>
<th>$n = 100$</th>
<th>$n = 300$</th>
<th>$n = 500$</th>
<th>$n = 700$</th>
<th>$n = 900$</th>
<th>$n = 1100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoopNets</td>
<td>2.66±.13</td>
<td>3.04±.13</td>
<td>3.41±.13</td>
<td>3.48±.08</td>
<td>3.59±.11</td>
<td>3.65±.07</td>
<td>3.79±.15</td>
</tr>
<tr>
<td>DCGAN</td>
<td>2.26±.16</td>
<td>2.50±.15</td>
<td>3.16±.15</td>
<td>3.05±.12</td>
<td>3.13±.09</td>
<td>3.34±.05</td>
<td>3.47±.06</td>
</tr>
<tr>
<td>EBGAN</td>
<td>2.23±.17</td>
<td>2.40±.14</td>
<td>2.62±.08</td>
<td>2.46±.09</td>
<td>2.65±.04</td>
<td>2.64±.04</td>
<td>2.75±.08</td>
</tr>
<tr>
<td>W-GAN</td>
<td>1.80±.09</td>
<td>2.19±.12</td>
<td>2.34±.06</td>
<td>2.62±.08</td>
<td>2.86±.10</td>
<td>2.88±.07</td>
<td>3.14±.06</td>
</tr>
<tr>
<td>VAE</td>
<td>1.62±.09</td>
<td>1.63±.06</td>
<td>1.65±.05</td>
<td>1.73±.04</td>
<td>1.67±.03</td>
<td>1.72±.02</td>
<td>1.73±.02</td>
</tr>
<tr>
<td>InfoGAN</td>
<td>2.21±.04</td>
<td>1.73±.01</td>
<td>2.15±.03</td>
<td>2.42±.05</td>
<td>2.47±.05</td>
<td>2.29±.03</td>
<td>2.08±.04</td>
</tr>
<tr>
<td>Method of</td>
<td>2.44±.27</td>
<td>2.38±.13</td>
<td>2.42±.09</td>
<td>2.94±.11</td>
<td>3.02±.06</td>
<td>3.08±.08</td>
<td>3.15±.06</td>
</tr>
<tr>
<td>Algorithm G</td>
<td>1.72±.07</td>
<td>1.94±.09</td>
<td>2.32±.09</td>
<td>2.40±.06</td>
<td>2.45±.05</td>
<td>2.54±.05</td>
<td>2.61±.06</td>
</tr>
<tr>
<td>Persistent CD</td>
<td>1.30±.08</td>
<td>1.94±.03</td>
<td>1.80±.02</td>
<td>1.53±.02</td>
<td>1.45±.04</td>
<td>1.35±.02</td>
<td>1.51±.02</td>
</tr>
</tbody>
</table>
Table: The performance of CoopNets, DCGAN, W-GAN, and VAE on LSUN bedrooms, CelebA and Cifar-10 datasets with respect to the Fréchet Inception Distance (FID).

<table>
<thead>
<tr>
<th></th>
<th>LSUN</th>
<th>CelebA</th>
<th>Cifar-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>W-GAN</td>
<td>67.72</td>
<td>52.54</td>
<td>48.40</td>
</tr>
<tr>
<td>DCGAN</td>
<td>70.40</td>
<td>21.40</td>
<td>37.70</td>
</tr>
<tr>
<td>VAE</td>
<td>243.47</td>
<td>50.53</td>
<td>126.32</td>
</tr>
<tr>
<td>Generator in CoopNets (ours)</td>
<td>64.30</td>
<td>16.98</td>
<td>35.25</td>
</tr>
<tr>
<td>Descriptor in CoopNets (ours)</td>
<td><strong>35.42</strong></td>
<td><strong>16.65</strong></td>
<td><strong>33.61</strong></td>
</tr>
</tbody>
</table>
Pattern completion

Table: Comparison of recovery performances of different methods in 3 experiments

<table>
<thead>
<tr>
<th>task</th>
<th>CoopNets</th>
<th>DCGAN</th>
<th>MRF $\ell_1$</th>
<th>MRF $\ell_2$</th>
<th>inter1</th>
<th>inter2</th>
<th>inter3</th>
<th>inter4</th>
<th>inter5</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M30</td>
<td>0.115</td>
<td>0.211</td>
<td>0.132</td>
<td>0.134</td>
<td>0.120</td>
<td>0.120</td>
<td>0.265</td>
<td>0.120</td>
<td>0.120</td>
</tr>
<tr>
<td>M40</td>
<td>0.124</td>
<td>0.212</td>
<td>0.148</td>
<td>0.149</td>
<td>0.135</td>
<td>0.135</td>
<td>0.314</td>
<td>0.135</td>
<td>0.135</td>
</tr>
<tr>
<td>M50</td>
<td>0.136</td>
<td>0.214</td>
<td>0.178</td>
<td>0.179</td>
<td>0.170</td>
<td>0.166</td>
<td>0.353</td>
<td>0.164</td>
<td>0.164</td>
</tr>
<tr>
<td>PSNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pattern completion
Pattern completion
[Han, et al., 2018]
Learning energy-based model, latent variable model, and inference model together.
Three models

Let $P_{\text{data}}(X)$ be the data distribution and $p_{\theta}(X)$ be the energy-based model. Let $q(h)q_{\alpha}(X|h)$ be the latent variable model. Let $\rho_{\phi}(h|X)$ be the inferential model. Consider joint distribution of $(X, h)$, three families of models can be considered:

- **$P_{\text{data}}$ distribution:** $P_{\text{data}}(h, X) = P_{\text{data}}(X)\rho_{\phi}(h|X)$
- **$P$ distribution:** $P(h, X) = p_{\theta}(X)\rho_{\phi}(h|X)$
- **$Q$ distribution:** $Q(h, X) = q(h)q_{\alpha}(X|h)$
Divergence triangle

\[
\max_{\theta} \min_{\alpha} \min_{\phi} [KL(P_{\text{data}} \| Q) + KL(Q \| P) - KL(P_{\text{data}} \| P)]
\]
Maximum likelihood and variational

\[
\text{MLE} : \min_{\theta} \text{KL}(P_{\text{data}}(X) \| q_\alpha(X))
\]

Variational Auto-encoder (VAE) [Kingma and Welling, 2014]

\[
\text{VAE} : \min_{\alpha} \min_{\phi} \text{KL}(P_{\text{data}} \| Q)
\]
Maximum likelihood and contrastive divergence

MLE: \( \min_{\alpha} KL(P_{\text{data}}(X) \parallel p_{\theta}(X)) \)

Adversarial Contrastive Divergence [Dai et al., 2017]

ACD: \( \min_{\theta} \max_{\phi} [KL(P_{\text{data}} \parallel P) - KL(Q \parallel P)] \)
Divergence triangle

\[
\max_{\theta} \min_{\alpha} \min_{\phi} \left[ \text{KL}(P_{\text{data}} \| Q) + \text{KL}(Q \| P) - \text{KL}(P_{\text{data}} \| P) \right],
\]

Moment matching and mode seeking
Wake and sleep (Hinton et al., 1998)
Actor and critic; policy and value
Results: 32x32

Figure: First row: Cifar10 dataset. Second row: SVHN dataset. From left to right: test images, test reconstruction, samples.
Results: 64x64

Figure: CelebA dataset. From left to right: original images, reconstruction, samples.
Fashion MNIST

**Figure:** Fashion dataset. original images, reconstruction, sample, sample reconstruction.
Conclusions

- Neural networks as powerful approximator and interpolator
- High dimensional linear spline with recursive partitioning
- Vector representation, embedding, encoder-decoder
- Computation vs representation, learned computation
- Energy-based model: value, critic, planning, optimal control, Lagrangian formulation
- Latent variable model: policy, actor, impulsive, initialization, Hamiltonian formulation
- More explicit models, sparsity and disentanglement, And-Or grammar, symbolic and logical reasoning