Restricted Boltzmann Machines

Sargur N. Srihari
srihari@cedar.buffalo.edu

RBMs or Harmonium

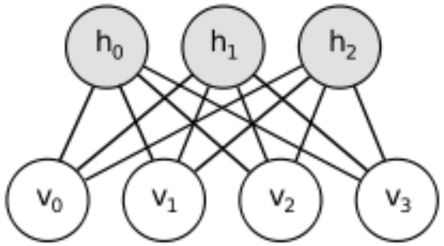
- RBMs are a quintessential example of how graphical models are used for deep learning
- RBM itself is not a deep model
- It has a single layer of latent units that may be used to learn a representation for the input
- RBMs can be used to build many deeper models

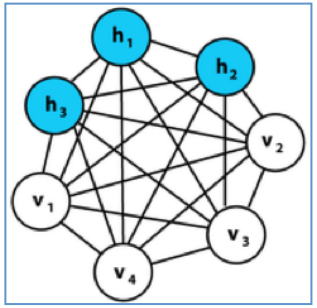
RBM Characteristics

- Units are organized into large groups called layers
- Connectivity between layers is described by a matrix
- Connectivity is relatively dense
- Allows efficient Gibbs sampling
- Learns latent variables whose semantics is not defined by the designer

Restricted Boltzmann Machine

- RBM is a special case of Boltzmann machines and Markov networks
- No visible-visible and hidden-hidden connections– Bipartite graph





General BM

- Used to learn features for input to neural networks in Deep Learning

Canonical RBM

- Energy-based model with binary visible / hidden units
- Its energy function is

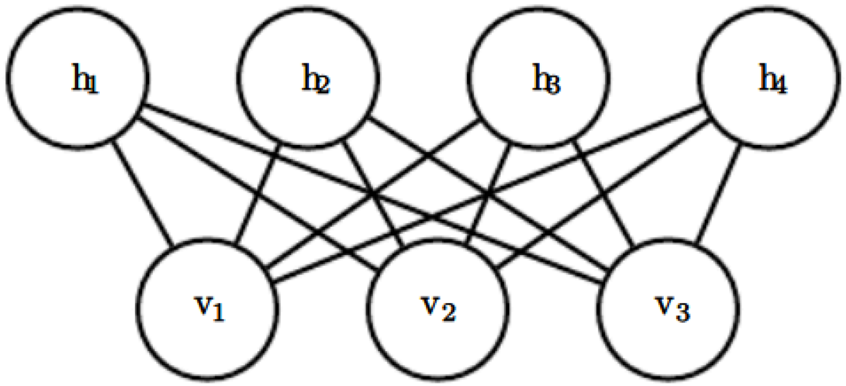
$$\mathrm{E}(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{h} - \boldsymbol{v}^{\mathrm{T}}\mathrm{W}\boldsymbol{h}$$

  – where $\boldsymbol{b}$, $\boldsymbol{c}$ and W are unconstrained, real-valued learnable parameters

- We can see that the model is divided into two groups of units $\boldsymbol{v}$ and $\boldsymbol{h}$ and the interaction between them is described by matrix W
- Model is graphically depicted next.

An RBM drawn as a Markov network

- The model is depicted graphically as



- No direct interactions between any two visible units or between any two hidden units
  - Hence the “restricted”, a general BM may arbitrary connections

Properties of RBMs

- Restrictions on RBM structure yields properties

  $p(\boldsymbol{h}|\boldsymbol{v}) = \prod_i p(h_i|\boldsymbol{v})$ and

  $p(\boldsymbol{v}|\boldsymbol{h}) = \prod_i p(v_i|\boldsymbol{h})$

- Individual conditionals are simple to compute
  - For binary RBM we obtain

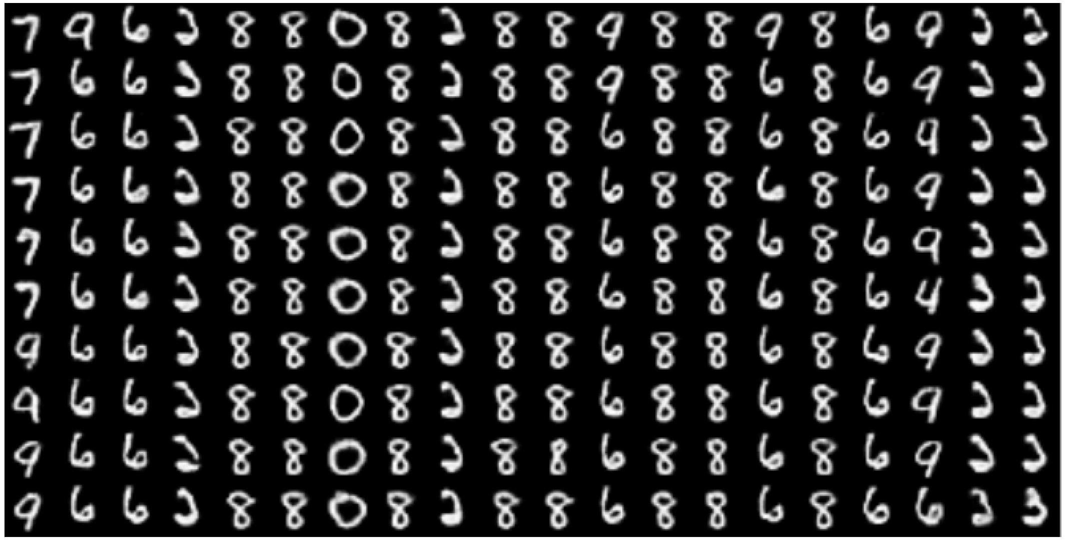
    $p(h_i = 1|\boldsymbol{v}) = \sigma\,(\boldsymbol{v}^{\mathrm{T}}\mathrm{W}_{:,i} + b_i)$

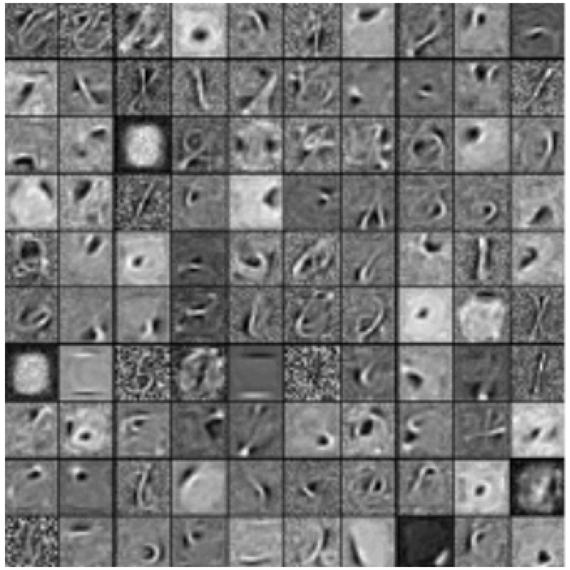
    $p(h_i = 0|\boldsymbol{v}) = 1\text{-}\,\sigma\,(\boldsymbol{v}^{\mathrm{T}}\mathrm{W}_{:,i} + b_i)$

  - Together these properties allow for *block Gibbs sampling* which alternate between sampling all $\boldsymbol{h}$ simultaneously and all $\boldsymbol{v}$ simultaneously
    - Shown next

Samples from a trained RBM

- Samples from a trained RBM on MNIST drawn using Gibbs sampling





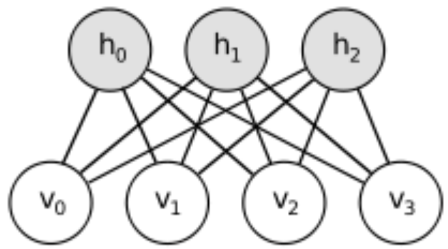
Corresponding weight vectors

  - Each column is a separate Gibbs process
  - Each row represents the output of another 1000 steps of Gibbs sampling
    - Successive samples are highly correlated

Derivatives of Energy Function

- Energy function: $\mathrm{E}(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{h} - \boldsymbol{v}^{\mathrm{T}}\mathrm{W}\boldsymbol{h}$
    - where $\boldsymbol{b}$, $\boldsymbol{c}$ and W are unconstrained, real-valued learnable parameters
- Since the energy function is a linear function of its parameters, it is easy to take derivatives
    – E.g., $\boxed{\dfrac{\partial}{\partial W_{i,j}} E(\boldsymbol{v},\boldsymbol{h}) = -v_i h_j}$
- These two properties, efficient Gibbs sampling and efficient derivatives make training convenient
    - Undirected models can be trained by computing such derivatives applied to samples from the model

Energy function of a RBM



- Energy function

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

  – where $\boldsymbol{b}$, $\boldsymbol{c}$ and W are unconstrained, real-valued learnable parameters

- Defining free energy as

$$\mathcal{F}(v) = -b'v - \sum_i \log \sum_{h_i} e^{h_i(c_i + W_i v)}.$$

- Due to structure of RBM

$$p(\boldsymbol{h}|\boldsymbol{v}) = \Pi_i\, p(h_i|\boldsymbol{v}) \text{ and}$$
$$p(\boldsymbol{v}|\boldsymbol{h}) = \Pi_i\, p(v_i|\boldsymbol{h})$$

RBM with binary units

- Using $v_j, h_i \in \{0,1\}$

$$p(h_i=1|\boldsymbol{v})=\sigma\,(\boldsymbol{v}^{\mathrm{T}}\mathrm{W}_{:,i}+b_i)$$
$$p(h_{\mathrm{i}}=0|\boldsymbol{v})=1\text{-}\,\sigma\,(\boldsymbol{v}^{\mathrm{T}}\mathrm{W}_{:,i}+b_i)$$

- Free energy simplifies to

$$\mathcal{F}(v) = -b'v - \sum_i \log(1 + e^{(c_i + W_i v)}).$$

- Update equations

$$-\frac{\partial \log p(v)}{\partial W_{ij}} = E_v[p(h_i|v) \cdot v_j] - v_j^{(i)} \cdot sigm(W_i \cdot v^{(i)} + c_i)$$
$$-\frac{\partial \log p(v)}{\partial c_i} = E_v[p(h_i|v)] - sigm(W_i \cdot v^{(i)})$$
$$-\frac{\partial \log p(v)}{\partial b_j} = E_v[p(v_j|h)] - v_j^{(i)}$$

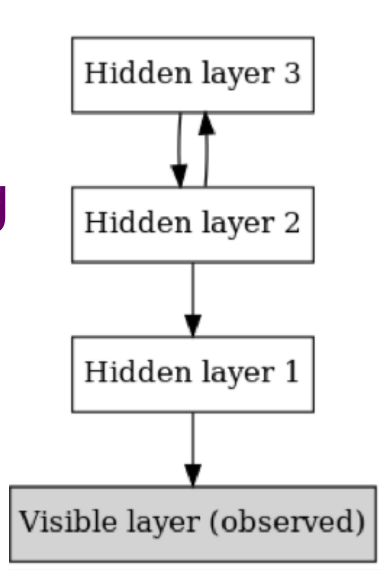
Training RBMs

- Contrastive Divergence
- A method to overcome exponential complexity in dealing with the partition function

Deep Belief Networks (DBNs)

- Consist of several layers of RBMs
  - Stacking RBMs
    - Fine tuning resulting deep network using gradient descent and back-propagation
- DBNs are Generative Models
  - Provide estimates of both

    $p(x \mid C_k)$ and $p(C_k \mid x)$
  - Conventional neural networks are discriminative
    - Directly estimate $p(C_k \mid x)$



Training several layers of RBMs

- Let X be a matrix of input feature vectors

1. Train an RBM on X to obtain weight matrix $\mathbf{W}$
   - Between lower two layers (input and hidden)
2. Transform X by RBM to produce new data X'
   - by sampling or by computing mean activation of hidden units
3. Repeat procedure with $X \leftarrow X'$ for next layer pair
   - Until top two layers of network are reached (output and hidden)