UNIVERSITÀ PARTHENOPE
DiST
Dipartimento di Scienze e Tecnologie

Machine Learning (part II)

Deep Generative Models

Angelo Ciaramella

Structured probabilistic models

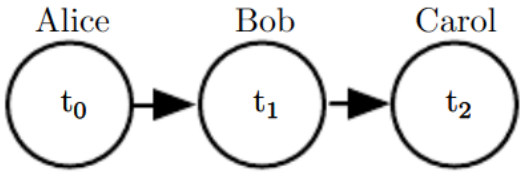
- Structured probabilistic models
  - the structure of the model is defined by a graph, these models are often also referred to as graphical models

- Challenge of Unstructured Modeling
  - Density estimation
    - Data generating distribution
  - Denoising
  - Missing value imputation
    - Distribution overs unobserved elements
  - Sampling
    - Speech synthesis

Direct models

- Directed graphical model
  - belief network or Bayesian network

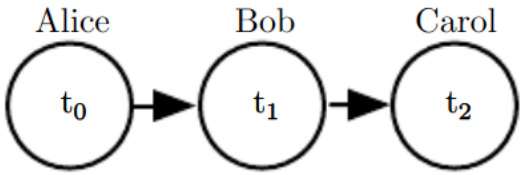
- Example - relay race
  - suppose we name Alice's finishing time t_0 , Bob's finishing time t_1, and Carol's finishing time t_2
    - Our estimate of t_2 depends directly on t1 but only indirectly on t_0



Direct models

- Directed graphical model
  - Genaral local conditional probability distributions

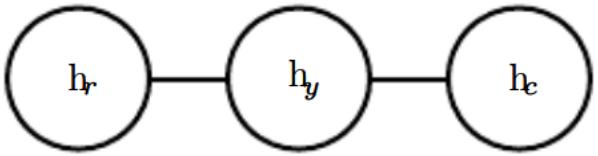
$$p(\mathbf{x}) = \Pi_i \, p(\mathrm{x}_i \mid Pa_{\mathcal{G}}(\mathrm{x}_i))$$



$$p(\mathrm{t}_0, \mathrm{t}_1, \mathrm{t}_2) = p(\mathrm{t}_0) p(\mathrm{t}_1 \mid \mathrm{t}_0) p(\mathrm{t}_2 \mid \mathrm{t}_1)$$

Undirect models

- Undirect models
  - Markov random fields (MRFs) or Markov networks
  - when the interactions seem to have no intrinsic direction, or to operate in both directions, it may be more appropriate to use an undirected model
- Example
  - whether or not you are sick, whether or not your coworker is sick, and whether or not your roommate is sick (coworker and roommate do not know each other)

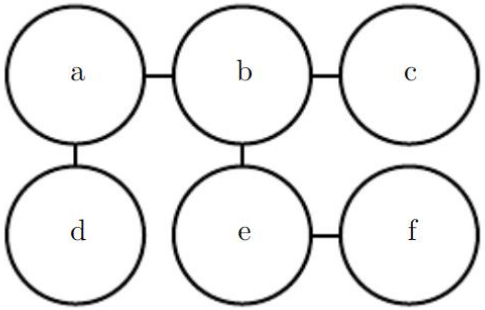


Undirect models

- Undirected graphical model
  - structured probabilistic model defined on an undirected graph
  - Clique potential
    - Factor $\phi(C)$
    - the affinity of the variables in that clique for being in each of their possible joint states

$$\tilde{p}(\mathbf{x}) = \Pi_{\mathcal{C}\in\mathcal{G}}\,\phi(\mathcal{C})$$

Unnormalized probability distribution



$$\frac{1}{Z}\,\phi_{\mathrm{a,b}}(\mathrm{a},\mathrm{b})\phi_{\mathrm{b,c}}(\mathrm{b},\mathrm{c})\phi_{\mathrm{a,d}}(\mathrm{a},\mathrm{d})\,\phi_{\mathrm{b,e}}(\mathrm{b},\mathrm{e})\phi_{\mathrm{e,f}}(\mathrm{e},\mathrm{f})$$

Partition function

- Unnormalized probability distribution
  - guaranteed to be non-negative everywhere, it is not guaranteed to sum or integrate to 1
  - valid probability distribution, we must use the corresponding normalized probability distribution

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x})$$

$$Z = \int \tilde{p}(\mathbf{x})d\mathbf{x}$$ intractable to compute

Z as a constant when the φ functions are held constant. Note that if the φ functions have parameters, then Z is a function of those parameters

Energy-based models

- Many interesting theoretical results about undirected models depend on the assumption that

$$\forall \mathbf{x}, \tilde{p}(\mathbf{x}) > 0$$

- A convenient way to enforce this condition is to use an Energy-Based Model (EBM) where

$$\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x}))$$

Boltzmann distribution

Energy function

Because exp(z) is positive for all z, this guarantees that no energy function will result in a probability of zero for any state x

Energy-based models

- Energy-based model is just a special kind of Markov network
  - the exponentiation makes each term in the energy function correspond to a factor for a different clique

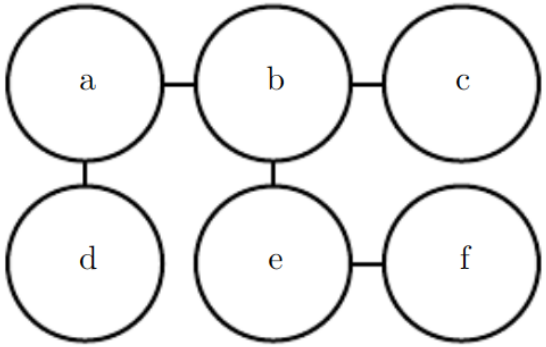


Figure 16.5: This graph implies that $E(\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{e},\mathrm{f})$ can be written as $E_{\mathrm{a,b}}(\mathrm{a},\mathrm{b}) + E_{\mathrm{b,c}}(\mathrm{b},\mathrm{c}) + E_{\mathrm{a,d}}(\mathrm{a},\mathrm{d}) + E_{\mathrm{b,e}}(\mathrm{b},\mathrm{e}) + E_{\mathrm{e,f}}(\mathrm{e},\mathrm{f})$ for an appropriate choice of the per-clique energy functions. Note that we can obtain the ϕ functions in Fig. 16.4 by setting each ϕ to the exponential of the corresponding negative energy, e.g., $\phi_{\mathrm{a,b}}(\mathrm{a},\mathrm{b}) = \exp(-E(\mathrm{a},\mathrm{b}))$.

Deep Belief Networks

- Deep belief networks (DBN)
  - first non-convolutional models to successfully admit training of deep architectures
  - Deep belief networks demonstrated that deep architectures can be successful
  - Generative models with several layers of latent variables
  - Latent variables are typically binary
  - Visible units may be binary or real
  - no intra-layer connections
    - Usually, every unit in each layer is connected to every unit in each neighboring layer
  - construct more sparsely connected DBNs
  - The connections between the top two layers are undirected
  - A DBN with only one hidden layer is a Restricted Boltzmann Machine (RBM)

Boltzmann Machines

- BM
  - connectionist approach to learning arbitrary probability distributions over binary vectors

- Energy function

$$\mathbf{x} \in \{0,1\}^d$$

$$P(\boldsymbol{x}) = \frac{\exp\left(-E(\boldsymbol{x})\right)}{Z}$$

joint probability distribution over the observed variables

$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{U} \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

Boltzmann Machines

- Powerful when
  - not all the variables are observed
  - latent variables act similarly to hidden units in MLP
  - Universal approximator of probability mass functions over discrete variables

- Energy function

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{v}^\top \boldsymbol{R}\boldsymbol{v} - \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{h} - \boldsymbol{h}^\top \boldsymbol{S}\boldsymbol{h} - \boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h}$$

- Learning
  - intractable partition function

Intractable partition functions

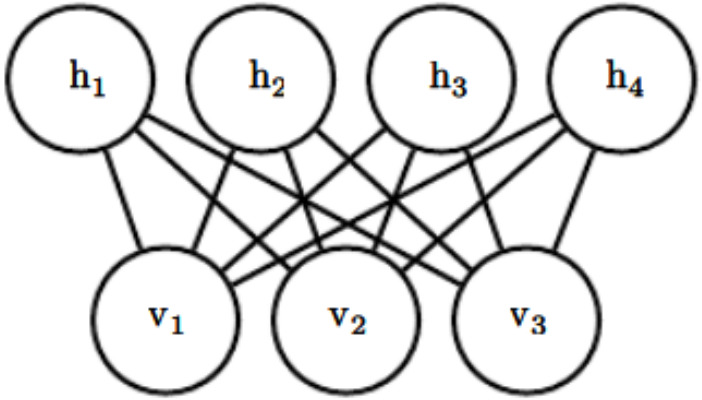
- Normalized probability distribution

$$p(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\tilde{p}(\mathbf{x};\boldsymbol{\theta})$$

- Techniques used for training and evaluating models
  - Log-Likelihood Gradient
  - Stochastic Maximum Likelihood
  - Markov Chain Monte-Carlo sampling
  - Contrastive Divergence (CD)
  - Pseudolikelihood
  - Score Matching and Ratio Matching
  - Noise-Contrastive Estimation
  - Annealed Importance Sampling
  - Bridge Sampling

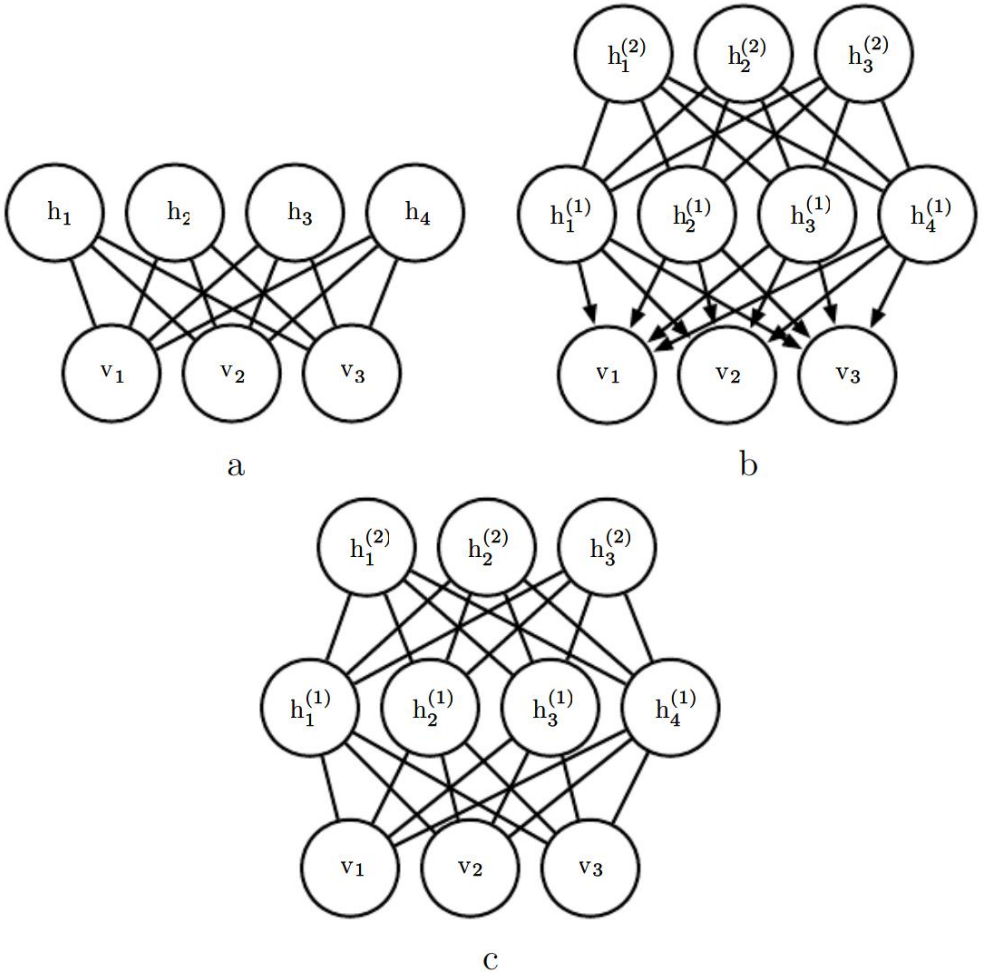
Restricted Boltzmann Machines

- Restricted Boltzmann Machines (RBM)
  - Energy based model (named Harmonium, 1986)
  - undirected probabilistic graphical model
  - a layer of observable variables and a single layer of latent variables
  - Restricted: No direct interactions between any two visible units or between any two hidden units



Restricted Boltzmann Machines

- Stacked Boltzmann Machines (deeper models)



RBM

- Joint probability distribution

$$P(\mathbf{v} = \boldsymbol{v}, \mathbf{h} = \boldsymbol{h}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h})\right)$$

- Energy function

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

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h}$$

RBM properties

- Structure

$$p(\mathbf{h} \mid \mathbf{v}) = \Pi_i p(\mathrm{h}_i \mid \mathbf{v})$$

$$p(\mathbf{v} \mid \mathbf{h}) = \Pi_i p(\mathrm{v}_i \mid \mathbf{h}).$$

$$P(\mathrm{h}_i = 1 \mid \mathbf{v}) = \sigma\left(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\right),$$
$$P(\mathrm{h}_i = 0 \mid \mathbf{v}) = 1 - \sigma\left(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\right)$$

Block Gibbs sampling

- Easy to take its derivatives

$$\frac{\partial}{\partial W_{i,j}} E(\mathbf{v}, \mathbf{h}) = -\mathrm{v}_i \mathrm{h}_j$$

RBM

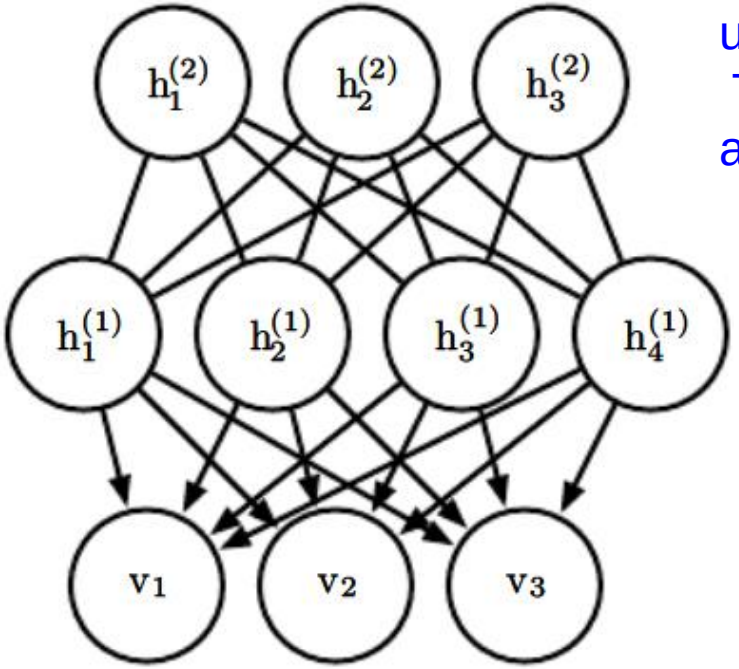
- Deriving the conditional distributions

$$
\begin{aligned}
P(\boldsymbol{h} \mid \boldsymbol{v}) &= \frac{P(\boldsymbol{h}, \boldsymbol{v})}{P(\boldsymbol{v})} \\
&= \frac{1}{P(\boldsymbol{v})} \frac{1}{Z} \exp\left\{ \boldsymbol{b}^\top \boldsymbol{v} + \boldsymbol{c}^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h} \right\} \\
&= \frac{1}{Z'} \exp\left\{ \boldsymbol{c}^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h} \right\} \\
&= \frac{1}{Z'} \exp\left\{ \sum_{j=1}^{n_h} c_j h_j + \sum_{j=1}^{n_h} \boldsymbol{v}^\top \boldsymbol{W}_{:,j} \boldsymbol{h}_j \right\} \\
&= \frac{1}{Z'} \prod_{j=1}^{n_h} \exp\left\{ c_j h_j + \boldsymbol{v}^\top \boldsymbol{W}_{:,j} \boldsymbol{h}_j \right\}
\end{aligned}
$$

Training RBM

- RBM training
  - training models that have intractable partition functions
    - CD
    - SML
    - Ratio matching
    - …

DBNs



The connections between the top two layers are undirected.
The connections between all other layers are directed

A deep belief network is a hybrid graphical model involving both directed and undirected connections

Deep belief networks demonstrated
that deep architectures can be successful

DBNs

- Probability distribution

$$P(\boldsymbol{h}^{(l)}, \boldsymbol{h}^{(l-1)}) \propto \exp\left(\boldsymbol{b}^{(l)\top}\boldsymbol{h}^{(l)} + \boldsymbol{b}^{(l-1)\top}\boldsymbol{h}^{(l-1)} + \boldsymbol{h}^{(l-1)\top}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)}\right),$$

$$P(h_i^{(k)} = 1 \mid \boldsymbol{h}^{(k+1)}) = \sigma\left(b_i^{(k)} + \boldsymbol{W}_{:,i}^{(k+1)\top}\boldsymbol{h}^{(k+1)}\right) \forall i, \forall k \in 1, \ldots, l-2,$$

$$P(v_i = 1 \mid \boldsymbol{h}^{(1)}) = \sigma\left(b_i^{(0)} + \boldsymbol{W}_{:,i}^{(1)\top}\boldsymbol{h}^{(1)}\right) \forall i.$$

$$\mathbf{v} \sim \mathcal{N}\left(\boldsymbol{v}; \boldsymbol{b}^{(0)} + \boldsymbol{W}^{(1)\top}\boldsymbol{h}^{(1)}, \boldsymbol{\beta}^{-1}\right)$$

Real-valued visible units

Training DBN

- Learning
  - Training an RBM to maximize by CD or SML

$$\mathbb{E}_{\mathbf{v} \sim p_{\text{data}}} \log p(\boldsymbol{v})$$

  - Second RBM maximize

$$\mathbb{E}_{\mathbf{v} \sim p_{\text{data}}} \mathbb{E}_{\mathbf{h}^{(1)} \sim p^{(1)}(\boldsymbol{h}^{(1)} | \boldsymbol{v})} \log p^{(2)}(\boldsymbol{h}^{(1)})$$

- procedure can be repeated indefinitely on many layers

DBF as generative model

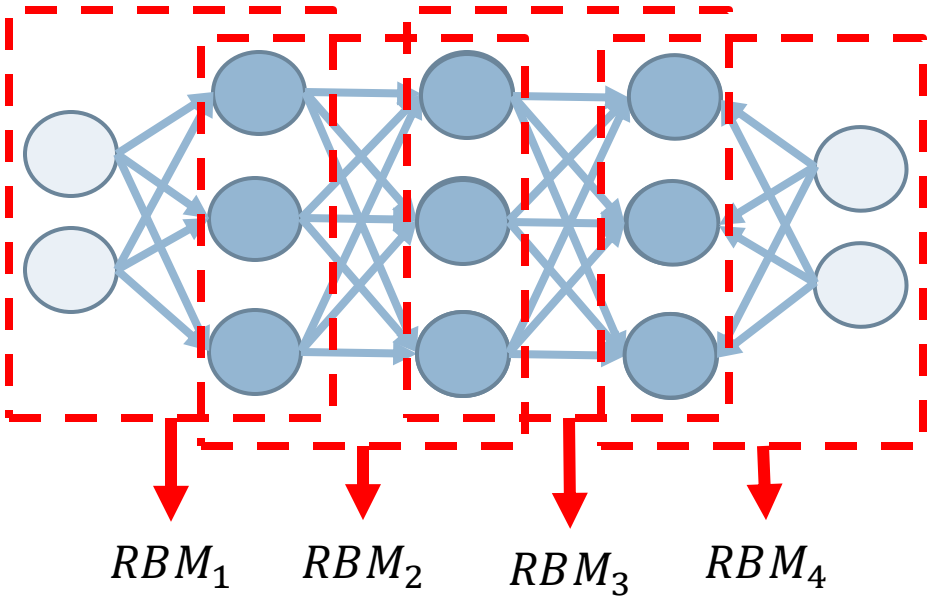
- Improve classification models
  - weights from the DBN and use them to define an MLP

$$\boldsymbol{h}^{(1)} = \sigma\left(b^{(1)} + \boldsymbol{v}^{\top}\boldsymbol{W}^{(1)}\right).$$

$$\boldsymbol{h}^{(l)} = \sigma\left(b_i^{(l)} + \boldsymbol{h}^{(l-1)\top}\boldsymbol{W}^{(l)}\right) \forall l \in 2, \ldots, m,$$

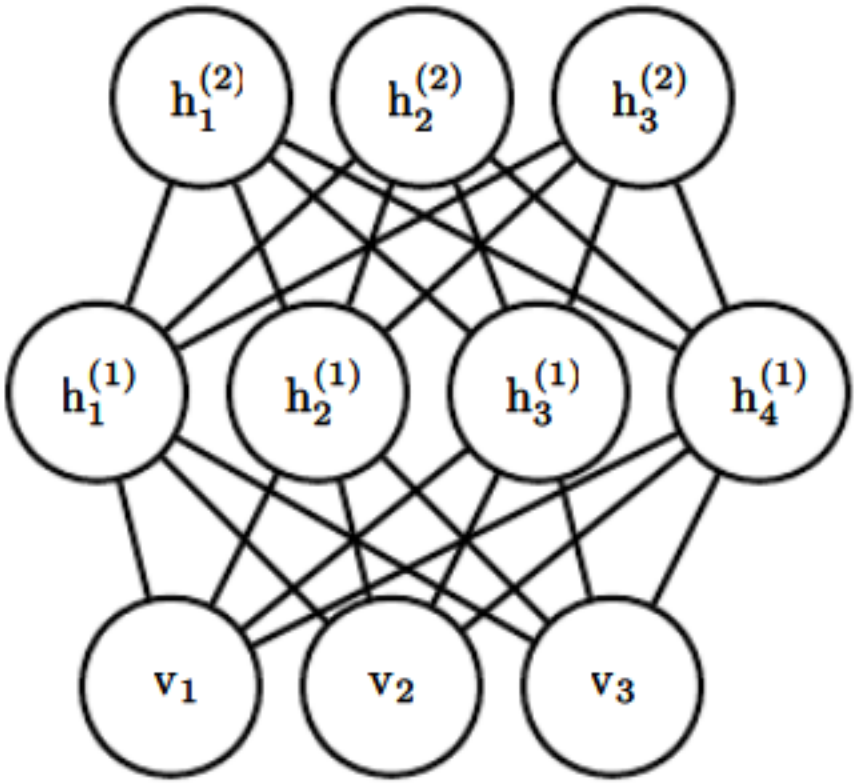
  - Training MLP for classification tasks (discriminative fine-tuning)

Deep Belief Networks



unsupervised, layer-wise, greedy pre-training

Deep Boltzmann Machines



entirely undirected model

Deep Boltzmann Machines

- joint probability

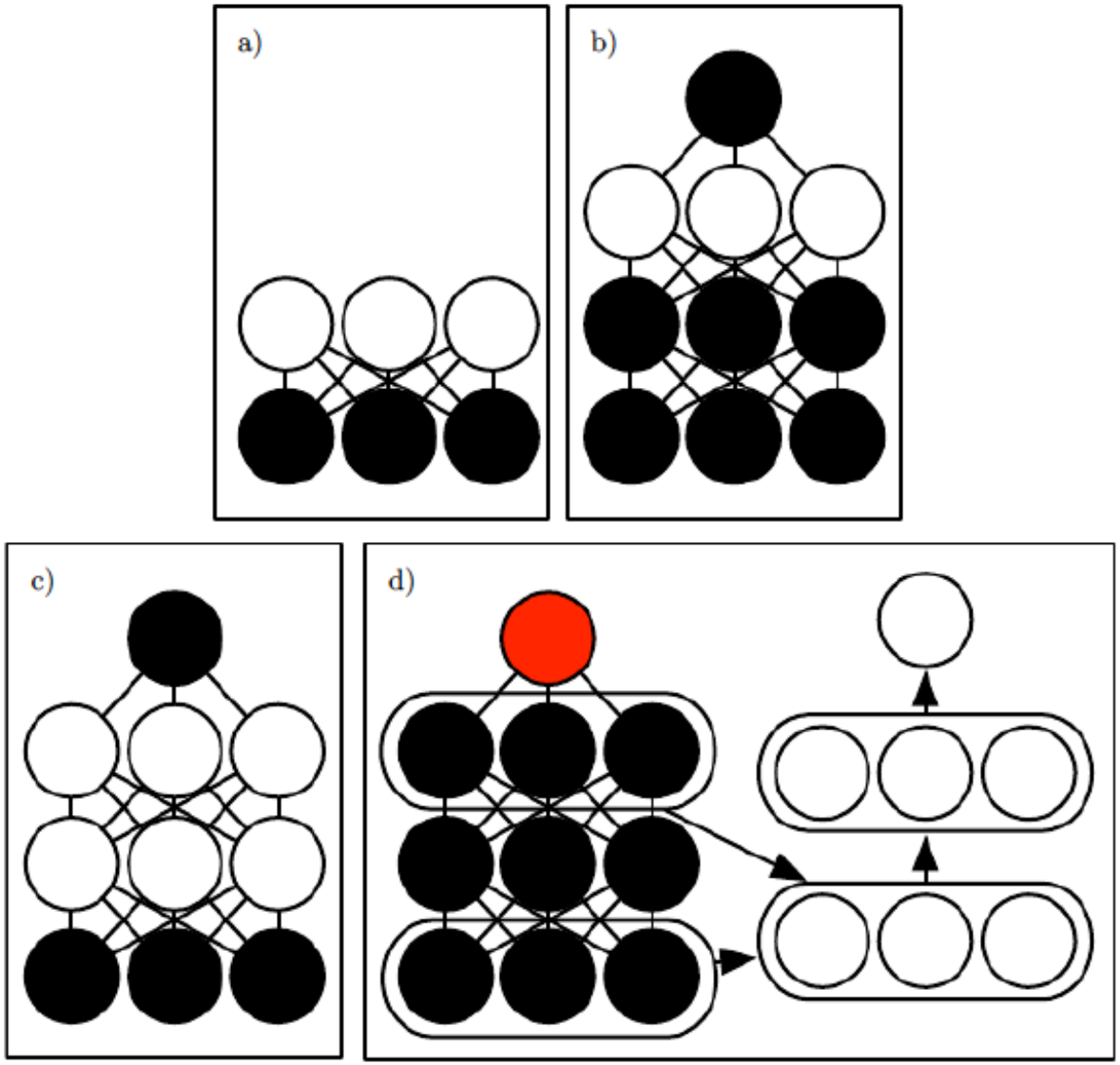
$$P\left(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}\right) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta})\right)$$

- Energy function

$$E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta}) = -\boldsymbol{v}^{\top} \boldsymbol{W}^{(1)} \boldsymbol{h}^{(1)} - \boldsymbol{h}^{(1)\top} \boldsymbol{W}^{(2)} \boldsymbol{h}^{(2)} - \boldsymbol{h}^{(2)\top} \boldsymbol{W}^{(3)} \boldsymbol{h}^{(3)}$$

- Learning
  - challenge of an intractable partition function
  - challenge of an intractable posterior distribution

Jointly Training DBM



Generative Adversarial Networks

- GANs
  - game theoretic scenario
    - generator network must compete against an adversary

$$\boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}^{(g)})$$

Generator network

$$d(\boldsymbol{x}; \boldsymbol{\theta}^{(d)})$$

Discriminator network

probability that x is a real training example rather than a fake sample drawn from the model

Generative Adversarial Networks

- Zero-sum game

$$v(\bar{\boldsymbol{\theta}}^{(g)}, \bar{\boldsymbol{\theta}}^{(d)})$$

payoff of the discriminator

- to maximize its own payoff

$$g^* = \arg\min_g \max_d v(g, d)$$

$$v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log d(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{model}}} \log (1 - d(\boldsymbol{x}))$$

This drives the discriminator to attempt to learn to correctly classify samples as real or fake. Simultaneously, the generator attempts to fool the classifier into believing its samples are real.

GANs

Images generated by GANs trained on the LSUN dataset