Machine Learning - SS 2011

Lecture 6, Deep Learning

Grégoire Montavon

Technische Universität Berlin

May 17, 2011

Deep learning

- Deep learning is the field of research that is concerned with learning automatically hierarchical representations of data.
- By learning a hierarchical model of data, deep learning aims to simultaneously deliver *high generalization* and *low computational cost* at prediction time.

A spectrum of machine learning tasks

Statistics

- There is not much structure in the data
- Simple models are typically sufficient to model the task
- Main challenge: separating the signal from the noise
- Prior knowledge can bring significant improvement

Artificial Intelligence

- There is a huge amount of structure in the data
- We need complex hierarchical models for the task
- Main challenge: discovering the structure of data
- Prior knowledge is already in the data

A spectrum of machine learning tasks

Statistics

Typical applications

- Brain computer interfaces
- Financial data analysis
- Statistical physics

Methods

- Generalized linear models
- Structured losses
- Kernels

Artificial Intelligence

Typical applications

- Computer vision
- Natural language processing
- Sequential reasoning

Methods

- Engineering
- Deep networks
- Unsupervised learning

Representing probability densities

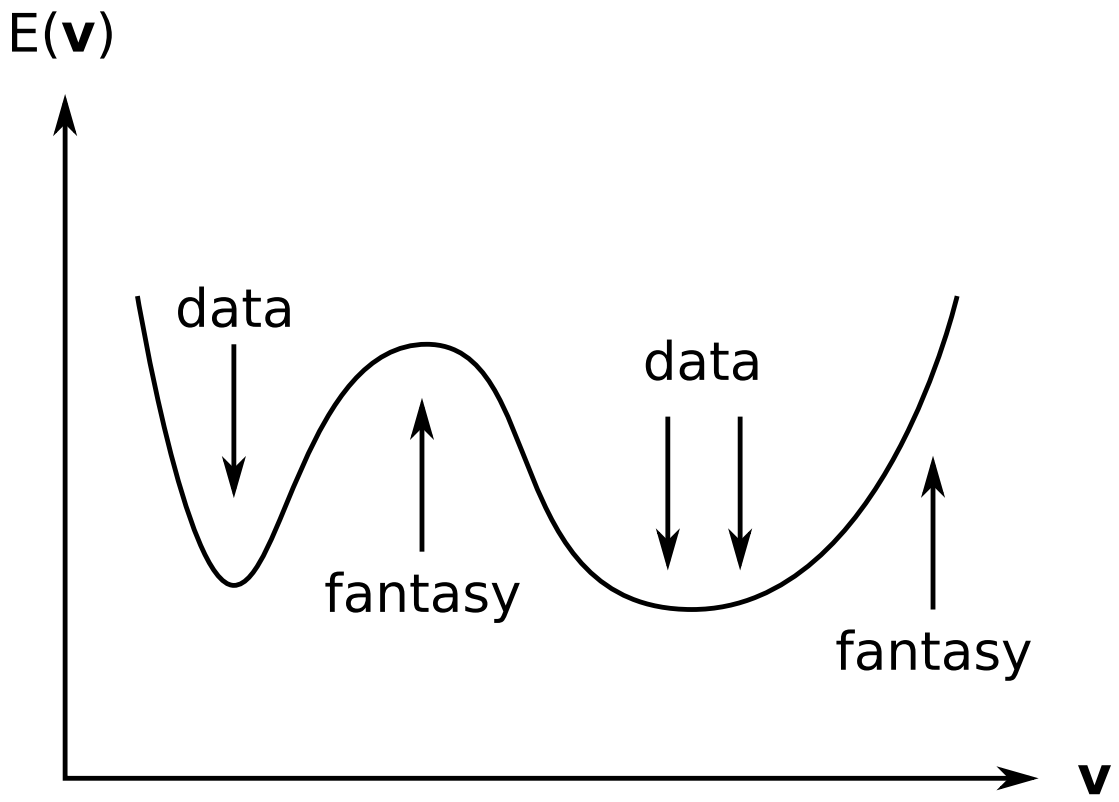
Problem:

- Let $\mathrm{p}(\boldsymbol{v}),\ \boldsymbol{v} \in \{0,1\}^d$ be an *unknown* probability distribution
- Estimate $\mathrm{p}(\boldsymbol{v})$ from a finite set of samples

Idea: learn an energy function $\mathrm{E}(\boldsymbol{v};\theta)$ of such that

- $\mathrm{E}(\boldsymbol{v};\theta)$ is low when $\boldsymbol{v} \in$ data
- $\mathrm{E}(\boldsymbol{v};\theta)$ is high when $\boldsymbol{v} \notin$ data
- $\mathrm{E}(\boldsymbol{v};\theta)$ is smooth

Representing probability densities



Find a model such that $\mathrm{E}(\boldsymbol{v};\theta)$ is low for $\boldsymbol{v} \in$ data and high elsewhere

A simple energy-based model

- Define an energy function:

$$\mathrm{E}(\boldsymbol{v};\theta) = -\sum_{i=1}^{d} v_i b_i \qquad\qquad \theta = \boldsymbol{b}$$

- Associate a probability to the energies:

$$p(\boldsymbol{v}) = \frac{1}{Z(\theta)} e^{-\mathrm{E}(\boldsymbol{v};\theta)}$$

  Where the partition function $Z(\theta) = \sum_{\boldsymbol{u}} e^{-\mathrm{E}(\boldsymbol{u};\theta)}$ forces probabilities to sum to 1.

- Learning rule:

$$b_i = b_i + \epsilon(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}})$$

- Problem: doesn't capture dependencies between variables

A simple energy-based model

- Define an energy function:

$$\mathrm{E}(\boldsymbol{v};\theta) = -\sum_{i=1}^{d} v_i b_i - \sum_{i,j} v_i v_j w_{ij} \qquad \theta = (W, \boldsymbol{b})$$

- Associate a probability to the energies:

$$p(\boldsymbol{v}) = \frac{1}{Z(\theta)} \mathrm{e}^{-\mathrm{E}(\boldsymbol{v};\theta)}$$

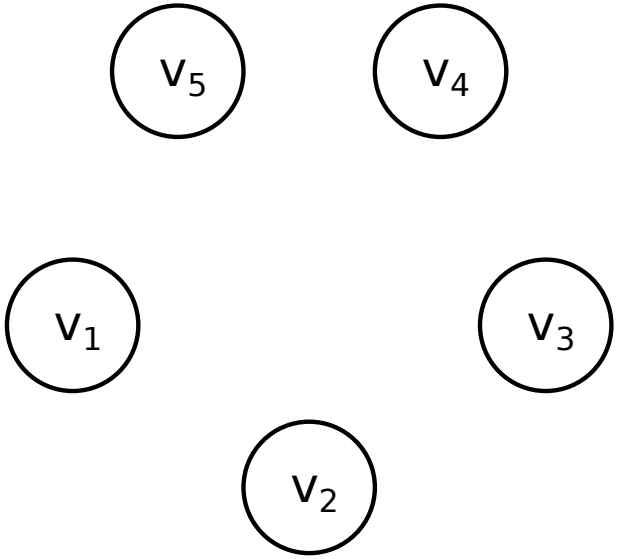
- Learning rule:

$$b_i = b_i + \epsilon\big(\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}\big)$$
$$w_{ij} = w_{ij} + \epsilon\big(\langle v_i v_j \rangle_{\text{data}} - \langle v_i v_j \rangle_{\text{model}}\big)$$

A simple energy-based model

Independent variables

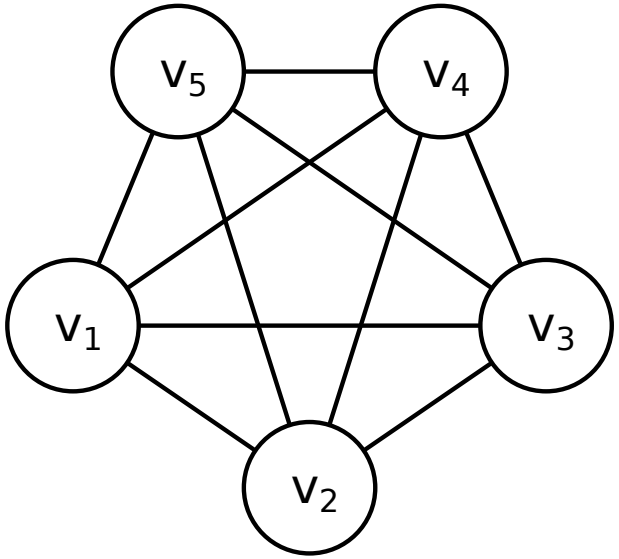
$\theta = \boldsymbol{b}$



$$\mathrm{E}(\boldsymbol{v};\theta) = -\sum_{i=1}^{d} v_i b_i$$

Dependent variables

$\theta = (W, \boldsymbol{b})$

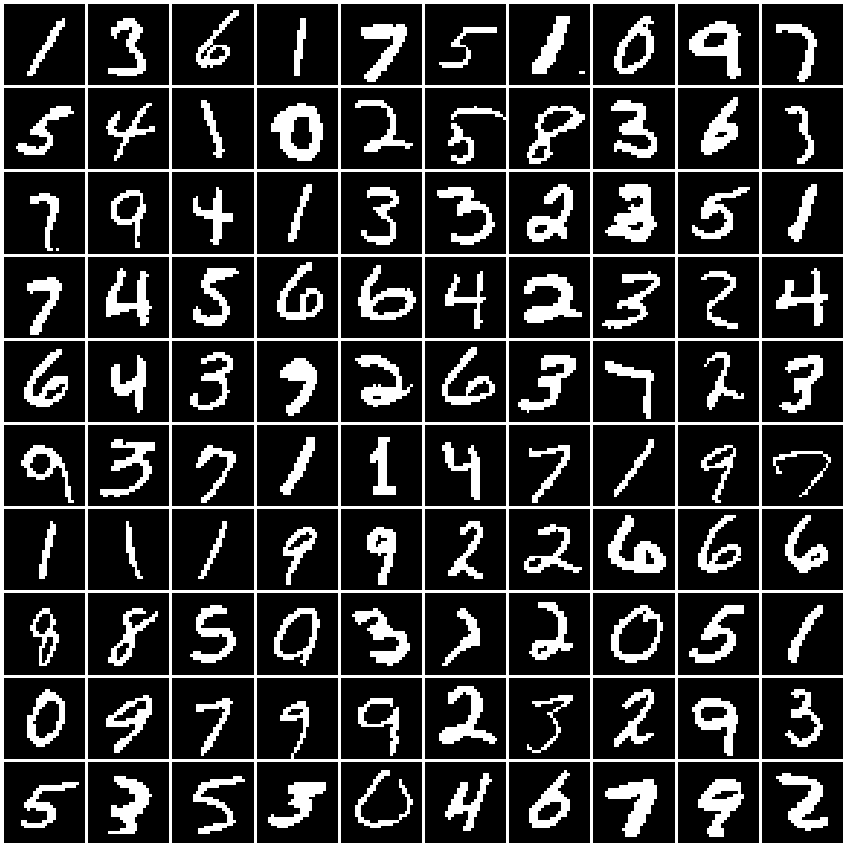


$$\mathrm{E}(\boldsymbol{v};\theta) = -\sum_{i=1}^{d} v_i b_i - \sum_{i,j} v_i v_j w_{ij}$$

A simple energy-based model

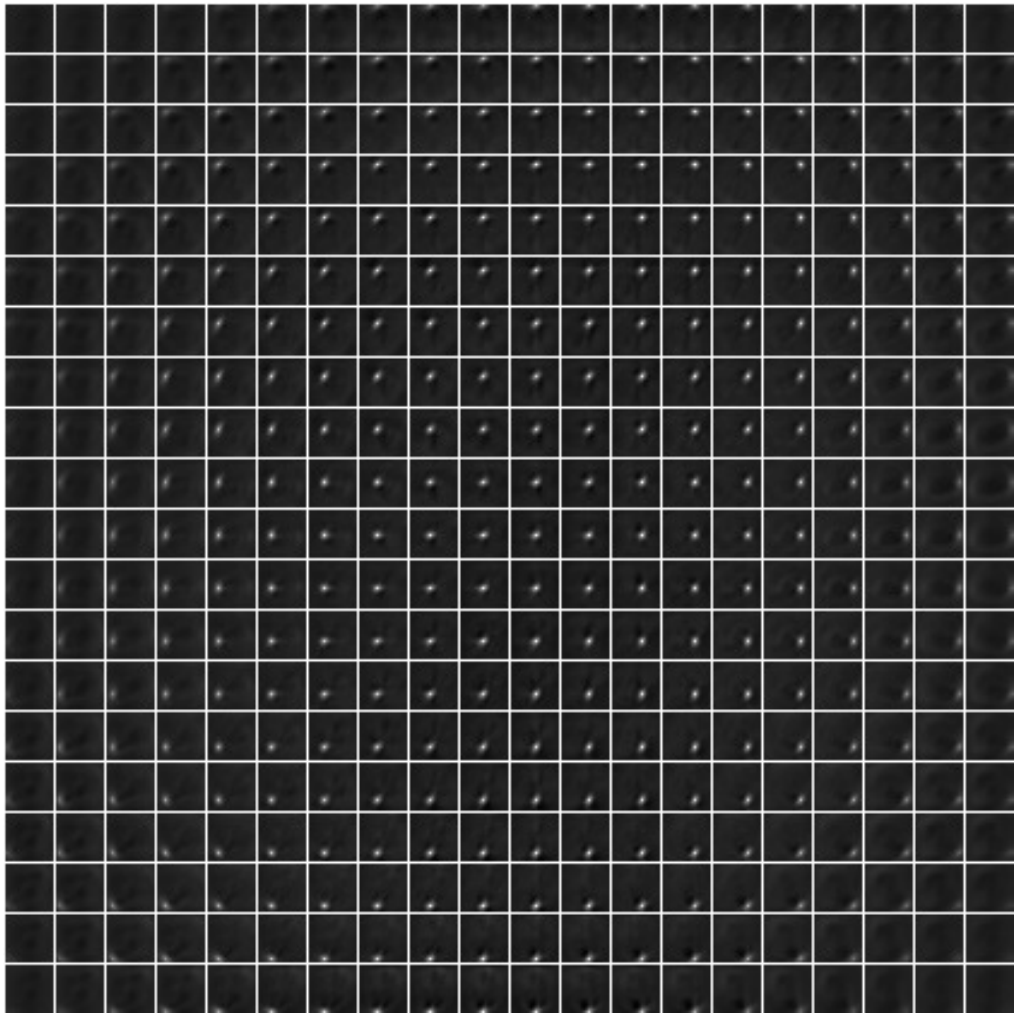
Example: handwritten digits

- Each pixel is a variable
- W models the relation between each pixels



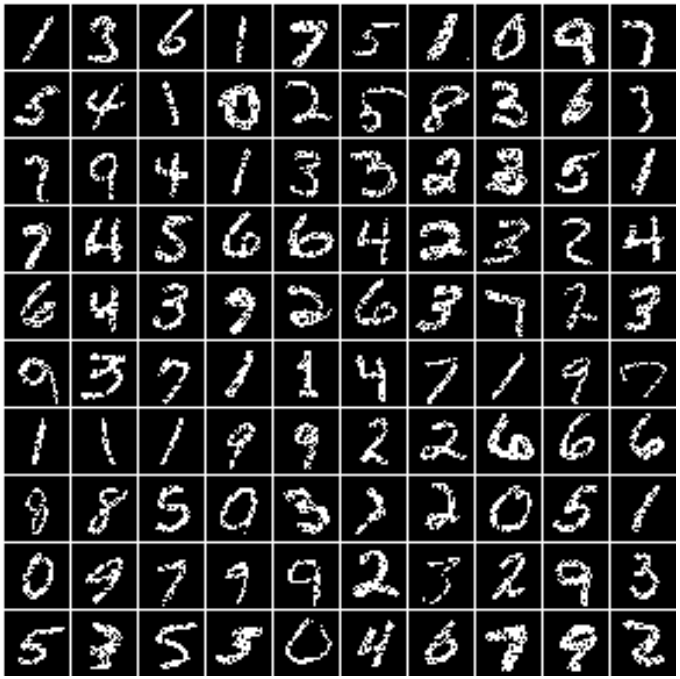
A simple energy-based model

W =

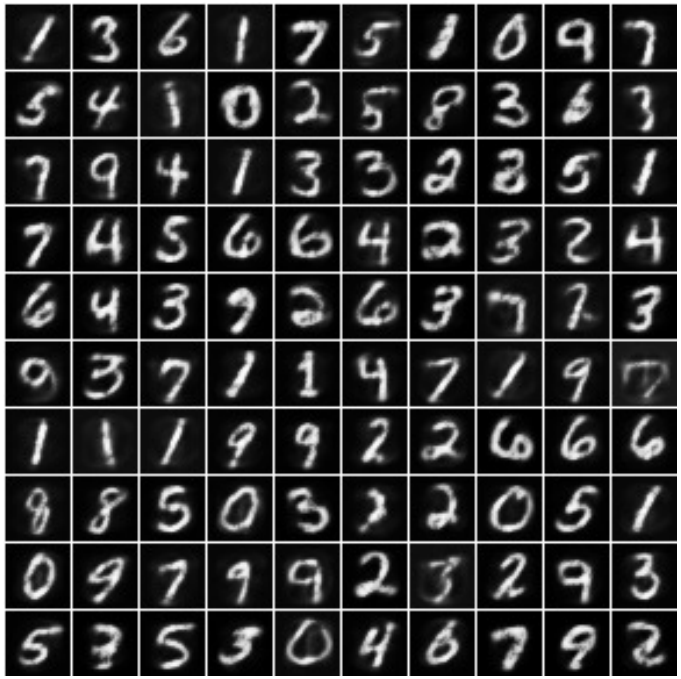


b =

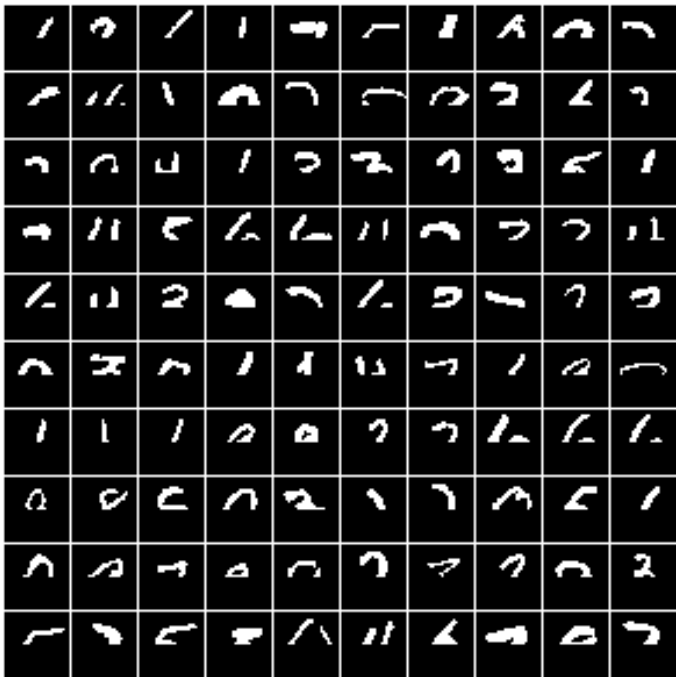
Denoising samples



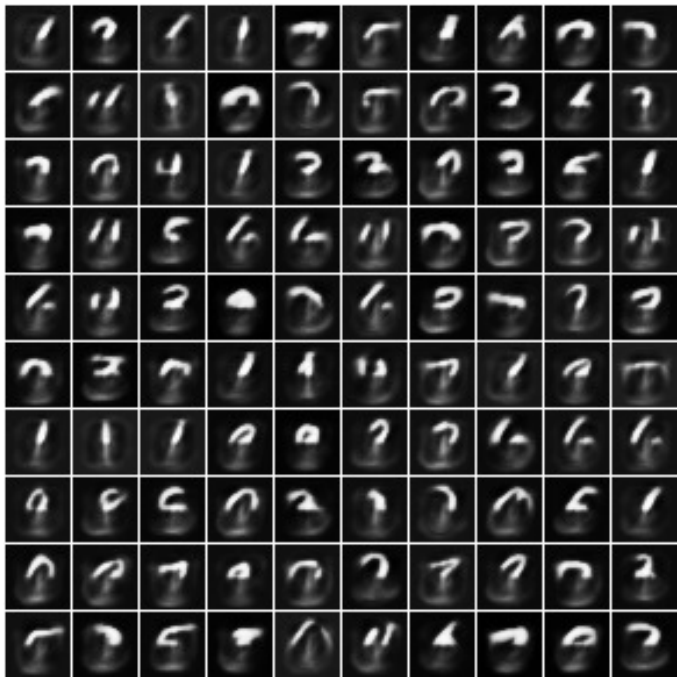
→



Completing samples



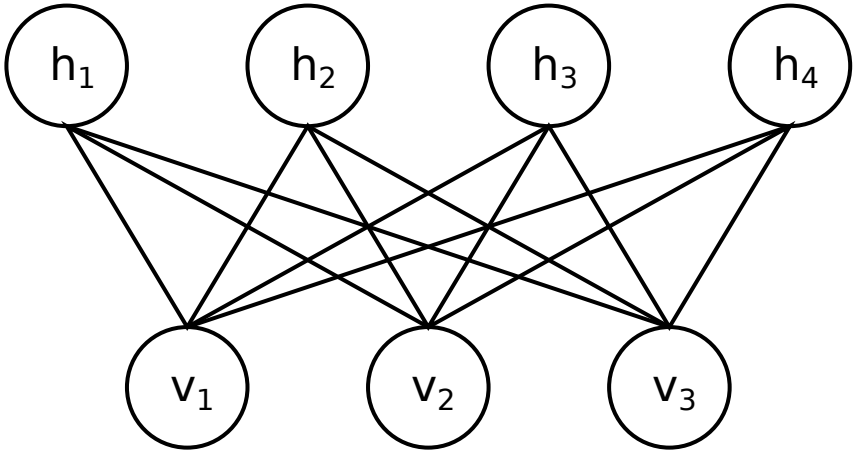
$\rightarrow$



On complicated problems, pairwise interactions between variables are not sufficient to model the distribution. Therefore, we need to introduce hidden variables.

The restricted Boltzmann machine (RBM)

- Let $\boldsymbol{v}$ be the visible variables (observations)
- Let $\boldsymbol{h}$ be the hidden variables (causes)



The energy function of the RBM is defined as follows:

$$\mathrm{E}(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\sum_{i \in \text{visible}} v_i a_i - \sum_{j \in \text{hidden}} h_j b_j - \sum_{i,j} v_i w_{ij} h_j$$

The restricted Boltzmann machine (RBM)

- The energy function of the RBM is defined as follows:

$$\mathrm{E}(\boldsymbol{v}, \boldsymbol{h}; \theta) = -\sum_{i \in \text{visible}} v_i a_i - \sum_{j \in \text{hidden}} h_j b_j - \sum_{i,j} v_i w_{ij} h_j$$

- A probability $p(\boldsymbol{v}, \boldsymbol{h})$ is associated to the energies:

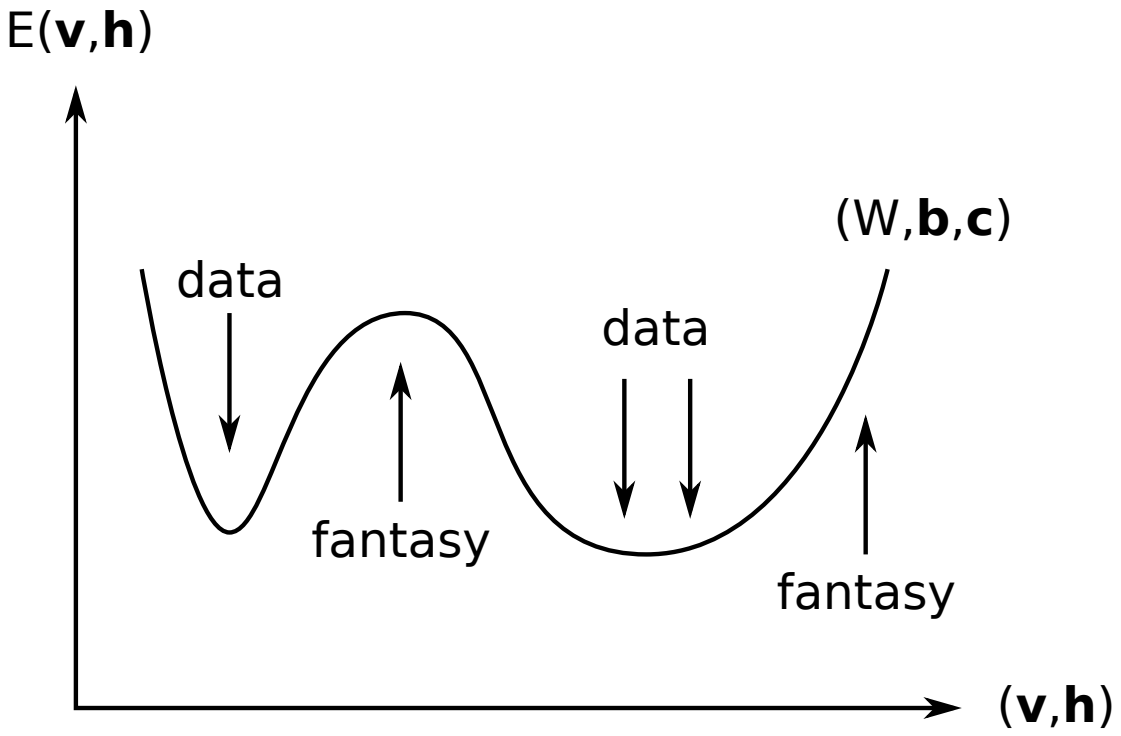
$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z(\theta)} \mathrm{e}^{-\mathrm{E}(\boldsymbol{v}, \boldsymbol{h}; \theta)}$$

- Sampling from the model:

$$p(\boldsymbol{h}|\boldsymbol{v}) = \sigma(W^\top \boldsymbol{v} + \boldsymbol{b}) \qquad \text{and} \qquad p(\boldsymbol{v}|\boldsymbol{h}) = \sigma(W\boldsymbol{h} + \boldsymbol{a})$$

where $\sigma(t) = \frac{1}{1+\mathrm{e}^{-t}}$ is the logistic function.

Training an RBM



Idea: find $W, \boldsymbol{b}, \boldsymbol{a}$ such that $\mathrm{E}(\boldsymbol{v}, \boldsymbol{h}; \theta)$ is low when $(\boldsymbol{v}, \boldsymbol{h}) \in$ data and high elsewhere

Training an RBM

- We want to minimize $\langle \log p(\boldsymbol{v}) \rangle_{\text{data}}$
- Gradient descent:

$$\left\langle \frac{\partial \log p(\boldsymbol{v})}{dw_{ij}} \right\rangle_{\text{data}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$$

- Update rule:

$$\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}})$$

- Unfortunately, we don't know $\langle v_i h_j \rangle_{\text{model}}$
- Idea, approximate $\langle v_i h_j \rangle_{\text{model}}$ by Gibbs sampling

Training an RBM

Idea: estimate $\langle v_i h_j \rangle_{\text{model}}$ by running one step of Gibbs sampling from the data.

```
while True:
```

```
  # Collect data statistics
```

$\boldsymbol{v}_{\text{data}} \sim \text{data}$

$\boldsymbol{h}_{\text{data}} \sim \text{p}(\boldsymbol{h}|\boldsymbol{v}_{\text{data}}; (W, a, b))$

```
  # Collect model statistics
```

$\boldsymbol{v}_{\text{model}} \sim \text{p}(\boldsymbol{v}|\boldsymbol{h}_{\text{data}}; (W, a, b))$

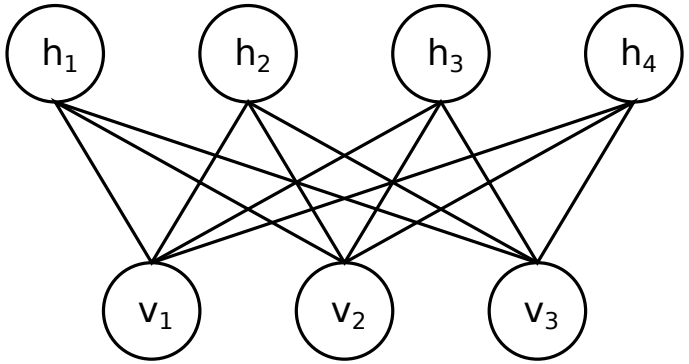
$\boldsymbol{h}_{\text{model}} \sim \text{p}(\boldsymbol{h}|\boldsymbol{v}_{\text{model}}; (W, a, b))$

```
  # Update model parameters
```

$W = W + \epsilon(\langle \boldsymbol{v}\boldsymbol{h}^\top \rangle_{\text{data}} - \langle \boldsymbol{v}\boldsymbol{h}^\top \rangle_{\text{model}})$

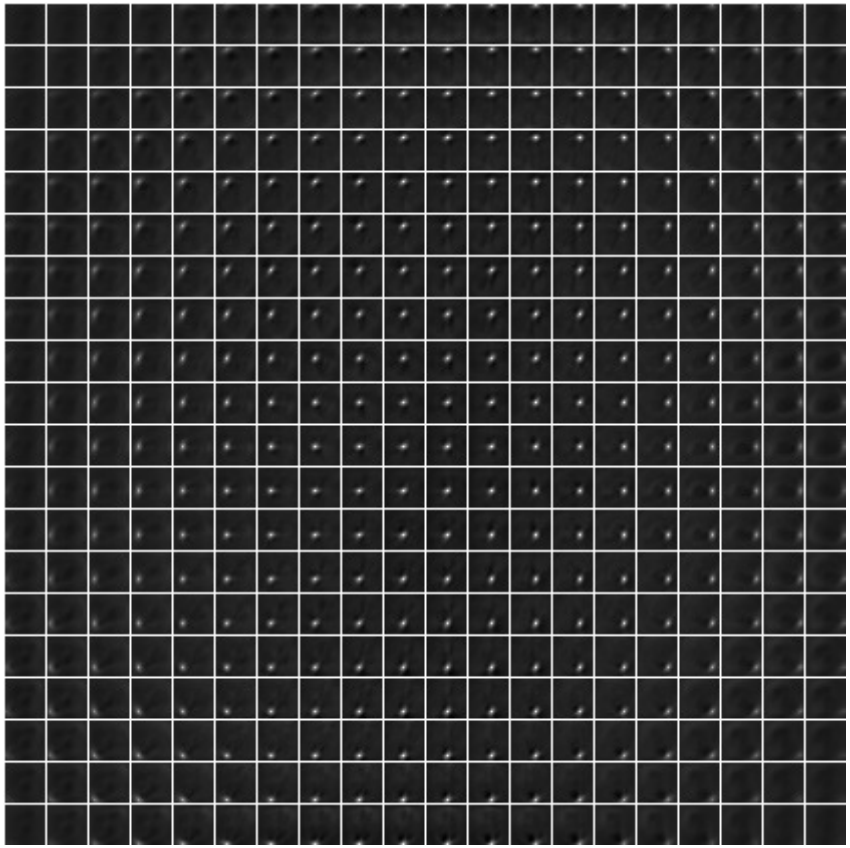
$\boldsymbol{a} = \boldsymbol{a} + \epsilon(\langle \boldsymbol{v} \rangle_{\text{data}} - \langle \boldsymbol{v} \rangle_{\text{model}})$

$\boldsymbol{b} = \boldsymbol{b} + \epsilon(\langle \boldsymbol{h} \rangle_{\text{data}} - \langle \boldsymbol{h} \rangle_{\text{model}})$

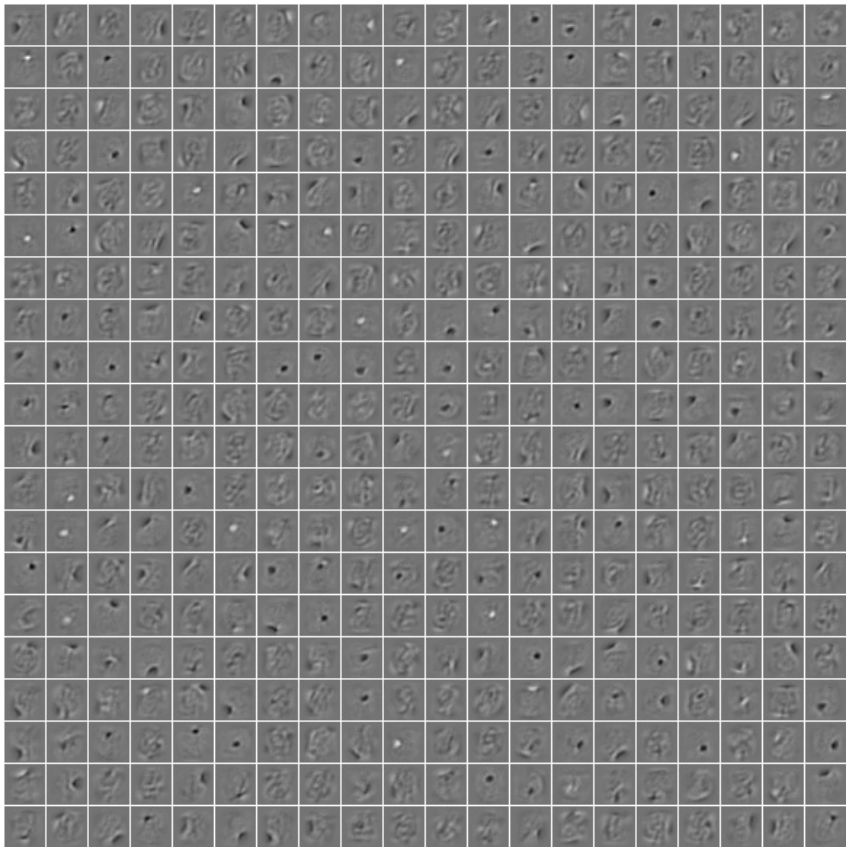


Training an RBM on handwritten digits

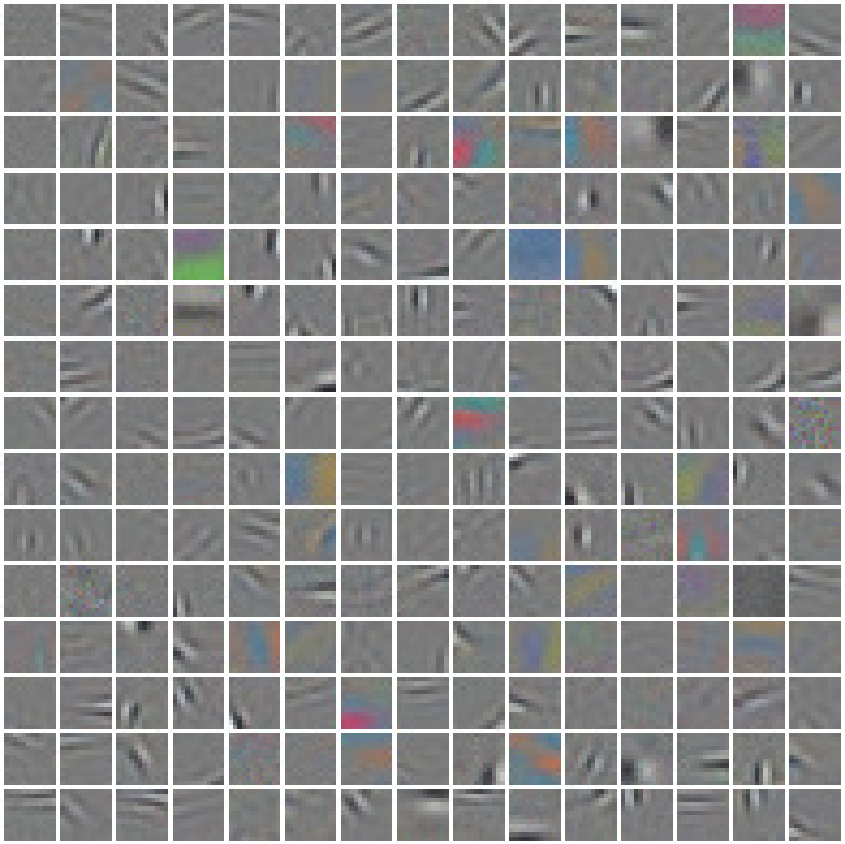
Without latent variables



With latent variables

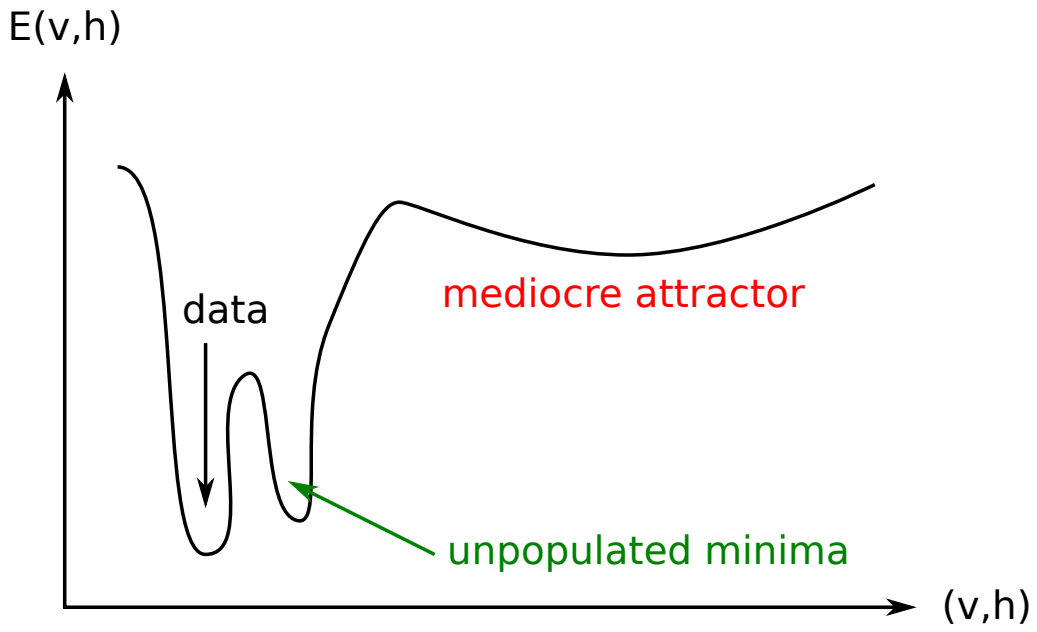


Training an RBM on natural image patches



- High resolution black and white filters
- Low-resolution color filters
- Similar to those observed in the monkey visual cortex

Obtaining unbiased samples from $p(\boldsymbol{v}, \boldsymbol{h})$



- Starting sampling from $(\boldsymbol{v}, \boldsymbol{h})_{\text{data}}$, it is difficult to move to other low-energy configurations of $(\boldsymbol{v}, \boldsymbol{h})$
- Starting sampling from $(\boldsymbol{v}, \boldsymbol{h}) \sim$ Bernouilli(p=0.5), it is difficult find any low-energy configuration of $(\boldsymbol{v}, \boldsymbol{h})$

Obtaining unbiased samples from $p(\boldsymbol{v}, \boldsymbol{h})$

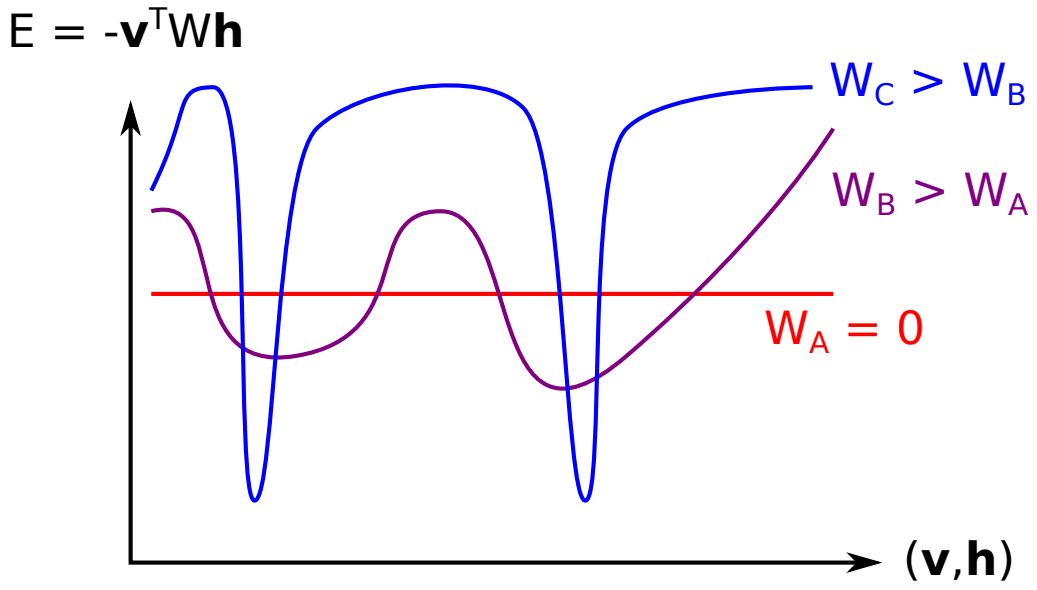
Contrastive divergence work well for some applications, but this is a hack. How can we approximate well $\langle v_i h_j \rangle_{\text{model}}$?

Simulated annealing

1. Start with a RBM with parameters $W, \boldsymbol{b}, \boldsymbol{a}$ set to zero.
2. Sample $(\boldsymbol{v}, \boldsymbol{h}) \sim \text{Bernouilli}(p = 0.5)$.
3. Update $(\boldsymbol{v}, \boldsymbol{h})$ by running k steps of Gibbs sampling while progressively raising the parameters $(W, \boldsymbol{b}, \boldsymbol{a})$ from zero to their original values.

If the process is slow enough, unbiased samples of $p(\boldsymbol{v}, \boldsymbol{h})$ are obtained.

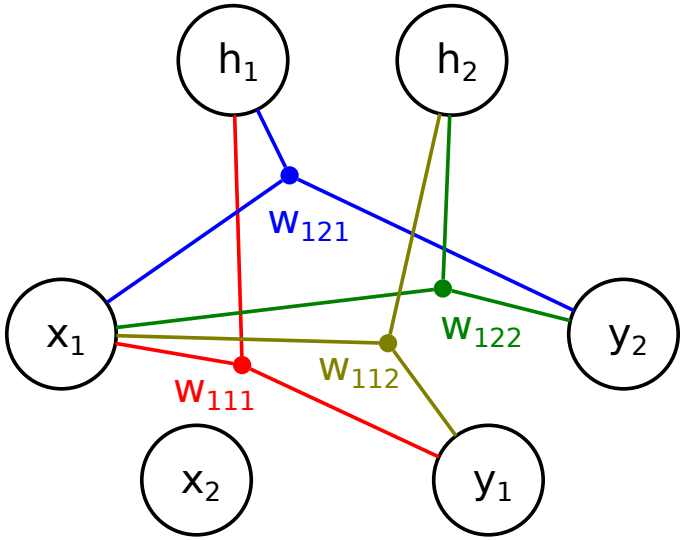
Obtaining unbiased samples from $p(\boldsymbol{v}, \boldsymbol{h})$



- When W is small, the RBM is "hot" and Gibbs sampling jumps randomly from one state to another.
- When W is large, the RBM is "cold" and Gibbs sampling can be stuck in the same state.
- By "cooling" progressively the RBM, we ensure that unbiased samples of $p(\boldsymbol{v}, \boldsymbol{h})$ are obtained.

Third-order restricted Boltzmann machines

- Visible units $\boldsymbol{x}$
- Visible units $\boldsymbol{y}$
- Hidden units $\boldsymbol{h}$
- Parameters $\theta = (W, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$



Energy function is defined as

$$\mathrm{E}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = -\sum_i x_i a_i - \sum_j y_j b_j - \sum_k h_k b_k - \sum_{ijk} x_i y_j h_k w_{ijk}$$

Third-order restricted Boltzmann machines

Energy function is defined as

$$\mathrm{E}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = -\sum_i x_i a_i - \sum_j y_j b_j - \sum_k h_k b_k - \sum_{ijk} x_i y_j h_k w_{ijk}$$

Conditional probabilities can be computed as:

$$p(x_i|\boldsymbol{y}, \boldsymbol{h}) = \sigma(\sum_{jk} y_j h_k w_{ijk} + a_i)$$

$$p(y_j|\boldsymbol{x}, \boldsymbol{h}) = \sigma(\sum_{ik} x_i h_k w_{ijk} + b_j)$$

$$p(h_k|\boldsymbol{x}, \boldsymbol{y}) = \sigma(\sum_{ij} x_i y_j w_{ijk} + c_k)$$

To reduce capacity, W can be factored: $w_{ijk} = \sum_f w^{\boldsymbol{x}}_{if} w^{\boldsymbol{y}}_{jf} w^{\boldsymbol{h}}_{kf}$

Third-order restricted Boltzmann machines

Key features:

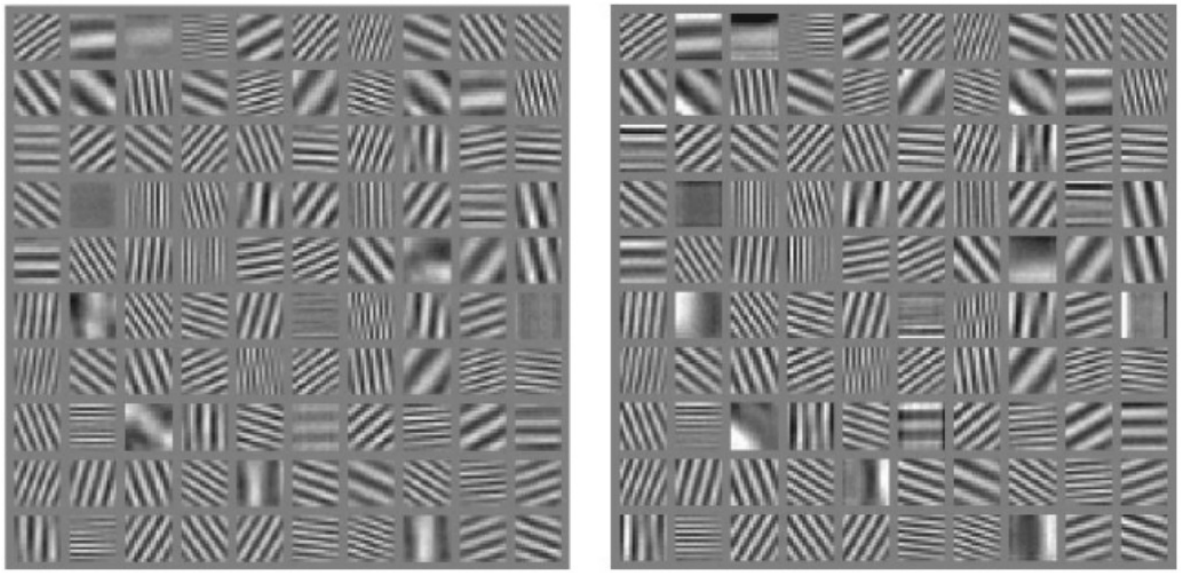
- This architecture models the interaction between $\boldsymbol{x}$ and $\boldsymbol{y}$ rather than $\boldsymbol{x}$ and $\boldsymbol{y}$ themselves.
- Hidden units $\boldsymbol{h}$ acts as a switch to determine whether $\boldsymbol{x}$ and $\boldsymbol{y}$ communicate.

Typical applications:

- Modeling system dynamics: $(\boldsymbol{x} = s(t)$ and $\boldsymbol{y} = s(t+1))$
- Modeling covariance matrices: $(\boldsymbol{x} = \boldsymbol{v}$ and $\boldsymbol{y} = \boldsymbol{v})$
- Transducers

Third-order restricted Boltzmann machines

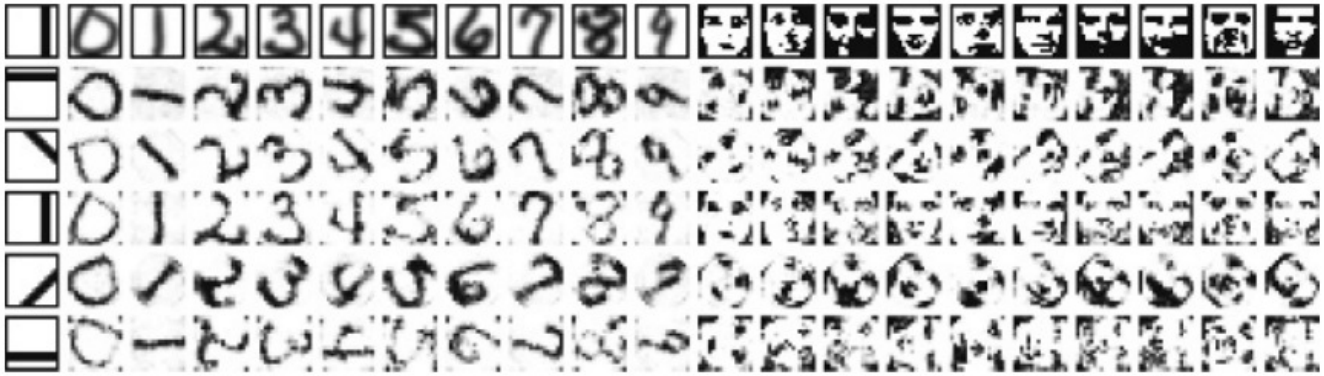
Modeling image transformations (Memisevic & Hinton, 2010)



When trained on image translations, the filters learned by RBM resemble the Fourier transform.

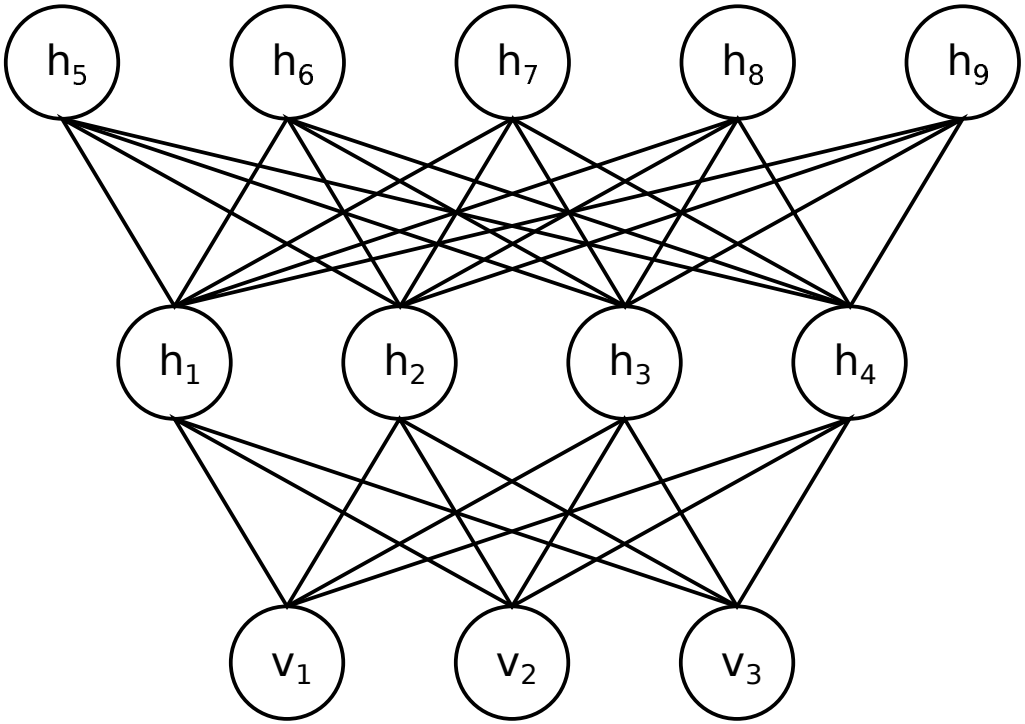
Third-order restricted Boltzmann machines

Modeling image transformations (Memisevic & Hinton, 2010)



Even if the RBM hasn't been trained on the faces, it can still rotate them.

Deep Boltzmann machines



- Theoretically higher representational power as RBM
- Difficult to optimize

Strategy:

- Initialize parameters with a greedy layer-wise pretraining

Does deep learning work?

MNIST (digit classification) - permutation invariant task

SVM	1.4 % error
Deep architecture	0.95% error

NORB (object categorization) - permutation invariant task

SVM	11.6 % error
Deep architecture	10.8% error

TIMIT (phone recognition)

GMM	33 % error
Deep architecture	20.5% error

Selected readings

Tutorial on RBMs

G. Hinton, A practical guide to training restricted Boltzmann machines, TR, 2010

A review on deep learning

Y. Bengio, Learning deep architectures for AI, Foundations and Trends in Machine Learning, 2009