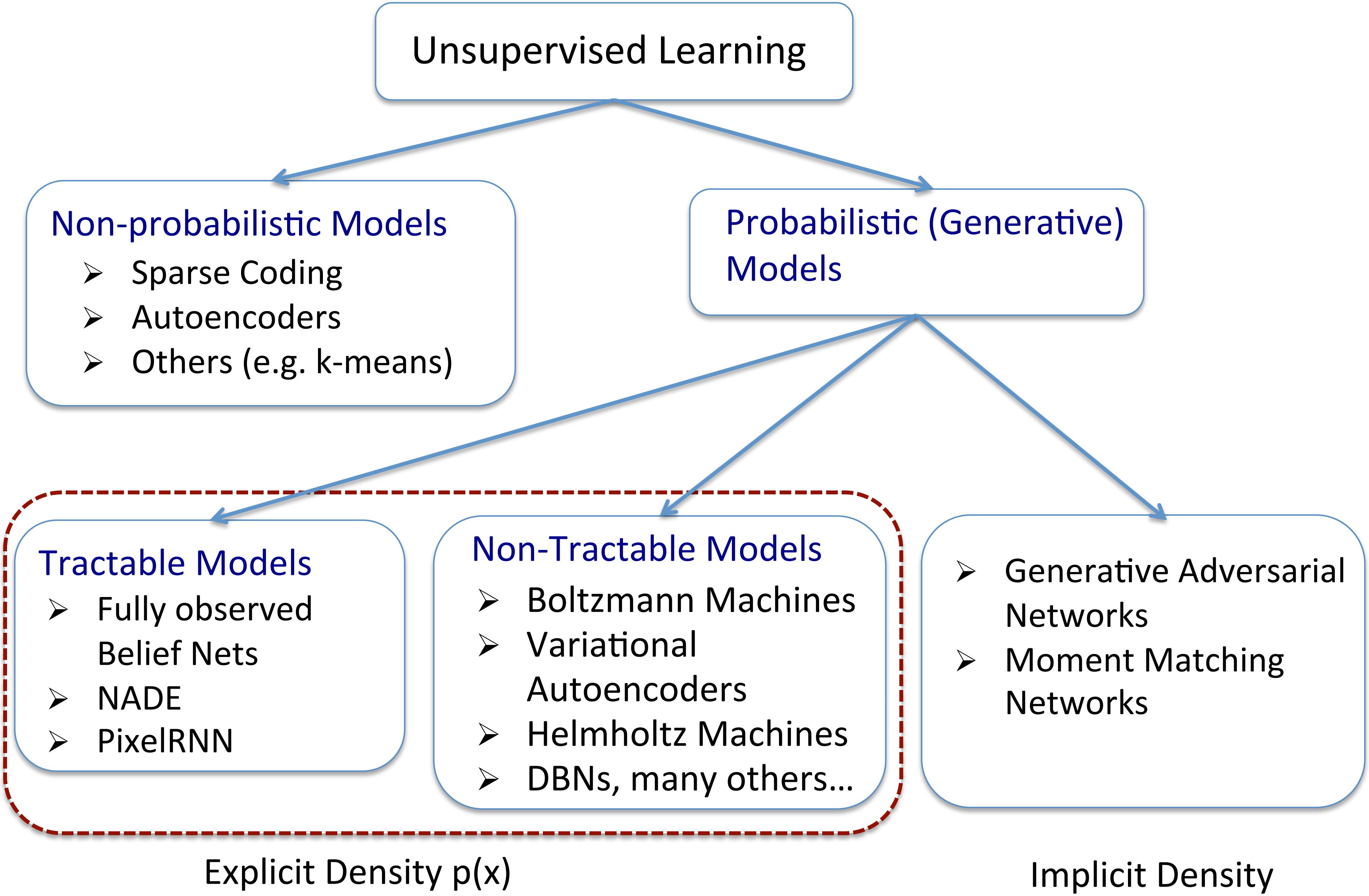
# 10707
# Deep Learning


Russ Salakhutdinov

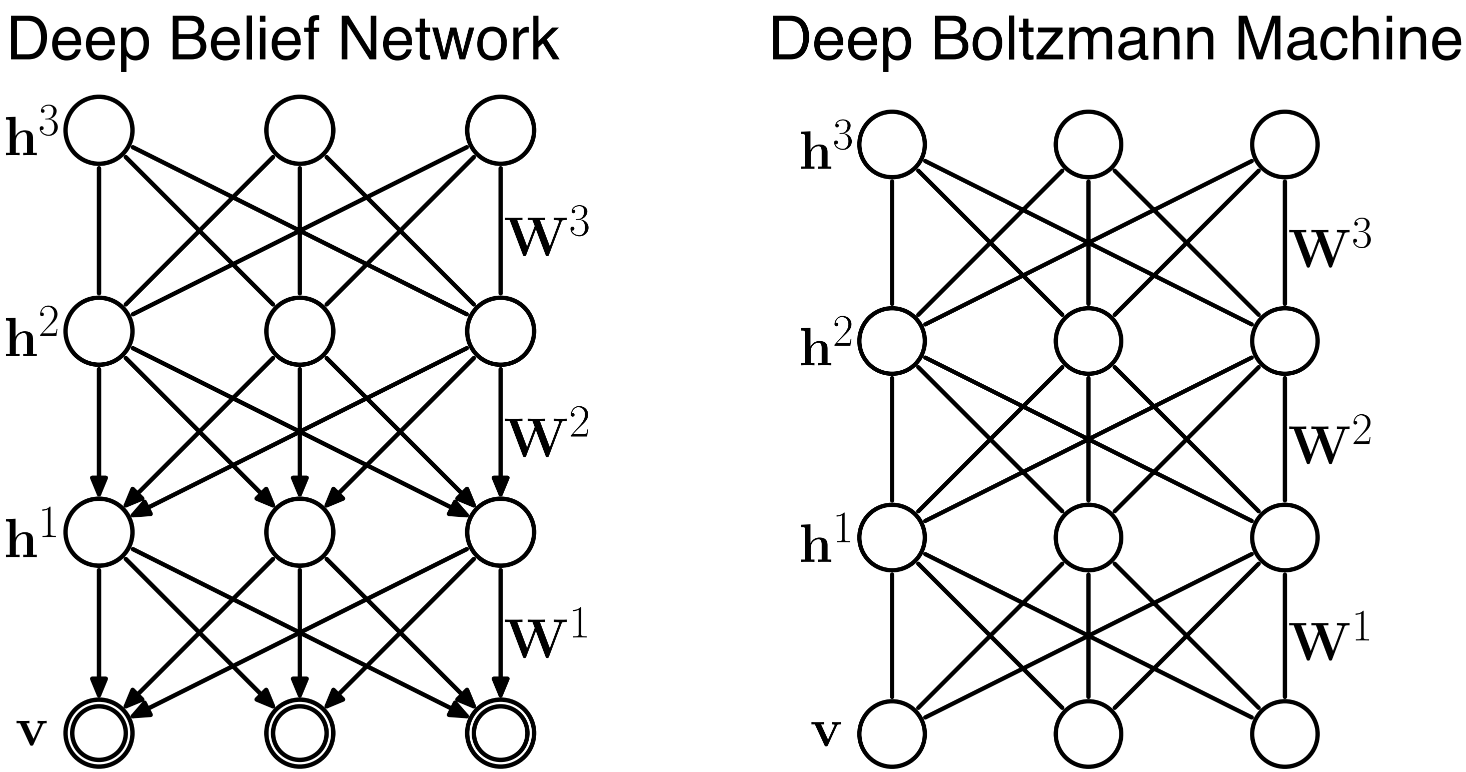
Machine Learning Department
rsalakhu@cs.cmu.edu


http://www.cs.cmu.edu/~rsalakhu/10707/

## Deep Boltzmann Machines I


Unsupervised Learning
Non-probabilistic Models
Sparse Coding
Autoencoders
Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
Fully observed Belief Nets
NADE
PixelRNN
Non-Tractable Models
Boltzmann Machines
Variational Autoencoders
Helmholtz Machines
DBNs, many others…
Generative Adversarial Networks
Moment Matching Networks
Explicit Density p(x)
Implicit Density

# DBNs vs. DBMs



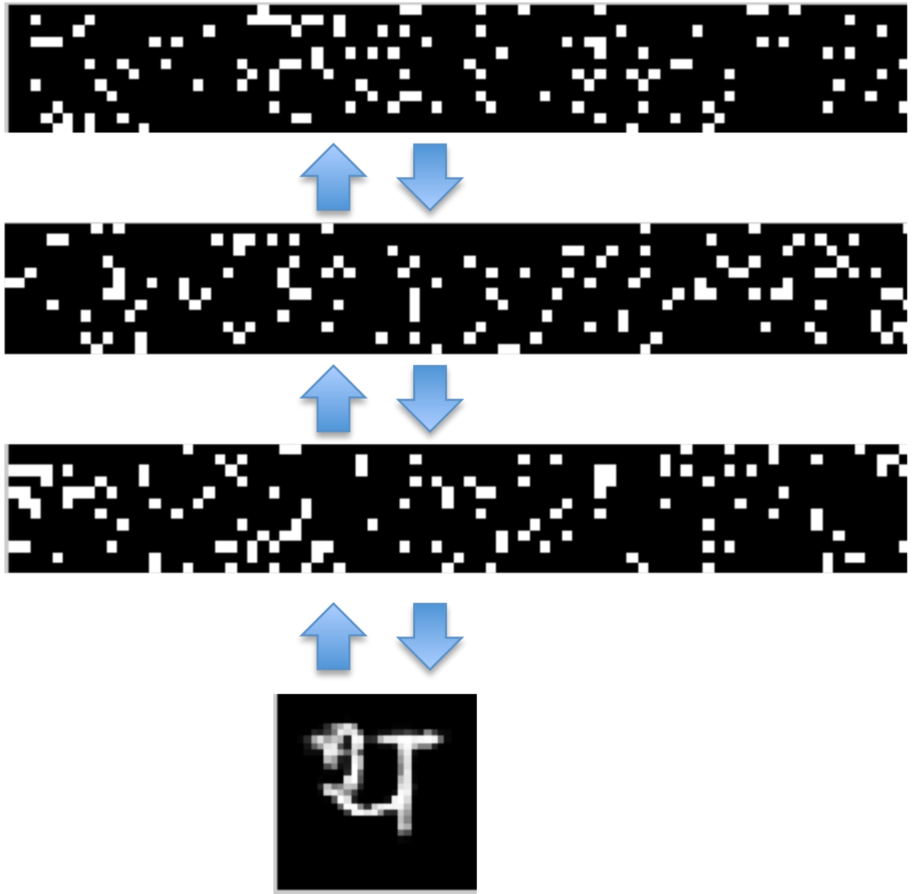

DBNs are hybrid models:

- Inference in DBNs is problematic due to **explaining away**.
- Only greedy pretrainig, **no joint optimization over all layers**.
- Approximate inference is feed-forward: **no bottom-up and top-down**.

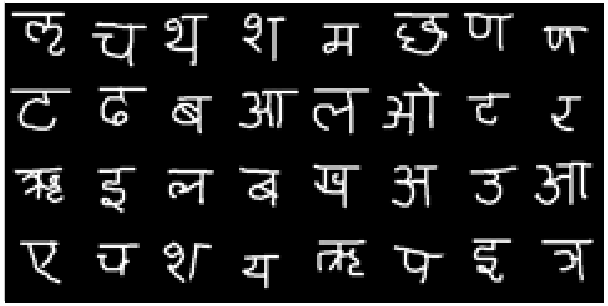
**Introduce a new class of models called Deep Boltzmann Machines.**

# Deep Generative Model

Sanskrit



Model P(image)



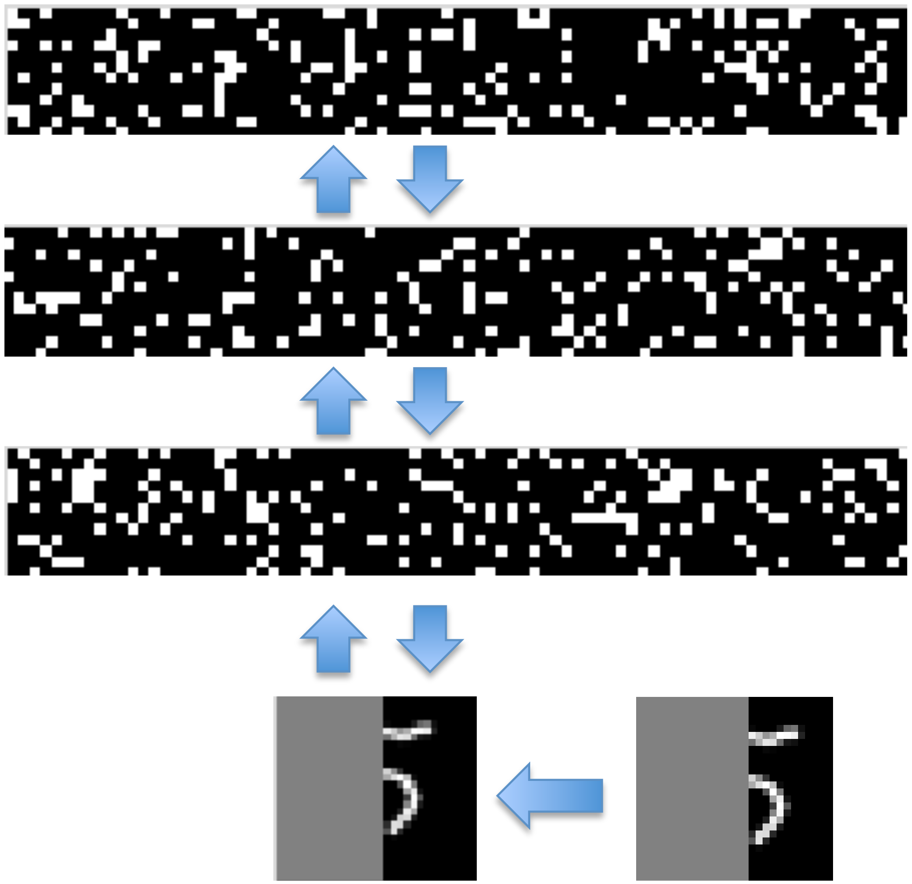
25,000 characters from 50 alphabets around the world.

- 3,000 hidden variables
- 784 observed variables (28 by 28 images)
- About 2 million parameters

Bernoulli Markov Random Field

# Deep Generative Model

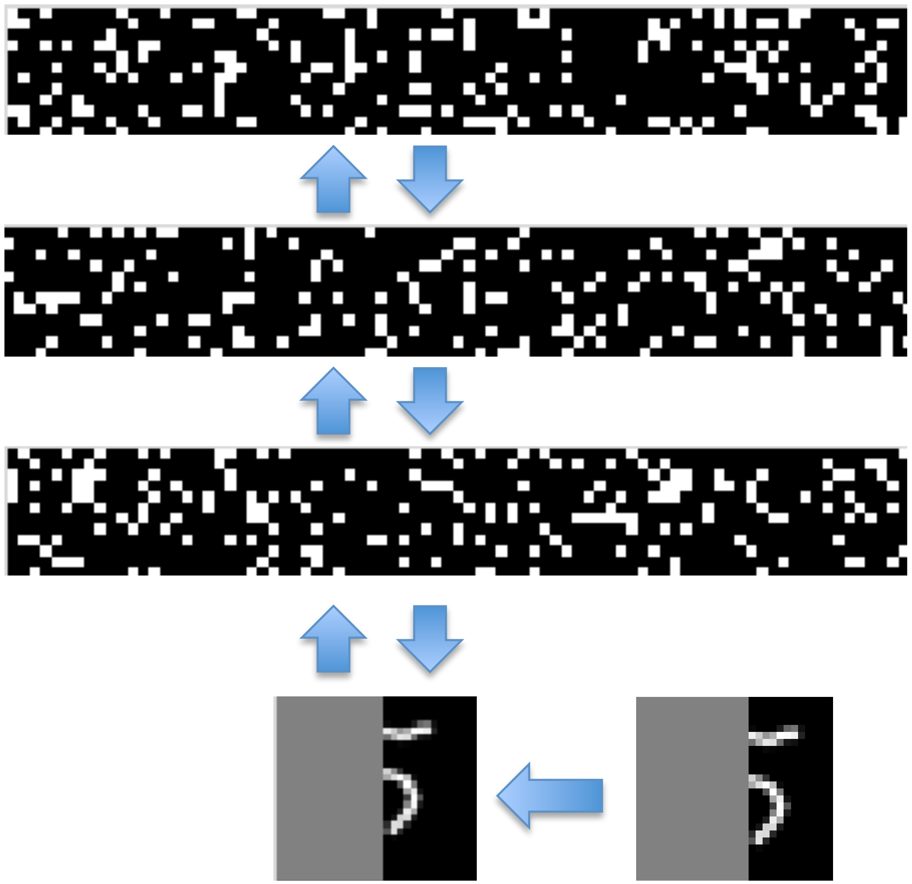
Conditional Simulation



P(image|partial image)

Bernoulli Markov Random Field

# Deep Generative Model



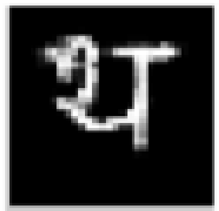
Conditional Simulation

P(image|partial image)

Why so difficult?

28

28



$2^{28\times 28}$ possible images!

Bernoulli Markov Random Field

# Fully Observed Models

- Explicitly model conditional probabilities:

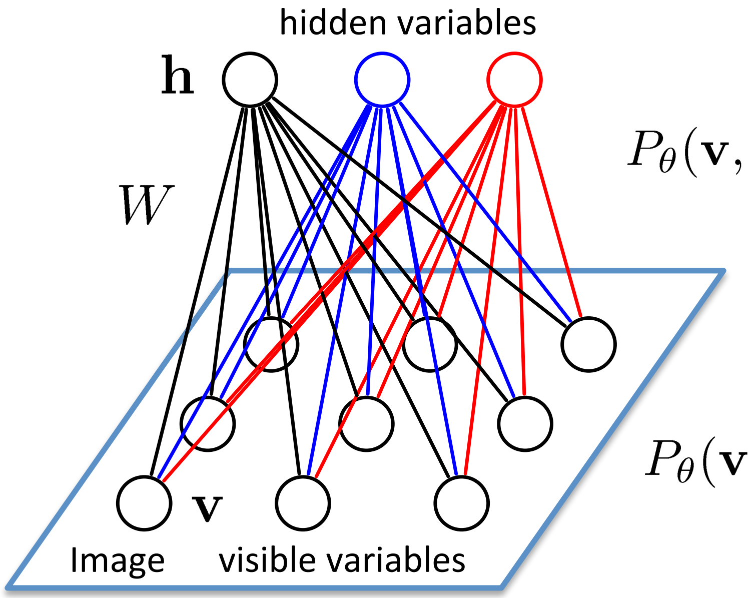
$$p_{\text{model}}(\boldsymbol{x}) = p_{\text{model}}(x_1) \prod_{i=2}^{n} p_{\text{model}}(x_i \mid x_1, \ldots, x_{i-1})$$

Each conditional can be a complicated neural network

- A number of successful models, including
  - ➢ NADE, RNADE (Larochelle, et.al. 20011)
  - ➢ Pixel CNN (van den Ord et. al. 2016)
  - ➢ Pixel RNN (van den Ord et. al. 2016)

Pixel CNN

# Restricted Boltzmann Machines




$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left( \overbrace{\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j}^{\textbf{Pair-wise}} + \overbrace{\sum_{i=1}^{D} v_i b_i + \sum_{j=1}^{F} h_j a_j}^{\textbf{Unary}} \right)$$

$$\theta = \{W, a, b\}$$

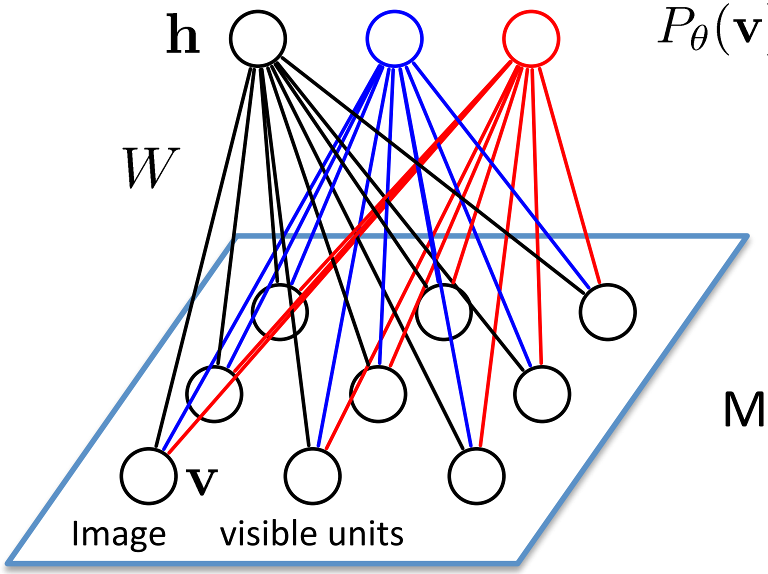
$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \frac{1}{1+\exp(-\sum_{j=1}^{F} W_{ij} v_i h_j - b_i)}$$

RBM is a Markov Random Field with:

- Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$.
- Stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.
- Bipartite connections.

Markov random fields, Boltzmann machines, log-linear models.

# Model Learning




$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}\right]$$

Given a set of *i.i.d.* training examples $\mathcal{D} = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N)}\}$ , we want to learn model parameters $\theta = \{W, a, b\}$.

Maximize log-likelihood objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)})$$

Derivative of the log-likelihood:

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log \left( \sum_{\mathbf{h}} \exp\left[\mathbf{v}^{(n)\top} W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}^{(n)}\right] \right) - \frac{\partial}{\partial W_{ij}} \log \mathcal{Z}(\theta)$$

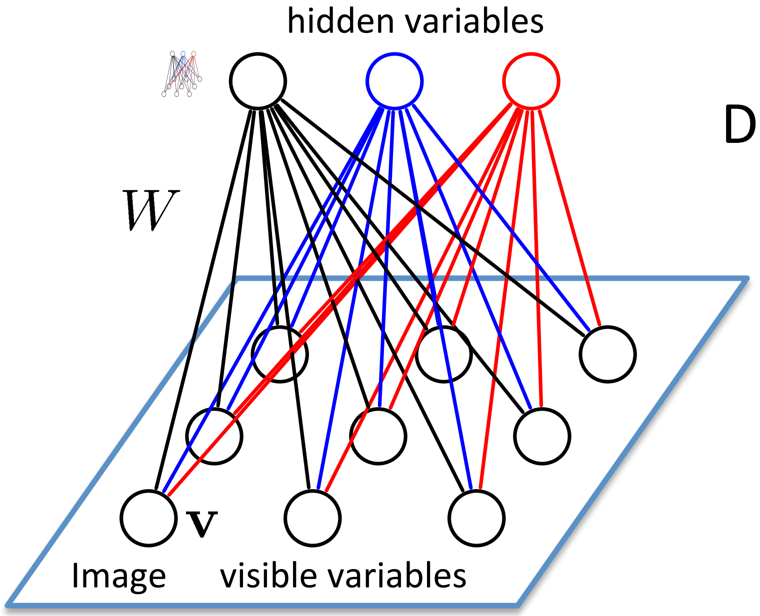
$$= \mathrm{E}_{P_{data}}[v_i h_j] - \underbrace{\mathrm{E}_{P_\theta}[v_i h_j]}$$

Difficult to compute: exponentially many configurations

$$P_{data}(\mathbf{v}, \mathbf{h}; \theta) = P(\mathbf{h}|\mathbf{v}; \theta) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N} \sum_{n} \delta(\mathbf{v} - \mathbf{v}^{(n)})$$

# Model Learning




Derivative of the log-likelihood:

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \mathbb{E}_{P_{data}}[v_i h_j] - \mathbb{E}_{P_\theta}[v_i h_j]$$

$$\sum_{\mathbf{v},\mathbf{h}} v_i h_j P_\theta(\mathbf{v},\mathbf{h})$$

Easy to compute exactly

Difficult to compute: exponentially many configurations.

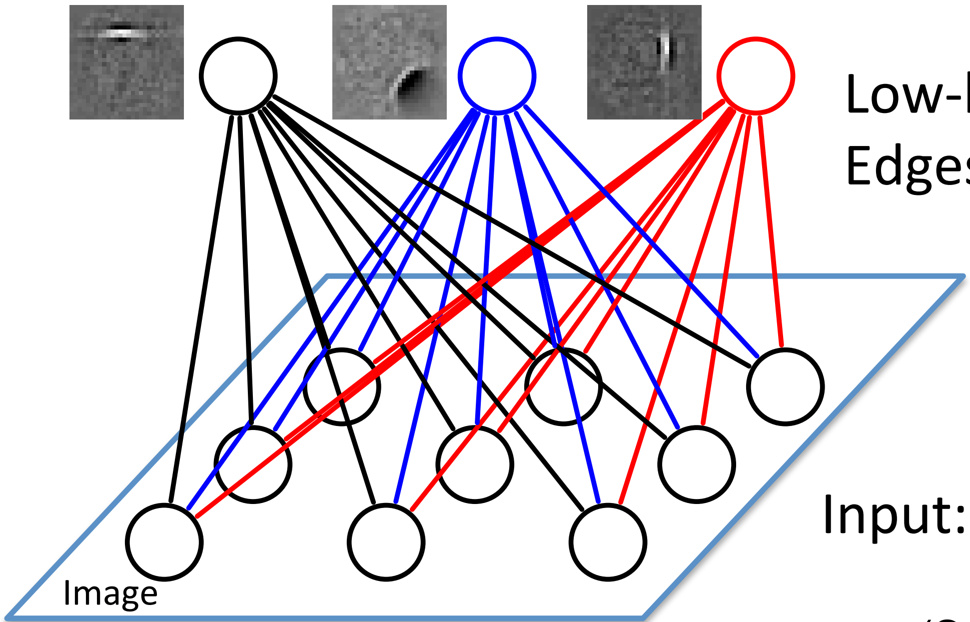
Use MCMC

$$P_{data}(\mathbf{v},\mathbf{h};\theta) = P(\mathbf{h}|\mathbf{v};\theta)P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_n \delta(\mathbf{v} - \mathbf{v}^{(n)})$$

**Approximate maximum likelihood learning**

# Deep Boltzmann Machines




Low-level features:
Edges

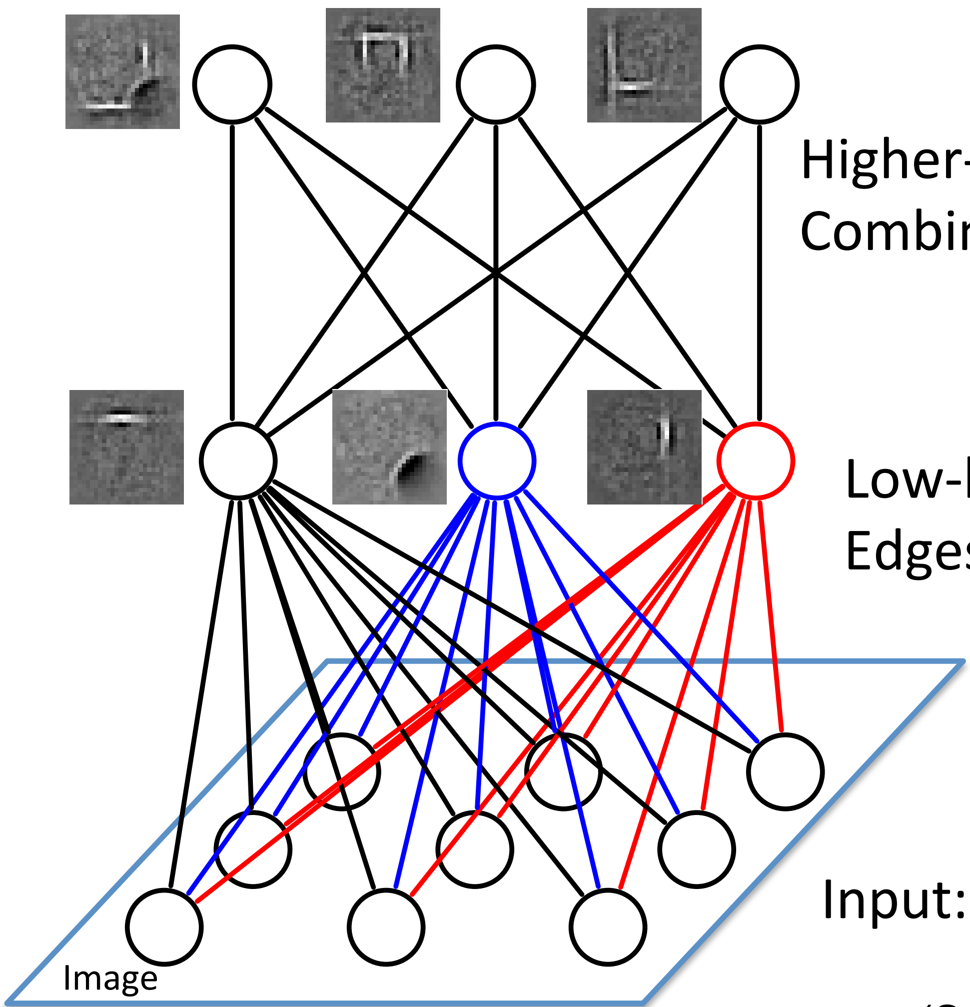
Built from **unlabeled** inputs.

Input: Pixels

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)

# Deep Boltzmann Machines

Learn simpler representations,
then compose more complex ones




Higher-level features:
Combination of edges

Low-level features:
Edges

Built from **unlabeled** inputs.

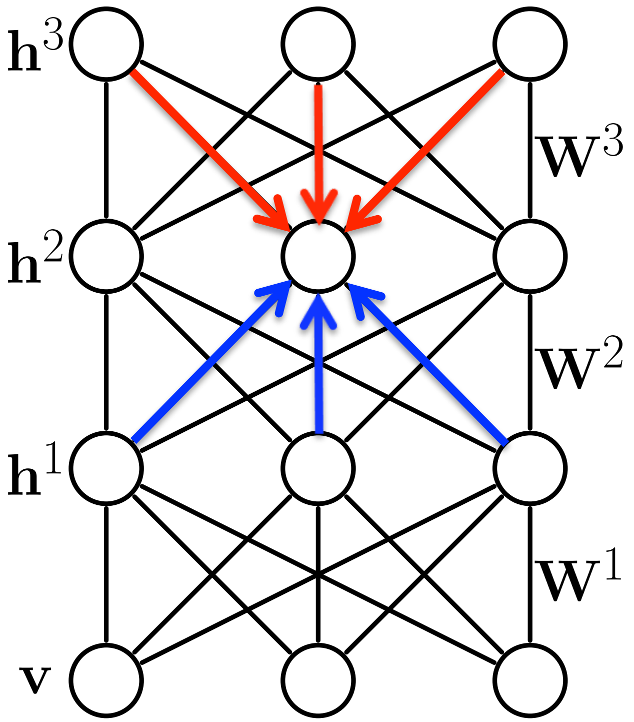
Input: Pixels

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)

# Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \underline{\mathbf{h}^{1\top} W^2 \mathbf{h}^2} + \underline{\mathbf{h}^{2\top} W^3 \mathbf{h}^3}\right]$$

Deep Boltzmann Machine




Input

$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.
- All connections are undirected.
- Bottom-up and Top-down:

$$P(h_k^2 = 1|\mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\sum_j W_{jk}^2 h_j^1 + \sum_m W_{km}^3 h_m^3\right)$$

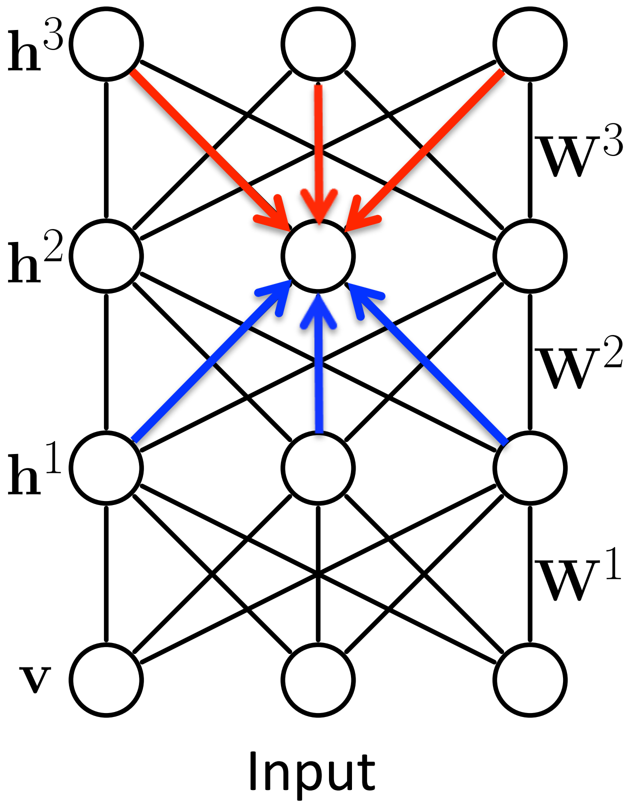
Bottom-up Top-Down

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio et.al.), Deep Belief Nets (Hinton et.al.)

# Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \underline{\mathbf{h}^{1\top} W^2 \mathbf{h}^2} + \underline{\mathbf{h}^{2\top} W^3 \mathbf{h}^3}\right]$$

Deep Boltzmann Machine




- Conditional Distributions:

$$P(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2) = \sigma\left(\sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 h_k^2\right)$$

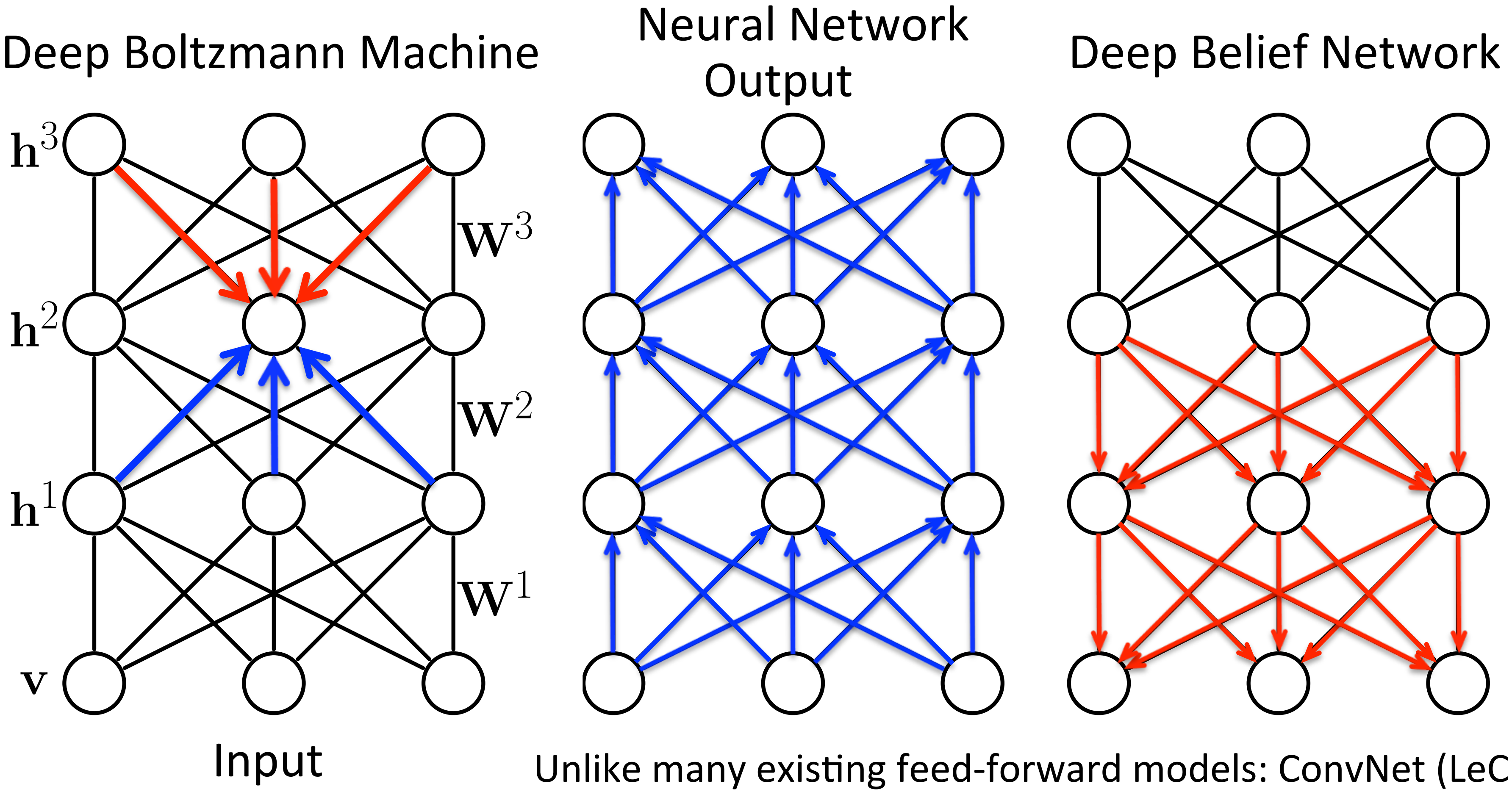
$$P(h_k^2 = 1|\mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\sum_j W_{jk}^2 h_j^1 + \sum_m W_{km}^3 h_m^3\right)$$

$$P(h_m^3 = 1|\mathbf{h}^2) = \sigma\left(\sum_k W_{km}^3 h_k^2\right)$$

- Note that exact computation of $P(\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3|\mathbf{v})$ is intractable.

# Mathematical Formulation

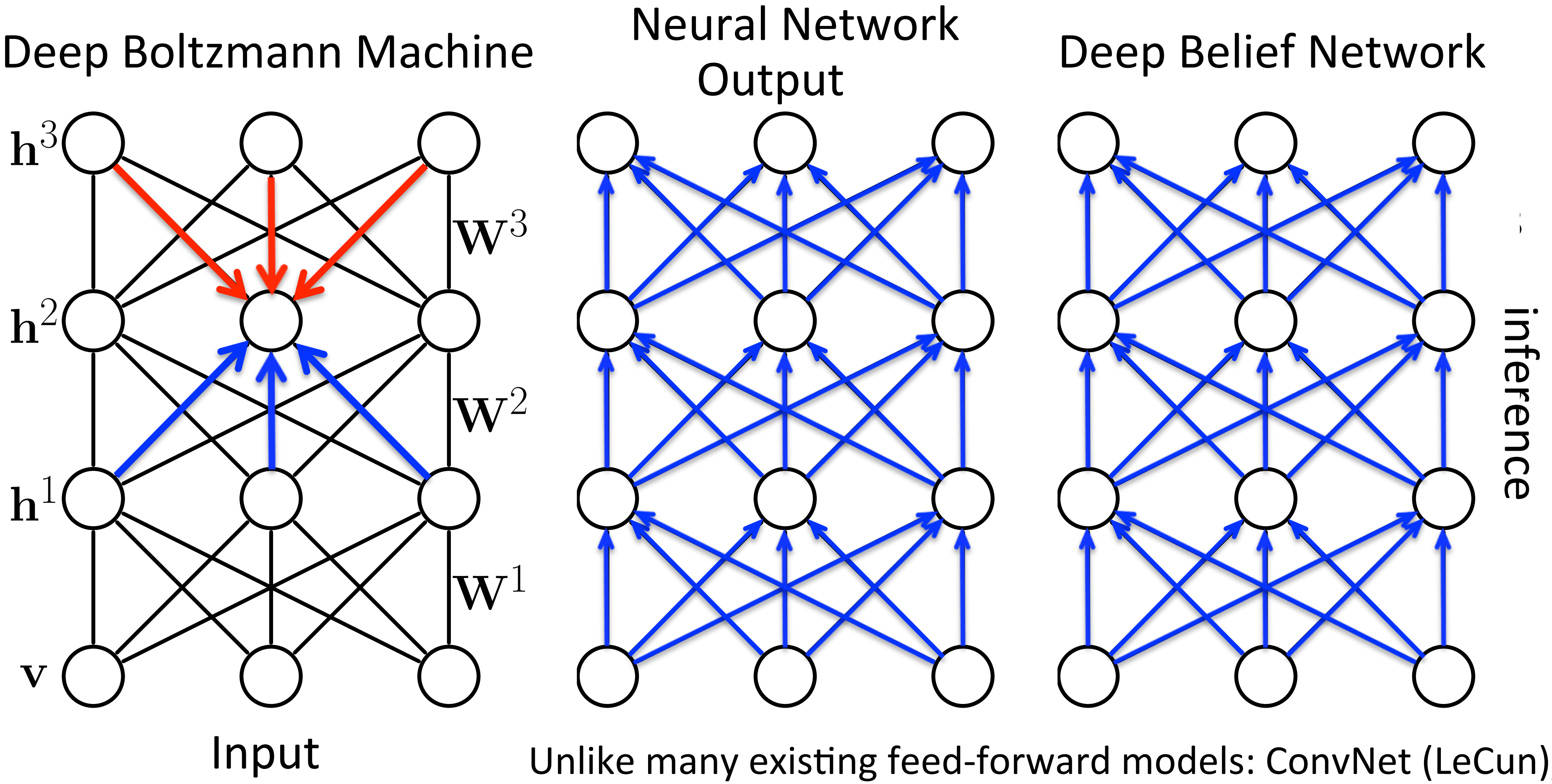
$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$




Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)

# Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

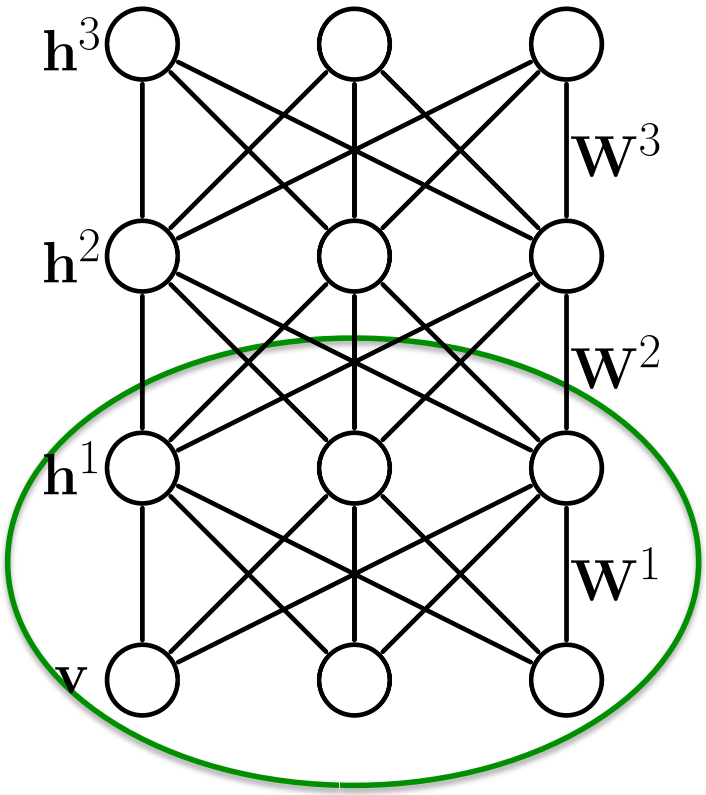



Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)

# Mathematical Formulation

$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine




$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.

Maximum likelihood learning:

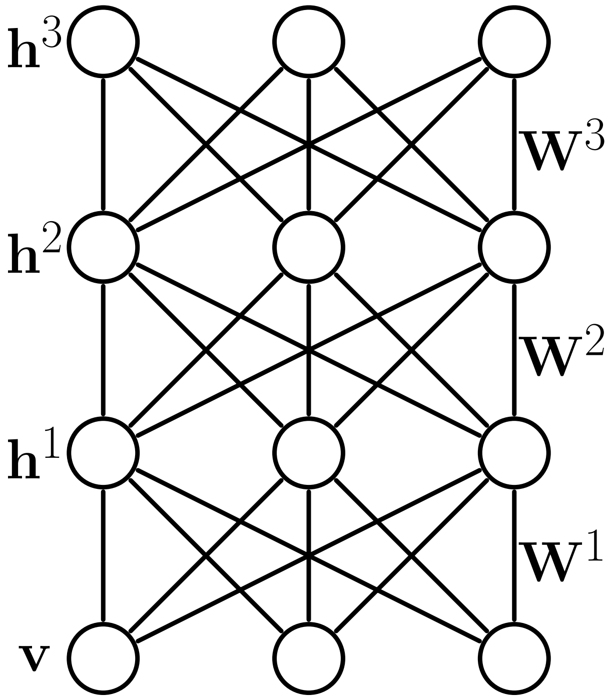
$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{1\top}] - \mathrm{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{1\top}]$$

**Problem:** Both expectations are intractable!

Learning rule for undirected graphical models: MRFs, CRFs, Factor graphs.

# Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}\right]$$




(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$

- Both expectations are intractable!

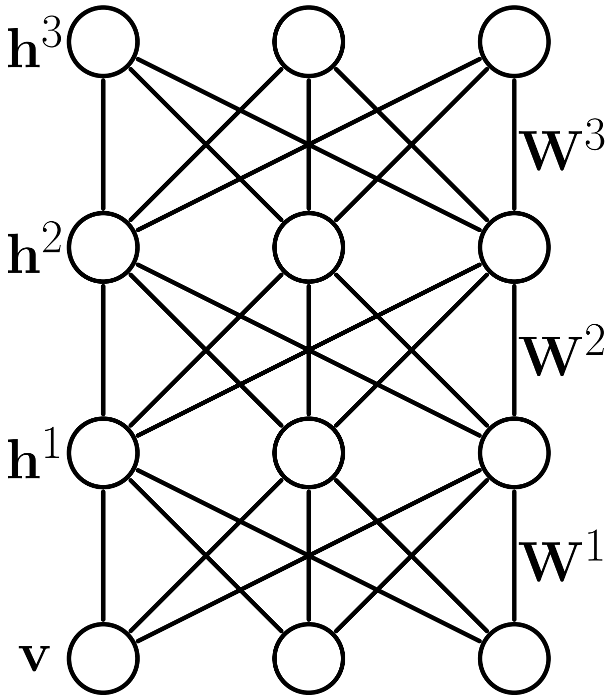
$$P_{data}(\mathbf{v}, \mathbf{h}^{\mathbf{1}}) = P_\theta(\mathbf{h}^{\mathbf{1}}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

Not factorial any more!

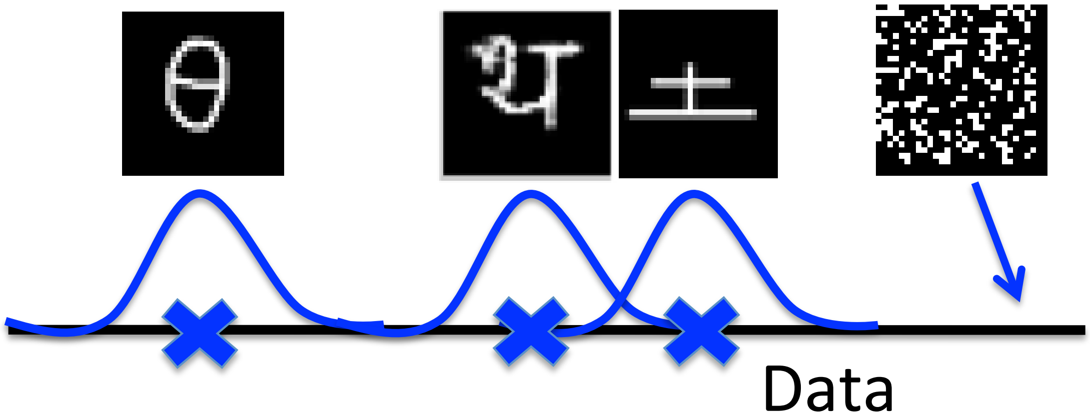
# Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)} \mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)} \mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)} \mathbf{h}^{(3)}\right]$$




(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$



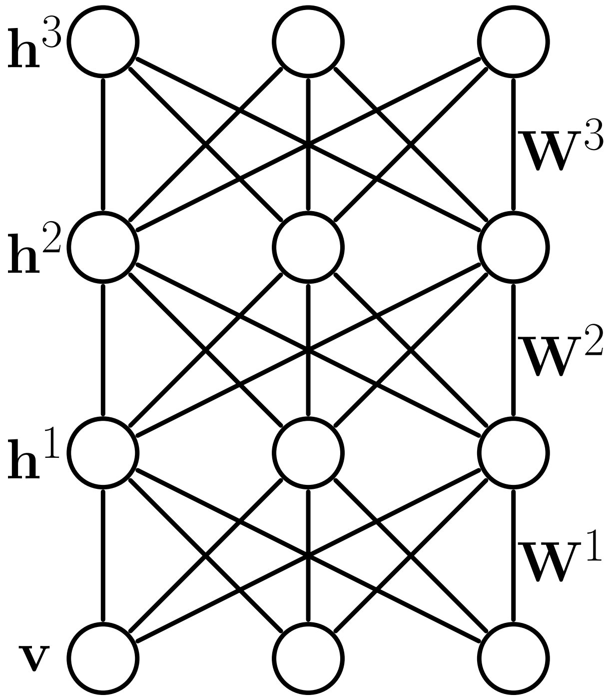

$$P_{data}(\mathbf{v}, \mathbf{h}^\mathbf{1}) = P_\theta(\mathbf{h}^\mathbf{1}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

Not factorial any more!

# Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}\right]$$




(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$

Variational Inference

Stochastic Approximation (MCMC-based)

$$P_{data}(\mathbf{v}, \mathbf{h}^{\mathbf{1}}) = P_\theta(\mathbf{h}^{\mathbf{1}}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

Not factorial any more!

# Previous Work

Many approaches for learning Boltzmann machines have been proposed over the last 20 years:

- Hinton and Sejnowski (1983),
- Peterson and Anderson (1987)
- Galland (1991)
- Kappen and Rodriguez (1998)
- Lawrence, Bishop, and Jordan (1998)
- Tanaka (1998)
- Welling and Hinton (2002)
- Zhu and Liu (2002)
- Welling and Teh (2003)
- Yasuda and Tanaka (2009)

Real-world applications – thousands of hidden and observed variables with millions of parameters.

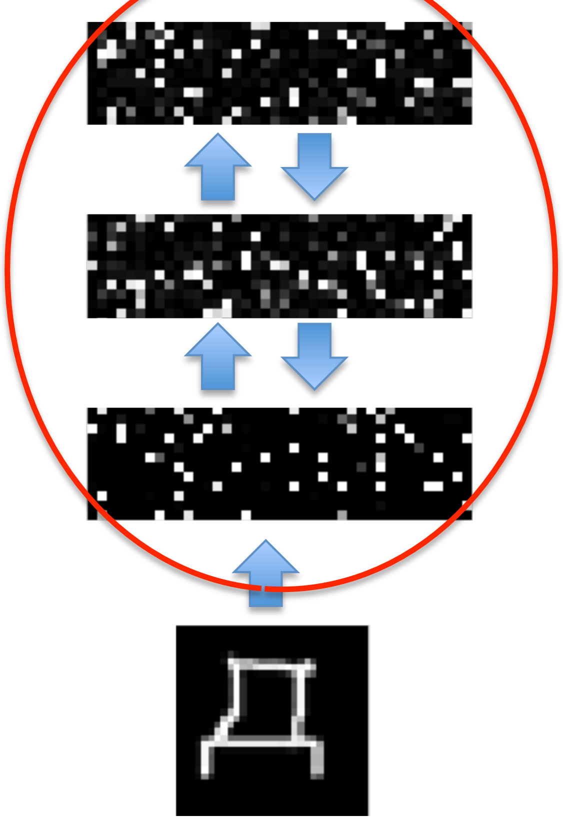
Many of the previous approaches were not successful for learning general Boltzmann machines with **hidden variables.**

Algorithms based on Contrastive Divergence, Score Matching, Pseudo-Likelihood, Composite Likelihood, MCMC-MLE, Piecewise Learning, cannot handle multiple layers of hidden variables.

# New Learning Algorithm

## Posterior Inference

Conditional



Approximate conditional

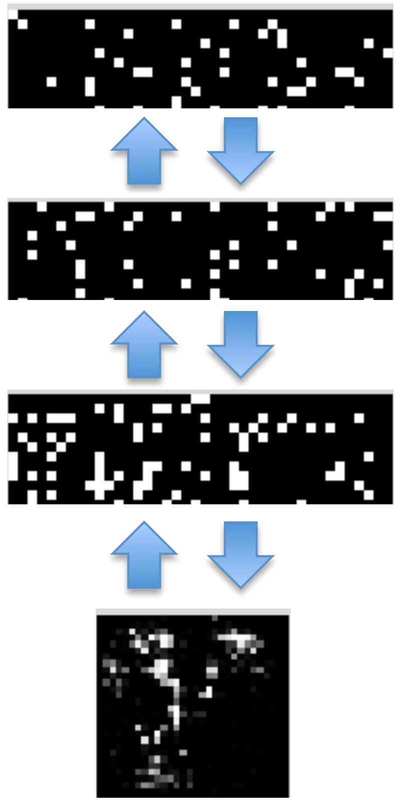
$$P_{data}(\mathbf{h}|\mathbf{v})$$

## Simulate from the Model

Approximate the joint distribution

$$P_{model}(\mathbf{h}, \mathbf{v})$$

Unconditional

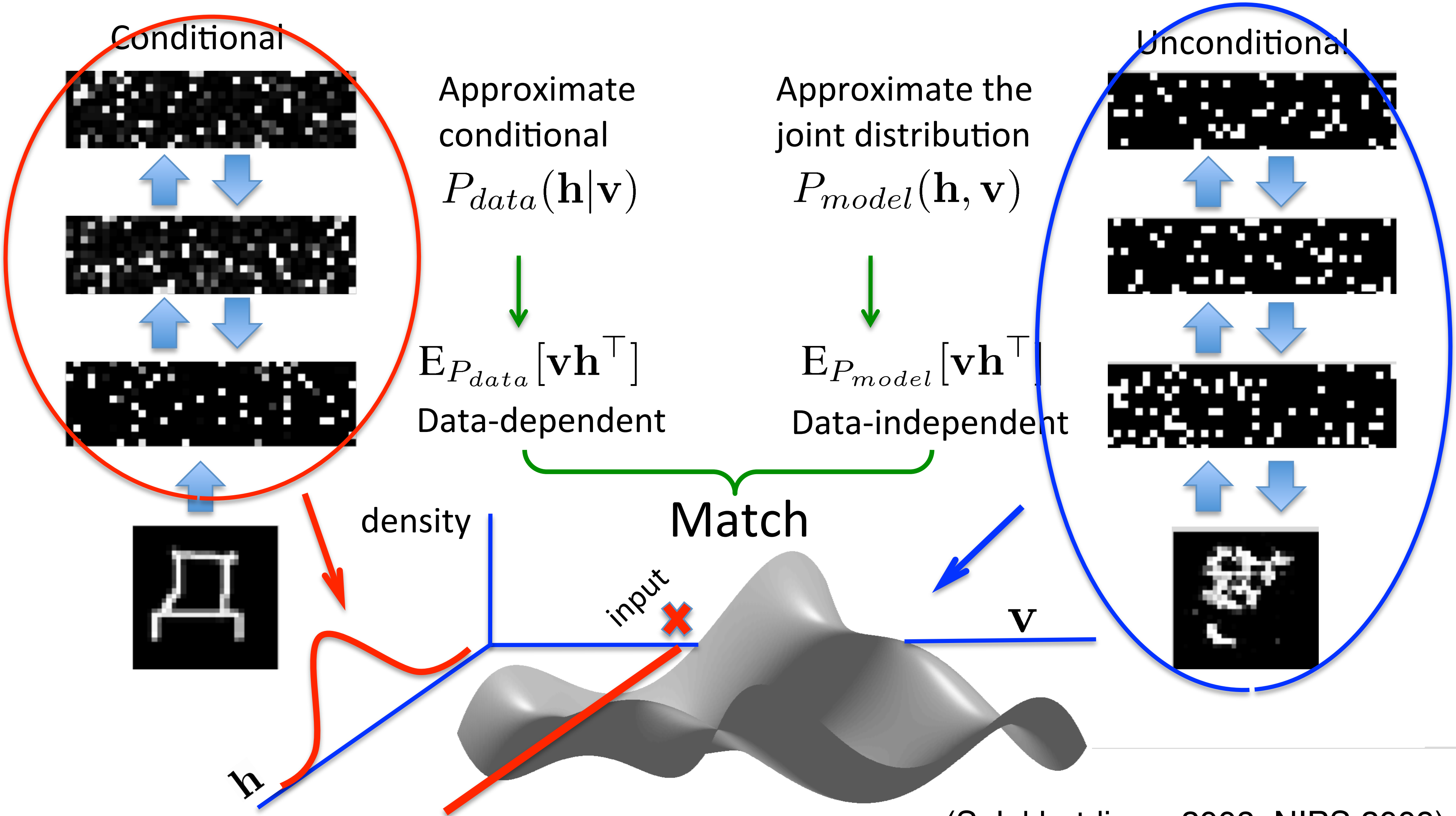


(Salakhutdinov, 2008; NIPS 2009)

# New Learning Algorithm

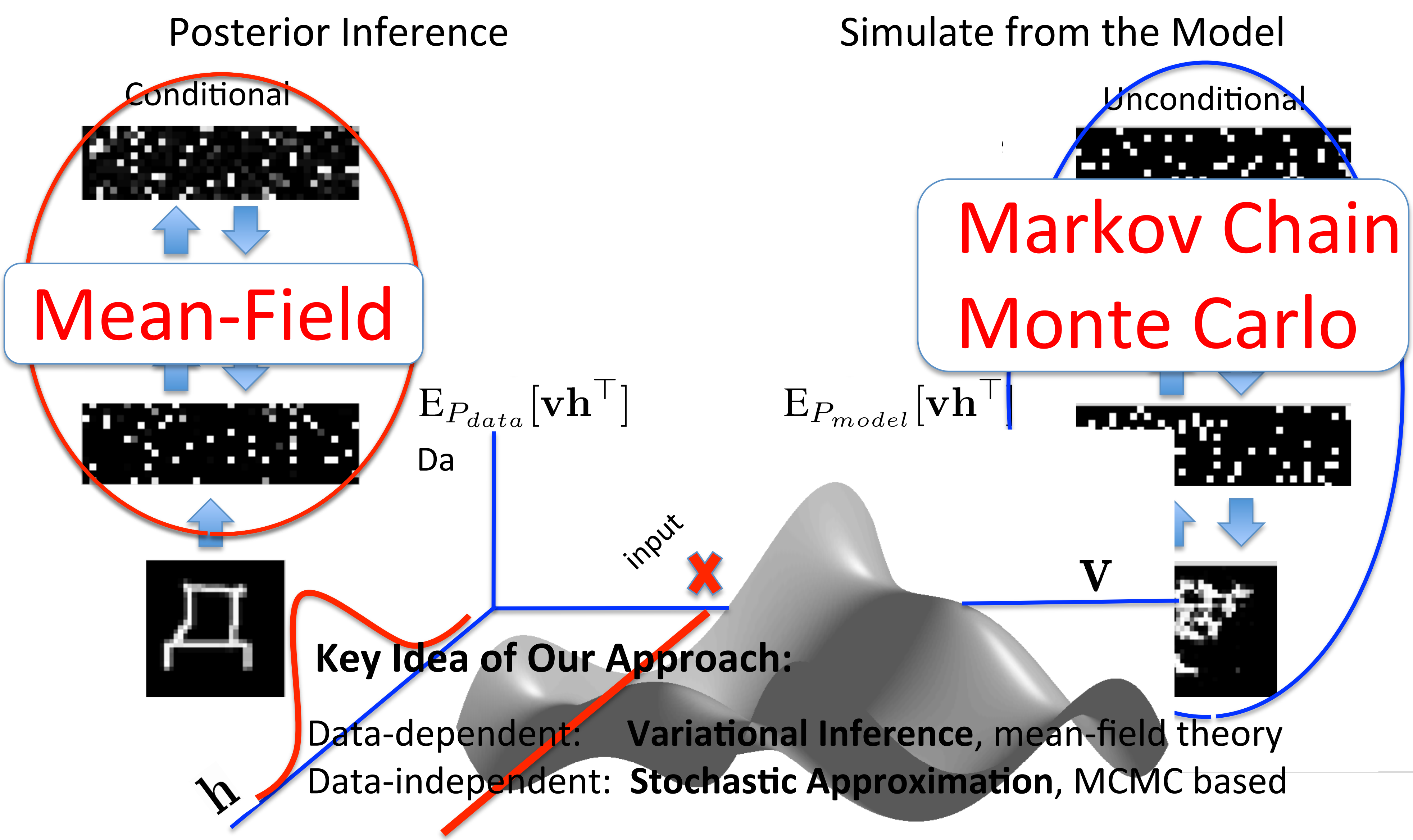
Posterior Inference

Simulate from the Model




(Salakhutdinov, 2008; NIPS 2009)

# New Learning Algorithm



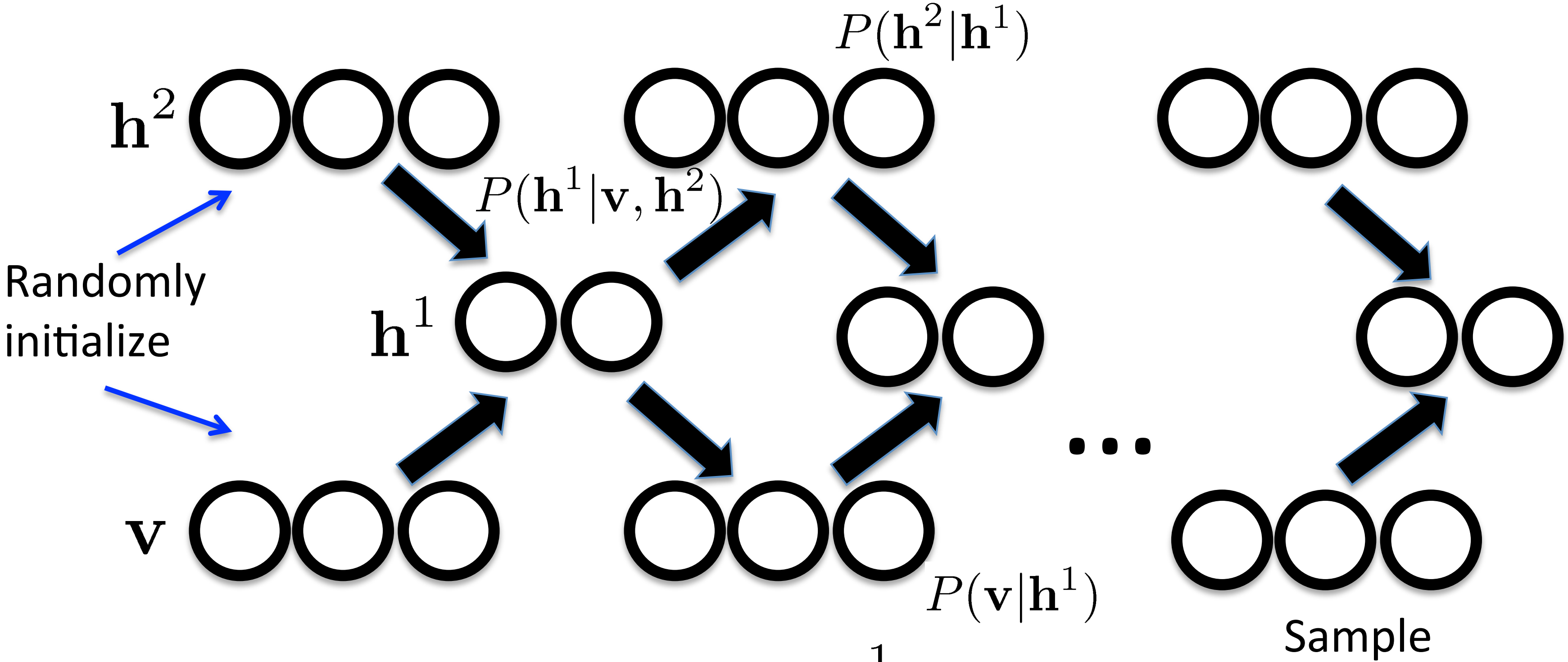

**Key Idea of Our Approach:**

Data-dependent: **Variational Inference**, mean-field theory

Data-independent: **Stochastic Approximation**, MCMC based

# Sampling from DBMs

Sampling from two-hidden layer DBM by running a Markov chain:

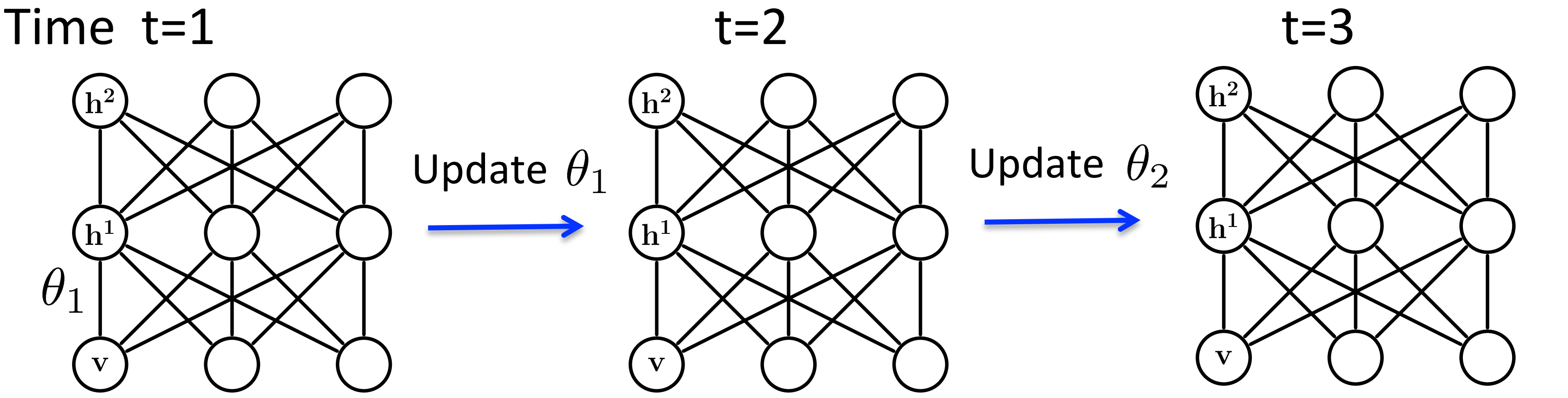



$$P(h^1_m = 1|\mathbf{v},\mathbf{h}^2) = \frac{1}{1+\exp(-\sum_i W^1_{im} v_i - \sum_j W^2_{mj} h^2_j)}$$

$$P(h^2_j = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W^2_{mj} h^1_m)}$$

$$P(v_i = 1|\mathbf{h}^1) = \frac{1}{1+\exp(-\sum_m W^1_{im} h^1_m)}$$

# Stochastic Approximation




$\mathbf{x}_1 \sim \mathrm{T}_{\theta_1}(\mathbf{x}_1 \leftarrow \mathbf{x}_0)$ $\quad$ $\mathbf{x}_2 \sim \mathrm{T}_{\theta_2}(\mathbf{x}_2 \leftarrow \mathbf{x}_1)$ $\quad$ $\mathbf{x}_3 \sim \mathrm{T}_{\theta_3}(\mathbf{x}_3 \leftarrow \mathbf{x}_2)$

Update $\theta_t$ and $\mathbf{x}_t$ sequentially, where $\mathbf{x} = \{\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2\}$

- Generate $\mathbf{x}_t \sim \mathrm{T}_{\theta_t}(\mathbf{x}_t \leftarrow \mathbf{x}_{t-1})$ by simulating from a Markov chain that leaves $P_{\theta_t}$ invariant (e.g. Gibbs or M-H sampler)
- Update $\theta_t$ by replacing intractable $\mathrm{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top]$ with a point estimate $[\mathbf{v}_t\mathbf{h}_t^\top]$

In practice we simulate several Markov chains in parallel.

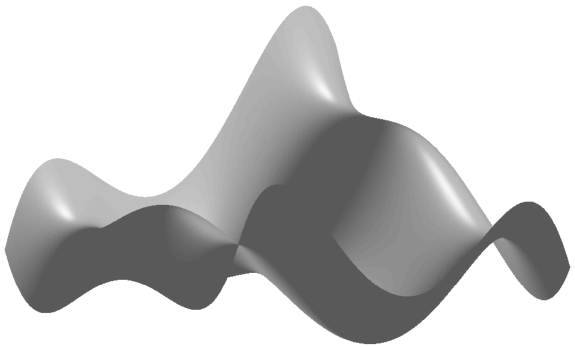

Robbins and Monro, Ann. Math. Stats, 1957
L. Younes, Probability Theory 1989

# Learning Algorithm

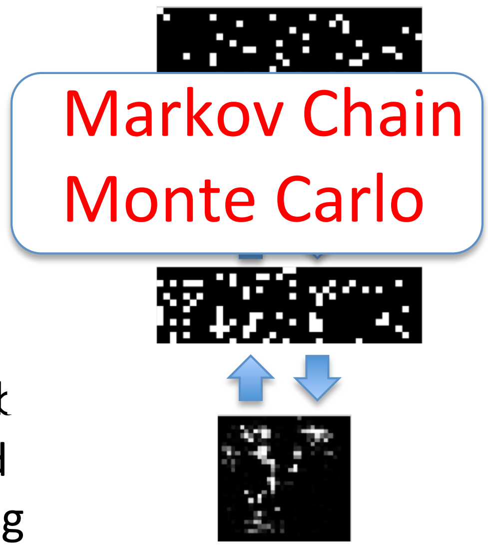
Update rule decomposes:

$$\theta_{t+1} = \theta_t + \alpha_t \underbrace{\left(\mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^\top] - \mathbb{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top]\right)}_{\text{True gradient}} + \alpha_t \underbrace{\left(\mathbb{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top] - \frac{1}{M}\sum_{m=1}^{M} \mathbf{v}_t^{(m)} \mathbf{h}_t^{(m)\top}\right)}_{\text{Perturbation term } \epsilon_t}$$

Almost sure convergence guarantees as learning rate $\alpha_t \to 0$



**Problem:** High-dimensional data: the probability landscape is highly multimodal.




**Key insight:** The transition operator can b
any valid transition operator – Tempered
Transitions, Parallel/Simulated Tempering

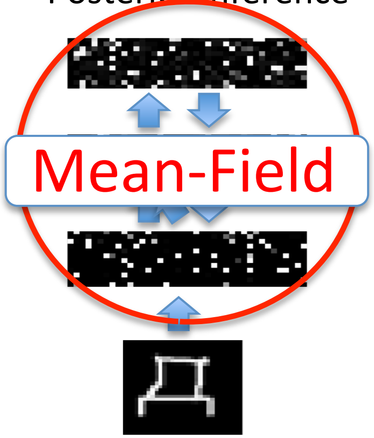
Connections to the theory of stochastic approximation and adaptive MCMC.

# Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\log P_\theta(\mathbf{v}) = \log \sum_{\mathbf{h}} P_\theta(\mathbf{h},\mathbf{v}) = \log \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \frac{P_\theta(\mathbf{h},\mathbf{v})}{Q_\mu(\mathbf{h}|\mathbf{v})}$$

Posterior Inference




$$\geq \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \log \frac{P_\theta(\mathbf{h},\mathbf{v})}{Q_\mu(\mathbf{h}|\mathbf{v})}$$

$$= \underbrace{\sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \underbrace{\log P^*_\theta(\mathbf{h},\mathbf{v})}_{\mathbf{v}^\top W^1 \mathbf{h}^1 + \mathbf{h}^{1\top} W^2 \mathbf{h}^2 + \mathbf{h}^{2\top} W^3 \mathbf{h}^3} - \log \mathcal{Z}(\theta) + \sum_{\mathbf{h}} Q_\mu(\mathbf{h}|\mathbf{v}) \log \frac{1}{Q_\mu(\mathbf{h}|\mathbf{v})}}_{\text{Variational Lower Bound}}$$

$$= \log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v}) || P_\theta(\mathbf{h}|\mathbf{v})\big)$$

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

Minimize KL between approximating and true distributions with respect to variational parameters $\mu$ .

(Salakhutdinov, 2008; Salakhutdinov & Larochelle, AI & Statistics 2010)

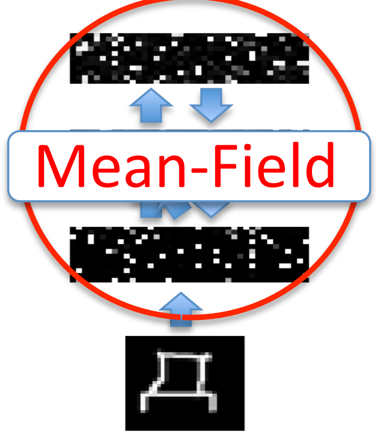
# Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

$$\log P_\theta(\mathbf{v}) \geq \underbrace{\log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)}_{\text{Variational Lower Bound}}$$

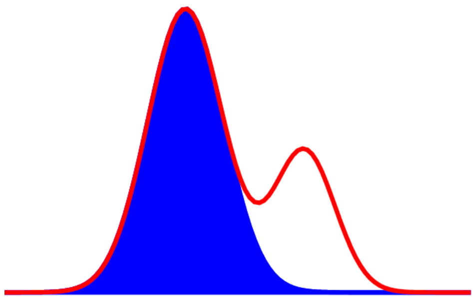
Posterior Inference




**Mean-Field:** Choose a fully factorized distribution:

$$Q_\mu(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{F} q(h_j|\mathbf{v}) \quad \text{with} \quad q(h_j = 1|\mathbf{v}) = \mu_j$$

**Variational Inference:** Maximize the lower bound w.r.t. Variational parameters $\mu$.



Nonlinear fixed-point equations:

$$\mu_j^{(1)} = \sigma\Big(\sum_i W_{ij}^1 v_i + \sum_k W_{jk}^2 \mu_k^{(2)}\Big)$$
$$\mu_k^{(2)} = \sigma\Big(\sum_j W_{jk}^2 \mu_j^{(1)} + \sum_m W_{km}^3 \mu_m^{(3)}\Big)$$
$$\mu_m^{(3)} = \sigma\Big(\sum_k W_{km}^3 \mu_k^{(2)}\Big)$$

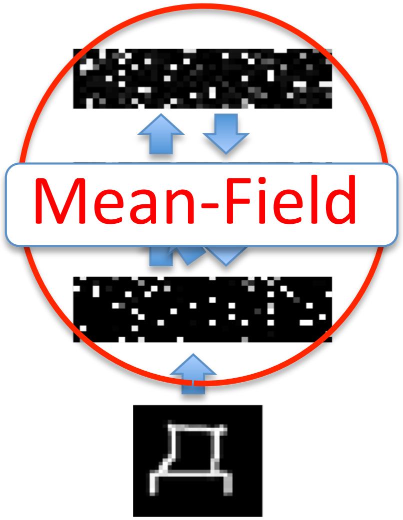
# Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

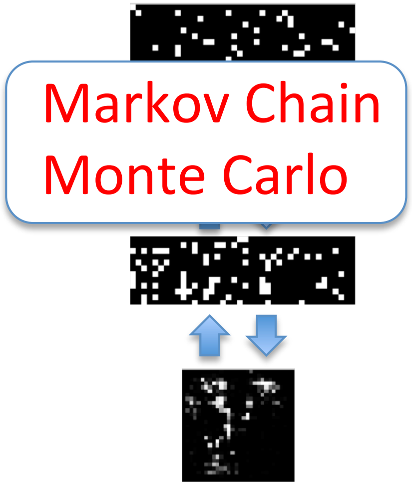
$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

$$\log P_\theta(\mathbf{v}) \geq \underbrace{\log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)}_{\text{Variational Lower Bound}}$$

Posterior Inference




Unconditional Simulation




**1. Variational Inference:** Maximize the lower bound w.r.t. variational parameters

**2. MCMC:** Apply stochastic approximation to update model parameters

Almost sure convergence guarantees to an asymptotically stable point.

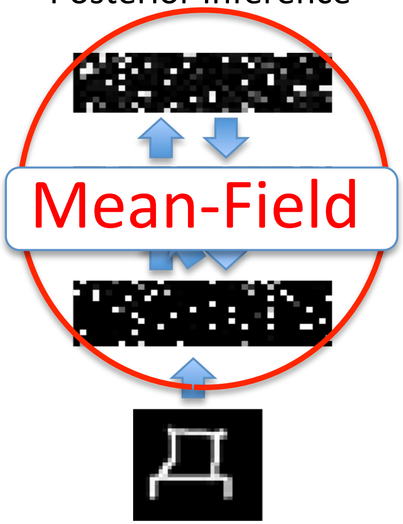
# Variational Inference

Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

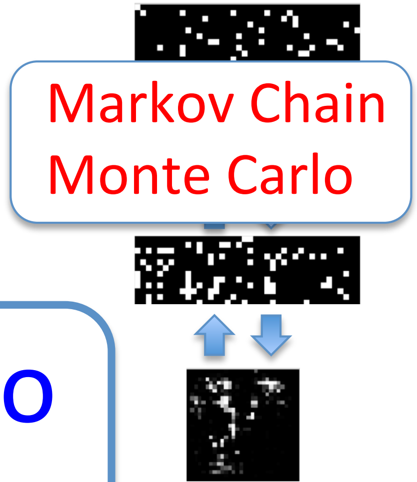
$$\mathrm{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

$$\log P_\theta(\mathbf{v}) \geq \underbrace{\log P_\theta(\mathbf{v}) - \mathrm{KL}(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v}))}_{\text{Variational Lower Bound}}$$

Posterior Inference




Unconditional Simulation




**1. V** wer bou

Fast Inference

**2. M** to u

Learning can scale to millions of examples

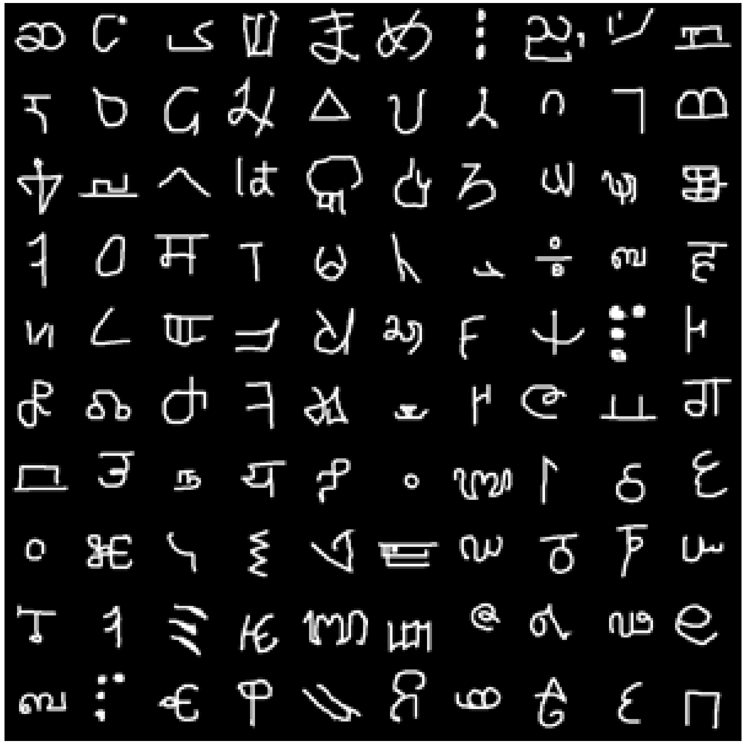
Almost sure convergence guarantees to an asymptotically stable point.

# Good Generative Model?

Handwritten Characters

# Good Generative Model?

Handwritten Characters



# Good Generative Model?

Handwritten Characters

Simulated

Real Data

# Good Generative Model?

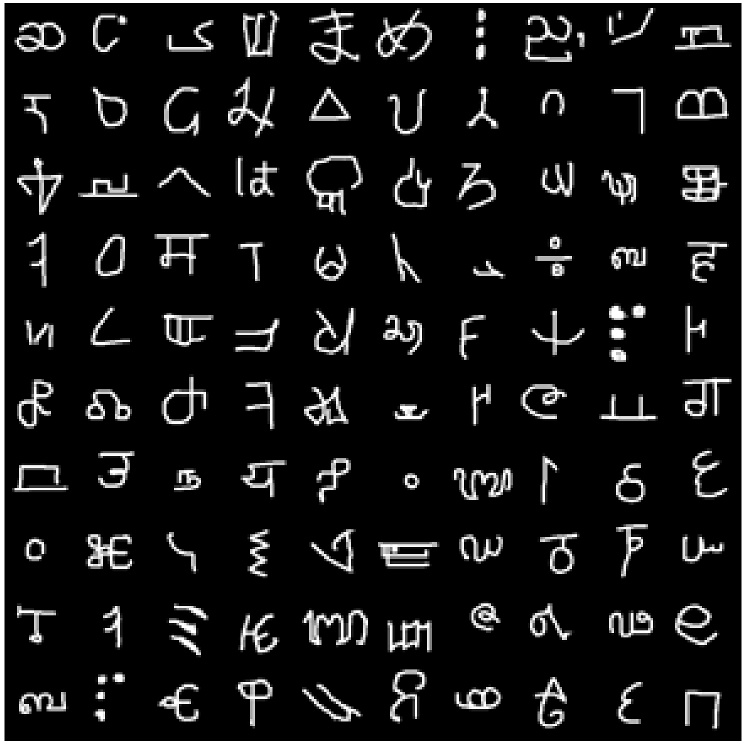
Handwritten Characters

Real Data

Simulated

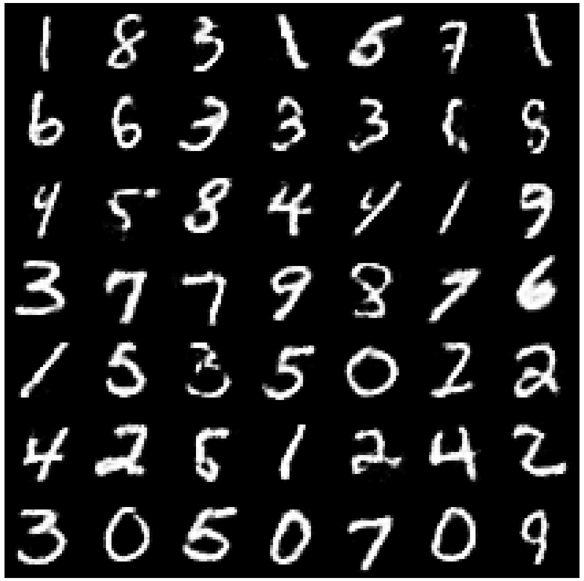
# Good Generative Model?

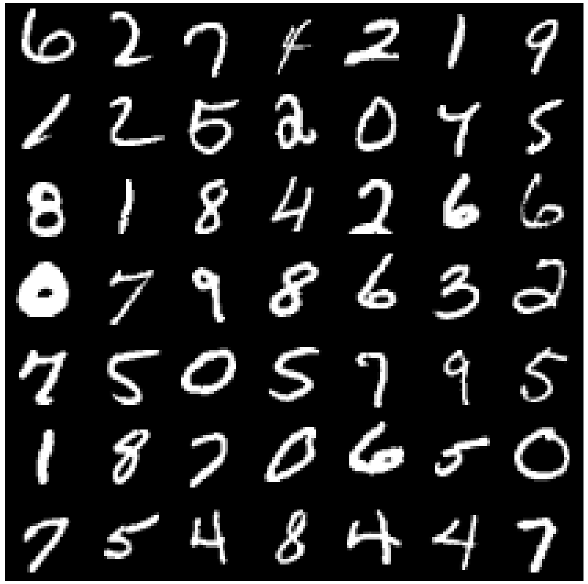
Handwritten Characters



# Good Generative Model?

MNIST Handwritten Digit Dataset





# Handwriting Recognition

MNIST Dataset
60,000 examples of 10 digits

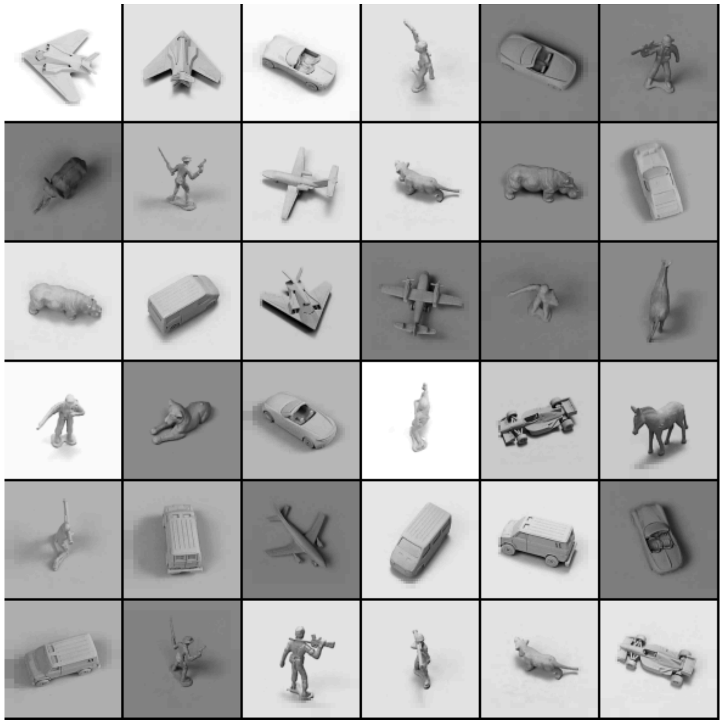
| Learning Algorithm | Error |
|---|---|
| Logistic regression | 12.0% |
| K-NN | 3.09% |
| Neural Net (Platt 2005) | 1.53% |
| SVM (Decoste et.al. 2002) | 1.40% |
| Deep Autoencoder (Bengio et. al. 2007) | 1.40% |
| Deep Belief Net (Hinton et. al. 2006) | 1.20% |
| **DBM** | **0.95%** |

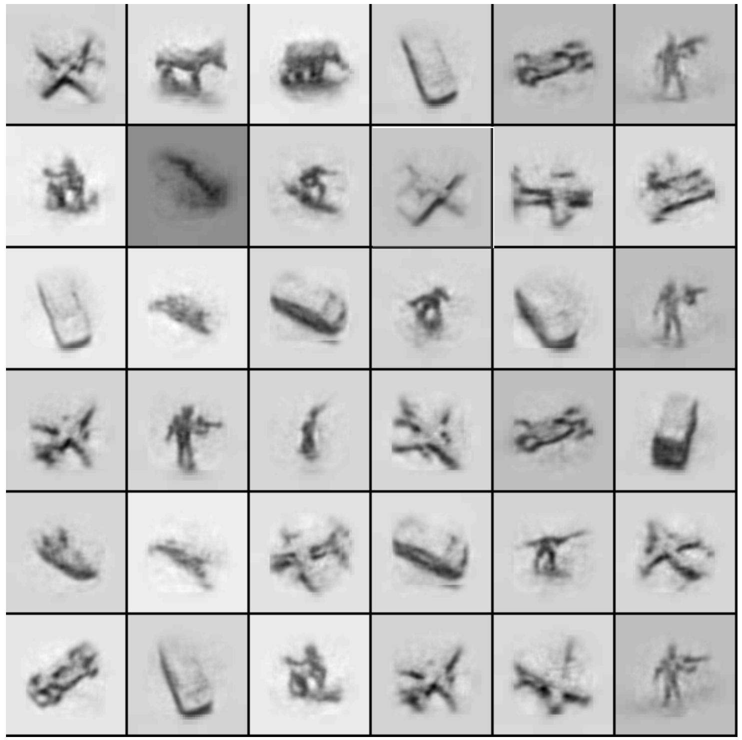
Optical Character Recognition
42,152 examples of 26 English letters

| Learning Algorithm | Error |
|---|---|
| Logistic regression | 22.14% |
| K-NN | 18.92% |
| Neural Net | 14.62% |
| SVM (Larochelle et.al. 2009) | 9.70% |
| Deep Autoencoder (Bengio et. al. 2007) | 10.05% |
| Deep Belief Net (Larochelle et. al. 2009) | 9.68% |
| **DBM** | **8.40%** |

Permutation-invariant version.

# Generative Model of 3-D Objects



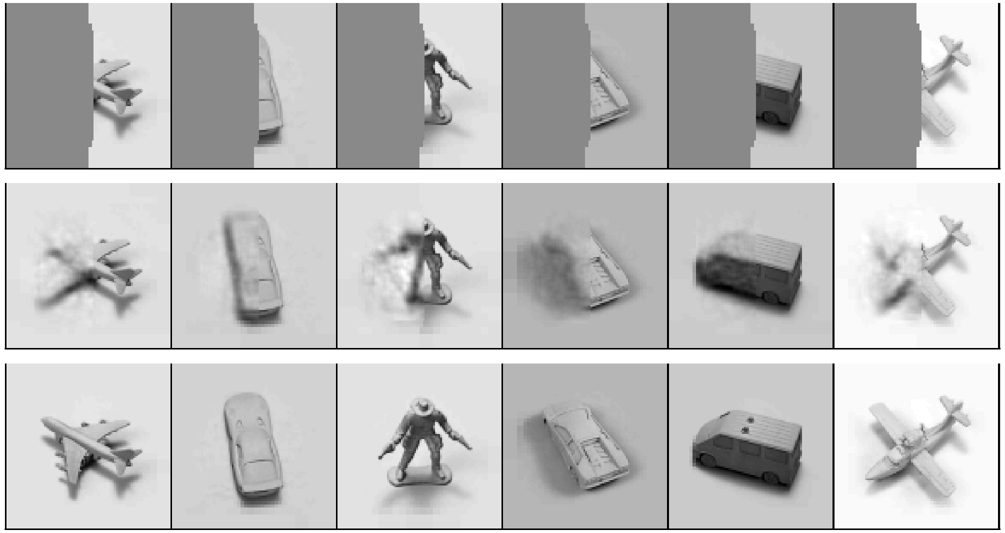


24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.

# 3-D Object Recognition

| Learning Algorithm | Error |
|---|---|
| Logistic regression | 22.5% |
| K-NN (LeCun 2004) | 18.92% |
| SVM (Bengio & LeCun 2007) | 11.6% |
| Deep Belief Net (Nair & Hinton 2009) | 9.0% |
| **DBM** | **7.2%** |

Pattern Completion

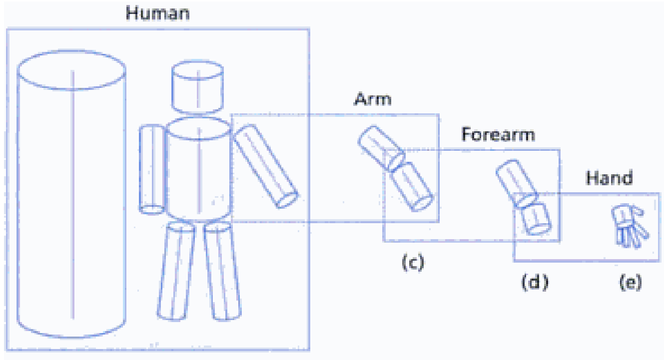


Permutation-invariant version.

# Learning Hierarchical Representations

Deep Boltzmann Machines:

Learning Hierarchical Structure in Features: edges, combination of edges.

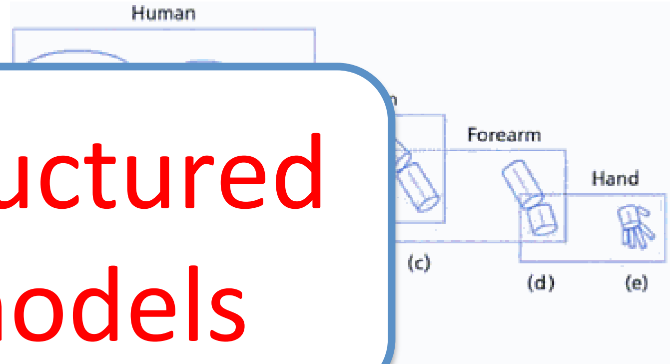



- Performs well in many application domains
- Fast Inference: fraction of a second
- Learning scales to millions of examples

# Learning Hierarchical Representations

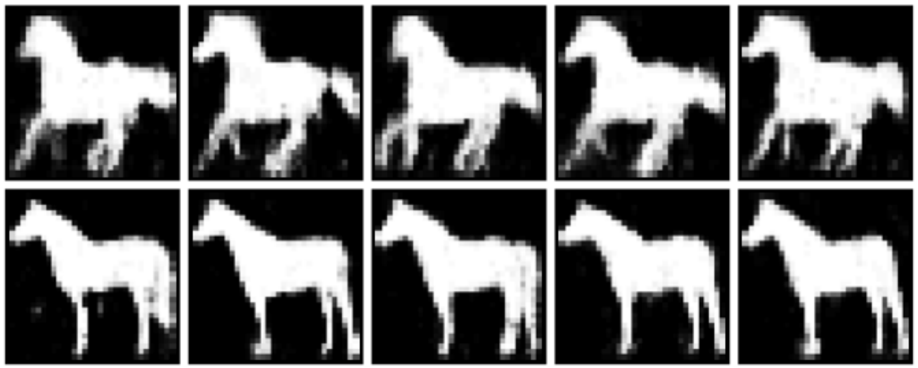
Deep Boltzmann Machines:

Learning Hi
in Features
of edges.




Need more structured
and robust models

**The Shape Boltzmann Machine: a Strong Model of Object Shape** (Eslami, Heess, Winn, CVPR 2012).



[Demo DBM]

**Hallucinations in Charles Bonnet Syndrome Induced by Homeostasis: a Deep Boltzmann Machine Model** (Reichert, Series, Storkey, NIPS 2012)