Deep Learning

Russ Salakhutdinov

Department of Computer Science
Department of Statistical Sciences
University of Toronto

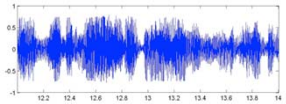
Mining for Structure

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

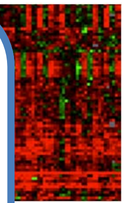
Images & Video

Text & Language

Speech & Audio



Gene Expression

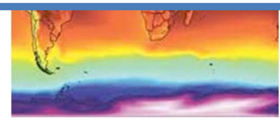


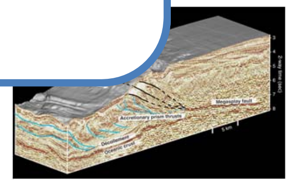
Prod

Reco

l Data

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



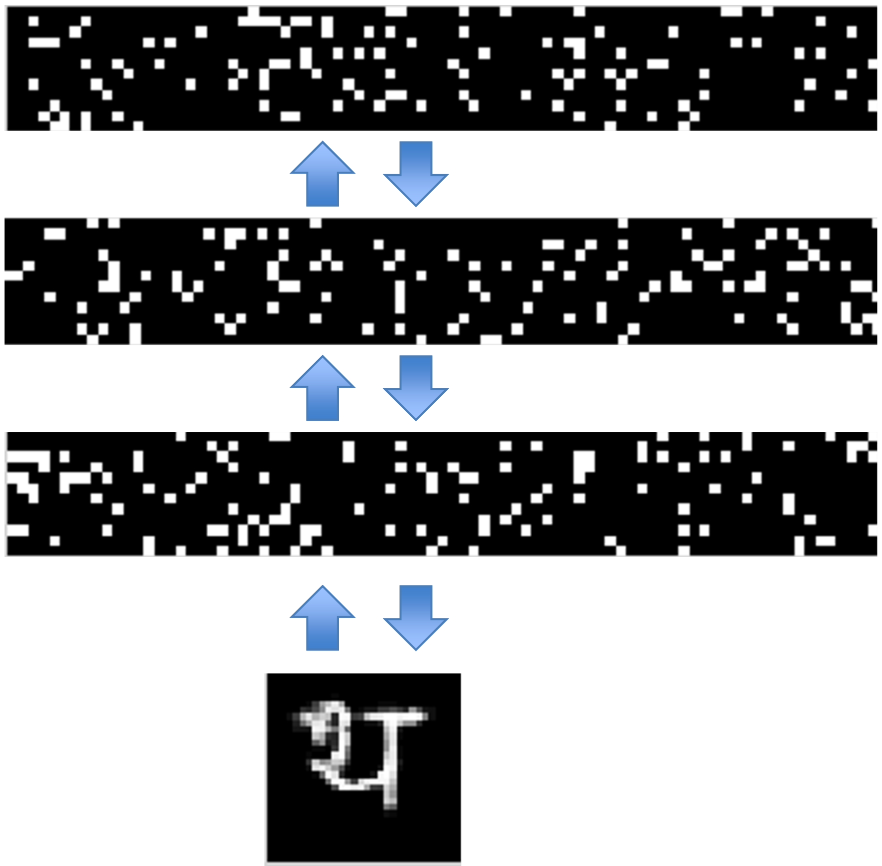


Mostly Unlabeled

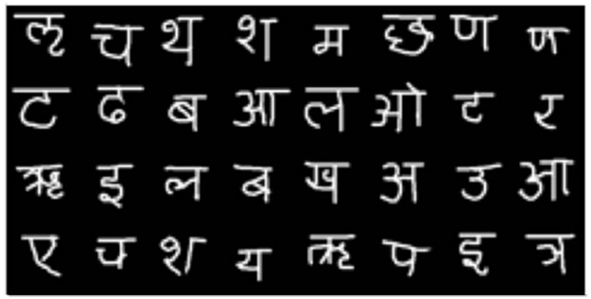
- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

Deep Boltzmann Machine

Sanskrit



Model P(image)

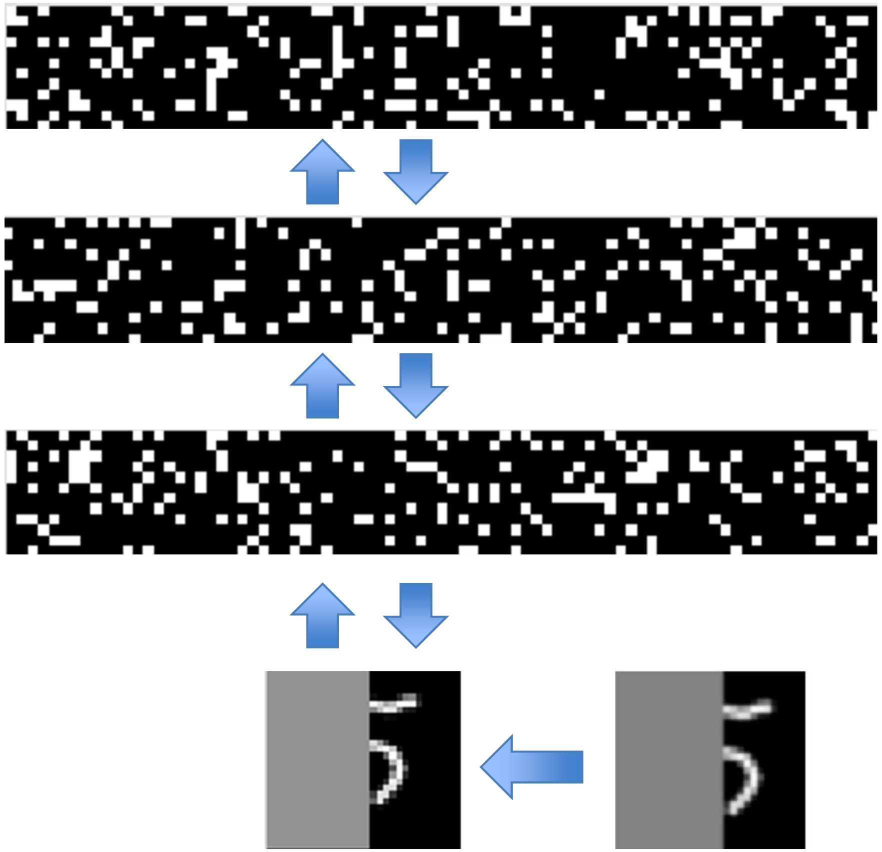


25,000 characters from 50 alphabets around the world.

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

Bernoulli Markov Random Field

Deep Boltzmann Machine

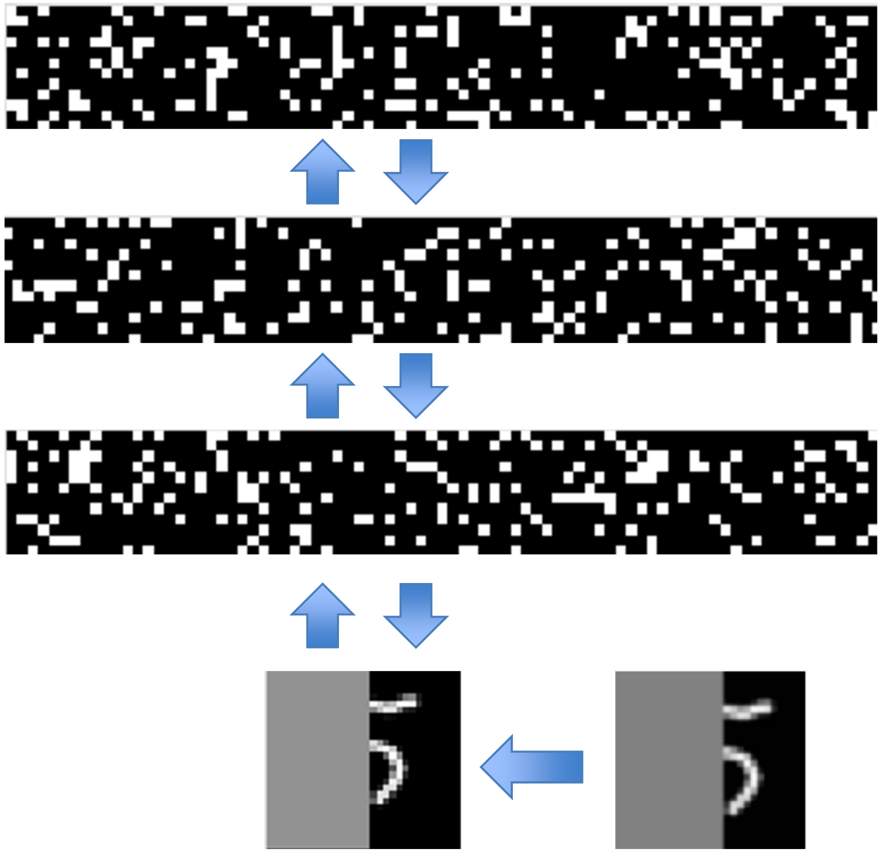


P(image|partial image)

Conditional Simulation

Bernoulli Markov Random Field

Deep Boltzmann Machine



Conditional
Simulation

P(image|partial image)

Why so difficult?

28

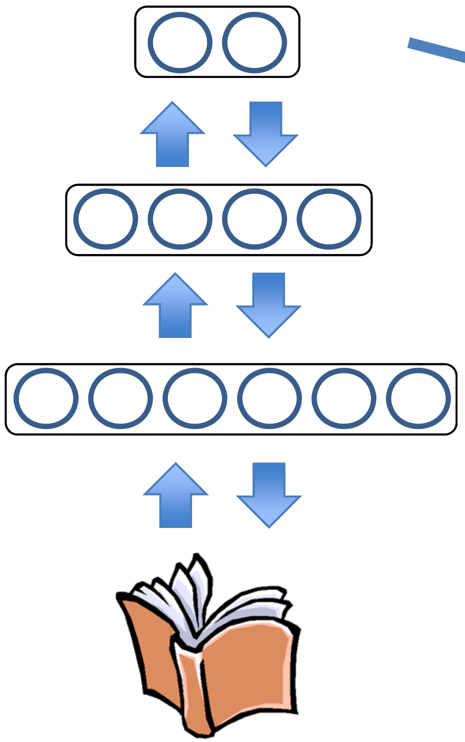
28

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

Bernoulli Markov Random Field

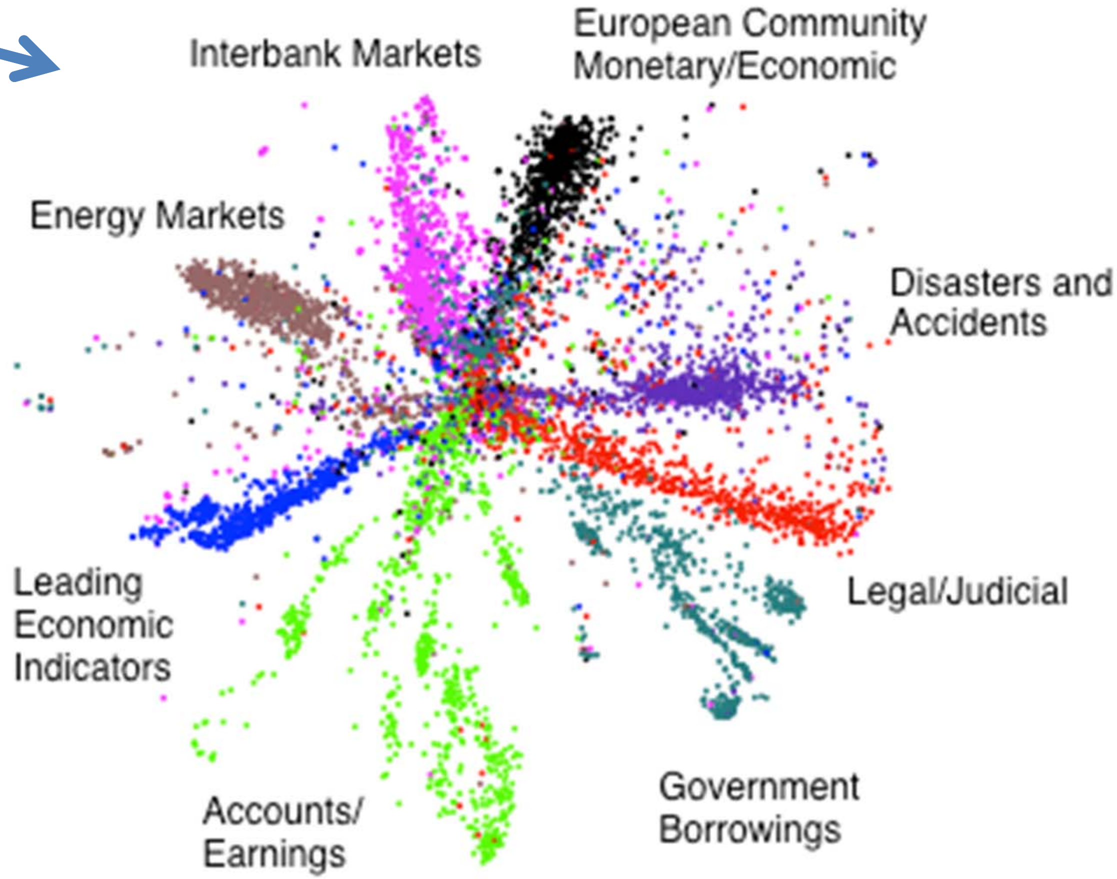
Deep Generative Model

Model P(document)

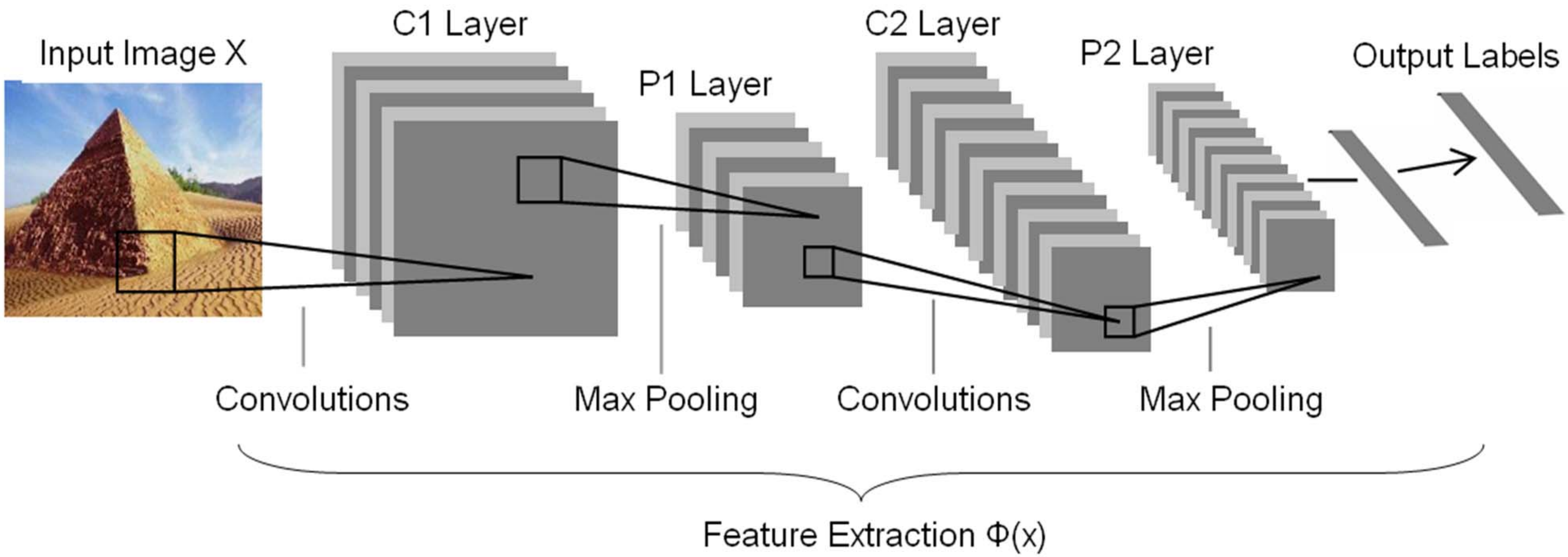


Bag of words

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



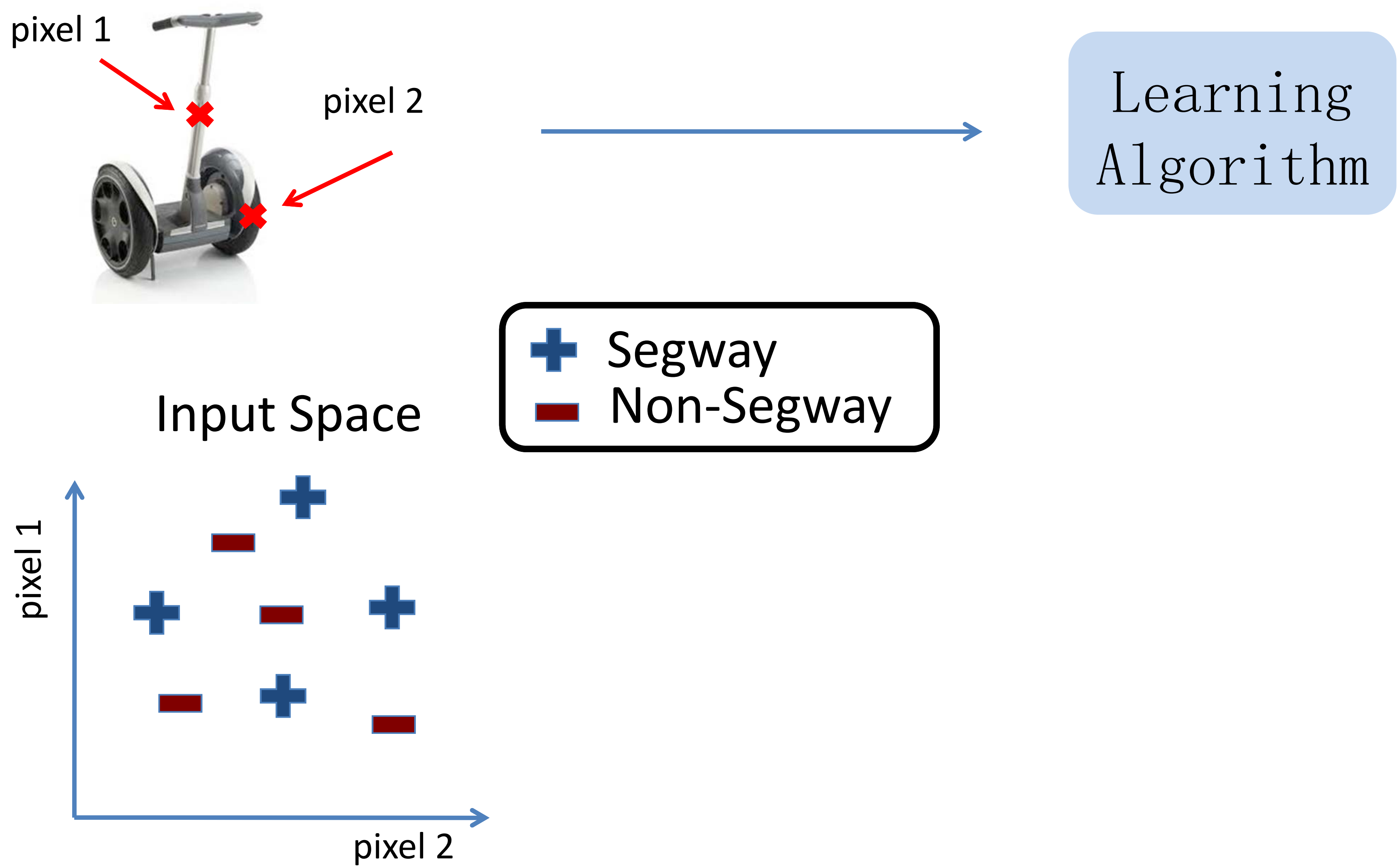
Convolutinal Deep Models for Image Recognition



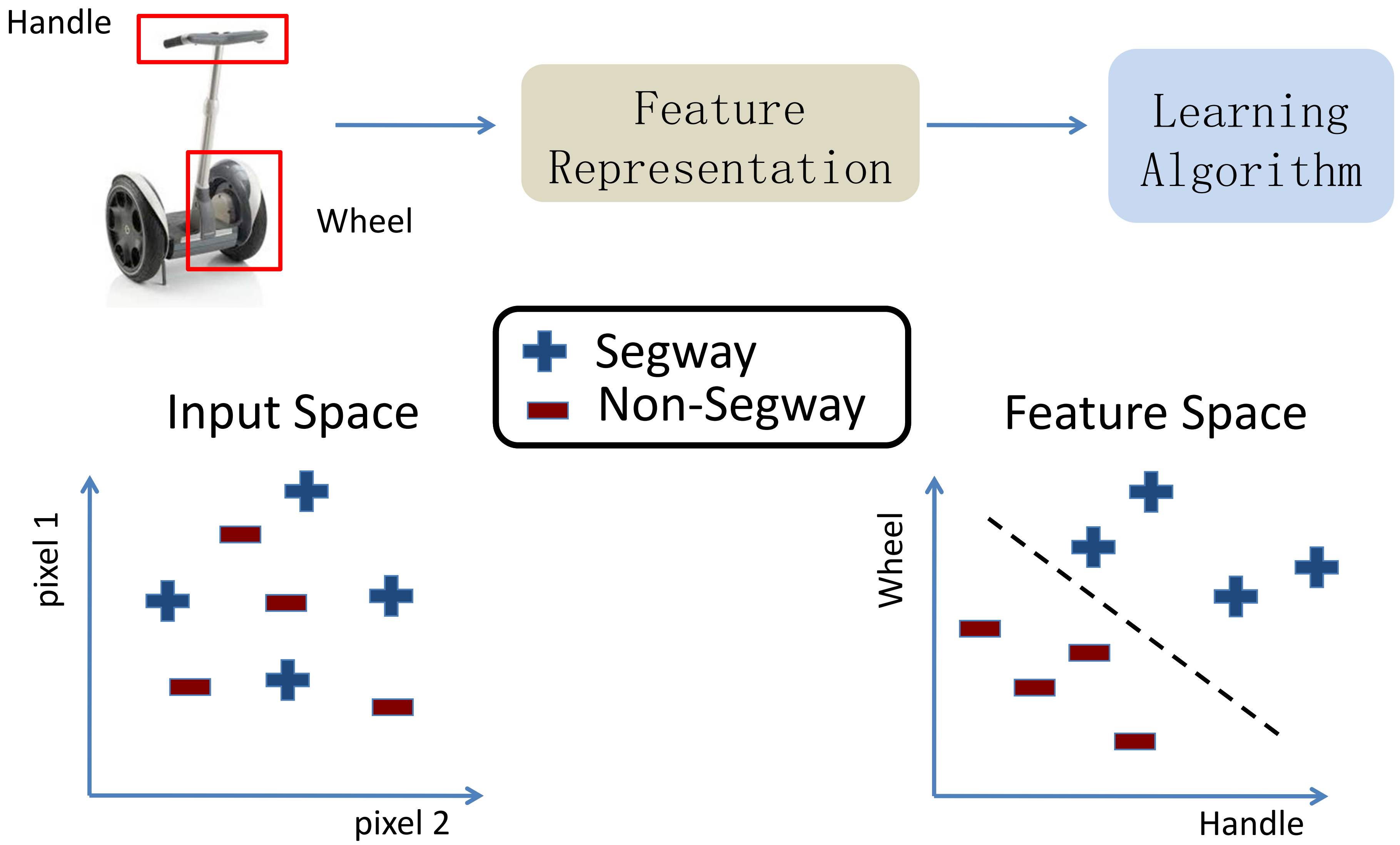
- Learning multiple layers of representation.

(LeCun, 1992)

Learning Feature Representations



Learning Feature Representations

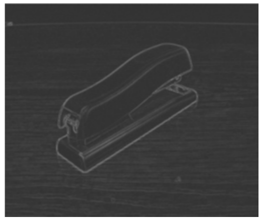


How is computer perception done?

Input Data → Low-level features → Learning Algorithm

Object detection

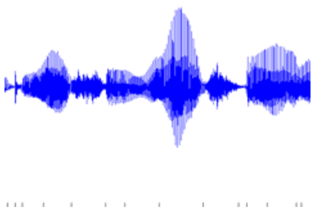
Image



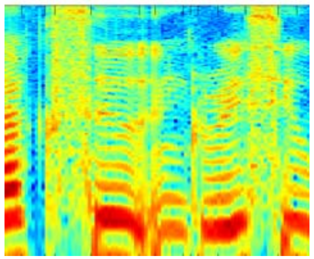
Low-level vision features

Recognition

Audio classification



Audio

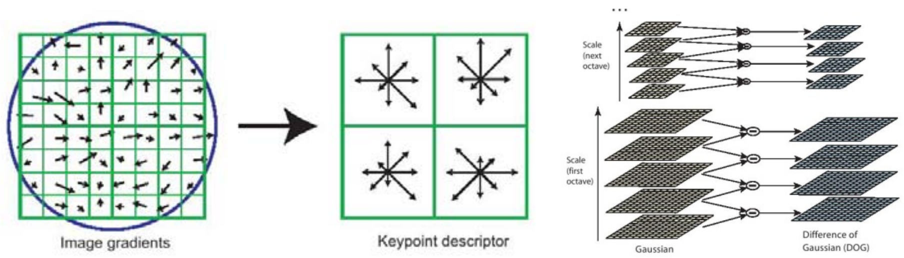


Low-level audio features

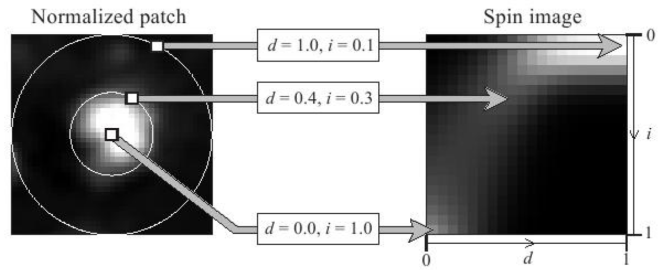
Speaker identification

Slide Credit: Honglak Lee

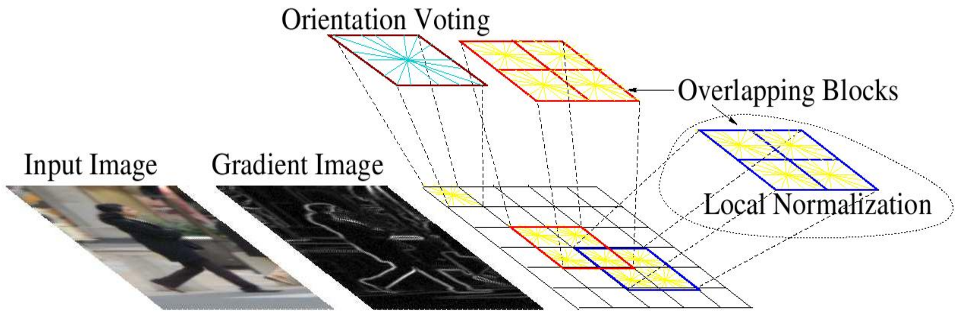
Computer vision features



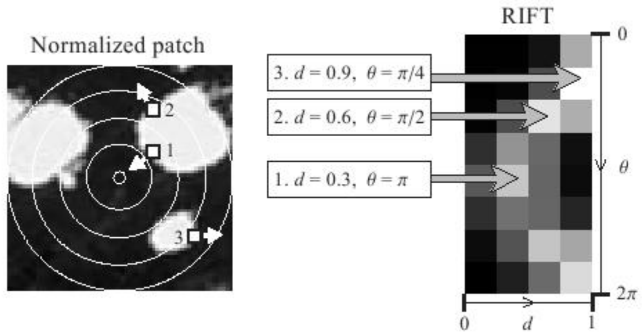
SIFT



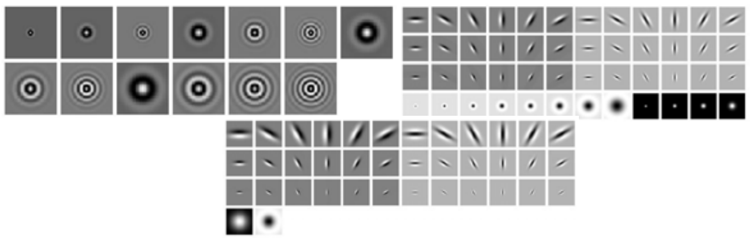
Spin image



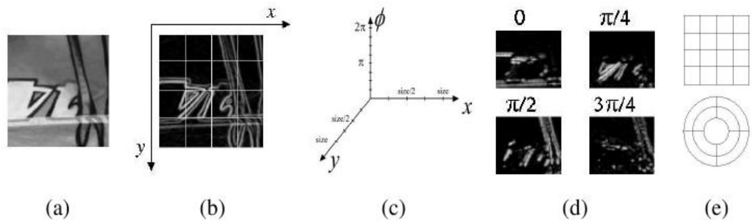
HoG



RIFT



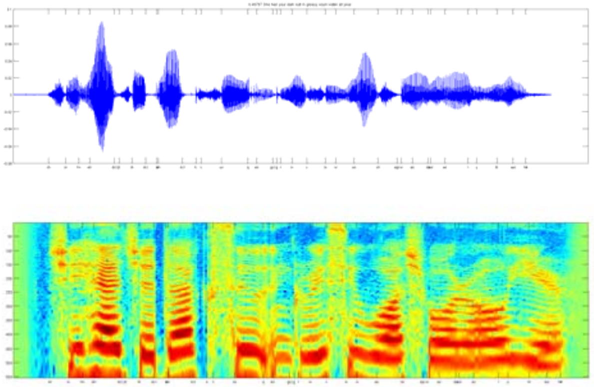
Textons



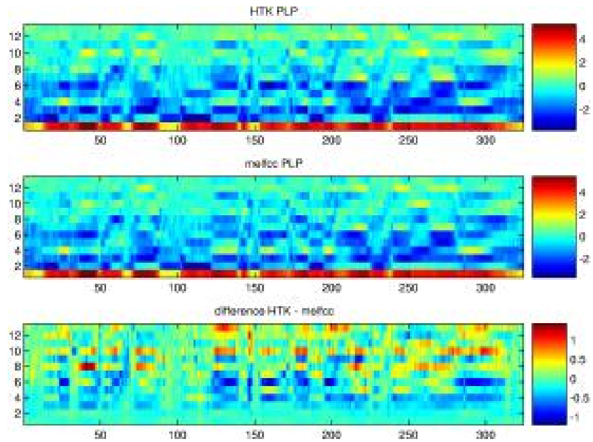
GLOH

Slide Credit: Honglak Lee

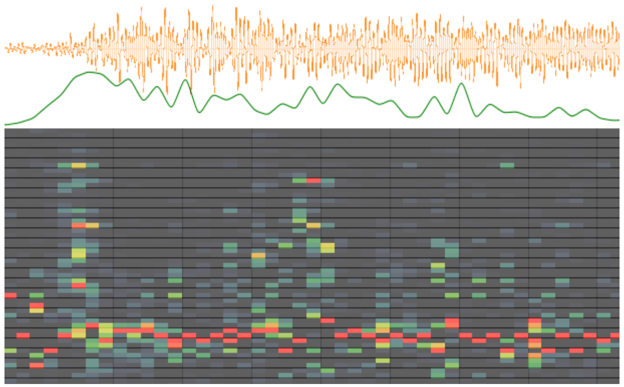
Audio features



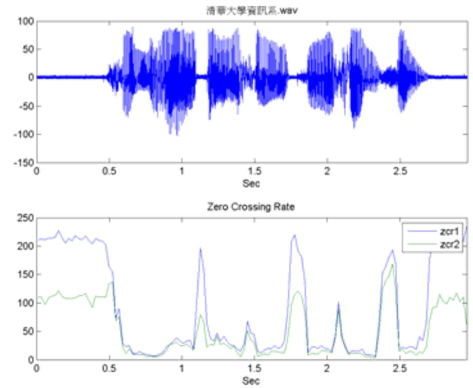
Spectrogram



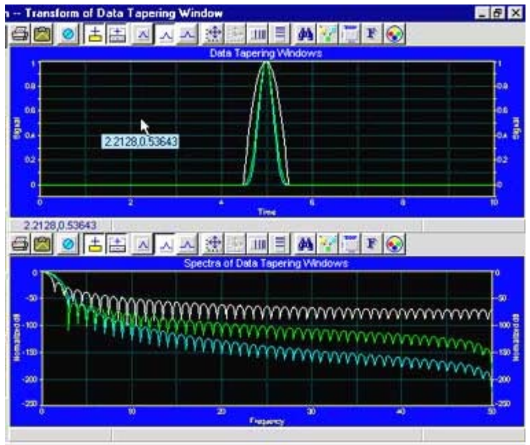
MFCC



Flux

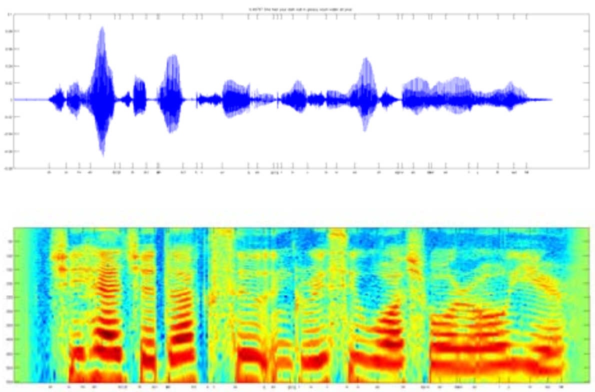


ZCR

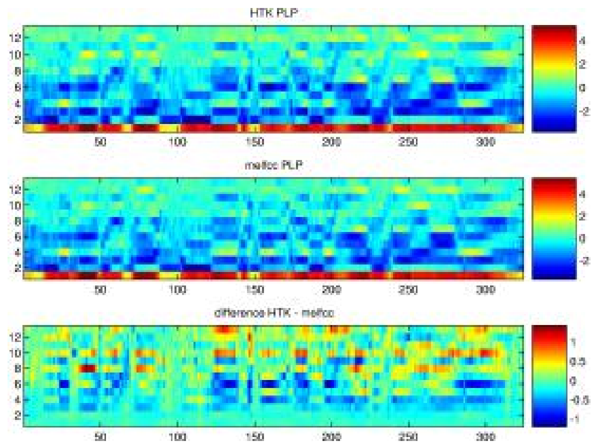


Rolloff

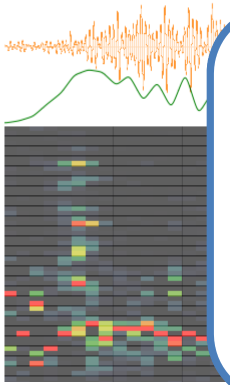
Audio features



Spectrogram



MFCC



Feature Learning: Can we learn meaningful features from unlabeled, partially labeled data?

Flux

ZCR

Rolloff

Talk Roadmap

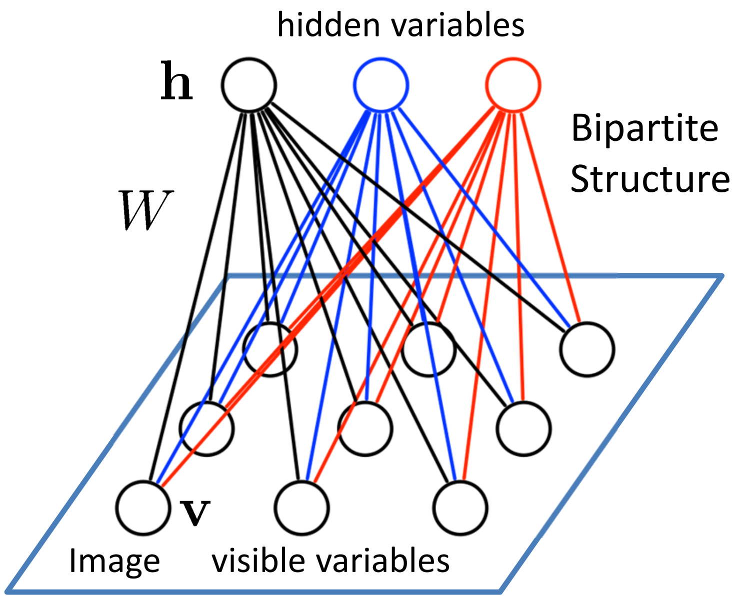
Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Advanced Deep Models.

- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning

Restricted Boltzmann Machines



- Undirected bipartite graphical model

- Stochastic binary visible variables:

$$\mathbf{v} \in \{0,1\}^D$$

- Stochastic binary hidden variables:

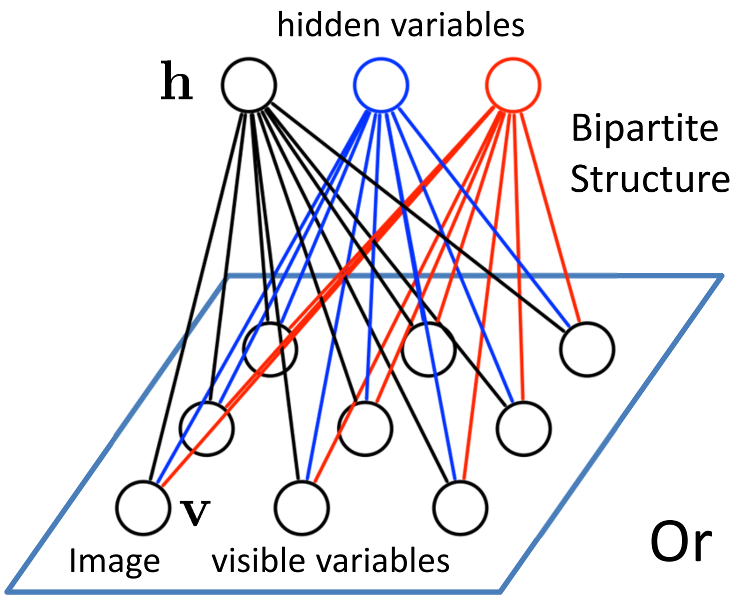
$$\mathbf{h} \in \{0,1\}^F$$

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.

Restricted Boltzmann Machines



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

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big)$$

Or

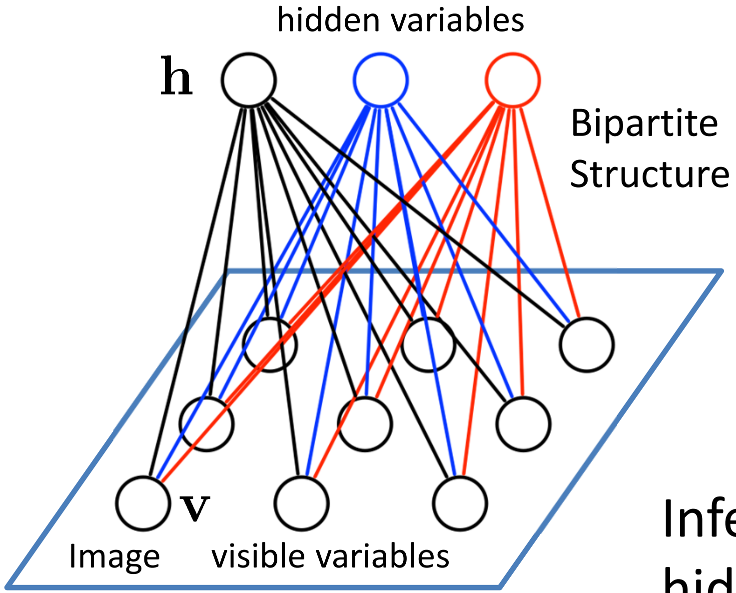
Pair-wise **Unary**

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

$$\mathcal{Z}(\theta) = \sum_{\mathbf{h},\mathbf{v}} \exp\big(-E(\mathbf{v}, \mathbf{h}; \theta)\big)$$

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

Restricted Boltzmann Machines



Restricted: No interaction between hidden variables

Inferring the distribution over the hidden variables is easy:

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

Factorizes: Easy to compute

Similarly:

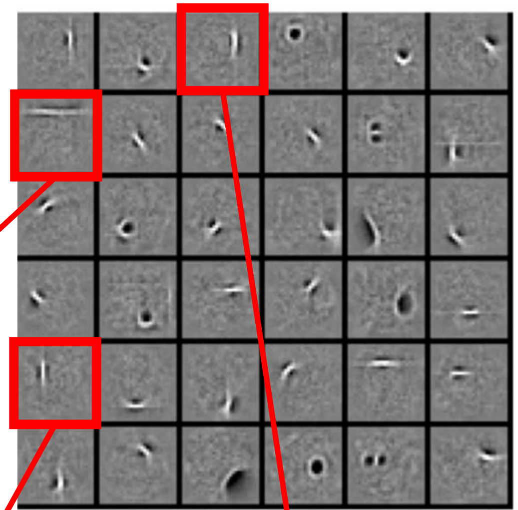
$$P(\mathbf{v}|\mathbf{h}) = \prod_i P(v_i|\mathbf{h}) \quad P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$

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

Learning Features

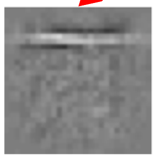
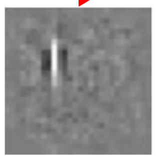
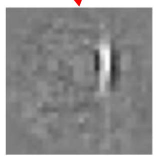
Observed Data
Subset of 25,000 characters

Learned W: "edges"
Subset of 1000 features



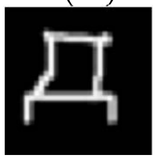
Most hidden variables are off

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

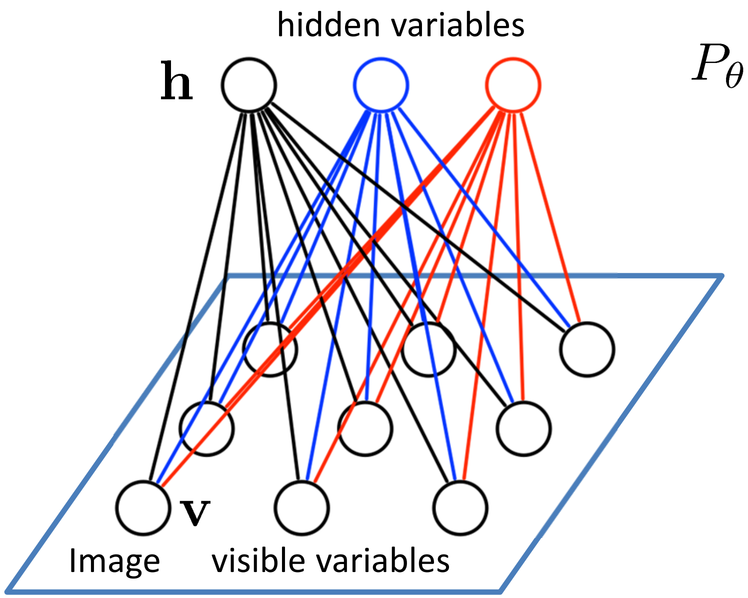
 $= \sigma\Big(0.99 \times$  $+ \; 0.97 \times$  $+ \; 0.82 \times$  $\ldots\Big)$

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

Logistic Function: Suitable for modeling binary images

Represent:  as $P(\mathbf{h}|\mathbf{v}) = [0,\ 0,\ 0.82,\ 0,\ 0,\ 0.99,\ 0,\ 0\ \ldots\ ]$

Model Learning



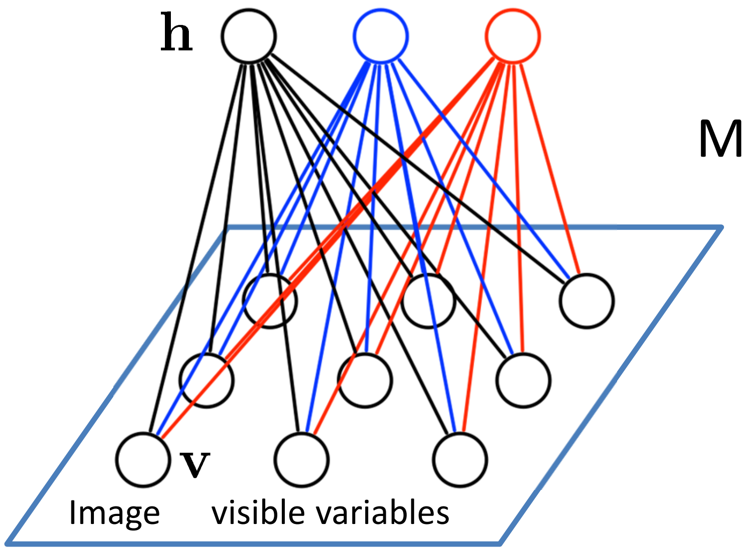
$$P_\theta(\mathbf{v}) = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}\right]$$

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 (penalized) log-likelihood objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)}) - \underbrace{\frac{\lambda}{N} ||W||_F^2}_{\text{Regularization}}$$

Model Learning



Maximize (penalized) log-likelihood objective:

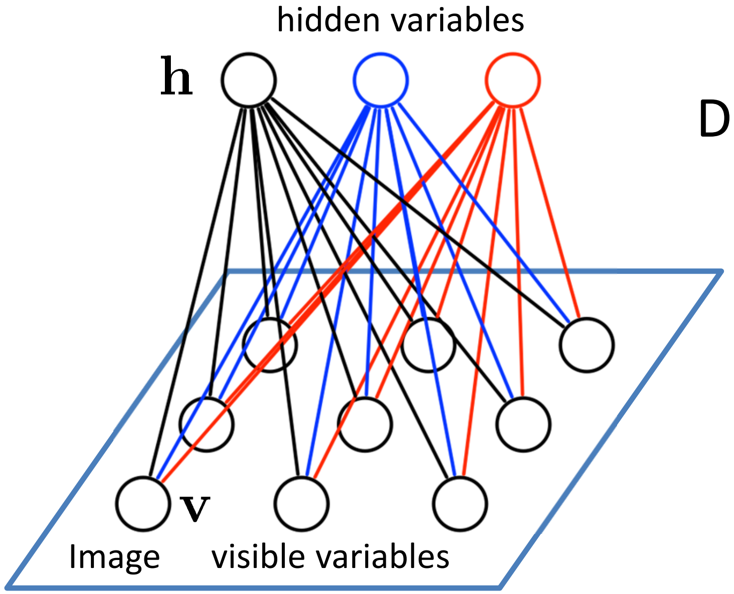
$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N}\log P_\theta(\mathbf{v}^{(n)}) - \underbrace{\frac{\lambda}{N}||W||_F^2}_{\text{Regularization}}$$

Derivative of the log-likelihood:

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial W_{ij}}\log\left(\sum_{\mathbf{h}}\exp\left[\mathbf{v}^{(n)\top}W\mathbf{h}+\mathbf{a}^\top\mathbf{h}+\mathbf{b}^\top\mathbf{v}^{(n)}\right]\right)-\frac{\partial}{\partial W_{ij}}\log\mathcal{Z}(\theta)-\frac{2\lambda}{N}W_{ij}$$

$$= \mathrm{E}_{P_{data}}[v_i h_j] - \mathrm{E}_{P_\theta}[v_i h_j] - \frac{2\lambda}{N}W_{ij}$$

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

Approximate Learning

- An approximation to the gradient of the log-likelihood objective:

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

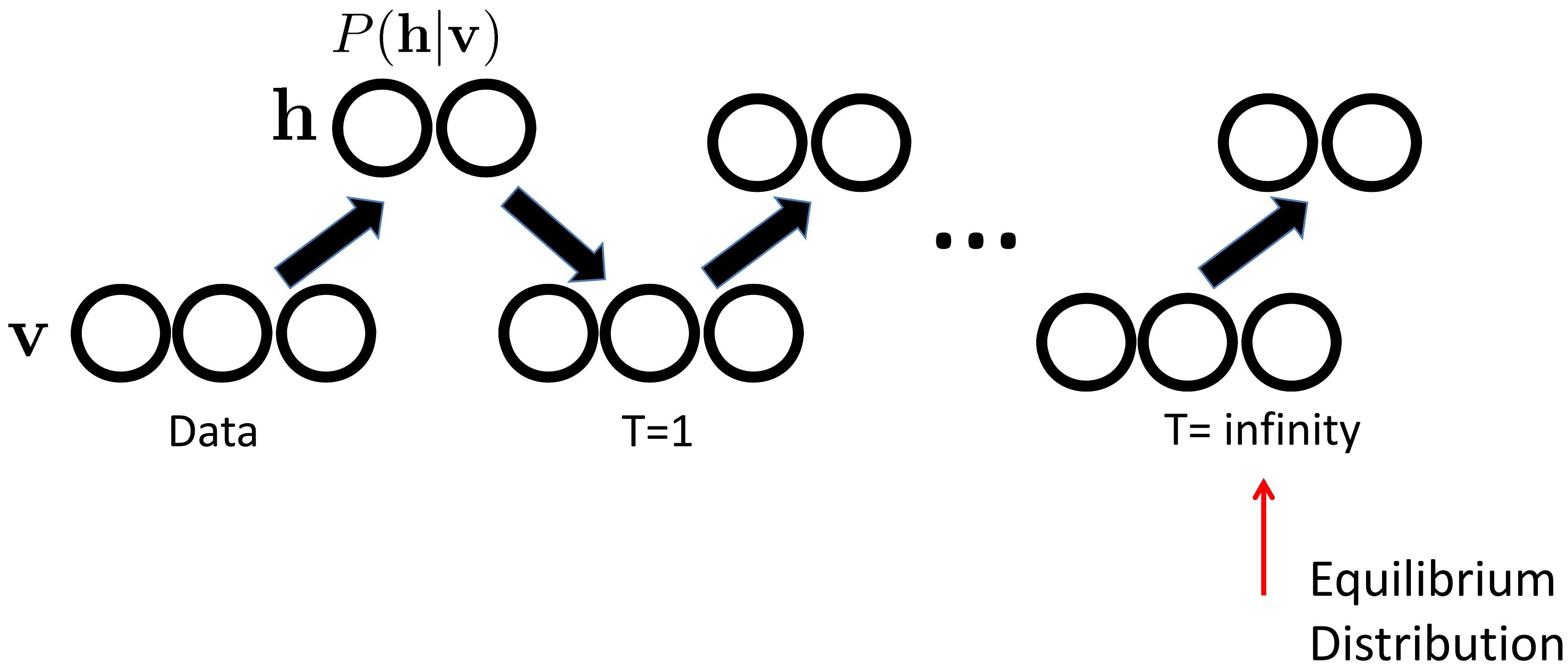
$$\sum_{\mathbf{v},\mathbf{h}} v_i h_j P_\theta(\mathbf{v},\mathbf{h})$$

- Replace the average over all possible input configurations by samples.
- Run MCMC chain (Gibbs sampling) starting from the observed examples.

> - Initialize $v^0 = v$
> - Sample h^0 from $P(h \mid v^0)$
> - For t=1:T
>   - Sample v^t from $P(v \mid h^{t-1})$
>   - Sample h^t from $P(h \mid v^t)$

Approximate ML Learning for RBMs

Run Markov chain (alternating Gibbs Sampling):

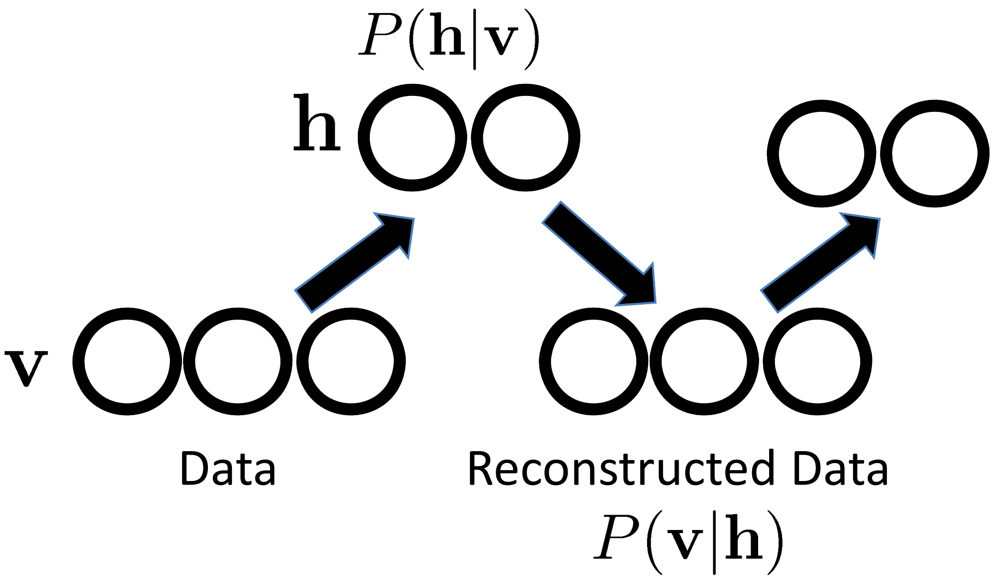


$$P(\mathbf{h}|\mathbf{v}) = \prod_j P(h_j|\mathbf{v}) \quad P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

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

Contrastive Divergence

A quick way to learn RBM:



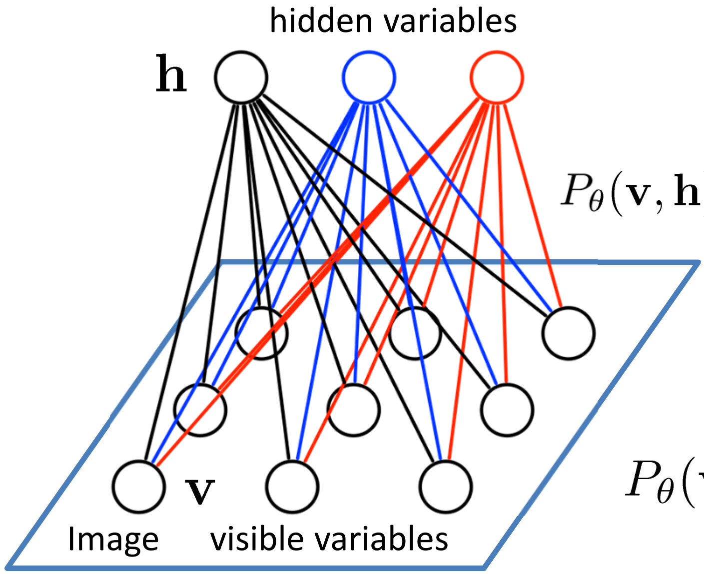
- Start with a training vector on the visible units.
- Update all the hidden units in parallel.
- Update the all the visible units in parallel to get a "reconstruction".
- Update the hidden units again.

Update model parameters:

$$\Delta W_{ij} = \mathrm{E}_{P_{data}}[v_i h_j] - \mathrm{E}_{P_1}[v_i h_j]$$

Implementation: ~10 lines of Matlab code.

RBMs for Real-valued Data



Pair-wise **Unary**

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

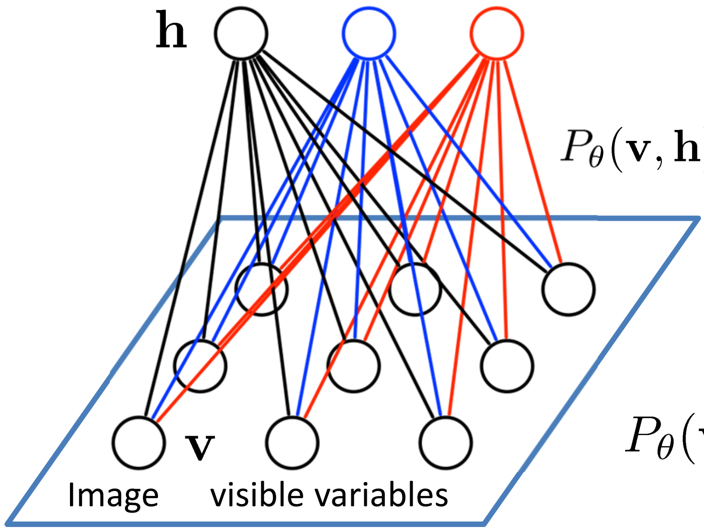
$$\theta = \{W, a, b\}$$

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

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.

RBMs for Real-valued Data



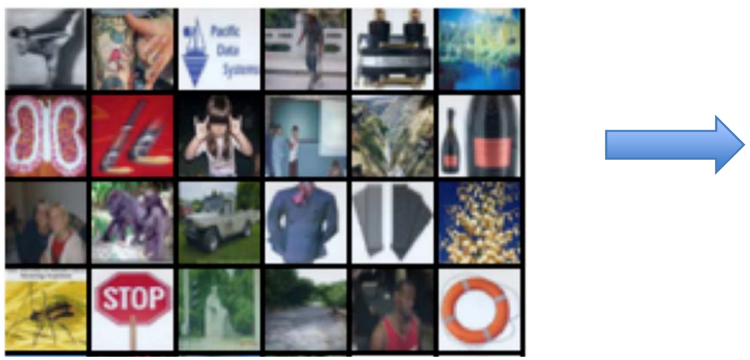
Pair-wise / Unary

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

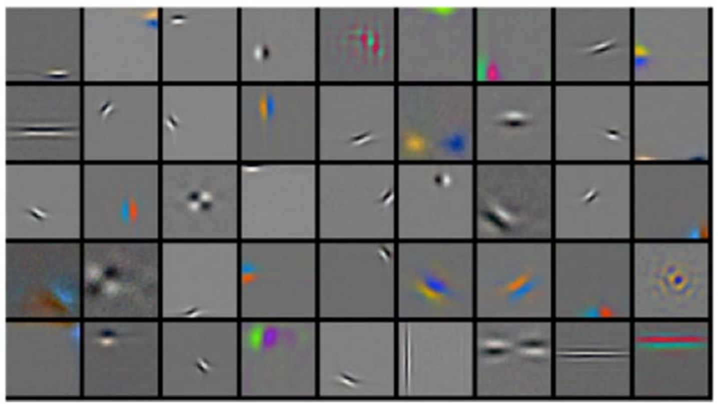
$$\theta = \{W, a, b\}$$

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

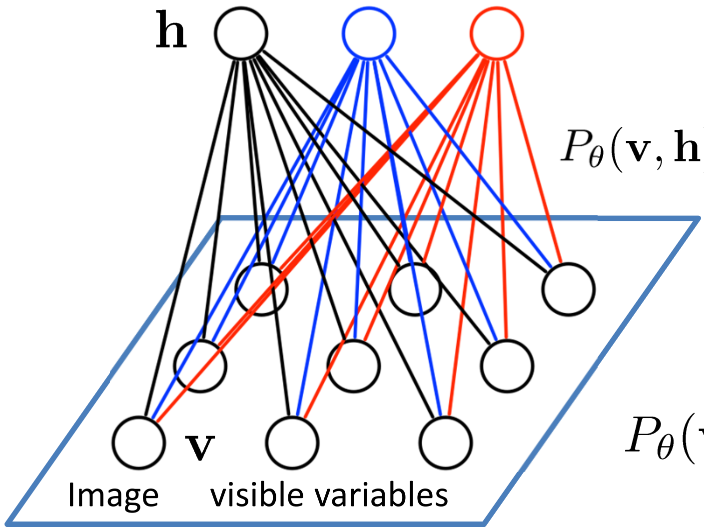
4 million **unlabelled** images



Learned features (out of 10,000)



RBMs for Real-valued Data



Pair-wise **Unary**

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

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

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

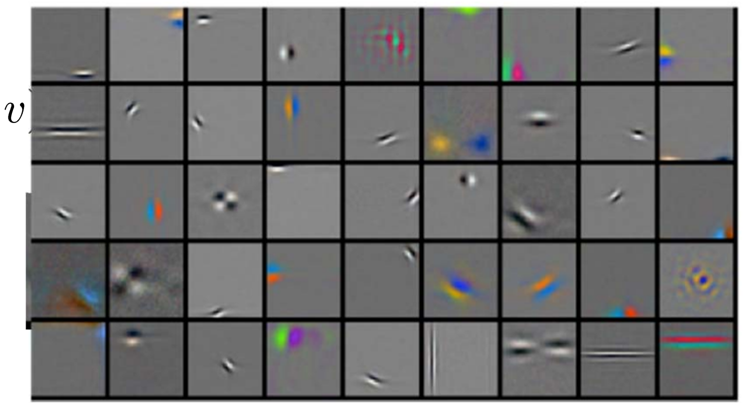
4 million **unlabelled** images

Ne

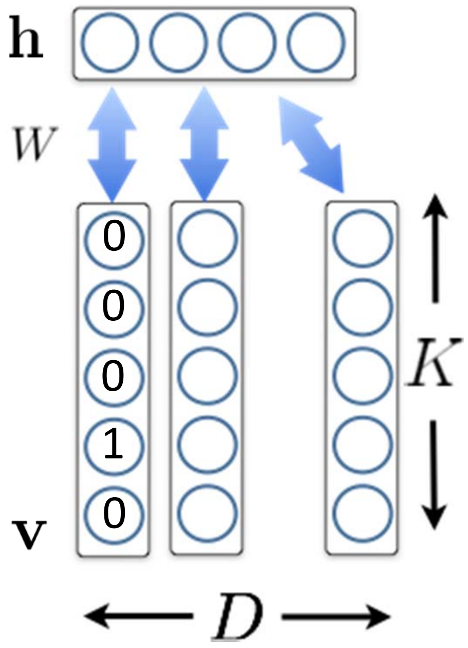
$p(h_{29} = 1|v)$

$+ \ 0.8 \ *$

Learned features (out of 10,000)



RBMs for Word Counts



Pair-wise   Unary

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

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

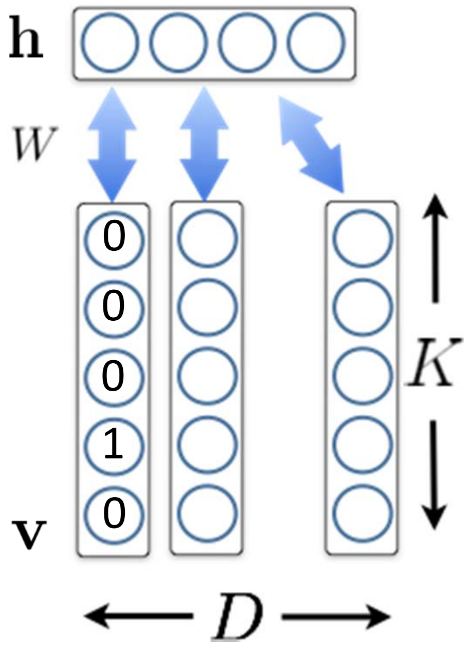
$$P_\theta(v_i^k = 1|\mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^F h_j W_{ij}^k\right)}{\sum_{q=1}^K \exp\left(b_i^q + \sum_{j=1}^F h_j W_{ij}^q\right)}$$

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

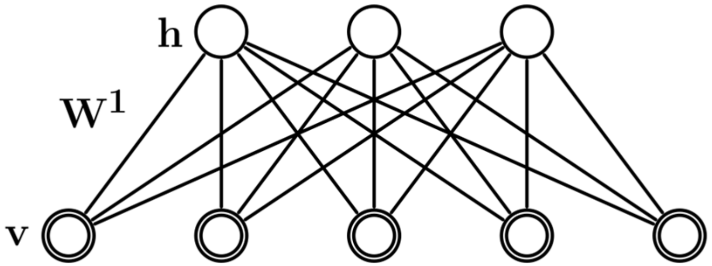
Learned features: ``topics''

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

Collaborative Filtering

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

Binary hidden: user preferences



Multinomial visible: user ratings

Netflix dataset:
480,189 users
17,770 movies
Over 100 million ratings

Learned features: ``genre''

Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt
Canadian Bacon
La Dolce Vita

Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

Friday the 13th
The Texas Chainsaw Massacre
Children of the Corn
Child's Play
The Return of Michael Myers

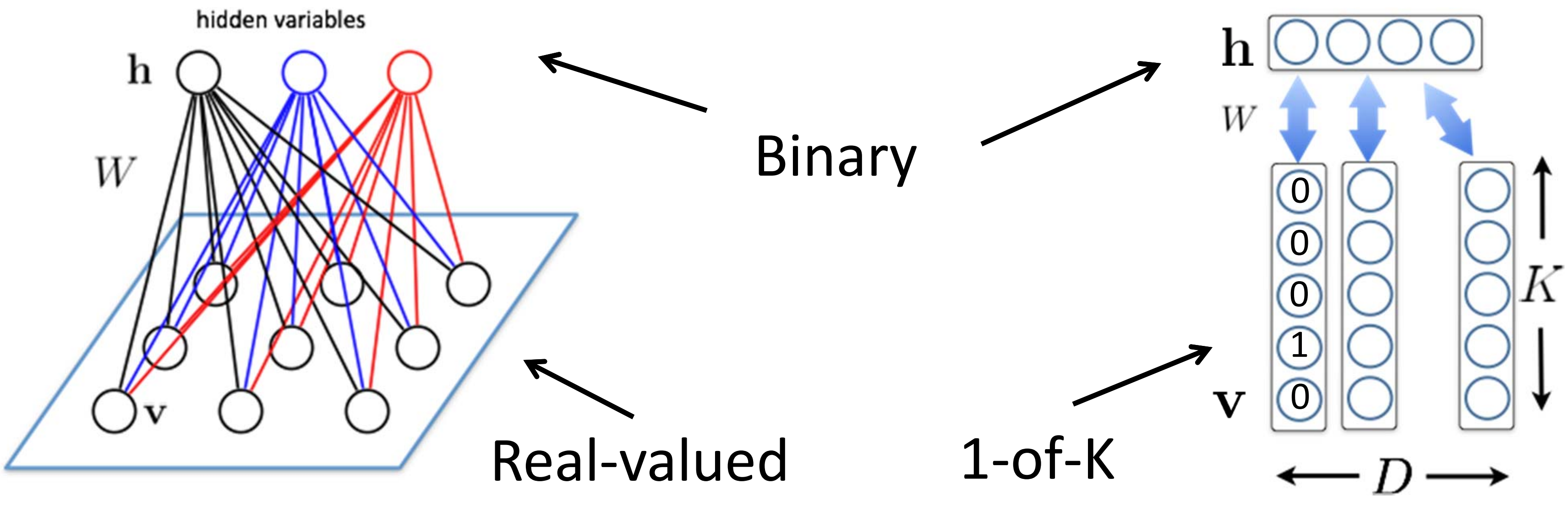
Scary Movie
Naked Gun
Hot Shots!
American Pie
Police Academy

State-of-the-art performance on the Netflix dataset.

(Salakhutdinov, Mnih, Hinton, ICML 2007)

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:

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

Marginalizing over hidden variables:

Product of Experts

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

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

Putin

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

(Salakhutdinov & Hinton, NIPS 2010)

Product of Experts

The joint distribution is given by:

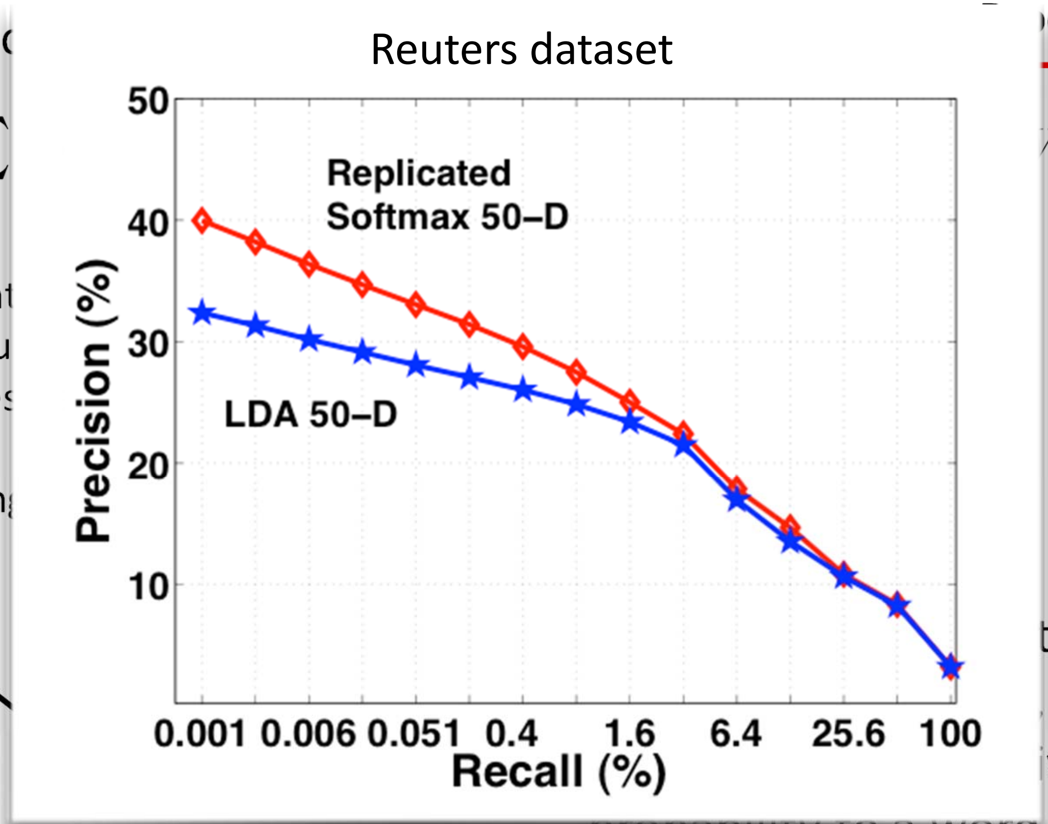
$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\Big(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\Big)$$

Marginalizing (

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}} \quad \ldots \quad W_{ij} v_i)\Big)$$

duct of Experts

government	clint
auhority	hou
power	pres
empire	bill
putin	con



tations allow the ”corruption” and ive very high probability to a word “Putin”.

Multiple Application Domains

- Natural Images
- Text/Documents
- Collaborative Filtering / Matrix Factorization
- Video
- Motion Capture
- Speech Perception

Same learning algorithm --
multiple input domains.

Limitations on the types of structure that can be represented by a single layer of low-level features!

Talk Roadmap

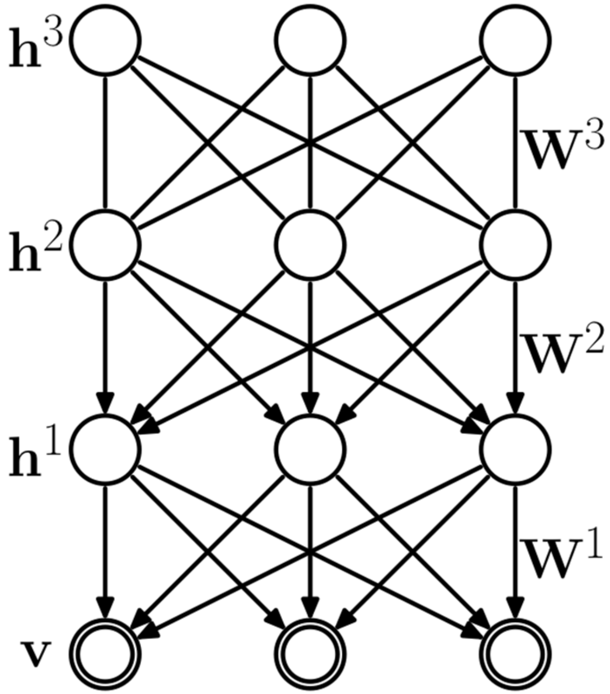
Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Advanced Deep Models.

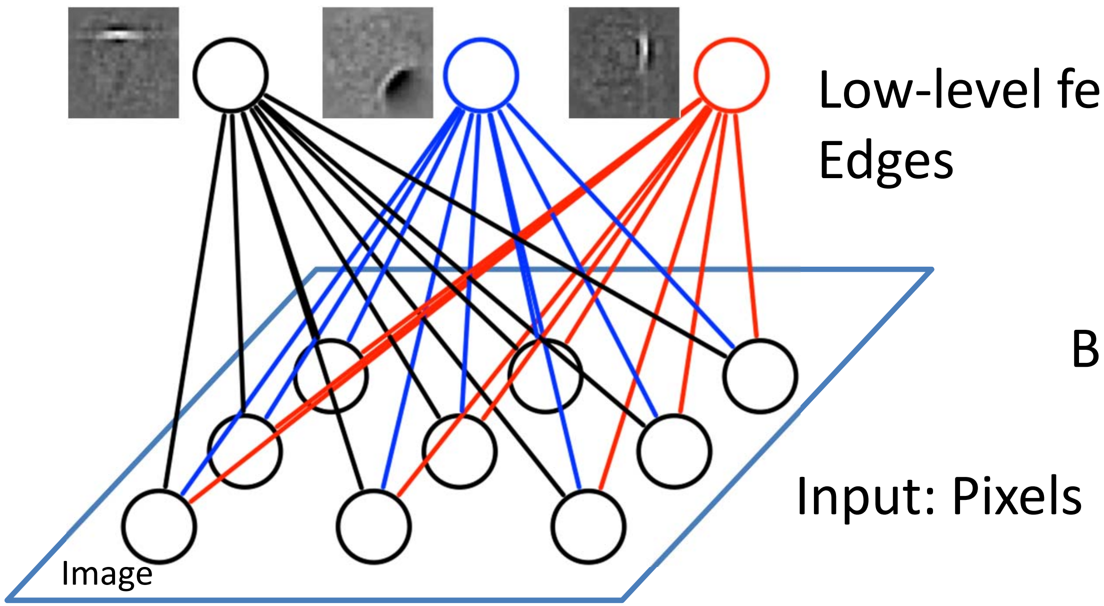
- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning

Deep Belief Network



- Probabilistic Generative model.
- Contains multiple layers of nonlinear representation.
- Fast, greedy layer-wise pretraining algorithm.
- Inferring the states of the latent variables in highest layers is easy.

Deep Belief Network

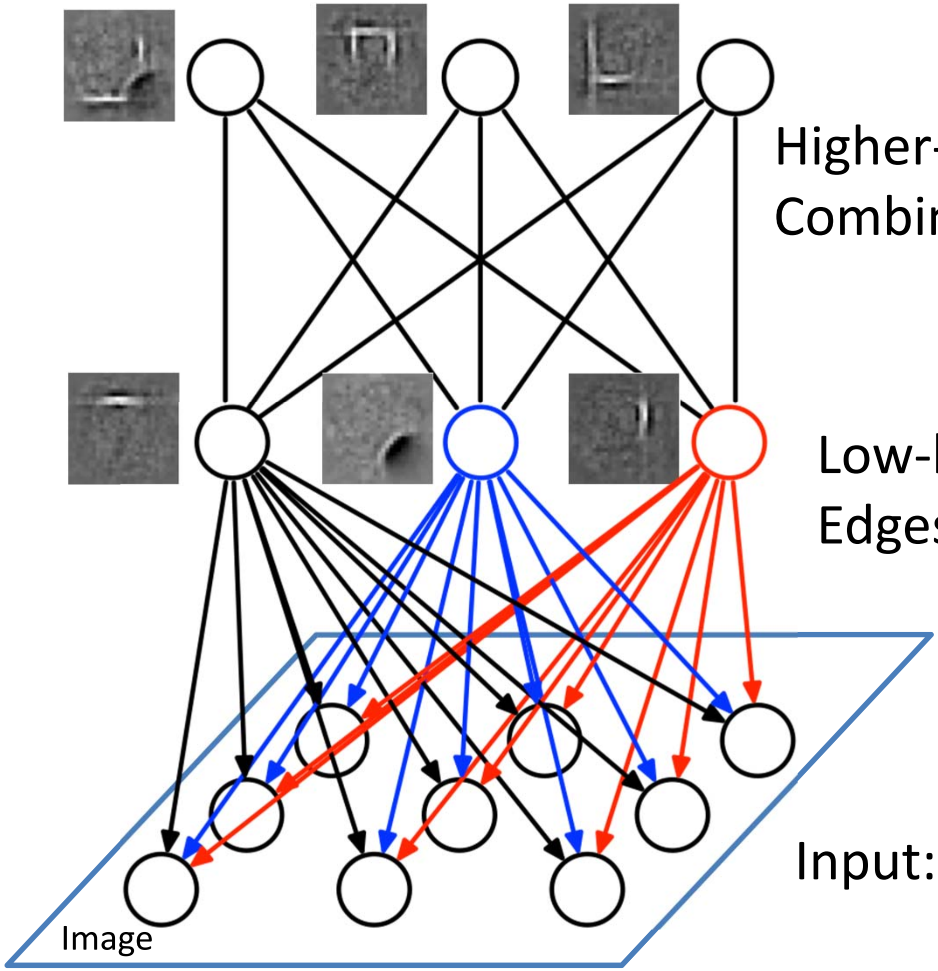


Built from **unlabeled** inputs.

(Hinton et.al. Neural Computation 2006)

Deep Belief Network

Internal representations capture higher-order statistical structure



Higher-level features:
Combination of edges

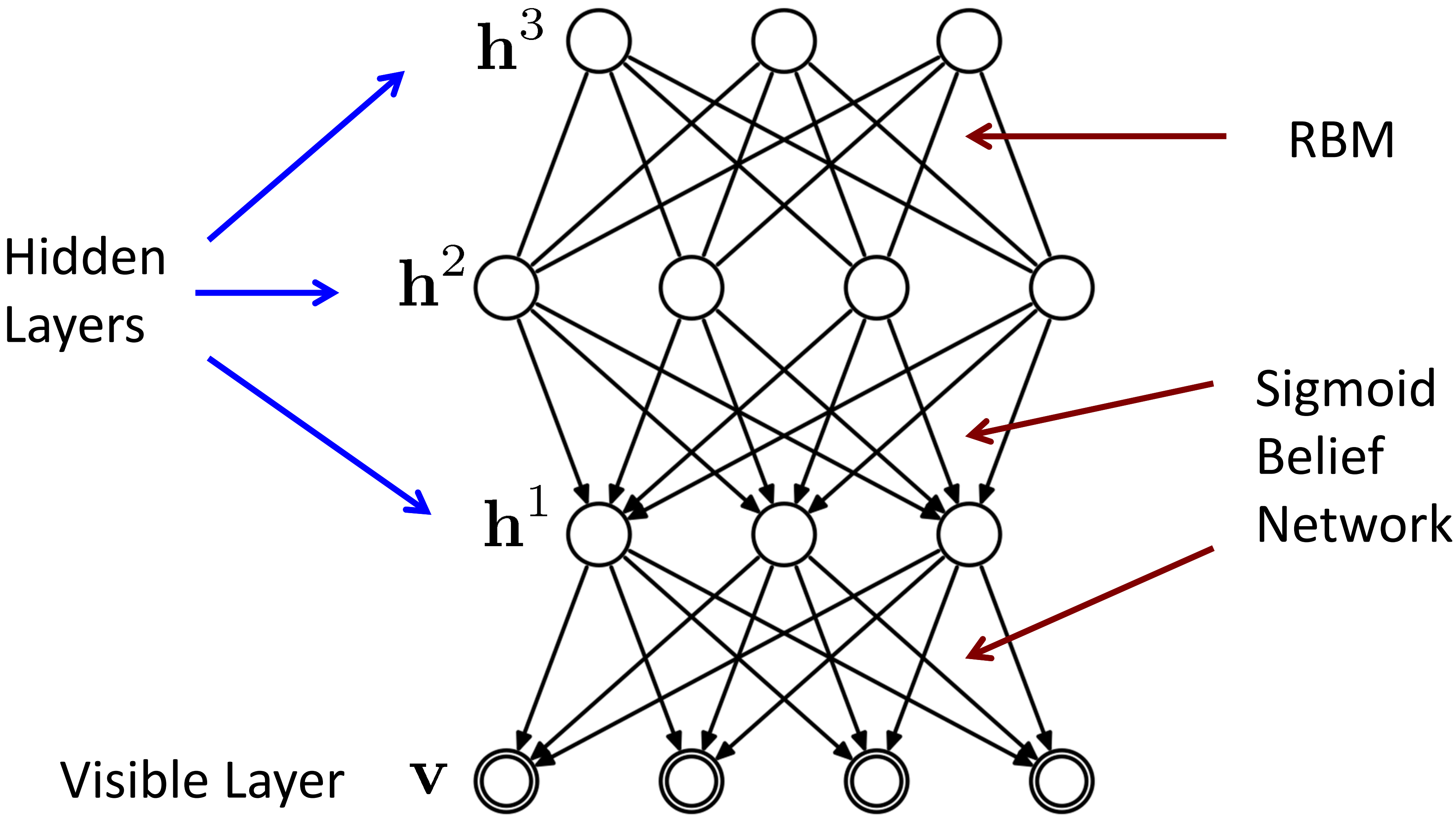
Low-level features:
Edges

Built from **unlabeled** inputs.

Input: Pixels

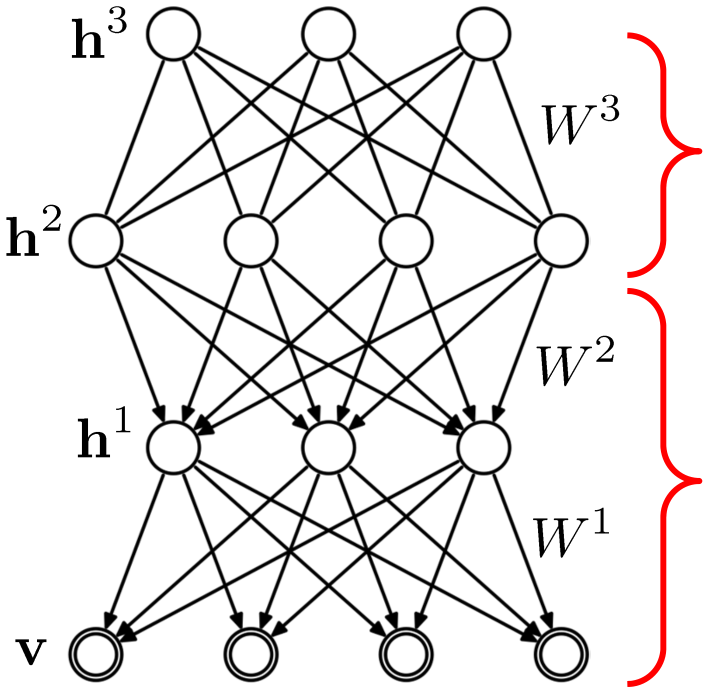
(Hinton et.al. Neural Computation 2006)

Deep Belief Network



Deep Belief Network

Deep Belief Network



RBM

Sigmoid Belief Network

The joint probability distribution factorizes:

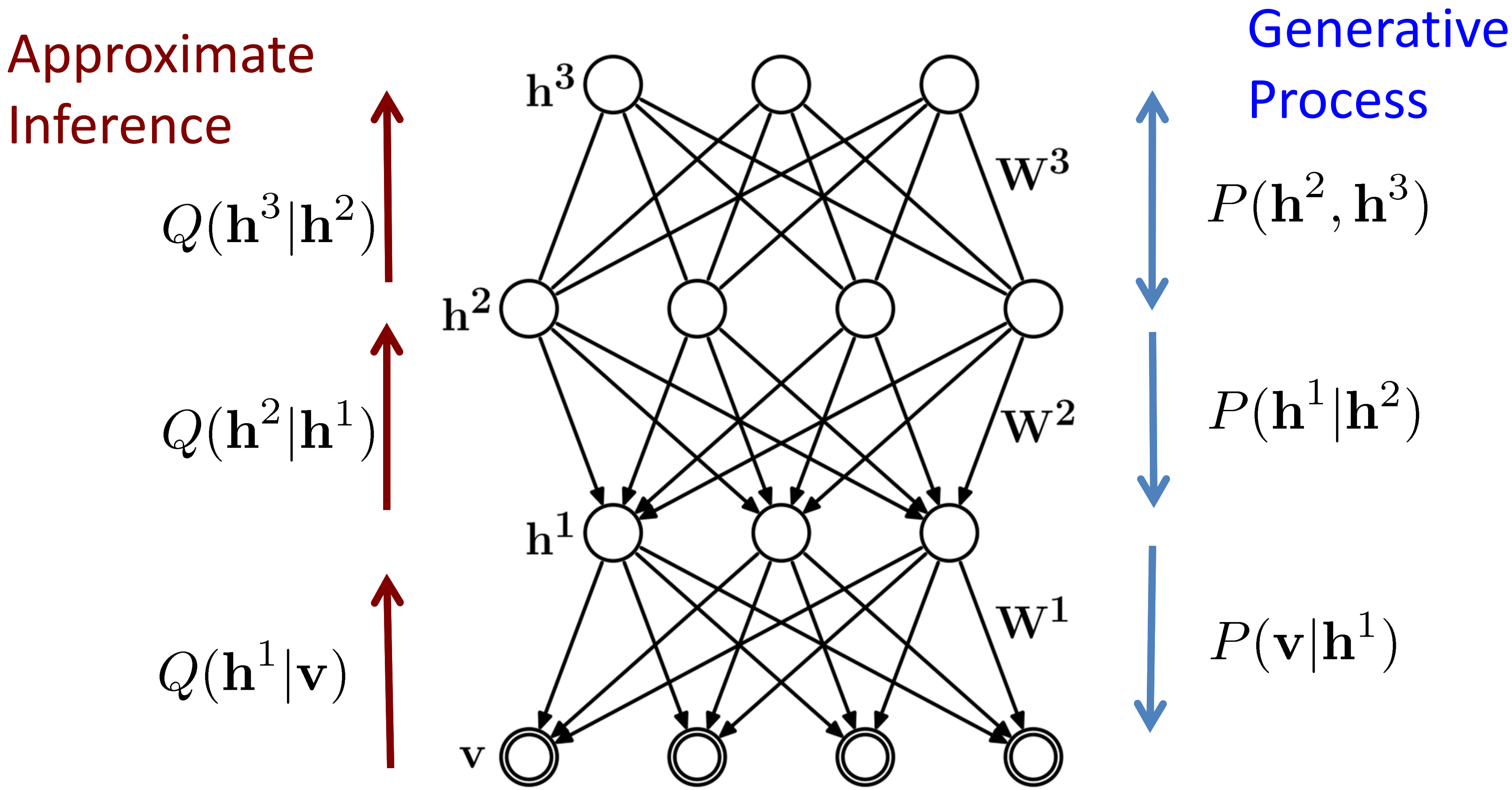
$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = \underbrace{P(\mathbf{v}|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)}_{\text{Sigmoid Belief Network}}\underbrace{P(\mathbf{h}^2, \mathbf{h}^3)}_{\text{RBM}}$$

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

$$P(\mathbf{h}^1|\mathbf{h}^2) = \prod_j P(h_j^1|\mathbf{h}^2) \qquad P(h_j^1 = 1|\mathbf{h}^2) = \frac{1}{1 + \exp\left(-\sum_k W_{jk}^2 h_k^2\right)}$$

$$P(\mathbf{v}|\mathbf{h}^1) = \prod_i P(v_i|\mathbf{h}^1) \qquad P(v_i = 1|\mathbf{h}^1) = \frac{1}{1 + \exp\left(-\sum_j W_{ij}^1 h_j^1\right)}$$

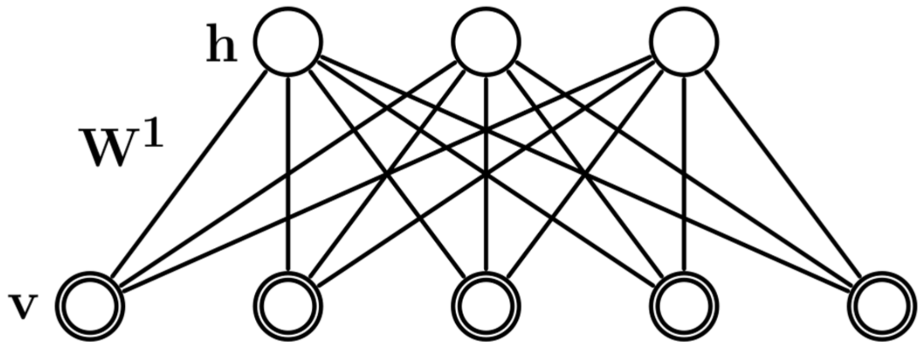
Deep Belief Network



$$Q(\mathbf{h}^t|\mathbf{h}^{t-1}) = \prod_j \sigma\left(\sum_i W^t h_i^{t-1}\right) \qquad P(\mathbf{h}^{t-1}|\mathbf{h}^t) = \prod_j \sigma\left(\sum_i W^t h_i^t\right)$$

DBN Layer-wise Training

- Learn an RBM with an input layer v and a hidden layer h.

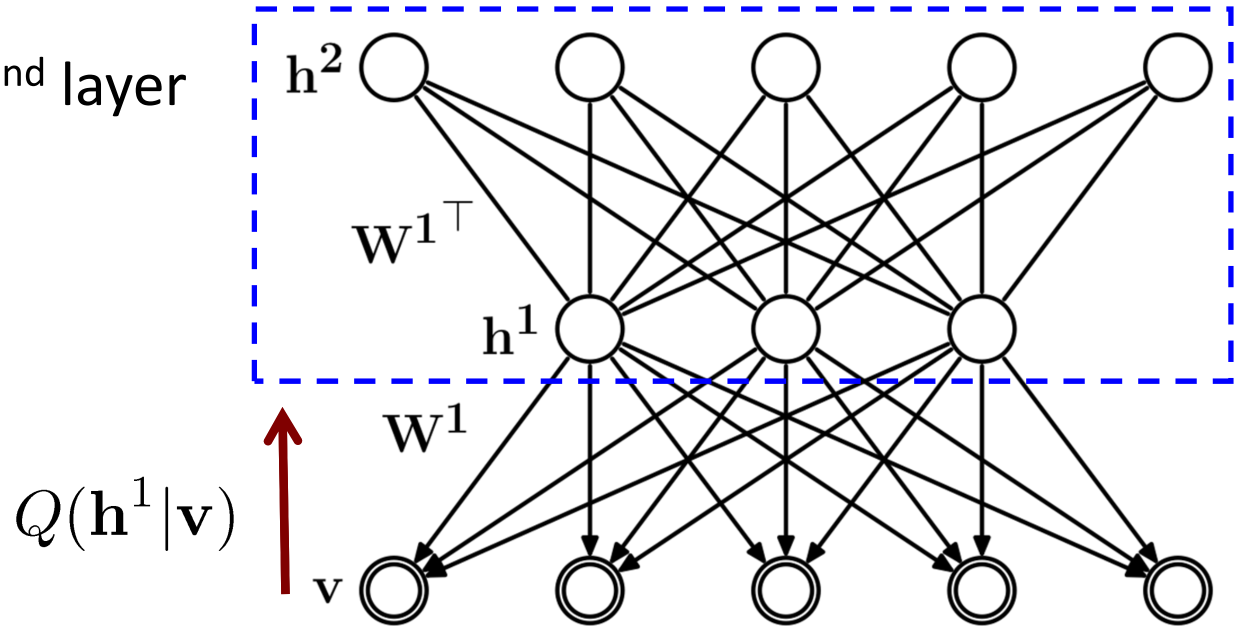


DBN Layer-wise Training

- Learn an RBM with an input layer v and a hidden layer h.

- Treat inferred values $Q(\mathbf{h}^1|\mathbf{v}) = P(\mathbf{h}^1|\mathbf{v})$ as the data for training 2nd-layer RBM.

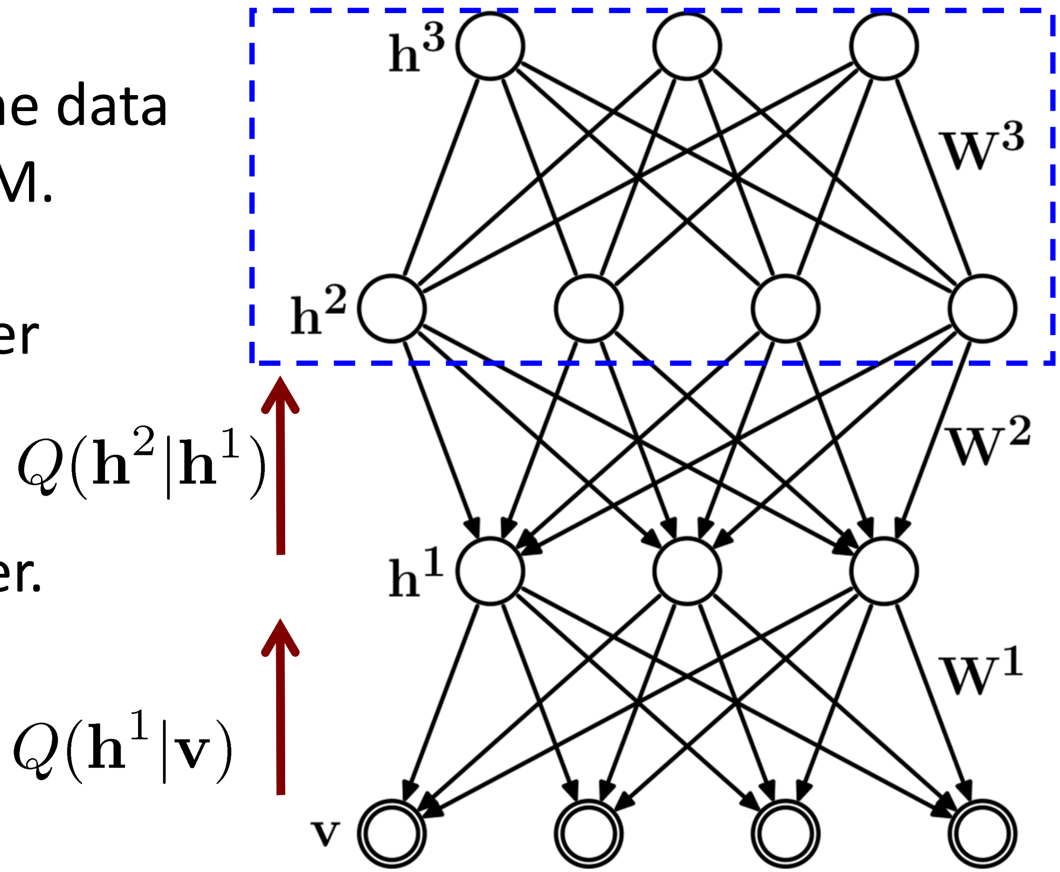
- Learn and freeze 2nd layer RBM.



DBN Layer-wise Training

- Learn an RBM with an input layer v and a hidden layer h.
- Treat inferred values $Q(\mathbf{h}^1|\mathbf{v}) = P(\mathbf{h}^1|\mathbf{v})$ as the data for training 2nd-layer RBM.
- Learn and freeze 2nd layer RBM.
- Proceed to the next layer.

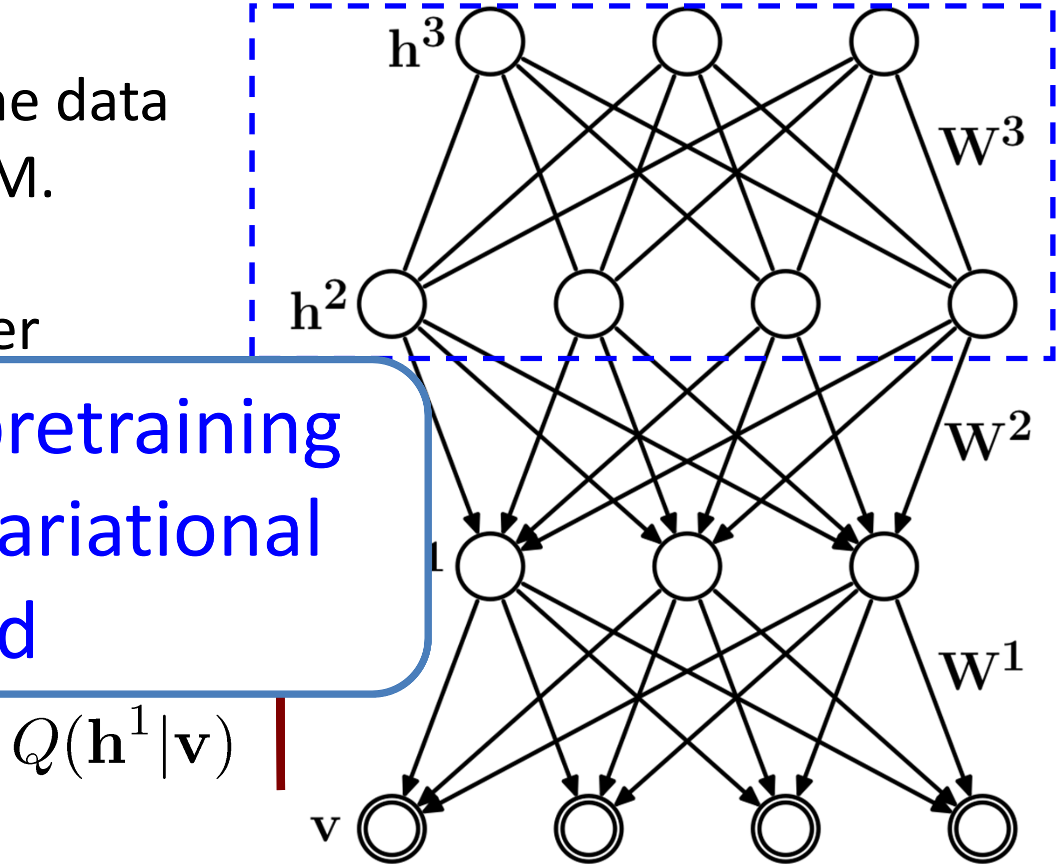
Unsupervised Feature Learning.



DBN Layer-wise Training

- Learn an RBM with an input layer v and a hidden layer h.
- Treat inferred values $Q(\mathbf{h}^1|\mathbf{v}) = P(\mathbf{h}^1|\mathbf{v})$ as the data for training 2nd-layer RBM.
- Learn and freeze 2nd layer RBM.
- Procee

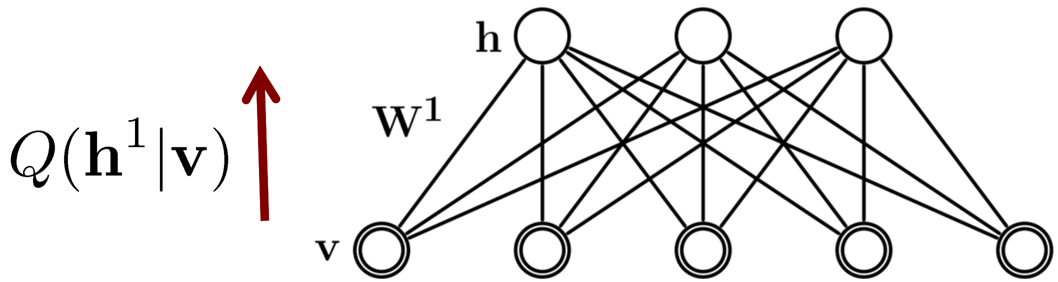
Unsupervised Feature Learning.



Layerwise pretraining improves variational lower bound

Why this Pre-training Works?

- Greedy pre-training improves variational lower bound!



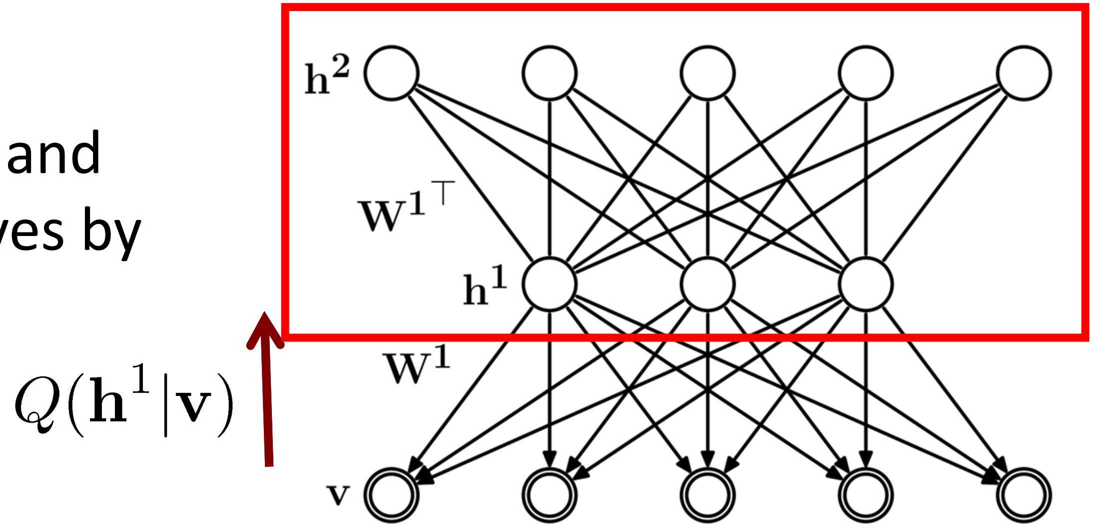
- For any approximating distribution $Q(\mathbf{h}^1|\mathbf{v})$

$$\log P_\theta(\mathbf{v}) = \sum_{\mathbf{h}^1} P_\theta(\mathbf{v}, \mathbf{h}^1)$$

$$\geq \sum_{\mathbf{h}^1} Q(\mathbf{h}^1|\mathbf{v}) \Big[\log P(\mathbf{h}^1) + \log P(\mathbf{v}|\mathbf{h}^1) \Big] + \mathcal{H}(Q(\mathbf{h}^1|\mathbf{v}))$$

Why this Pre-training Works?

- Greedy training improves variational lower bound.
- RBM and 2-layer DBN are equivalent when $W^2 = {W^1}^\top$.
- The lower bound is tight and the log-likelihood improves by greedy training.



- For any approximating distribution $Q(\mathbf{h}^1|\mathbf{v})$

$$\log P_\theta(\mathbf{v}) = \sum_{\mathbf{h}^1} P_\theta(\mathbf{v}, \mathbf{h}^1)$$

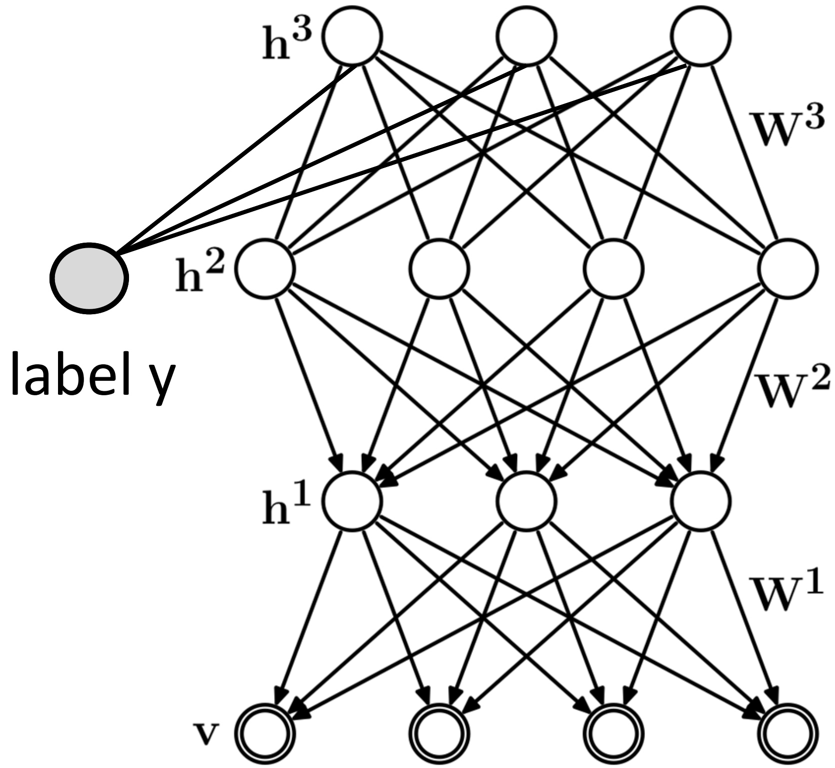
$$\geq \sum_{\mathbf{h}^1} Q(\mathbf{h}^1|\mathbf{v}) \Big[\log P(\mathbf{h}^1) + \log P(\mathbf{v}|\mathbf{h}^1) \Big] + \mathcal{H}(Q(\mathbf{h}^1|\mathbf{v}))$$

Train 2nd-layer RBM

Supervised Learning with DBNs

- If we have access to label information, we can train the joint generative model by maximizing the joint log-likelihood of data and labels

$$\log P(\mathbf{y}, \mathbf{v})$$



- Discriminative fine-tuning:
  - Use DBN to initialize a multilayer neural network.
  - Maximize the conditional distribution:

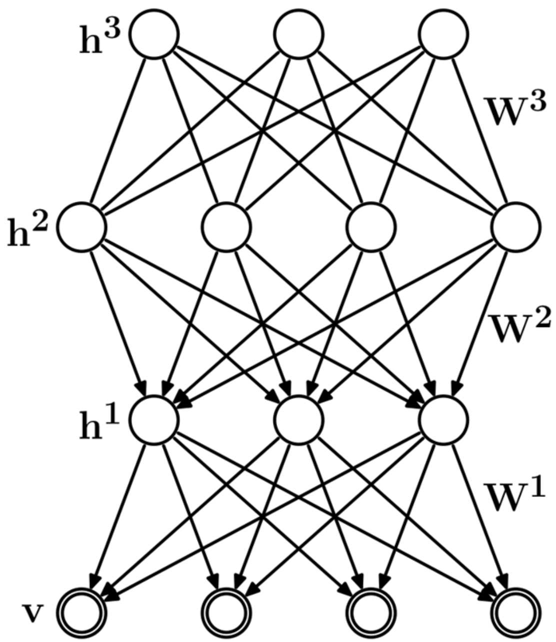
$$\log P(\mathbf{y}|\mathbf{v})$$

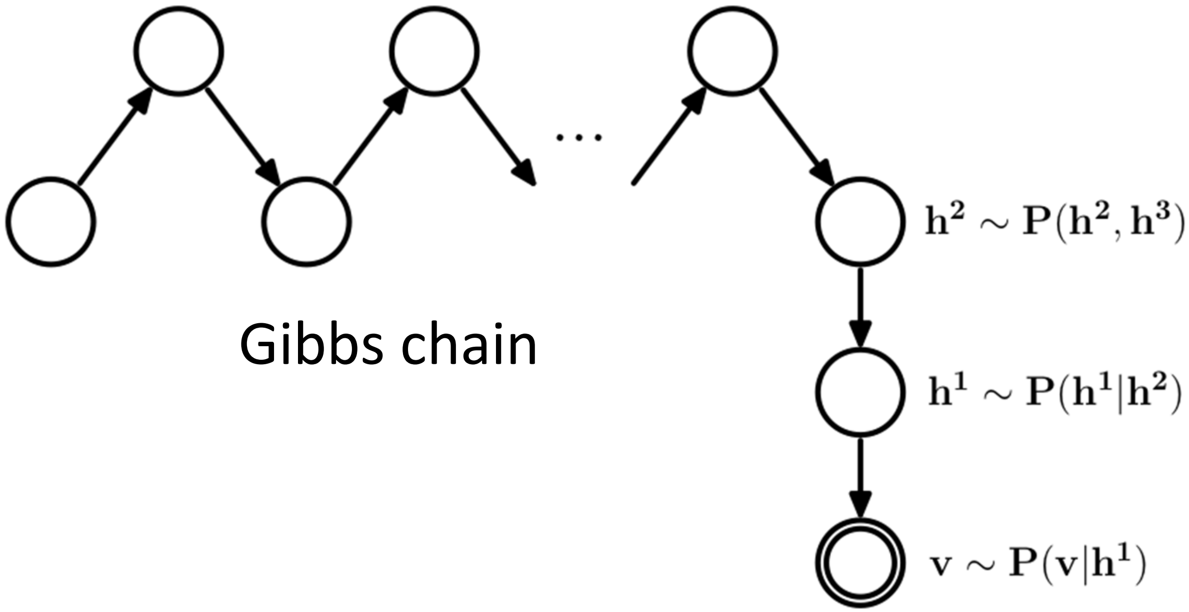
Sampling from DBNs

- To sample from the DBN model:

$$P(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = P(\mathbf{v}|\mathbf{h}^1) P(\mathbf{h}^1|\mathbf{h}^2) P(\mathbf{h}^2, \mathbf{h}^3)$$

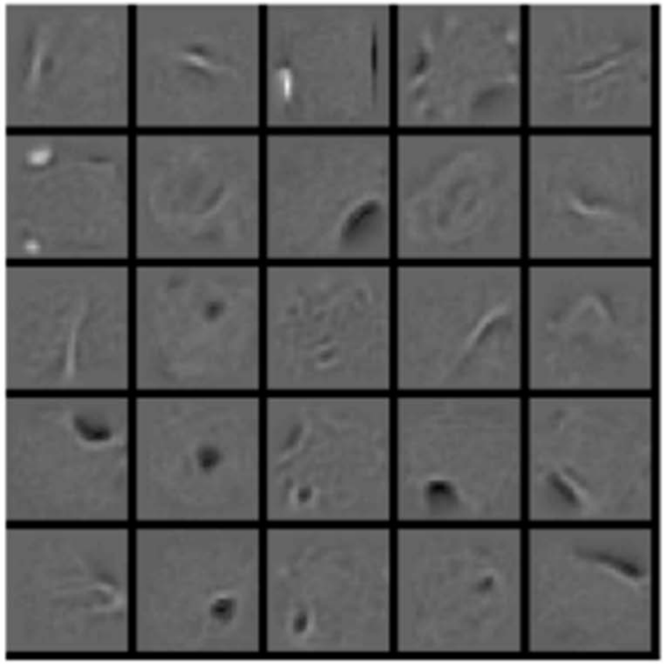
- Sample h^2 using alternating Gibbs sampling from RBM.
- Sample lower layers using sigmoid belief network.



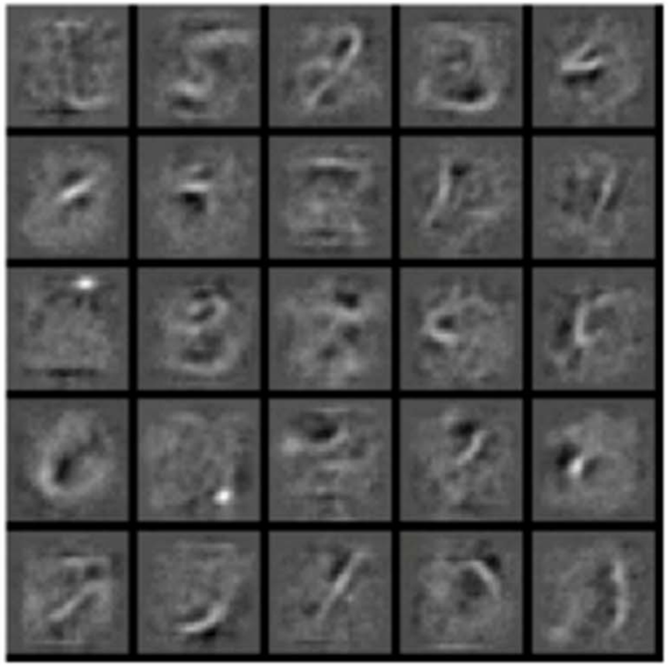


Learned Features

1^{st}-layer features

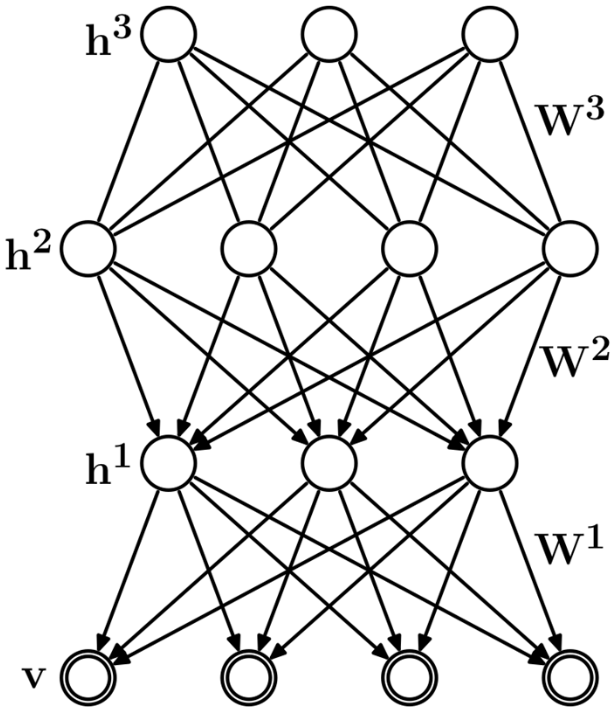


2^{nd}-layer features

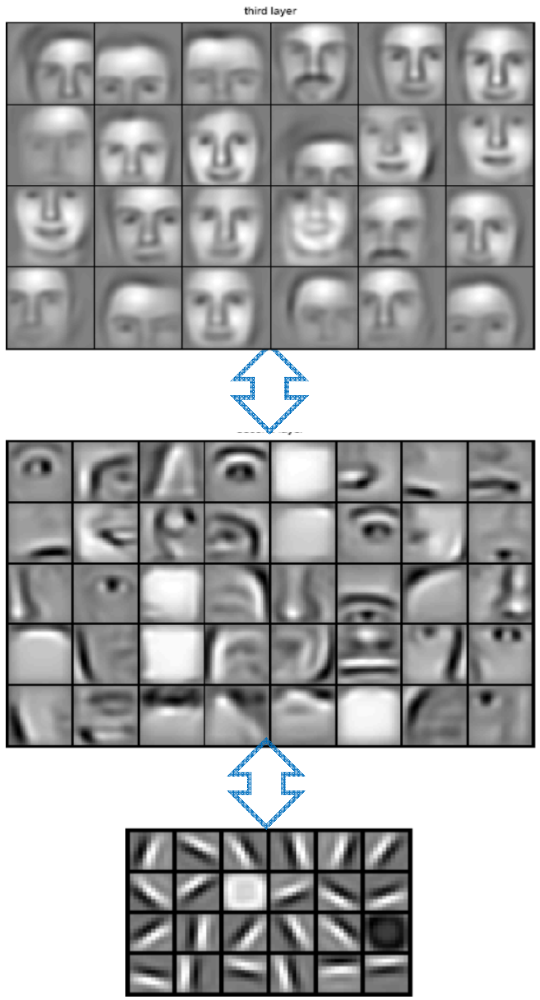


Learning Part-based Representation

Convolutional DBN



Faces



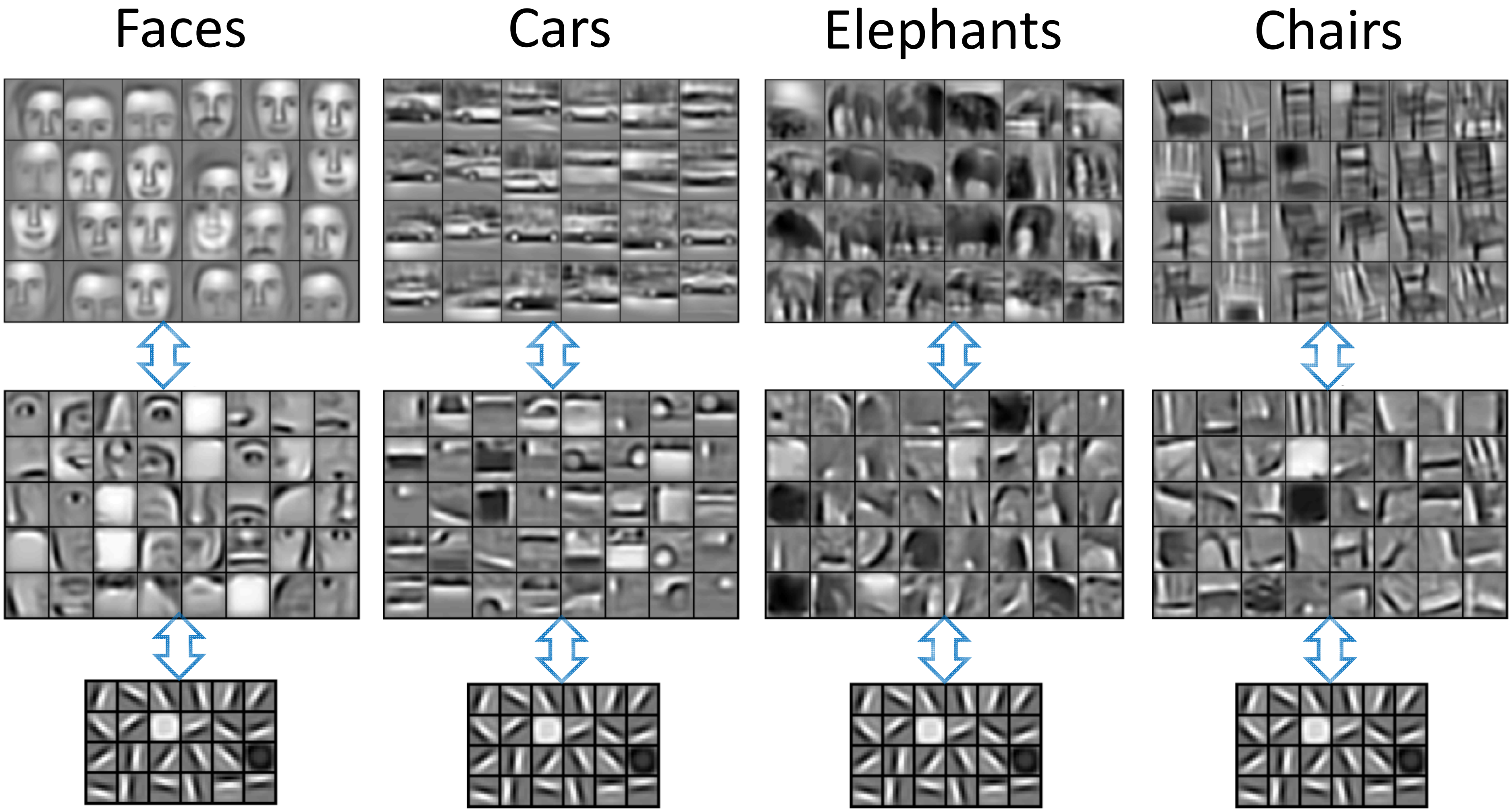
Groups of parts.

Object Parts

Trained on face images.

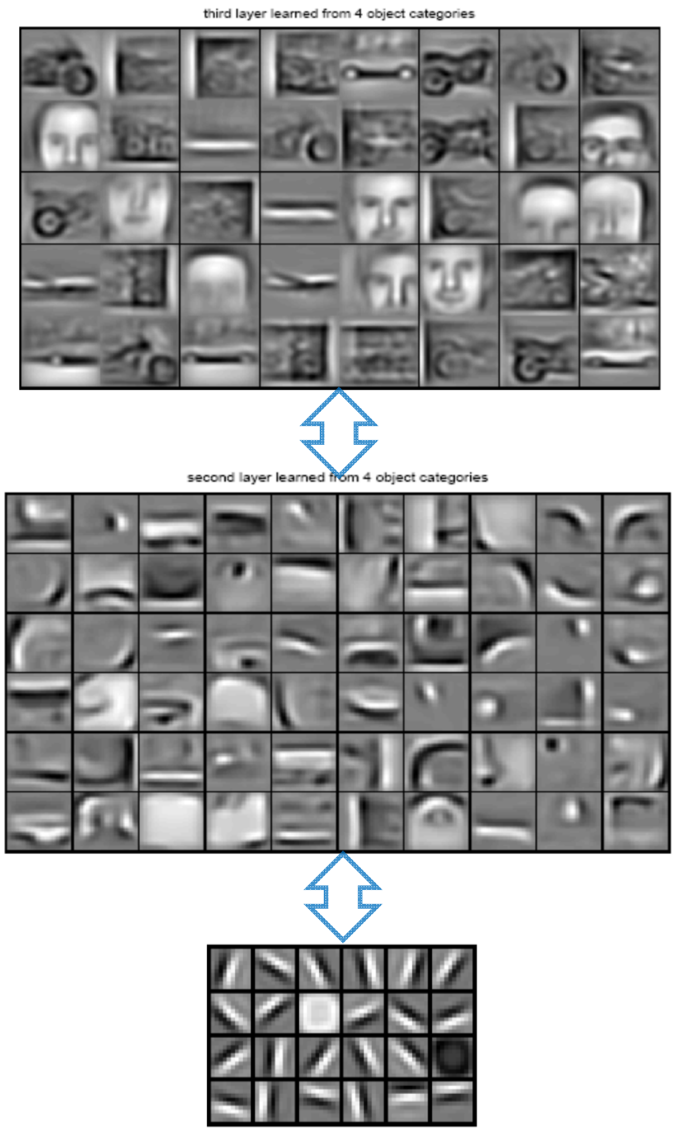
Lee et.al., ICML 2009

Learning Part-based Representation



Lee et.al., ICML 2009

Learning Part-based Representation



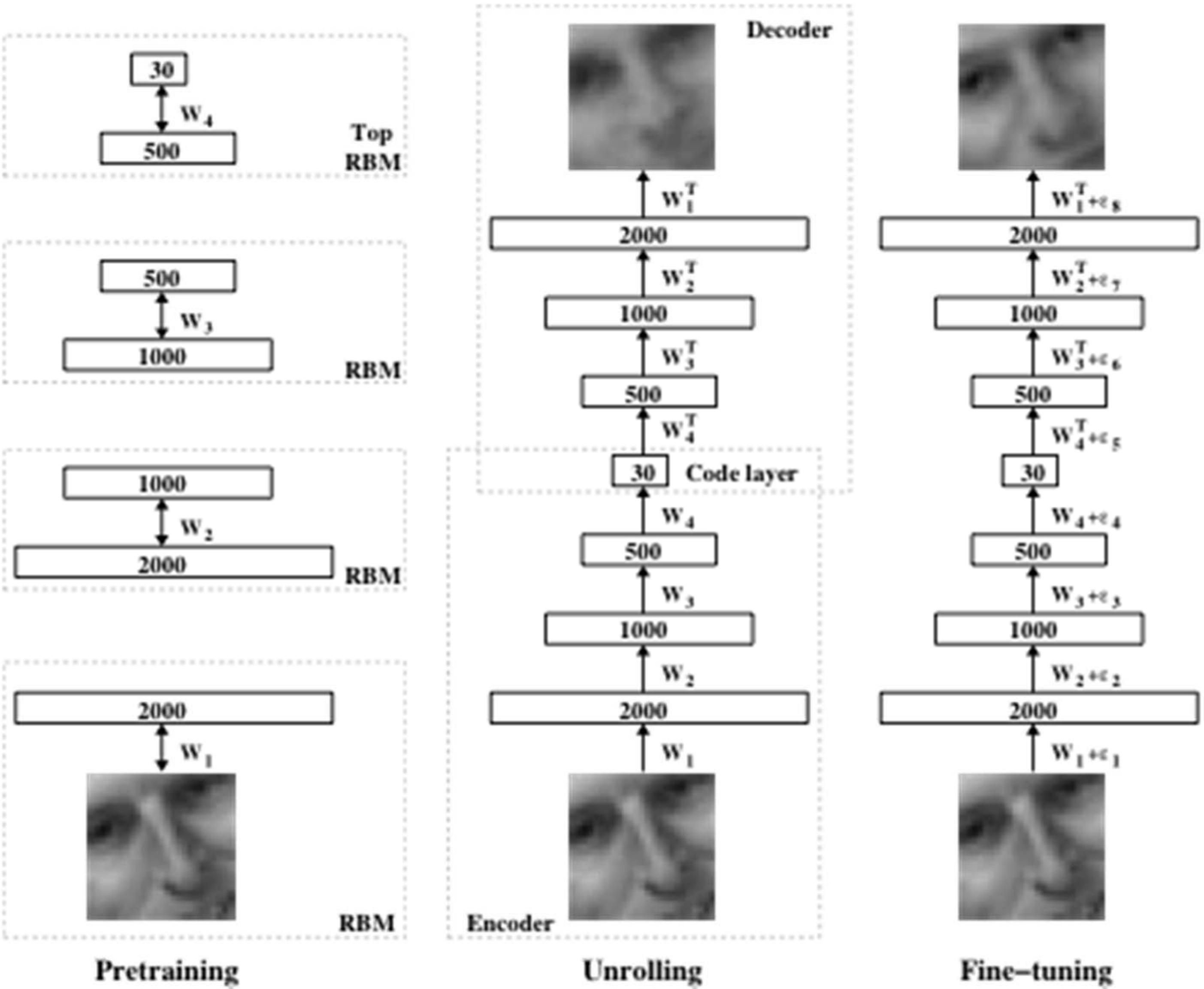
Groups of parts.

Class-specific object parts

Trained from multiple classes (cars, faces, motorbikes, airplanes).

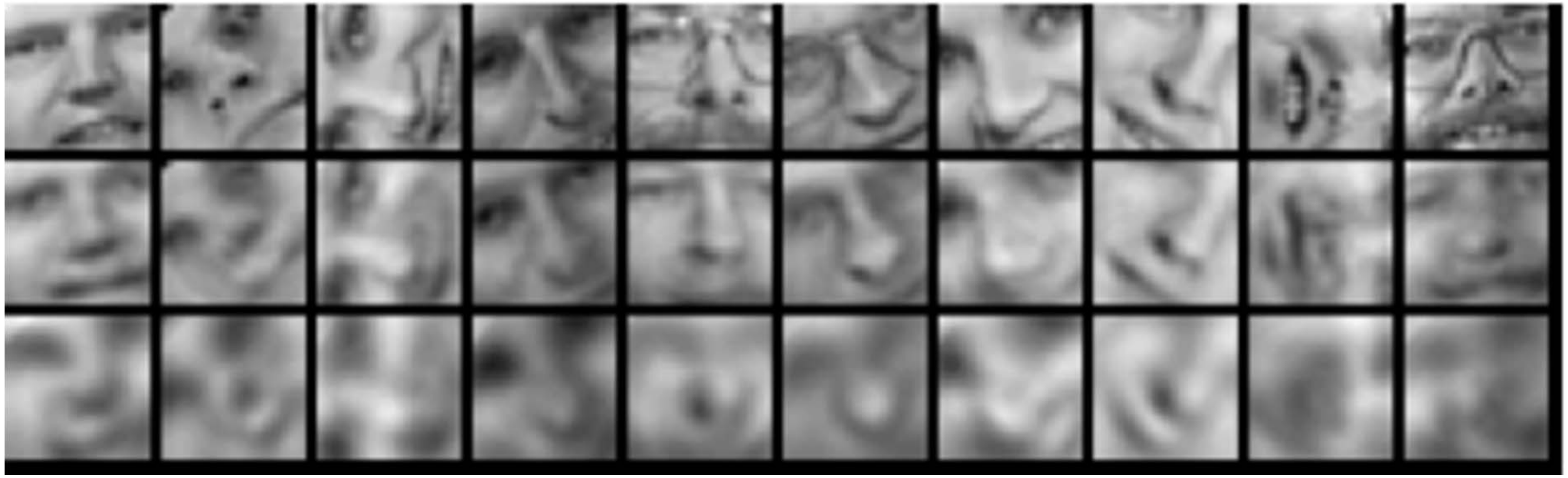
Lee et.al., ICML 2009

Deep Autoencoders



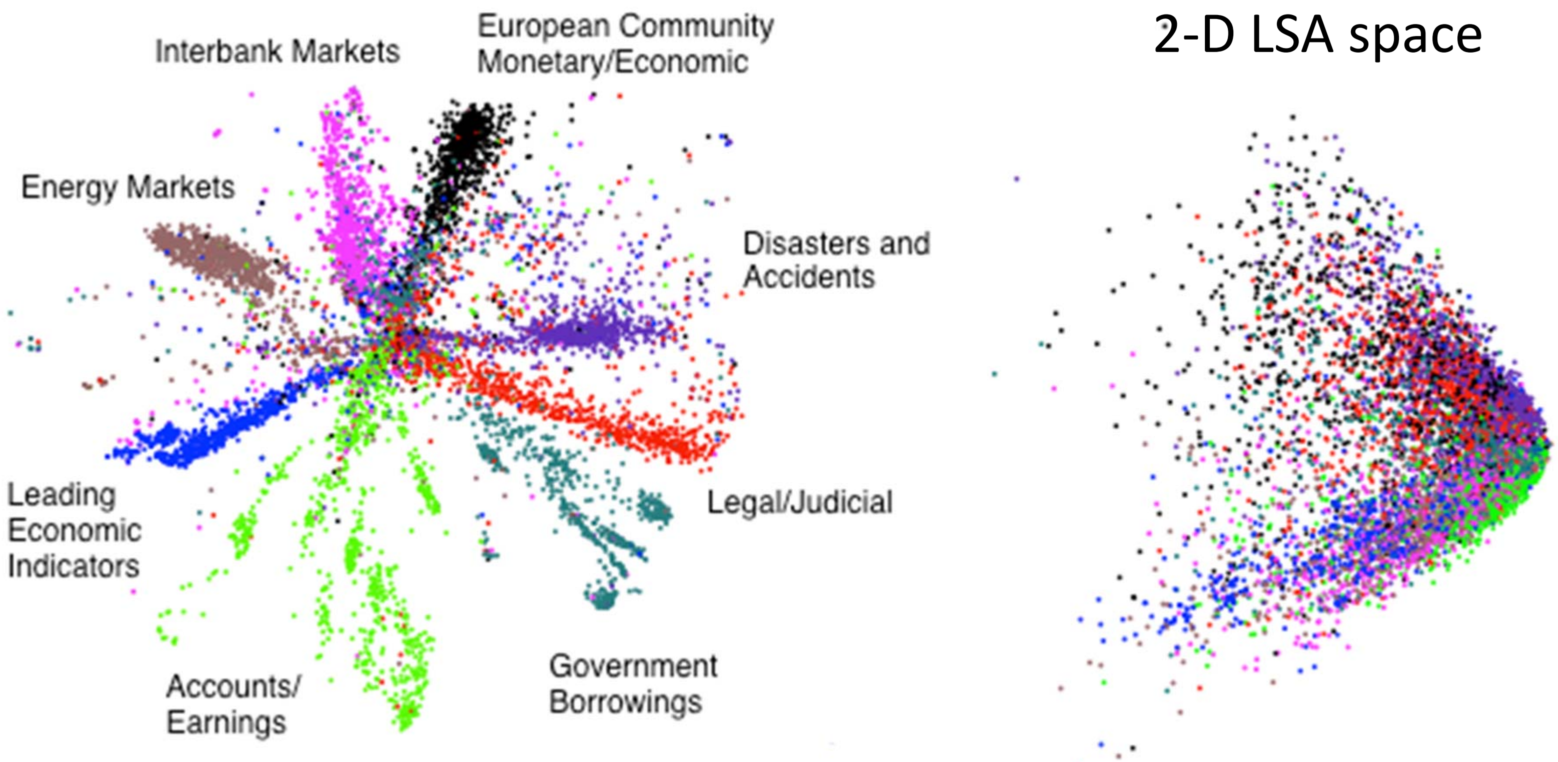
Deep Autoencoders

- We used 25x25 – 2000 – 1000 – 500 – 30 autoencoder to extract 30-D real-valued codes for Olivetti face patches.



- **Top**: Random samples from the test dataset.
- **Middle**: Reconstructions by the 30-dimensional deep autoencoder.
- **Bottom**: Reconstructions by the 30-dimentinoal PCA.

Information Retrieval

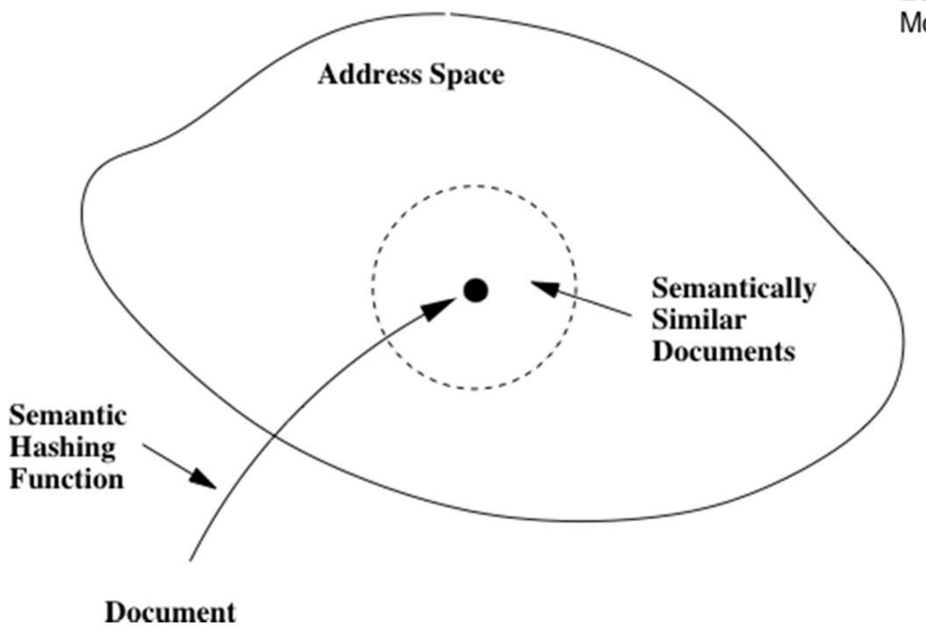


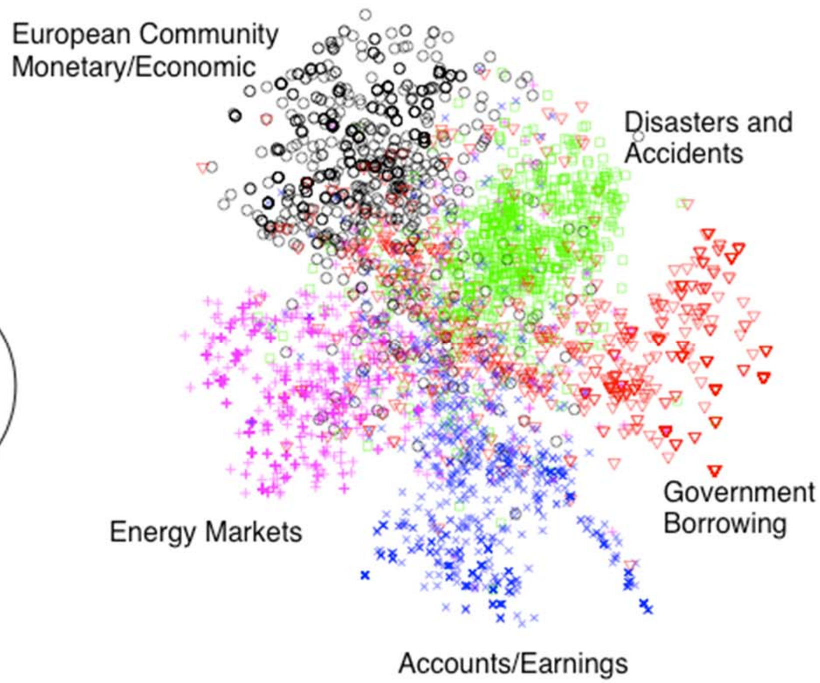
• The Reuters Corpus Volume II contains 804,414 newswire stories (randomly split into **402,207 training** and **402,207 test).**

• "Bag-of-words" representation: each article is represented as a vector containing the counts of the most frequently used 2000 words in the training set.

(Hinton and Salakhutdinov, Science 2006)

Semantic Hashing



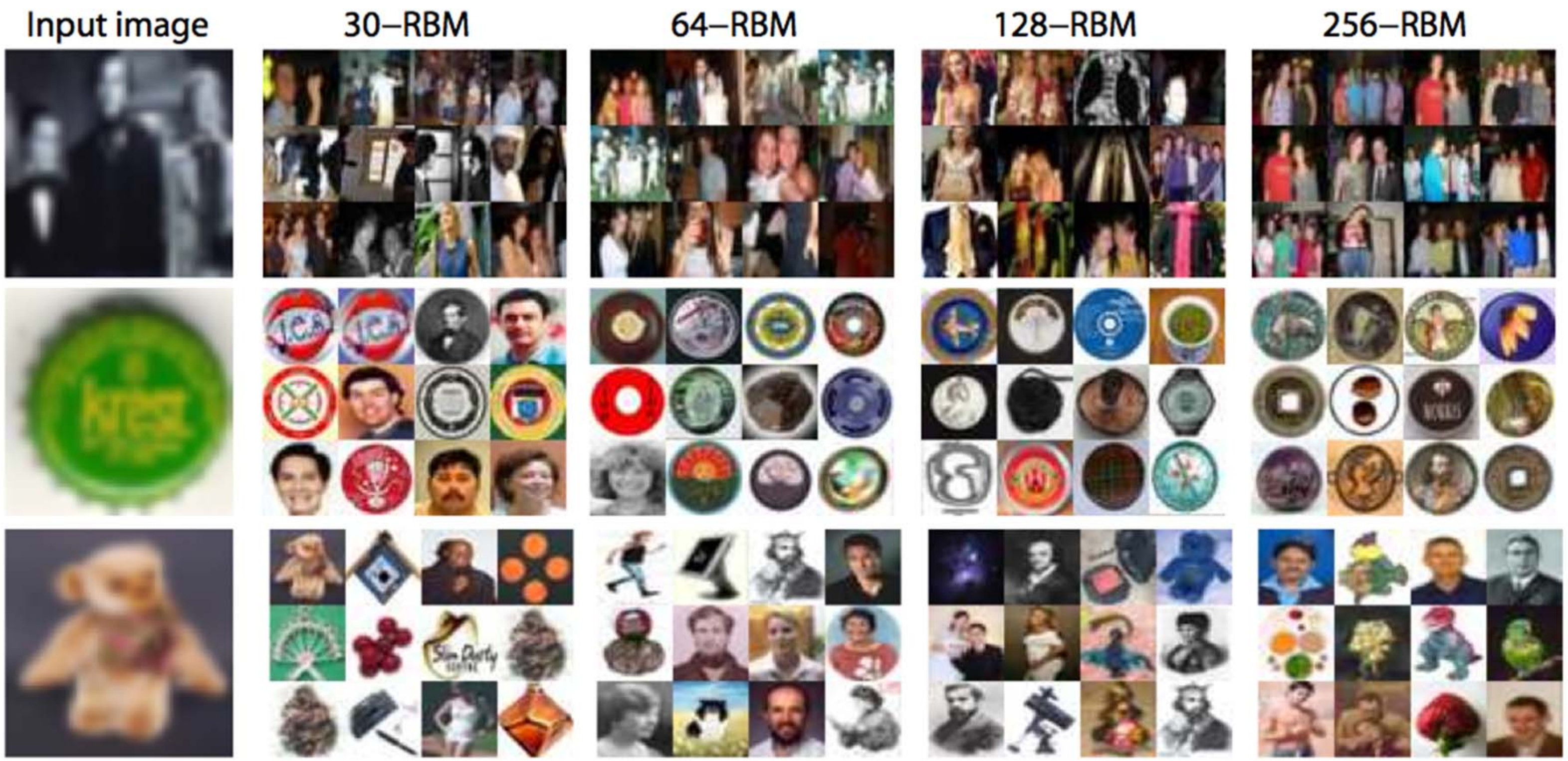


- Learn to map documents into **semantic 20-D binary codes.**
- Retrieve similar documents stored at the nearby addresses **with no search at all.**

(Salakhutdinov and Hinton, SIGIR 2007)

Searching Large Image Database using Binary Codes

- Map images into binary codes for fast retrieval.



- Small Codes, Torralba, Fergus, Weiss, CVPR 2008
- Spectral Hashing, Y. Weiss, A. Torralba, R. Fergus, NIPS 2008
- Kulis and Darrell, NIPS 2009, Gong and Lazebnik, CVPR 20111
- Norouzi and Fleet, ICML 2011,

Talk Roadmap

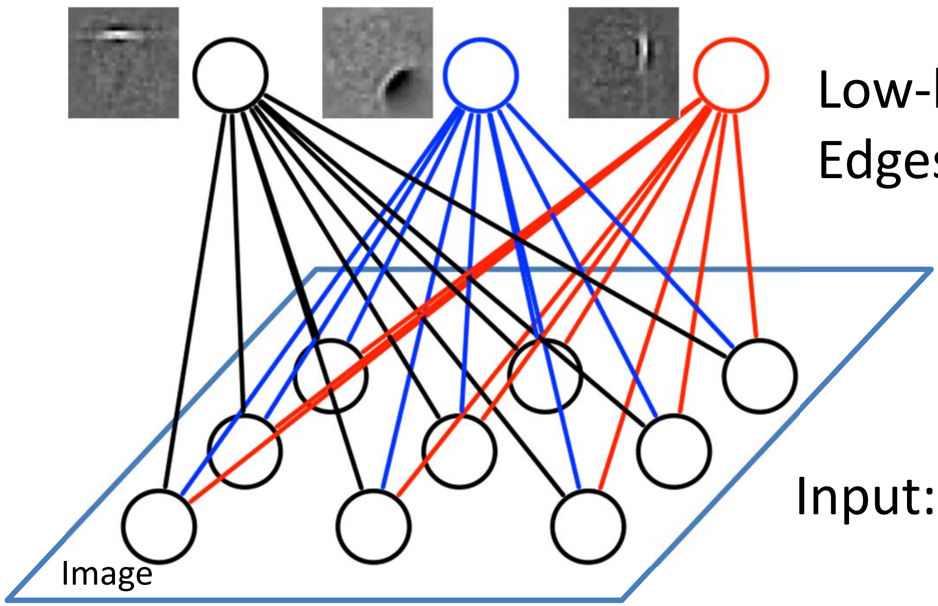
Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Advanced Deep Models.

- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning

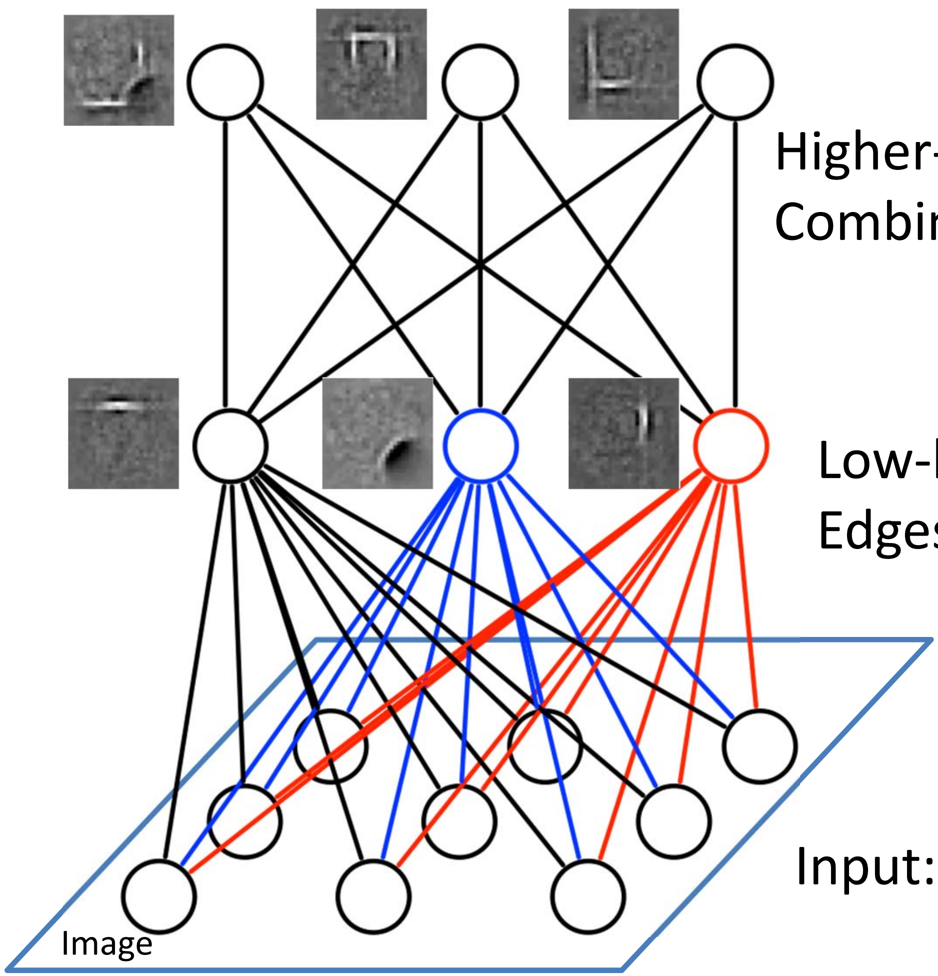
Deep Boltzmann Machines



Built from **unlabeled** inputs.

Deep Boltzmann Machines

Learn simpler representations,
then compose more complex ones



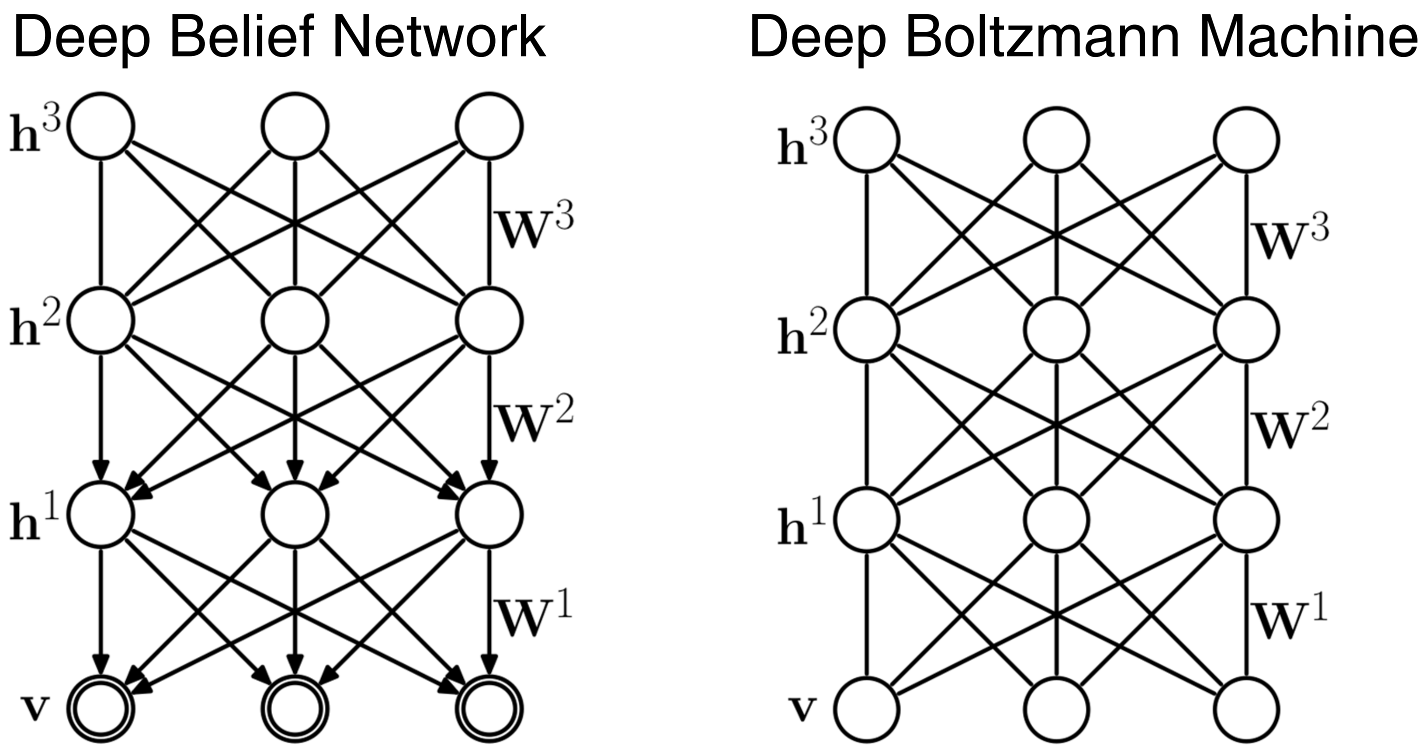
Higher-level features:
Combination of edges

Low-level features:
Edges

Built from **unlabeled** inputs.

Input: Pixels

DBNs vs. DBMs



DBNs are hybrid models:

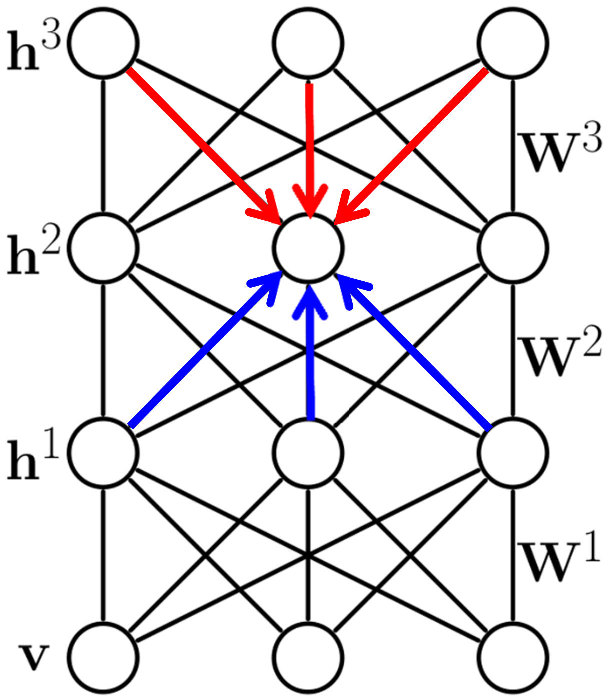
- Inference in DBNs is problematic due to **explaining away**.
- Only greedy pretrainig, **no joint optimization over all layers**.
- Approximate inference is feed-forward: **no bottom-up and top-down**.

Introduce a new class of models called Deep Boltzmann Machines.

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



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

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

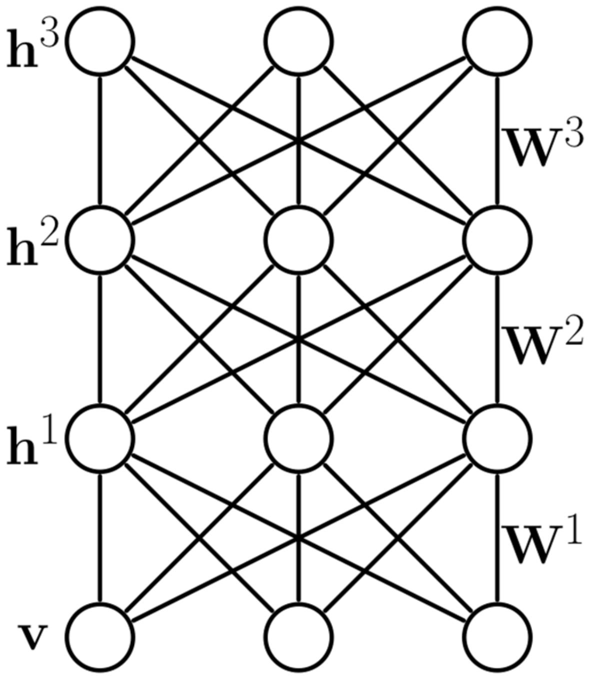
$$P(h_j^2 = 1 | \mathbf{h}^1, \mathbf{h}^3) = \sigma\left(\sum_k W_{kj}^3 h_k^3 + \sum_m W_{mj}^2 h_m^1\right)$$

Top-down Bottom-up

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

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!

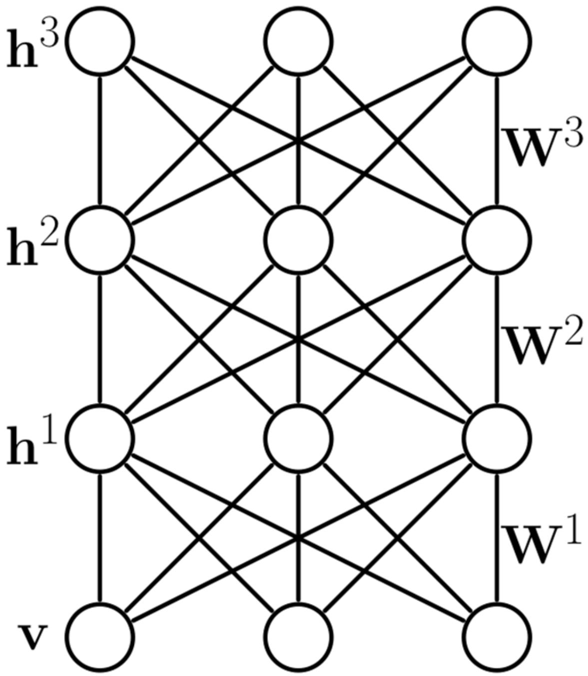
$$P_{data}(\mathbf{v}, \mathbf{h^1}) = P_\theta(\mathbf{h^1}|\mathbf{v}) P_{data}(\mathbf{v})$$

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

Not factorial any more!

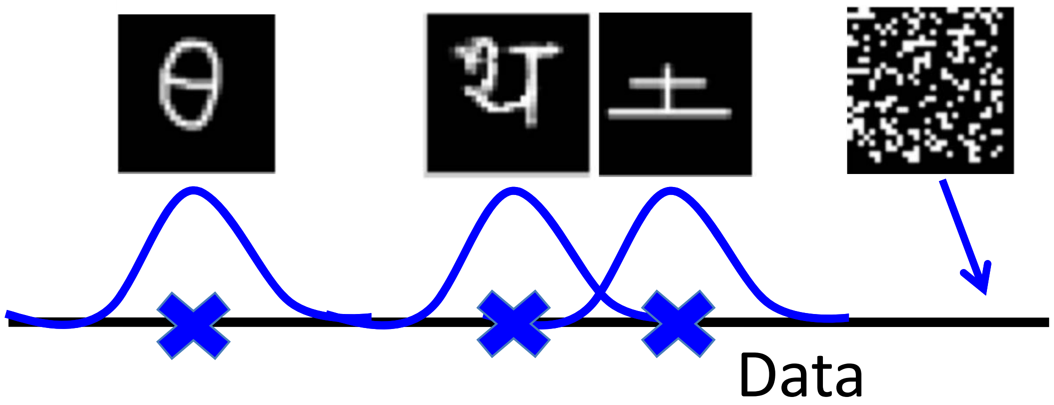
Approximate Learning

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



(Approximate) Maximum Likelihood:

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



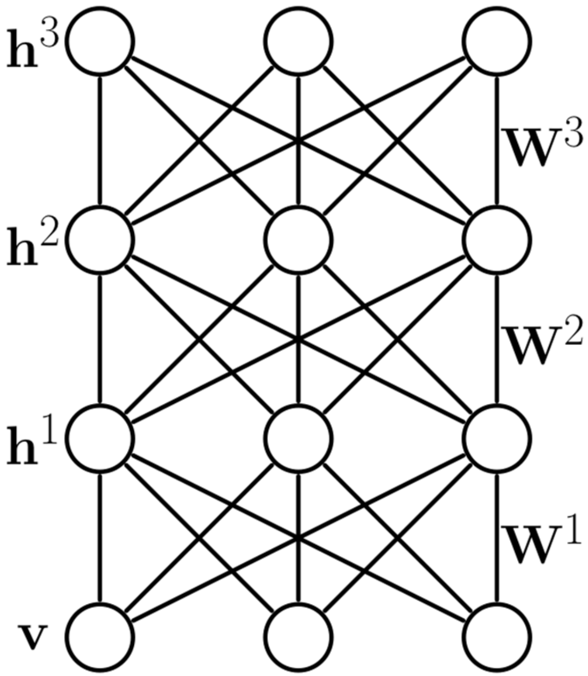
$$P_{data}(\mathbf{v}, \mathbf{h}^{\mathbf{1}}) = P_\theta(\mathbf{h}^{\mathbf{1}}|\mathbf{v}) P_{data}(\mathbf{v})$$

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

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)

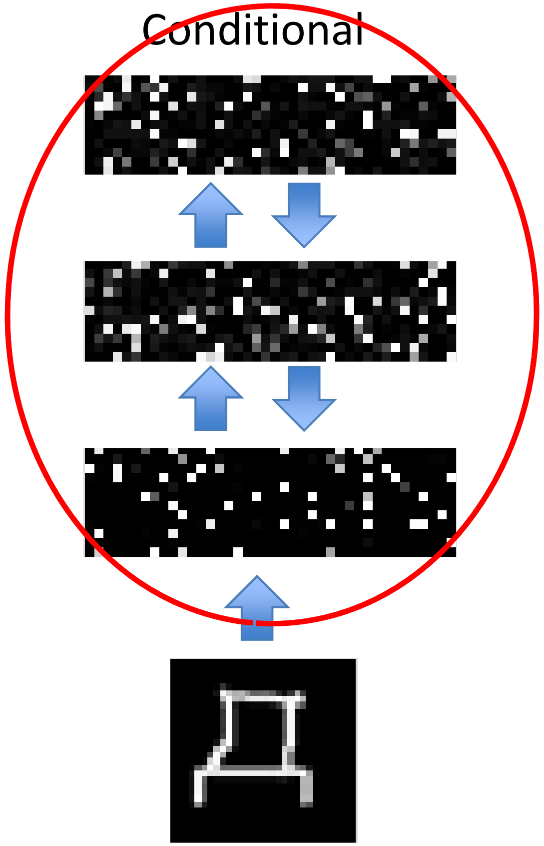
Real-world applications – thousands of hidden and observed variables with millions of parameters.

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



Approximate conditional

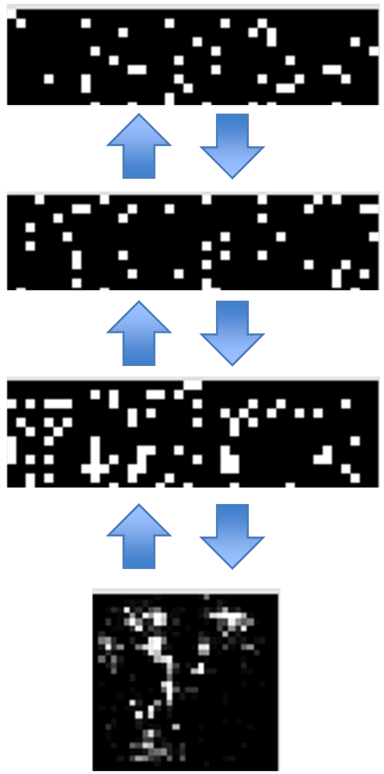
$P_{data}(\mathbf{h}|\mathbf{v})$

Simulate from the Model

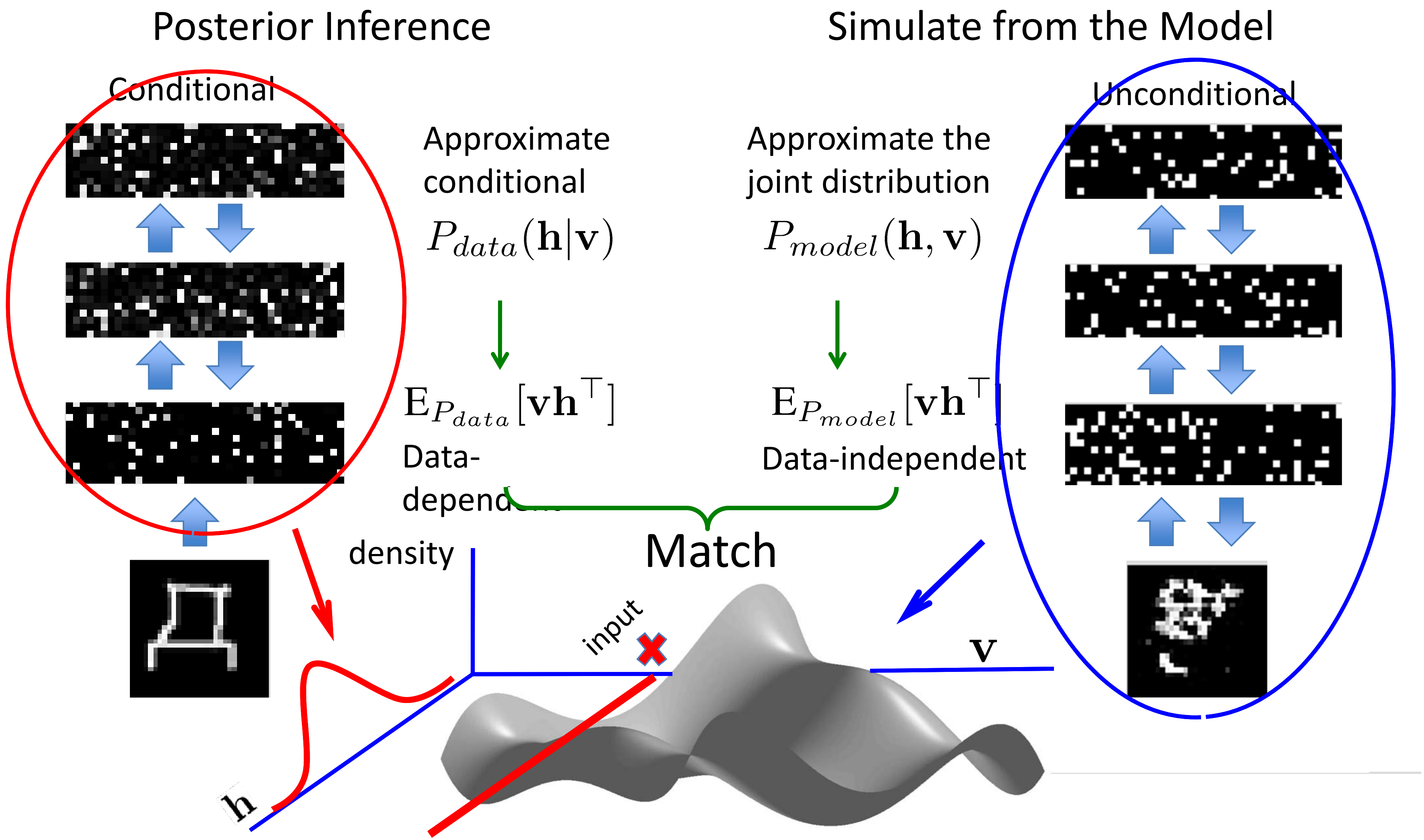
Approximate the joint distribution

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

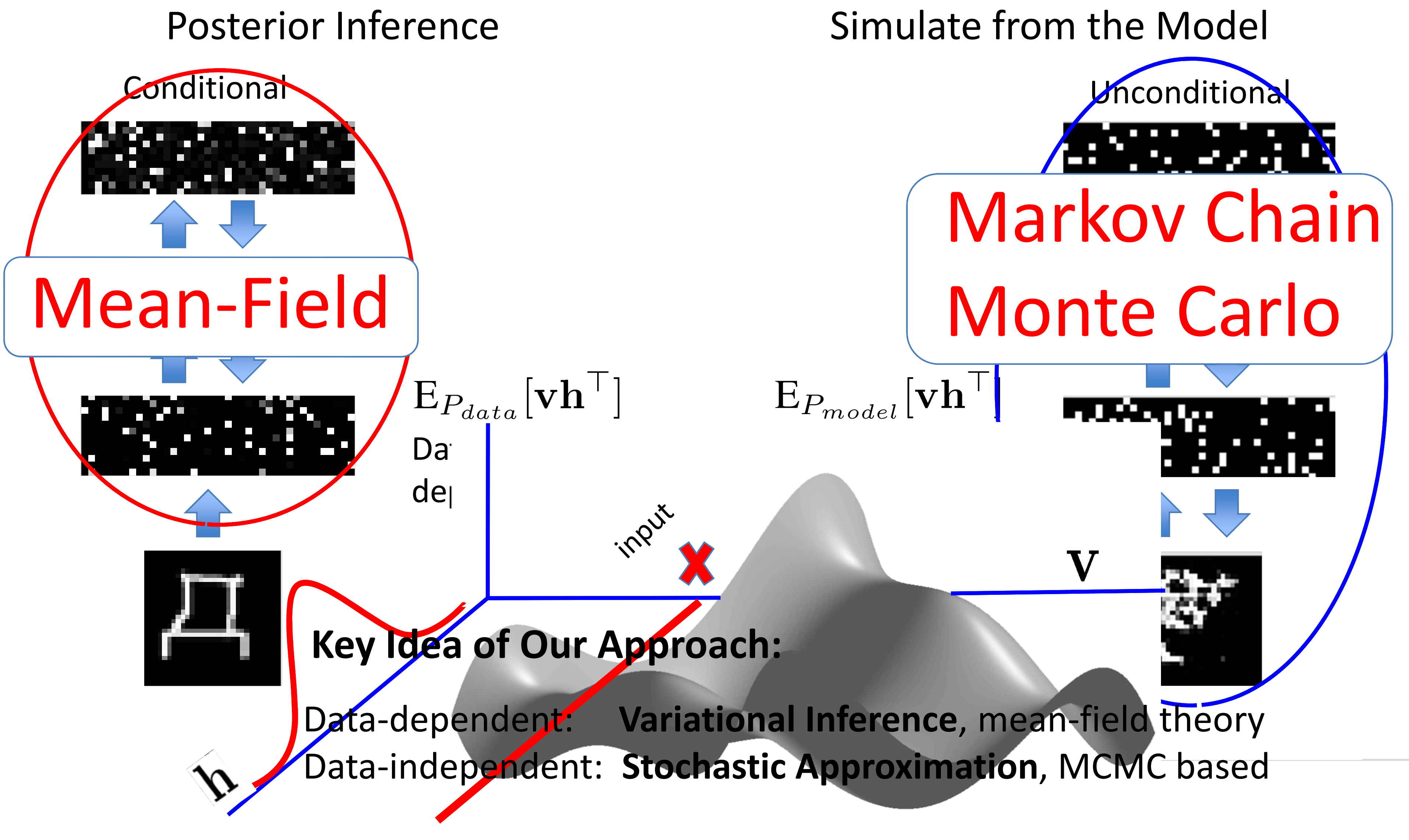
Unconditional



New Learning Algorithm



New Learning Algorithm



Key Idea of Our Approach:

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

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

Variational Inference

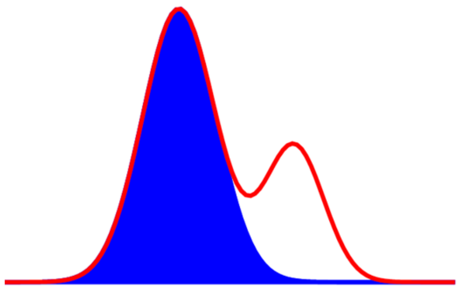
(Salakhutdinov, 2008; Salakhutdinov & Larochelle, AI & Statistics 2010)

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

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

$$\log P_\theta(\mathbf{v}) \geq \log P_\theta(\mathbf{v})$$

$$= \sum_{\mathbf{h}} Q_\mu$$



Mean-F

(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

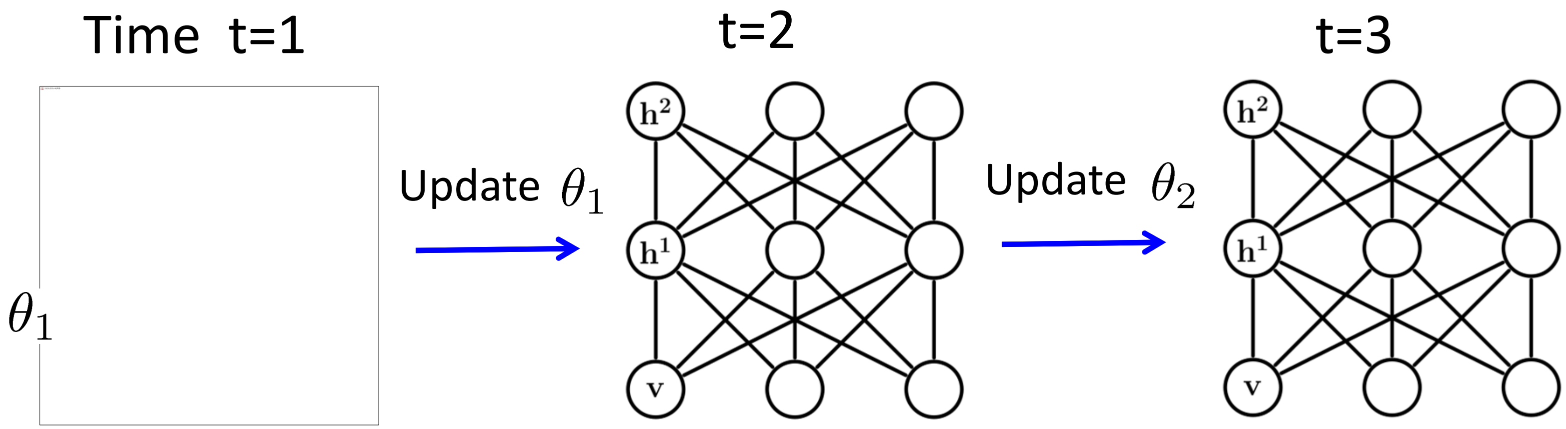
Variational In
lower bound w.r.t. variational parameters μ .

Nonlinear fixed-point equations:

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

$$\mu_m^{(3)} = \sigma\Big(\sum_k W_{km}^3 \mu_k^{(2)}\Big)$$

Stochastic Approximation



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

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, 1954
L. Younes, Probability Theory 1989

Learning Algorithm

Update rule decomposes:

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

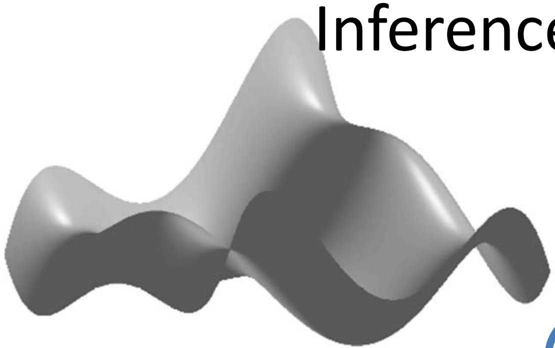
True gradient

Perturbation term ϵ_t

Variational Inference

MCMC

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



Problem: High-dimensional data:

y multimodal.

Fast Inference

Key insight: The transition operator can be

Learning can scale to millions of examples

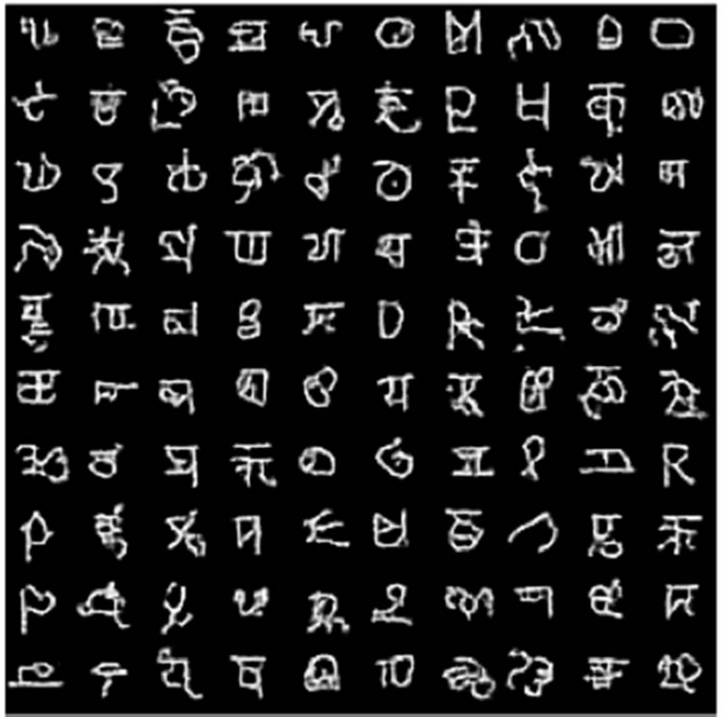
Connections to the theory of stochastic approximation and adaptive MCMC.

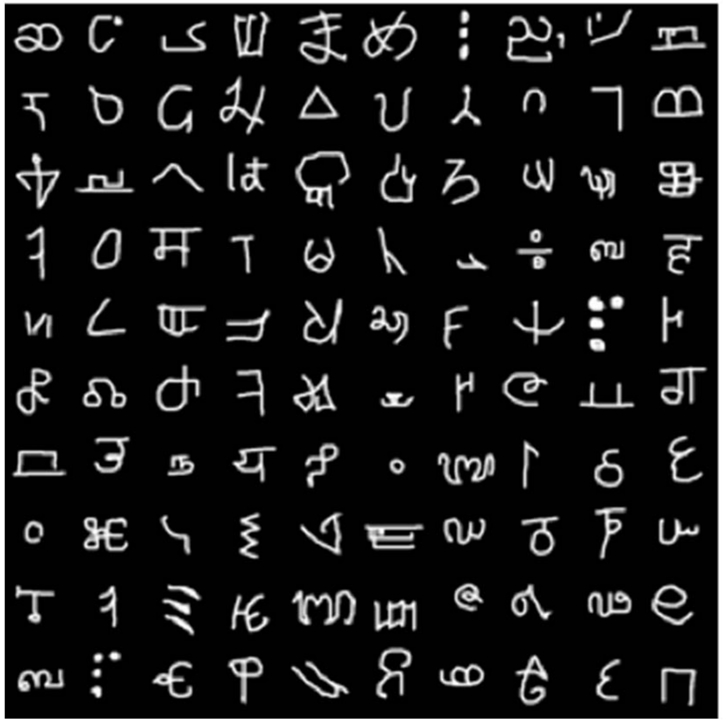
Good Generative Model?

Handwritten Characters

Good Generative Model?

Handwritten Characters





Good Generative Model?

Handwritten Characters

Simulated

Real Data

Good Generative Model?

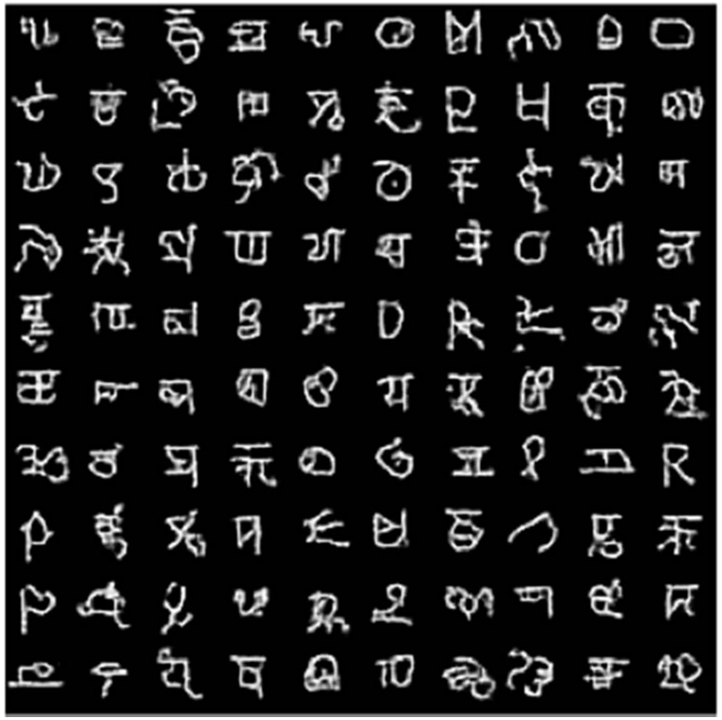
Handwritten Characters

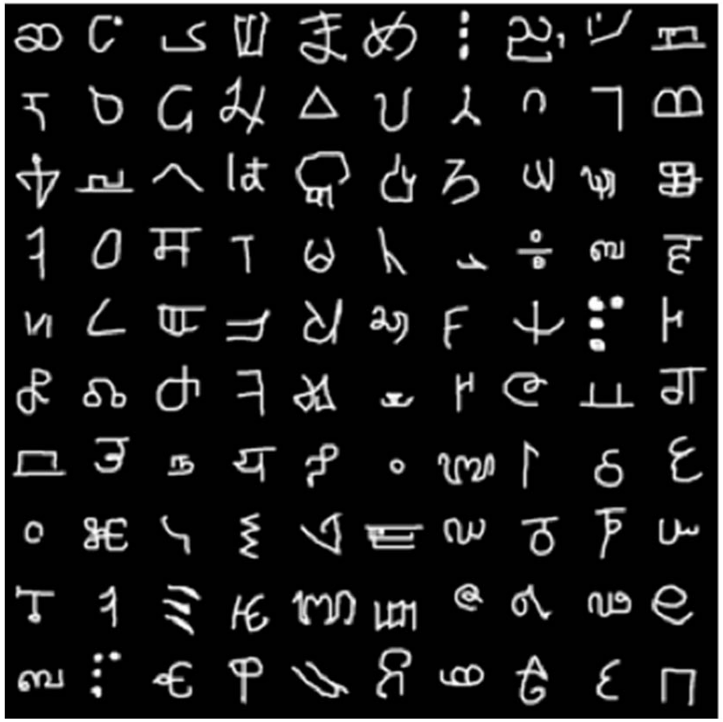
Real Data

Simulated

Good Generative Model?

Handwritten Characters





Good Generative Model?

MNIST Handwritten Digit Dataset

Handwriting Recognition

MNIST Dataset
60,000 examples of 10 digits

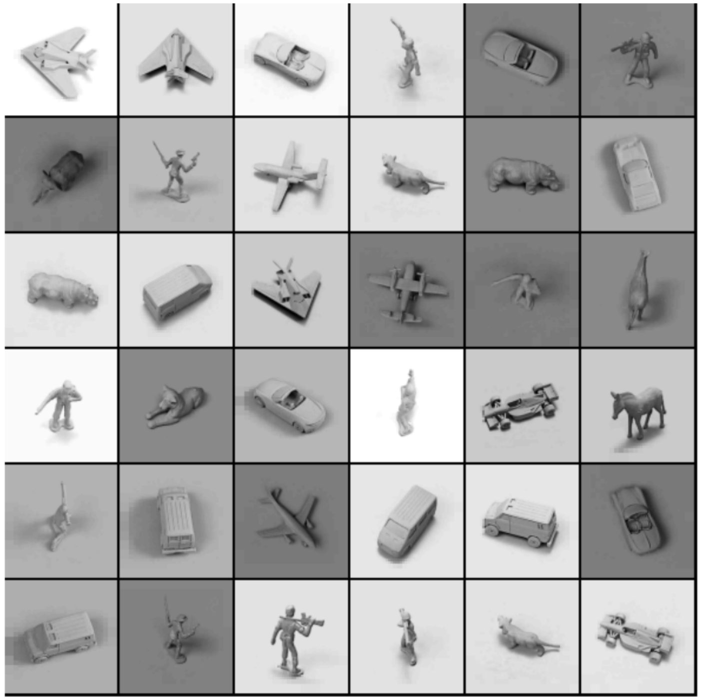
Learning Algorithm	Error
Logistic regression	12.0%
K-NN	3.09%
Neural Net (Platt 2005)	1.53%
SVM (Decoste et.al. 2002)	1.40%
Deep Autoencoder (Bengio et. al. 2007)	1.40%
Deep Belief Net (Hinton et. al. 2006)	1.20%
DBM	**0.95%**

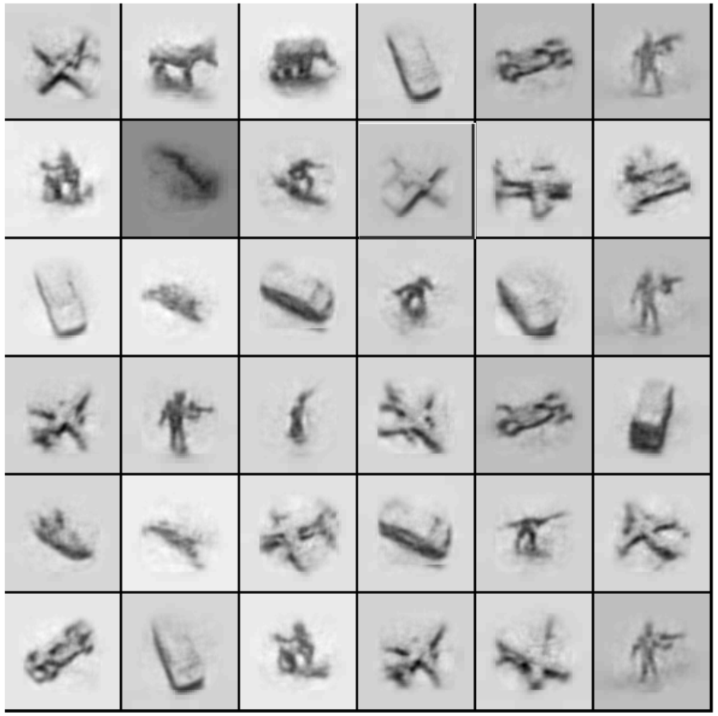
Optical Character Recognition
42,152 examples of 26 English letters

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

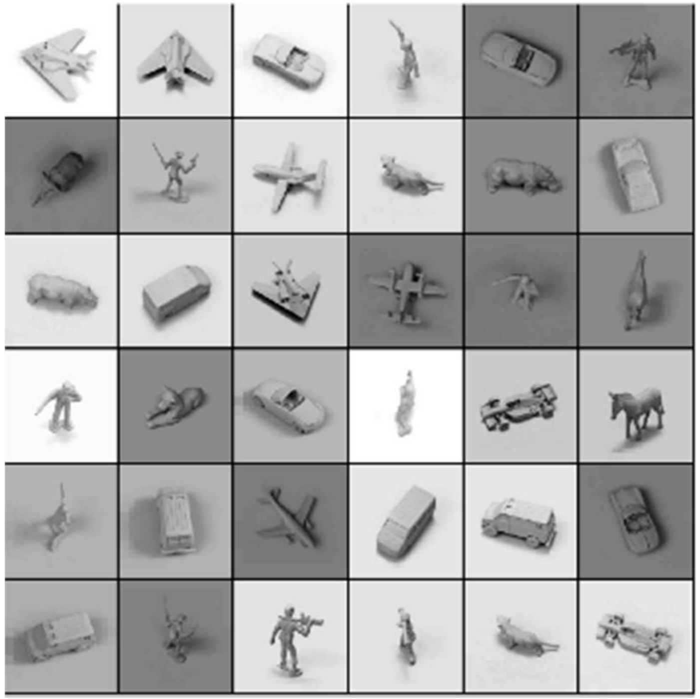




24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.

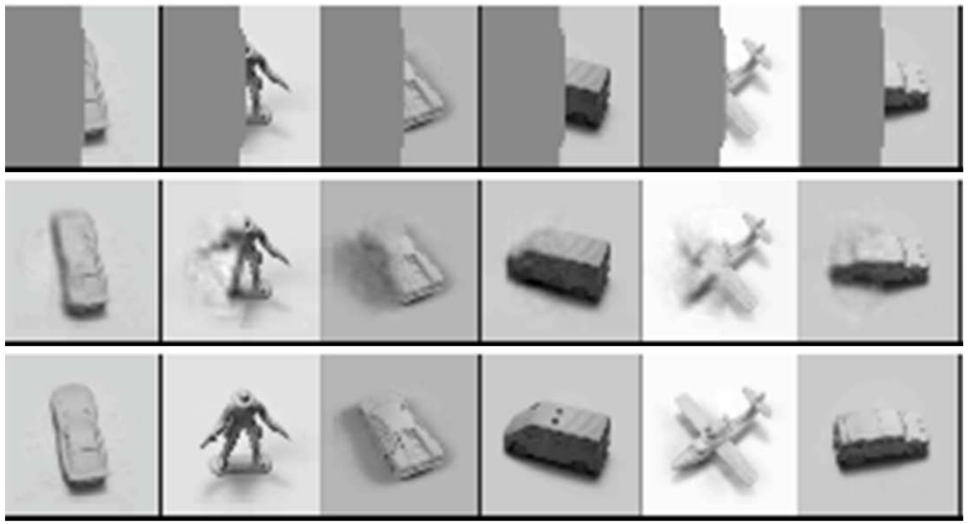
3-D object Recognition

NORB Dataset: 24,000 examples



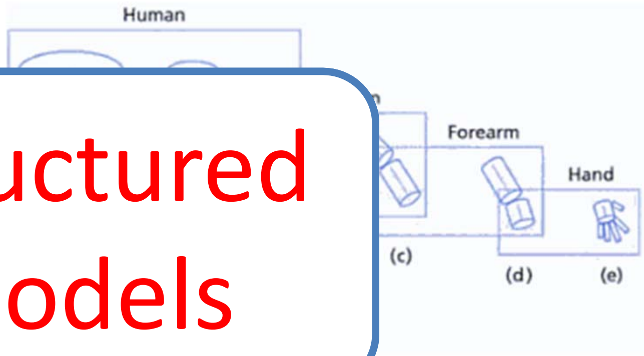
Learning Algorithm	Error
Logistic regression	22.5%
K-NN (LeCun 2004)	18.92%
SVM (Bengio & LeCun 2007)	11.6%
Deep Belief Net (Nair & Hinton 2009)	9.0%
DBM	**7.2%**

Pattern Completion



Learning Hierarchical Representations

Deep Boltzmann Machines:



Learning Hi
in Features
of edges.

Need more structured
and robust models

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

Talk Roadmap

Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Advanced Deep Models.

- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning

Face Recognition

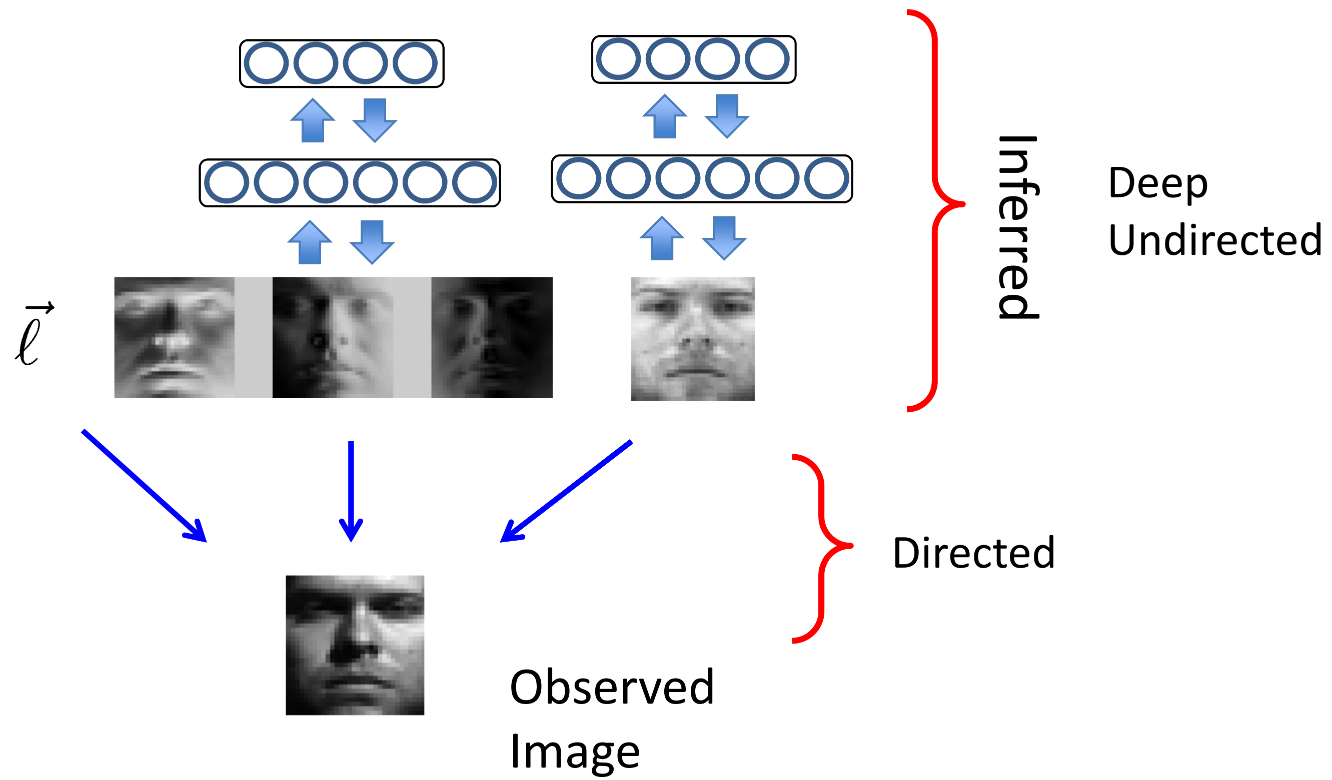
Yale B Extended Face Dataset
4 subsets of increasing illumination variations

Subset 1
Subset 2
Subset 3
Subset 4

Due to extreme illumination variations, deep models perform quite poorly on this dataset.

Deep Lambertian Model

Consider More Structured Models: undirected + directed models.



Combines the elegant properties of the Lambertian model with the Gaussian DBM model.

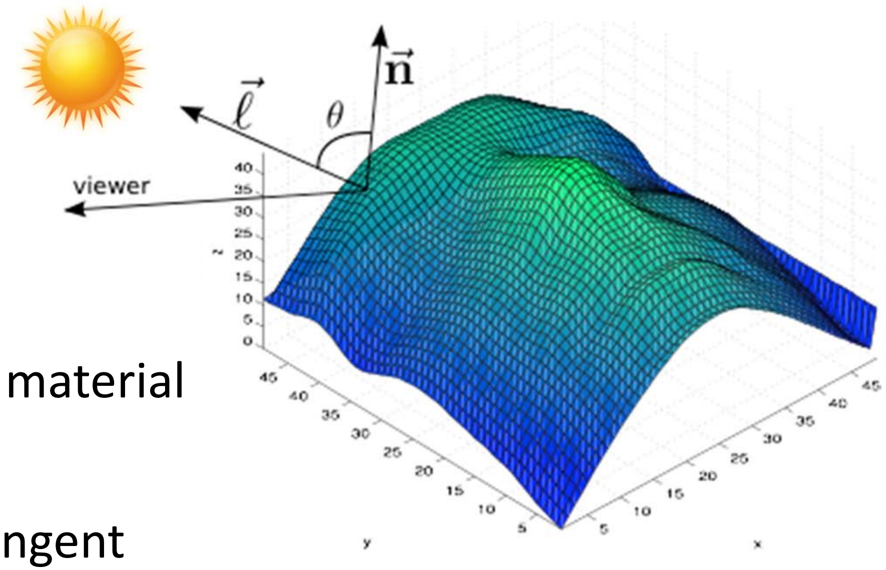
(Tang et. Al., ICML 2012, Tang et. al. CVPR 2012)

Lambertian Reflectance Model

- A simple model of the image formation process.

$$I = a \times |\vec{\ell}||\vec{\mathbf{n}}| \cos(\theta)$$

Image albedo — Light source — Surface normal



- Albedo -- diffuse reflectivity of a surface, material dependent, illumination independent.
- Surface normal -- perpendicular to the tangent plane at a point on the surface.

- Images with different illumination can be generated by varying light directions

Deep Lambertian Model

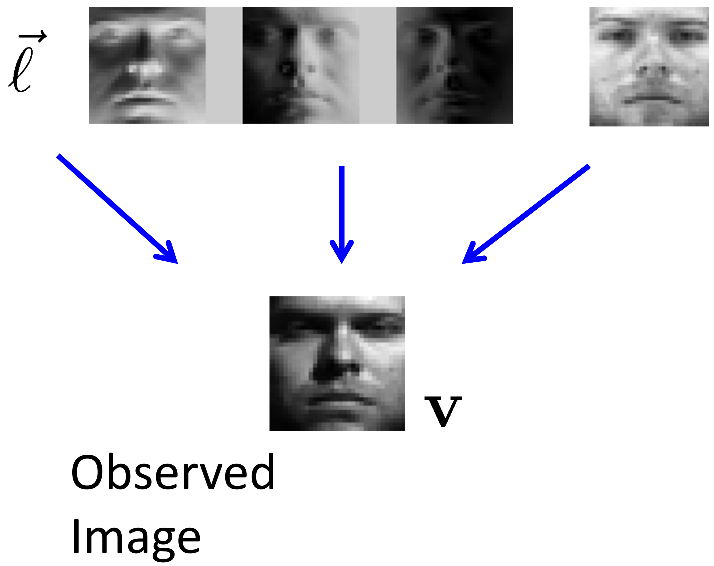
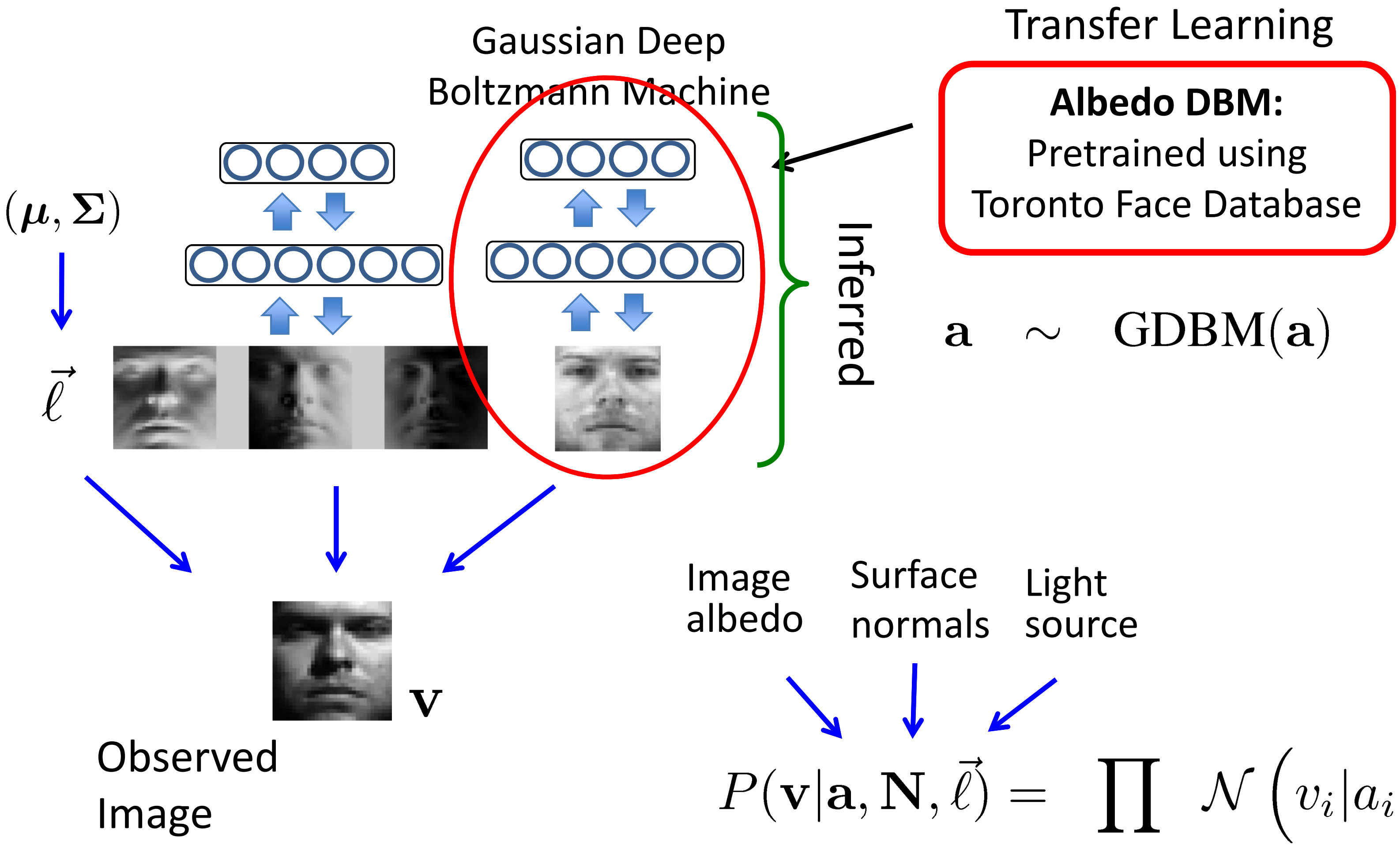


Image albedo — Surface normals — Light source

$$P(\mathbf{v}|\mathbf{a},\mathbf{N},\vec{\ell}) = \prod_{i \in \text{pixels}} \mathcal{N}\left(v_i | a_i(\vec{\mathbf{n}}_i^\top \vec{\ell}), \sigma^2\right)$$

$$\mathbf{a} \in \mathbb{R}^D, \quad \mathbf{N} \in \mathbb{R}^{D\times 3}, \quad \ell \in \mathbb{R}^3$$

Deep Lambertian Model



$$\mathbf{a} \sim \text{GDBM}(\mathbf{a})$$

Image albedo, Surface normals, Light source

$$P(\mathbf{v}|\mathbf{a}, \mathbf{N}, \vec{\ell}) = \prod_{i \in \text{pixels}} \mathcal{N}\left(v_i | a_i(\vec{\mathbf{n}}_i^\top \vec{\ell}), \sigma^2\right)$$

$$\mathbf{a} \in \mathbb{R}^D, \quad \mathbf{N} \in \mathbb{R}^{D \times 3}, \quad \ell \in \mathbb{R}^3$$

Inference: Variational Inference.

Learning: Stochastic Approximation

Yale B Extended Face Dataset

• 38 subjects, ~ 45 images of varying illuminations per subject, divided into 4 subsets of increasing illumination variations.

• 28 subjects for training, and 10 for testing.

Face Relighting

One Test Image

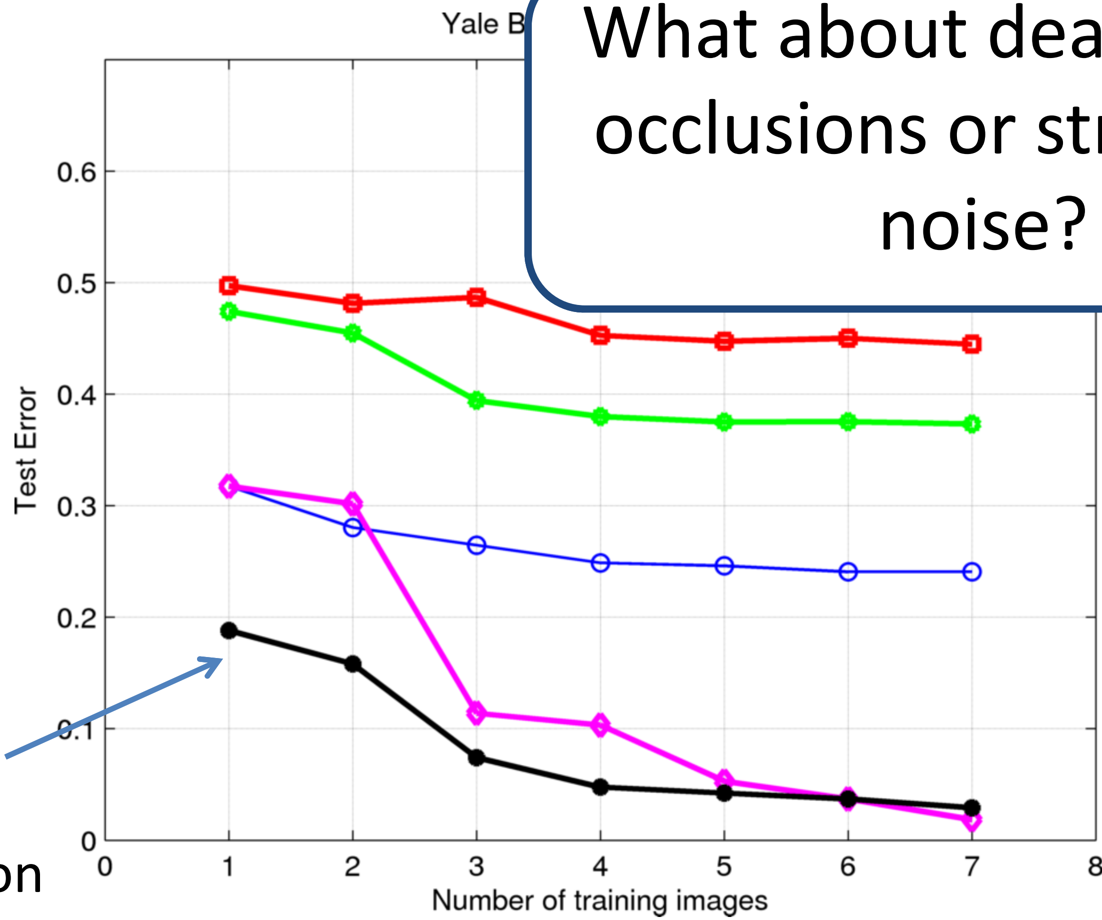
Observed

Inferred albedo

Face Relighting

Recognition Results

Recognition as function of the number of training images for 10 test subjects.



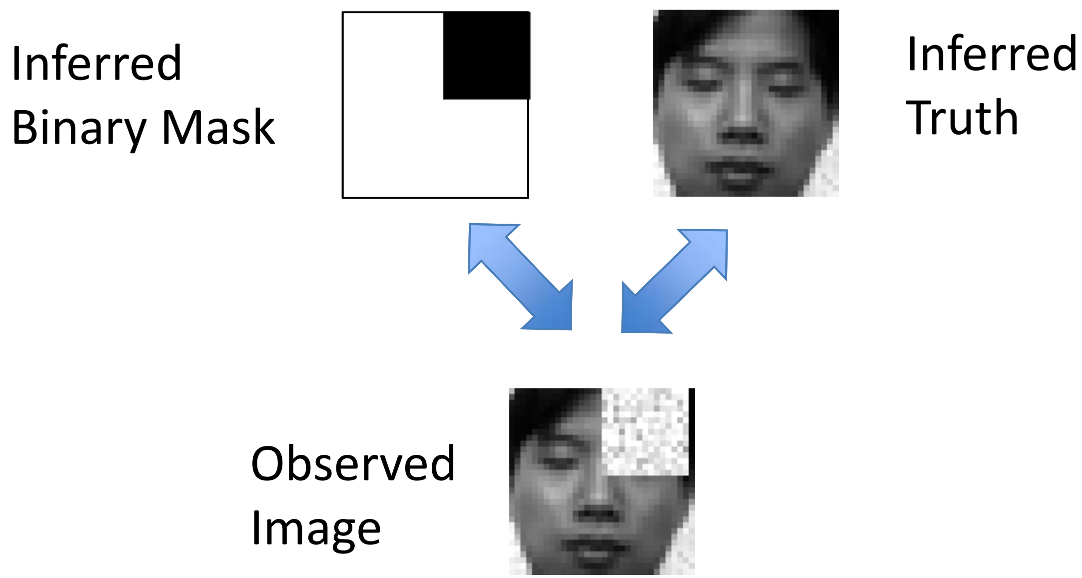
What about dealing with occlusions or structured noise?

One-Shot Recognition

Robust Boltzmann Machines

- Build more structured models that can deal with occlusions or structured noise.

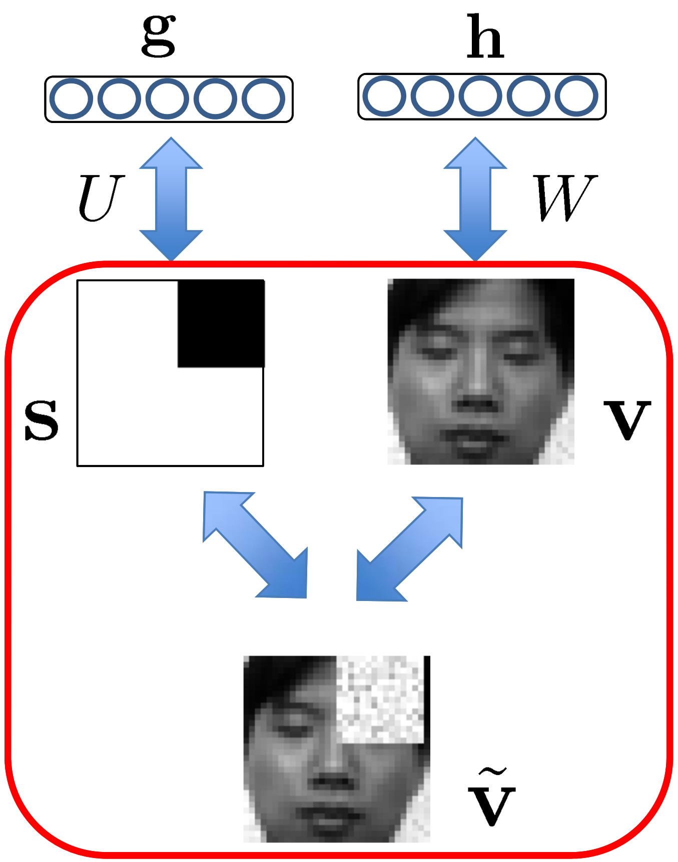
$$\log P(\tilde{\mathbf{v}}, \mathbf{v}, \mathbf{s}, \mathbf{h}, \mathbf{g}) \sim$$



(Tang et. Al., ICML 2012, Tang et. al. CVPR 2012)

Robust Boltzmann Machines

- Build more structured models that can deal with occlusions or structured noise.



Observed Image

$$\log P(\tilde{\mathbf{v}}, \mathbf{v}, \mathbf{s}, \mathbf{h}, \mathbf{g}) \sim$$

$$-\frac{1}{2}\sum_{i\in\text{pixels}} \frac{(v_i - b_i)^2}{\sigma_i^2} + \mathbf{v}^\top W \mathbf{h}$$

Gaussian RBM, modeling clean faces

Binary RBM modeling occlusions

$$-\frac{1}{2}\sum_{i\in\text{pixels}} \gamma_i s_i (v_i - \tilde{v}_i)^2$$

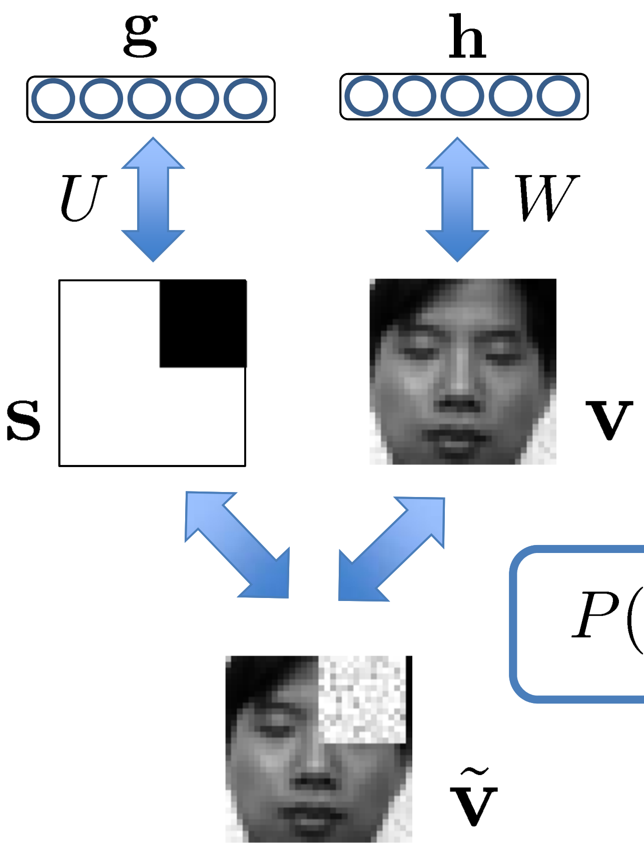
Binary pixel-wise Mask

Gaussian noise model

(Tang et. Al., ICML 2012, Tang et. al. CVPR 2012)

Robust Boltzmann Machines

- Build more structured models that can deal with occlusions or structured noise.



$$\log P(\tilde{\mathbf{v}}, \mathbf{v}, \mathbf{s}, \mathbf{h}, \mathbf{g}) \sim$$

$$-\frac{1}{2}\sum_{i \in \text{pixels}} \frac{(v_i - b_i)^2}{\sigma_i^2} + \mathbf{v}^\top W\mathbf{h} + \mathbf{s}^\top U\mathbf{g}$$

Gaussian RBM, modeling clean faces

Binary RBM modeling occlusions

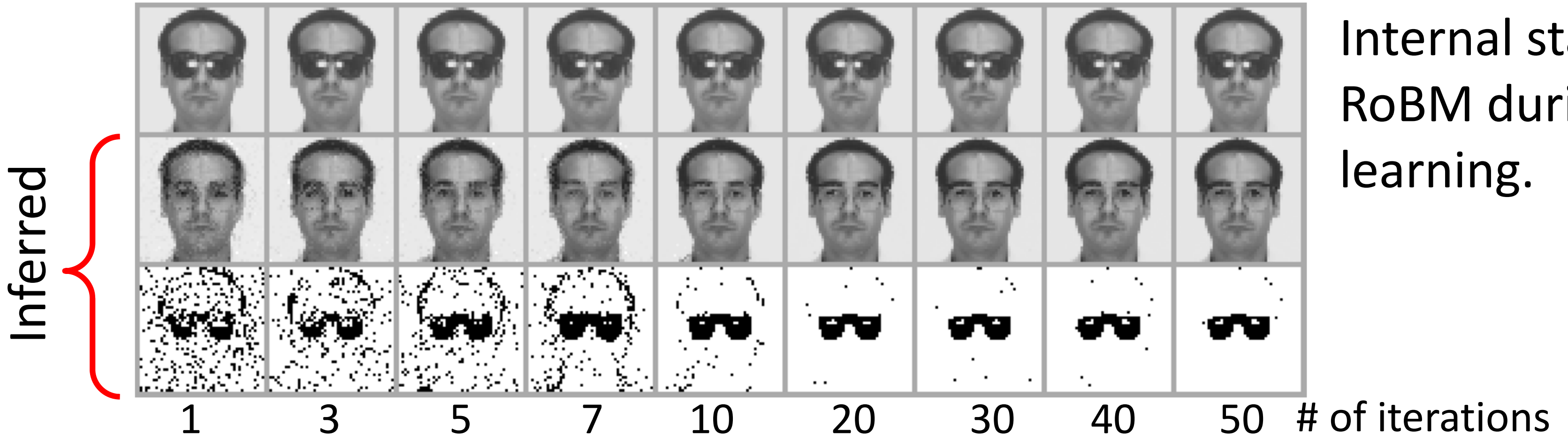
$P(\tilde{\mathbf{v}}|\mathbf{h}, \mathbf{g})$ is a heavy-tailed distribution

Binary pixel-wise Mask

Gaussian noise model

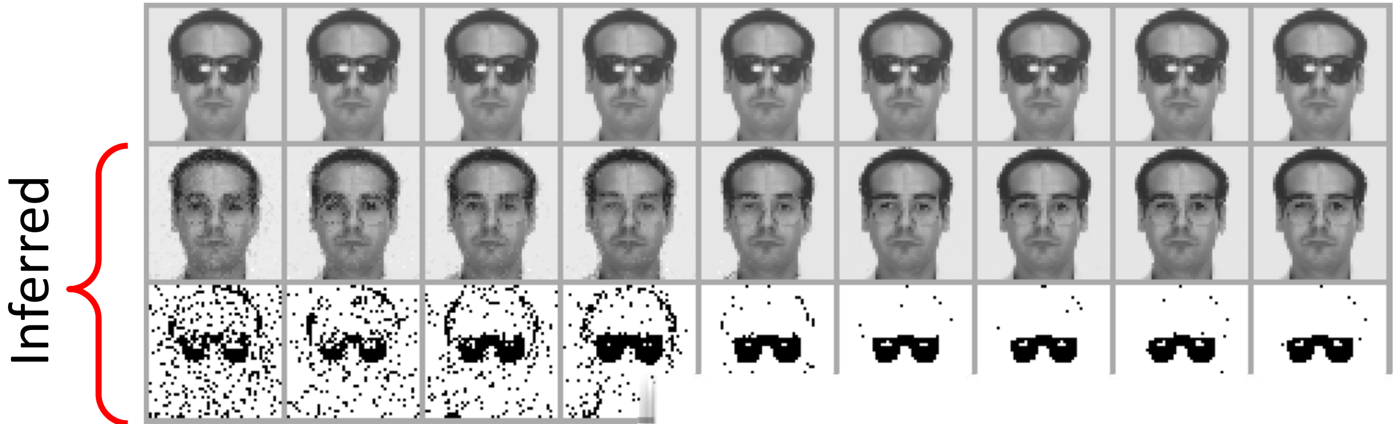
Inference: Variational Inference.
Learning: Stochastic Approximation

Recognition Results on AR Face Database



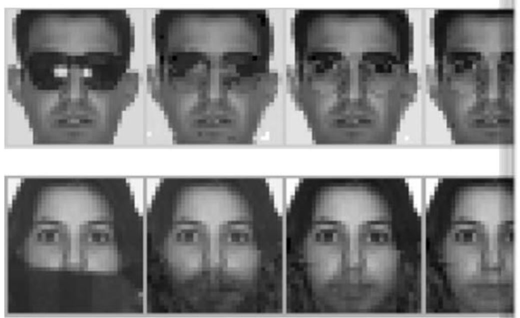
Internal states of RoBM during learning.

Recognition Results on AR Face Database



Internal states of RoBM during learning.

Inference on the



# of iteratio

Learning Algorithm	Sunglasses	Scarf
Robust BM	84.5%	80.7%
RBM	61.7%	32.9%
Eigenfaces	66.9%	38.6%
LDA	56.1%	27.0%
Pixel	51.3%	17.5%

Transfer Learning

How can we learn a novel concept – a high dimensional statistical object – from few examples.

Supervised Learning

Segway

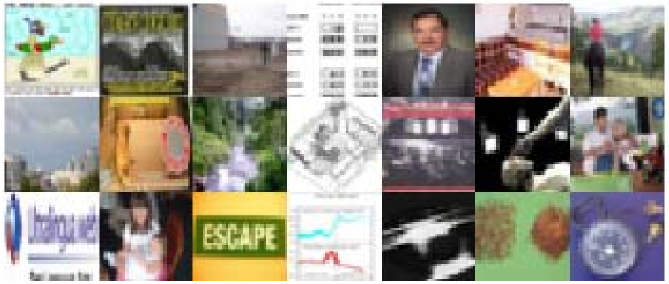
Motorcycle

Test:

Transfer Learning

Background Knowledge

Millions of unlabeled images



Some labeled images

Bicycle

Dolphin

Elephant

Tractor

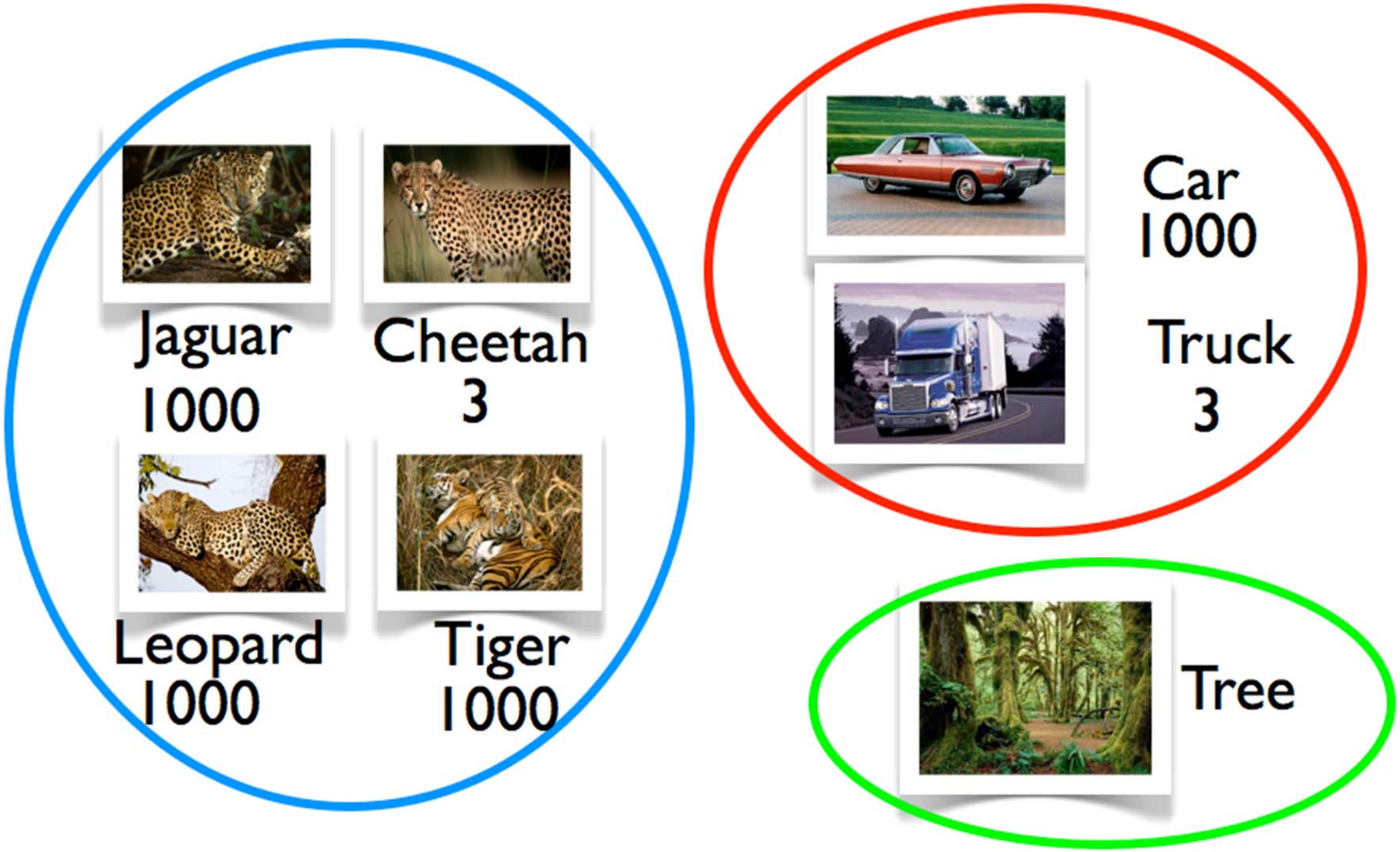
Learn to Transfer Knowledge

Learn novel concept from one example

Test:
What is this?

An Example

Structure in classes!



Slide Credit: Nitish Srivastava

Hierarchical-Deep Models

(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

HD Models: Integrate hierarchical Bayesian models with deep models.

One-Shot Learning

Hierarchical Bayes:

- Learn **hierarchies of categories** for sharing abstract knowledge.

Shared higher-level features

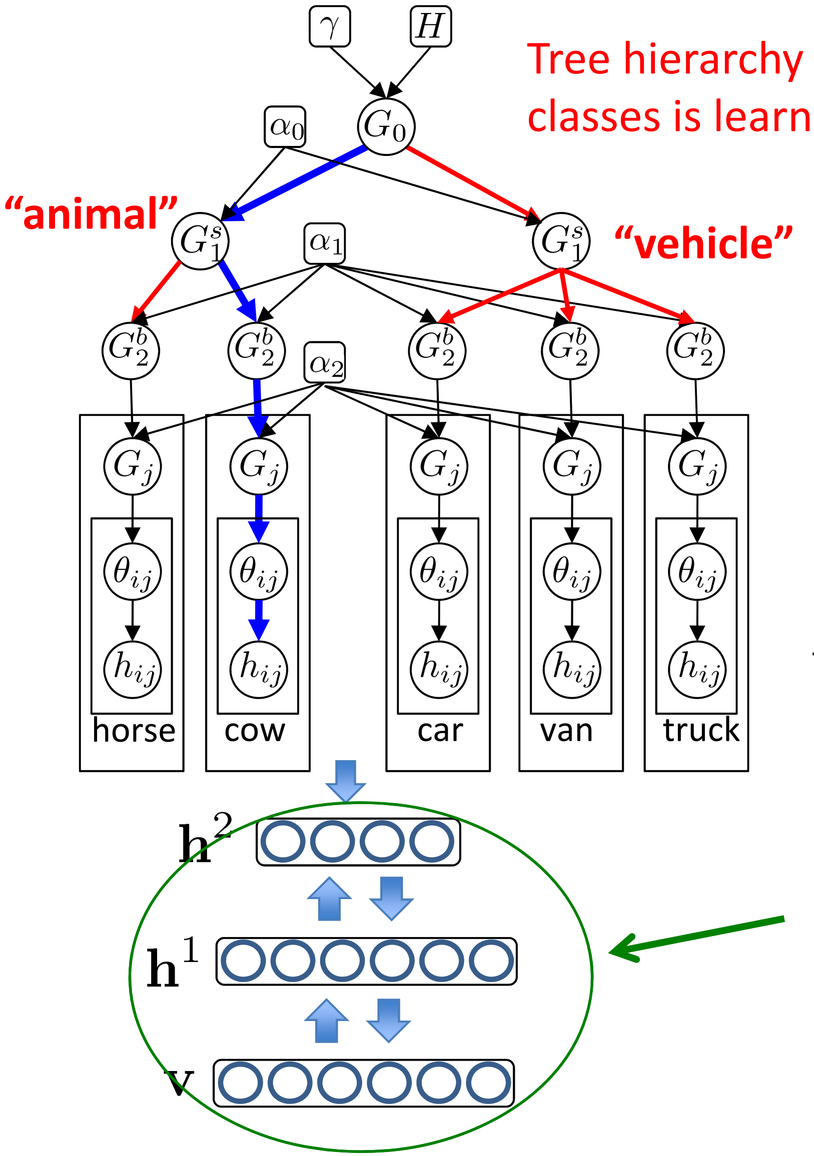
Deep Models:

- Learn **hierarchies of features.**
- **Unsupervised feature learning** – no need to rely on human-crafted input features.

Shared low-level features

Hierarchical-Deep Models

(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)



$\mathbf{z} \sim \mathrm{nCRP}$ (**Nested Chinese Restaurant Process**)
prior: a nonparametric prior over tree structures

$\mathbf{h}^3|\mathbf{z} \sim \mathrm{HDP}$ (**Hierarchical Dirichlet Process**) prior:
a nonparametric prior allowing categories to share higher-level features, or parts.

$\mathbf{v}|\mathbf{h}^3 \sim \mathrm{DBM}$ **Deep Boltzmann Machine**

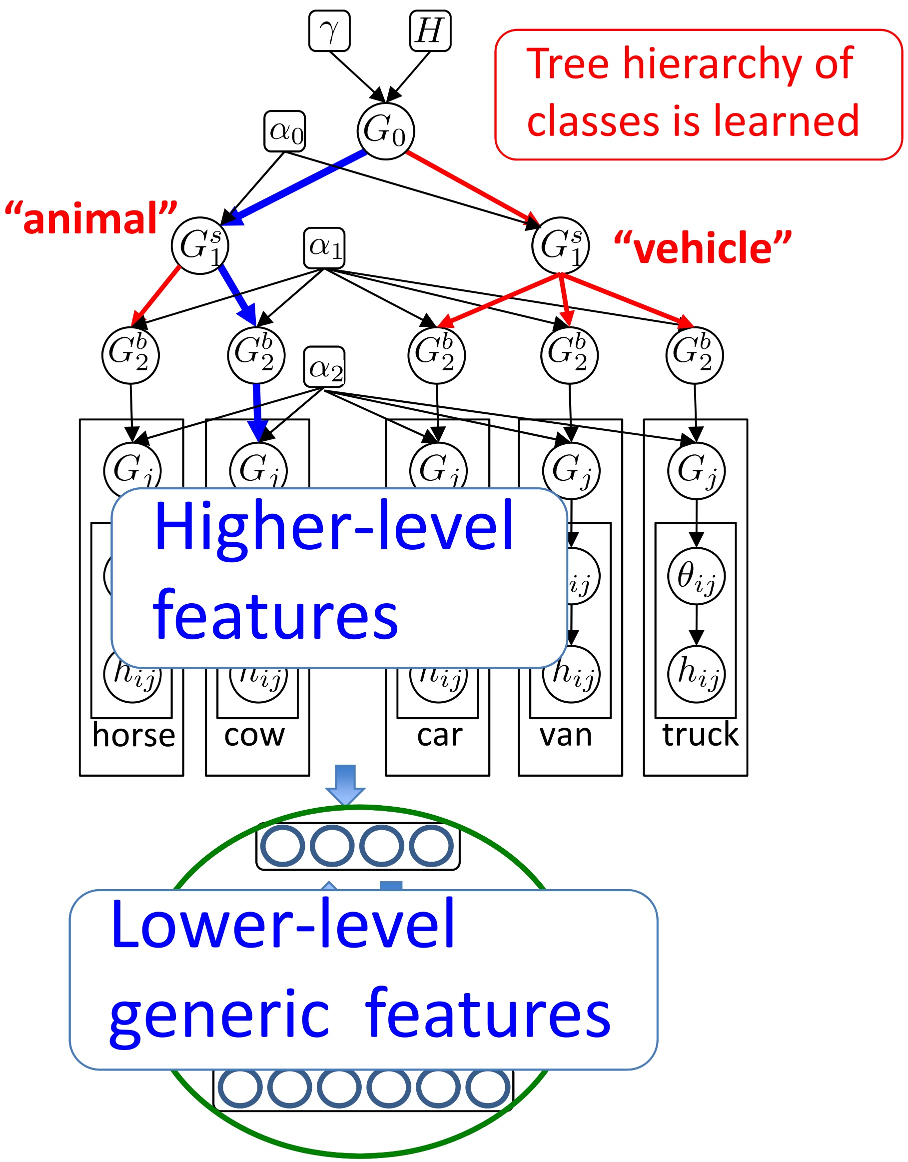
Enforce approximate global consistency through many local constraints.

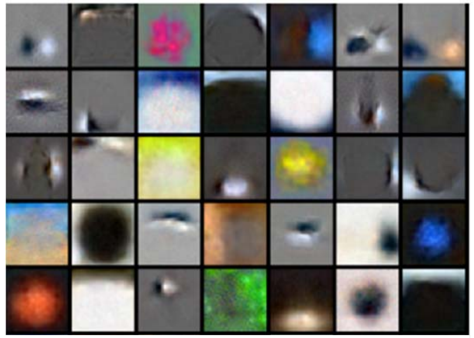
Incorporate prior knowledge to deal with occlusions, corrupted or missing data.

Images, Handwritten characters, Motion capture datasets.

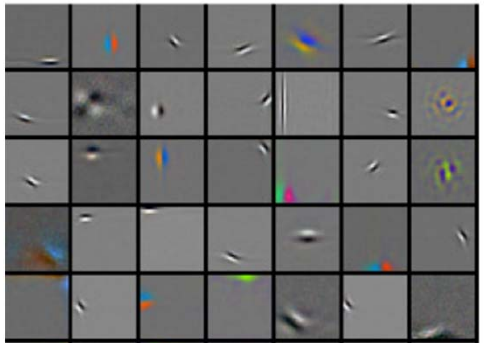
CIFAR Object Recognition

(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)



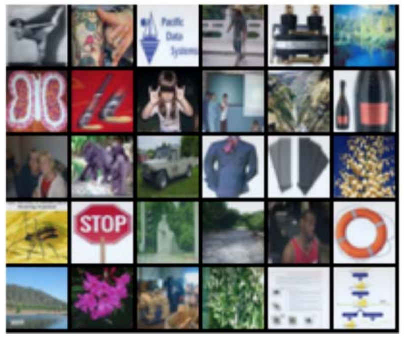


Learned high-level features



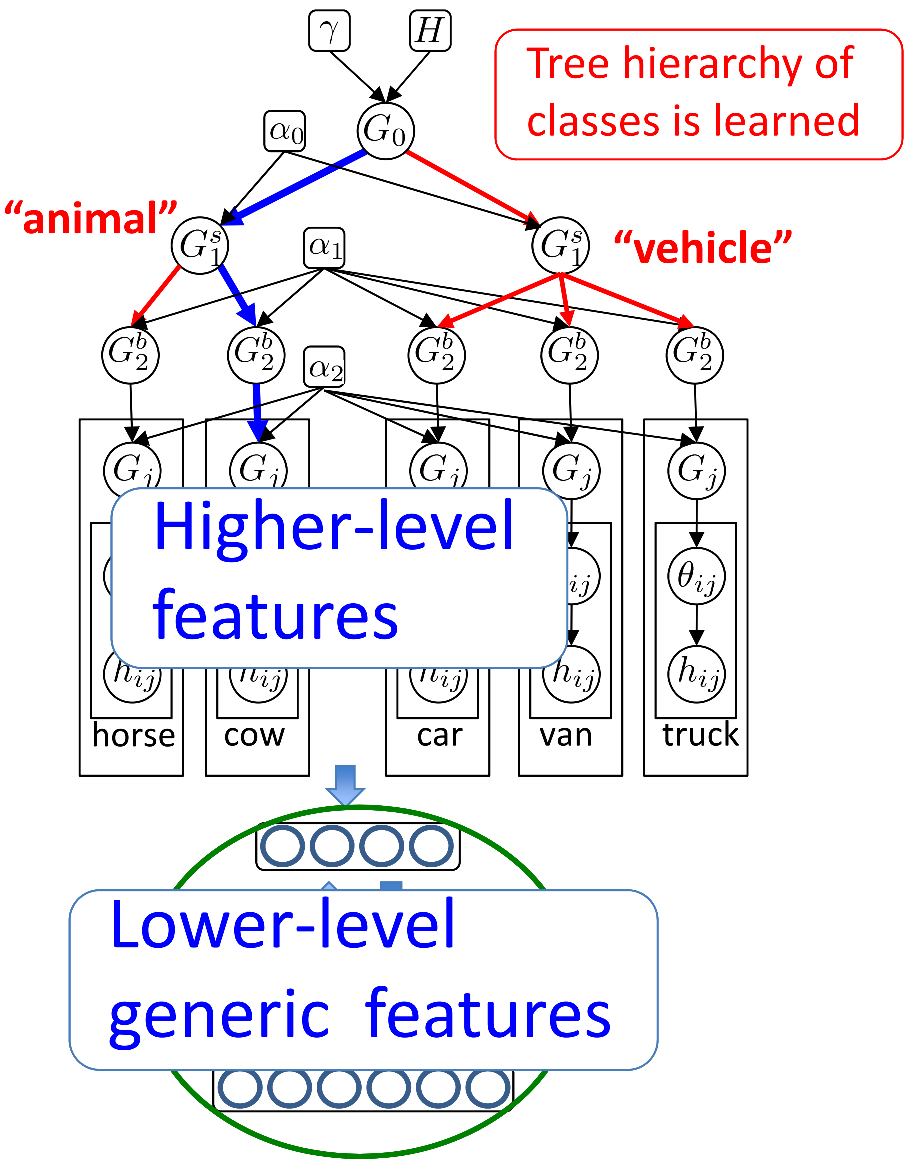
DBM generic features

4 million Images

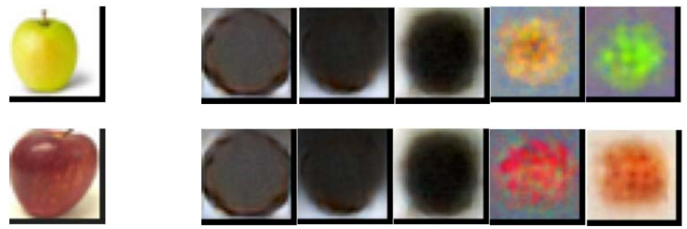


CIFAR Object Recognition

(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

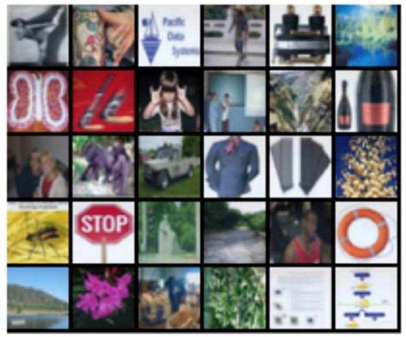


Each image is made up of learned high-level features features.



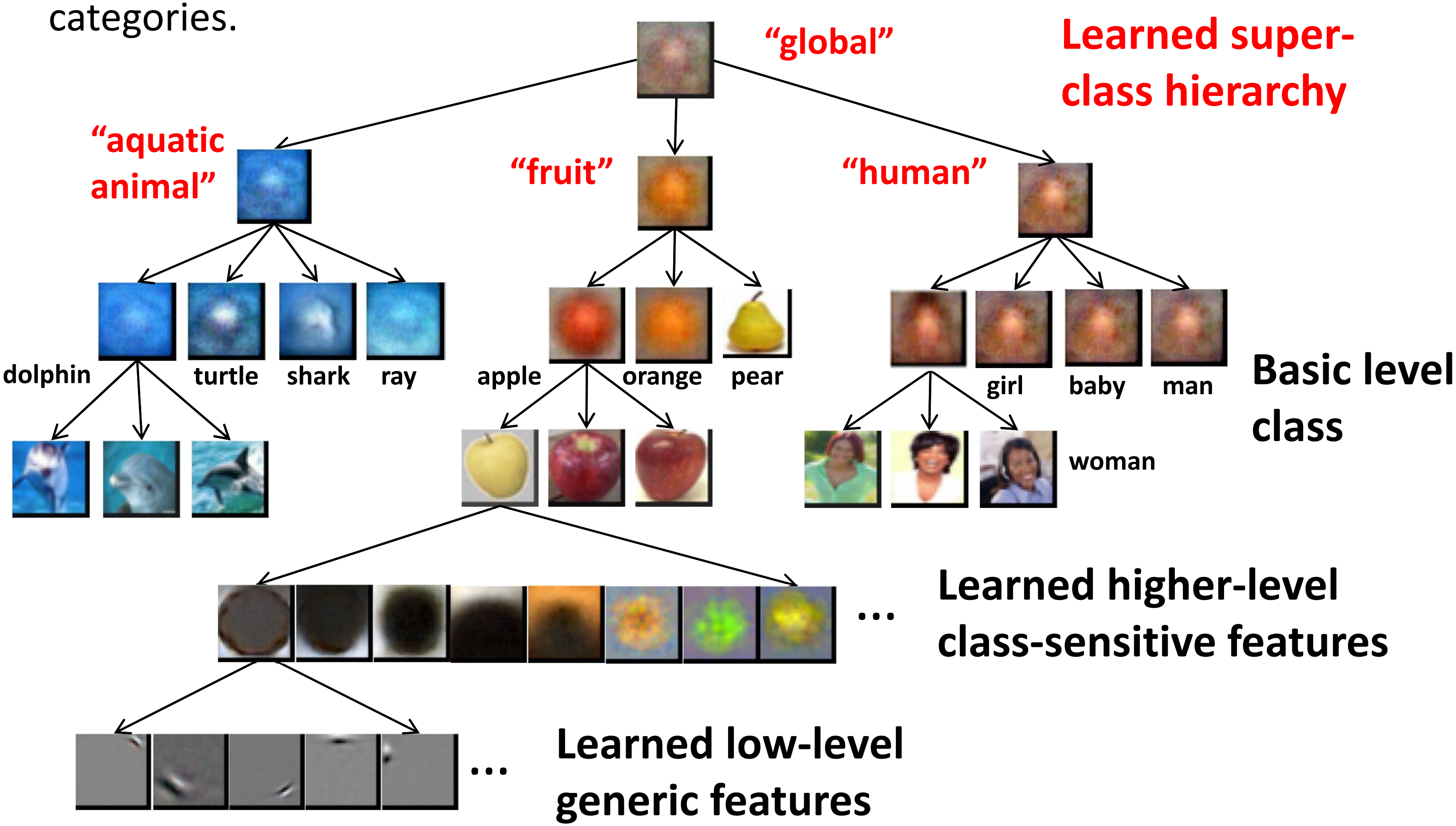
Each higher-level feature is made up of lower-level features.

4 million Images



Learning Category Hierarchy

The model learns how to share the knowledge across many visual categories.



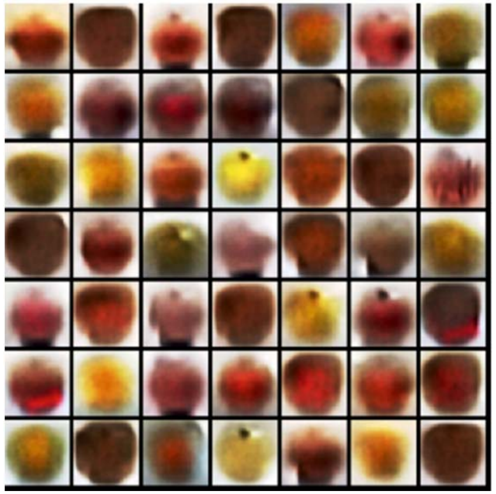
Learning from 3 Examples

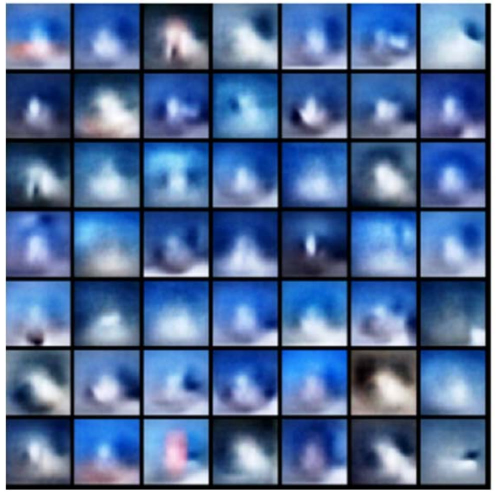
Given only 3 Examples

Willow Tree

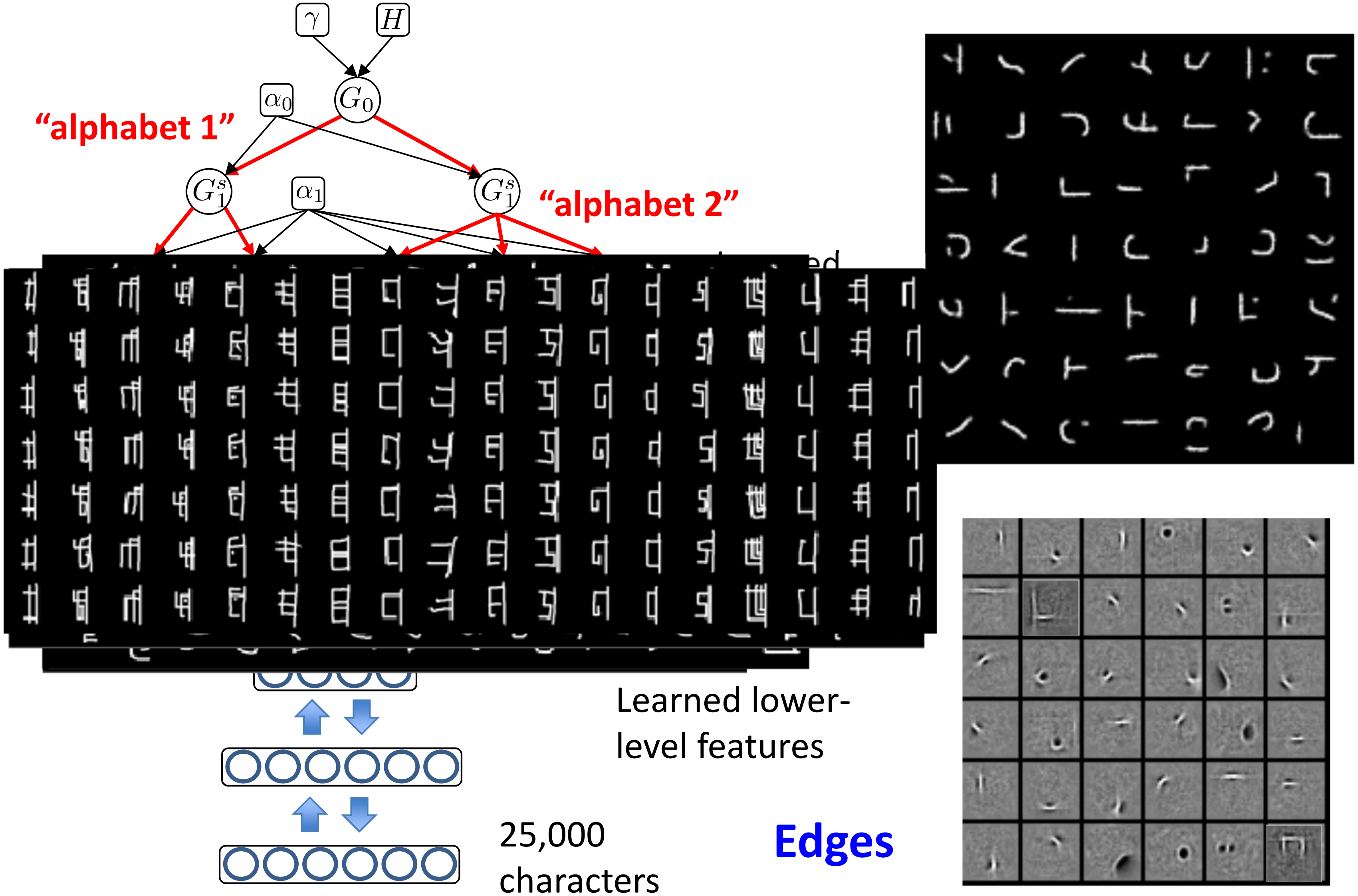
Rocket

Generated Samples

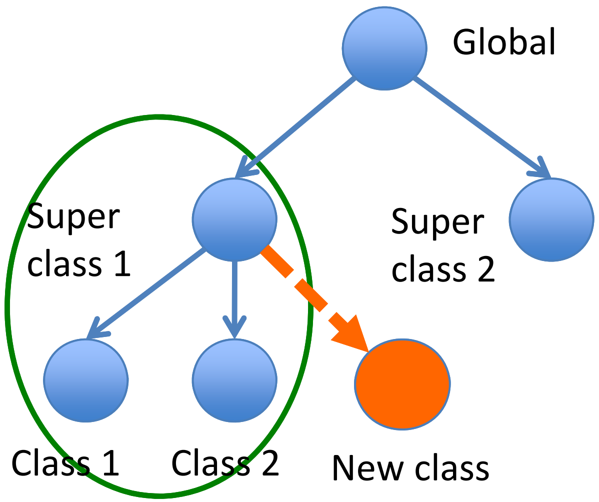




Handwritten Character Recognition



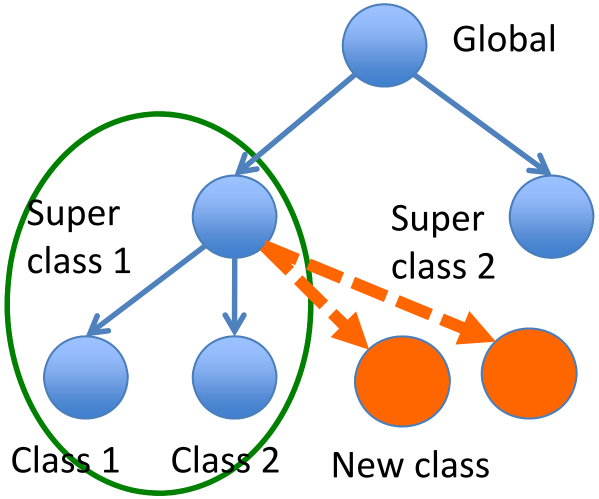
Simulating New Characters



Real data within super class

Simulated new characters

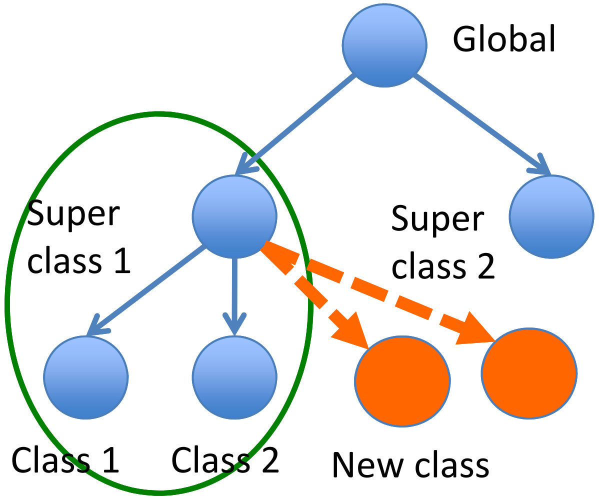
Simulating New Characters



Real data within super class

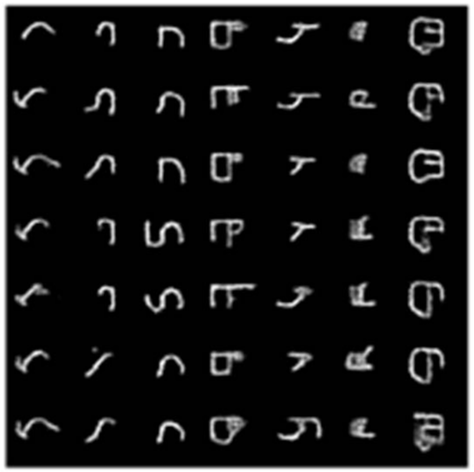
Simulated new characters

Simulating New Characters

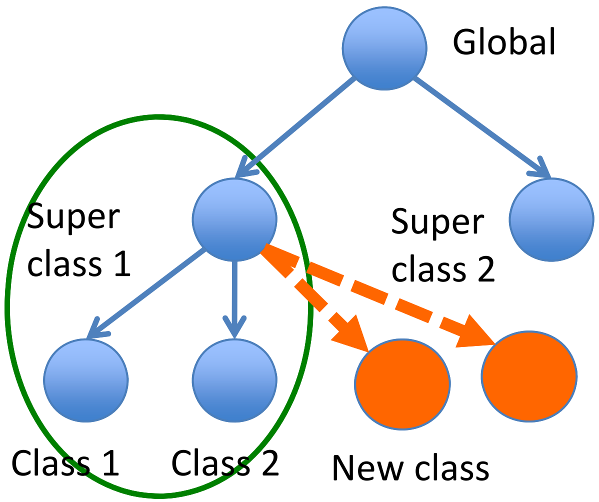


Real data within super class

Simulated new characters

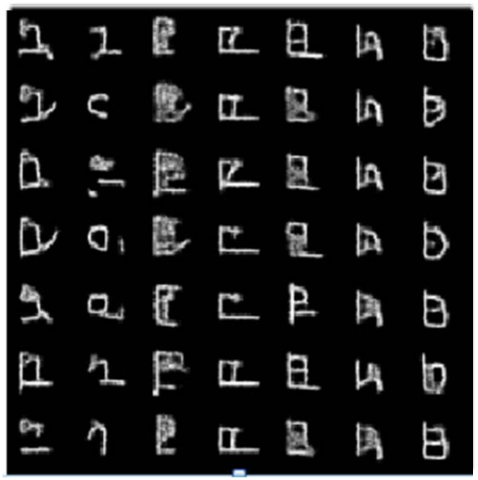


Simulating New Characters

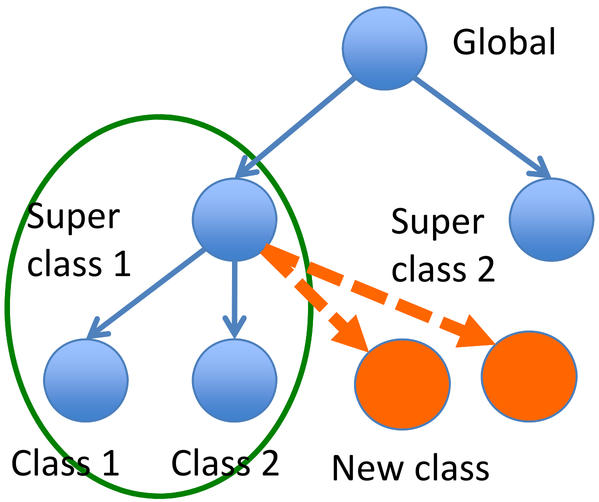


Real data within super class

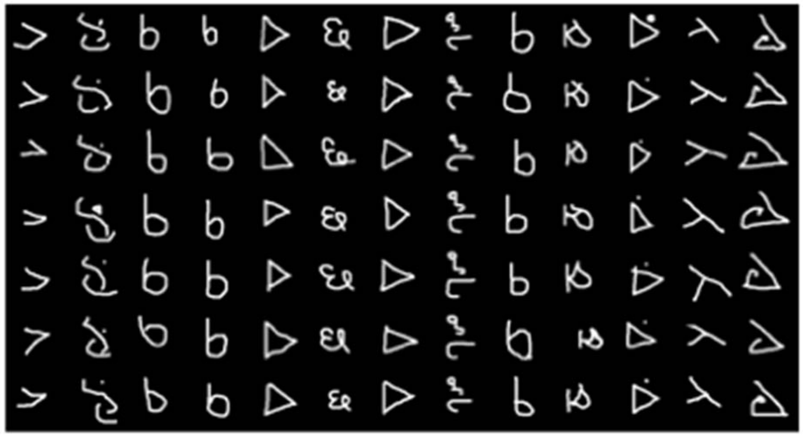
Simulated new characters



Simulating New Characters

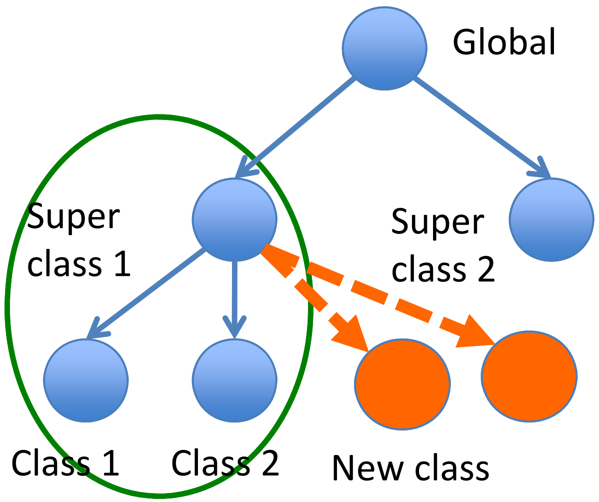


Real data within super class

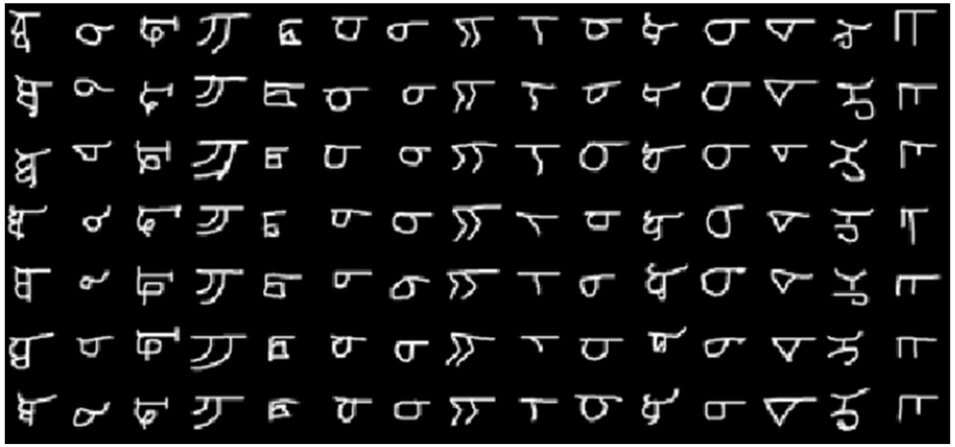


Simulated new characters

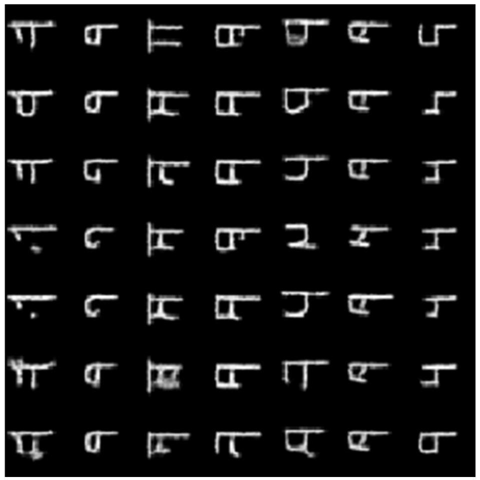
Simulating New Characters



Real data within super class

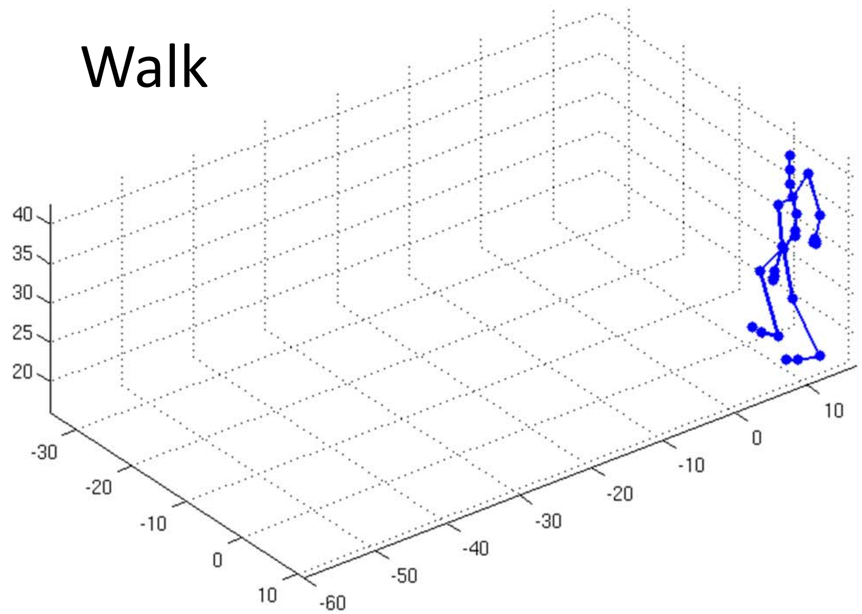


Simulated new characters

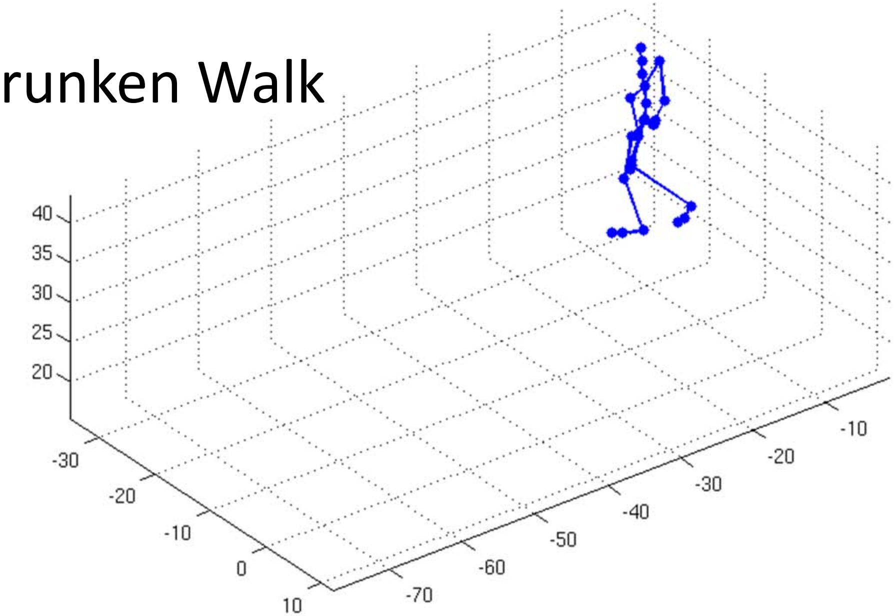


Motion Capture

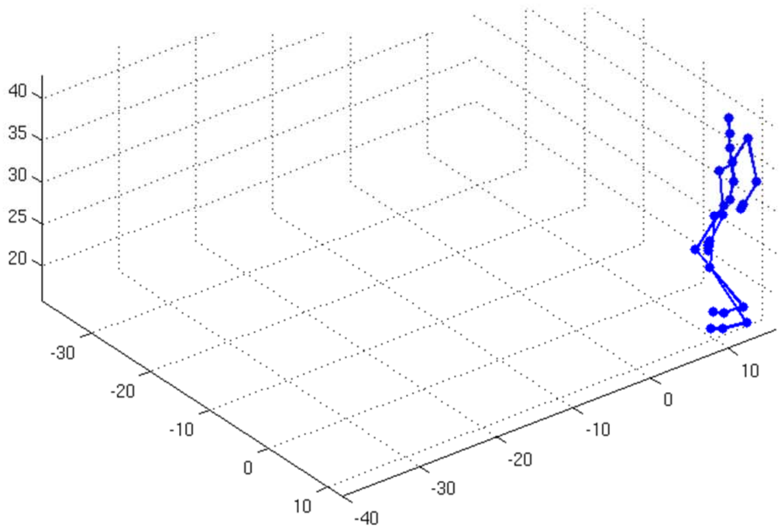
Walk



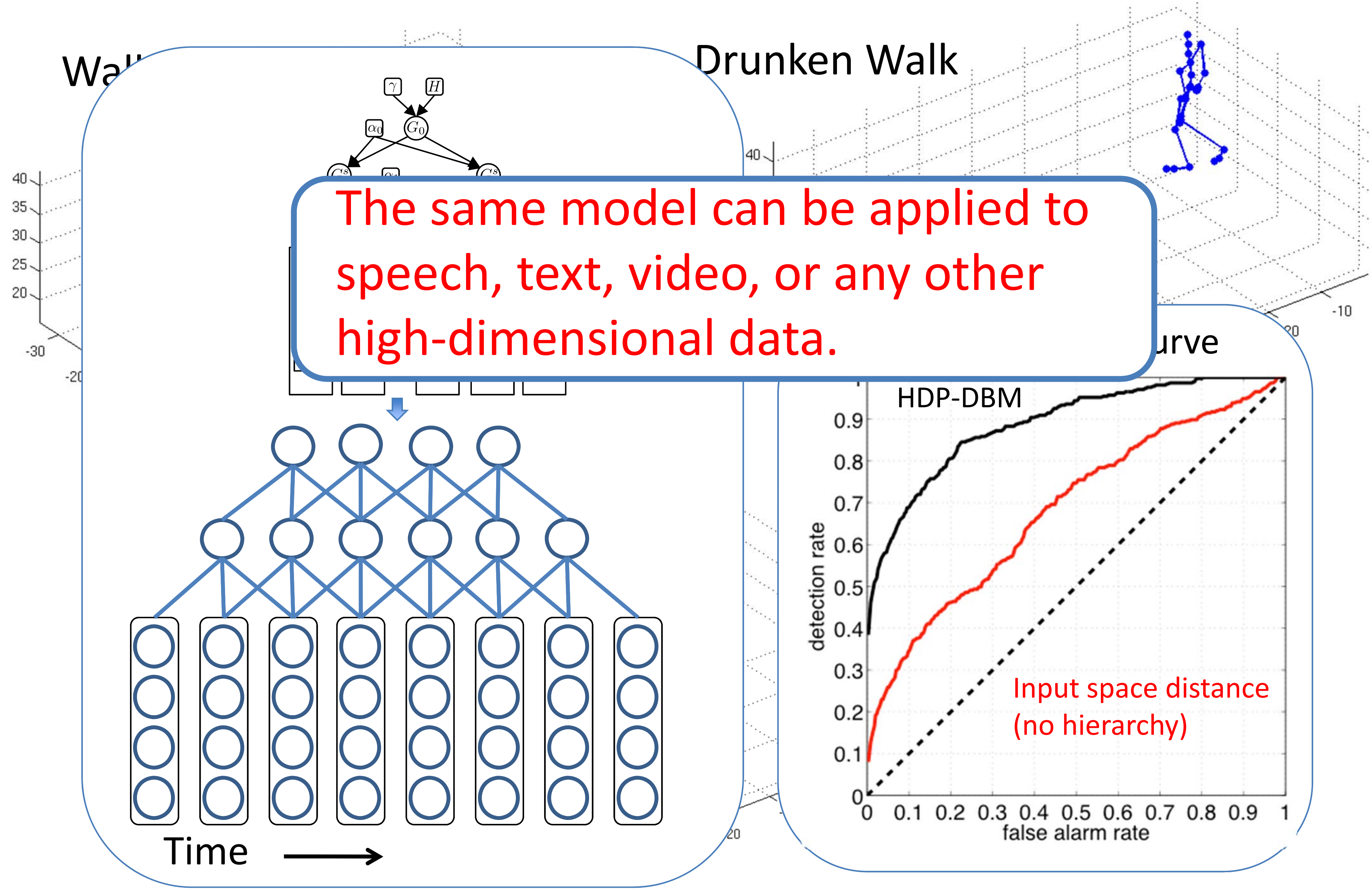
Drunken Walk



Sexy Walk



Motion Capture



The same model can be applied to speech, text, video, or any other high-dimensional data.

Talk Roadmap

Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.
- Deep Belief Networks: Learning Part-based Hierarchies.

Part 2: Advanced Deep Models.

- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning

Data – Collection of Modalities

- Multimedia content on the web - image + text + audio.

- Product recommendation systems.

- Robotics application

M

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

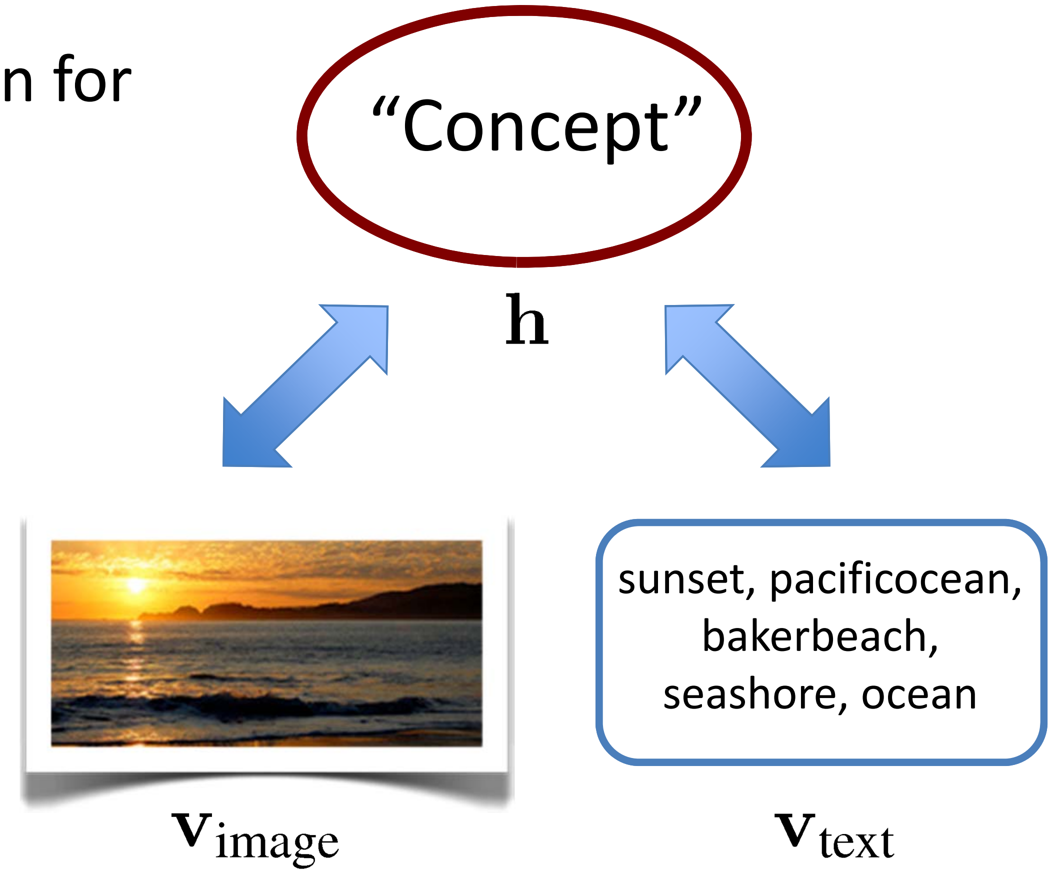
→

Building a Probabilistic Model

- Learn a joint density model: $P(\mathbf{h}, \mathbf{v}_{\text{image}}, \mathbf{v}_{\text{text}})$.

- **h**: "fused" representation for classification, retrieval.

$$P(\mathbf{h}|\mathbf{v}_{\text{image}}, \mathbf{v}_{\text{text}})$$



Building a Probabilistic Model

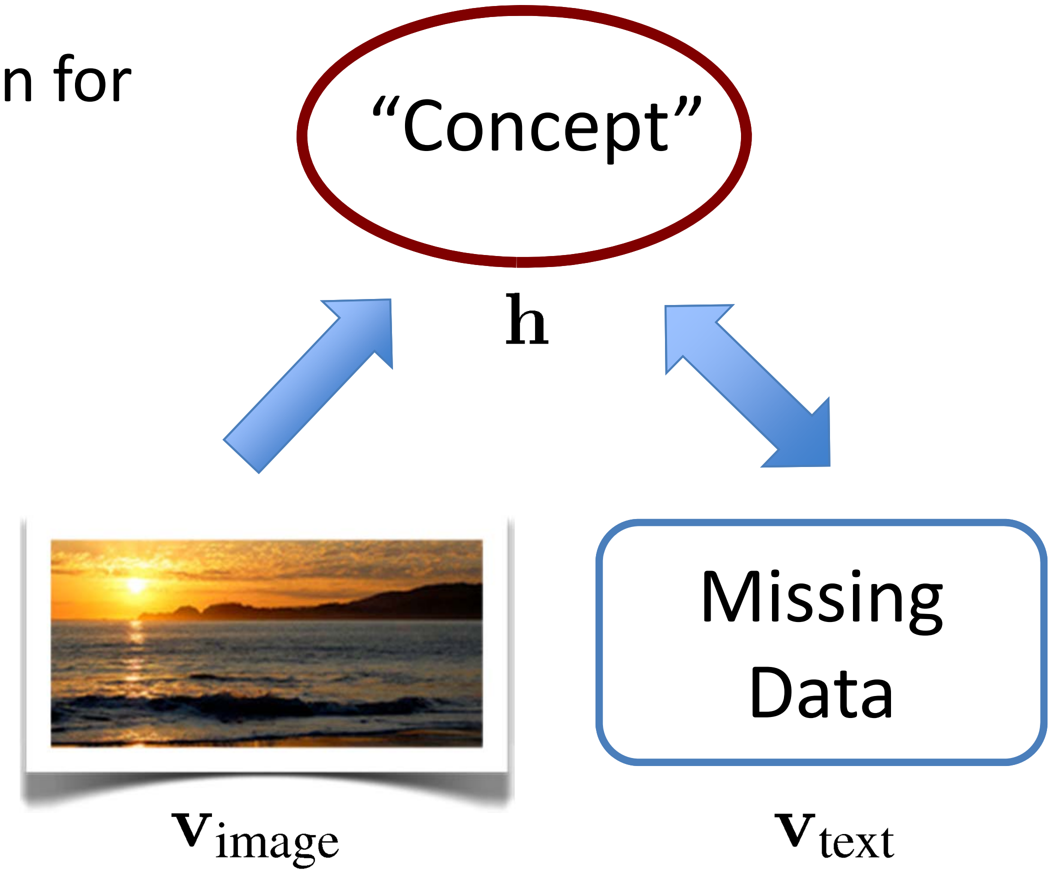
- Learn a joint density model: $P(\mathbf{h}, \mathbf{v}_{\text{image}}, \mathbf{v}_{\text{text}})$.

- **h**: "fused" representation for classification, retrieval.

- Generate data from conditional distributions for

  - Image Annotation

$$P(\mathbf{h}, \mathbf{v}_{\text{text}} | \mathbf{v}_{\text{image}})$$

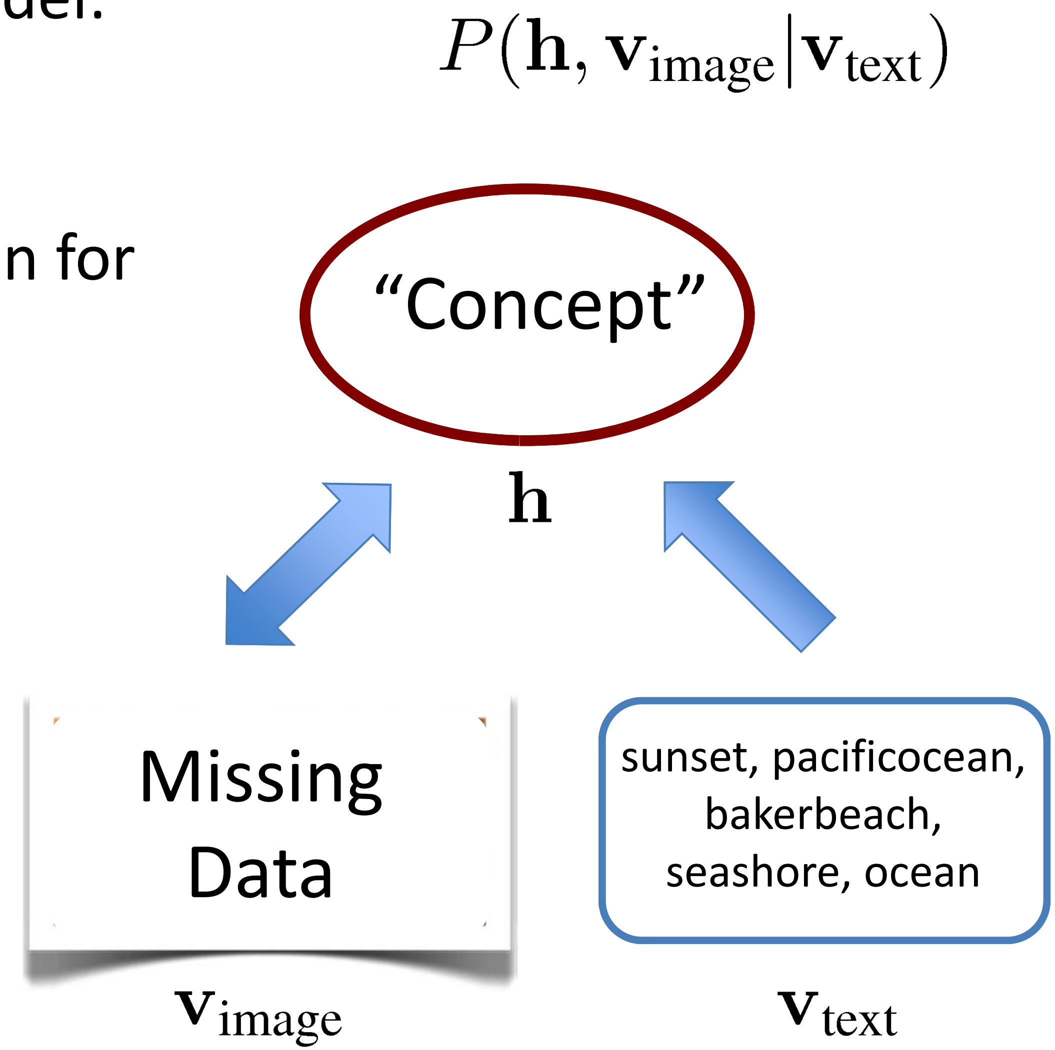


Building a Probabilistic Model

- Learn a joint density model: $P(\mathbf{h}, \mathbf{v}_{\text{image}}, \mathbf{v}_{\text{text}})$.

- **h**: "fused" representation for classification, retrieval.

- Generate data from conditional distributions for
  - Image Annotation
  - Image Retrieval



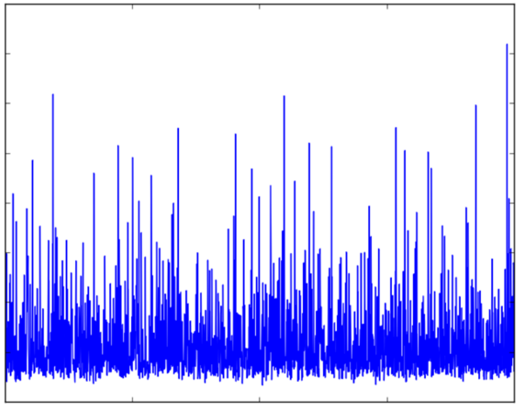
Challenges - I

Image

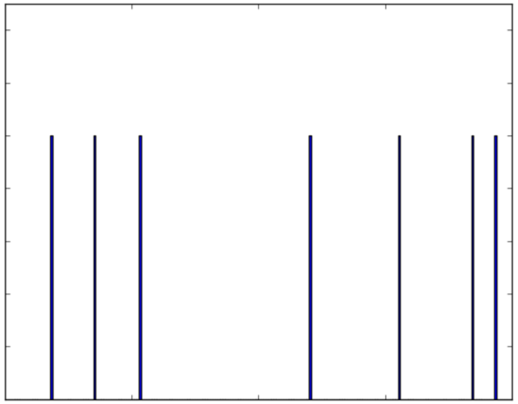
Text

sunset, pacificocean, bakerbeach, 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

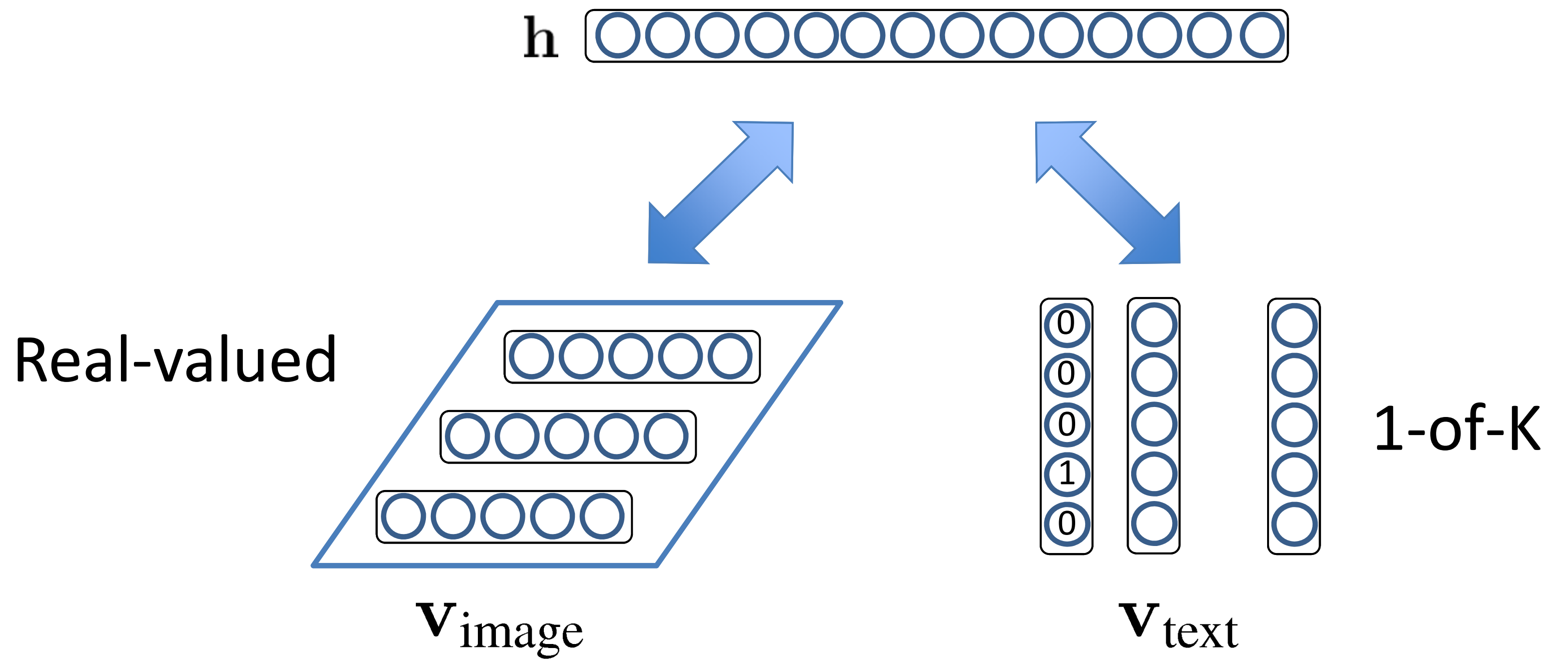
Noisy and missing data

Challenges - II

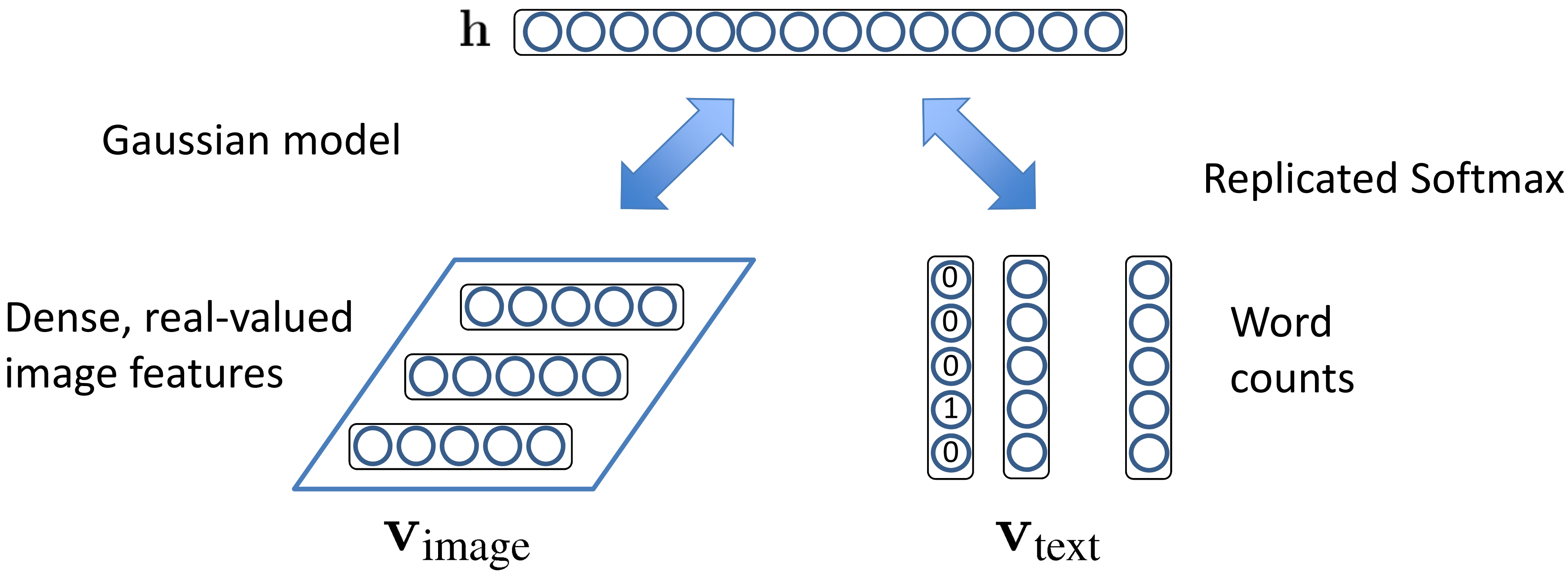
Image	Text	Text generated by the model
	pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves
	mickikrimmel, mickipedia, headshot	portrait, girl, woman, lady, blonde, pretty, gorgeous, expression, model
	< no text>	night, 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.

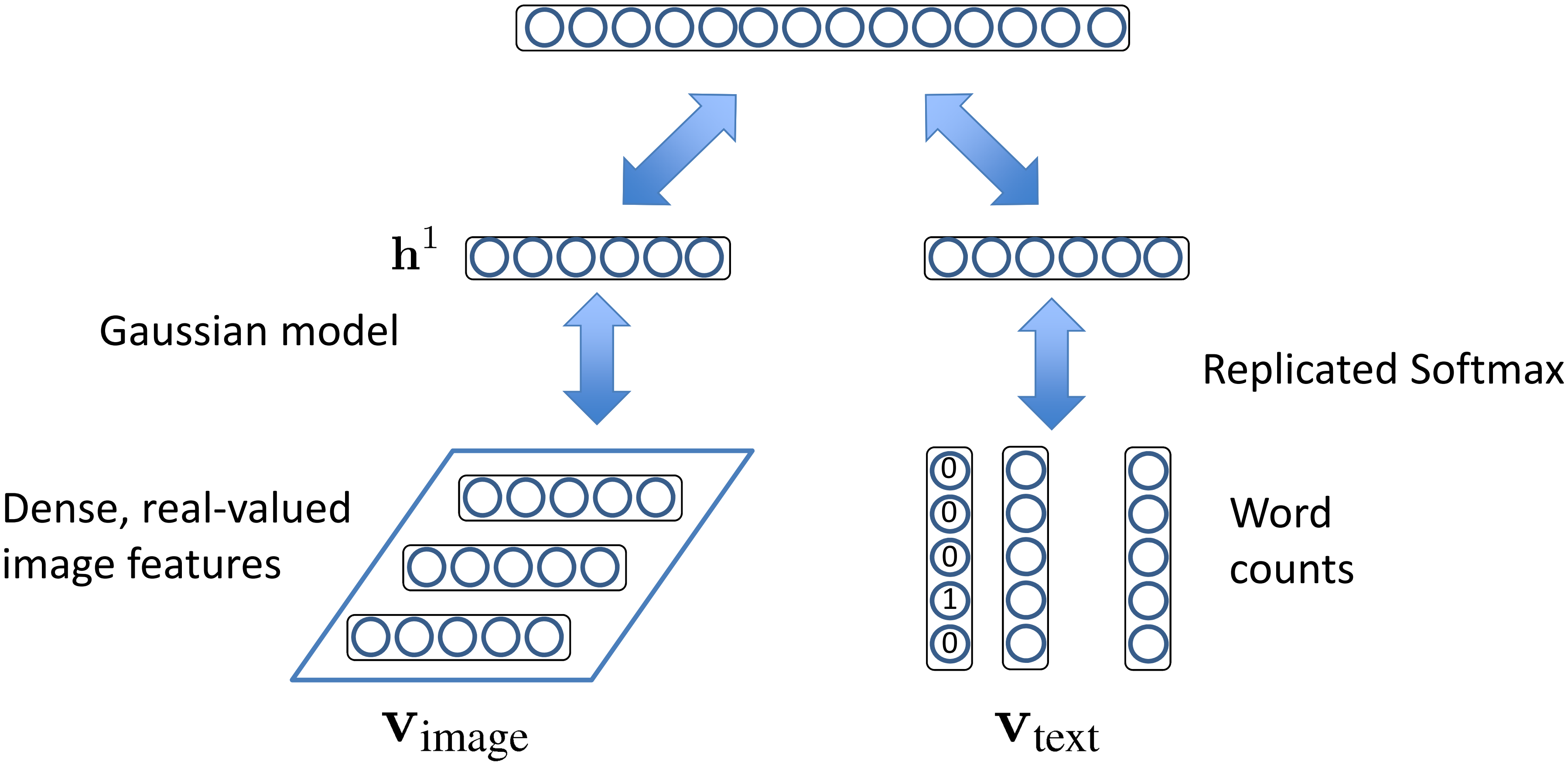


Multimodal DBM



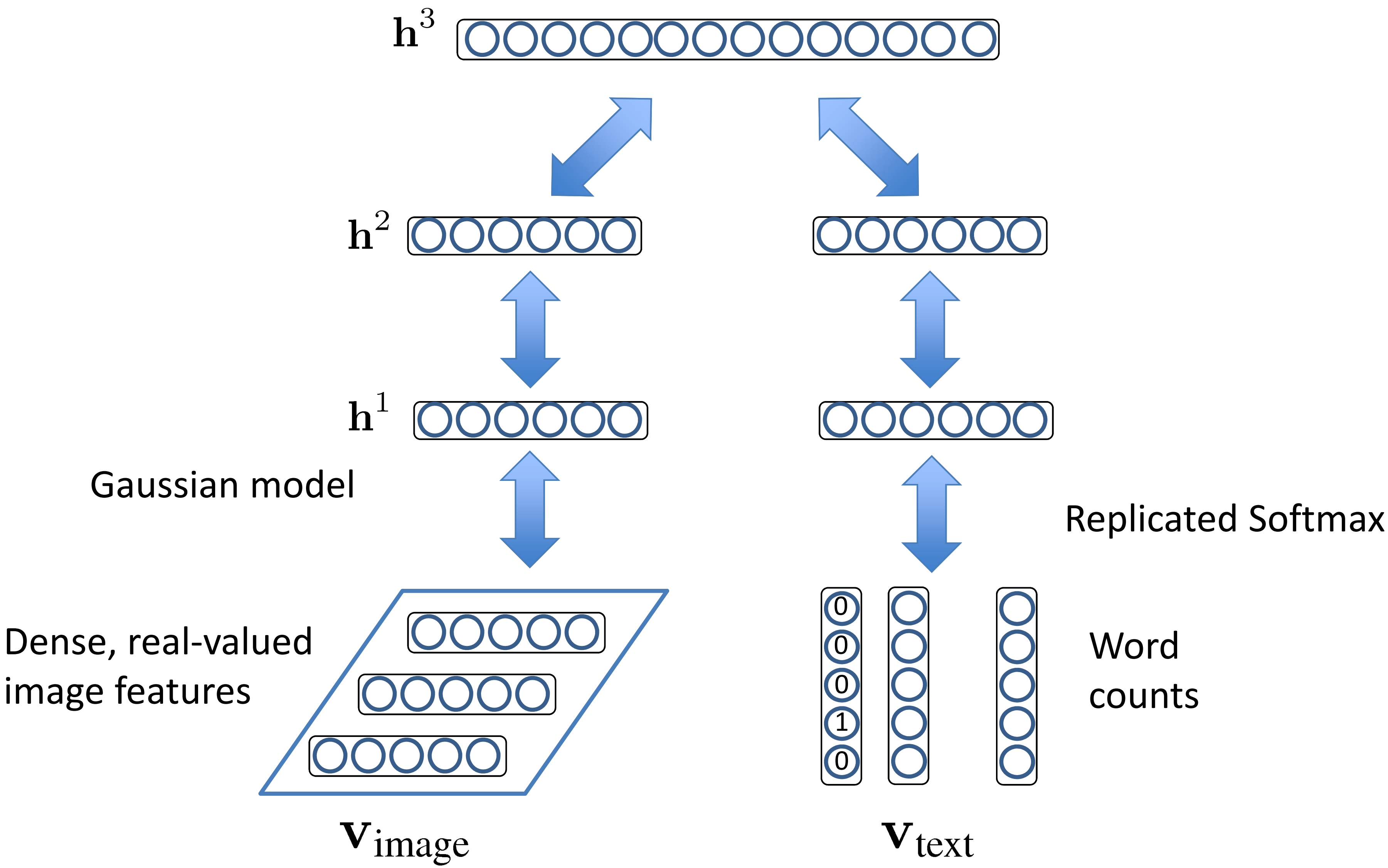
(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM



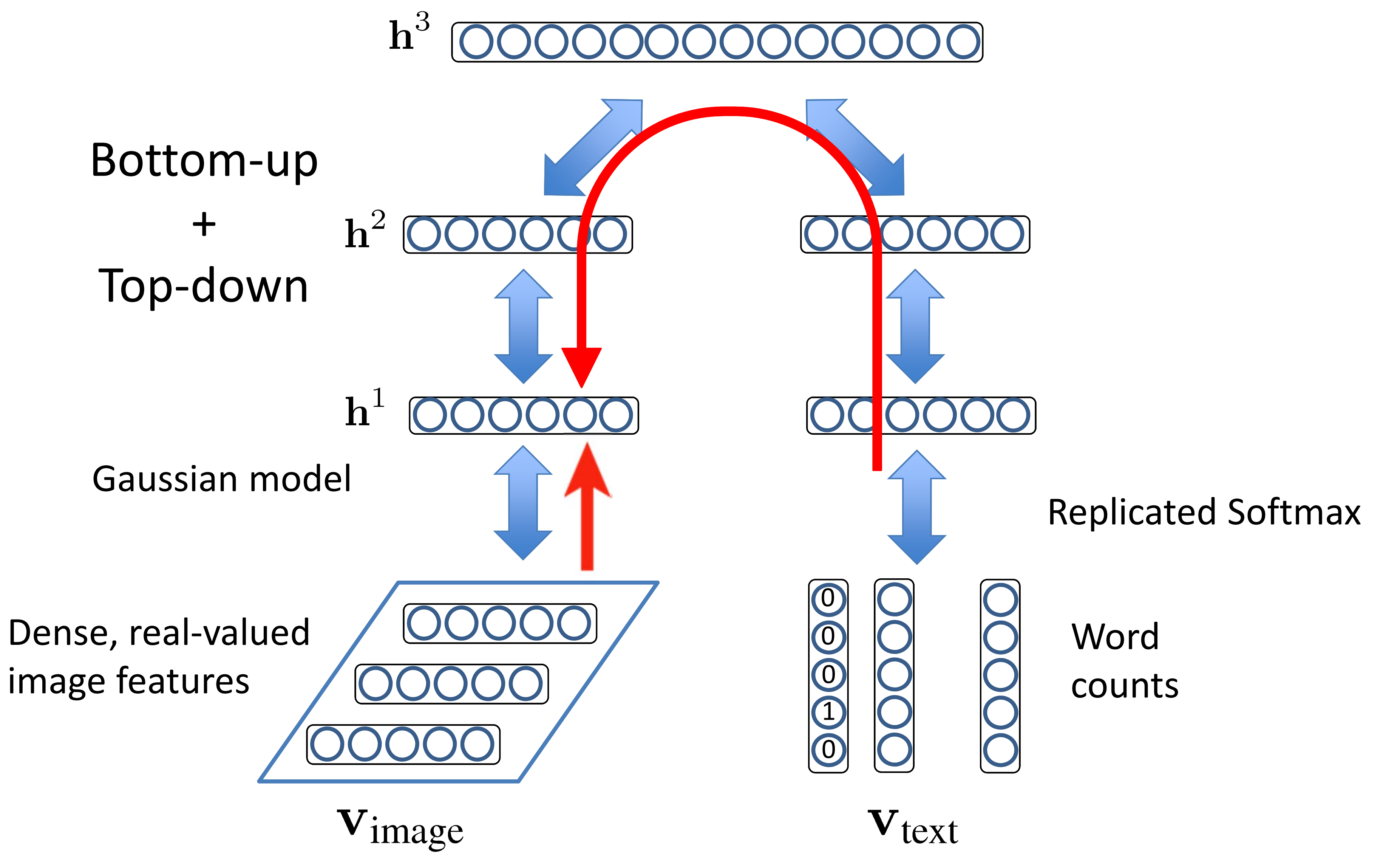
(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM



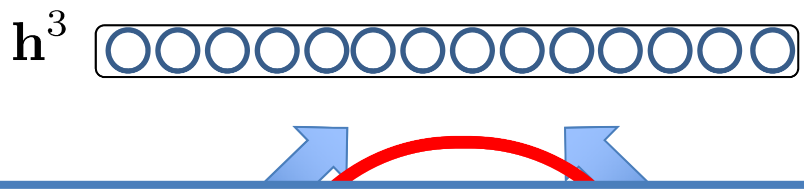
(Srivastava & Salakhutdinov, NIPS 2012)

Multimodal DBM



(Srivastava & Salakhutdinov, NIPS 2012)

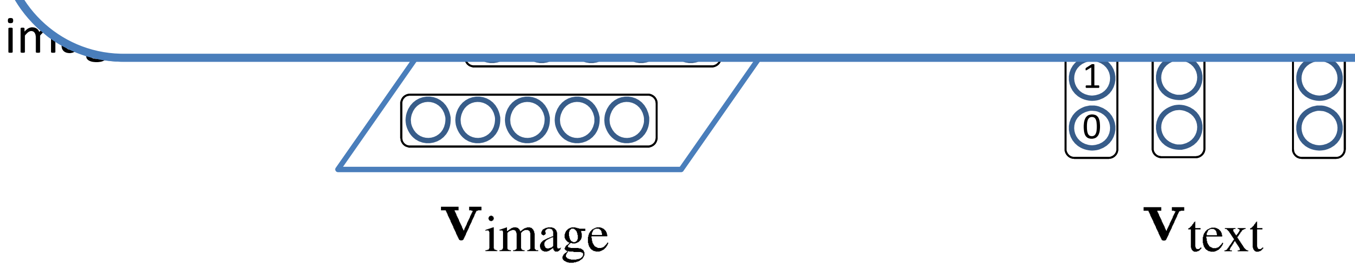
Multimodal DBM



$$P(\mathbf{v}^m, \mathbf{v}^t; \theta) = \sum_{\mathbf{h}^{(2m)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}} P(\mathbf{h}^{(2m)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}) \Big(\sum_{\mathbf{h}^{(1m)}} P(\mathbf{v}_m, \mathbf{h}^{(1m)} | \mathbf{h}^{(2m)}) \Big) \Big(\sum_{\mathbf{h}^{(1t)}} P(\mathbf{v}^t, \mathbf{h}^{(1t)} | \mathbf{h}^{(2t)}) \Big)$$

$$\frac{1}{\mathcal{Z}(\theta, M)} \sum_{\mathbf{h}} \exp \Big(\underbrace{- \sum_i \frac{(v_i^m)^2}{2\sigma_i^2} + \sum_{ij} \frac{v_i^m}{\sigma_i} W_{ij}^{(1m)} h_j^{(1m)} + \sum_{jl} W_{jl}^{(2m)} h_j^{(1m)} h_l^{(2m)}}_{\text{Gaussian Image Pathway}}$$

$$\underbrace{+ \sum_{jk} W_{kj}^{(1t)} h_j v_k^t + \sum_{jl} W_{jl}^{(2t)} h_j^{(1t)} h_l^{(2t)}}_{\text{Replicated Softmax Text Pathway}} \underbrace{+ \sum_{lp} W^{(3t)} h_l^{(2t)} h_p^{(3)} + \sum_{lp} W^{(3m)} h_l^{(2m)} h_p^{(3)}}_{\text{Joint } 3^{rd} \text{ Layer}} \Big)$$



(Srivastava & Salakhutdinov, NIPS 2012)

Text Generated from Images

Given	Generated
	dog, cat, pet, kitten, pupp 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

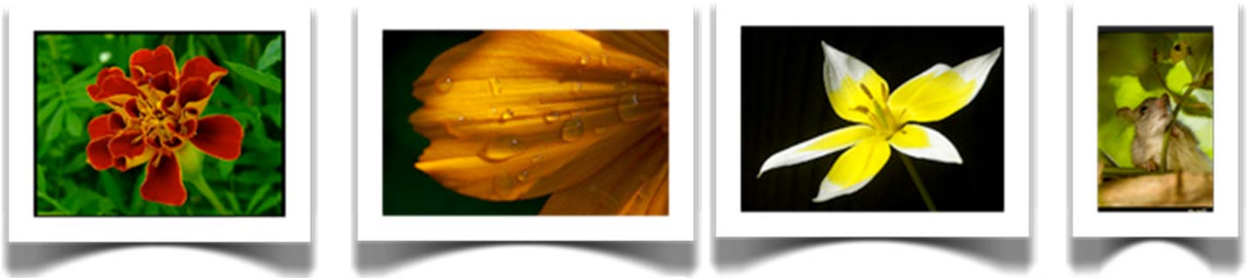
Given	Generated
	portrait, women, army, soldier, mother, postcard, soldiers
	obama, barackobama, election, politics, president, hope, change, sanfrancisco, convention, rally
	water, glass, beer, bottle, drink, wine, bubbles, splash, drops, drop

Images from Text

Given | Retrieved

water, red, sunset

nature, flower, red, green



blue, green, yellow, colors

chocolate, cake

MIR-Flickr Dataset

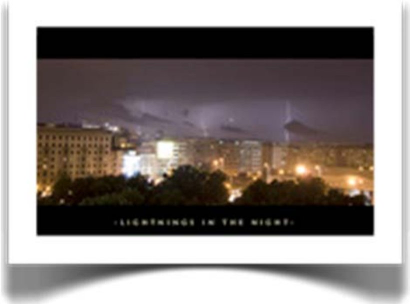
- 1 million images along with user-assigned tags.

sculpture, beauty, stone

d80

nikon, abigfave, goldstaraward, d80, nikond80

food, cupcake, vegan



anawesomeshot, theperfectphotographer, flash, damniwishidtakenthat, spiritofphotography

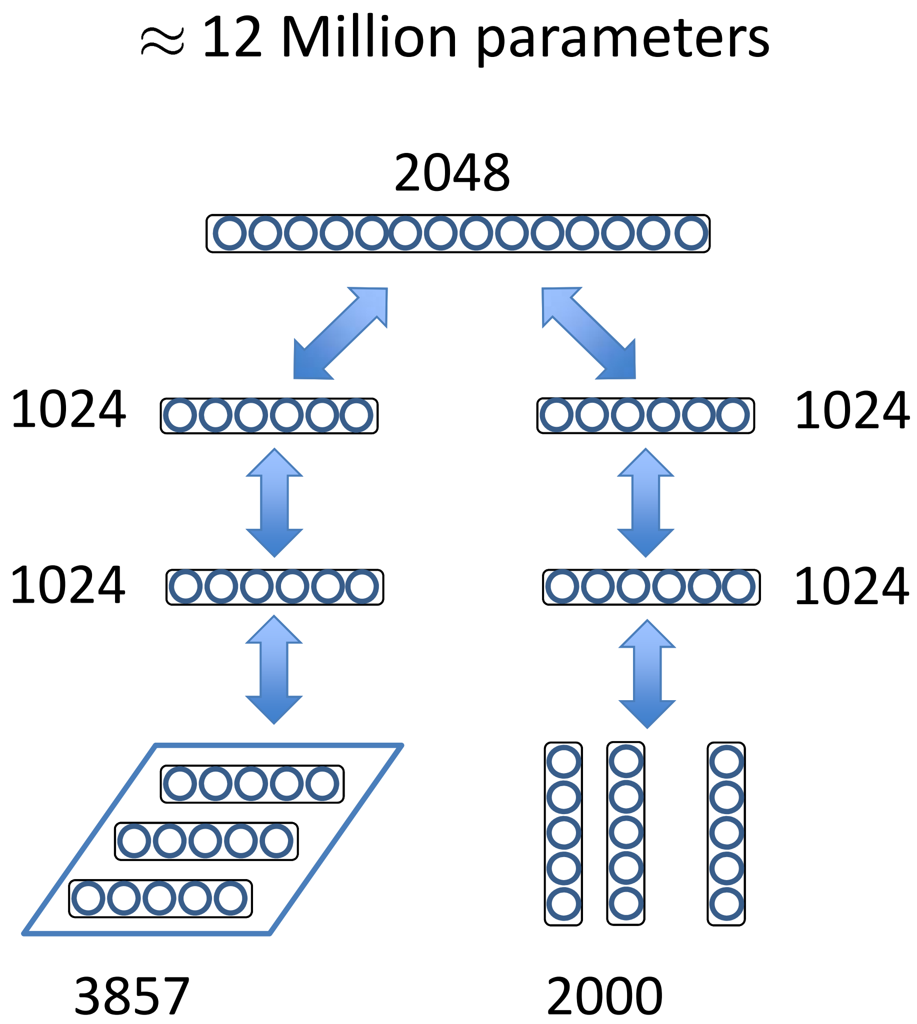
nikon, green, light, photoshop, apple, d70

white, yellow, abstract, lines, bus, graphic

sky, geotagged, reflection, cielo, bilbao, reflejo

Huiskes et. al.

Data and Architecture



- Image features: Gist, SIFT, MPEG-7 descriptors - 3857-dims.

- 200 most frequent tags.

- 25K labeled subset (15K training, 10K testing)

- 38 classes - *sky, tree, baby, car, cloud* …

Results

- 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

Similar Features, 15K labeled examples

Results

• 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
DBM-Unlablled+Dropout	**0.641**	0.888
MKL [Guillaumin et. al.]	0.623	

Similar Features, 15K labeled examples

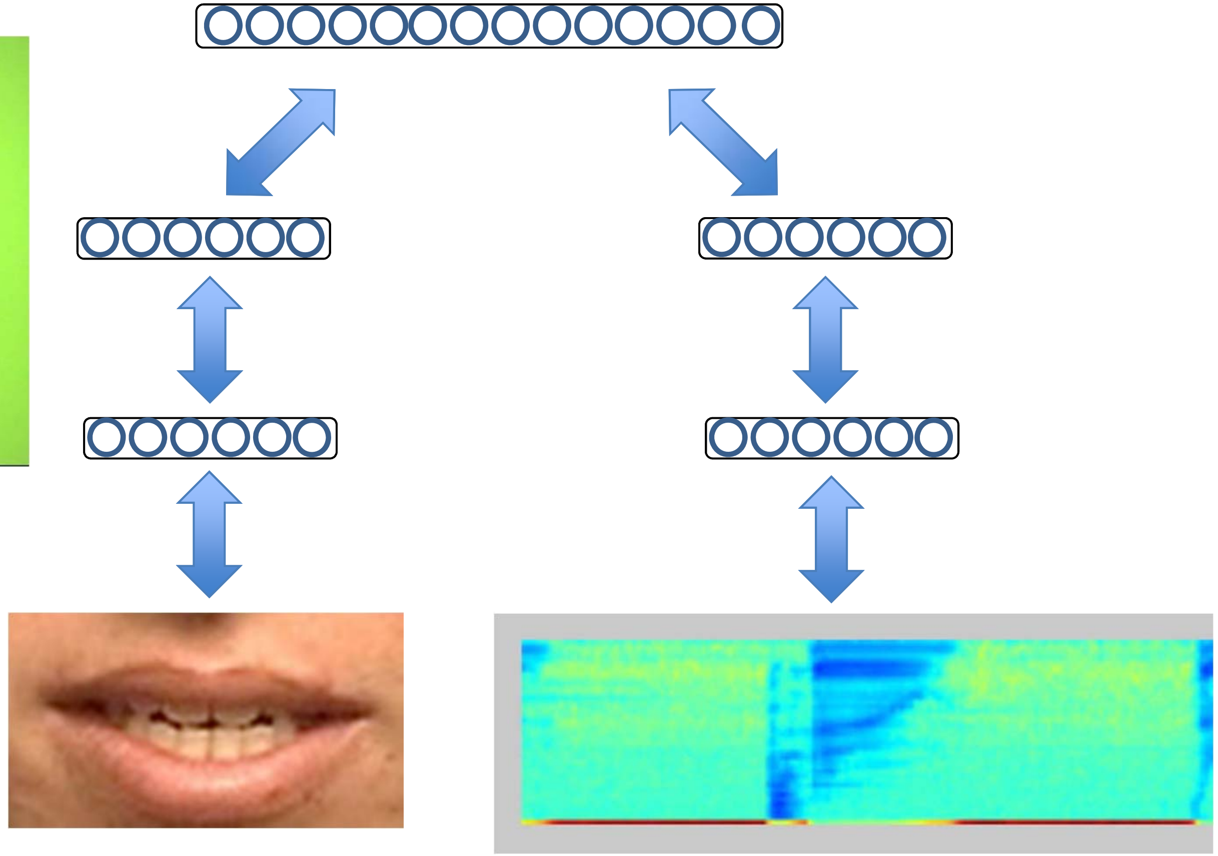
+ 1 Million unlabelled

State-of-the-art

• Multiple Kernel Learning uses 37,152 image features, compared to our model that uses 3,857 features.

Video and Audio

Cuave Dataset



Multi-Modal Models

Images

Text & Language

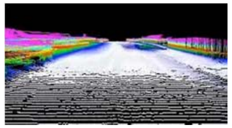
WIKIPEDIA
The Free Encyclopedia

REUTERS

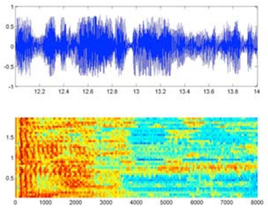
Associated Press

Video

Laser scans



Speech &
Audio



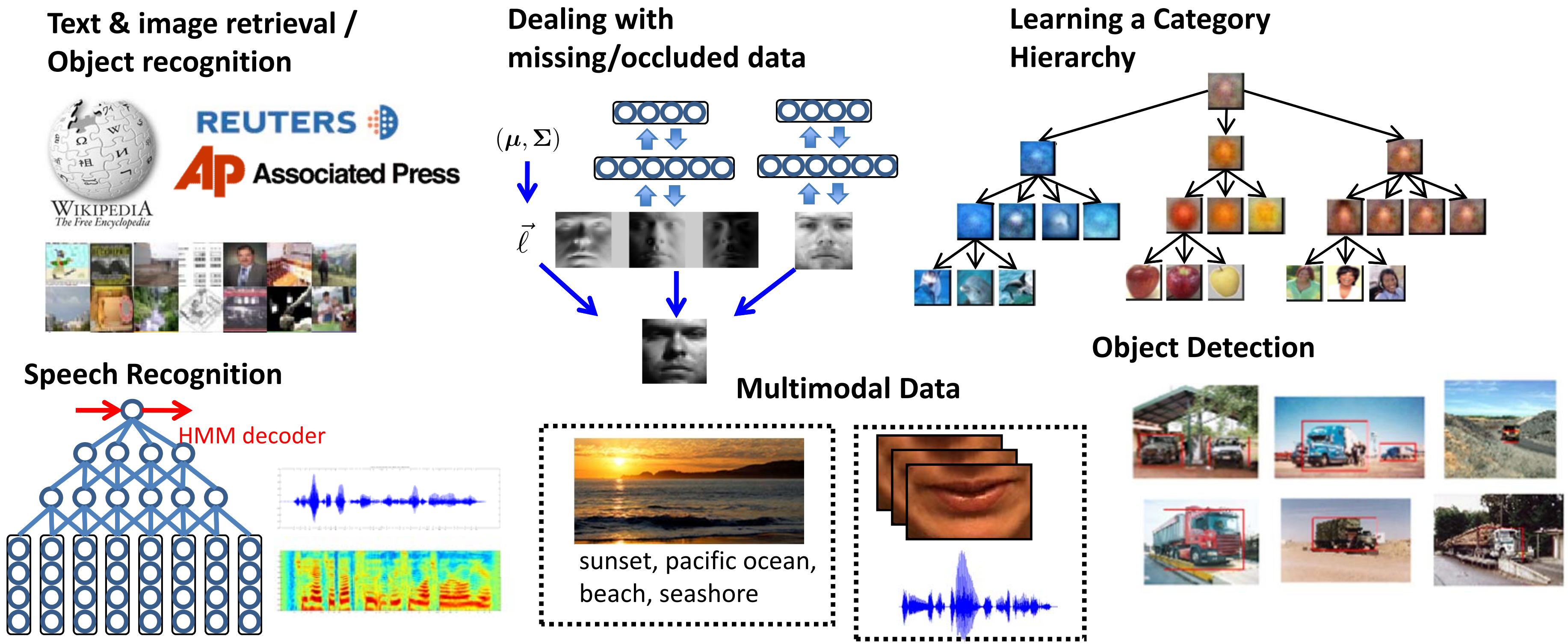
Time series
data

Develop learning systems that come
closer to displaying human like intelligence

One of Key Challenges:
Inference

Summary

- Efficient learning algorithms for Hierarchical Models. Learning more adaptive, robust, and structured representations.



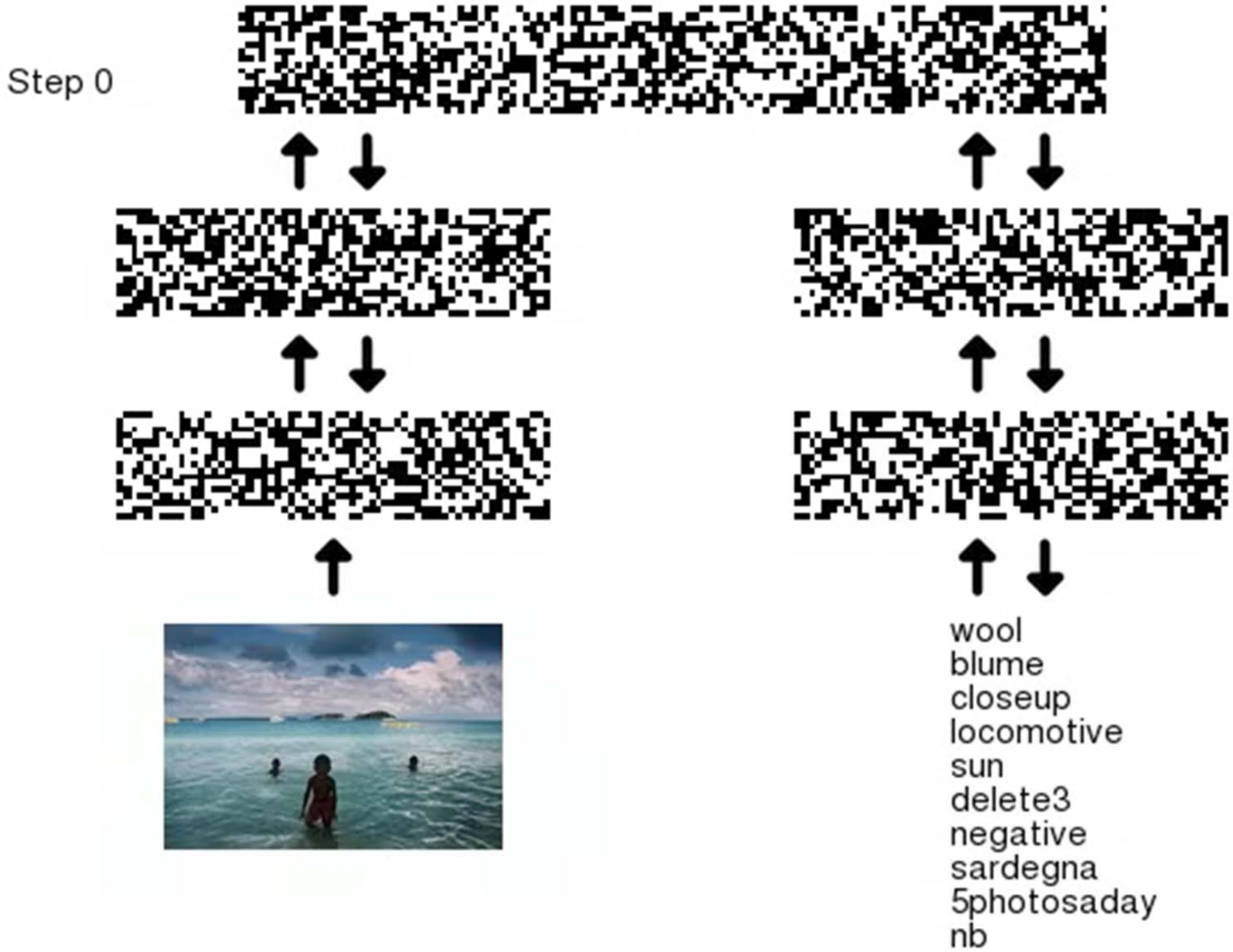
- Hierarchical models can improve current state-of-the art in many application domains:
  - Object recognition and detection, text and image retrieval, handwritten character and speech recognition, and others.

Thank you

Code for learning RBMs, DBNs, and DBMs is available at:
http://www.utstat.toronto.edu/~rsalakhu/

Demo: http://deeplearning.cs.toronto.edu/

Generating Text from Images



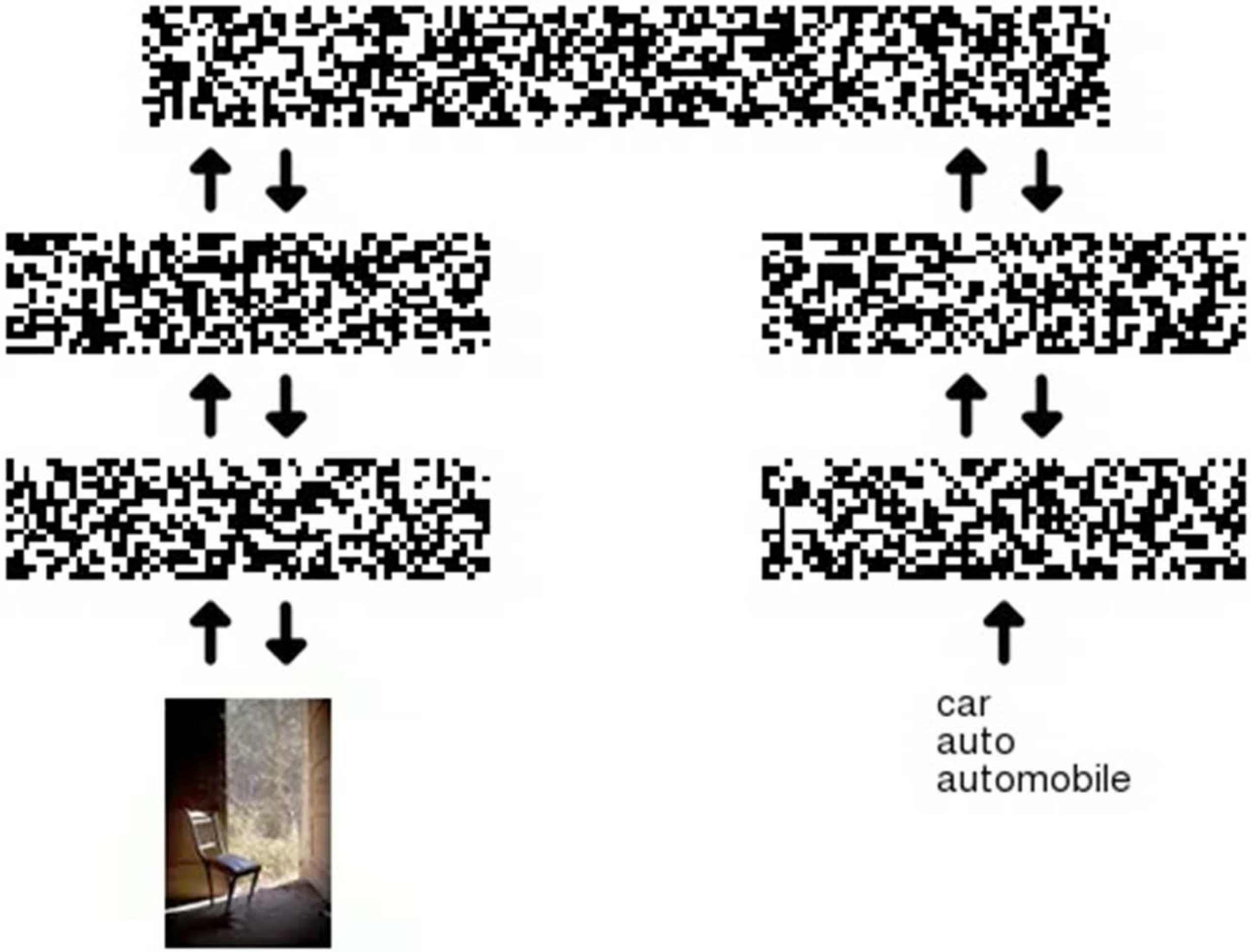
Samples drawn after every 50 steps of Gibbs updates

Sample at step 0
wool
blume
closeup
locomotive
sun
delete3
negative
sardegna
5photosaday
nb

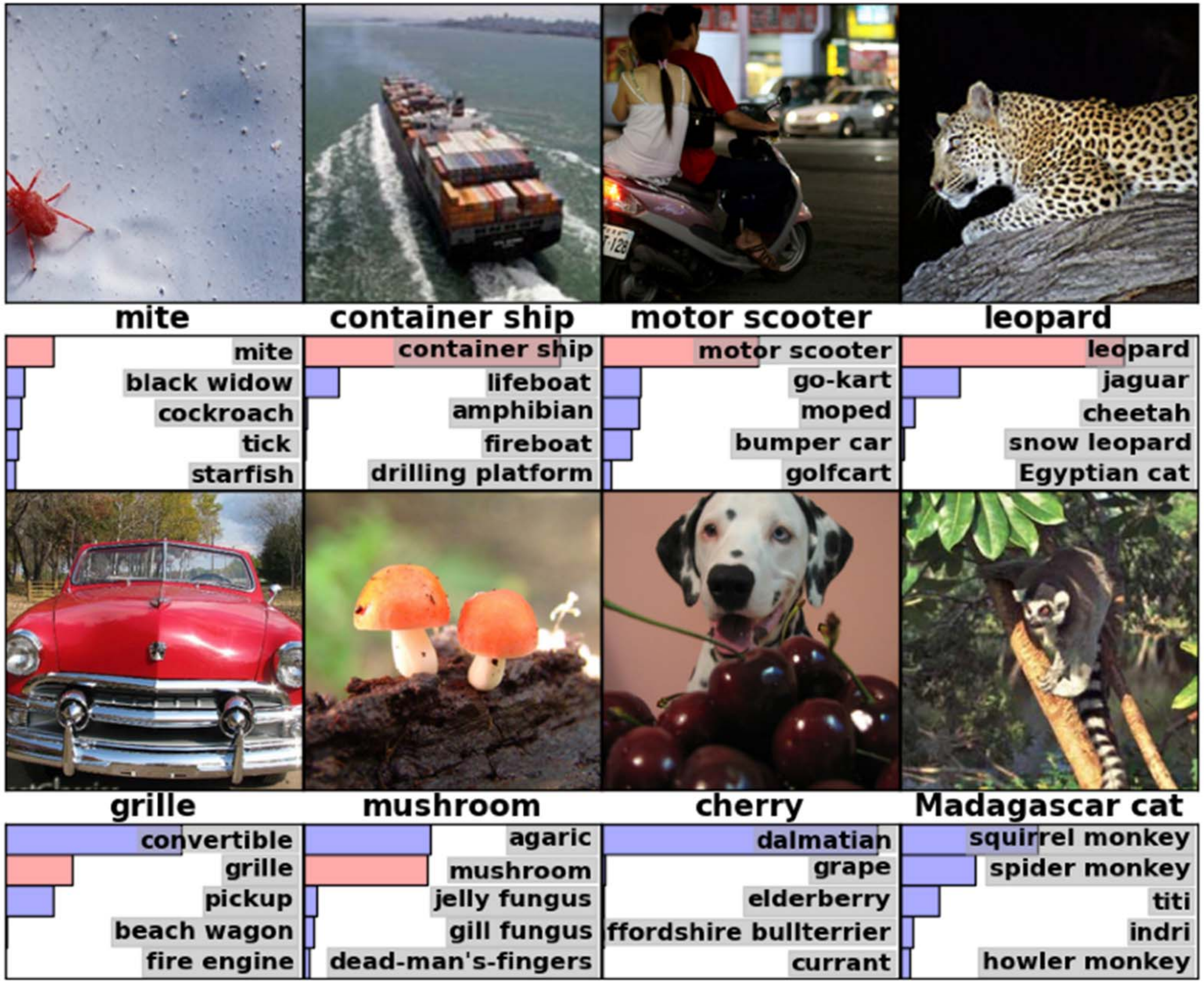
Images from Text

Step 0

Sample drawn after every 50 steps of Gibbs sampling



Convolutinal Deep Models for Image Recognition



(Krizhevsky et. al., NIPS 2012)