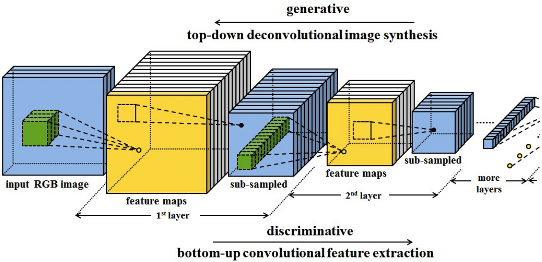
Cooperative Learning of Energy-based Model and Latent Variable Model via MCMC Teaching

Ying Nian Wu

Department of Statistics
University of California, Los Angeles

UC Irvine, Oct 25, 2018

A Tale of Two Nets



Bottom-up ConvNet	Top-down ConvNet
energy	**latent variables**
$\Uparrow$	$\Downarrow$
signal	signal
(a) Descriptor Net	(b) Generator Net
Energy-based Model	**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.)

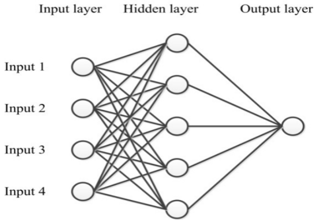
Modes of learning

- 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

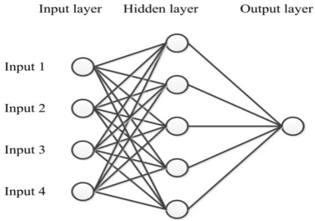


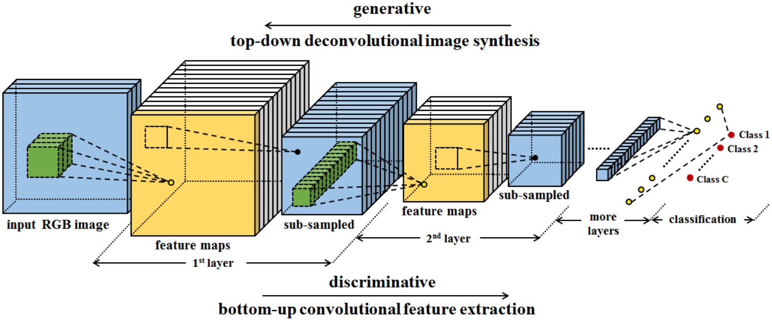
obs	input	hidden	output
1	$X_1^\top$	$h_1^\top$	y_1
2	$X_2^\top$	$h_2^\top$	y_2
...			
n	$X_n^\top$	$h_n^\top$	y_n

Supervised learning

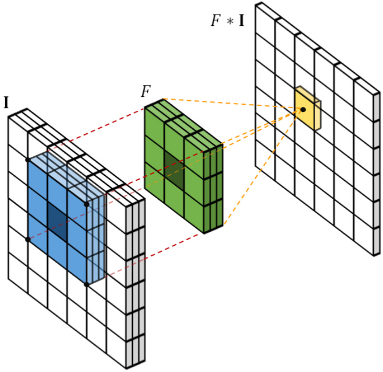
$$
\begin{array}{rc}
\text{output}: & Y_i \\
 & \uparrow \\
\text{hidden}: & h_i \\
 & \uparrow \\
\text{input}: & X_i
\end{array}
$$

ConvNet

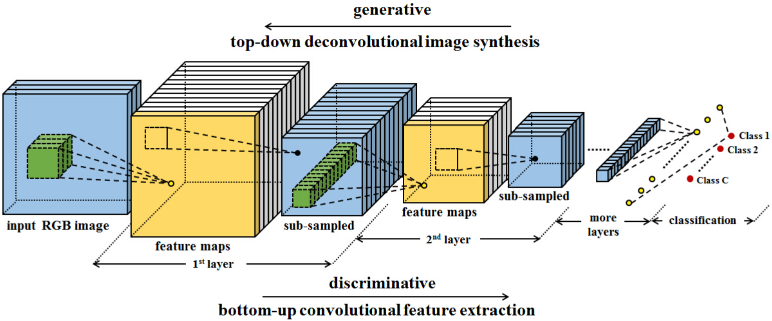

Input layer
Hidden layer
Output layer
Input 1
Input 2
Input 3
Input 4


generative
top-down deconvolutional image synthesis
input RGB image
feature maps
sub-sampled
feature maps
sub-sampled
more layers
classification
Class 1
Class 2
Class C
1st layer
2nd layer
discriminative
bottom-up convolutional feature extraction

Filtering



ConvNet



[Le Cun et al. 1998; Krizhevsky et al. 2012]

Coordinate-wise non-linearity

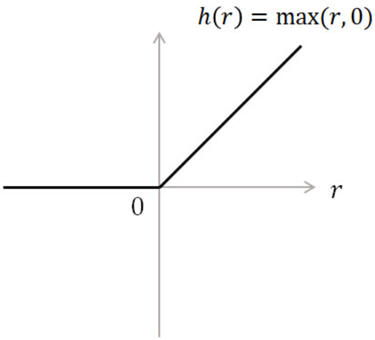
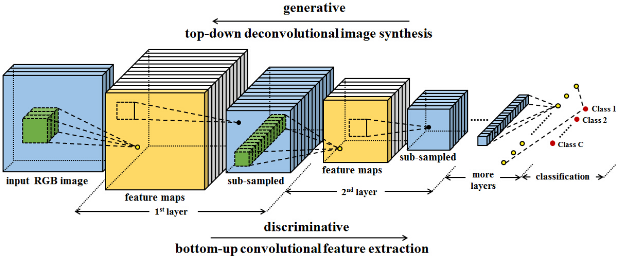


Figure: Rectified Linear Unit (ReLU).

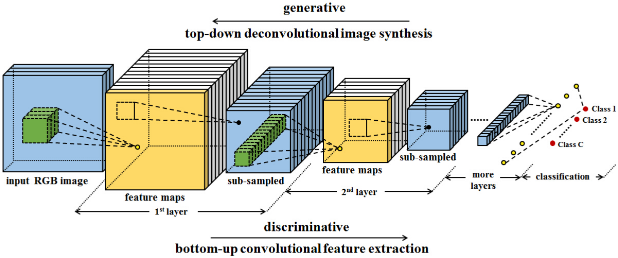
ConvNet



L-layer network

$$X \to h^{(1)} \to ...h^{(l-1)} \to h^{(l)} \to ... \to h^{(L)} \to \hat{Y},$$

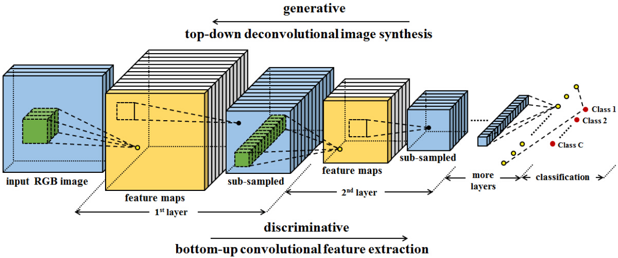
ConvNet



$$h^{(l)} = f_l(W_l h^{(l-1)} + b_l),$$

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

Back-propagation



$$\partial h^{(l)}/\partial h^{(l-1)} = f_l'(W_l h^{(l-1)} + b_l)W_l$$

End-to-end training

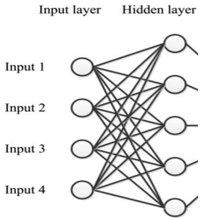
ConvNet

$$h^{(l)} = f_l(W_l h^{(l-1)} + b_l),$$

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



obs	input	hidden	output
1	$X_1^\top$	$h_1^\top$	?
2	$X_2^\top$	$h_2^\top$	?
...			
n	$X_n^\top$	$h_n^\top$	?

Recall supervised learning

$$\begin{array}{rc} \text{output}: & Y_i \\ & \uparrow \\ \text{hidden}: & h_i \\ & \uparrow \\ \text{input}: & X_i \end{array}$$

Unsupervised learning

Energy-based model

$$
\begin{array}{cc}
\text{energy}: & f(X_i;\theta) \\
 & \uparrow \\
\text{features}: & h_i \\
 & \uparrow \\
\text{input}: & X_i
\end{array}
$$

Latent variable model

$$
\begin{array}{cc}
\text{prior}: & p(h) \\
 & \downarrow \\
\text{hidden}: & h_i \\
 & \downarrow \\
\text{input}: & X_i
\end{array}
$$

Unsupervised learning

Auto-encoder: encoder and decoder

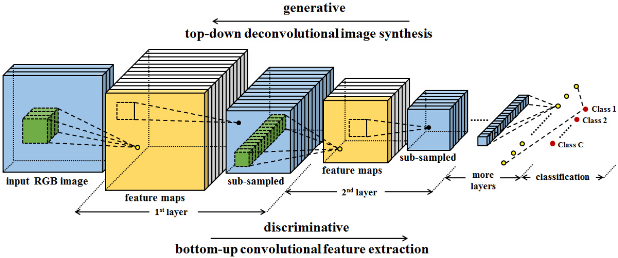
$$\begin{array}{rc} \text{code}: & h_i \\ & \uparrow\downarrow \\ \text{input}: & X_i \end{array}$$

Embedding: relative relationship

$$\begin{array}{c} \leftarrow h_i \rightarrow \\ | \\ \leftarrow X_i \rightarrow \end{array}$$

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

ConvNet



Bottom-up ConvNet	Top-down ConvNet
energy	**hidden variables** h
$\Uparrow$	$\Downarrow$
signal X	signal X
(a) $f(X;\theta)$	(b) $X = g(h;\alpha)$

Energy-based model: descriptor net

[Xie et al. 2016]

$$X \to h^{(1)} \to \ldots \to h^{(L)} \to f(X;\theta)$$

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

$p_0(X)$ 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:

$$\mathcal{E}(X;\theta) = \|X\|^2/2s^2 - f(X;\theta)$$

Relationship with discriminative net

[Xie et al. 2016]

$$X \to h^{(1)} \to ... \to h^{(L)} \to f(X;\theta_k)$$

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

$$\Pr(k|X) = \frac{\exp(f(X;\theta_k) + b_k)}{\sum_{k=0}^{K} \exp(f(X;\theta_k) + b_k)}, \tag{1}$$

Maximum likelihood

$$X_i \sim P_{\rm data}, i = 1, ..., n.$$

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

Log-likelihood:

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

minimize ${\rm KL}(P_{\rm data}|p_\theta)$
Maximum likelihood learning:

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

Energy-based model: descriptor net

Langevin revision:

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

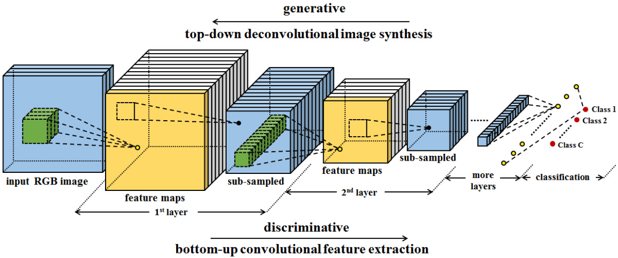
Density shifting:

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

Adversarial interpretation:

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

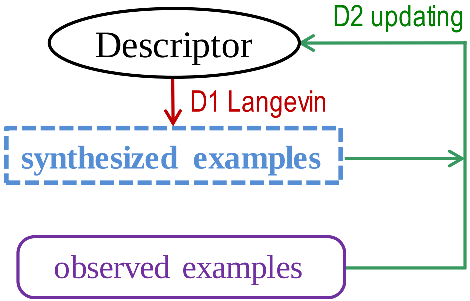
Back-propagation



$\partial f(X;\theta)/\partial\theta$ for updating θ
$\partial f(X;\theta)/\partial X$ for sampling X
The two derivatives share the same chain rule:

$$\partial h^{(l)}/\partial h^{(l-1)} = f_l'(W_l h^{(l-1)} + b_l)W_l$$

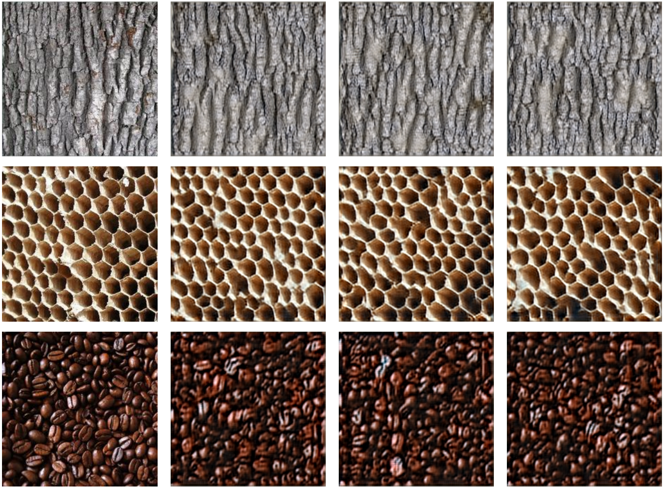
Analysis by synthesis



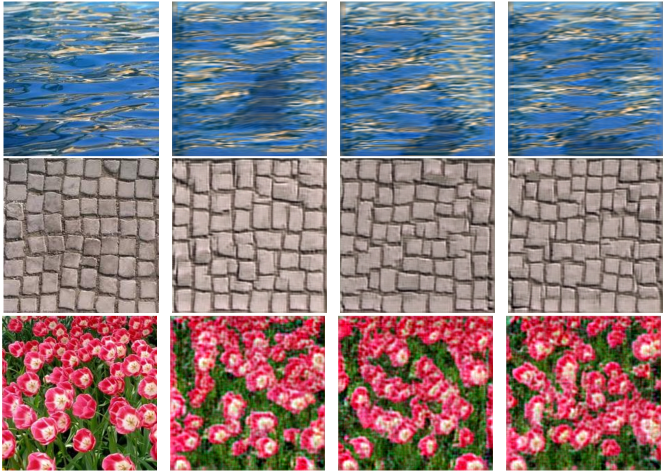
D1: Dreaming

D2: Make the dreaming more realistic

Texture

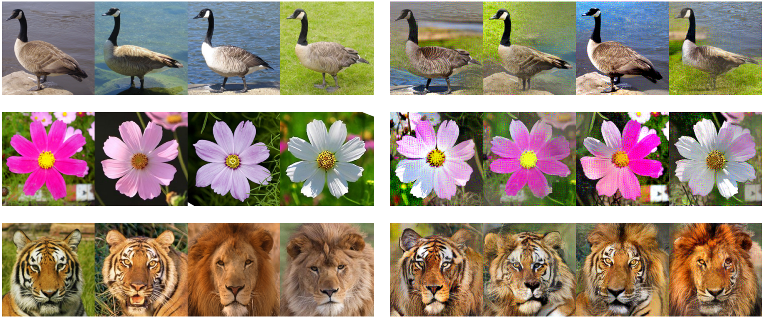


Texture



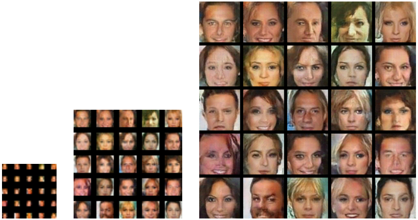
Object

[Lu et al. 2016]



Multi-grid

[Gao, Lu, Zhou, Zhu, Wu, 2018]



Forest road Volcano Hotel room Building facade

Multi-grid

	Real images	DCGAN	Multi-grid
Inception score	11.237	6.581	6.565

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.

Test error rate with # of labeled images	1,000	2,000	4,000
Persistent CD	45.74	39.47	34.18
One-step CD	44.38	35.87	30.45
Wasserstein GAN	43.15	38.00	32.56
Deep directed generative models	44.99	34.26	27.44
DCGAN	38.59	32.51	29.37
Single-grid CD	36.69	30.87	25.60
Multi-grid CD	**30.23**	**26.54**	**22.83**

Learning features

Table: Classification error of CNN classifier trained on the features of three grids learned from SVHN.

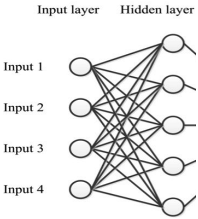
Test error rate with # of labeled images	1,000	2,000	4,000
DGN	36.02	-	-
Virtual adversarial	24.63	-	-
Auxiliary deep generative model	22.86	-	-
Supervised CNN with the same structure	39.04	22.26	15.24
Multi-grid CD + CNN classifier	**19.73**	**15.86**	**12.71**

Learning prior

Learning prior

	Mask	PCD	CD1	SCD	CE	MCD
Error	Mask	0.056	0.081	0.066	0.045	**0.042**
	Doodle	0.055	0.078	0.055	0.050	**0.045**
	Pepper	0.069	0.084	0.054	0.060	**0.036**
PSNR	Mask	12.81	12.66	15.97	**17.37**	16.42
	Doodle	12.92	12.68	14.79	15.40	**16.98**
	Pepper	14.93	15.00	15.36	17.04	**19.34**

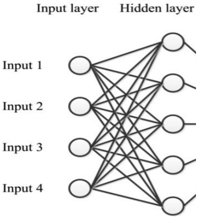
Linear latent variable models



Top-down from hidden variables (factors, sources, causes, code)
$X_i = W h_i + \epsilon_i$, $i = 1, ..., n$.

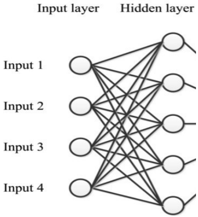
- Loading/connection weights: $x_{ij} = \sum_{k=1}^{d} w_{jk} h_{ik}$
- Basis vectors: $X_i = \sum_{k=1}^{d} W_k h_{ik}$.
- Matrix factorization: $(X_1, ..., X_n) = W(h_1, ..., h_n)$
- Distributed representation, embedding, disentangle

Factor analysis



$h_i \sim \mathrm{N}(0, I_d)$
$X_i = W h_i + \epsilon_i$, $\epsilon_i \sim \mathrm{N}(0, \sigma^2 I_p)$
$d < p$, $i = 1, ..., n$.
decathlon $p = 10$, $h_i =$ (strength, speed, endurance), $d = 3$
Dimension reduction, principal component analysis
Disentangle, independent causes
Generalizing $h_i \sim p(h)$

Make it deep



Factor analysis

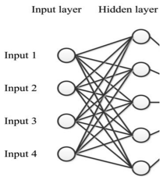
$h_i \sim \mathrm{N}(0, I_d)$

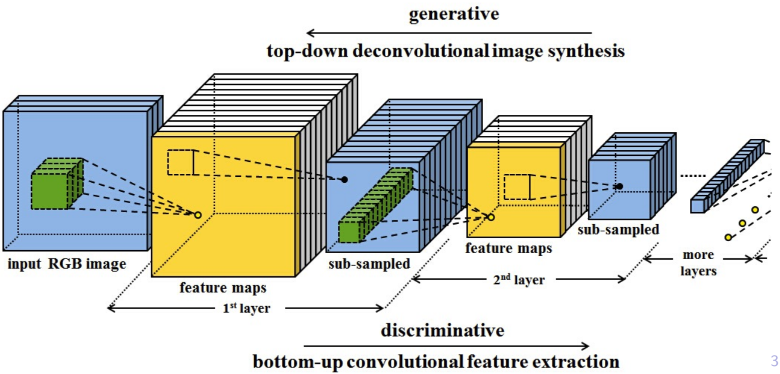
$X_i = W h_i + \epsilon_i$, $\epsilon_i \sim \mathrm{N}(0, \sigma^2 I_p)$

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

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

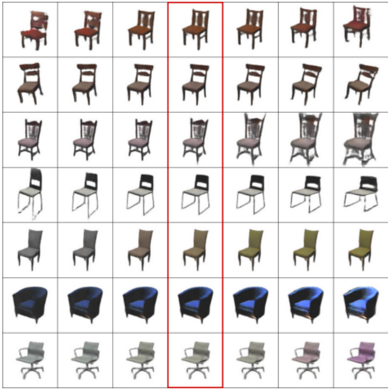
Make it deep


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


generative
top-down deconvolutional image synthesis
input RGB image
feature maps
sub-sampled
feature maps
sub-sampled
more layers
1st layer
2nd layer
discriminative
bottom-up convolutional feature extraction

Generator with known factors

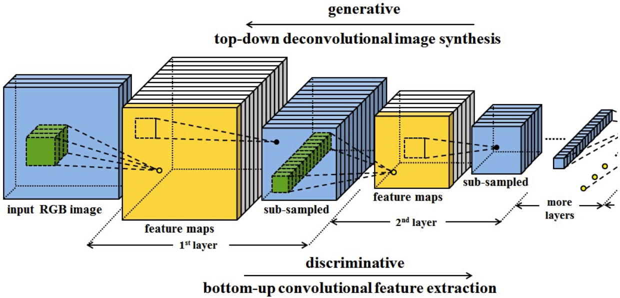
[Dosovitskiy et al., 2016]



$(h_i, X_i) : X_i = g(h_i; \alpha)$ supervised

$$h \to h^{(L)} \to ... \to h^{(1)} \to X$$

Latent variable model: generator network



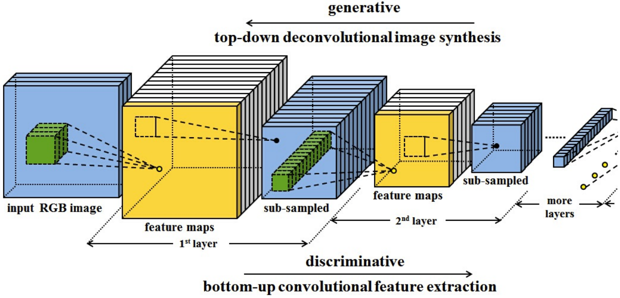
[Goodfellow et al. 2014; Kingma and Welling 2013]

$$h \sim \mathrm{N}(0, I_d)$$
$$X = g(h; \alpha) + \epsilon$$

$$h^{(l-1)} = g_l(W_l h^{(l)} + b_l)$$
$$h^{(L+1)} = h; \; X = h^{(0)}$$

Alternating back-propagation

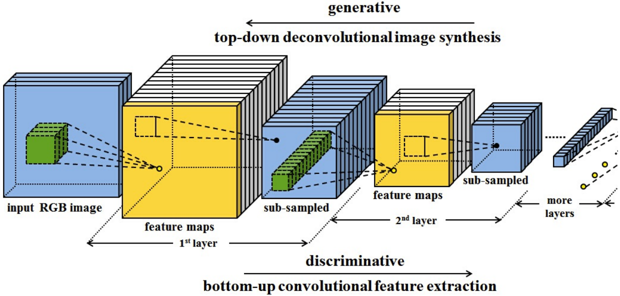
[Han et al. 2017]



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

- Inference: $h_i \leftarrow h_i + \gamma \partial L_i / \partial h_i$
- Learning: $\alpha \leftarrow \alpha + \gamma \partial L / \partial \alpha$

Alternating back-propagation



$$h^{(l-1)} = g_l(W_l h^{(l)} + b_l),$$

$$h^{(L)} = h;\ X = h^{(0)} = g(h; W)$$

$$\partial g(h;\alpha)/\partial\alpha; \partial g(h;\alpha)/\partial h$$

$\partial h^{(l-1)}/\partial h^{(l)}$ shared computation

Alternating back-propagation

Joint distribution:

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

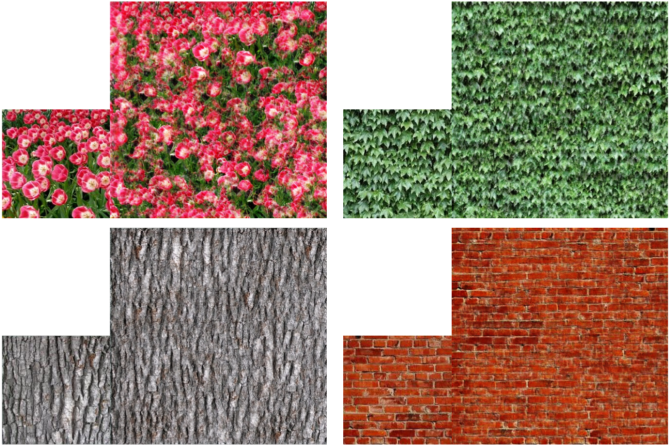
Inference: $h \sim p(h|X;\alpha)$ via Langevin dynamics

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

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

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

Generator network



[Han et al. 2017]

Generator network

$$h = (h_1, h_2)$$

Generator network

Generative Models

Ying Nian Wu

Background

Energy-based model

Latent variable model

Cooperative learning

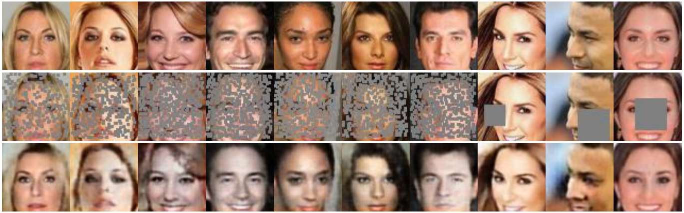
Divergence triangle

$d = 100$

Incomplete data

$\|X_i - g(h_i; \alpha)\|^2$: sum over visible pixels

experiment	P.5	P.7	P.9	M20	M30
error	.0571	.0662	.0771	.0773	.1035



[Han et al. 2017]

Non-linear dimension reduction

Reconstruction error on testing examples

experiment	$d = 20$	$d = 60$	$d = 100$	$d = 200$
ABP	.0810	.0617	.0549	.0523
PCA	.1038	.0820	.0722	.0621

Shared representation

Figure: Face rotation results on testing images. First column: face image under standard pose (0°). 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]

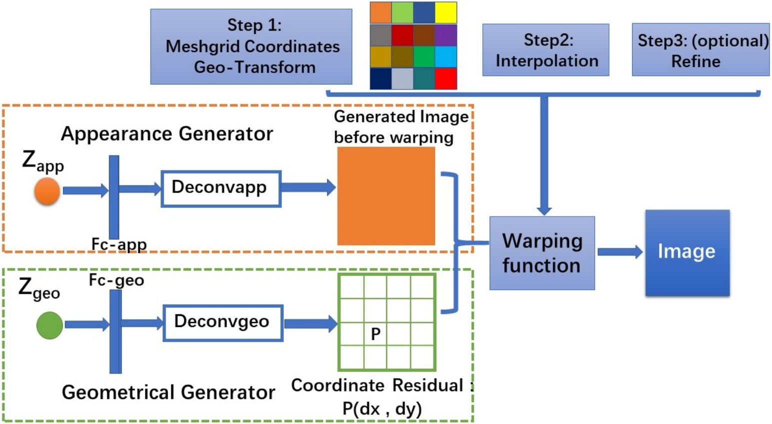


Figure: Two generator networks: one appearance generator and one geometric generator.

Training images

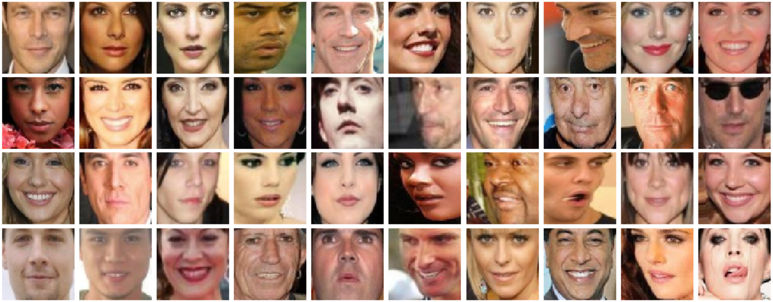


Figure: Example training images from CelebA. The training set contains 10000 images from CelebA, and they are cropped to 64×64 pixels by the OpenFace. These faces have different colors, illuminations, identities, viewing angles, shapes, and expressions.

Appearance latent vector

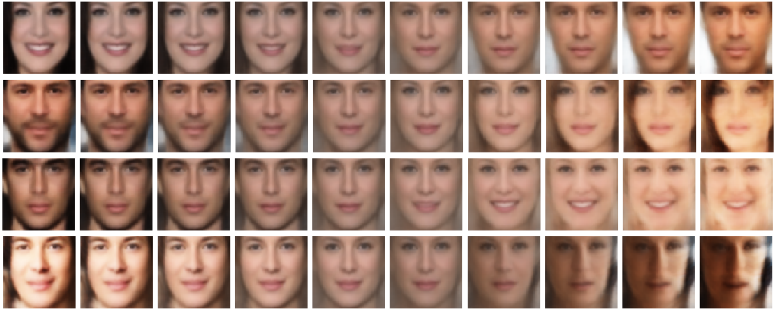


Figure: Each dimension of the appearance latent vector encodes appearance information such as color, illumination and gender.

Geometric latent vector

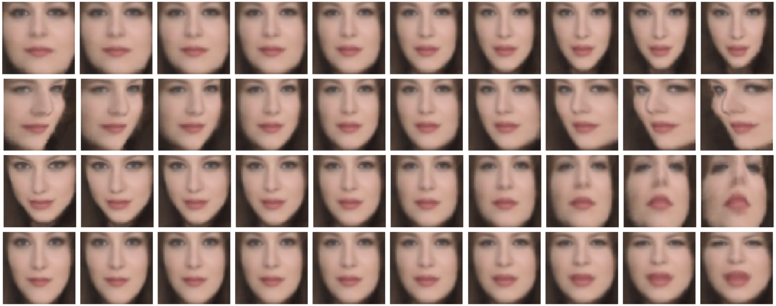


Figure: Each dimension of the geometric latent vector encodes fundamental geometric information such as shape and viewing angle. In the first line, the shape of the face changes from fat to thin from left to the right. In the second line, the pose of the face varies from left to right.

Transferring

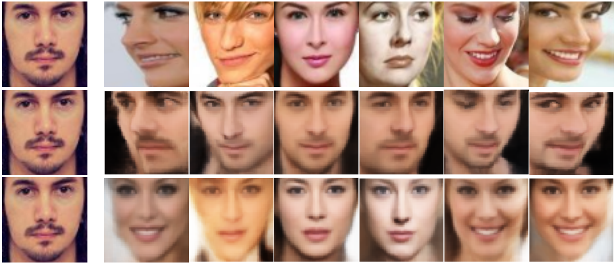


Figure: The first row shows the 7 unseen faces from CelebA. The second row shows the generated faces by transferring and recombining the first row's 2nd-7th faces' geometric vectors with the first row's 1th face's appearance vector. The third row shows the generated faces by transferring and recombining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

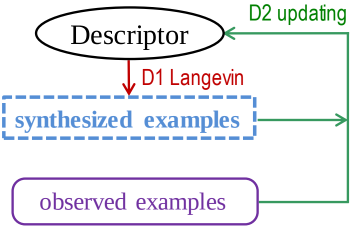
Cooperative learning

[Xie et al. 2017]

Bottom-up ConvNet	Top-down ConvNet
energy	**latent variables**
$\Uparrow$	$\Downarrow$
signal	signal
(a) Descriptor Net (teacher)	(b) Generator Net (student)

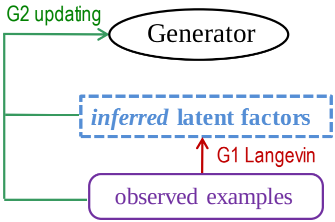
Student writes initial draft
Teacher revises
Student learns from revision
Teacher learns from outside review

Cooperative learning



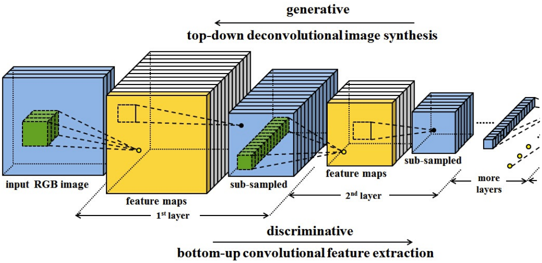
D1 needs to dream hard, but generator is a better dreamer
D2 learns without thinking

Cooperative learning



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

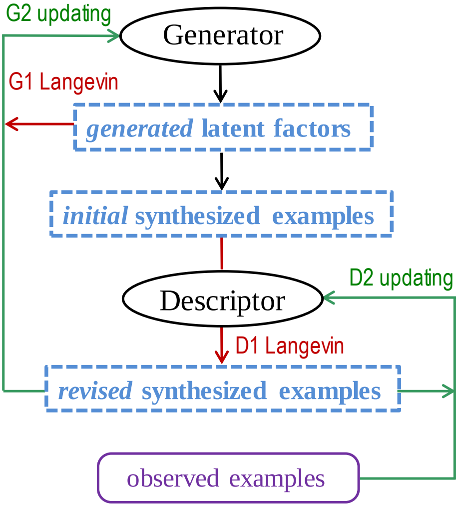
Generator reconstructs the revised, knowing latent factors

Descriptor shifts from revised towards observed

Generator shifts from initial towards revised

Jump-starting each other's Langevin

Cooperative learning



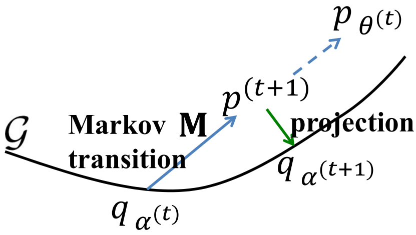
Cooperative learning

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

$$\begin{tikzcd}
\hat{h}_i \arrow[d, Rightarrow, "\alpha^{(t)}"'] \arrow[dr, Rightarrow, "\alpha^{(t+1)}"] & \\
\hat{X}_i \arrow[r, dashed, "\theta^{(t)}"'] & \tilde{X}_i
\end{tikzcd}
\qquad
\begin{tikzcd}
\hat{h}_i \arrow[r, dashed, "\alpha^{(t)}"] \arrow[d, Rightarrow, "\alpha^{(t)}"'] & \tilde{h}_i \arrow[d, Rightarrow, "\alpha^{(t+1)}"] \\
\hat{X}_i \arrow[r, dashed, "\theta^{(t)}"'] & \tilde{X}_i
\end{tikzcd}$$

Key: $\hat{h}_i$ known, no need for inference

MCMC teaching



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

Descriptor:

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

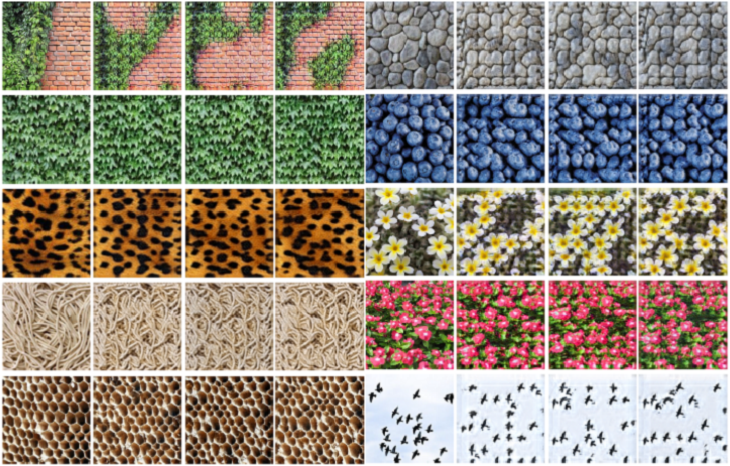
Generator:

$$\min_{\alpha} \mathrm{KL}(M_{\theta} q_{\alpha_t} \| q_{\alpha}).$$

MCMC teaching

$$
\begin{array}{llcl}
\text{MCMC}: & P^{(t)} & \xrightarrow{\text{Markov transition}} & P^{(t+1)} \\
 & \Updownarrow & & \Updownarrow \\
\text{Generator}: & \alpha^{(t)} & \xrightarrow{\text{Parameter updating}} & \alpha^{(t+1)}
\end{array}
$$

Texture



Scene

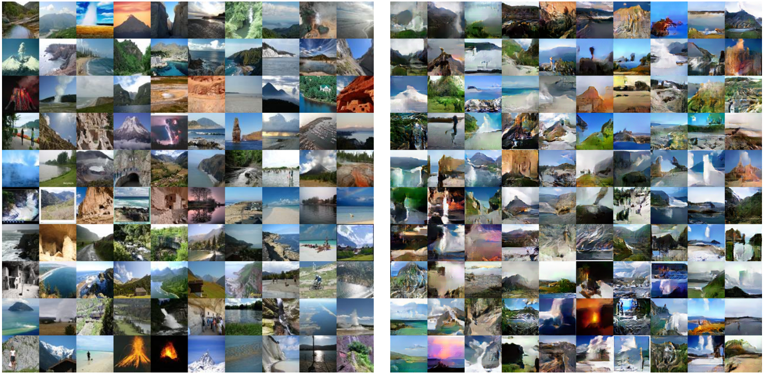
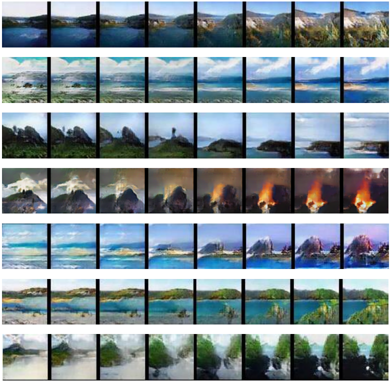


Figure: Images generated by CoopNets learned from 10 Imagenet scene categories. The training set consists of 1100 images randomly sampled from each category.

Interpolation



Synthesis quality

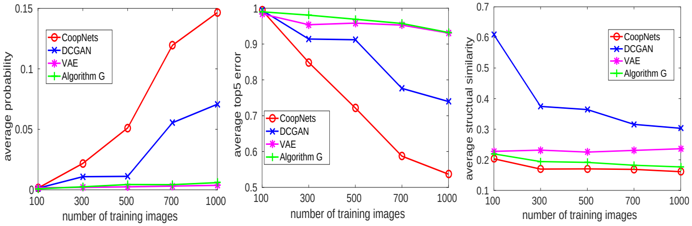


Figure: Left: Average softmax class probability on single Imagenet category versus the number of training images. Middle: Top 5 classification error. Below: Average pairwise structural similarity.

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.

	$n = 50$	$n = 100$	$n = 300$	$n = 500$	$n = 700$	$n = 900$	$n = 1100$
CoopNets	**2.66±.13**	**3.04±.13**	**3.41±.13**	**3.48±.08**	**3.59±.11**	**3.65±.07**	**3.79±.15**
DCGAN	2.26±.16	2.50±.15	3.16±.15	3.05±.12	3.13±.09	3.34±.05	3.47±.06
EBGAN	2.23±.17	2.40±.14	2.62±.08	2.46±.09	2.65±.04	2.64±.04	2.75±.08
W-GAN	1.80±.09	2.19±.12	2.34±.06	2.62±.08	2.86±.10	2.88±.07	3.14±.06
VAE	1.62±.09	1.63±.06	1.65±.05	1.73±.04	1.67±.03	1.72±.02	1.73±.02
InfoGAN	2.21±.04	1.73±.01	2.15±.03	2.42±.05	2.47±.05	2.29±.03	2.08±.04
Method of	2.44±.27	2.38±.13	2.42±.09	2.94±.11	3.02±.06	3.08±.08	3.15±.06
Algorithm G	1.72±.07	1.94±.09	2.32±.09	2.40±.06	2.45±.05	2.54±.05	2.61±.06
Persistent CD	1.30±.08	1.94±.03	1.80±.02	1.53±.02	1.45±.04	1.35±.02	1.51±.02

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).

	LSUN	CelebA	Cifar-10
W-GAN	67.72	52.54	48.40
DCGAN	70.40	21.40	37.70
VAE	243.47	50.53	126.32
Generator in CoopNets (ours)	64.30	16.98	35.25
Descriptor in CoopNets (ours)	**35.42**	**16.65**	**33.61**

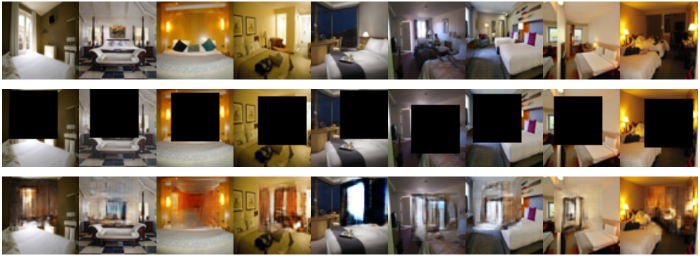
Pattern completion

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

	task	CoopNets	DCGAN	MRFℓ_1	MRFℓ_2	inter1	inter2	inter3	inter4	inter5
error	M30	**0.115**	0.211	0.132	0.134	0.120	0.120	0.265	0.120	0.120
	M40	**0.124**	0.212	0.148	0.149	0.135	0.135	0.314	0.135	0.135
	M50	**0.136**	0.214	0.178	0.179	0.170	0.166	0.353	0.164	0.164
PSNR	M30	**16.893**	12.116	15.739	15.692	16.203	16.635	9.524	16.665	16.648
	M40	**16.098**	11.984	14.834	14.785	15.065	15.644	8.178	15.698	15.688
	M50	**15.105**	11.890	13.313	13.309	13.220	14.009	7.327	14.164	14.161

Pattern completion

Pattern completion



Divergence triangle

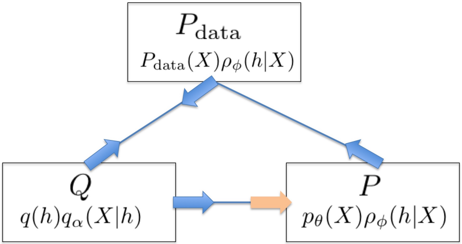
[Han, et al., 2018]
Learning energy-based model, latent variable model, and inference model together.

Three models

Let $P_{\rm data}(X)$ 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:

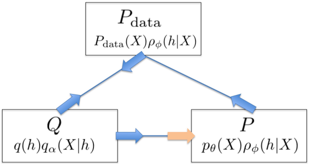
- $P_{\rm data}$ **distribution**: $P_{\rm data}(h, X) = P_{\rm data}(X)\rho_\phi(h|X)$
- P **distribution**: $P(h, X) = p_\theta(X)\rho_\phi(h|X)$
- Q **distribution**: $Q(h, X) = q(h)q_\alpha(X|h)$

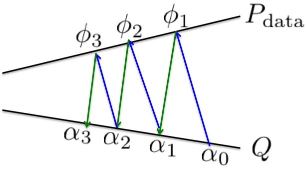
Divergence triangle



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

Maximum likelihood and variational



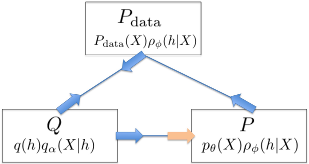


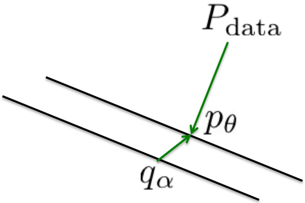
$$\text{MLE} : \min_\theta \text{KL}(P_{\text{data}}(X)\|q_\alpha(X))$$

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



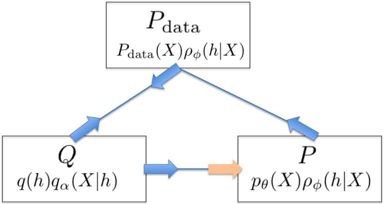


$$\text{MLE} : \min_{\alpha} \text{KL}(P_{\text{data}}(X) \| p_\theta(X))$$

Adversarial Contrasive Divergence [Dai et al., 2017]

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

Divergence triangle



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

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

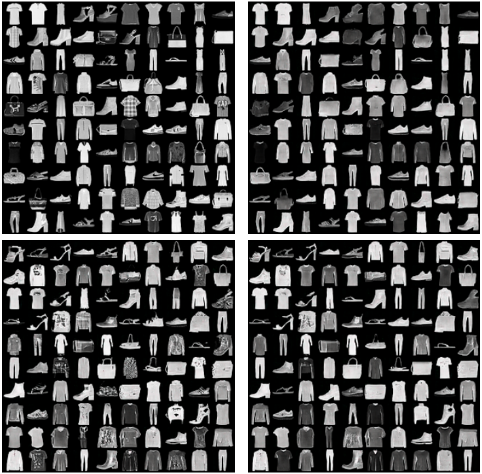


Figure: Fashion dataset. original images, reconstruction, sample, sample reconstruction.

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