Deep Learning

Russ Salakhutdinov

Deep Learning Summer School

Department of Computer Science
University of Toronto → CMU
Canadian Institute for Advanced Research

Current Student and Postdocs

PhD Students

Lei Jimmy Ba

Ryan Kiros

Chris Maddison

Nitish Srivastava

Charlie Tang

Postdocs

Yura Burda

Roger Grosse

Master Students

Shikhar Sharma

Yukun Zhu

Undergrads

Emilio Parisotto

Elman Mansimov

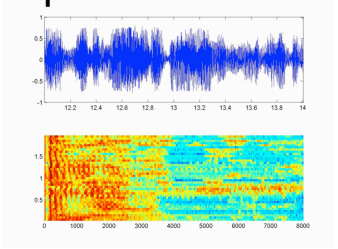
Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.

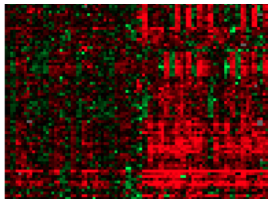
Images & Video

Text & Language

Speech & Audio

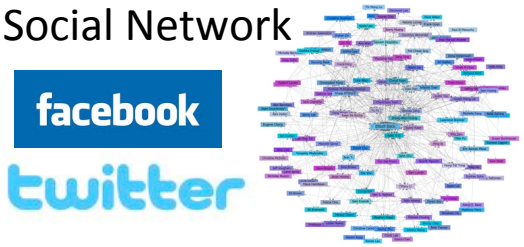


Gene Expression

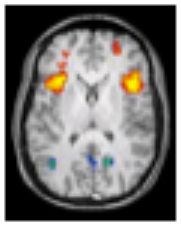


Product Recommendation

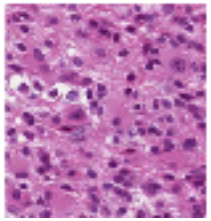
Relational Data/ Social Network



fMRI



Tumor region



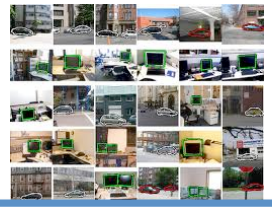
Mostly Unlabeled

- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.

Images & Video

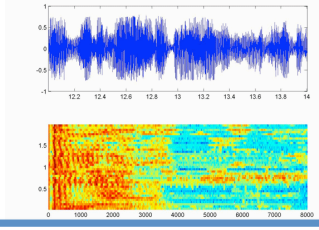


Text & Language

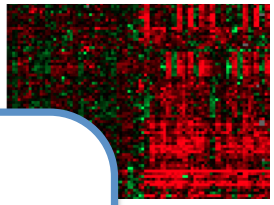
REUTERS

Associated Press

Speech & Audio



Gene Expression



Product

Recomm

ama

NETFLI

Deep Learning Models that support inferences and discover structure at multiple levels.

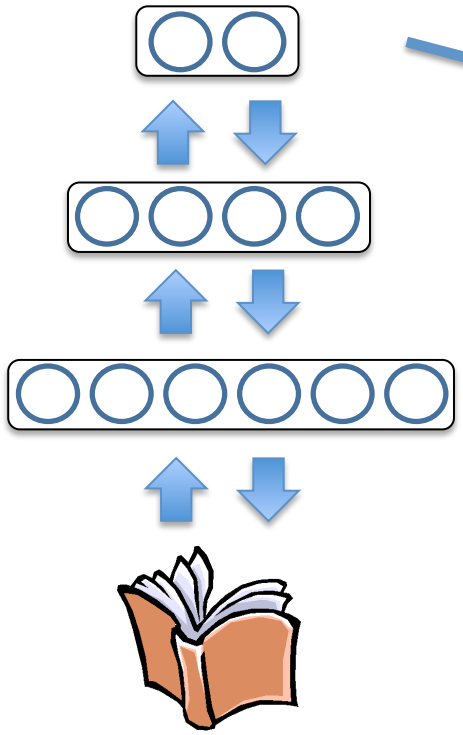
on

Mostly Unlabeled

- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

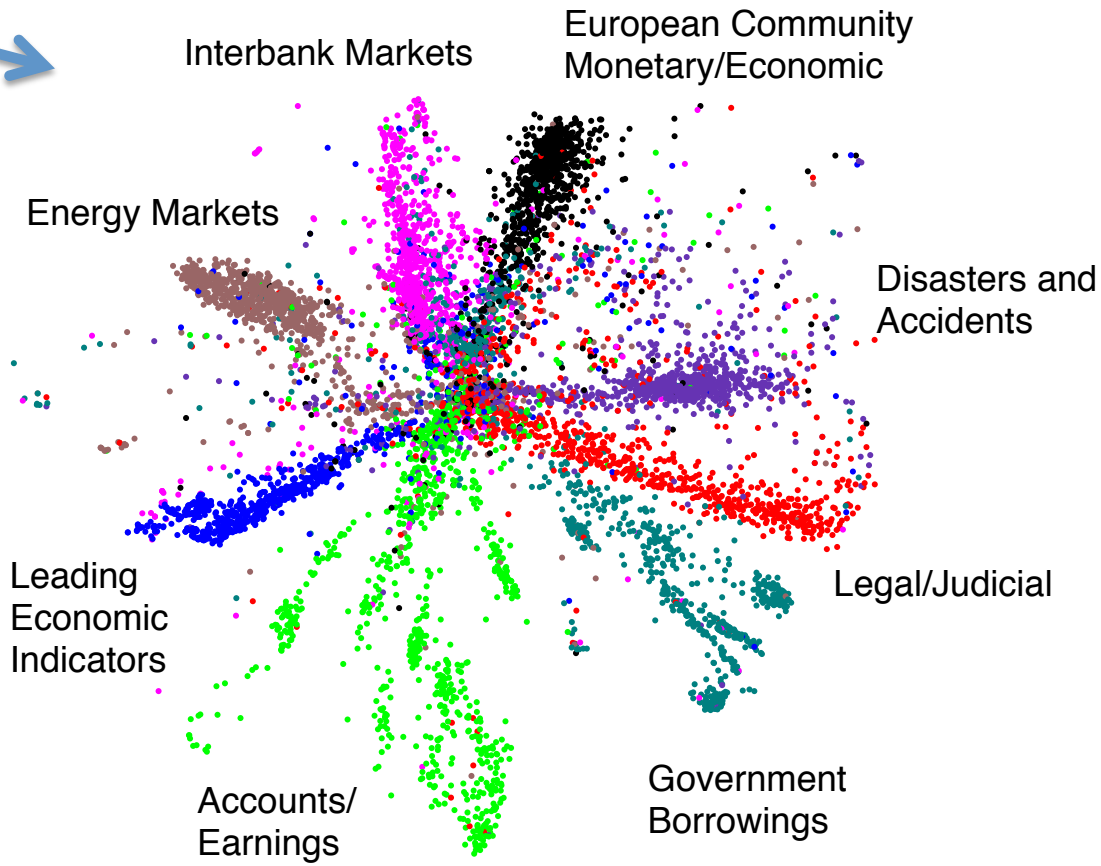
Deep Generative Model

Model P(document)



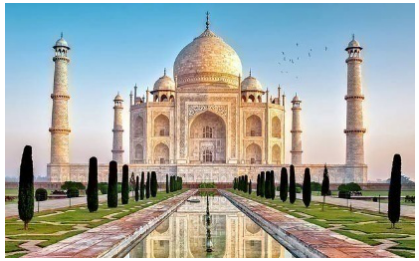
Bag of words

Reuters dataset: 804,414
newswire stories: **unsupervised**



(Hinton & Salakhutdinov, Science 2006)

Multimodal Data



mosque, tower, building, cathedral, dome, castle

ski, skiing, skiers, skiiers, snowmobile

kitchen, stove, oven, refrigerator, microwave

bowl, cup, soup, cups, coffee

beach

snow

Example: Understanding Images

TAGS:

strangers, coworkers, conventioneers, attendants, patrons

Nearest Neighbor Sentence:

people taking pictures of a crazy person

Model Samples

- a group of people in a crowded area .
- a group of people are walking and talking .
- a group of people, standing around and talking .
- a group of people that are in the outside .

Caption Generation

a car is parked in
the middle of nowhere .

a wooden table and chairs
arranged in a room .

there is a cat sitting on a shelf .

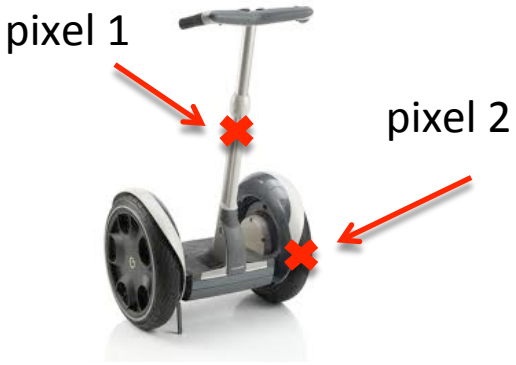
a ferry boat on a marina
with a group of people .

a little boy with a bunch
of friends on the street .

Talk Roadmap

- Learning Deep Models
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines
- Multi-Modal Learning with DBMs
- Evaluating Deep Generative Models

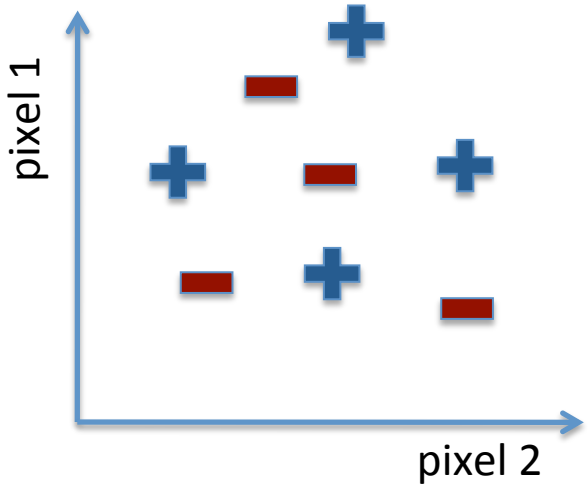
Learning Feature Representations



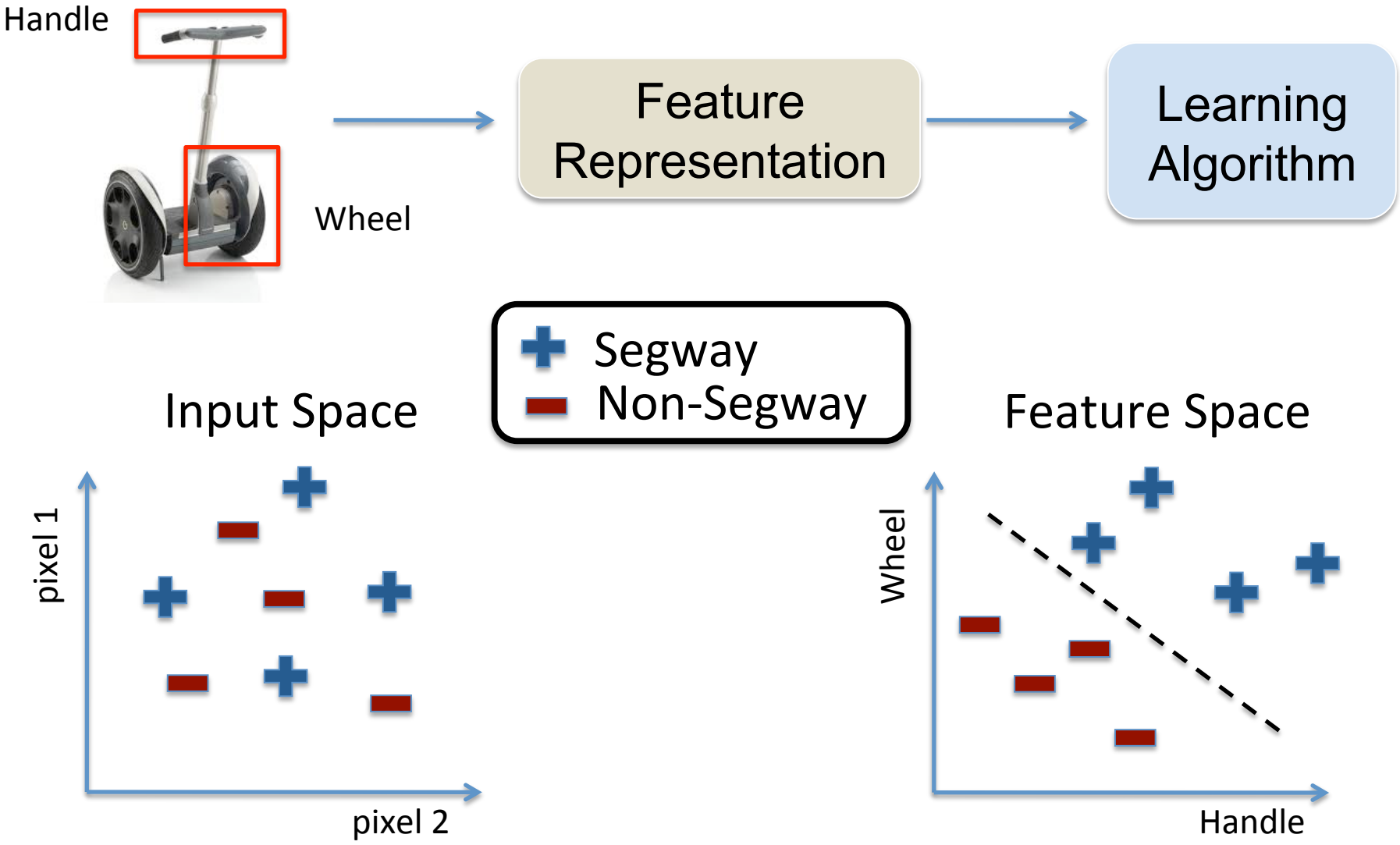
Learning Algorithm

Segway
Non-Segway

Input Space



Learning Feature Representations

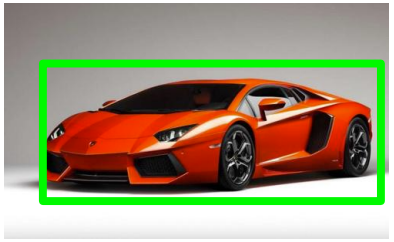


Traditional Approaches

Object detection

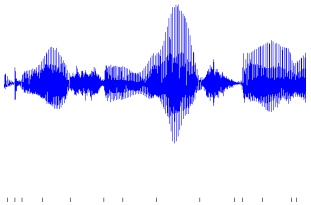
Image

vision features

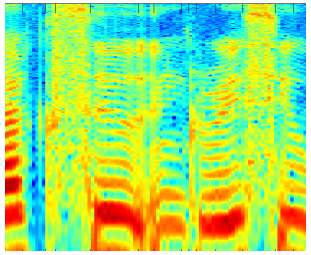


Recognition

Audio classification



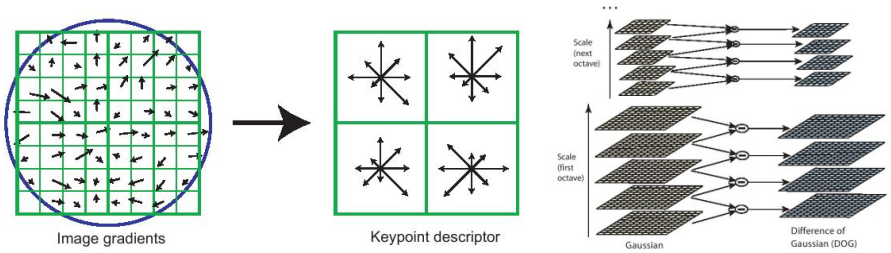
Audio



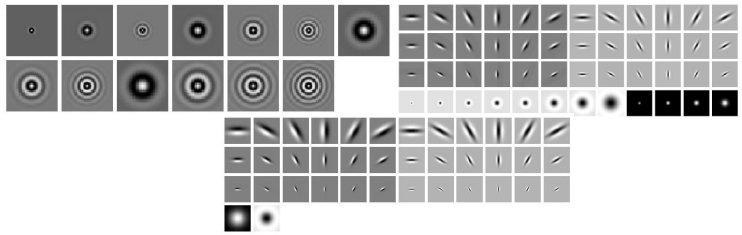
audio features

Speaker identification

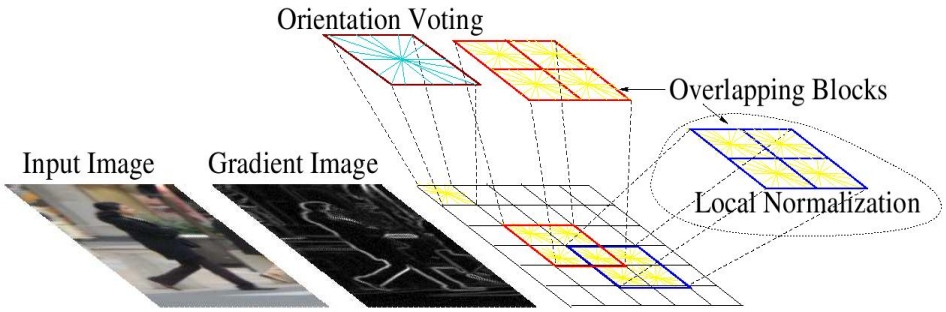
Computer Vision Features



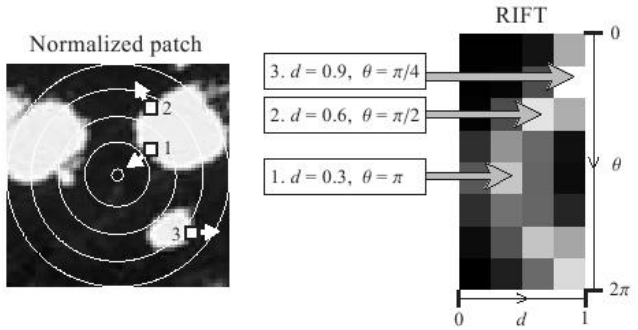
SIFT



Textons

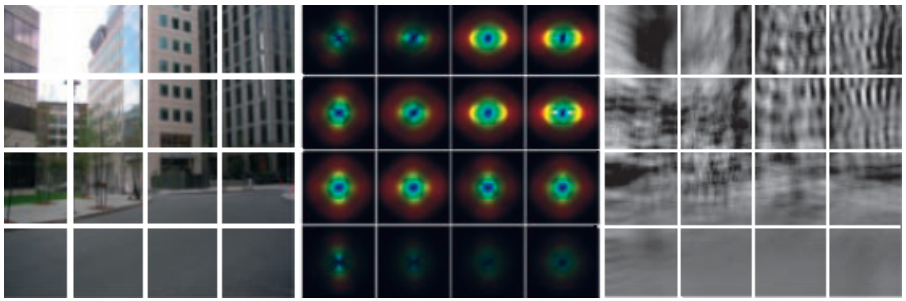


HoG



RIFT

GIST



Computer Vision Features

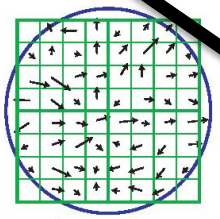
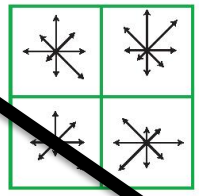
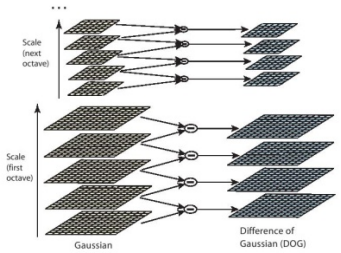


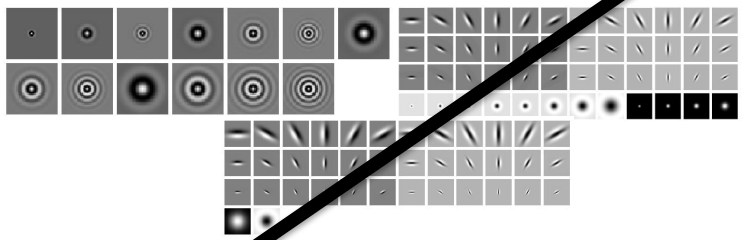
Image gradients



Keypoint descriptor

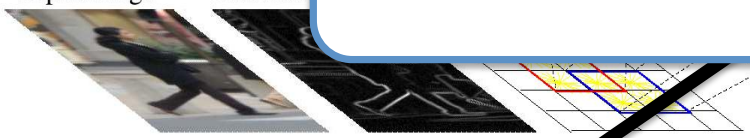


SIFT

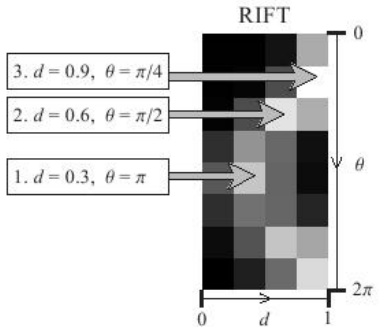


Textons

Deep Learning

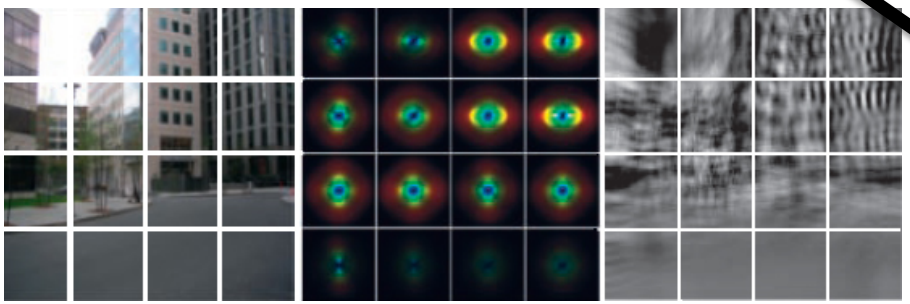


HoG

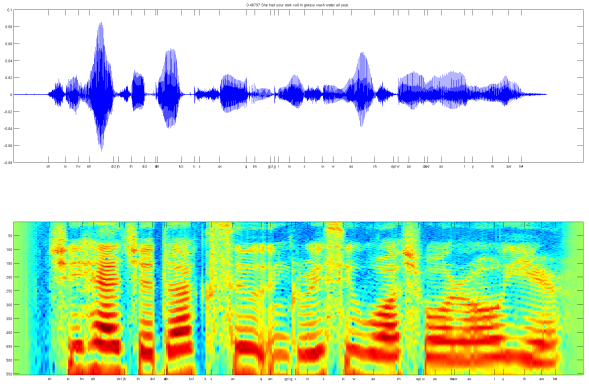


RIFT

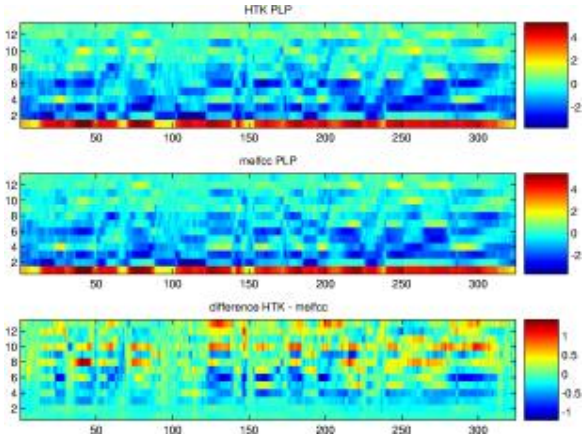
GIST



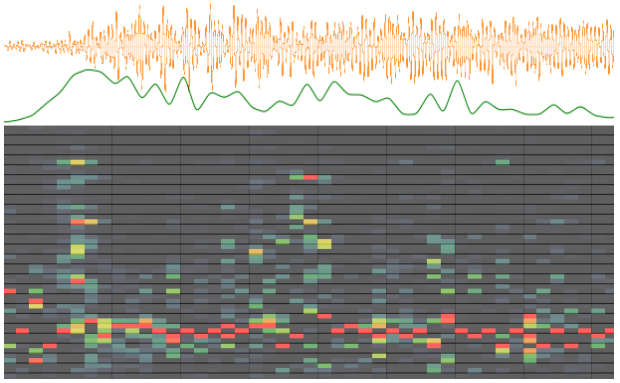
Audio Features



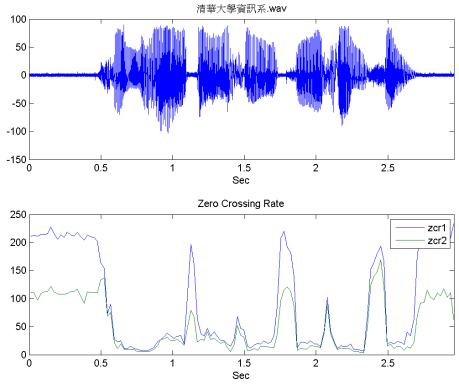
Spectrogram



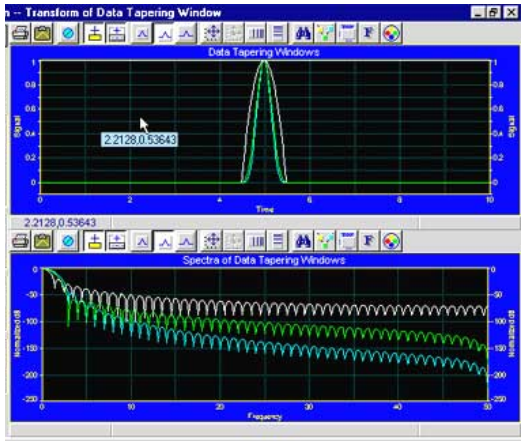
MFCC



Flux

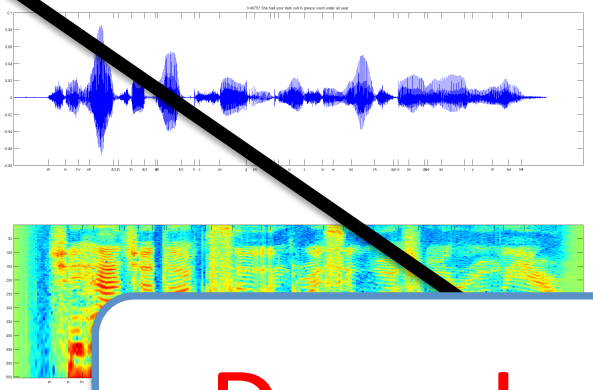


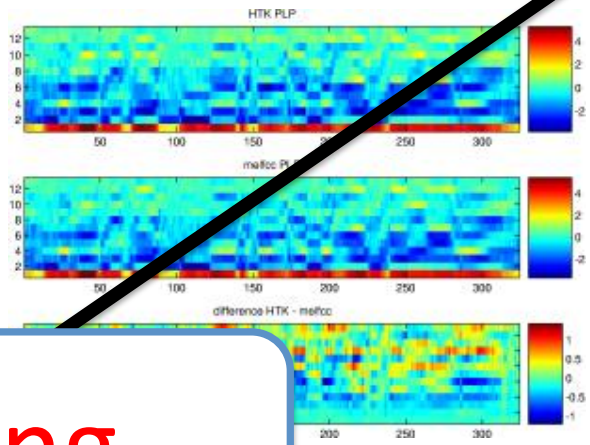
ZCR



Rolloff

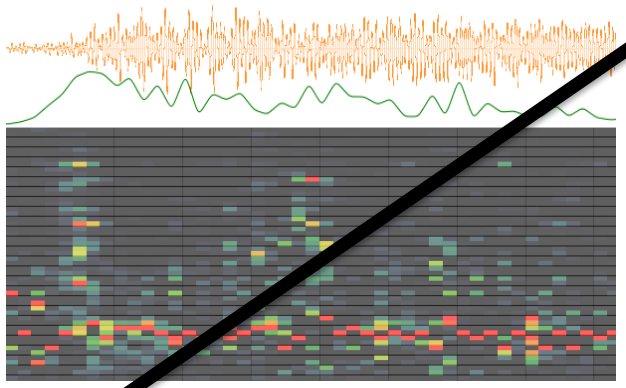
Audio Features



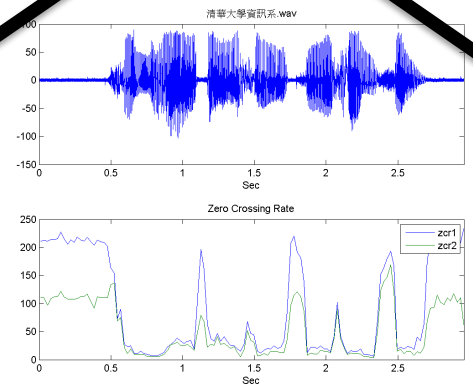


MFCC

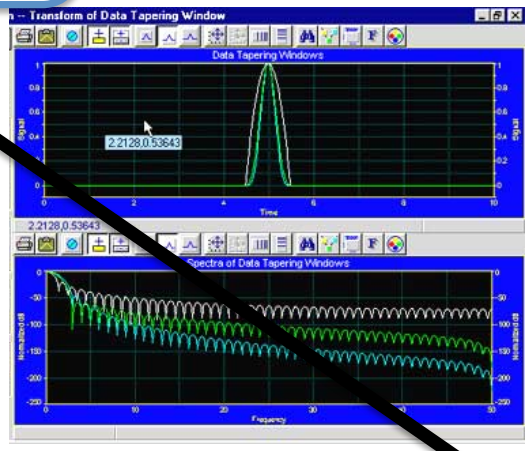
Deep Learning



Flux

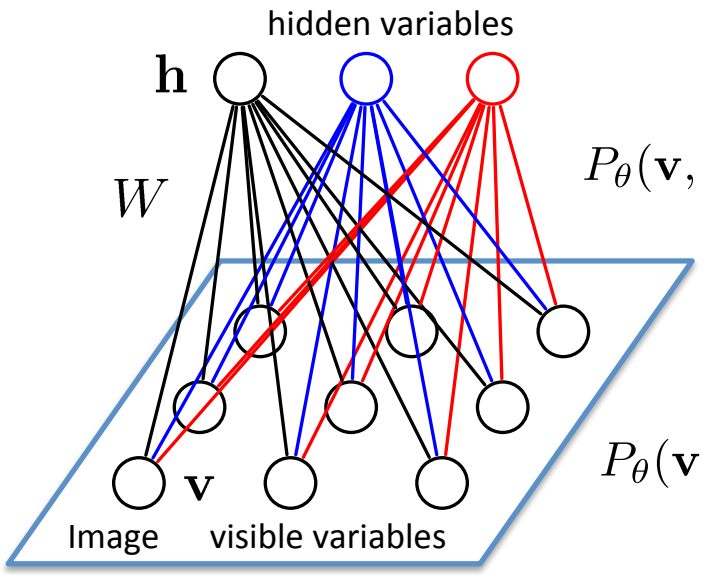


ZCR



Rolloff

Restricted Boltzmann Machines



Pair-wise Unary

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

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

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

RBM is a Markov Random Field with:

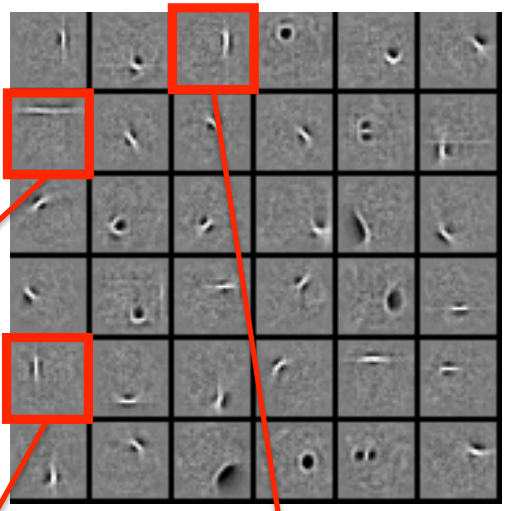
- Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$.
- Stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.
- Bipartite connections.

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

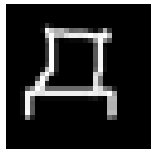
Learning Features

Observed Data
Subset of 25,000 characters

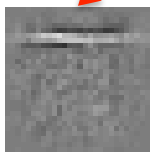
Learned W: "edges"
Subset of 1000 features



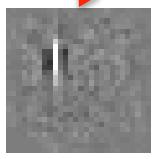
New Image:



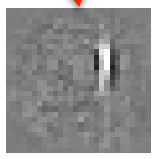
$$= \sigma\left(0.99 \times \right.$$



$$+ \quad 0.97 \times$$



$$+ \quad 0.82 \times$$



$$\left. \cdots \right)$$

$p(h_7 = 1|v)$

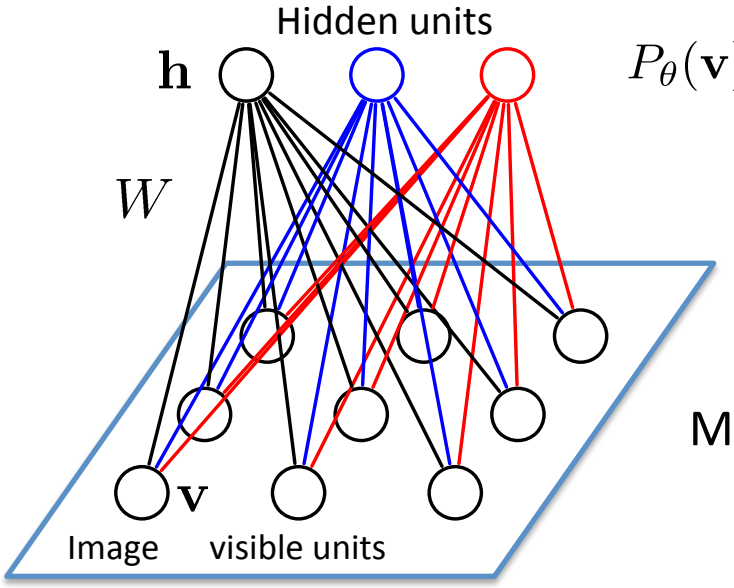
$p(h_{29} = 1|v)$

Sparse representations

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Logistic Function: Suitable for modeling binary images

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:

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

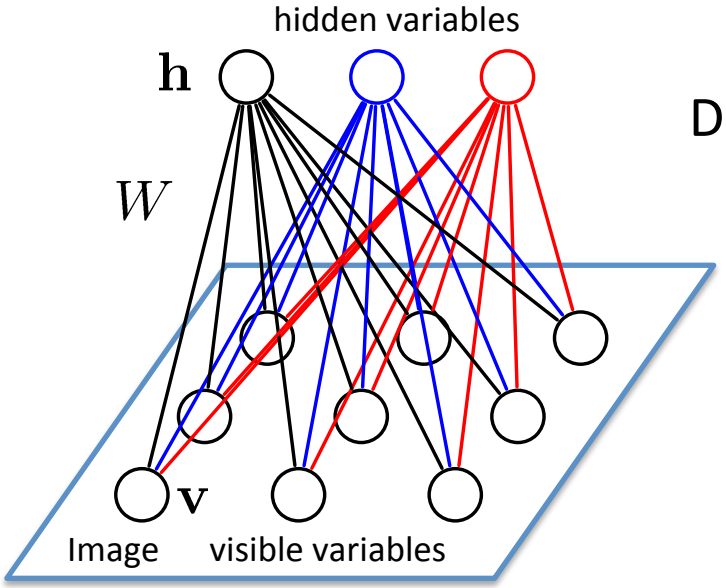
$$= \mathrm{E}_{P_{data}}[v_i h_j] - \underbrace{\mathrm{E}_{P_\theta}[v_i h_j]}$$

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.

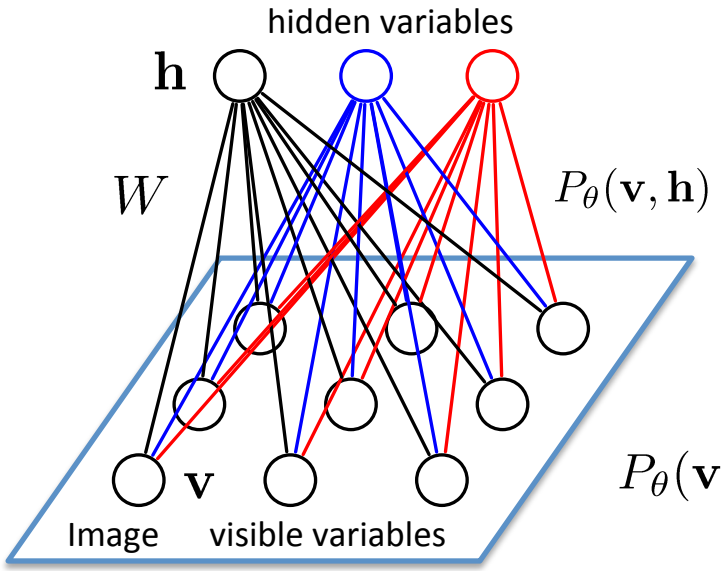
Use MCMC

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

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

Approximate maximum likelihood learning

RBMs for Real-valued Data



$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} + \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{F} a_j h_j \right)$$

Pair-wise

Unary

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

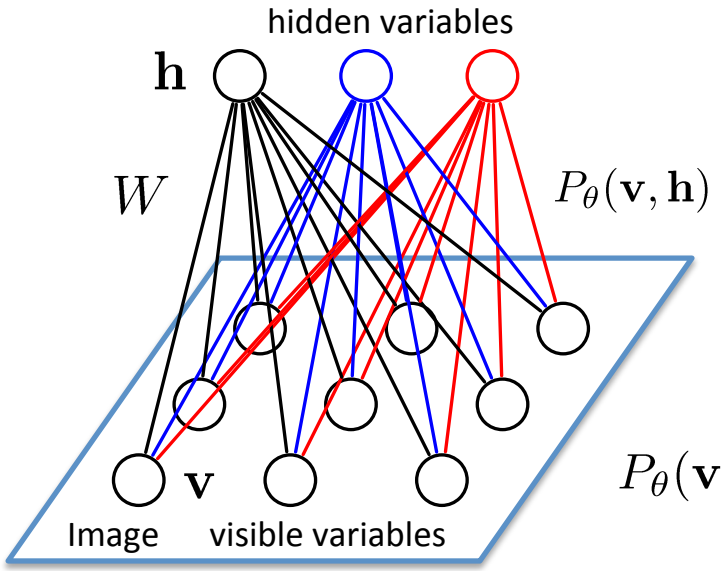
$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \mathcal{N}\left(b_i + \sum_{j=1}^{F} W_{ij} h_j, \sigma_i^2 \right)$$

Gaussian-Bernoulli RBM:

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

(Salakhutdinov & Hinton, NIPS 2007; Salakhutdinov & Murray, ICML 2008)

RBMs for Real-valued Data



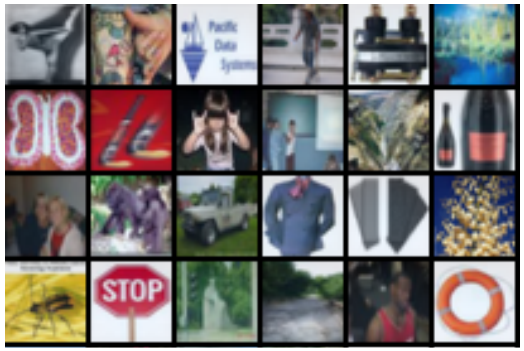
Pair-wise **Unary**

$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} + \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{F} a_j h_j\right)$$

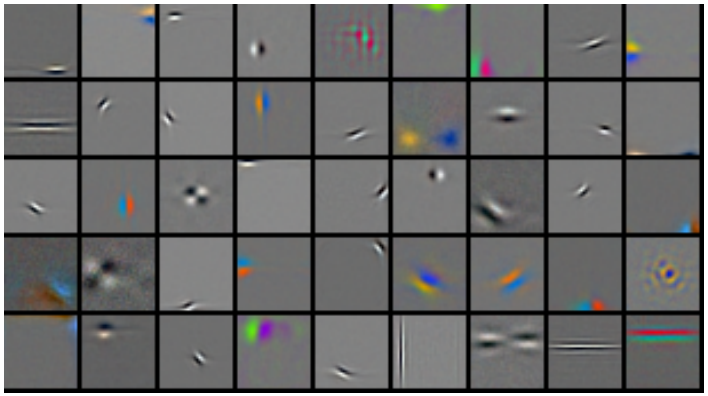
$$\theta = \{W, a, b\}$$

$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \mathcal{N}\left(b_i + \sum_{j=1}^{F} W_{ij} h_j, \sigma_i^2\right)$$

4 million **unlabelled** images



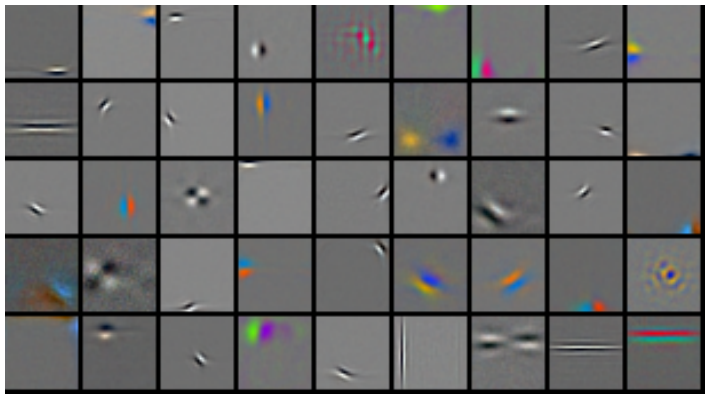
Learned features (out of 10,000)



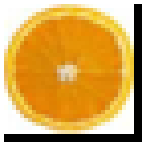
RBMs for Real-valued Data

4 million **unlabelled** images

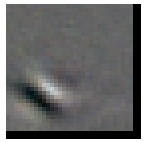
Learned features (out of 10,000)



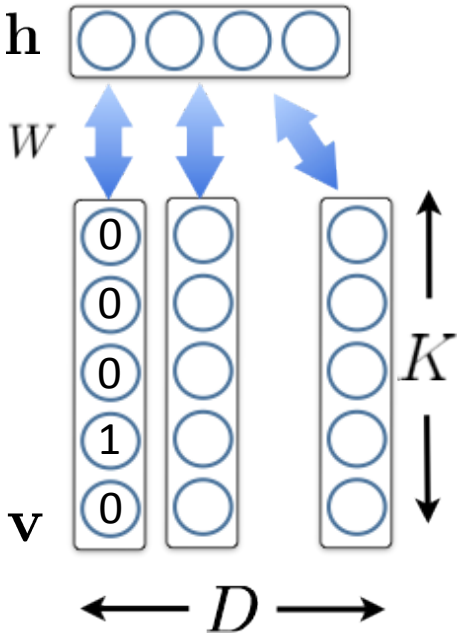
New Image



$p(h_7 = 1|v)$ $p(h_{29} = 1|v)$

$= 0.9 *$  $+ 0.8 *$  $+ 0.6 *$  $\ldots$

RBMs for Word Counts



Pair-wise Unary

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

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

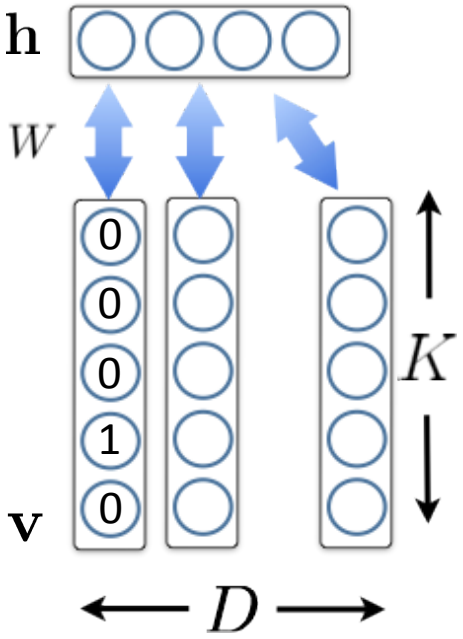
$$P_\theta(v_i^k = 1 | \mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k\right)}{\sum_{q=1}^{K} \exp\left(b_i^q + \sum_{j=1}^{F} h_j W_{ij}^q\right)}$$

Replicated Softmax Model: undirected topic model:

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

(Salakhutdinov & Hinton, NIPS 2010, Srivastava & Salakhutdinov, NIPS 2012)

RBMs for Word Counts



Pair-wise Unary

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

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

$$P_\theta(v_i^k = 1|\mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k\right)}{\sum_{q=1}^{K} \exp\left(b_i^q + \sum_{j=1}^{F} h_j W_{ij}^q\right)}$$

WIKIPEDIA The Free Encyclopedia

REUTERS

AP Associated Press

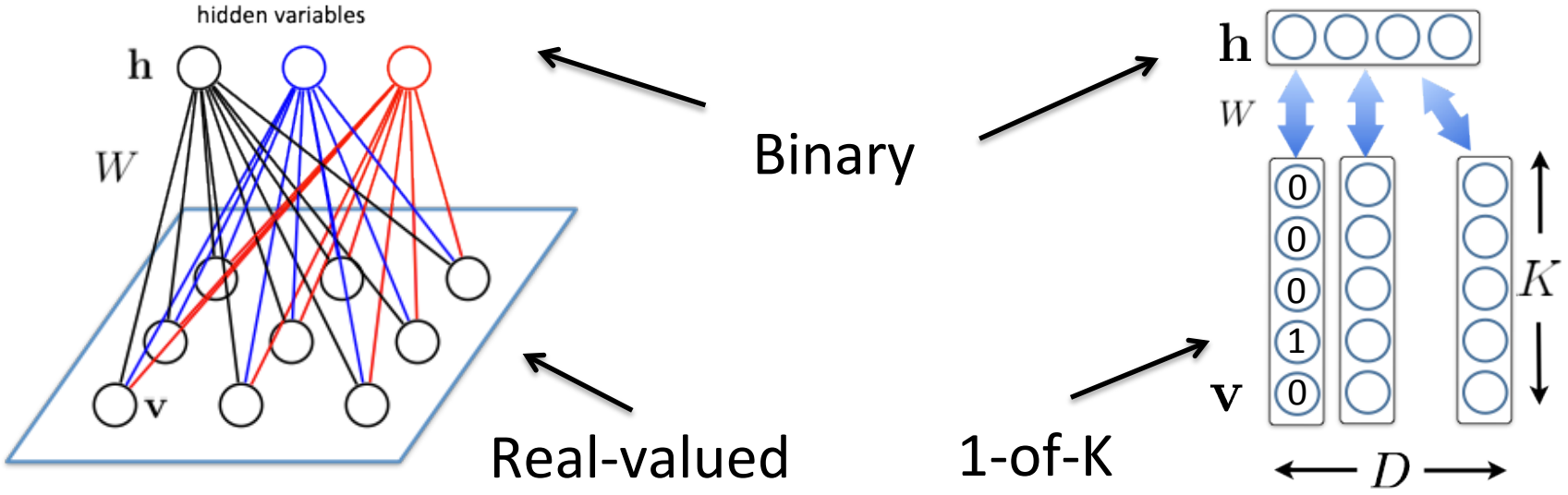
Reuters dataset:
804,414 **unlabeled**
newswire stories
Bag-of-Words

Learned features: ``topics''

russian	clinton	computer	trade	stock
russia	house	system	country	wall
moscow	president	product	import	street
yeltsin	bill	software	world	point
soviet	congress	develop	economy	dow

Different Data Modalities

- Binary/Gaussian/Softmax RBMs: All have binary hidden variables but use them to model different kinds of data.



- It is easy to infer the states of the hidden variables:

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

Product of Experts

The joint distribution is given by:

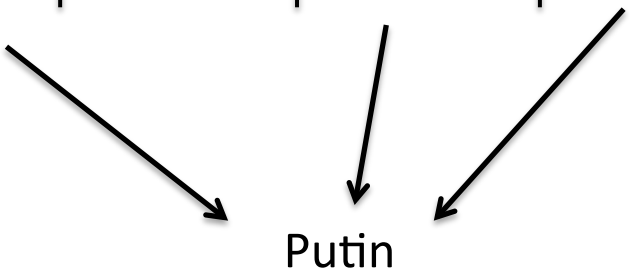
$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\right)$$

Marginalizing over hidden variables:

Product of Experts

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}} P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \prod_i \exp(b_i v_i) \prod_j \left(1 + \exp(a_j + \sum_i W_{ij} v_i)\right)$$

government auhority power empire putin	clinton house president bill congress	bribery corruption dishonesty putin fraud	oil barrel exxon putin drill	stock wall street point dow	…



Topics “government”, ”corruption” and ”oil” can combine to give very high probability to a word “Putin”.

Product of Experts

The joint distribution is given by:

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\right)$$

Marginalizing o

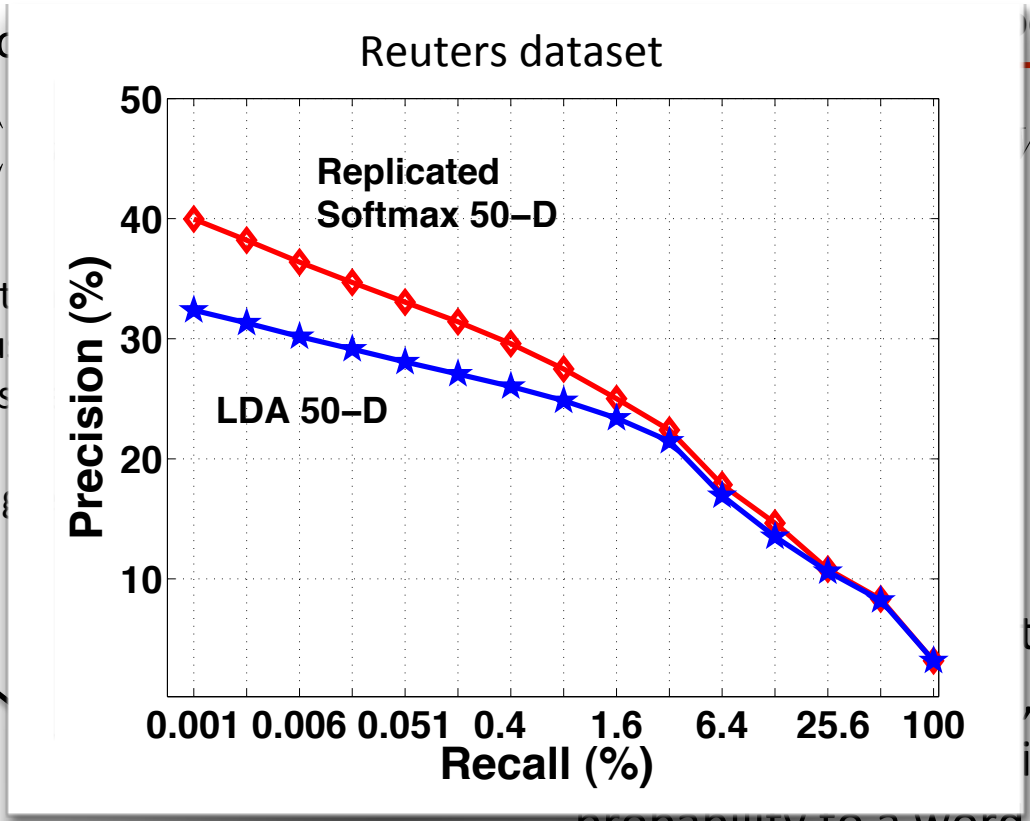
$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}}$$

duct of Experts

$$W_{ij} v_i)\Big)$$

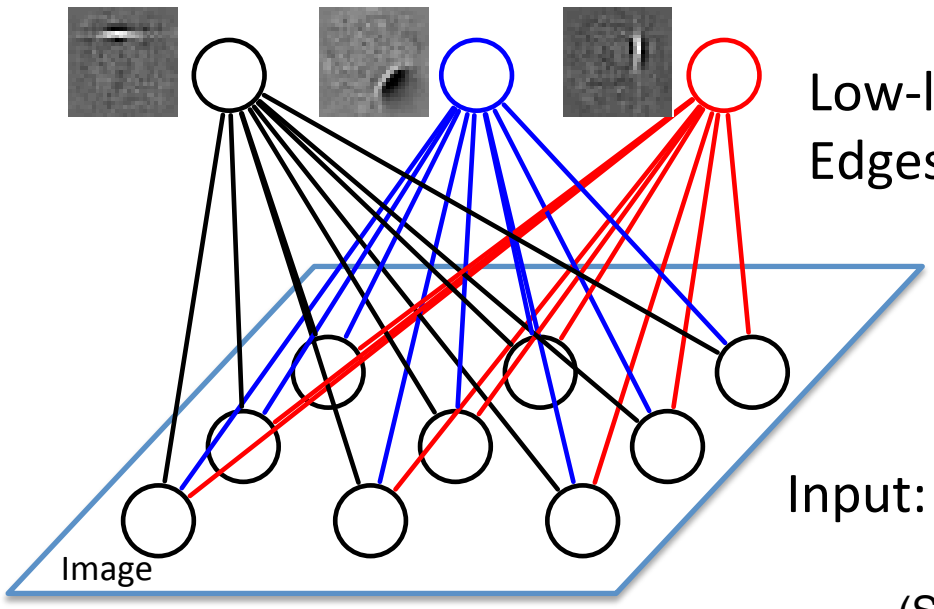
government
auhority
power
empire
putin

clint
hou
pres
bill
cong



tations allow the
, "corruption" and
ive very high
probability to a word "Putin".

Deep Boltzmann Machines



Low-level features:
Edges

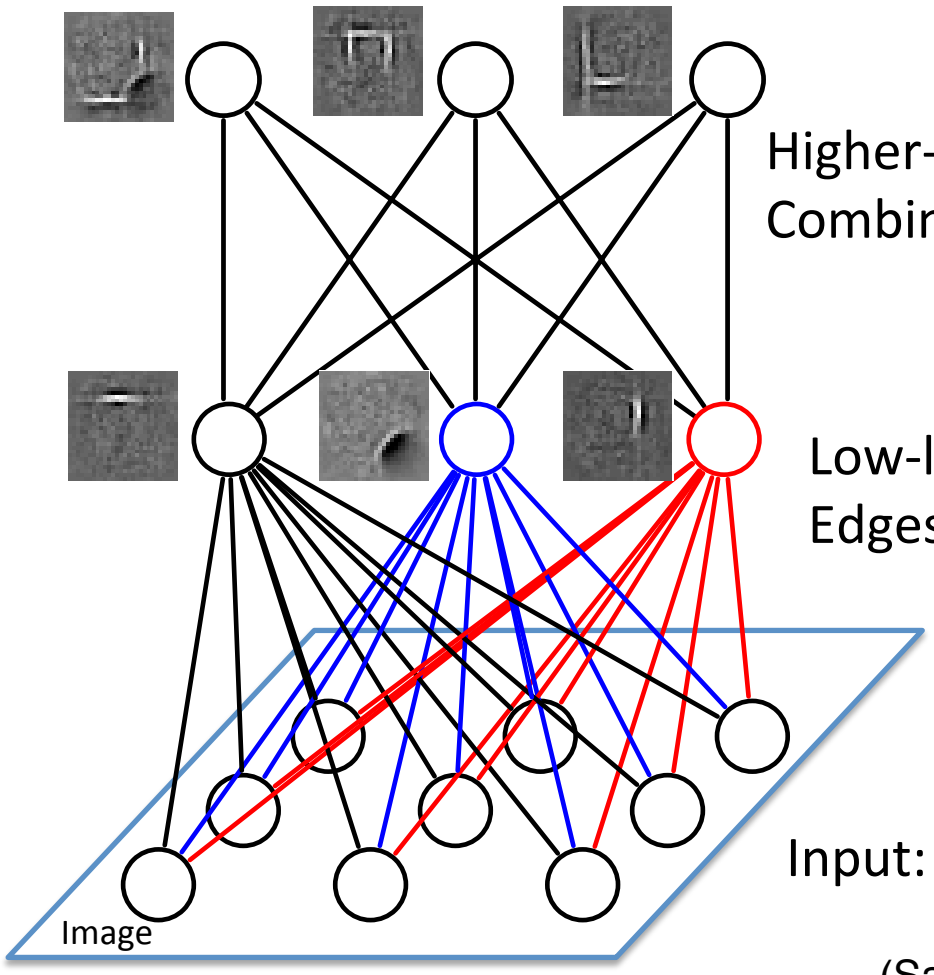
Built from **unlabeled** inputs.

Input: Pixels

(Salakhutdinov & Hinton, Neural Computation 2012)

Deep Boltzmann Machines

Learn simpler representations,
then compose more complex ones



Higher-level features:
Combination of edges

Low-level features:
Edges

Input: Pixels

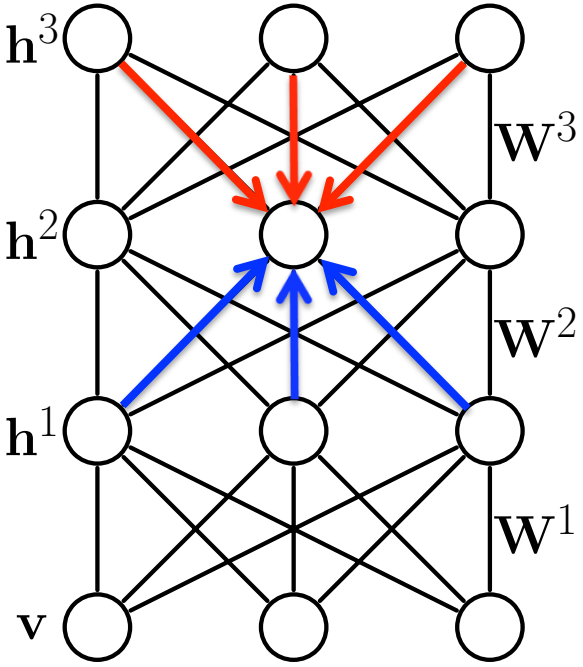
Built from **unlabeled** inputs.

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)

Model Formulation

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

Same as RBMs



Input

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

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

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

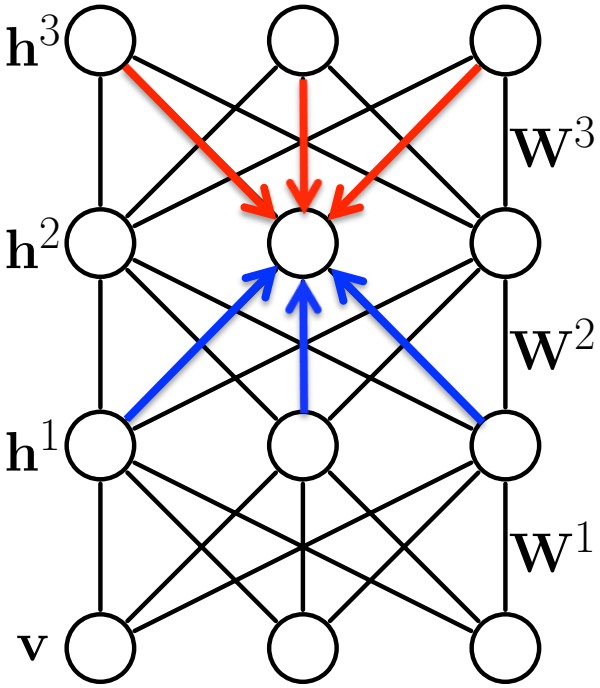
Top-down Bottom-up

- Hidden variables are dependent even when **conditioned on the input**.

Mathematical Formulation

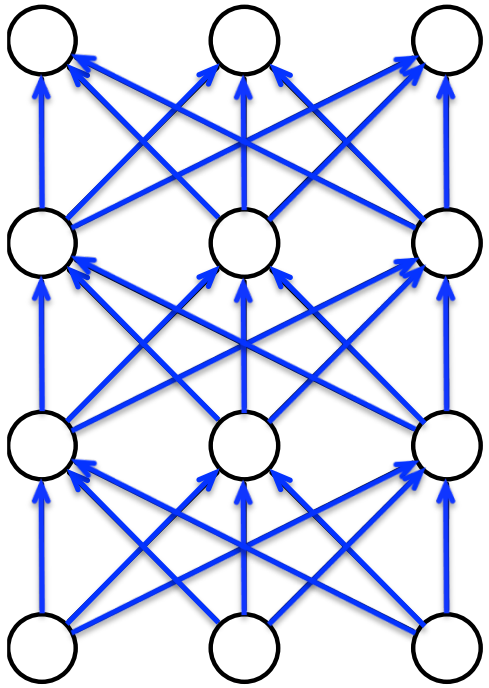
$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}^1,\mathbf{h}^2,\mathbf{h}^3} \exp\left[\mathbf{v}^\top W^1 \mathbf{h}^1 + {\mathbf{h}^1}^\top W^2 \mathbf{h}^2 + {\mathbf{h}^2}^\top W^3 \mathbf{h}^3\right]$$

Deep Boltzmann Machine

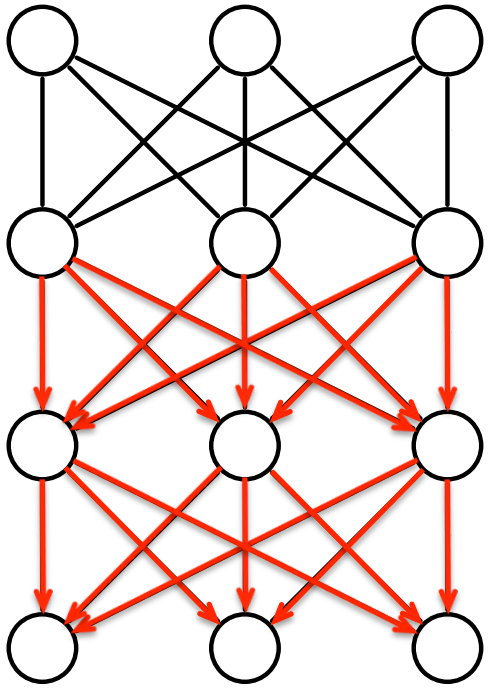


Input

Neural Network Output



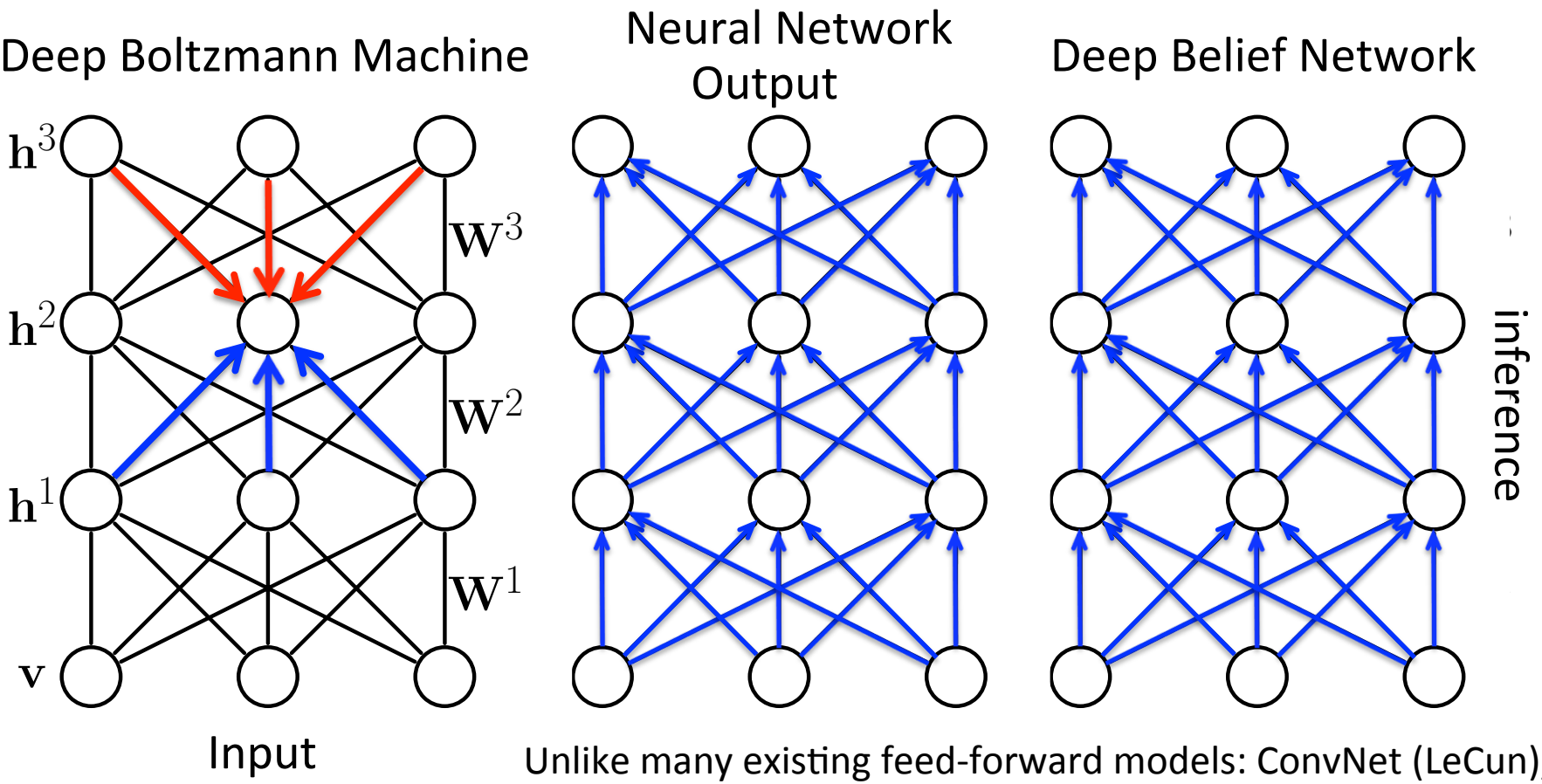
Deep Belief Network



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

Mathematical Formulation

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

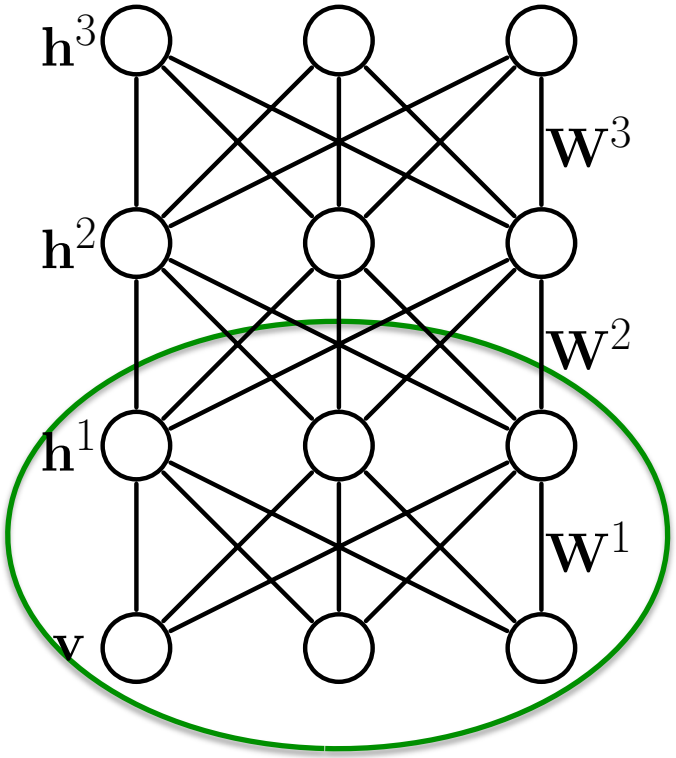


Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio), Deep Belief Nets (Hinton)

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:

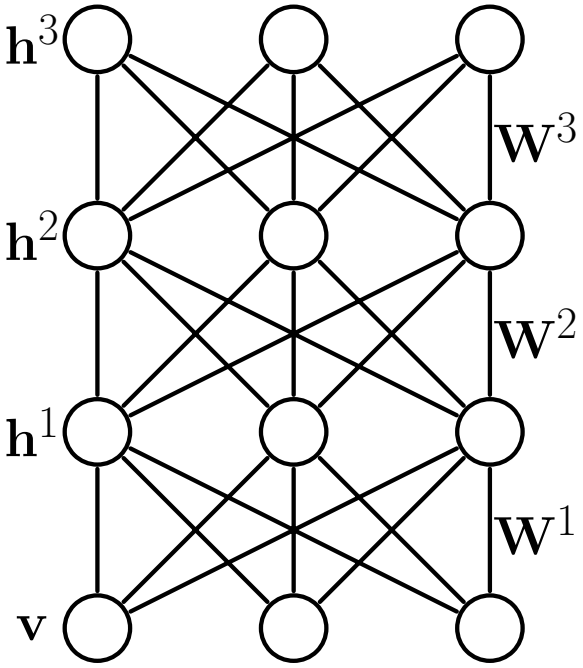
$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathrm{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{1\top}] - \mathrm{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{1\top}]$$

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:

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

- Both expectations are intractable!

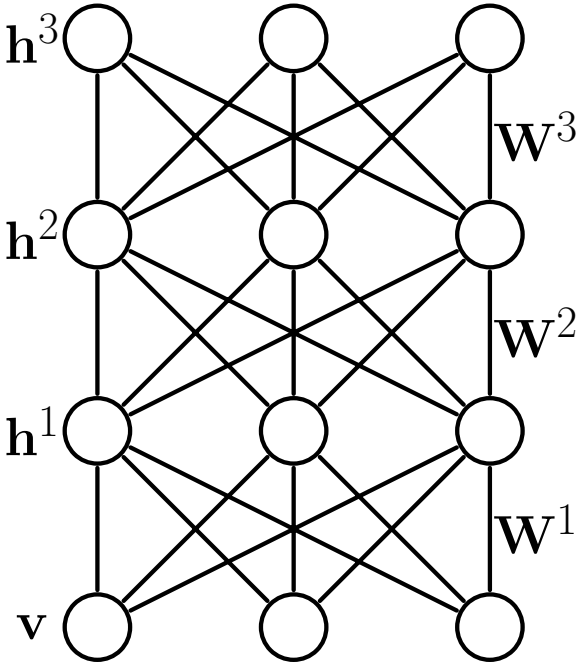
$$P_{data}(\mathbf{v}, \mathbf{h^1}) = P_\theta(\mathbf{h^1}|\mathbf{v}) P_{data}(\mathbf{v})$$

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

Not factorial any more!

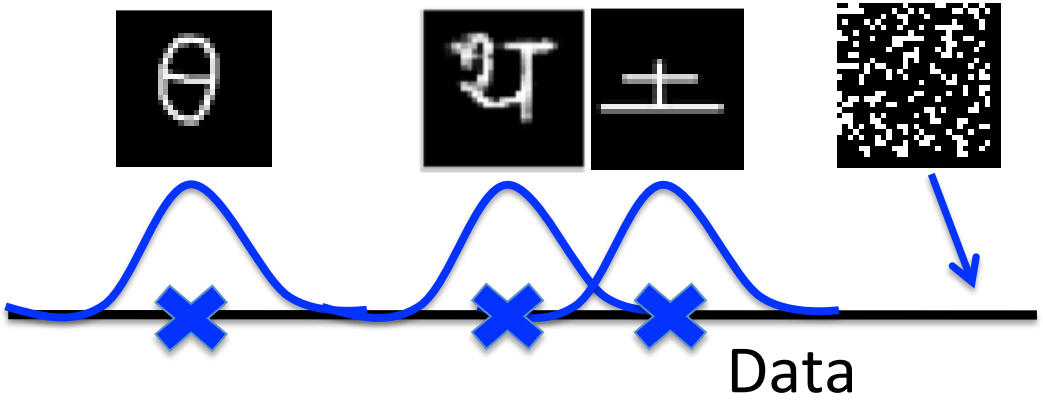
Approximate Learning

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



(Approximate) Maximum Likelihood:

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



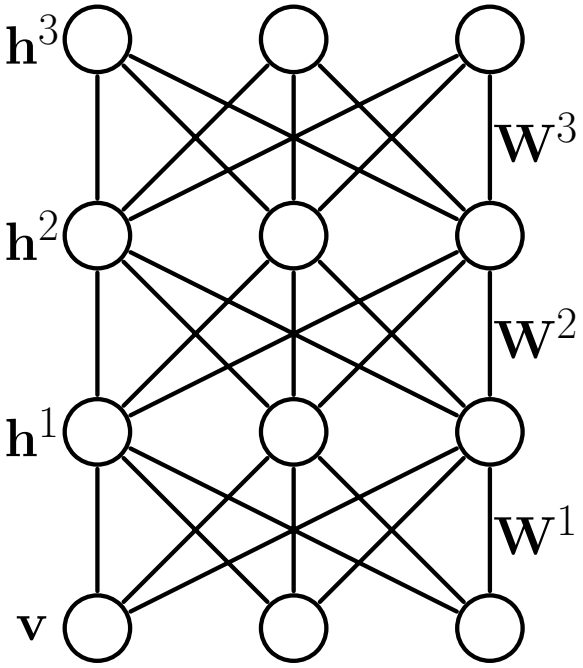
$$P_{data}(\mathbf{v}, \mathbf{h^1}) = P_\theta(\mathbf{h^1}|\mathbf{v}) P_{data}(\mathbf{v})$$

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

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.

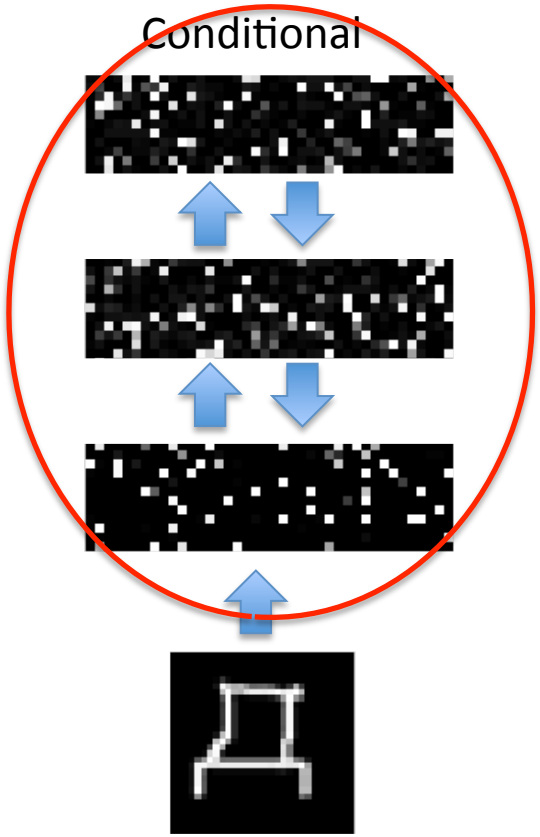
Many of the previous approaches were not successful for learning general Boltzmann machines with **hidden variables.**

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

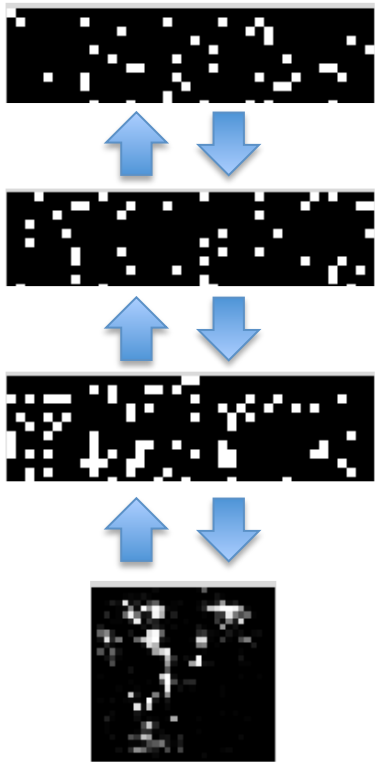
$P_{data}(\mathbf{h}|\mathbf{v})$

Simulate from the Model

Approximate the joint distribution

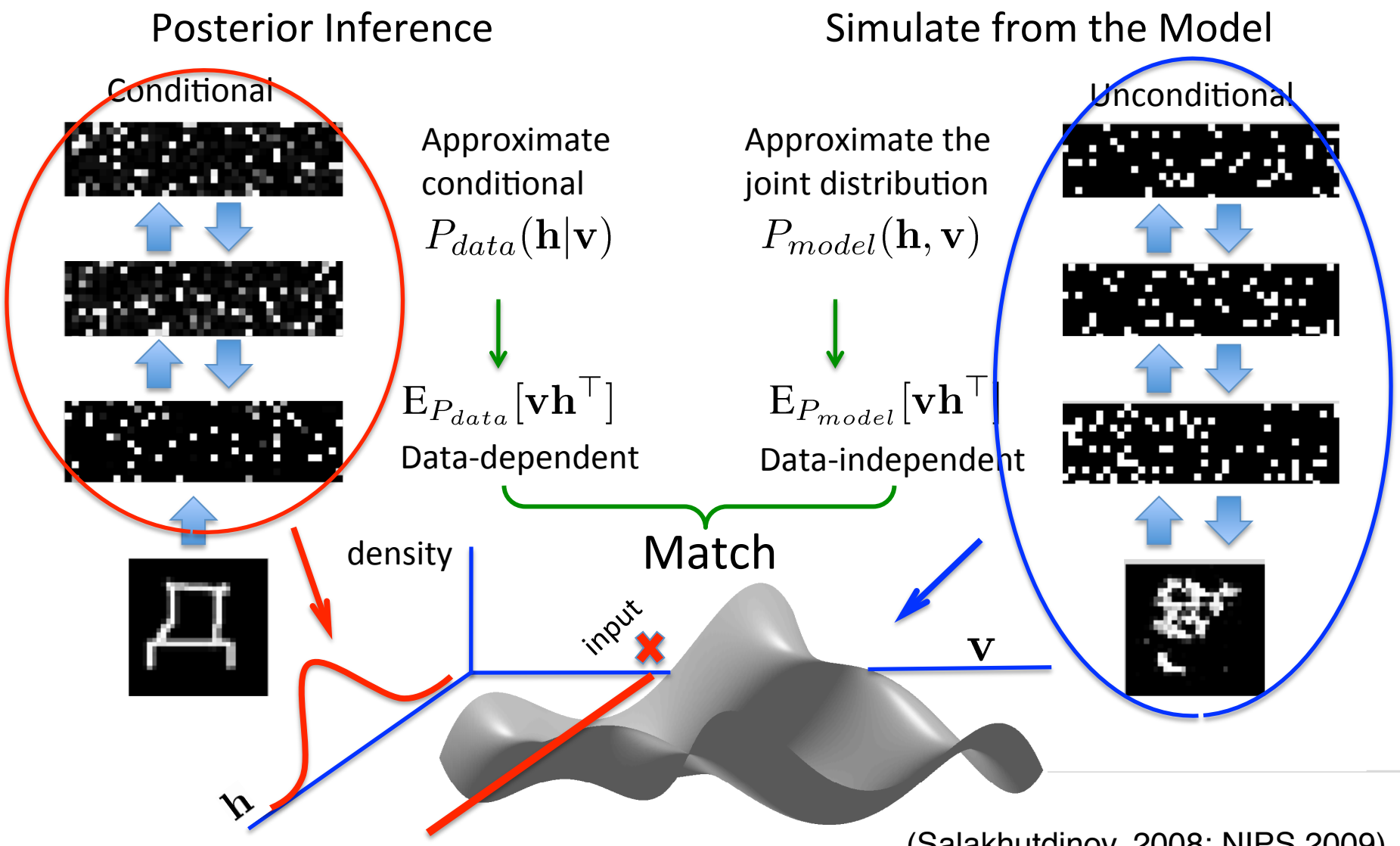
$P_{model}(\mathbf{h}, \mathbf{v})$

Unconditional



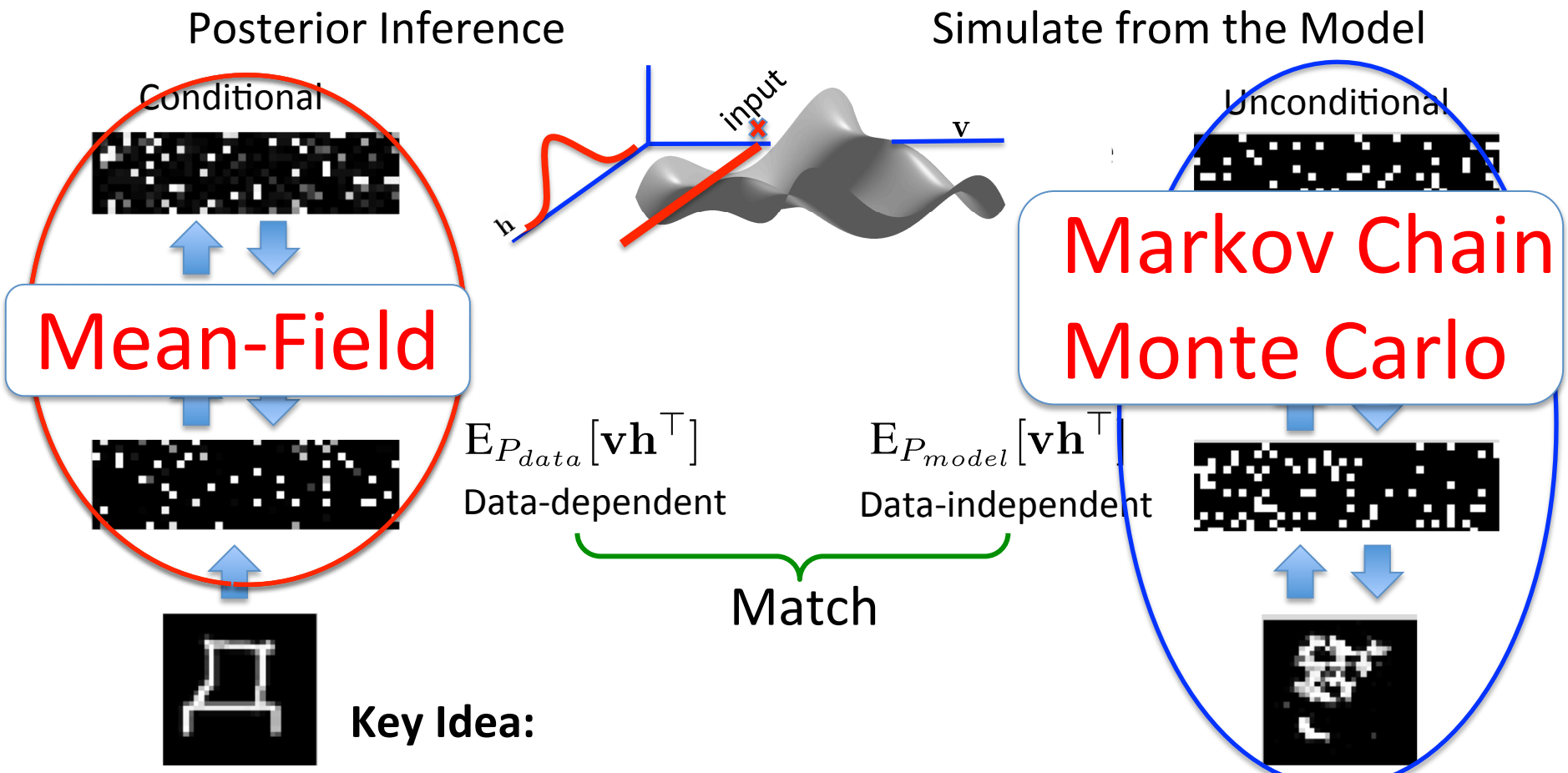
(Salakhutdinov, 2008; NIPS 2009)

New Learning Algorithm



(Salakhutdinov, 2008; NIPS 2009)

New Learning Algorithm

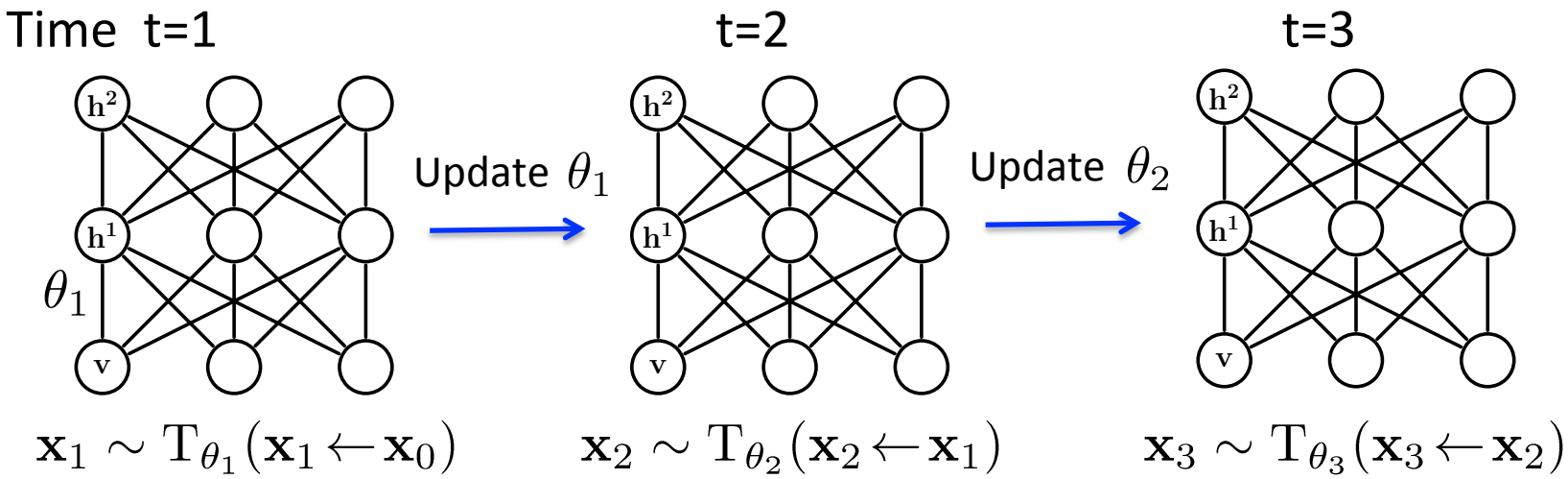


Key Idea:

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

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

Stochastic Approximation



Update θ_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_{θ_t} invariant (e.g. Gibbs or M-H sampler)
- Update θ_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.

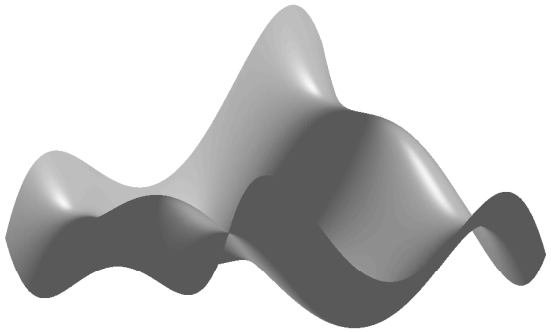
Robbins and Monro, Ann. Math. Stats, 1957
L. Younes, Probability Theory 1989

Learning Algorithm

Update rule decomposes:

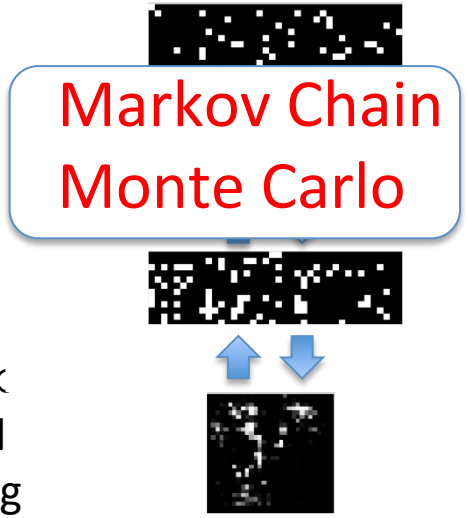
$$\theta_{t+1} = \theta_t + \alpha_t \underbrace{\left(\mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^\top] - \mathbb{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top] \right)}_{\text{True gradient}} + \alpha_t \underbrace{\left(\mathbb{E}_{P_{\theta_t}}[\mathbf{v}\mathbf{h}^\top] - \frac{1}{M} \sum_{m=1}^{M}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

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



(Salakhutdinov, ICML 2010, NIPS 2011, Srivastava & Salakhutdinov, NIPS 2012, Grosse et.al., 2013, Burda et.al., 2015);

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



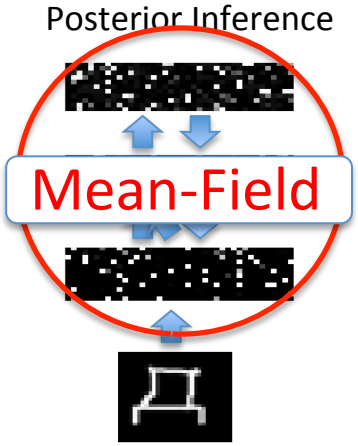
Key insight: The transition operator can b any valid transition operator – Tempered Transitions, Parallel/Simulated Tempering

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 $Q_\mu(\mathbf{h}|\mathbf{v})$:

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



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

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

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 μ .

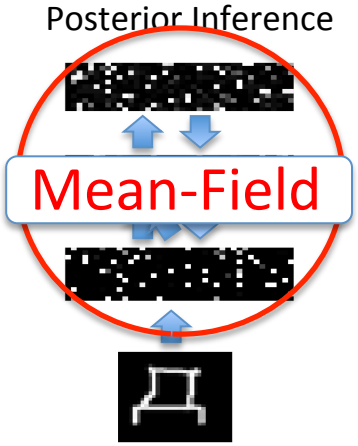
(Salakhutdinov, 2008; Salakhutdinov & Larochelle, AI & Statistics 2010)

Variational Inference

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

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

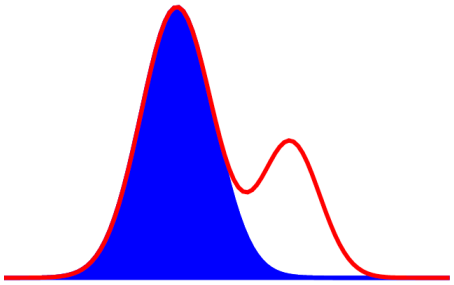
$$\log P_\theta(\mathbf{v}) \geq \underbrace{\log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)}_{\text{Variational Lower Bound}}$$



Mean-Field: Choose a fully factorized distribution:

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

Variational Inference: Maximize the lower bound w.r.t. Variational parameters μ.



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

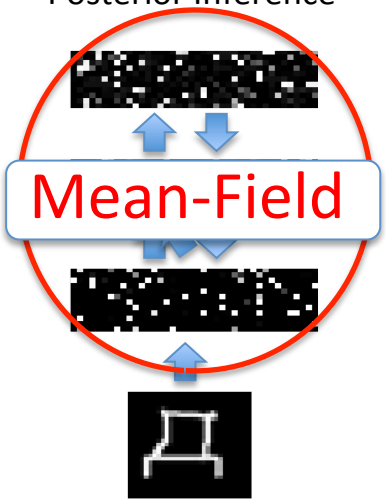
Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

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

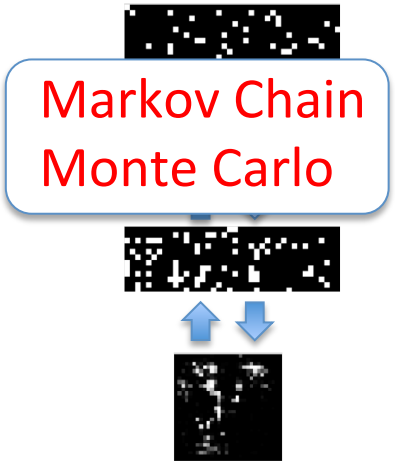
$$\log P_\theta(\mathbf{v}) \geq \log P_\theta(\mathbf{v}) - \mathrm{KL}\big(Q_\mu(\mathbf{h}|\mathbf{v})||P_\theta(\mathbf{h}|\mathbf{v})\big)$$

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

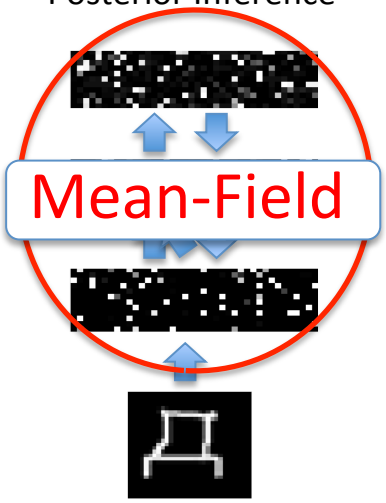
Approximate intractable distribution $P_\theta(\mathbf{h}|\mathbf{v})$ with simpler, tractable distribution $Q_\mu(\mathbf{h}|\mathbf{v})$:

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

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

Variational Lower Bound

Posterior Inference



Unconditional Simulation

1. V wer bou

Fast Inference

2. M to u

Learning can scale to millions of examples

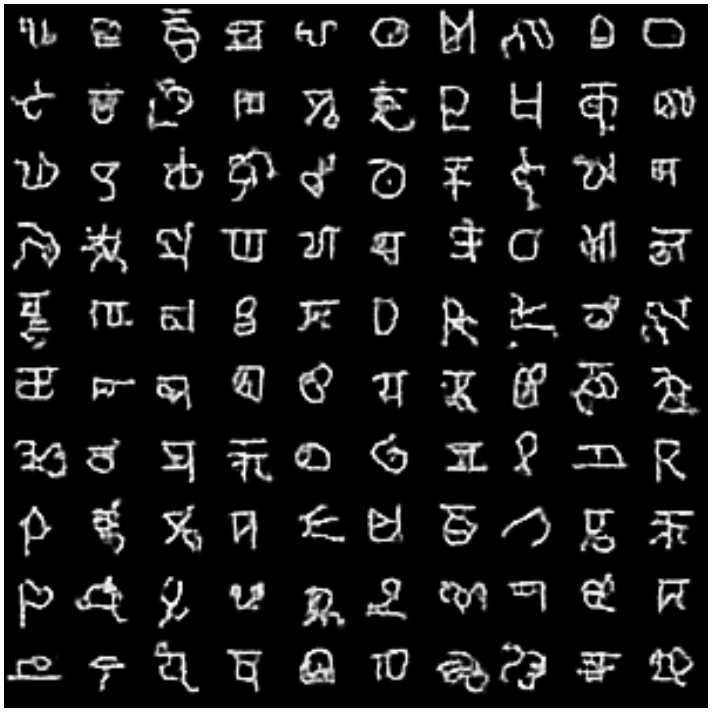
Almost sure convergence guarantees to an asymptotically stable point.

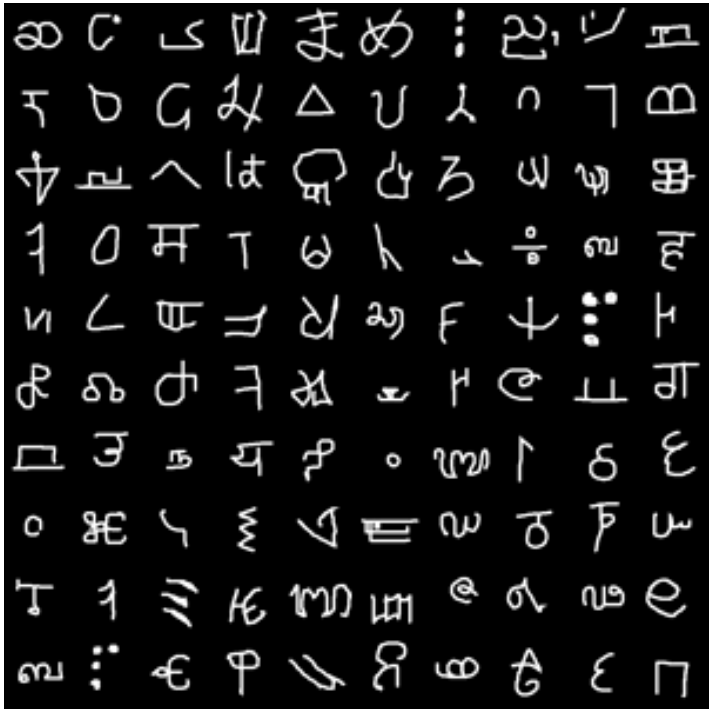
Good Generative Model?

Handwritten Characters

Good Generative Model?

Handwritten Characters





Good Generative Model?

Handwritten Characters

Simulated

Real Data

Good Generative Model?

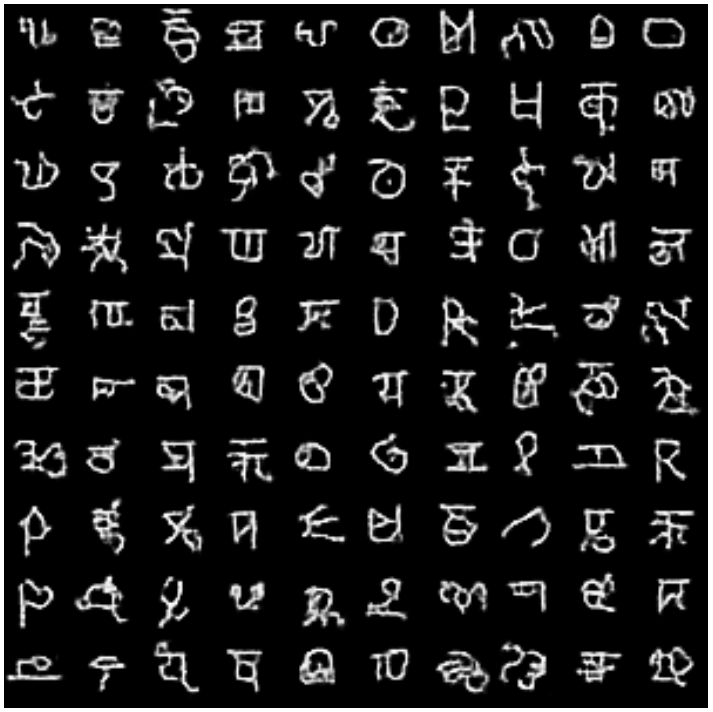
Handwritten Characters

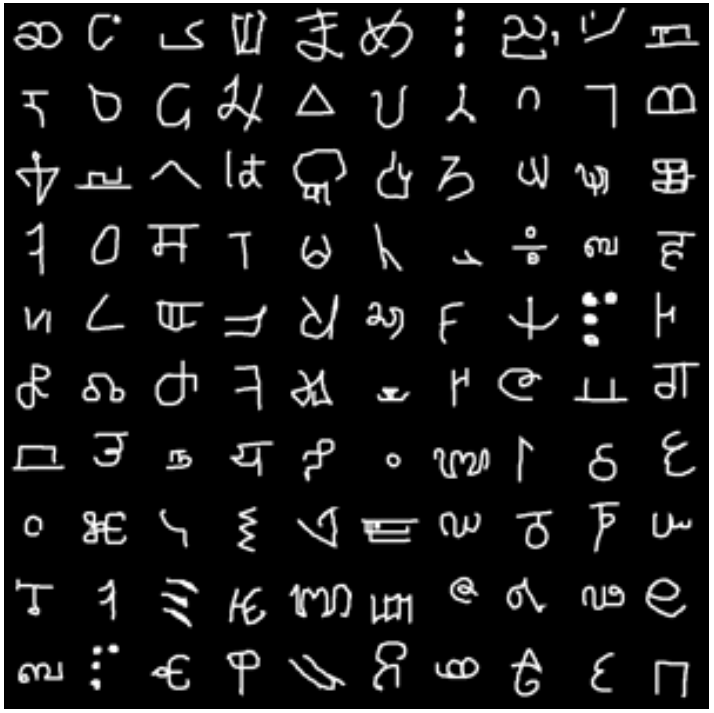
Real Data

Simulated

Good Generative Model?

Handwritten Characters





Good Generative Model?

MNIST Handwritten Digit Dataset

Handwriting Recognition

MNIST Dataset
60,000 examples of 10 digits

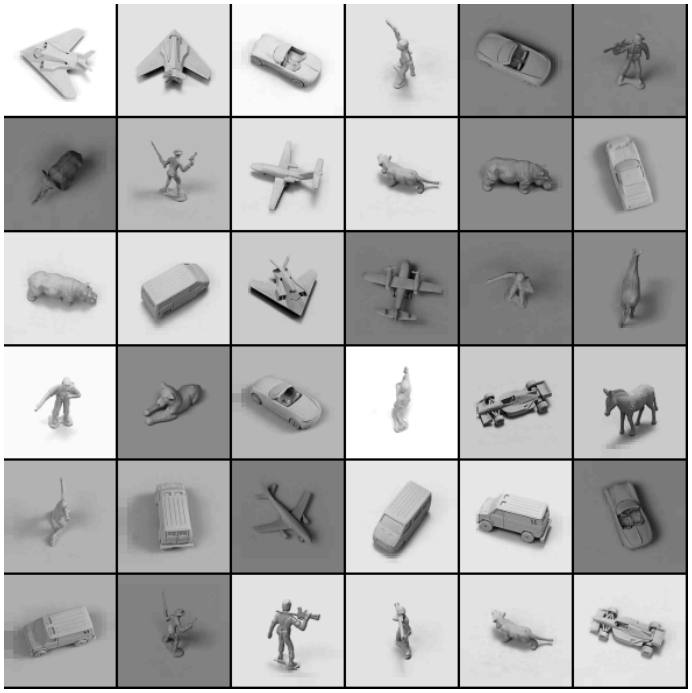
Learning Algorithm	Error
Logistic regression	12.0%
K-NN	3.09%
Neural Net (Platt 2005)	1.53%
SVM (Decoste et.al. 2002)	1.40%
Deep Autoencoder (Bengio et. al. 2007)	1.40%
Deep Belief Net (Hinton et. al. 2006)	1.20%
DBM	**0.95%**

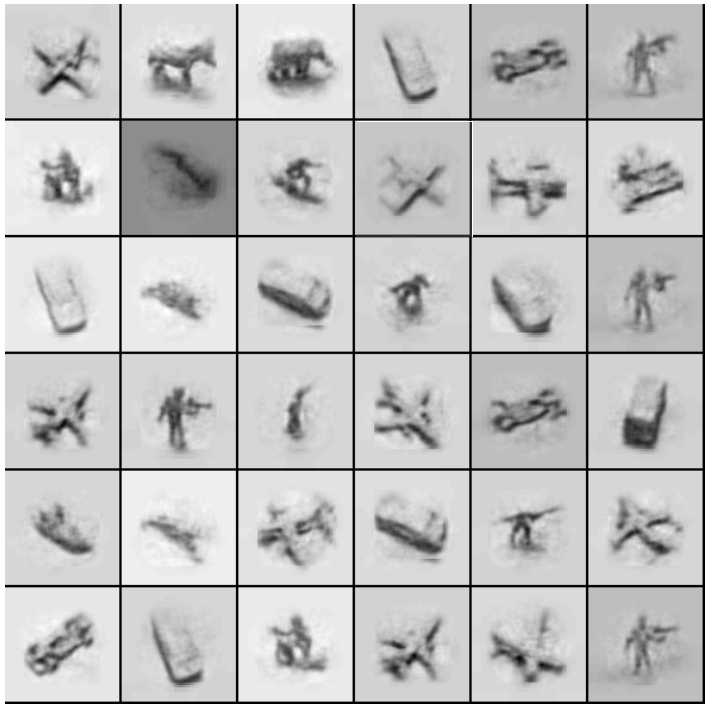
Optical Character Recognition
42,152 examples of 26 English letters

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



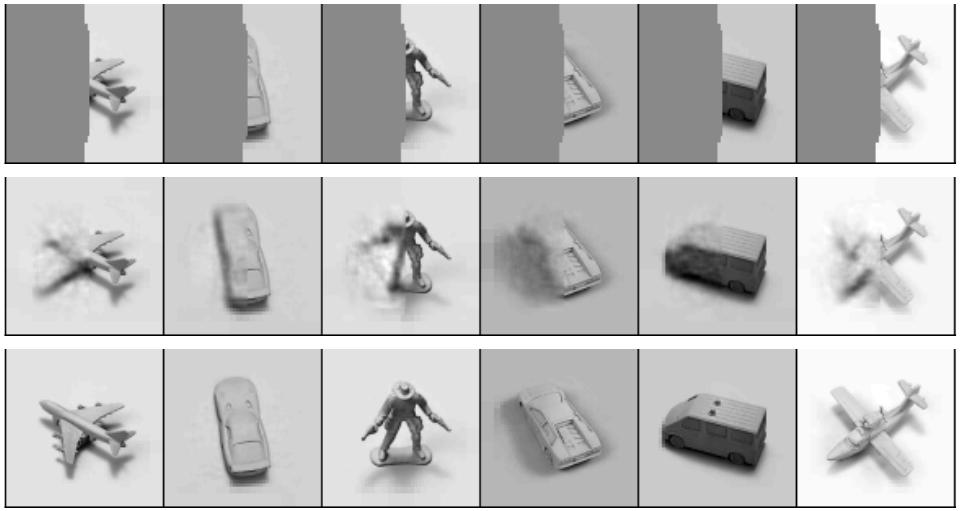


24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.

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.

Talk Roadmap

- Learning Deep Models
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines

- Multi-Modal Learning with DBMs

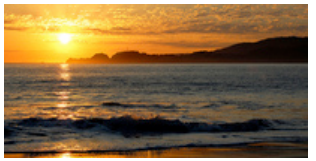
  Srivastava & Salakhutdinov, JMLR 2014, NIPS 2012

Nitish Srivastava

- Evaluating Deep Generative Models

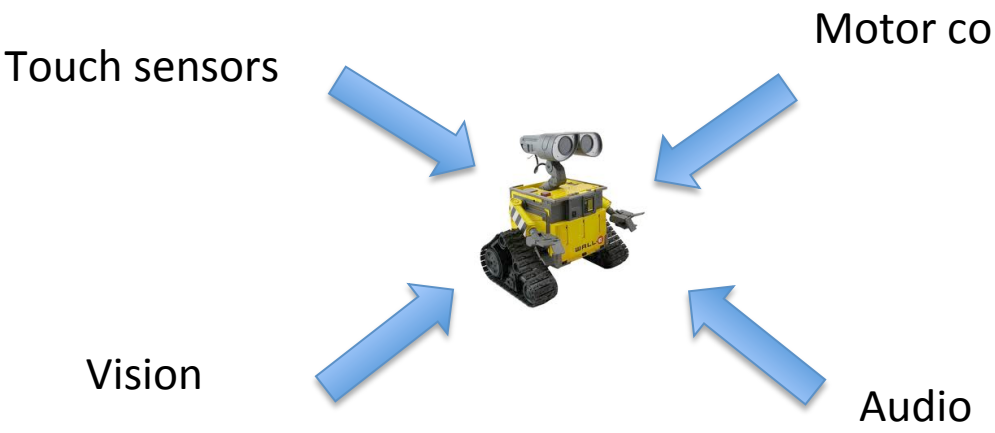
Data – Collection of Modalities

- Multimedia content on the web - image + text + audio.
- Product recommendation systems.
- Robotics applications.



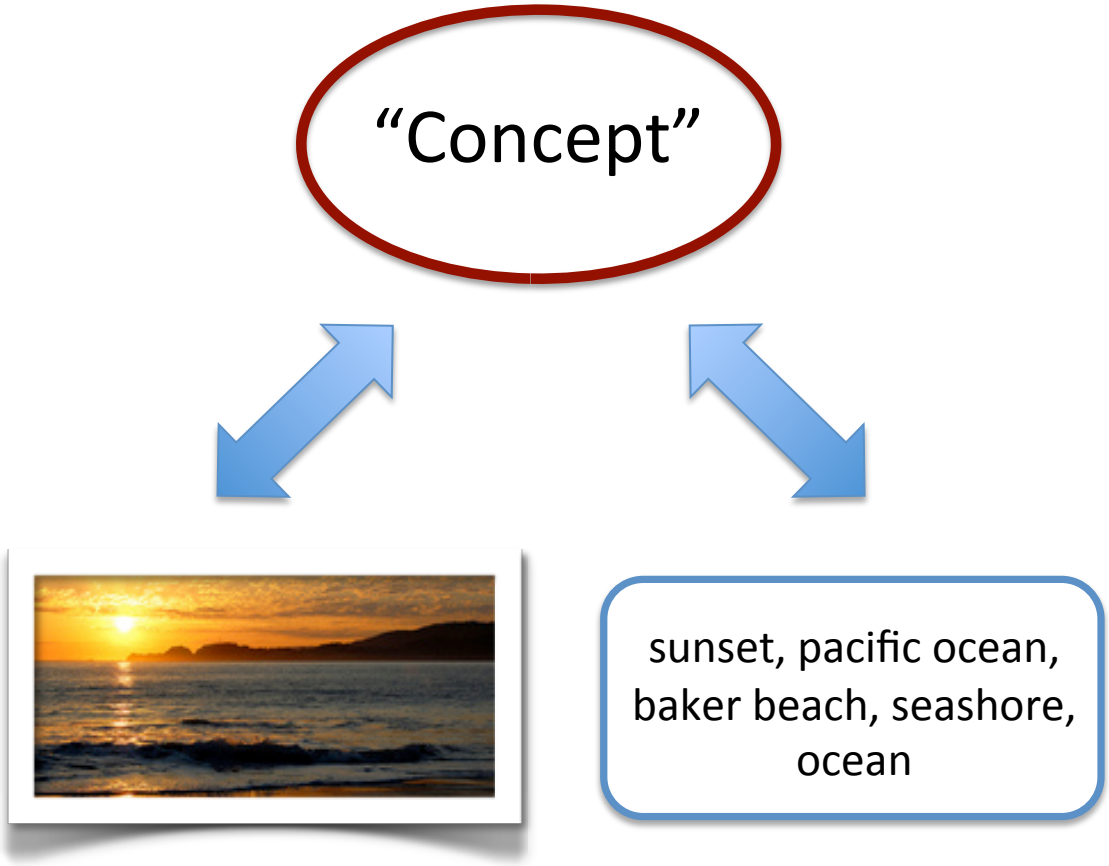
sunset, pacificocean, bakerbeach, seashore, ocean

car, automobile



Shared Concept

"Modality-free" representation



"Modality-full" representation

Multi-Modal Input

- Improve Classification

pentax, k10d, kangarooisland
southaustralia, sa australia
australiansealion 300mm

→ SEA / NOT SEA

- Fill in Missing Modalities

→ beach, sea, surf,
strand, shore,
wave, seascape,
sand, ocean, waves

- Retrieve data from one modality when queried using data from another modality

beach, sea, surf,
strand, shore,
wave, seascape,
sand, ocean, waves

→ 

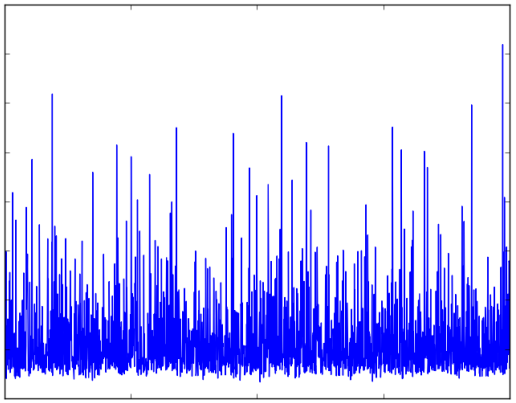
Challenges - I

Image

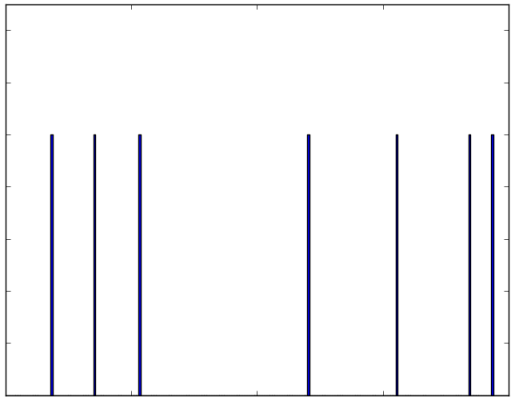
Text

sunset, pacific ocean, baker beach, seashore, ocean

Dense



Sparse



Very different input representations

- Images – real-valued, dense
- Text – discrete, sparse

Difficult to learn cross-modal features from low-level representations.

Challenges - II

Image	Text
	pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion
	mickikrimmel, mickipedia, headshot
	< no text>
	unseulpixel, naturey, crap

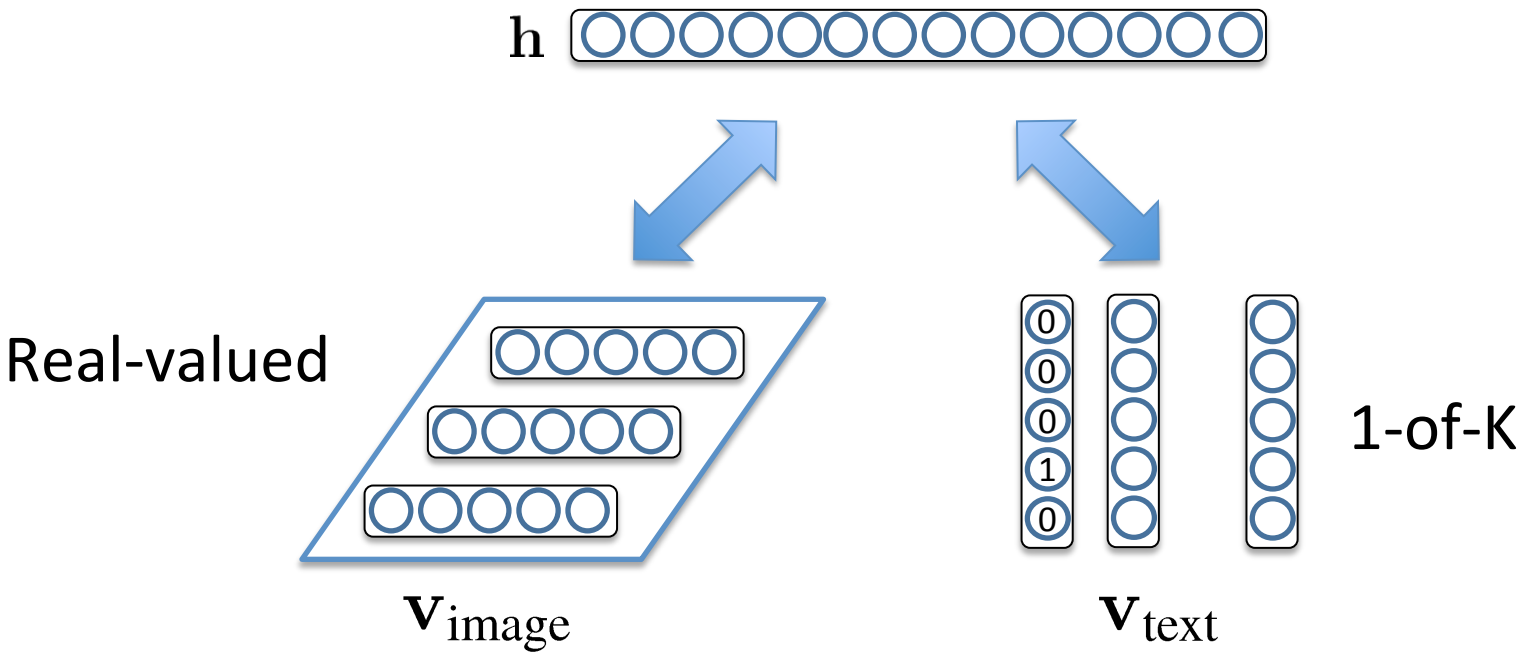
Noisy and missing data

Challenges - II

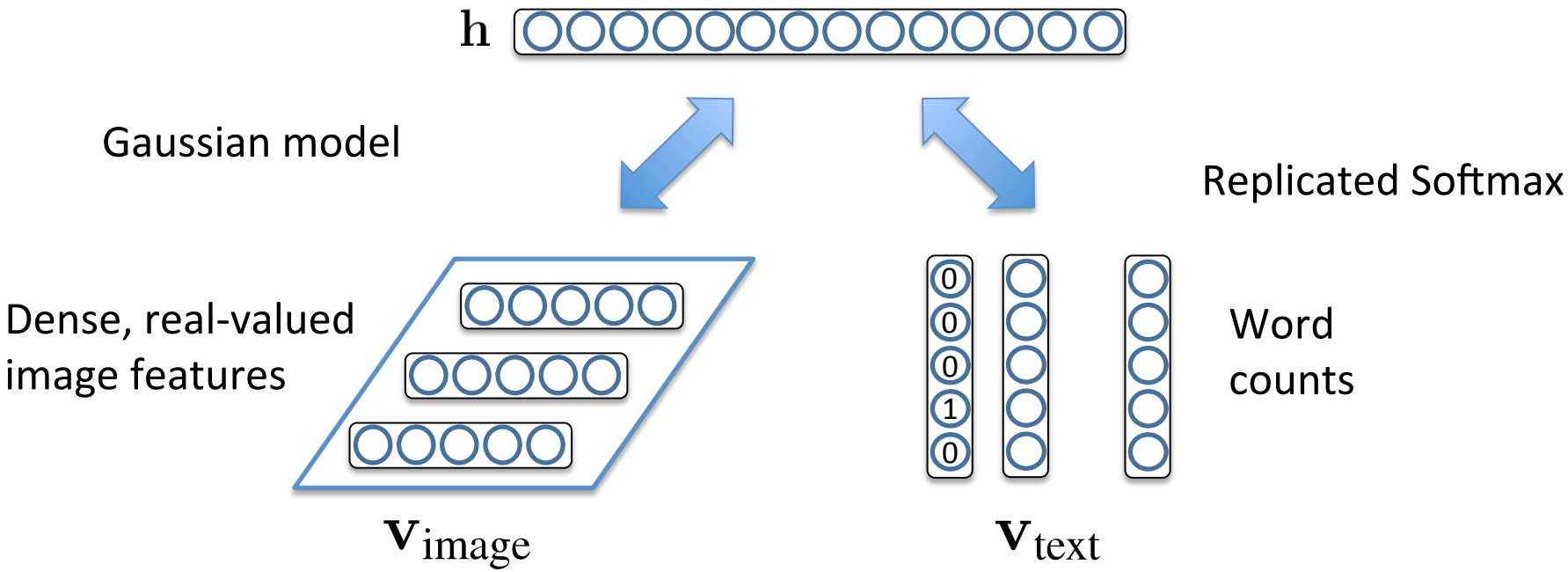
Image	Text	Text generated by the model
	pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves
	mickikrimmel, mickipedia, headshot	portrait, girl, woman, lady, blonde, pretty, gorgeous, expression, model
	< no text>	night, notte, traffic, light, lights, parking, darkness, lowlight, nacht, glow
	unseulpixel, naturey, crap	fall, autumn, trees, leaves, foliage, forest, woods, branches, path

A Simple Multimodal Model

- Use a joint binary hidden layer.
- **Problem**: Inputs have very different statistical properties.
- Difficult to learn cross-modal features.

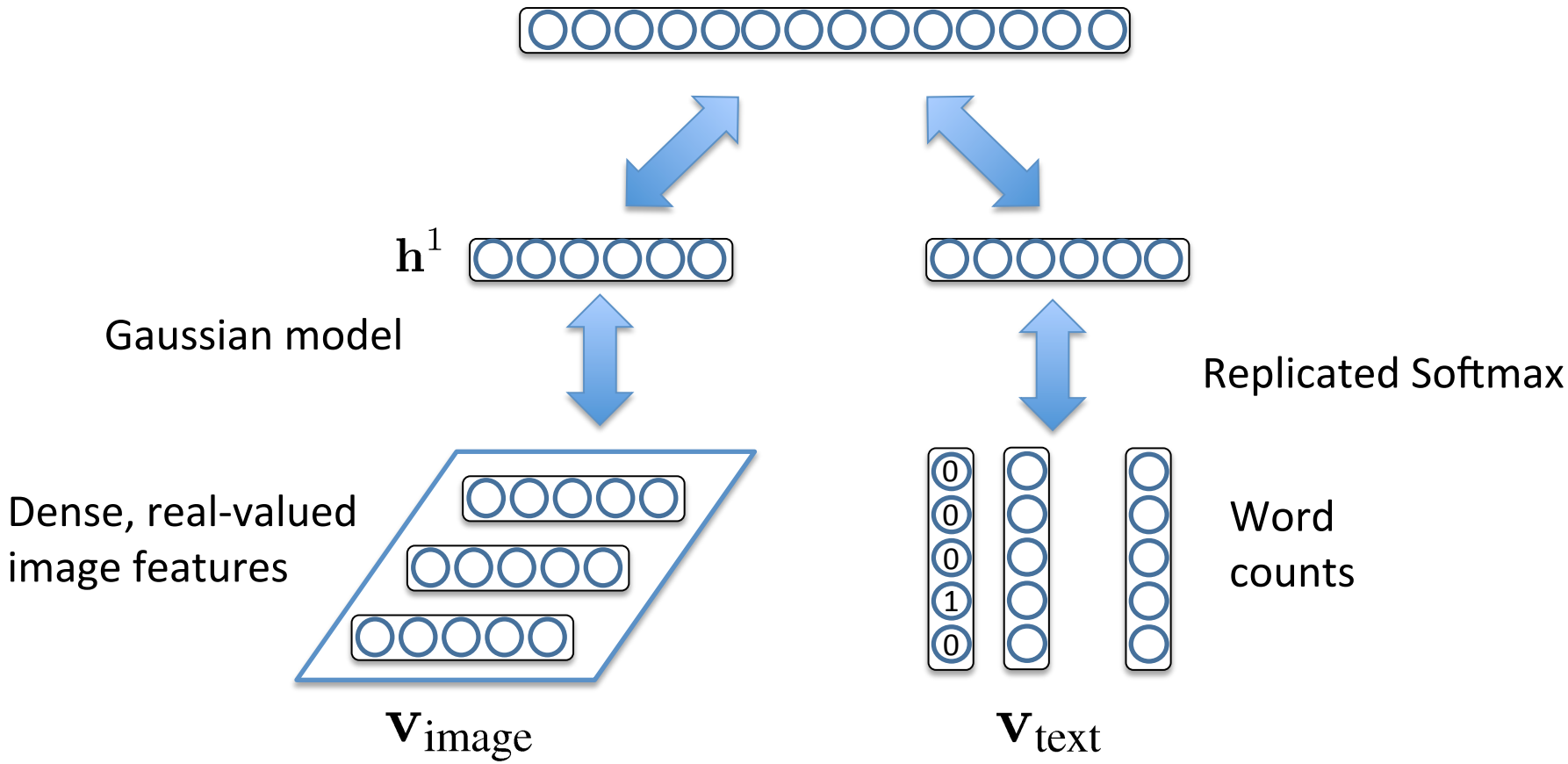


Multimodal DBM



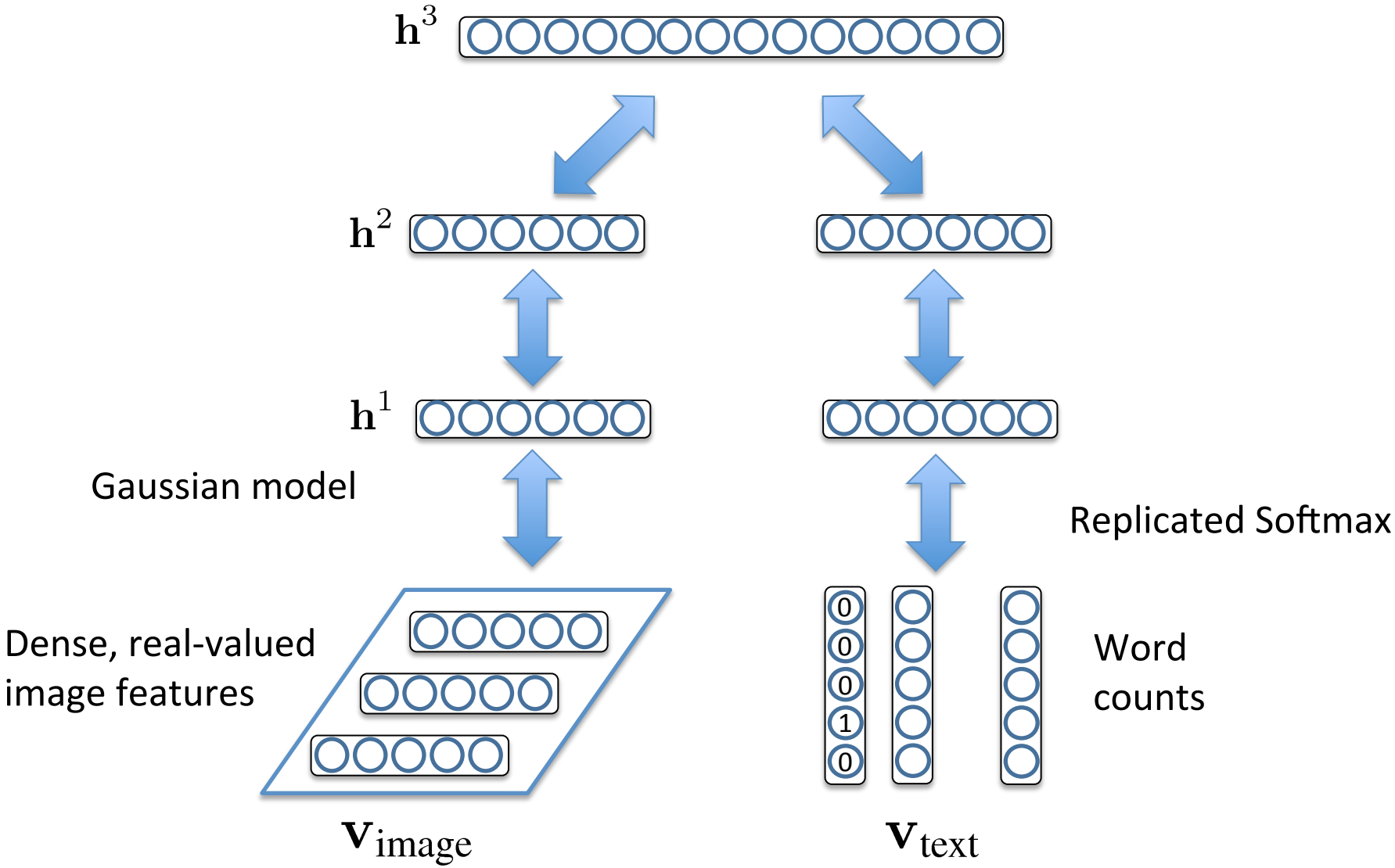
(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM



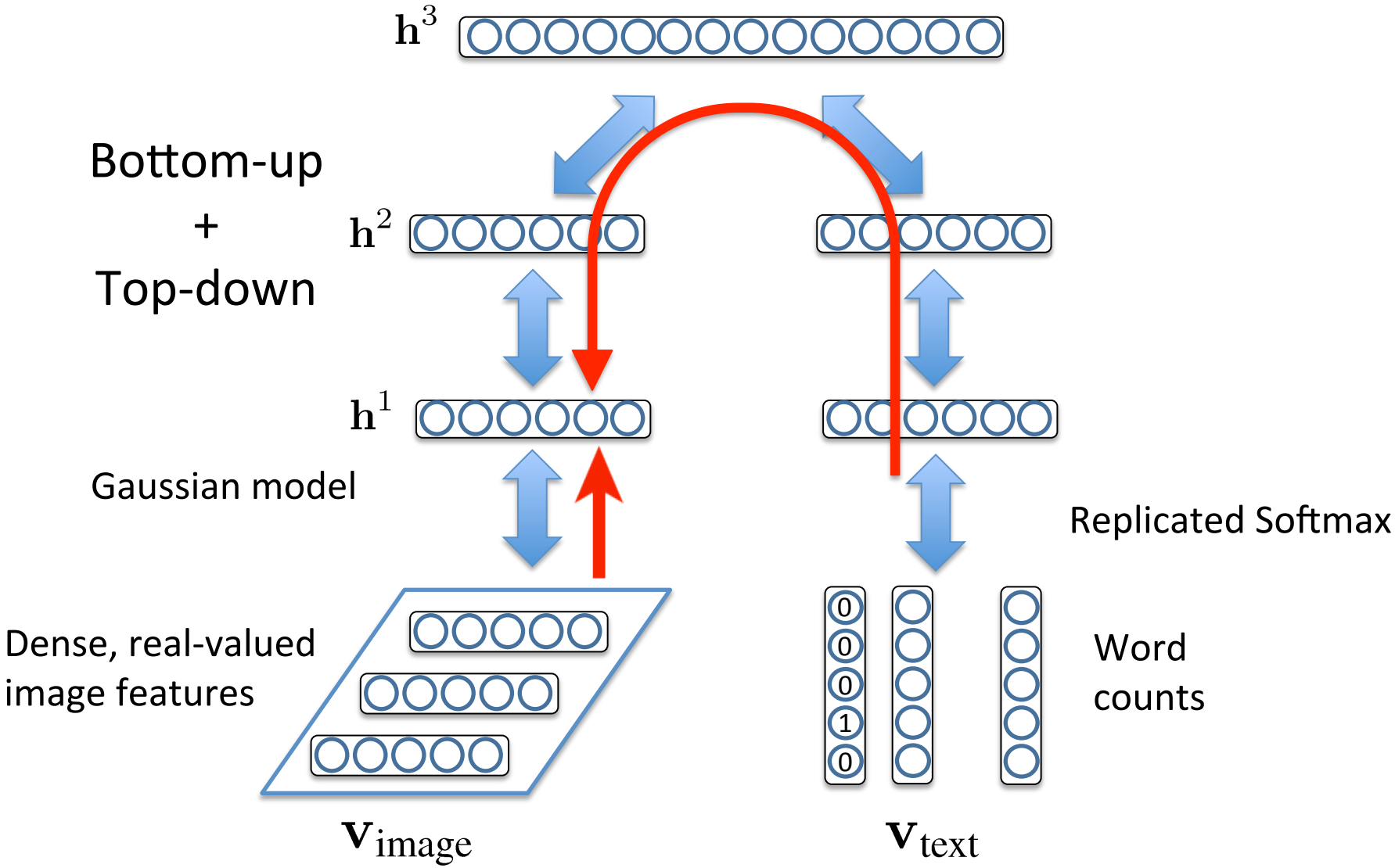
(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM

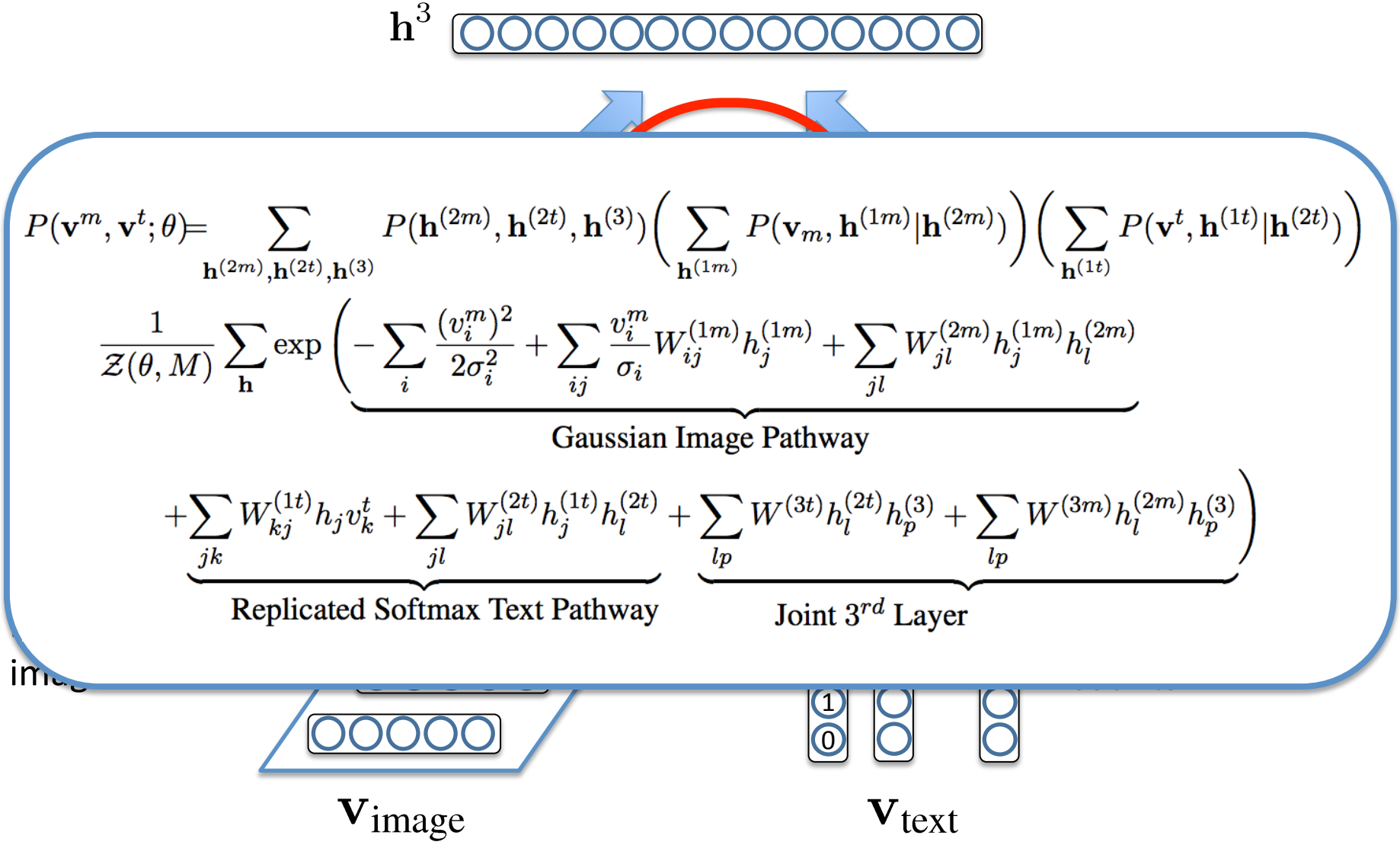


(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM



Multimodal DBM



Text Generated from Images

Given	Generated
	dog, cat, pet, kitten, puppy, ginger, tongue, kitty, dogs, furry
	sea, france, boat, mer, beach, river, bretagne, plage, brittany
	portrait, child, kid, ritratto, kids, children, boy, cute, boys, italy

Given	Generated
	insect, butterfly, insects, bug, butterflies, lepidoptera
	graffiti, streetart, stencil, sticker, urbanart, graff, sanfrancisco
	canada, nature, sunrise, ontario, fog, mist, bc, morning

Text Generated from Images

Given	Generated
	portrait, women, army, soldier, mother, postcard, soldiers
	obama, barackobama, election, politics, president, hope, change, sanfrancisco, convention, rally
	water, glass, beer, bottle, drink, wine, bubbles, splash, drops, drop

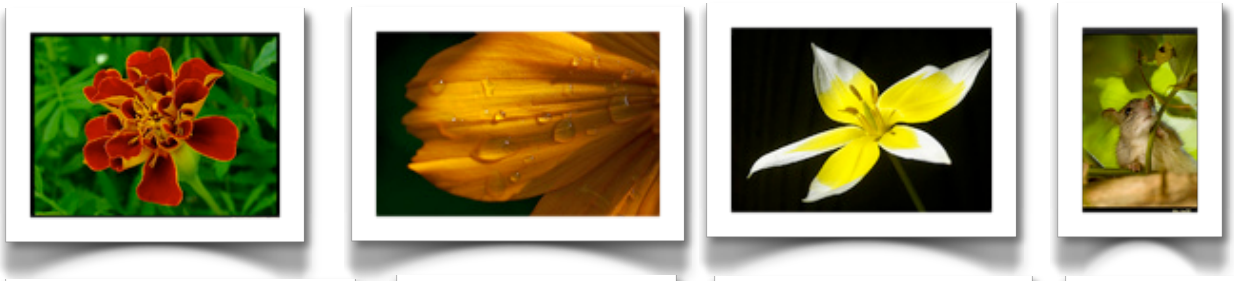
Images from Text

Given

Retrieved

water, red, sunset

nature, flower, red, green



blue, green, yellow, colors

chocolate, cake

MIR-Flickr Dataset

- 1 million images along with user-assigned tags.

sculpture, beauty, stone

d80

nikon, abigfave, goldstaraward, d80, nikond80

food, cupcake, vegan

anawesomeshot, theperfectphotographer, flash, damniwishidtakenthat, spiritofphotography

nikon, green, light, photoshop, apple, d70

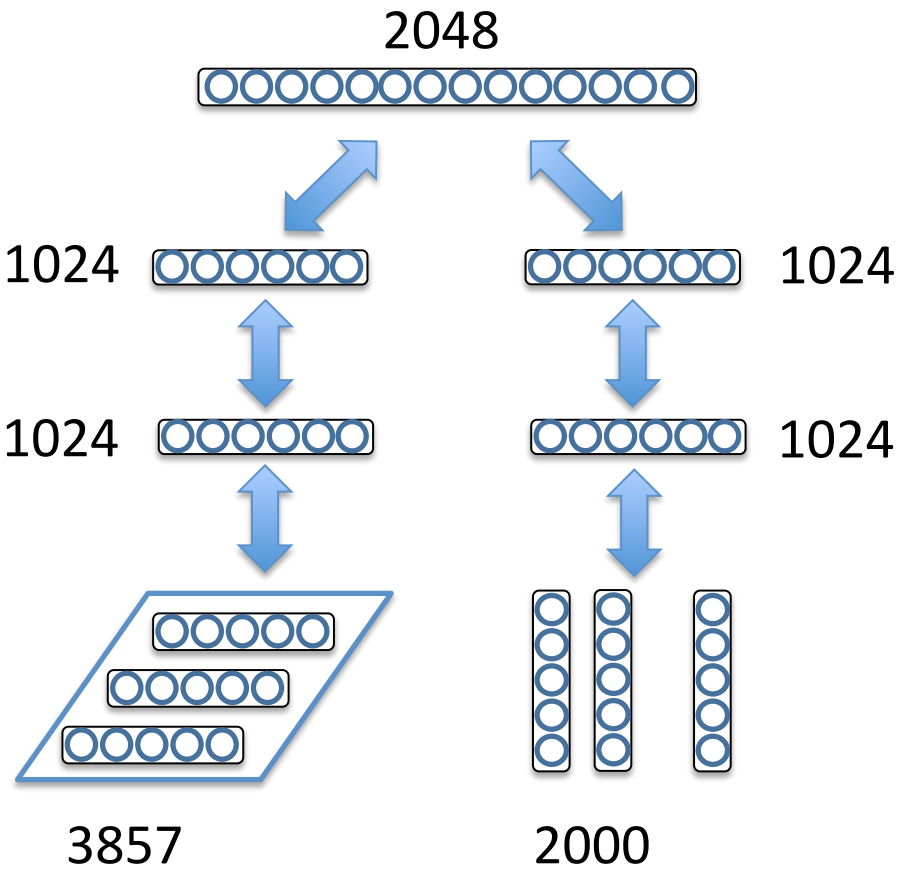
white, yellow, abstract, lines, bus, graphic

sky, geotagged, reflection, cielo, bilbao, reflejo

Huiskes et. al.

Data and Architecture

≈ 12 Million parameters



- 200 most frequent tags.
- 25K labeled subset (15K training, 10K testing)
- Additional 1 million unlabeled data
- 38 classes - *sky, tree, baby, car, cloud* …

Results

- Logistic regression on top-level representation.
- Multimodal Inputs

Mean Average Precision

Learning Algorithm	MAP	Precision@50
Random	0.124	0.124
LDA [Huiskes et. al.]	0.492	0.754
SVM [Huiskes et. al.]	0.475	0.758
DBM-Labelled	0.526	0.791

Labeled 25K examples

Results

- Logistic regression on top-level representation.
- Multimodal Inputs

Mean Average Precision

Learning Algorithm	MAP	Precision@50
Random	0.124	0.124
LDA [Huiskes et. al.]	0.492	0.754
SVM [Huiskes et. al.]	0.475	0.758
DBM-Labelled	0.526	0.791
Deep Belief Net	0.638	0.867
Autoencoder	0.638	0.875
DBM	0.641	0.873

Labeled 25K examples

+ 1 Million unlabelled

Generating Sentences

- More challenging problem.
- How can we generate complete descriptions of images?

Input

Output

A man skiing down the snow covered mountain with a dark sky in the background.

Second Half of Tutorial

Talk Roadmap

- Learning Deep Models
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines
- Multi-Modal Learning with DBMs

- Evaluating Deep Generative Models

Yura Burda

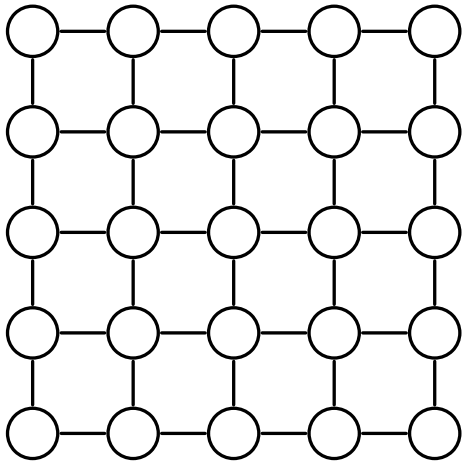
Roger Grosse

Burda, Grosse, Salakhutdinov,
AI & Statistics 2015

Markov Random Fields

Graphical Models: Powerful framework for representing dependency structure between random variables.

$$P_\theta(\mathbf{x}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(-E(\mathbf{x};\theta)\right) = \frac{f_\theta(\mathbf{x})}{\mathcal{Z}(\theta)}$$

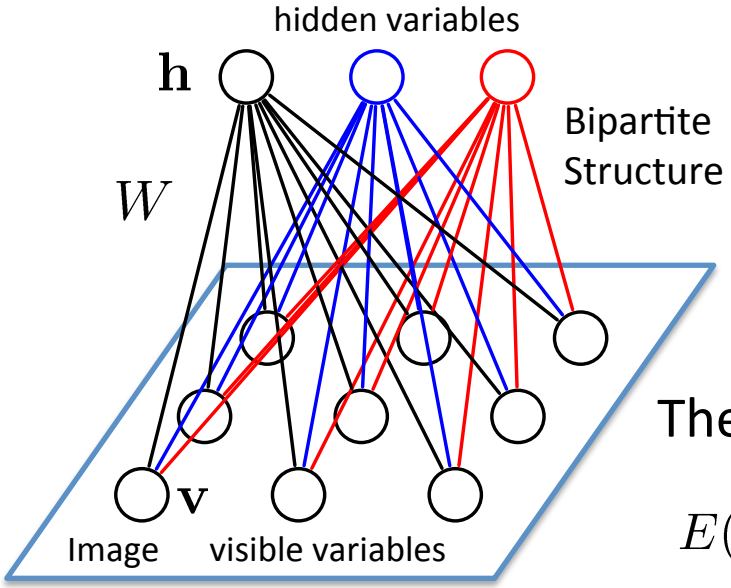


Partition function: difficult to compute

$$\mathcal{Z}(\theta) = \sum_{\mathbf{x}} \exp\left(-E(\mathbf{x};\theta)\right)$$

- **Goal**: Obtain good estimates of $\mathcal{Z}(\theta)$.

Restricted Boltzmann Machines



Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$ are connected to stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.

The energy of the joint configuration:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$\theta = \{W, a, b\}$ model parameters.

Probability of the joint configuration is given by the Boltzmann distribution:

$$P_\theta(\mathbf{v}) = \frac{1}{\mathcal{Z}(\theta)} \underbrace{\sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right)}_{\text{Tractable}} = \underbrace{\frac{f_\theta(\mathbf{v})}{\mathcal{Z}(\theta)}}_{\text{Intractable}}.$$

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

Model Selection

- Model Selection / Complexity Control?
- Suppose we have two MRFs with parameters θ_A and θ_B.
- Each MRF has different number of hidden units and was trained using different learning rates and different numbers of CD steps.
- On the validation set, we need to compute:

$$\frac{P(\mathbf{v};\theta_A)}{P(\mathbf{v};\theta_B)} = \frac{f_{\theta_A}(\mathbf{v})}{f_{\theta_B}(\mathbf{v})} \times \frac{\mathcal{Z}(\theta_B)}{\mathcal{Z}(\theta_A)}.$$

- This requires knowing the ratio of partition functions.

Generative Model

- Which model is a better generative model?

Model A

Model B

Model Selection

- More generally, how can we choose between models?

RBM samples

Mixture of Bernoulli's

Compare $P(\mathbf{x})$ on the validation set: $P(\mathbf{x}) = f(\mathbf{x})/\mathcal{Z}$.

Need an estimate of Partition Function $\mathcal{Z}$

Model Selection

- More generally, how can we choose between models?

RBM samples

Mixture of Bernoulli's

MoB, test log-probability: -137.64 nats/digit

RBM, test log-probability: -86.35 nats/digit

Difference of about 50 nats!

Simple Importance Sampling

- Two distributions defined on $\mathcal{X}$ with probability distribution functions $p_{\text{ini}}(\mathbf{x}) = f_{\text{ini}}(\mathbf{x})/\mathcal{Z}_0$ and $p_{\text{tgt}}(\mathbf{x}) = f_{\text{tgt}}(\mathbf{x})/\mathcal{Z}_{\text{tgt}}$

Proposal, easy to sample from distribution

Intractable, target distribution

- Under mild conditions:

$$\mathcal{Z}_{\text{tgt}} = \sum_{\mathbf{x}} f_{\text{tgt}}(\mathbf{x}) = \sum_{\mathbf{x}} \frac{f_{\text{tgt}}(\mathbf{x})}{p_{\text{ini}}(\mathbf{x})} \times p_{\text{ini}}(\mathbf{x})$$

- Get an unbiased estimate by using Monte Carlo approximation:

$$\mathcal{Z}_{\text{tgt}} \approx \frac{1}{M} \sum_{m=1}^{M} \frac{f_{\text{tgt}}(\mathbf{x}^{(m)})}{p_{\text{ini}}(\mathbf{x}^{(m)})} = \frac{1}{M} \sum_{m=1}^{M} w^{(m)} \qquad \mathbf{x}^{(m)} \sim p_{\text{ini}}$$

- In high-dimensional spaces, the variance will be high (or infinite).

Annealing Between Distributions

- Consider a sequence of intermediate distributions:

  $p_0, p_1, \ldots, p_K$ with $p_0 = p_{\text{ini}}$ and $p_K = p_{\text{tgt}}$.

- One general way is to use **geometric averages**:

$$p_\beta(\mathbf{x}) = f_\beta(\mathbf{x})/\mathcal{Z}_\beta = f_{\text{ini}}(\mathbf{x})^{1-\beta} f_{\text{tgt}}(\mathbf{x})^\beta / \mathcal{Z}_\beta$$

  with $0 = \beta_0 < \beta_1 < \ldots < \beta_K = 1$ chosen by the user.

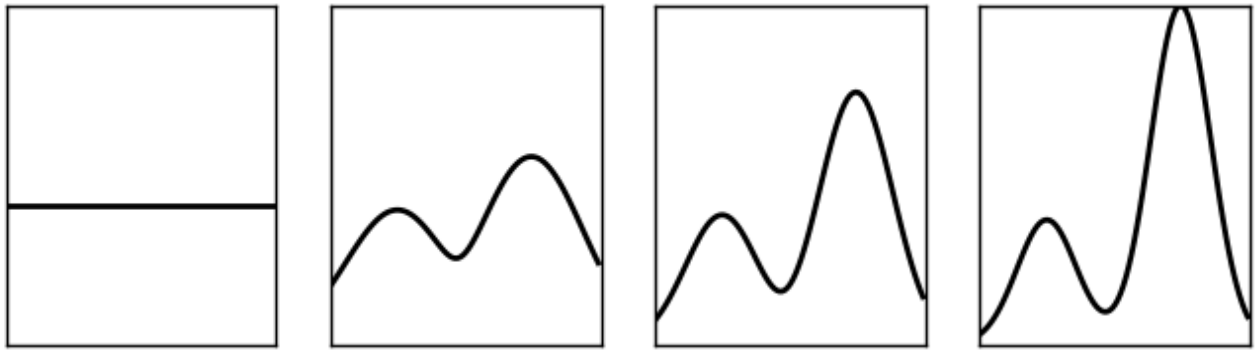
- If p_{ini} is the uniform distribution, then:

$$p_\beta(\mathbf{x}) = f_{\text{tgt}}(\mathbf{x})^\beta / \mathcal{Z}_\beta$$

  hence the term annealing.

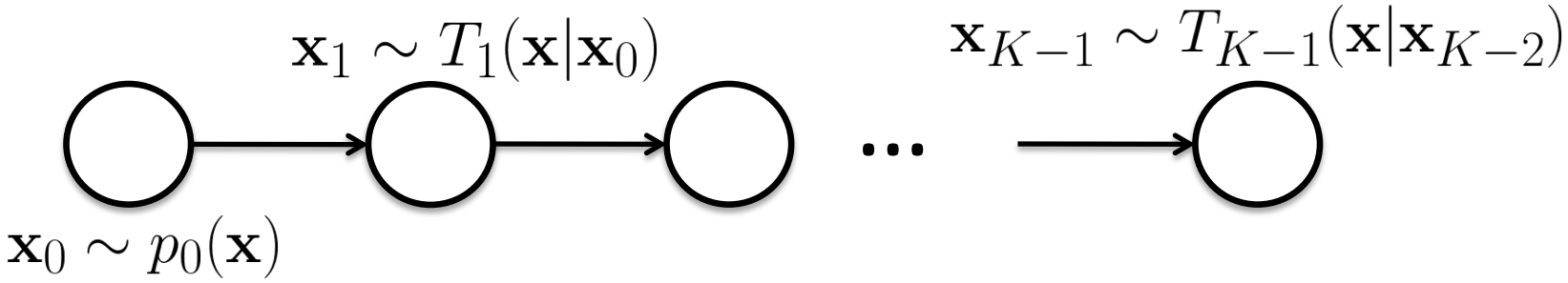
Annealing Between Distributions

- Move gradually from hotter distribution to colder distribution:



- Need to define transition operator $T_k(\mathbf{x}'|\mathbf{x})$ that leaves p_k invariant (e.g. Gibbs sampling) – Easy to implement!

Annealed Importance Sampling Run

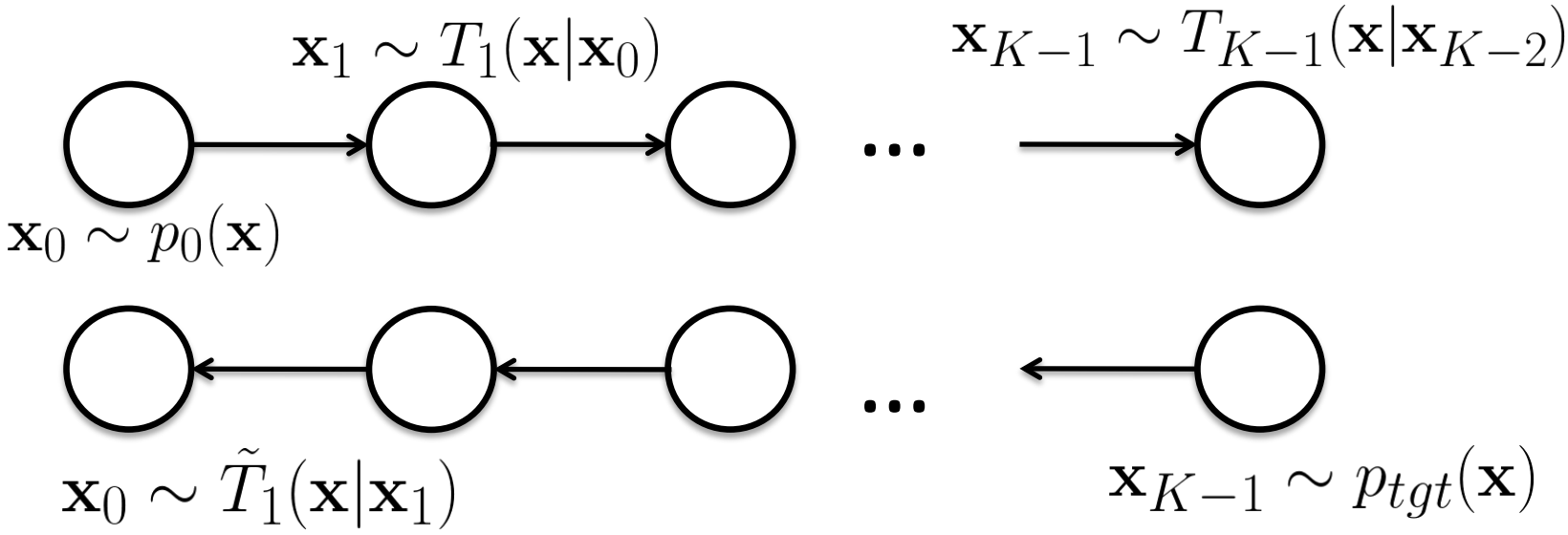


- Generate: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{K-1}$
  - Sample $\mathbf{x}_0 \sim p_0(\mathbf{x})$
  - Sample $\mathbf{x}_1 \sim T_1(\mathbf{x}|\mathbf{x}_0)$
  - …
  - Sample $\mathbf{x}_{K-1} \sim T_{K-1}(\mathbf{x}|\mathbf{x}_{K-2})$

$$w^{(m)} = \frac{f_{tgt}(\mathbf{x}_{K-1})}{p_0(\mathbf{x}_0)} \frac{f_1(\mathbf{x}_0)}{f_1(\mathbf{x}_1)} \frac{f_2(\mathbf{x}_1)}{f_2(\mathbf{x}_2)} \cdots \frac{f_{K-1}(\mathbf{x}_{K-2})}{f_{K-1}(\mathbf{x}_{K-1})}$$

- We obtain an unbiased estimator: $\mathbb{E}[w] = \mathcal{Z}_{\text{tgt}}$

AIS is Importance Sampling



- Forward Markov chain:

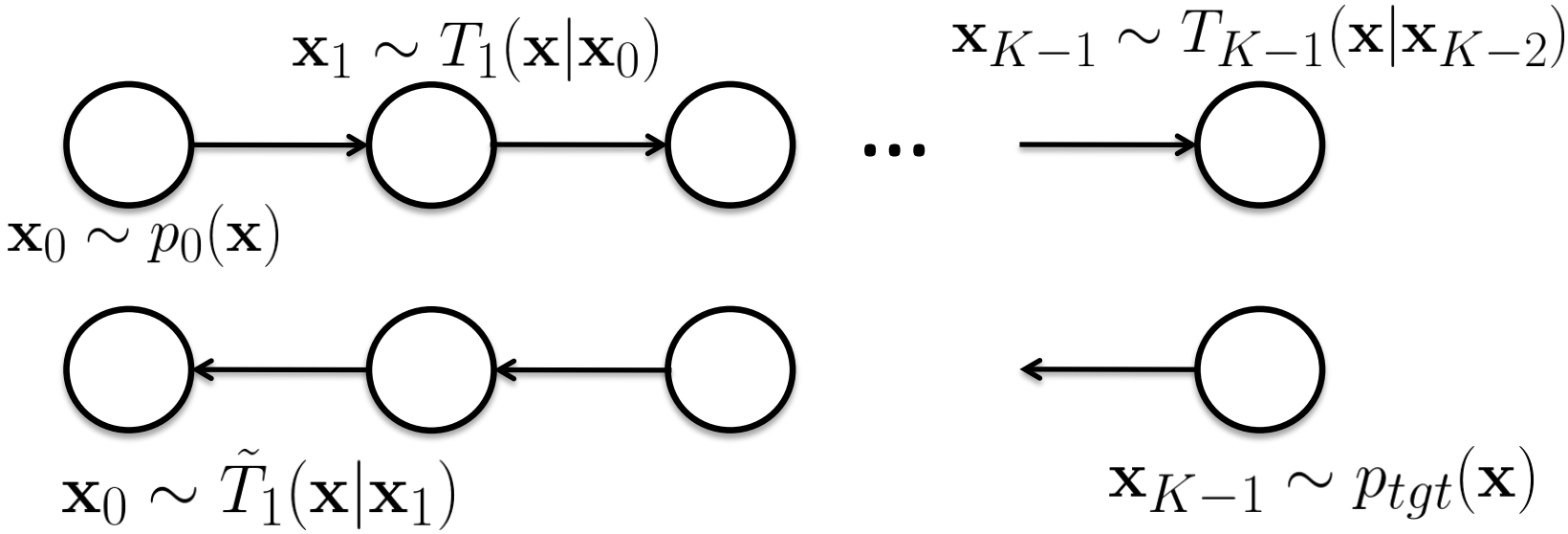
$$q_{\text{fwd}}(\mathbf{x}_0, ..., \mathbf{x}_{K-1}) = p_0(\mathbf{x}_0) \prod_{k=1}^{K-1} T_k(\mathbf{x}_k|\mathbf{x}_{k-1})$$

- Reverse Markov chain (merely a theoretical construct):

$$f_{\text{rev}}(\mathbf{x}_0, ..., \mathbf{x}_{K-1}) = f_{\text{tgt}}(\mathbf{x}_{K-1}) \prod_{k=1}^{K-1} \tilde{T}_k(\mathbf{x}_{k-1}|\mathbf{x}_k)$$

$$\tilde{T}_k(\mathbf{x}' \,|\, \mathbf{x}) = T_k(\mathbf{x} \,|\, \mathbf{x}') p_k(\mathbf{x}') / p_k(\mathbf{x})$$

AIS is Importance Sampling

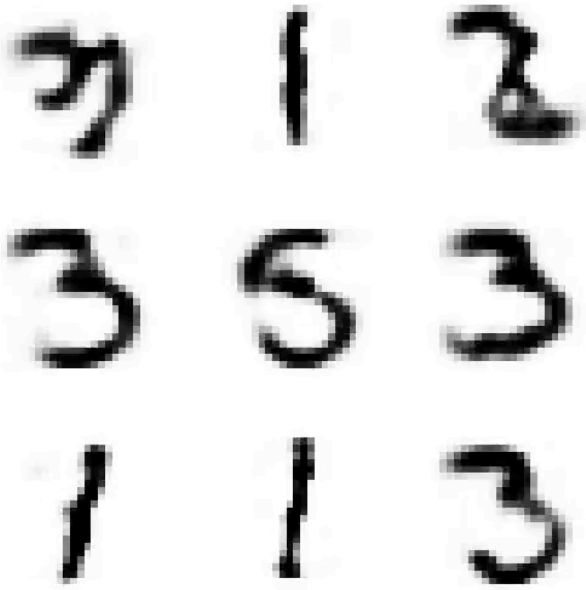


- AIS is a simple importance sampling on extended space:

$$\mathcal{Z}_{tgt} = \mathbb{E}_{q_{\text{fwd}}}\left[\frac{f_{\text{rev}}}{q_{\text{fwd}}}\right] = \mathbb{E}_{q_{\text{fwd}}}[w]$$

RBMs with Geometric Averages

- Restricted Boltzmann Machines trained on MNIST.



Samples from target distribution

AIS with geometric averages

Problems with Undirected Models

- AIS provides an unbiased estimator: $\mathbb{E}[\hat{\mathcal{Z}}_{\text{tgt}}] = \mathcal{Z}_{\text{tgt}}$. In general, we are interested in estimating $\log \mathcal{Z}_{\text{tgt}}$
- By Jensen's inequality:

$$\mathbb{E}[\log \hat{\mathcal{Z}}_{\text{tgt}}] \leq \log \mathbb{E}[\hat{\mathcal{Z}}_{\text{tgt}}] = \log \mathcal{Z}_{\text{tgt}}$$

- By Markov's inequality: very unlikely to overestimate $\log \mathcal{Z}_{\text{tgt}}$

$$\Pr(\log \hat{\mathcal{Z}}_{\text{tgt}} > \log \mathcal{Z}_{\text{tgt}} + b) \leq e^{-b}$$

Stochastic lower bound!

- Compute log-probability on the test set:

$$\log p(\mathbf{x}) = \log f(\mathbf{x}) - \log \mathcal{Z}_{\text{tgt}}$$

overestimate

underestimate

Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):

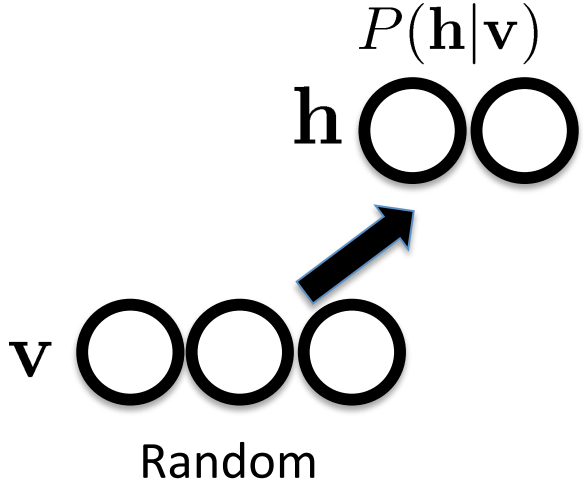
Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):

Random

Motivation: RBM Sampling

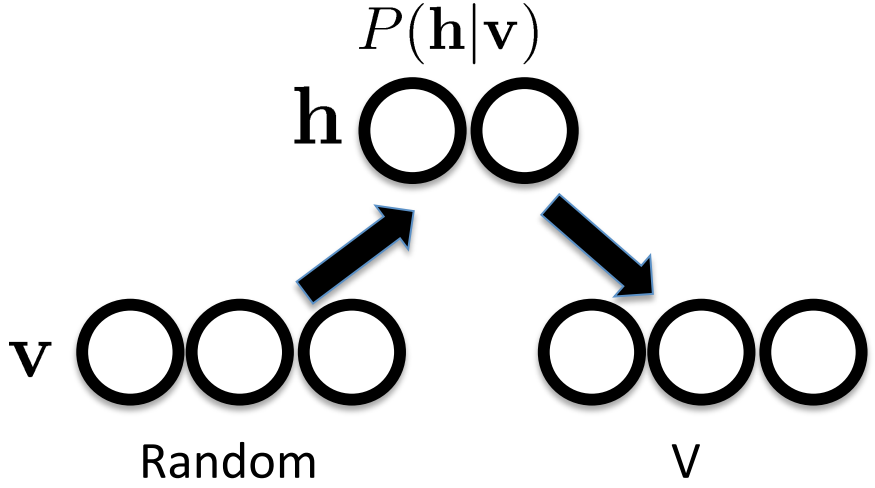
Run Markov chain (alternating Gibbs Sampling):



$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):

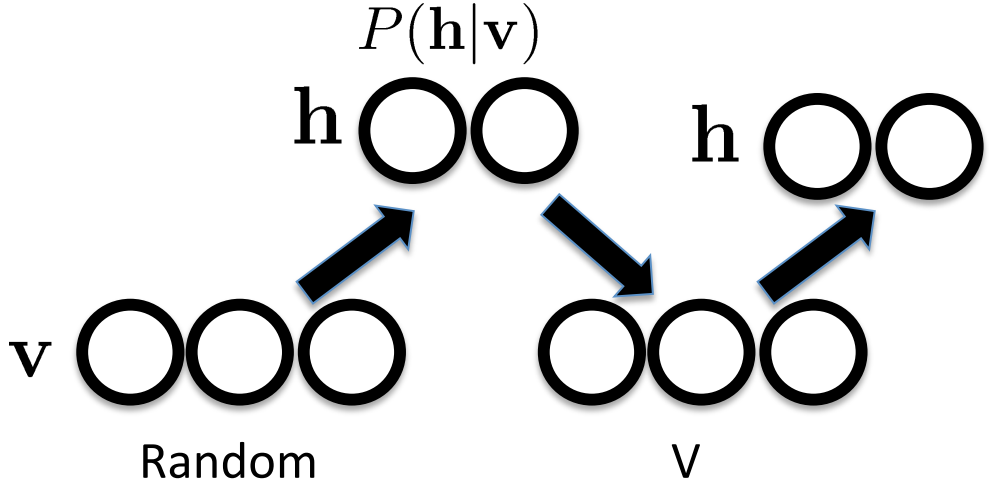


$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):

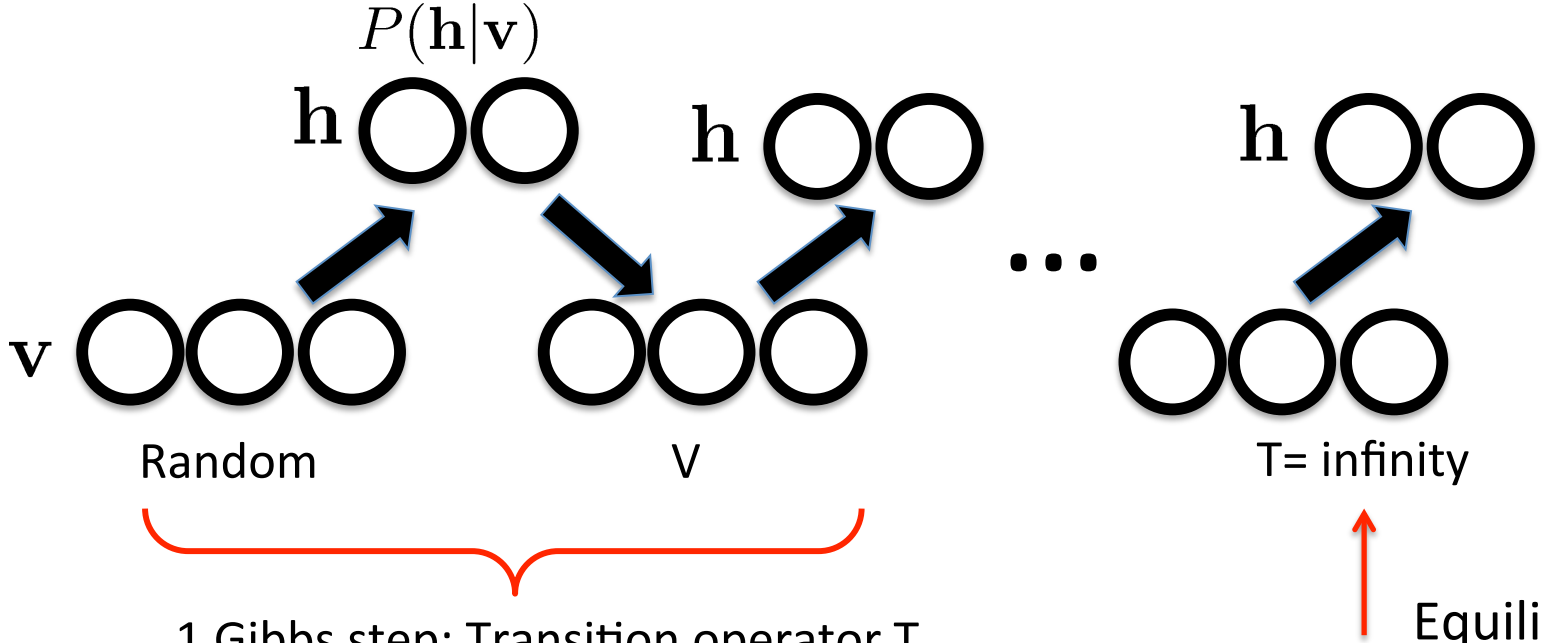


$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):

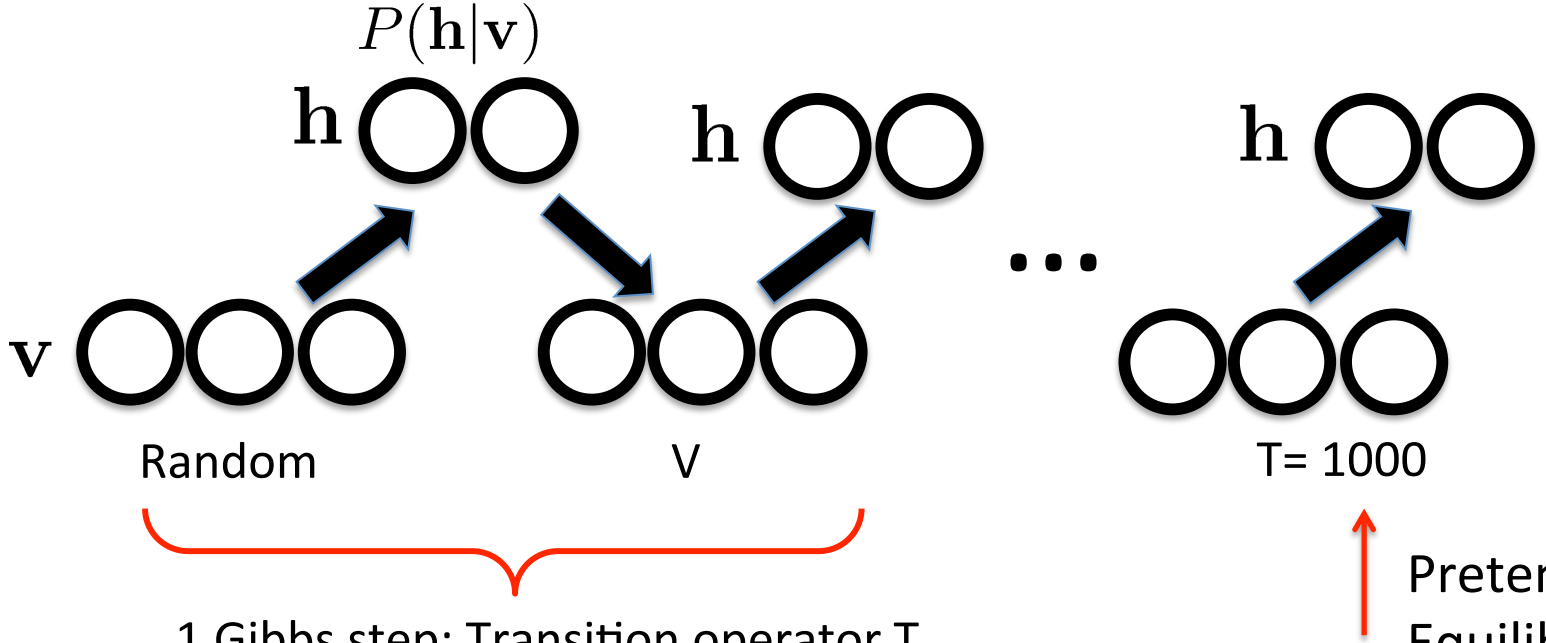


$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

Motivation: RBM Sampling

Run Markov chain (alternating Gibbs Sampling):



$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

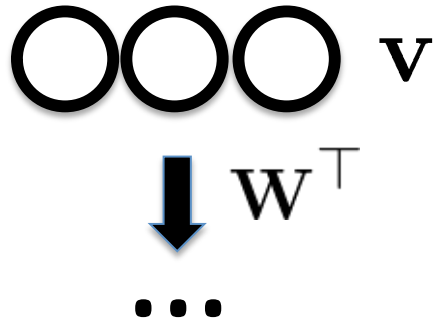
$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

Unrolled RBM as a Deep Generative Model

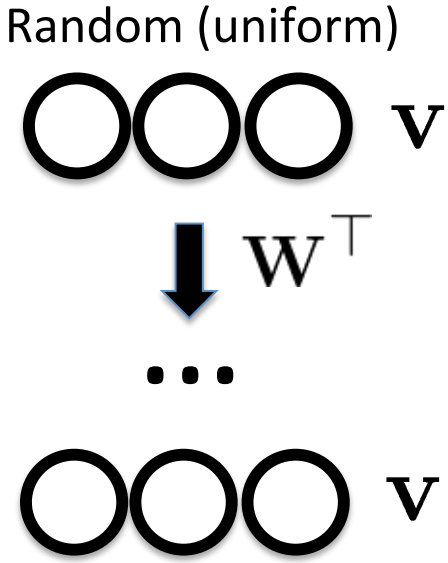
Random (uniform)

Unrolled RBM as a Deep Generative Model

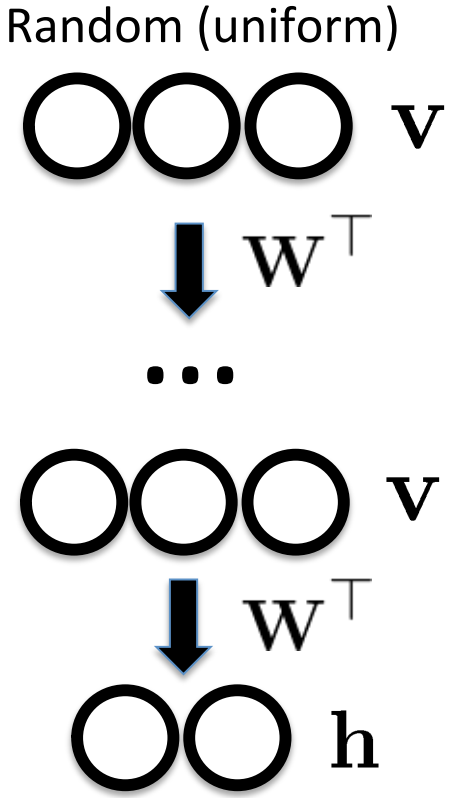
Random (uniform)



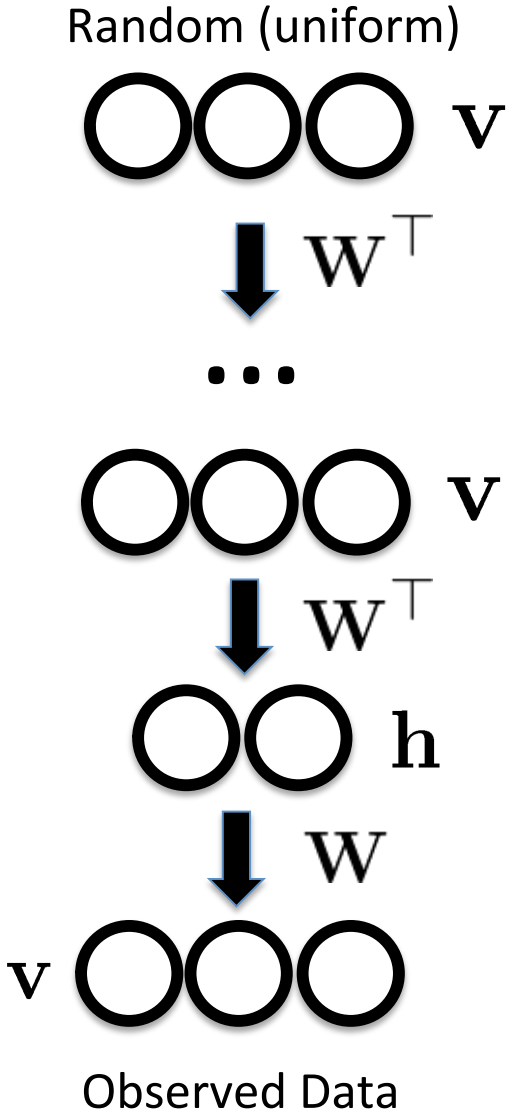
Unrolled RBM as a Deep Generative Model



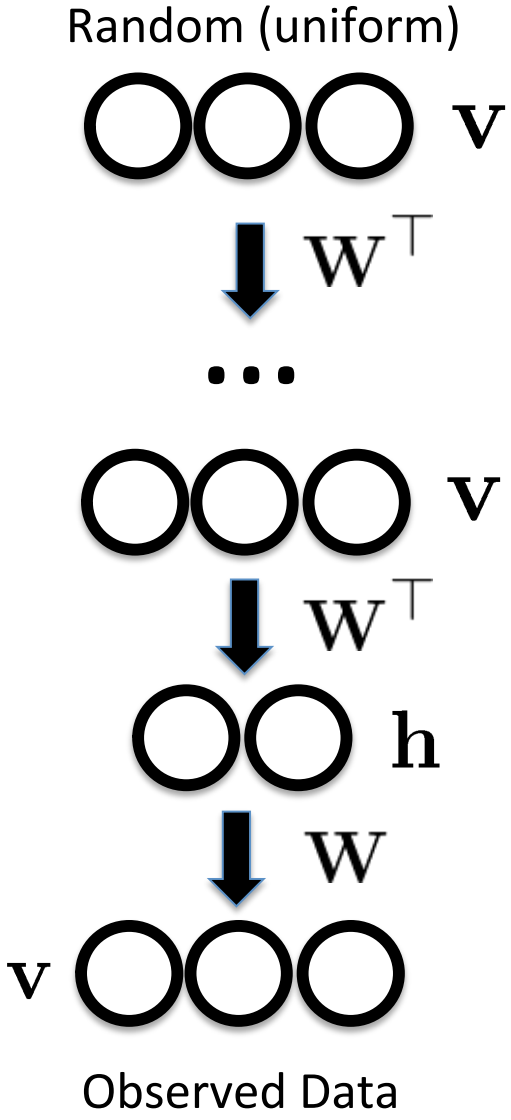
Unrolled RBM as a Deep Generative Model



Unrolled RBM as a Deep Generative Model



Unrolled RBM as a Deep Generative Model

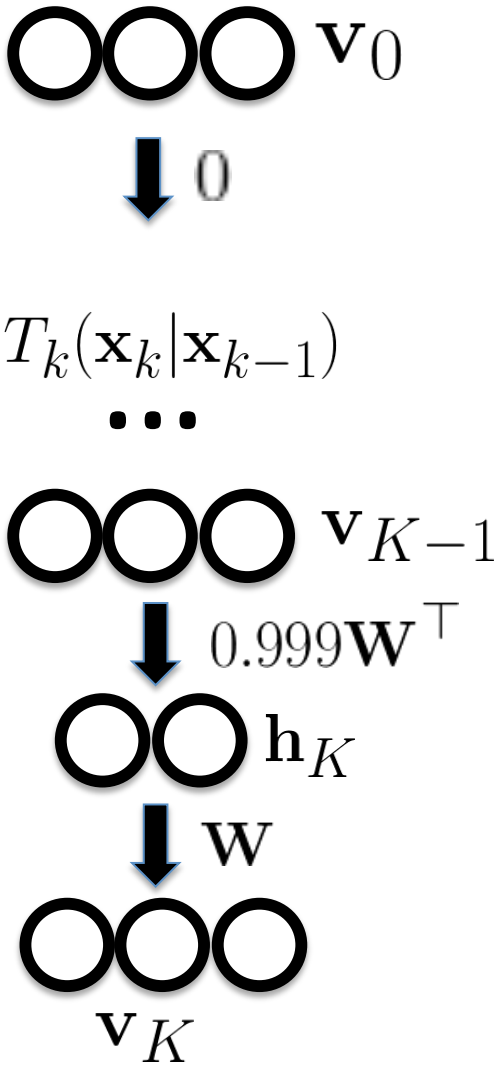


• If we use infinite number of layers, then:

$$P_{gen}(\mathbf{v}) = P_{RBM}(\mathbf{v})$$

• Otherwise, deep generative model is just an approximation to an RBM.

Reverse AIS Estimator (RAISE)



- Let us consider $\mathbf{x} = \{\mathbf{v}, \mathbf{h}\}$ where v is observed and h is unobserved.

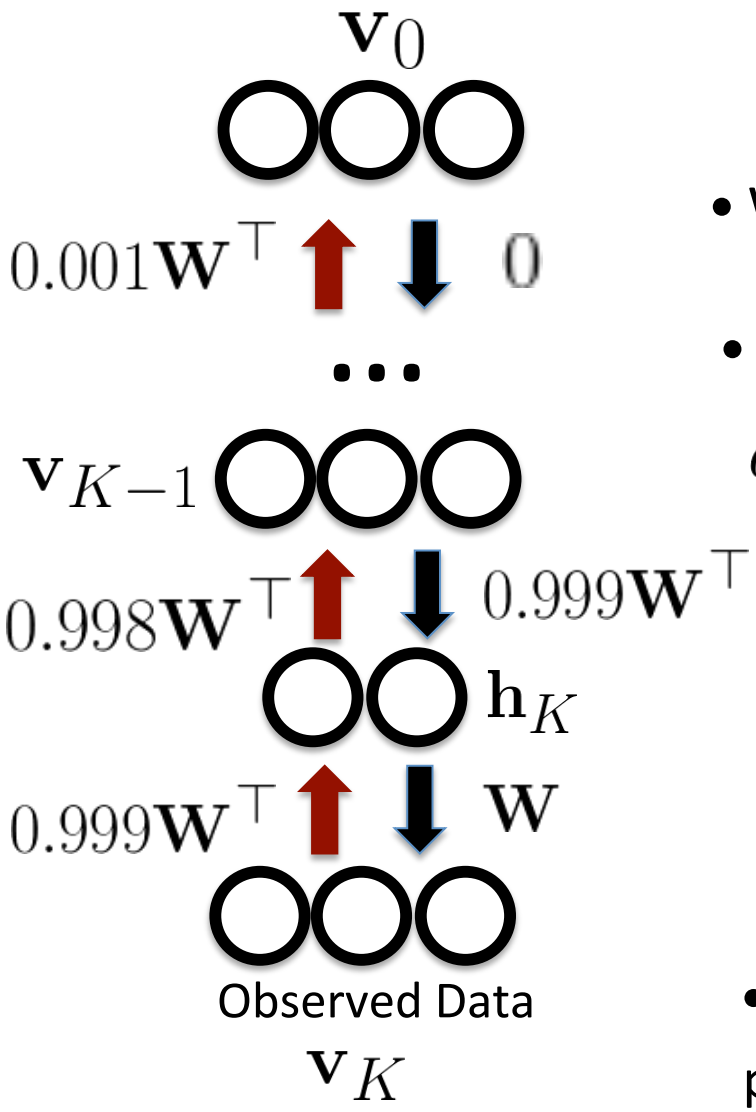
- Define the following generative process (*sequence of AIS distributions*):

$$p_{\text{fwd}}(\mathbf{x}_{0:K}) = p_0(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k|\mathbf{x}_{k-1})$$

- Generative model, that we call the **annealing model**:

$$p_{\text{ann}}(\mathbf{v}_K) = \sum_{\mathbf{x}_{0:K-1}, \mathbf{h}_K} p_{\text{fwd}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K, \mathbf{v}_K)$$

Reverse AIS Estimator (RAISE)



- As K goes to infinity:

$$P_{\text{ann}}(\mathbf{x}) = P_{\text{RBM}}(\mathbf{x})$$

- We would like to estimate $p(\mathbf{v}_{\text{test}})$.

- We use reverse chain as our proposal:

$$q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K | \mathbf{v}_{\text{test}}) = p_{\text{tgt}}(\mathbf{h}_K | \mathbf{v}_{\text{test}}) \prod_{k=1}^{K} \tilde{T}_k(\mathbf{x}_{k-1} | \mathbf{x}_k)$$

Assume tractable, which is the case for RBMs

- Can be easily extended to non-tractable posteriors, e.g. DBMs, DBNs.

Reverse AIS Estimator (RAISE)

- We now have our generative model (theoretical construct):

$$p_{\text{fwd}}(\mathbf{x}_{0:K}) = p_0(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k|\mathbf{x}_{k-1})$$

- Proposal starts at the data and melts the distribution:

$$q_{\text{rev}}(\mathbf{x}_{0:K-1}, \mathbf{h}_K|\mathbf{v}_{\text{test}}) = p_{\text{tgt}}(\mathbf{h}_K|\mathbf{v}_{\text{test}}) \prod_{k=1}^{K} \tilde{T}_k(\mathbf{x}_{k-1}|\mathbf{x}_k)$$

- We then obtain:

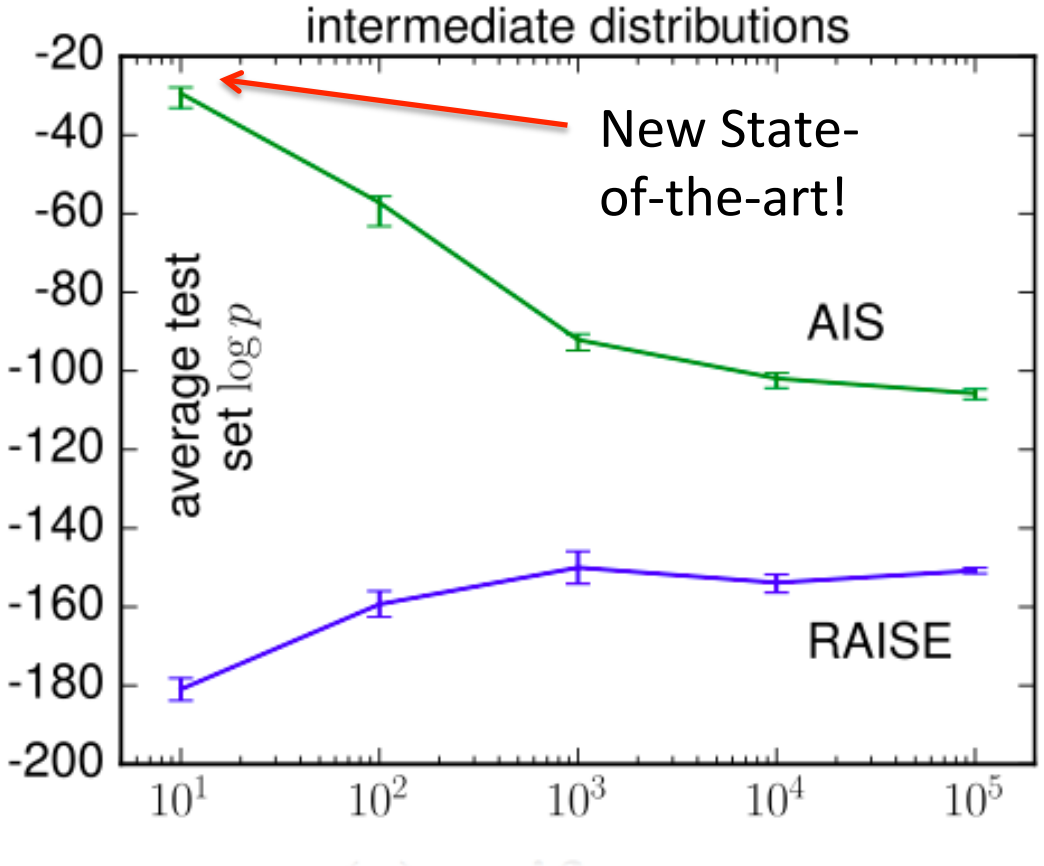
$$P_{\text{ann}}(\mathbf{v}_{\text{test}}) = \mathbb{E}_{q_{\text{rev}}}\left[\frac{f_{\text{fwd}}}{q_{\text{rev}}}\right]$$

$$= \mathbb{E}_{q_{\text{rev}}}\left[\frac{f_{\text{tgt}}(\mathbf{v}_{\text{test}})}{\mathcal{Z}_0} \prod_{k=1}^{K-1} \frac{f_k(\mathbf{x}_k)}{f_{k+1}(\mathbf{x}_k)}\right] = \mathbb{E}_{q_{\text{rev}}}[w]$$

- Tends to underestimate rather than overestimate log-probs!

MNIST

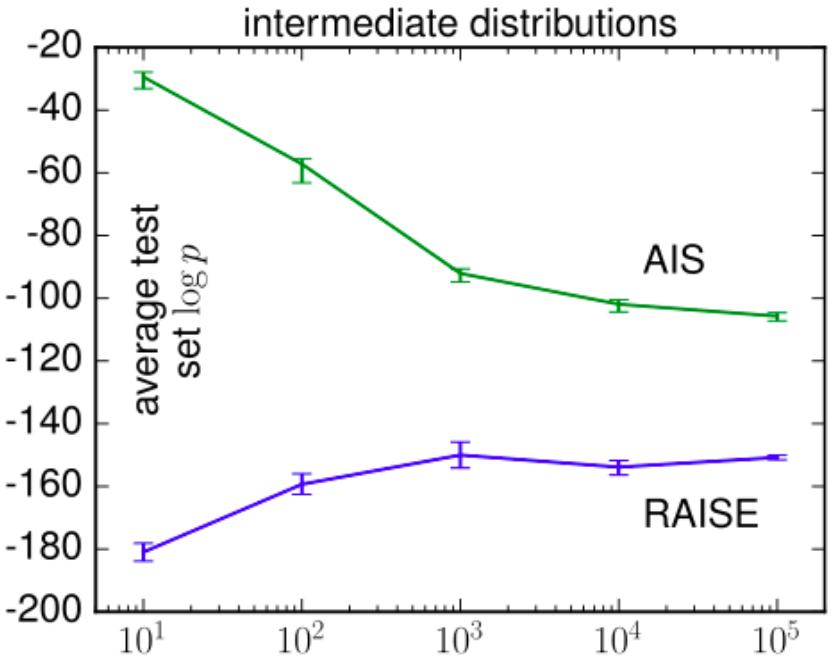
- RBM with 500 hidden units trained on MNIST with CD1.
- **Initial distribution is uniform**: AIS with geometric averages is off by 20 nats, even when using 100K intermediate distributions!



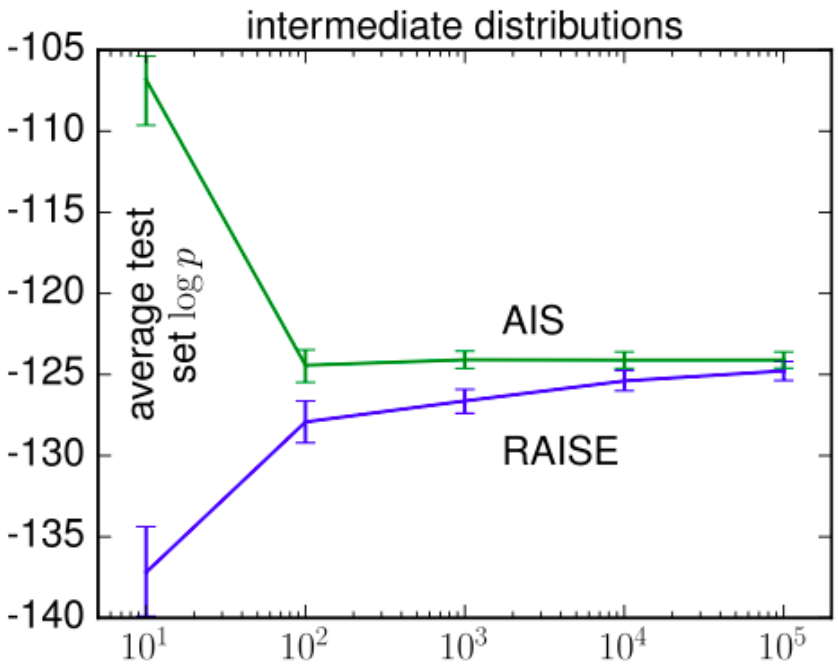
MNIST

- The only different is the choice of initial distribution.

Uniform

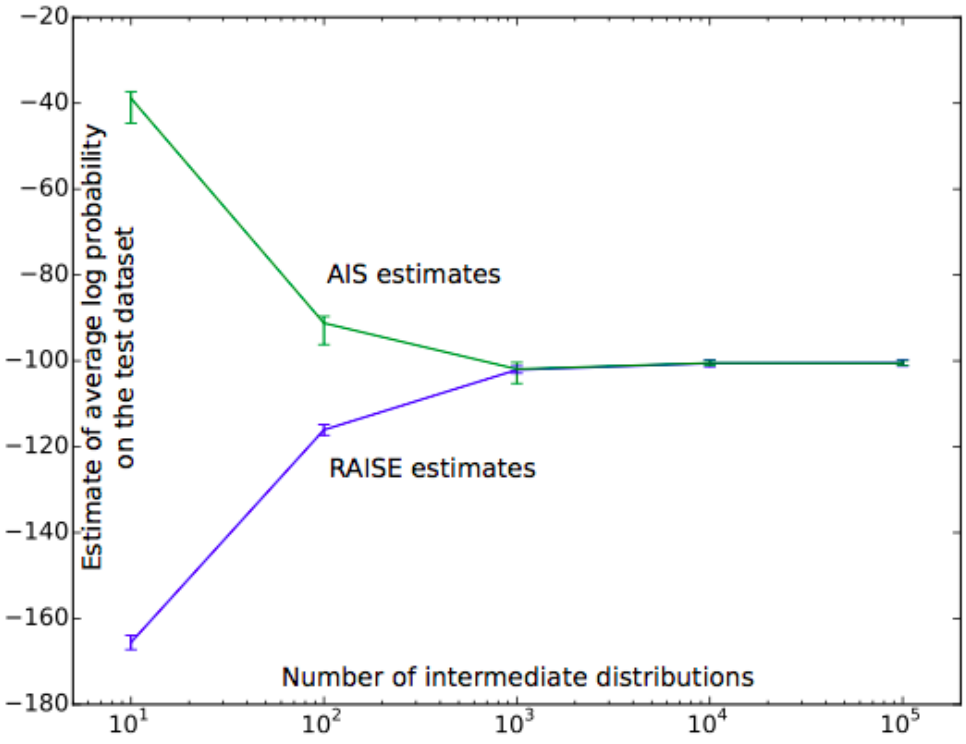


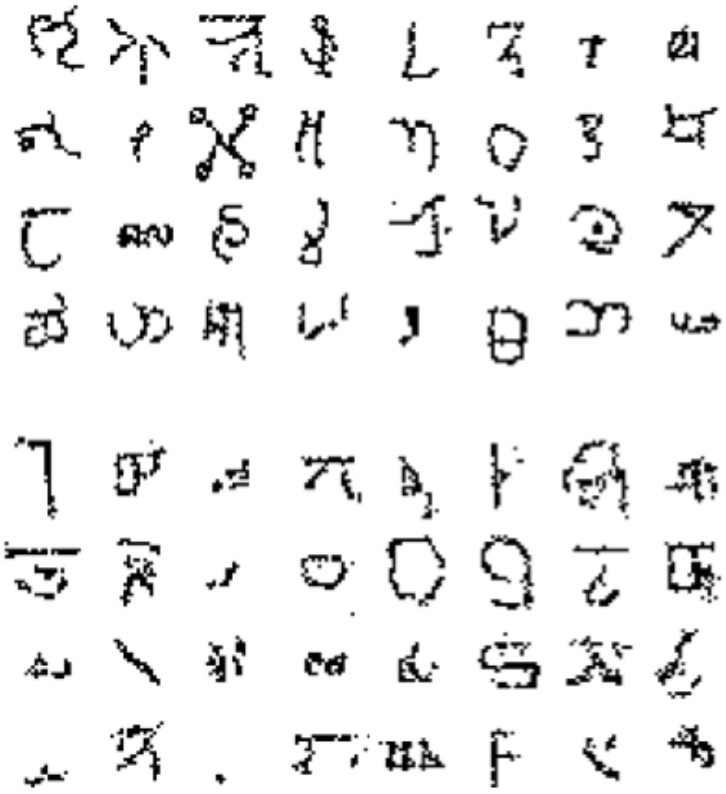
Base rates



Omniglot Dataset

- RBM with 500 hidden units trained on Omniglot with PCD.





MNIST and Omniglot Results

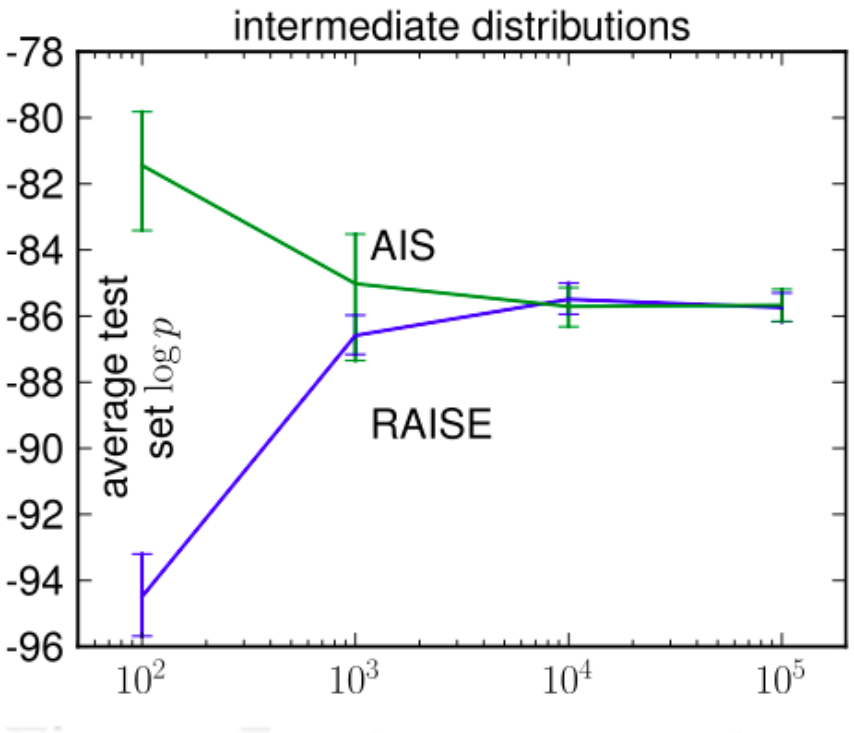
Model	exact	CSL	RAISE	uniform AIS	gap
mnistCD1-20	-164.50	-185.74	-165.33	-164.51	0.82
mnistPCD-20	-150.11	-152.13	-150.58	-150.04	0.54
mnistCD1-500	—	-566.91	-150.78	-106.52	44.26
mnistPCD-500	—	-138.76	-101.07	-99.99	1.08
mnistCD25-500	—	-145.26	-88.51	-86.42	2.09
omniPCD-1000	—	-144.25	-100.47	-100.45	0.02

- RAISE errs on the side of underestimating the log-likelihood.
- Note that the gap is very small!
- CSL: Conservative Sampling-based Log-likelihood (CSL) estimator of Bengio et. al.

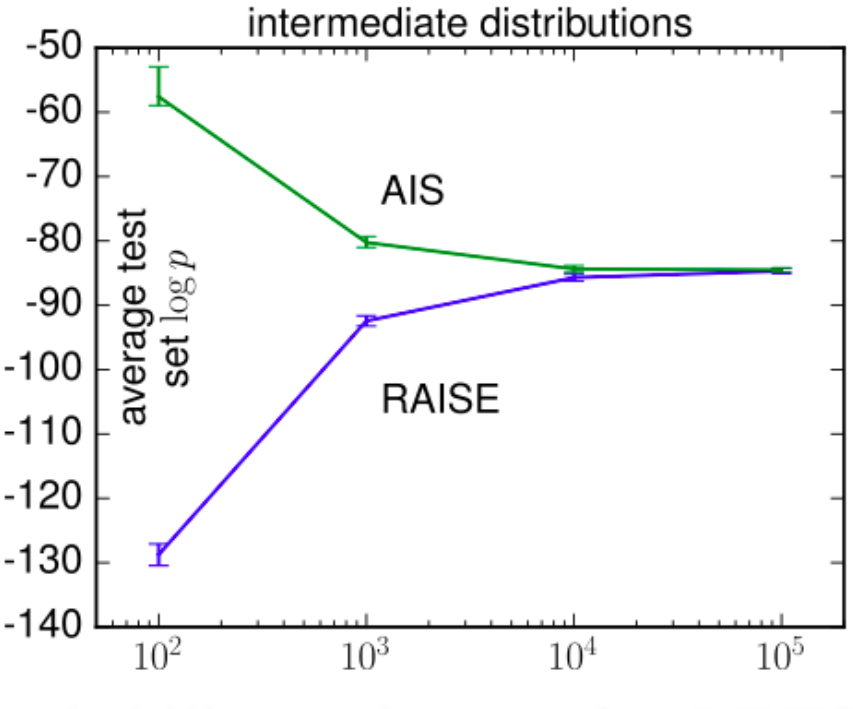
Y. Bengio, L. Yao, and K. Cho. Bounding the test log-likelihood of generative models.

DBMs and DBNs

Deep Boltzmann Machine



Deep Believe Network



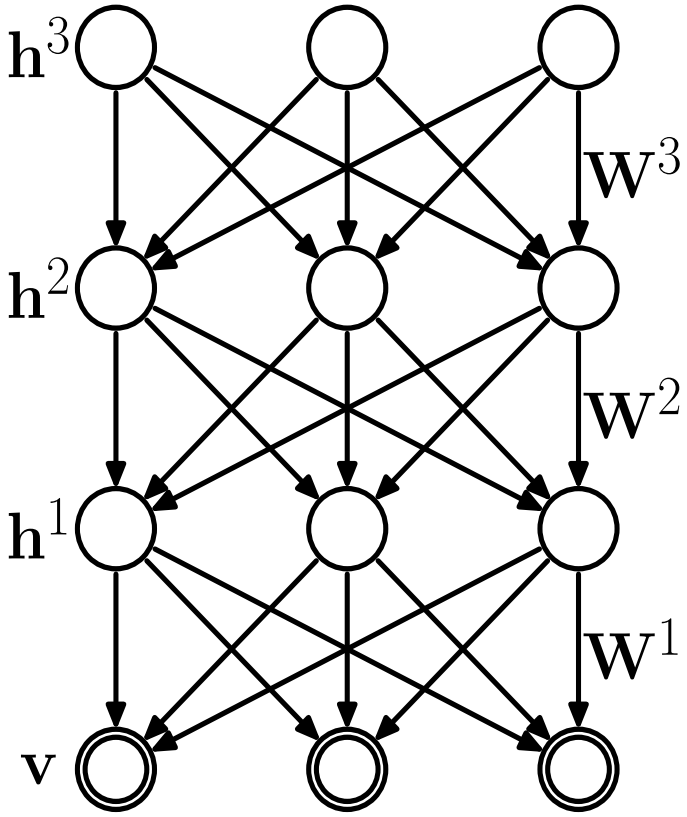
Helmholtz Machines

Approximate Inference

$Q(\mathbf{h}^3|\mathbf{h}^2)$

$Q(\mathbf{h}^2|\mathbf{h}^1)$

$Q(\mathbf{h}^1|\mathbf{v})$



Input

Generative Process

$P(\mathbf{h}^2, \mathbf{h}^3)$

$P(\mathbf{h}^1|\mathbf{h}^2)$

$P(\mathbf{v}|\mathbf{h}^1)$

Yura Burda

Conclusions

- RAISE produces accurate, yet conservative, estimates of log-probabilities for RBMs, DBMs, and DBNs.

- Using both RAISE and AIS, one can judge the accuracy of one's results by measuring the agreement of the two estimators .

- RAISE is simple to implement (same as AIS), so it gives a simple and practical way to evaluate MRF test log-probabilities.

- These ideas serve as a starting point for learning very deep generative models!

End of Part 1