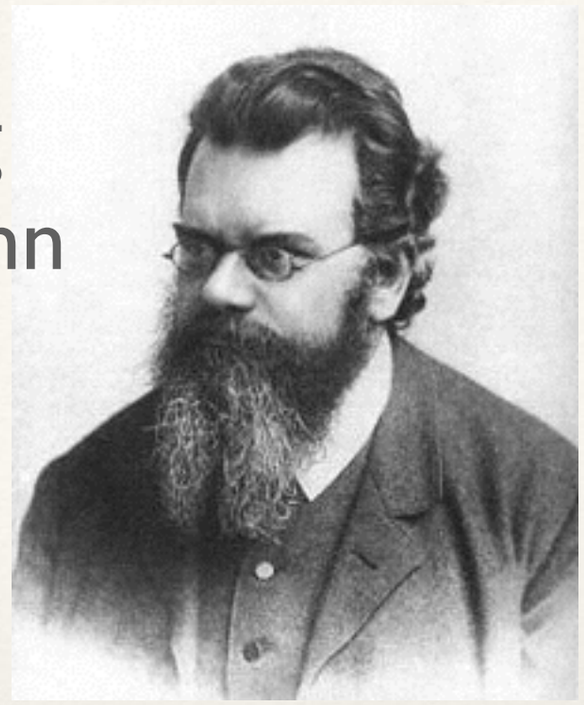
Hans Bethe

Ludwig Boltzmann



Fall 2015 Deep Learning CMPSCI 697L

Deep Learning Lecture 3: Graphical Models

Sridhar Mahadevan
Autonomous Learning Lab
UMass Amherst

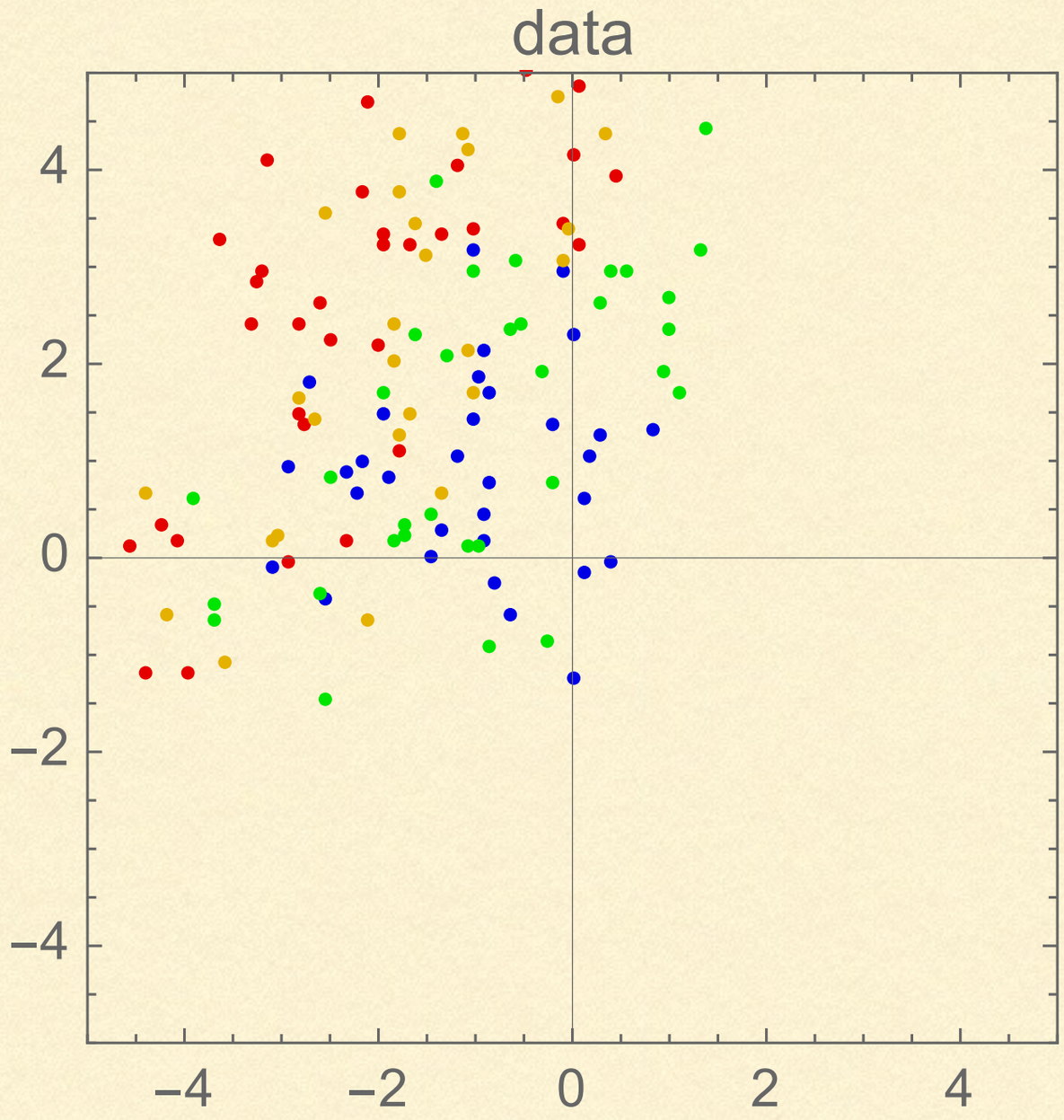
UMASS CS 50 YEARS
COLLEGE OF COMPUTER SCIENCE

RECAP FROM LAST LECTURE

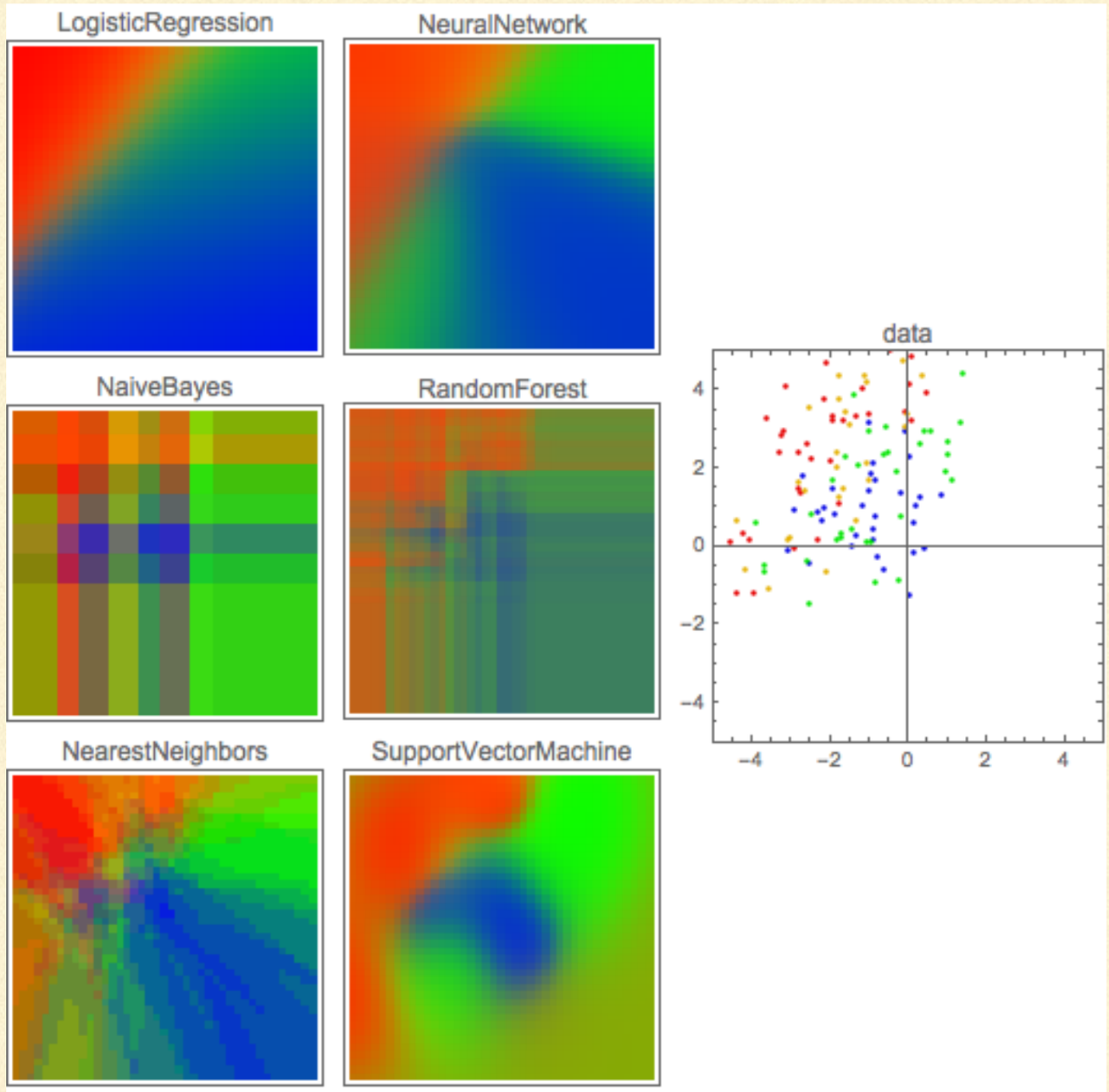
- Perceptron: linear unit with hard decision boundary
  - Restricted to learning linearly separable decision boundaries
- Logistic regression
  - Uses logistic function as a smooth nonlinear activation function
- Feedforward network
  - Can represent arbitrary continuous functions

SAMPLE DATASET

Colors indicate class labels



CLASSIFIER PERFORMANCE



PARALLEL DISTRIBUTED MODELS

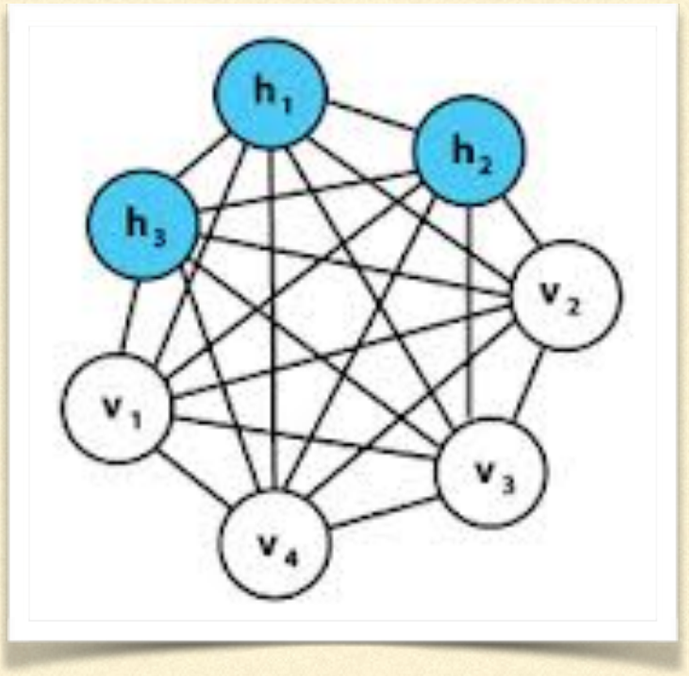
- Two areas of research in ML have explored PDP models
  - Neural nets
  - Graphical models
- In graphical models, we explore distributed representations of probability distributions
  - Undirected models: Markov random fields and variants
  - Directed models: Bayesian networks

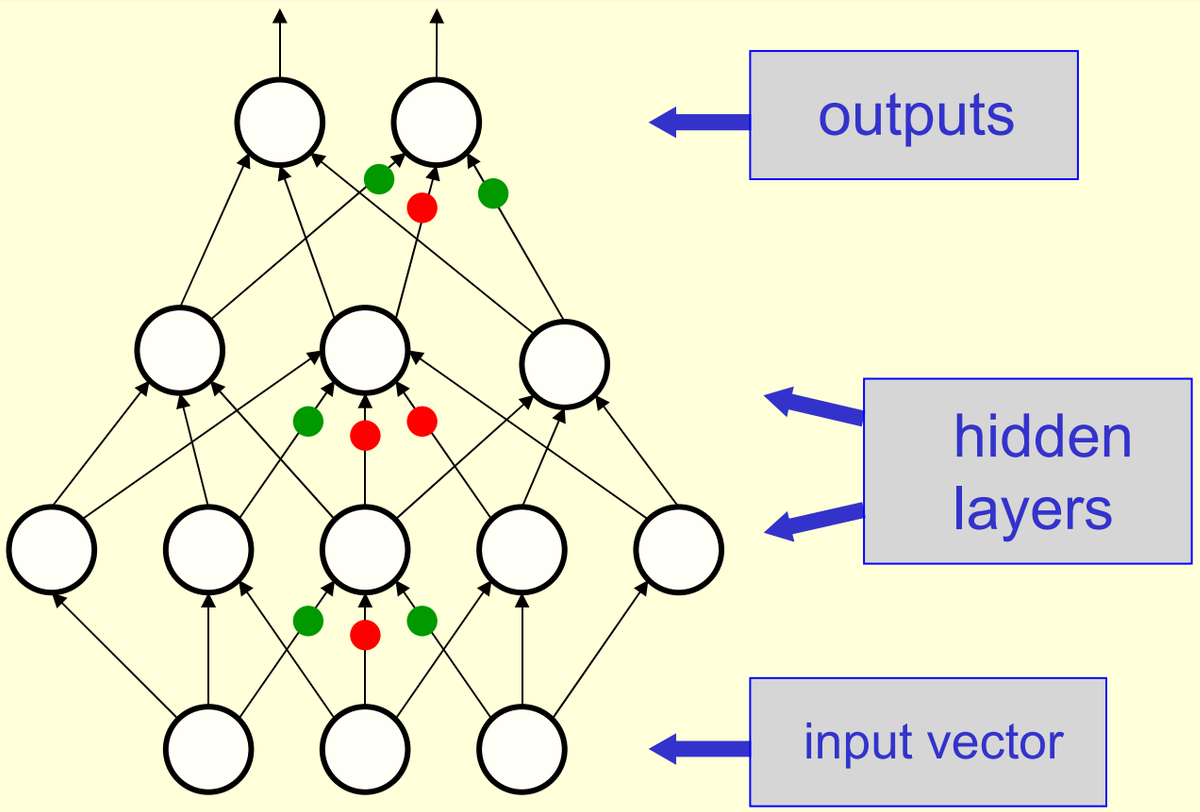
ENERGY AND PROBABILITY: PHYSICS AND MACHINE LEARNING

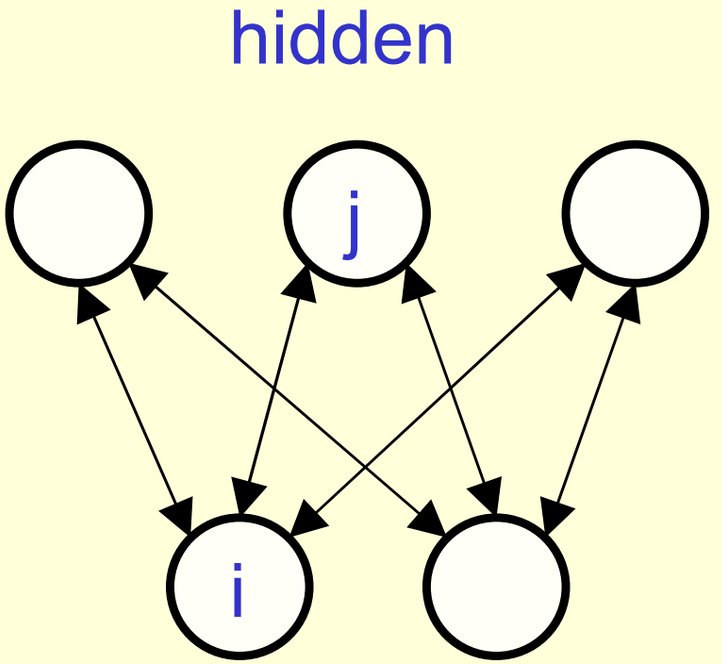
ENERGY AND PROBABILITY

$$P(x) = \frac{e^{-E(x)}}{Z}$$

$$Z = \sum_{x} e^{-E(x)}$$

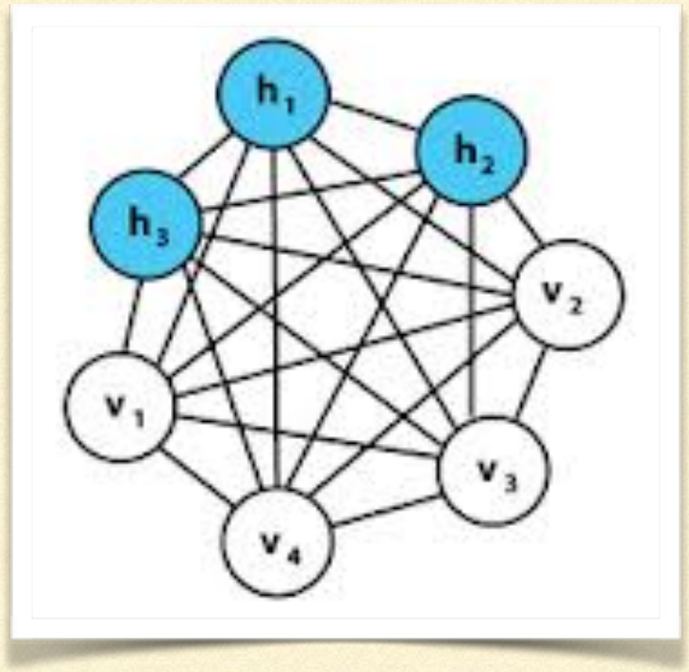

h1
h2
h3
v2
v1
v3
v4


outputs
hidden layers
input vector


hidden
j
i
visible

LATENT VARIABLE ENERGY MODELS

$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-E(x)}}{Z}$$



BE THE FREE ENERGY

$$F(x) = -\log \sum_h e^{-E(x,h)}$$

$$P(x) = \frac{e^{-F(x)}}{Z}$$

$$Z = \sum_x e^{-F(x)}$$

Hans Bethe

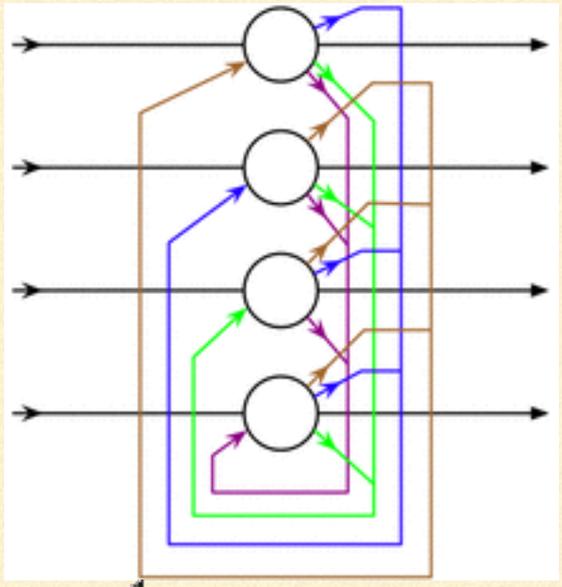
LOG LIKELIHOOD TRAINING

$$\frac{\partial \log P(x)}{\partial \theta} = \frac{\partial F(x)}{\partial \theta} - \sum_x P(x) \frac{\partial F(x)}{\partial \theta}$$

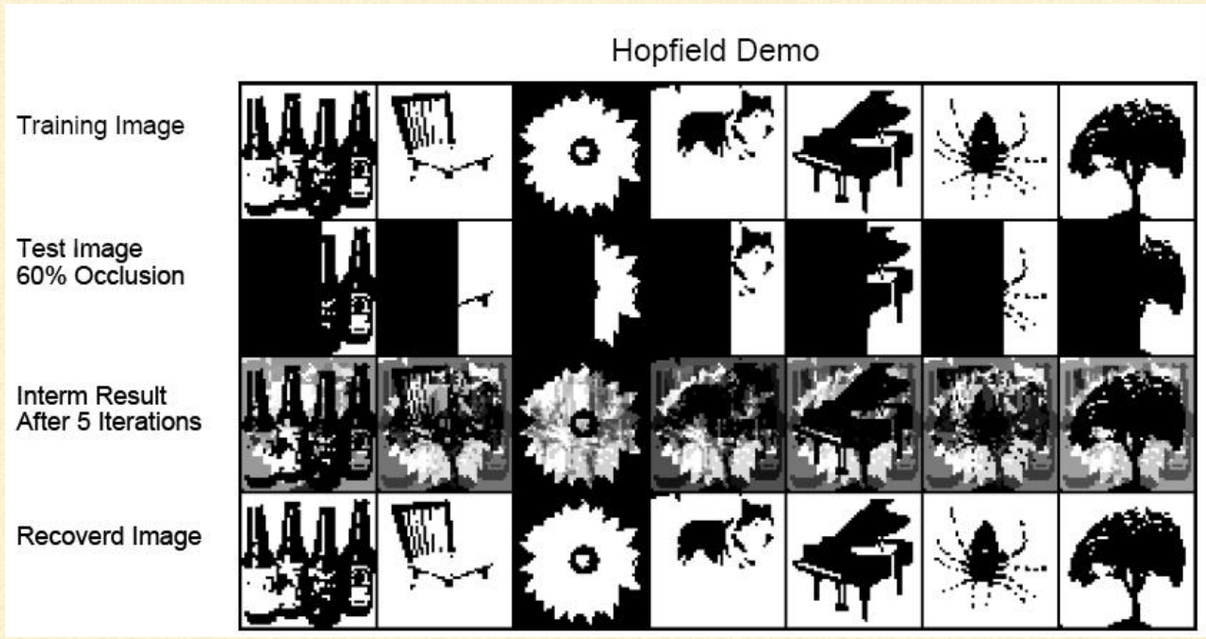
Increase likelihood of data, by minimizing free energy

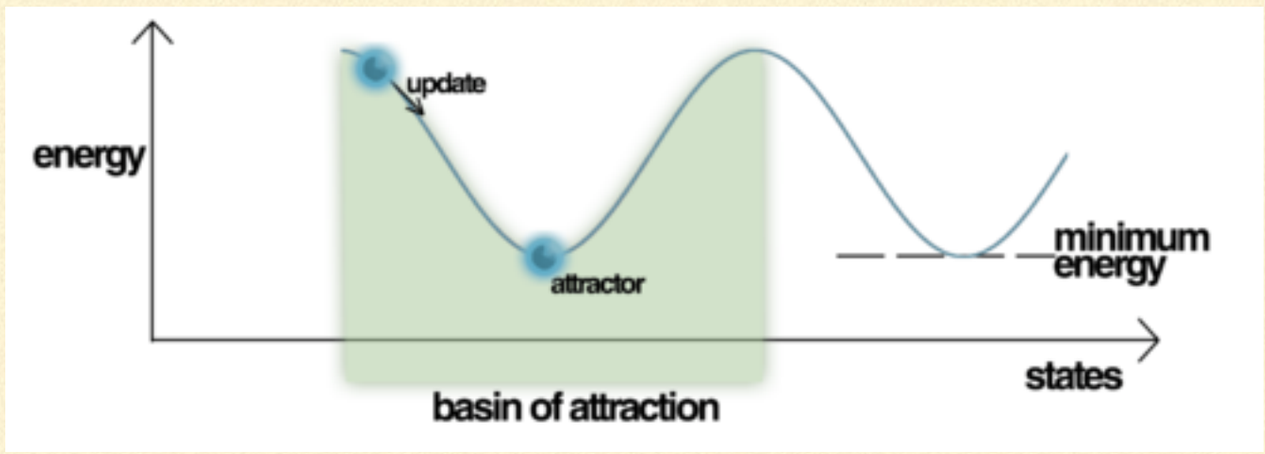
Decrease likelihood

HOPFIELD NETWORKS



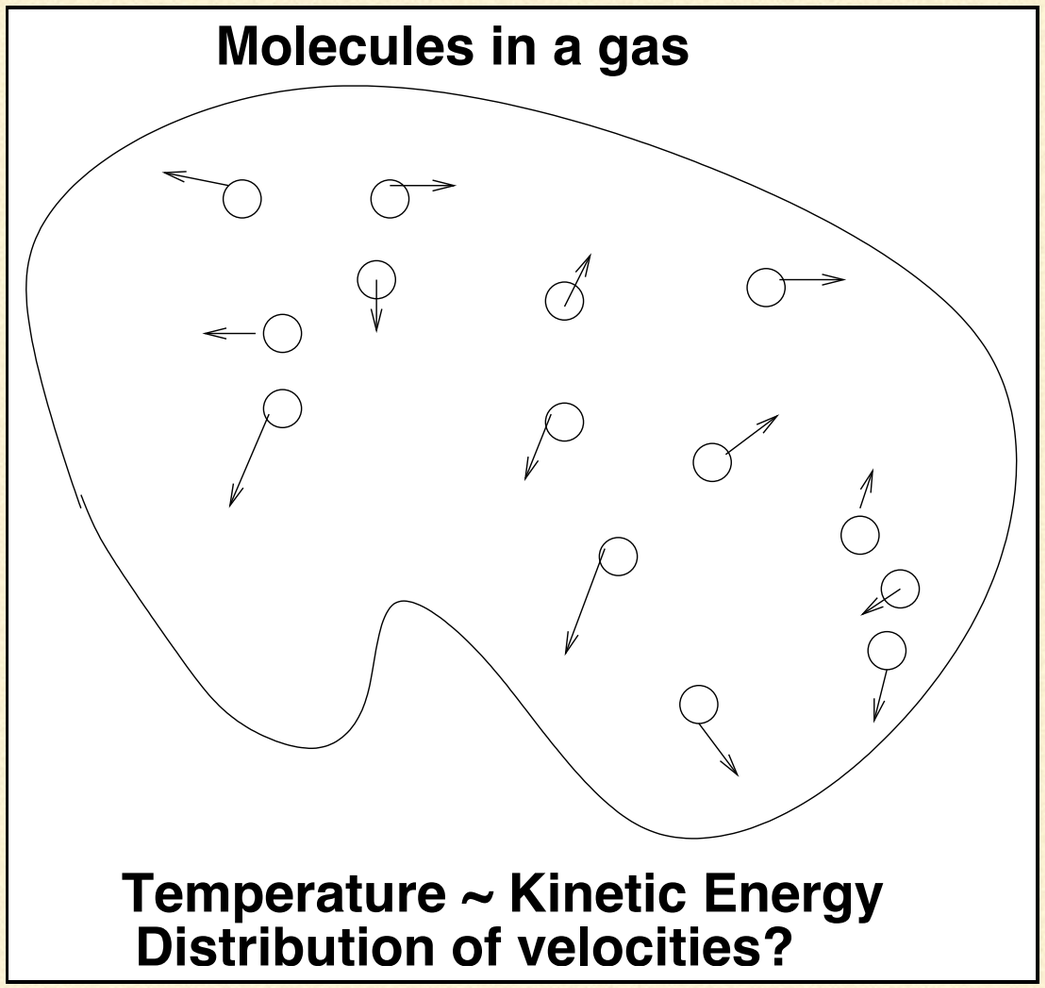
$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$





Content addressable memory

STATISTICAL PHYSICS



STATISTICAL PHYSICS AND ML

- The concept of entropy was investigated originally in statistical physics and thermodynamics.
- The temperature of a gas is proportional to the average kinetic energy of the molecules in the gas.
- The *distribution* of velocities at a given temperature is a maximum entropy distribution (also known as the Maxwell-Boltzmann distribution).
- *Boltzmann machines* are a type of neural network that get their inspiration from statistical physics.


S = k. log W
LVDWIG
BOLTZMANN
1844 – 1906
HENRIETTE
BOLTZMANN
GEB. EDLE von AIGENTLER
1854 – 1938
DR PHIL. PAULA
BOLTZMANN
GEB. CHIARI
1891 – 1977
ARTHUR
BOLTZMANN
DIPL. ING. DR PHIL. HOFRAT
1881 – 1932
LVDWIG
BOLTZMANN
1923 – 1943
ZTER MÄNNLICHER NACHKOMME.
GEFALLEN BEI SMOLENSK

Reprinted from AMERICAN JOURNAL OF PHYSICS, Vol. 33, No. 5, 391–398, May, 1965
Printed in U. S. A.

Gibbs vs Boltzmann Entropies*

E. T. JAYNES
Department of Physics, Washington University, St. Louis, Missouri

(Received 27 March 1964; in final form, 5 November 1964)

The status of the Gibbs and Boltzmann expressions for entropy has been a matter of some confusion in the literature. We show that: (1) the Gibbs H function yields the correct entropy as defined in phenomenological thermodynamics; (2) the Boltzmann H yields an "entropy" that is in error by a nonnegligible amount whenever interparticle forces affect thermodynamic properties; (3) Boltzmann's other interpretation of entropy, $S=k \log W$, is consistent with the Gibbs H, and derivable from it; (4) the Boltzmann H theorem does not constitute a demonstration of the second law for dilute gases; (5) the dynamical invariance of the Gibbs H gives a simple proof of the second law for arbitrary interparticle forces; (6) the second law is a special case of a general requirement for any macroscopic process to be experimentally reproducible. Finally, the "anthropomorphic" nature of entropy, on both the statistical and phenomenological levels, is stressed.

MAX ENTROPY PRINCIPLE

- Consider estimating a joint distribution over two variables u and v, given some *constraints*

P(u,v)	0	1	
0	?	?	
1	?	?	
	0.6		1.0

- The *maximum entropy* framework suggests picking values that make the least commitments, while being consistent with the contraints that are given.

MAX ENTROPY DISTRIBUTIONS

- Consider maximizing the entropy $h(P)$ over all distributions P satisfying the following constraints:
  - $P(x) \geq 0$ (where $x \in S$, the support of P).
  - $\sum_{x \in S} P(x) = 1$
  - $\sum_{x \in S} P(x) r_i(x) = \alpha_i$ for $1 \leq i \leq m$.

DERIVING THE MAXENT DISTRIBUTION

- Writing out the Lagrangian, we get

$$\Lambda(P) = -\sum_{x \in S} P(x) \ln P(x) + \lambda_0 (\sum_{x \in S} P(x)) + \sum_{i=1}^{m} \lambda_i (\sum_{x \in S} P(x) r_i(x) - \alpha_i)$$

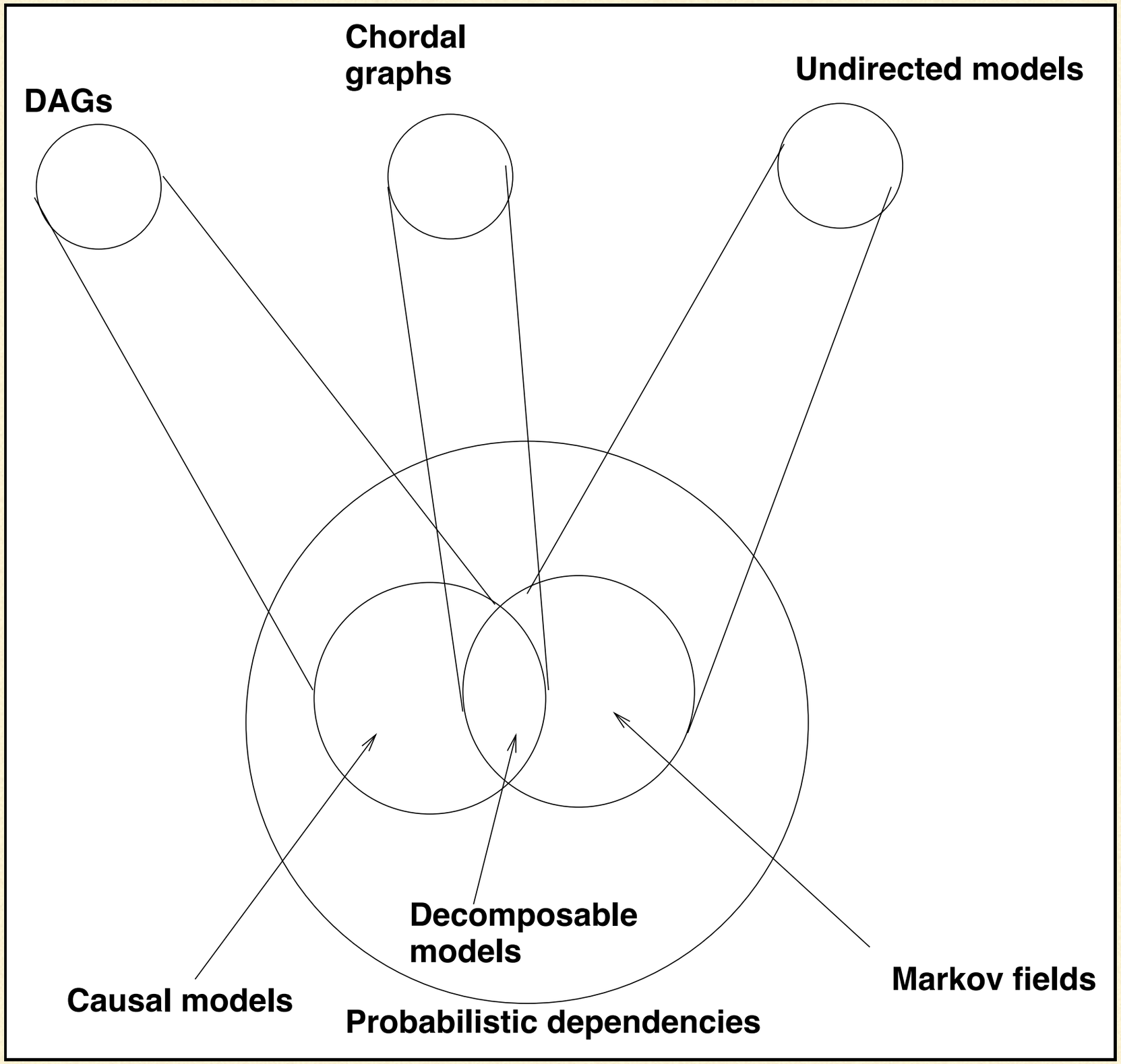
- Taking the gradient of this Lagrangian w.r.t. $P(x)$ we get

$$\begin{aligned}
&= \frac{\partial}{\partial P} \left(-\sum_{x \in S} P(x) \ln P(x) + \lambda_0 (\sum_{x \in S} P(x)) + \sum_{i=1}^{m} \lambda_i (\sum_{x \in S} P(x) r_i(x) - \alpha_i) \right) \\
&= \left(-\ln P(x) - 1 + \lambda_0 + \sum_{i=1}^{m} \lambda_i r_i(x) \right) \\
\Rightarrow \quad & P(x) = e^{\left(\lambda_0 - 1 + \sum_{i=1}^{m} \lambda_i r_i(x) \right)} \quad x \in S
\end{aligned}$$

EXAMPLE

- Find the maximum entropy distribution that satisfies the following constraints:
  - $S = [a, b]$
- Since there are no other constraints, the form of the distribution must be $P(x) = \int_a^b e^{\lambda_0} dx = 1$
- This is because all $\lambda_i = 0, i > 0$.
- Solving this integral, we find $[\int_a^b e^{\lambda_0} dx = 1 \Rightarrow e^{\lambda_0}(b - a) = 1$
- which immediately gives us the *uniform distribution* because $P(x) = e^{\lambda_0} = \frac{1}{b-a}$

REVIEW OF GRAPHICAL MODELS


DAGs
Chordal graphs
Undirected models
Causal models
Decomposable models
Markov fields
Probabilistic dependencies

DEPENDENCY MODEL

- Conditional independence:

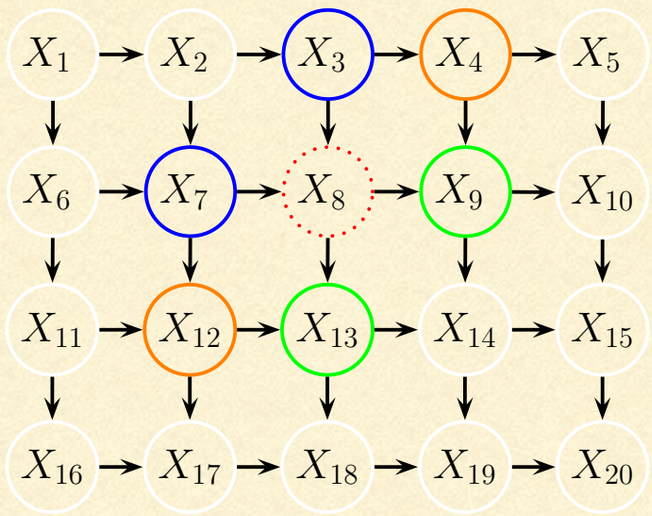
$$P(X|Y,Z) = P(X|Z) \textit{ denoted as } I(X,Z,Y)$$

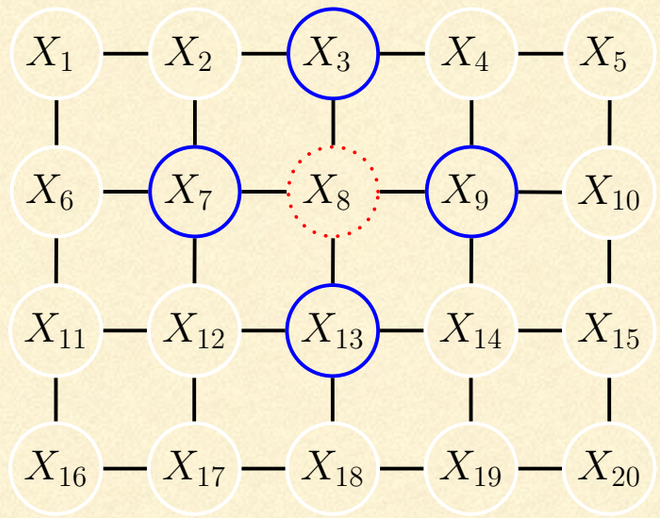
- The *graphoid* axioms [Pearl and Paz, 1985]
  - Symmetry: $I(X,Z,Y) \Leftrightarrow I(Y,Z,X)$
  - Decomposition:
    $I(X,Z,Y \cup W) \Rightarrow I(X,Z,Y) \wedge I(X,Z,W)$
  - Weak Union: $I(X,Z,Y \cup W) \Rightarrow I(X,Z \cup Y,W)$
  - Contraction:
    $I(X,Z,Y) \wedge I(X,Z,W \cup Y) \Rightarrow I(X,Z,W)$

UNDIRECTED GRAPHICAL MODEL

- Conditional independence in an undirected graph G reduces to graph separability.
- Graph separability is monotonic: if $< X|Z|Y >_G$, then $I(X|Z \cup A|Y >_G$ for any A. This is the *converse* of weak union.

MARKOV BLANKET IN DIRECTED VS UNDIRECTED MODELS





ENTROPY VS LIKELIHOOD

- Maximum entropy is the dual of maximum likelihood.
- ME involves finding the “maximally general” distribution that satisfies a set of prespecified constraints
- The notion of “maximal generality” is made precise using the concept of entropy
- The distribution that satisfies the ME constraint can be shown to be in an exponential form.
- The concept of maximum entropy comes from statistical physics, and this whole area of machine learning borrows heavily from physics.

DISTRIBUTIONS ON UNDIRECTED MODELS

- The factorization property of directed graphs ensures that every distribution defined in this way must satisfy all d-separation properties in the graph.
- In undirected models, a similar property holds: a set of nodes A is conditionally independent of a set of nodes B given the separating set C, if every path from a node in A to a node in B goes through a node in C.
- We say that a probability distribution P is *globally Markov* with respect to an undirected graph G iff for every disjoint set of nodes A, B, and C, if $A \perp B|C$, then the distribution also satisfies the same property.

FACTORIZATION IN UNDIRECTED MODELS

- Define a clique C to be a maximal set of nodes such that each node is connected to every other node in the set.
- Define the distribution $P(S) = \frac{1}{Z} \prod_C \psi_C(S_C)$ where each ψ_C is an arbitrary potential function on clique C, and Z is a normalizer.
- Theorem: For any undirected graph G, any distribution which satisfies the factorization property will be globally Markov.
- **Hammersley Clifford Theorem**: For strictly positive distributions, the global Markov property is equivalent to the factorization property.

LOG LINEAR MODELS

$$\log \psi_c(\mathbf{y}_c) \triangleq \boldsymbol{\phi}_c(\mathbf{y}_c)^T \boldsymbol{\theta}_c$$

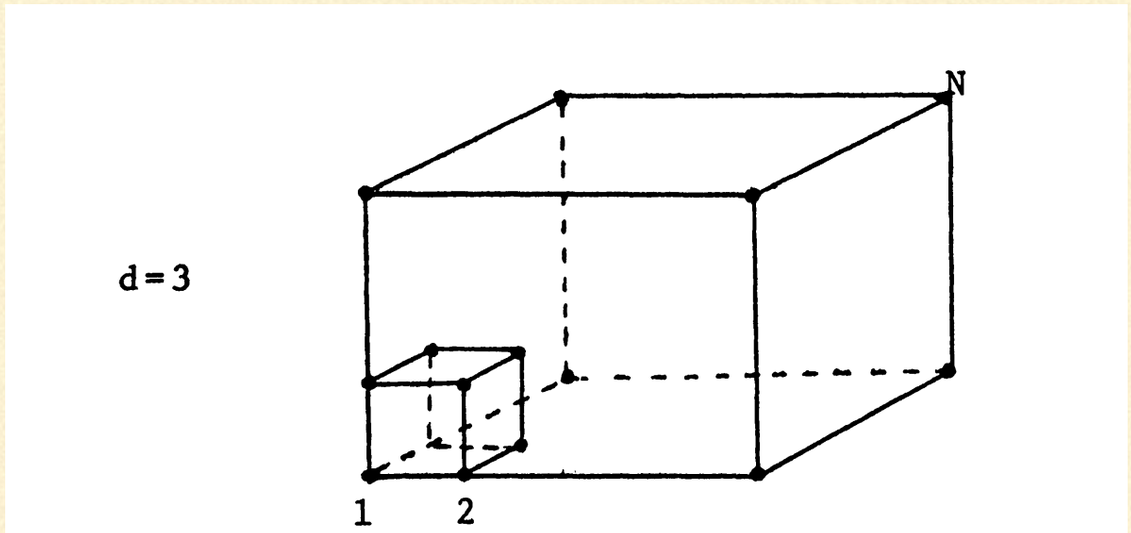
$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \sum_c \boldsymbol{\phi}_c(\mathbf{y}_c)^T \boldsymbol{\theta}_c - Z(\boldsymbol{\theta})$$

Pairwise MRF

$$\boldsymbol{\phi}_{st}(y_s, y_t) = [\ldots, \mathbb{I}(y_s = j, y_t = k), \ldots]$$

$$\psi_{st}(y_s = j, y_t = k) = \exp([\boldsymbol{\theta}_{st}^T \boldsymbol{\phi}_{st}]_{jk}) = \exp(\theta_{st}(j,k))$$

ISING MODELS IN PHYSICS



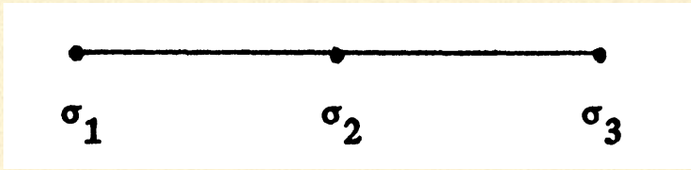
"Energy" models using the Hamiltonian

$$H = H(\sigma) = -\sum_{\langle i,j \rangle} E\sigma_i\sigma_j - \sum_i J\sigma_i,$$

partition function:

$$Z = Z(\beta, E, J, N) = \sum_{\pm 1} e^{-\beta H(\sigma)}.$$

EXAMPLE: ONE-DIMENSIONAL ISING MODEL



$$H = -E(\sigma_1\sigma_2 + \sigma_2\sigma_3) - J(\sigma_1 + \sigma_2 + \sigma_3).$$

$$\text{Prob}(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z}.$$

$$Z = e^{-\beta H(1,1,1)} + e^{-\beta H(1,1,-1)} + e^{-\beta H(1,-1,1)} + e^{-\beta H(1,-1,-1)}$$

TRAINING UNDIRECTED MODELS

Method	Restriction	Exact MLE?
Closed form	Only Chordal MRF	Exact
IPF	Only Tabular / Gaussian MRF	Exact
Gradient-based optimization	Low tree width	Exact
Max-margin training	Only CRFs	N/A
Pseudo-likelihood	No hidden variables	Approximate
Stochastic ML	-	Exact (up to MC error)
Contrastive divergence	-	Approximate
Minimum probability flow	Can integrate out the hiddens	Approximate

EXACT GRADIENT METHODS

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}} \exp\left(\sum_c \boldsymbol{\theta}_c^T \boldsymbol{\phi}_c(\mathbf{y})\right)$$

$$\ell(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_i \log p(\mathbf{y}_i|\boldsymbol{\theta}) = \frac{1}{N}\sum_i \left[\sum_c \boldsymbol{\theta}_c^T \boldsymbol{\phi}_c(\mathbf{y}_i) - \log Z(\boldsymbol{\theta})\right]$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}_c} = \frac{1}{N}\sum_i \left[\boldsymbol{\phi}_c(\mathbf{y}_i) - \frac{\partial}{\partial \boldsymbol{\theta}_c}\log Z(\boldsymbol{\theta})\right]$$

$$\frac{\partial \log Z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_c} = \mathbb{E}\left[\boldsymbol{\phi}_c(\mathbf{y})|\boldsymbol{\theta}\right] = \sum_{\mathbf{y}} \boldsymbol{\phi}_c(\mathbf{y}) p(\mathbf{y}|\boldsymbol{\theta})$$

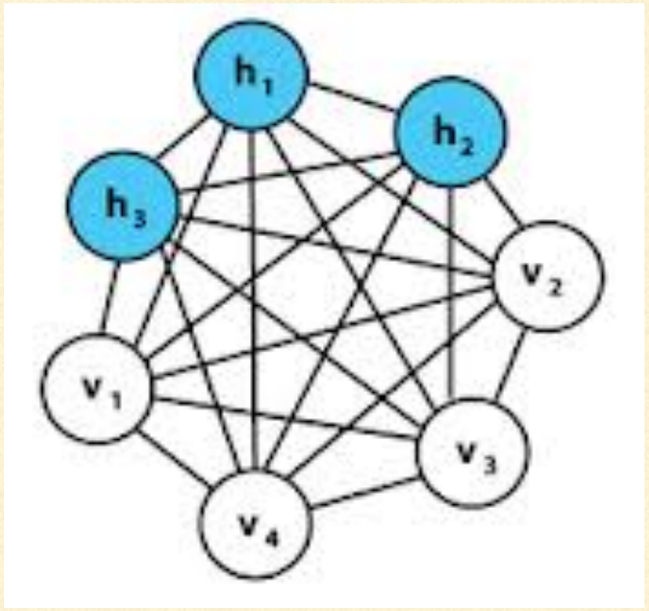
$$\frac{\partial \ell}{\partial \boldsymbol{\theta}_c} = \left[\frac{1}{N}\sum_i \boldsymbol{\phi}_c(\mathbf{y}_i)\right] - \mathbb{E}\left[\boldsymbol{\phi}_c(\mathbf{y})\right]$$

STOCHASTIC GRADIENT METHOD

Algorithm 19.1: Stochastic maximum likelihood for fitting an MRF

Initialize weights $\boldsymbol{\theta}$ randomly;
$k = 0$, $\eta = 1$;
for *each epoch* **do**
  for *each minibatch of size* B **do**
    for *each sample* $s = 1 : S$ **do**
      Sample $\mathbf{y}^{s,k} \sim p(\mathbf{y}|\boldsymbol{\theta}_k)$;
    $\hat{E}(\boldsymbol{\phi}(\mathbf{y})) = \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{\phi}(\mathbf{y}^{s,k})$;
    for *each training case* i *in minibatch* **do**
      $\mathbf{g}_{ik} = \boldsymbol{\phi}(\mathbf{y}_i) - \hat{E}(\boldsymbol{\phi}(\mathbf{y}))$;
    $\mathbf{g}_k = \frac{1}{B}\sum_{i \in B} \mathbf{g}_{ik}$;
    $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \mathbf{g}_k$;
    $k = k + 1$;
    Decrease step size η;

BOLTZMANN MACHINES



- Boltzmann machines were developed in the 1980s
- They are a class of undirected graphical models

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\frac{1}{2}\mathbf{v}^\top \mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^\top \mathbf{J}\mathbf{h} - \mathbf{v}^\top \mathbf{W}\mathbf{h}$$

$$p(\mathbf{v}; \theta) = \frac{p^*(\mathbf{v}; \theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_h \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right)$$

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

$$p(h_j = 1|\mathbf{v}, \mathbf{h}_{-j}) = \sigma\Big(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1\setminus j}^{P} J_{jm} h_j\Big), \quad (4)$$

$$p(v_i = 1|\mathbf{h}, \mathbf{v}_{-i}) = \sigma\Big(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1\setminus i}^{D} L_{ik} v_j\Big), \quad (5)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function.

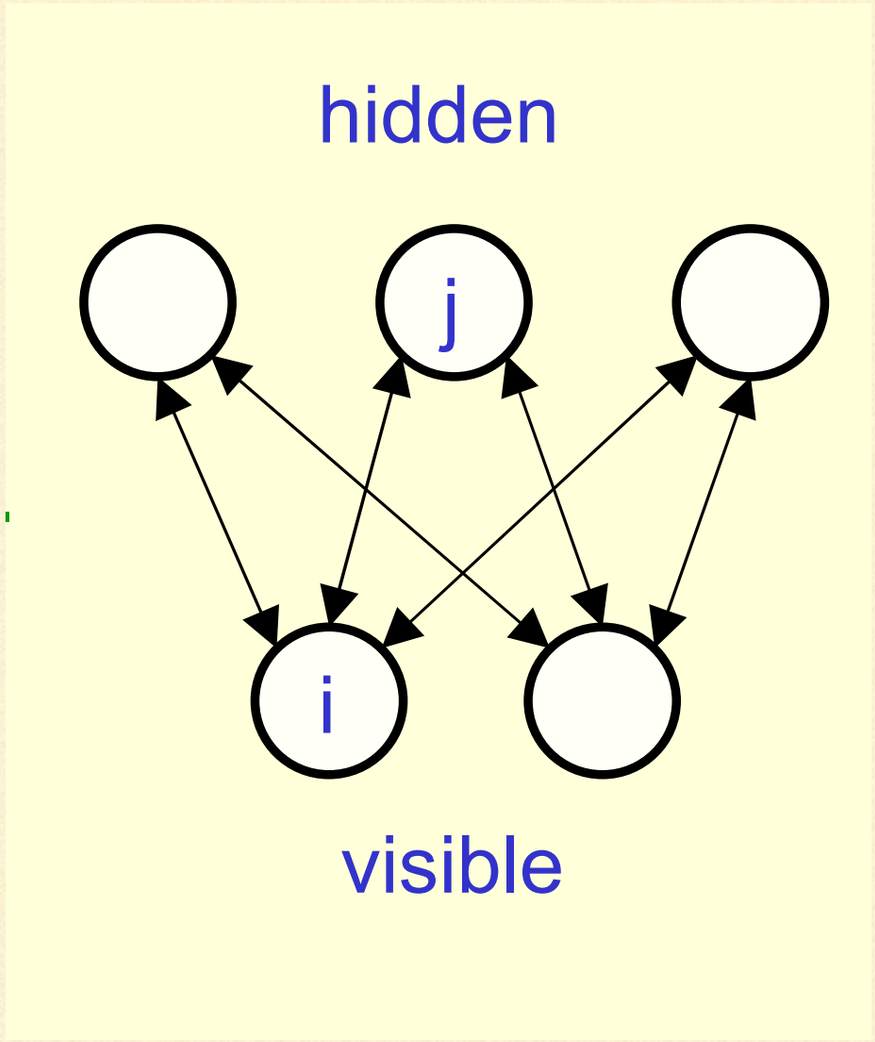
LEARNING BOLTZMANN MACHINES

$$
\begin{aligned}
\Delta \mathbf{W} &= \alpha \left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{h}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{h}^\top] \right), \\
\Delta \mathbf{L} &= \alpha \left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{v}\mathbf{v}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{v}\mathbf{v}^\top] \right), \\
\Delta \mathbf{J} &= \alpha \left(\mathrm{E}_{P_{\mathrm{data}}}[\mathbf{h}\mathbf{h}^\top] - \mathrm{E}_{P_{\mathrm{model}}}[\mathbf{h}\mathbf{h}^\top] \right),
\end{aligned}
$$

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

- A gradient based learning method for Boltzmann machines was developed by Hinton and Sejnowski (1983)
- Exact learning is intractable in Boltzmann machines because computing the expectations takes time exponential in the number of hidden units

RESTRICTED BOLTZMANN MACHINES

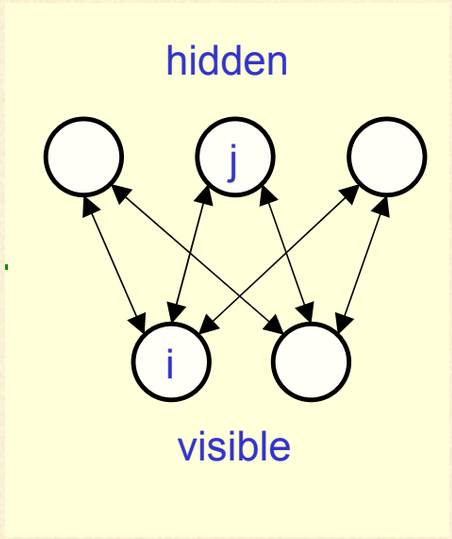


- Undirected graphical models used in deep learning
- Bipartite structure makes inference tractable
- Pairwise Markov random field

$$p(\mathbf{h}|\mathbf{v},\boldsymbol{\theta}) = \prod_k p(h_k|\mathbf{v},\boldsymbol{\theta})$$

$$p(\mathbf{h},\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{r=1}^{R} \prod_{k=1}^{K} \psi_{rk}(v_r, h_k)$$

BINARY RESTRICTED BOLTZMANN MACHINES



- Binary visible nodes and binary hidden nodes
- Joint distribution has the form shown below

$$
\begin{aligned}
p(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta})} \exp(-E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})) \\
E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) &\triangleq -\sum_{r=1}^{R}\sum_{k=1}^{K} v_r h_k W_{rk} - \sum_{r=1}^{R} v_r b_r - \sum_{k=1}^{K} h_k c_k \\
&= -(\mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{v}^T \mathbf{b} + \mathbf{h}^T \mathbf{c}) \\
Z(\boldsymbol{\theta}) &= \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}))
\end{aligned}
$$

INFERENCE IN AN RBM

$$p(\mathbf{h}|\mathbf{v},\boldsymbol{\theta}) = \prod_{k=1}^{K} p(h_k|\mathbf{v},\boldsymbol{\theta}) = \prod_k \mathrm{Ber}(h_k|\mathrm{sigm}(\mathbf{w}_{:,k}^T\mathbf{v}))$$

$$p(\mathbf{v}|\mathbf{h},\boldsymbol{\theta}) = \prod_r p(v_r|\mathbf{h},\boldsymbol{\theta}) = \prod_r \mathrm{Ber}(v_r|\mathrm{sigm}(\mathbf{w}_{r,:}^T\mathbf{h}))$$

$$\begin{aligned}\mathbb{E}\left[\mathbf{h}|\mathbf{v}\boldsymbol{\theta}\right] &= \mathrm{sigm}(\mathbf{W}^T\mathbf{v}) \\ \mathbb{E}\left[\mathbf{v}|\mathbf{h},\boldsymbol{\theta}\right] &= \mathrm{sigm}(\mathbf{W}\mathbf{h})\end{aligned}$$

- Inference in an RBM is greatly simplified by its bipartite structure

OTHER TYPES OF RBMS

- Categorical RBMs:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) \triangleq -\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{c=1}^{C} v_r^c h_k W_{rk}^c - \sum_{r=1}^{R}\sum_{c=1}^{C} v_r^c b_r^c - \sum_{k=1}^{K} h_k c_k$$

The full conditionals are given by

$$p(v_r|\mathbf{h}, \boldsymbol{\theta}) = \text{Cat}(\mathcal{S}(\{b_r^c + \sum_k h_k W_{rk}^c\}_{c=1}^{C}))$$

$$p(h_k = 1|\mathbf{c}, \boldsymbol{\theta}) = \text{sigm}(c_k + \sum_r \sum_c v_r^c W_{rk}^c)$$

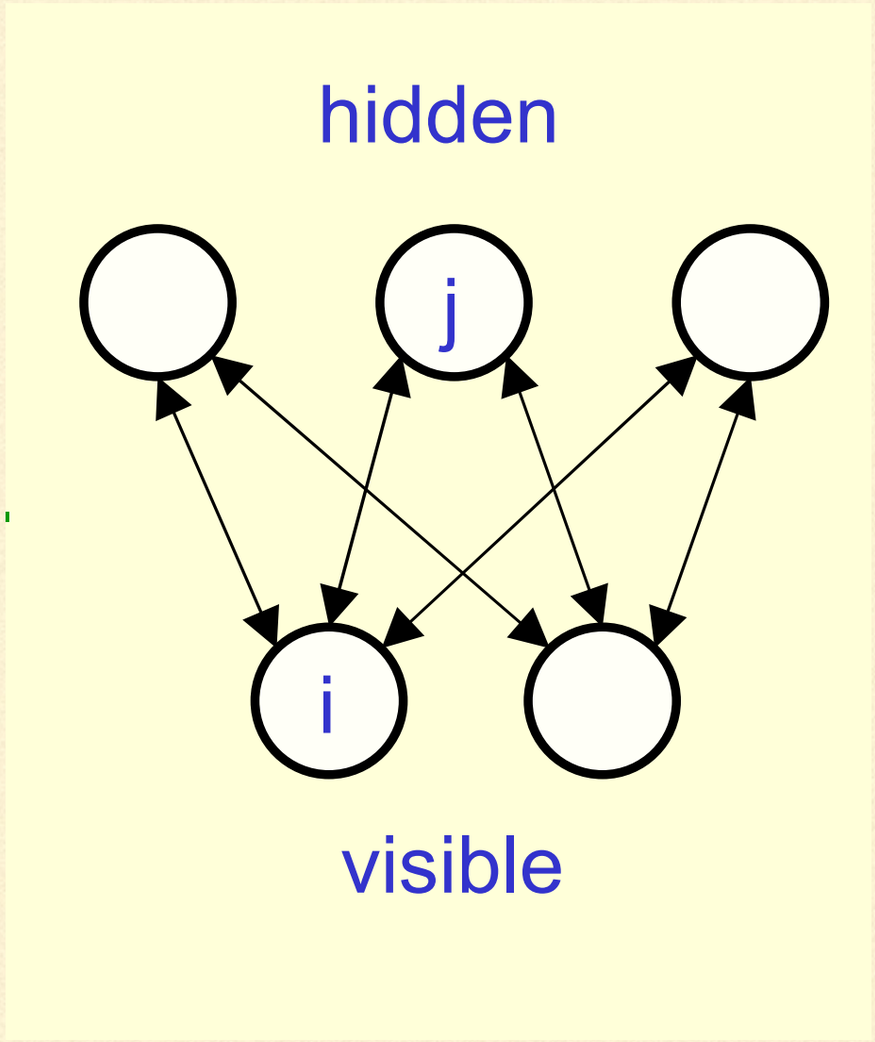
- Gaussian RBMs:

$$E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = -\sum_{r=1}^{R}\sum_{k=1}^{K} W_{rk} h_k v_r - \frac{1}{2}\sum_{r=1}^{R}(v_r - b_r)^2 - \sum_{k=1}^{K} a_k h_k$$

$$p(v_r|\mathbf{h}, \boldsymbol{\theta}) = \mathcal{N}(v_r|b_r + \sum_k w_{rk} h_k, 1)$$

$$p(h_k = 1|\mathbf{v}, \boldsymbol{\theta}) = \text{sigm}\left(c_k + \sum_r w_{rk} v_r\right)$$

LEARNING IN RBMS



- The learning rule for training RBMs is surprisingly simple:

$$\frac{\partial \ell}{\partial w_{rk}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[v_r h_k | \mathbf{v}_i, \boldsymbol{\theta}\right] - \mathbb{E}\left[v_r h_k | \boldsymbol{\theta}\right]$$

- This accounts for their popularity
- However, efficient learning requires making quite a few approximations
  - Gibbs sampling and contrastive divergence methods

LEARNING RBMS

binary state of visible unit i

binary state of hidden unit j

$$E(v,h) = -\sum_{i,j} v_i h_j w_{ij}$$

Energy with configuration v on the visible units and h on the hidden units

weight between units i and j

$$-\frac{\partial E(v,h)}{\partial w_{ij}} = v_i h_j$$

- Deep learning models use gradient-based methods to derive maximum likelihood estimators for RBMs
- Gradient-based ML estimators enable scaling deep learning models to large datasets

$$
\begin{aligned}
F(\mathbf{v}) &\triangleq \sum_{\mathbf{h}} E(\mathbf{v},\mathbf{h}) = \sum_{\mathbf{h}} \exp\left(\sum_{r=1}^{R}\sum_{k=1}^{K} v_r h_k W_{rk}\right) \\
&= \sum_{\mathbf{h}} \prod_{k=1}^{K} \exp\left(\sum_{r=1}^{R} v_r h_k W_{rk}\right) \\
&= \prod_{k=1}^{K} \sum_{h_r \in \{0,1\}} \exp\left(\sum_{r=1}^{R} v_r h_r W_{rk}\right) \\
&= \prod_{k=1}^{K} \left(1 + \exp(\sum_{r=1}^{R} v_r W_{rk})\right)
\end{aligned}
$$

Next we write the (scaled) log-likelihood in the following form:

$$
\ell(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \log p(\mathbf{v}_i|\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} F(\mathbf{v}_i|\boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})
$$

Using the fact that $Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}} \exp(-F(\mathbf{v};\boldsymbol{\theta}))$ we have

$$
\begin{aligned}
\nabla \ell(\boldsymbol{\theta}) &= -\frac{1}{N}\sum_{i=1}^{N} \nabla F(\mathbf{v}_i) - \frac{\nabla Z}{Z} \\
&= -\frac{1}{N}\sum_{i=1}^{N} \nabla F(\mathbf{v}_i) + \sum_{\mathbf{v}} \nabla F(\mathbf{v}) \frac{\exp(-F(\mathbf{v}))}{Z} \\
&= -\frac{1}{N}\sum_{i=1}^{N} \nabla F(\mathbf{v}_i) + \mathbb{E}\left[\nabla F(\mathbf{v})\right]
\end{aligned}
$$

Plugging in the free energy (Equation 27.117), one can show that

$$
\frac{\partial}{\partial w_{rk}} F(\mathbf{v}) = -v_r \mathbb{E}\left[h_k|\mathbf{v},\boldsymbol{\theta}\right] = -\mathbb{E}\left[v_r h_k|\mathbf{v},\boldsymbol{\theta}\right]
$$

See Chapter 27 Section 27.1 in Murphy's textbook

GIBBS SAMPLING

- $x_1^{s+1} \sim p(x_1|x_2^s, x_3^s)$
- $x_2^{s+1} \sim p(x_2|x_1^{s+1}, x_3^s)$
- $x_3^{s+1} \sim p(x_3|x_1^{s+1}, x_2^{s+1})$

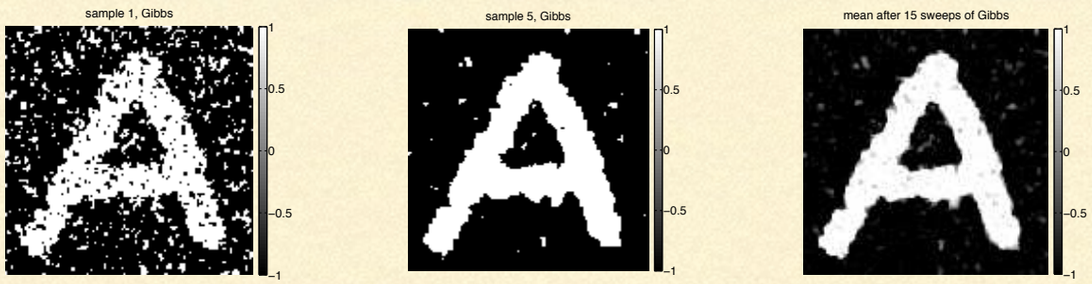
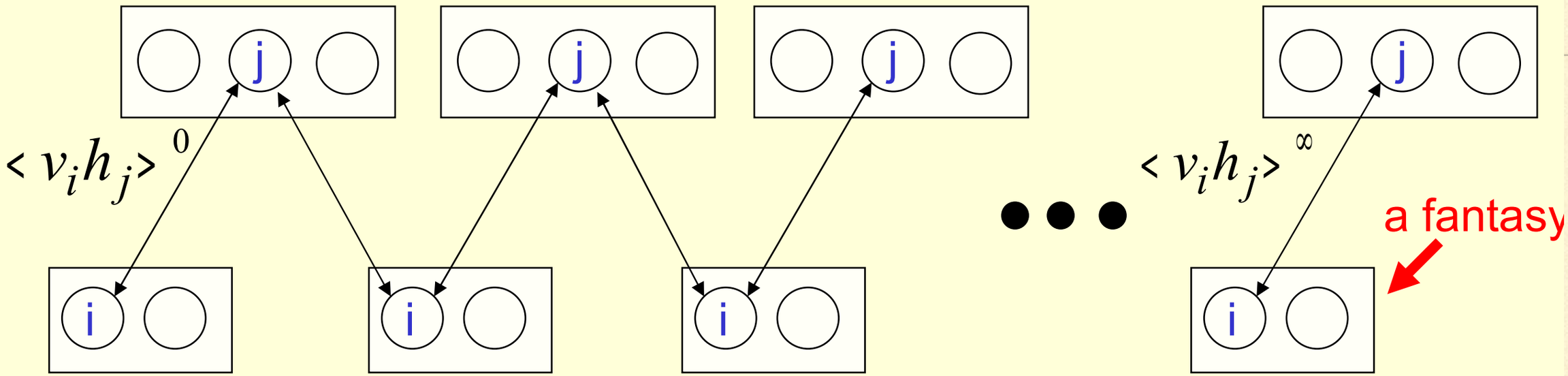


Image Denoising with Gibbs Sampling

- Gibbs sampling is a widely used sampling method in statistics to sample from complex distributions
- It is widely used in graphical models
- It is an example of a class of sampling algorithms called Markov Chain Monte Carlo (MCMC)
- Gibbs sampling can be slow because it only samples one variable at a time

A picture of the maximum likelihood learning algorithm for an RBM

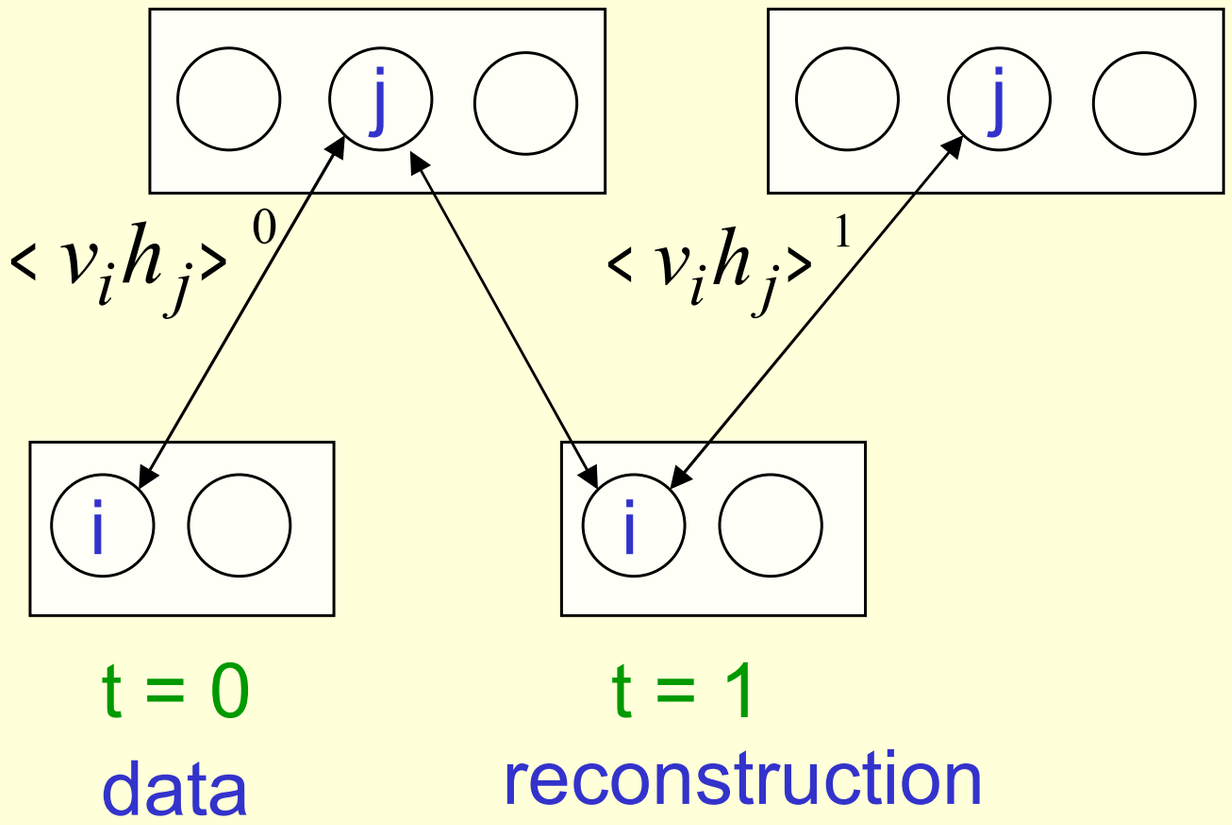


Start with a training vector on the visible units.

Then alternate between updating all the hidden units in parallel and updating all the visible units in parallel.

$$\frac{\partial \log p(v)}{\partial w_{ij}} = < v_i h_j >^0 - < v_i h_j >^\infty$$

A quick way to learn an 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.

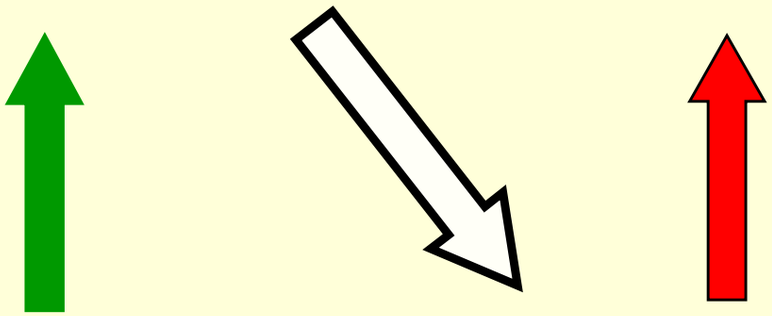
$$\Delta w_{ij} = \varepsilon \left(< v_i h_j >^0 - < v_i h_j >^1 \right)$$

How to learn a set of features that are good for reconstructing images of the digit 2

50 binary feature neurons

50 binary feature neurons

Increment weights between an active pixel and an active feature



Decrement weights between an active pixel and an active feature

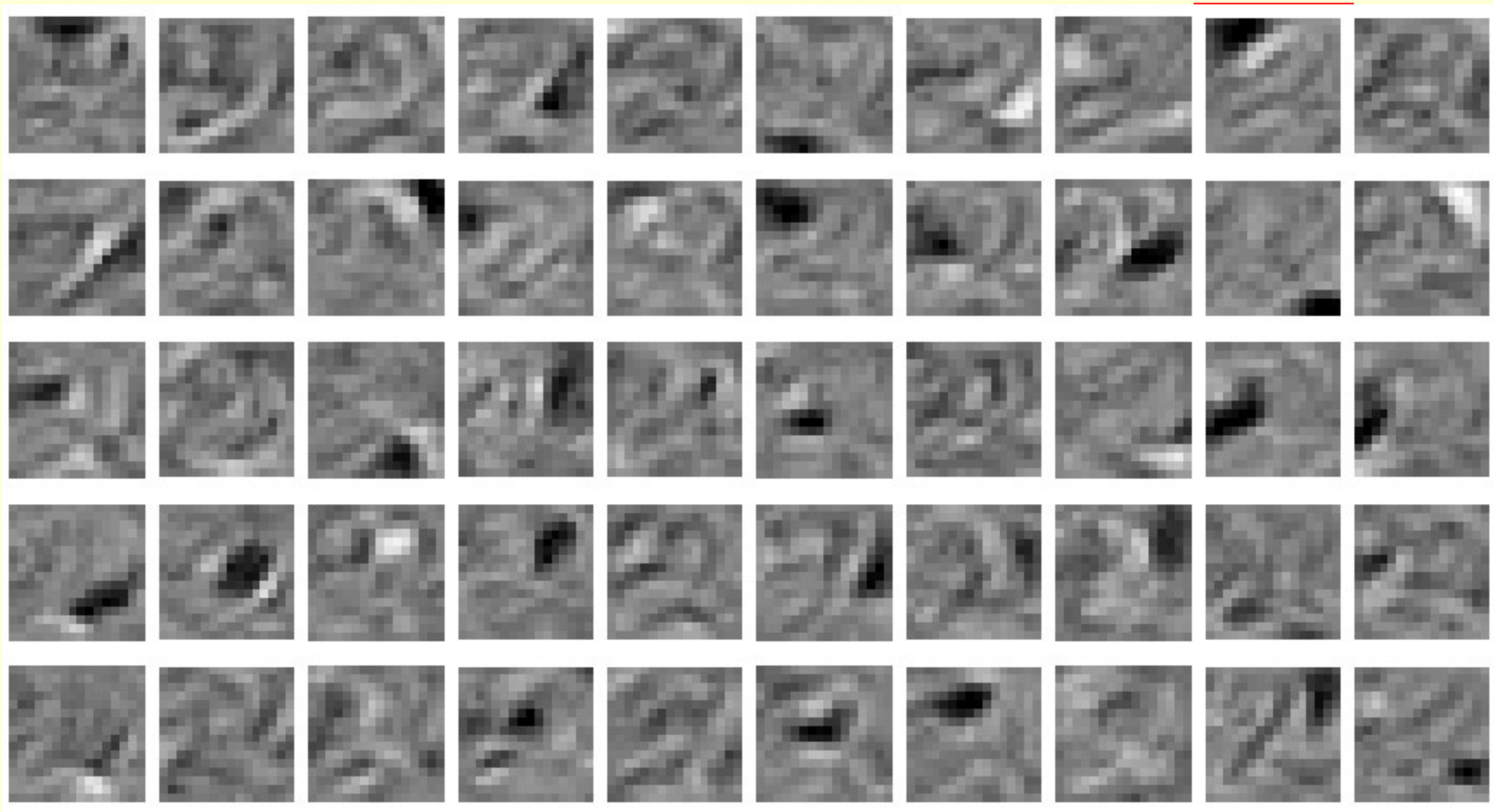
16 x 16 pixel image

data

16 x 16 pixel image

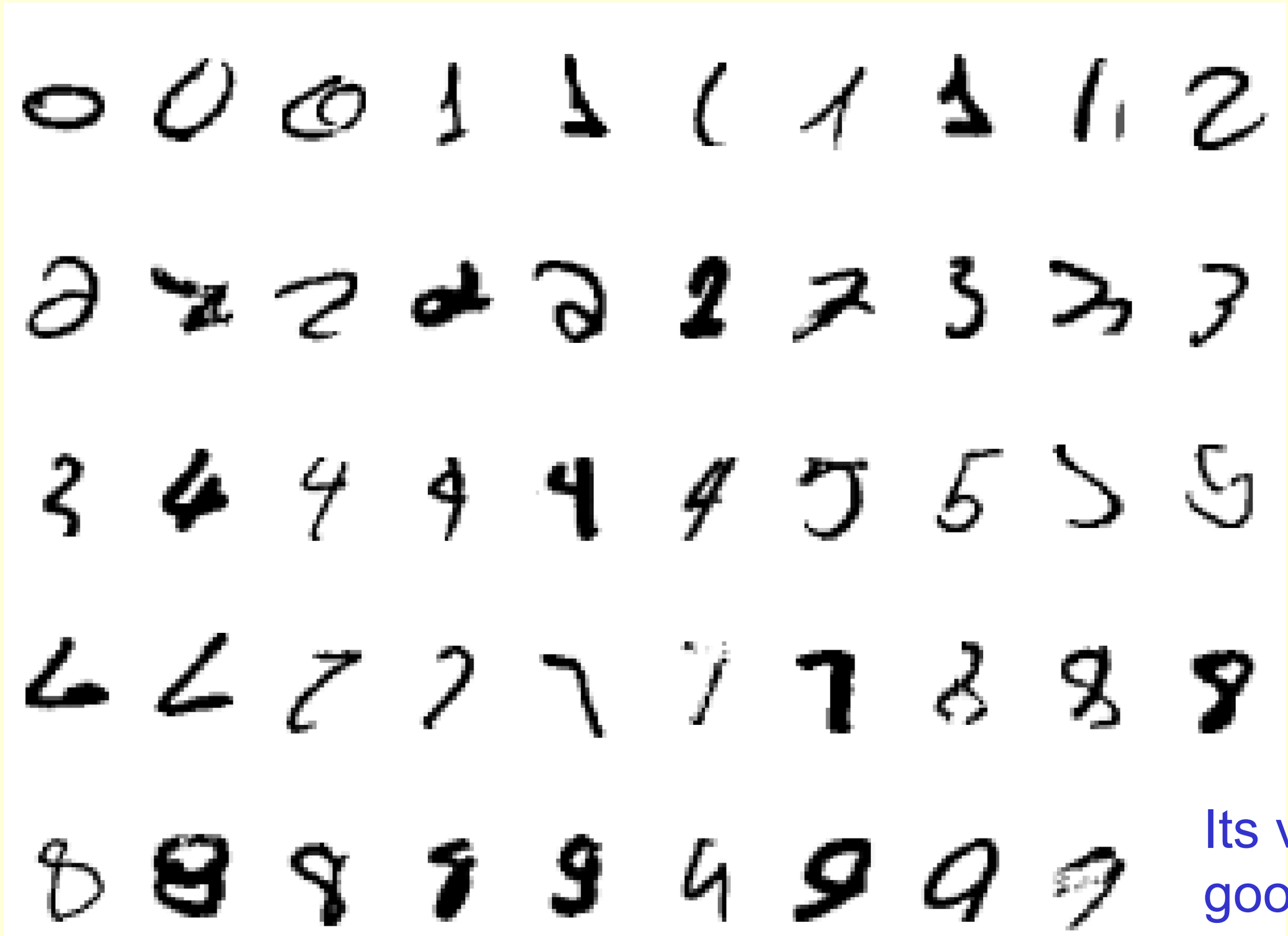
reconstruction

The final 50 x 256 weights

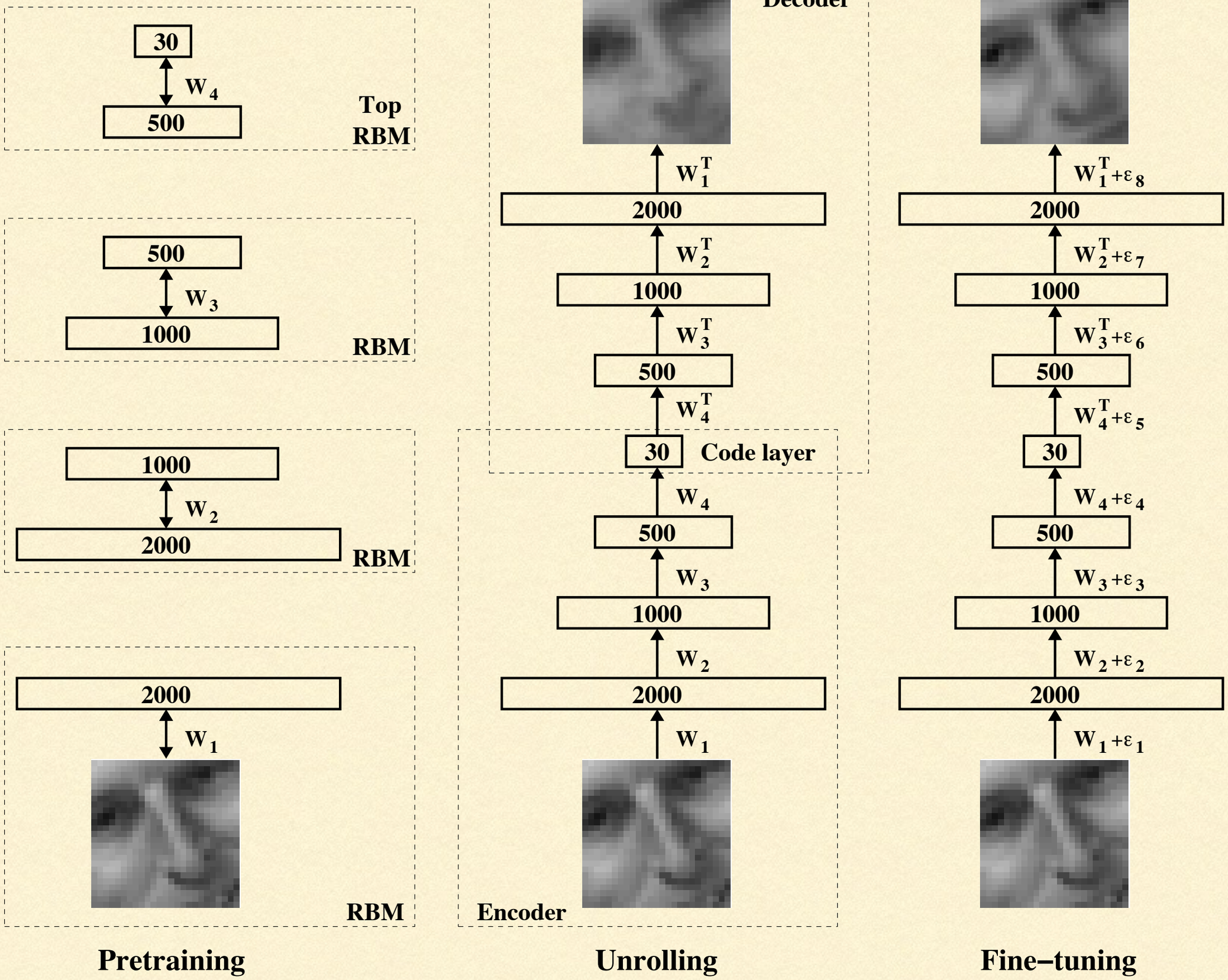


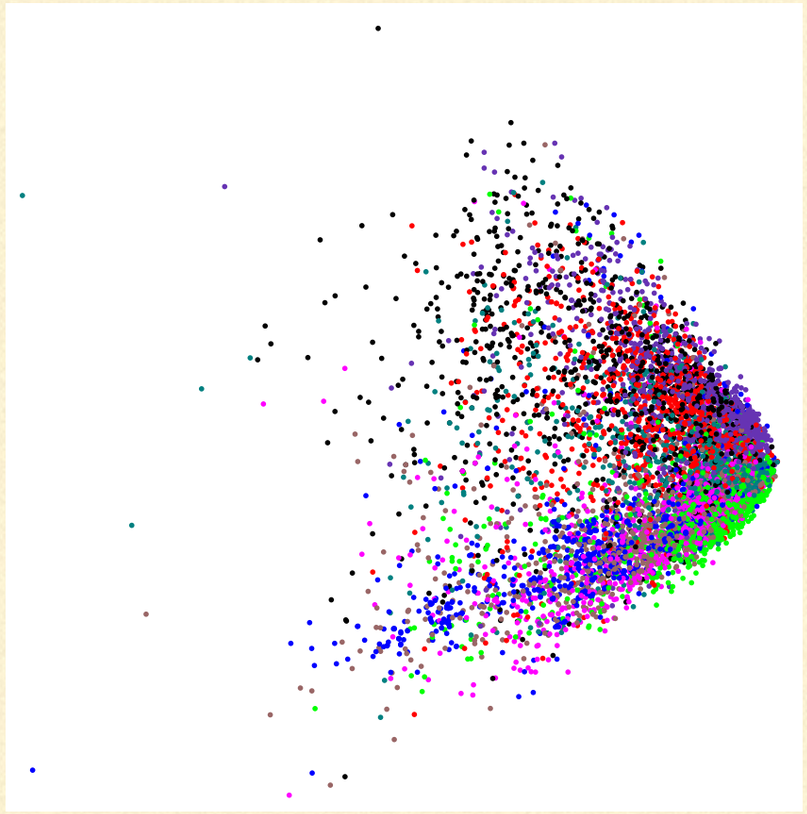
Each neuron grabs a different feature.

Examples of correctly recognized handwritten digits that the neural network had never seen before

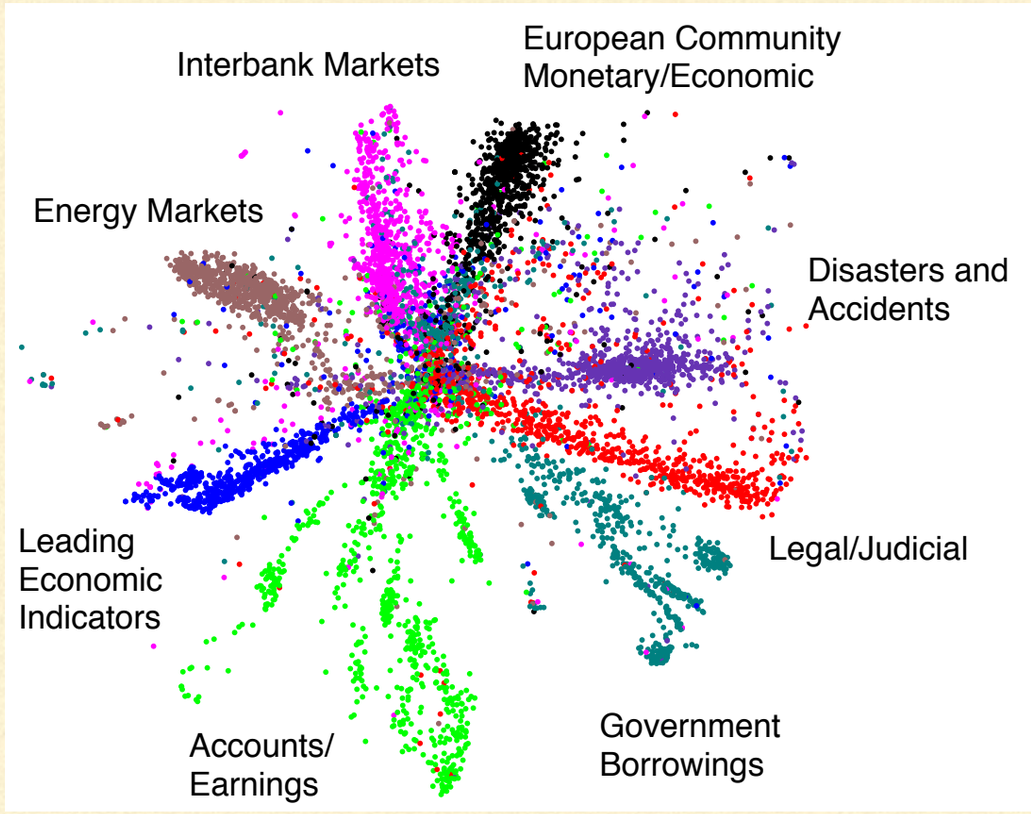


Its very good


30
W4
500
Top RBM
500
W3
1000
RBM
1000
W2
2000
RBM
2000
W1
RBM
Pretraining
Decoder
W1T
2000
W2T
1000
W3T
500
W4T
30
Code layer
W4
500
W3
1000
W2
2000
W1
Encoder
Unrolling
W1T+ε8
2000
W2T+ε7
1000
W3T+ε6
500
W4T+ε5
30
W4+ε4
500
W3+ε3
1000
W2+ε2
2000
W1+ε1
Fine–tuning



Latent Semantic Indexing



Deep Autoencoders

Results on a Reuters news collection
(see Chapter 28, Murphy ML text)

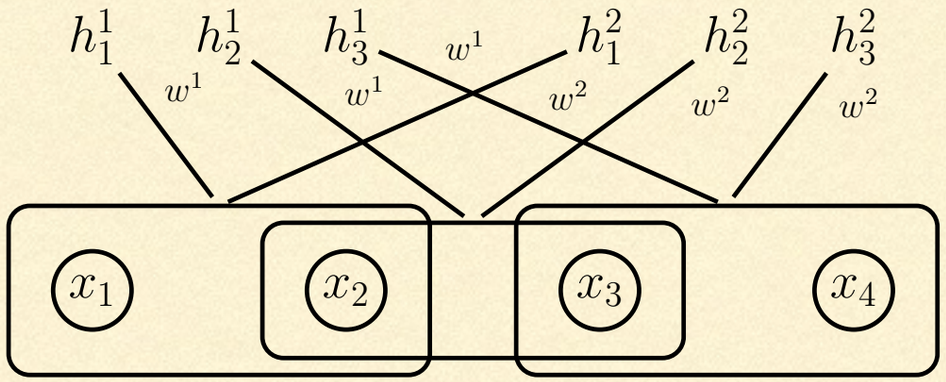
LEARNING IMAGE FEATURES

faces, cars, airplanes, motorbikes

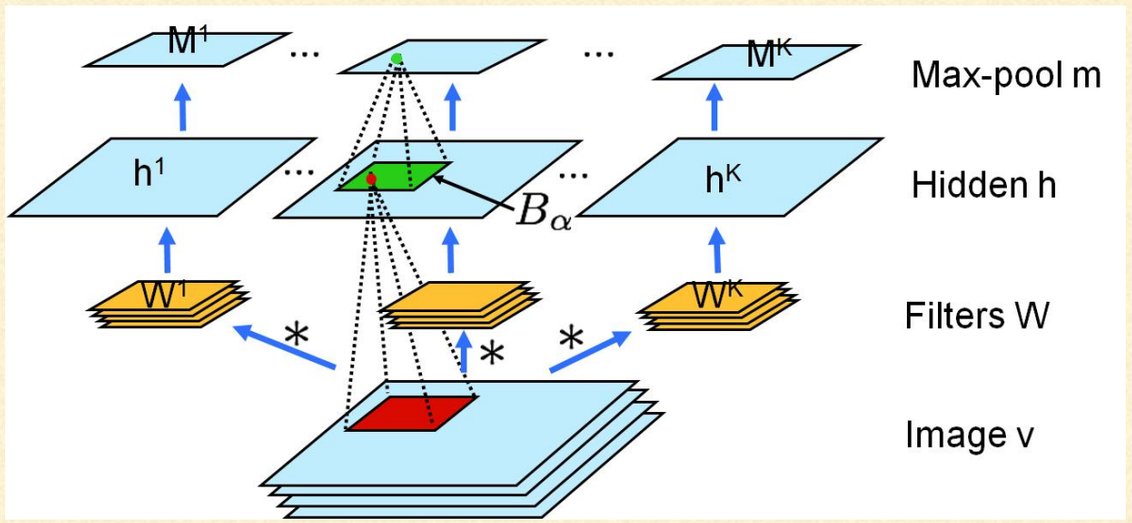
(a)

(b)

CONVOLUTIONAL NETWORKS



(a)



(b)

CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
linear classifier (1-layer NN)	deskewing	8.4	LeCun et al. 1998
pairwise linear classifier	deskewing	7.6	LeCun et al. 1998
K-Nearest Neighbors			
K-nearest-neighbors, Euclidean (L2)	none	5.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	deskewing, noise removal, blurring	1.80	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring	1.73	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 pixel shift	1.33	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 pixel shift	1.22	Kenneth Wilder, U. Chicago

K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
K-NN, Tangent Distance	subsampling to 16x16 pixels	1.1	LeCun et al. 1998
K-NN, shape context matching	shape context feature extraction	0.63	Belongie et al. IEEE PAMI 2002
Boosted Stumps			
boosted stumps	none	7.7	Kegl et al., ICML 2009
products of boosted stumps (3 terms)	none	1.26	Kegl et al., ICML 2009
boosted trees (17 leaves)	none	1.53	Kegl et al., ICML 2009
stumps on Haar features	Haar features	1.02	Kegl et al., ICML 2009
product of stumps on Haar f.	Haar features	0.87	Kegl et al., ICML 2009
Non-Linear Classifiers			
40 PCA + quadratic classifier	none	3.3	LeCun et al. 1998
1000 RBF + linear classifier	none	3.6	LeCun et al. 1998
SVMs			
SVM, Gaussian Kernel	none	1.4	
SVM deg 4 polynomial	deskewing	1.1	LeCun et al. 1998
Reduced Set SVM deg 5 polynomial	deskewing	1.0	LeCun et al. 1998
Virtual SVM deg-9 poly [distortions]	none	0.8	LeCun et al. 1998
Virtual SVM, deg-9 poly, 1-pixel jittered	none	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 1-pixel jittered	deskewing	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 2-pixel jittered	deskewing	0.56	DeCoste and Scholkopf, MLJ 2002

3-layer NN, 300+100 hidden units	none	3.05	LeCun et al. 1998
3-layer NN, 300+100 HU [distortions]	none	2.5	LeCun et al. 1998
3-layer NN, 500+150 hidden units	none	2.95	LeCun et al. 1998
3-layer NN, 500+150 HU [distortions]	none	2.45	LeCun et al. 1998
3-layer NN, 500+300 HU, softmax, cross entropy, weight decay	none	1.53	Hinton, unpublished, 2005
2-layer NN, 800 HU, Cross-Entropy Loss	none	1.6	Simard et al., ICDAR 2003
2-layer NN, 800 HU, cross-entropy [affine distortions]	none	1.1	Simard et al., ICDAR 2003
2-layer NN, 800 HU, MSE [elastic distortions]	none	0.9	Simard et al., ICDAR 2003
2-layer NN, 800 HU, cross-entropy [elastic distortions]	none	0.7	Simard et al., ICDAR 2003
NN, 784-500-500-2000-30 + nearest neighbor, RBM + NCA training [no distortions]	none	1.0	Salakhutdinov and Hinton, AI-Stats 2007
6-layer NN 784-2500-2000-1500-1000-500-10 (on GPU) [elastic distortions]	none	0.35	Ciresan et al. Neural Computation 10, 2010 and arXiv 1003.0358, 2010
committee of 25 NN 784-800-10 [elastic distortions]	width normalization, deslanting	0.39	Meier et al. ICDAR 2011
deep convex net, unsup pre-training [no distortions]	none	0.83	Deng et al. Interspeech 2010

FURTHER READING

- Chapter 27 and Chapter 28, Murphy's textbook on ML

- Hinton, G. E., Osindero, S. and Teh, Y. (2006), *A fast learning algorithm for deep belief nets*, Neural Computation, 18, pp 1527-1554.

- Hinton, G. E. and Salakhutdinov, R. R. (2006), *Reducing the dimensionality of data with neural networks*, Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.