Bolzmann Machine

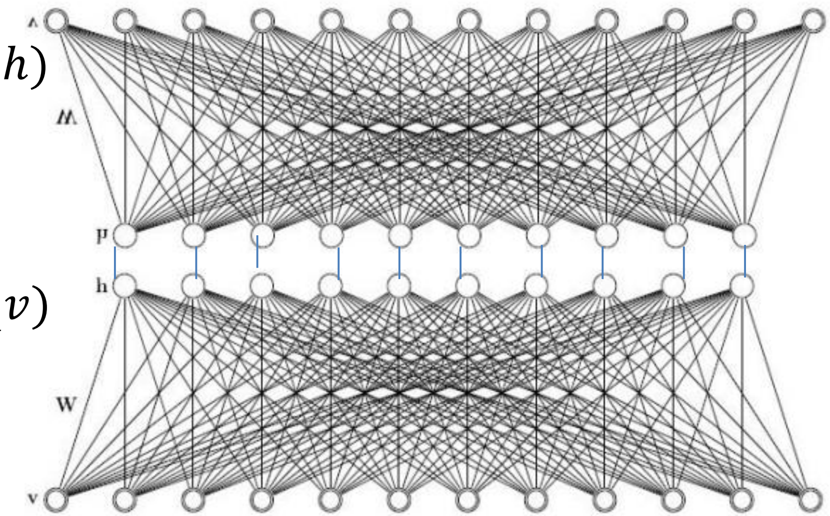
- Unsupervised Modelling of Binary Data
- What is Boltzmann Machine ?
- Restricted Boltzmann Machine (RBM)
- RBM Learning
- Contrast Divergence (CD)
- Example

Unsupervised Modelling of Binary Data

1 0 0 1 1 0 0 1 0 0 0 1

$o = \sigma(W_2 h)$

$h = \sigma(W_1 v)$

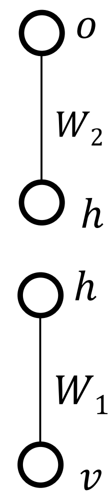


1 0 0 1 1 0 0 1 0 0 0 1

0 1 0 1 0 0 1 1 0 1 0 1

....

If no desired outputs ?



Modeling binary data

- Given a training set of binary vectors, fit a model that will assign a probability to other binary vectors

Name	Harry Potter	Avatar	LOTR3	Gladiator	Titanic	Glitter
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Prefer SF/fantasy

Prefer Oscar winner

$$p(x) = \prod_j \left(\boxed{x_j p_j} + \boxed{(1 - x_j)(1 - p_j)} \right)$$

If component j of vector x is on

If component j of vector x is off

Modeling binary data

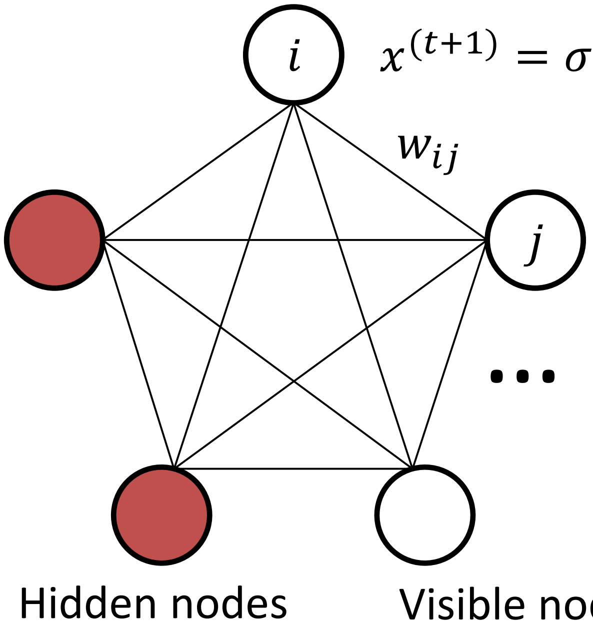
- Modelling with Boltzmann Machine

Name	Harry Potter	Avatar	LOTR3	Gladiator	Titanic	*Glitter*
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Prefer SF/fantasy

Prefer Oscar winner

$x^{(t+1)} = \sigma(Wx^{(t)})$



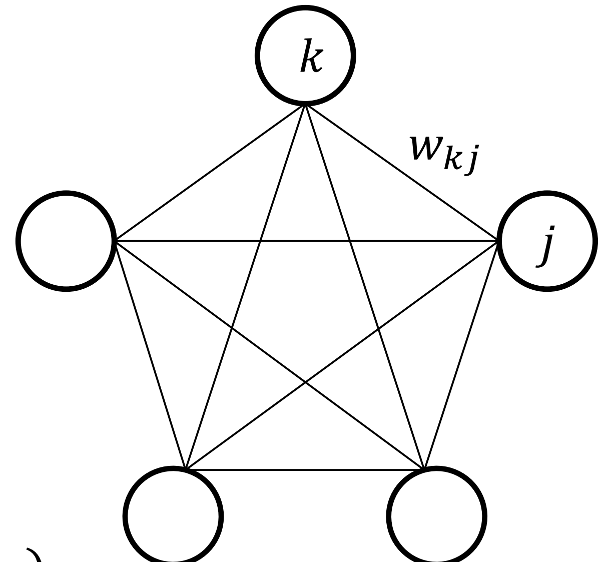
- w_{ij} represents a correlation between nodes
- $p(v) = \sum_h p(h)p(v|h)$

Boltzmann Machine

- Probability distribution on binary vectors x

$$x^{(t+1)} = \sigma(Wx^{(t)})$$

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



$$E(x) = -\frac{1}{2}x^T W x - \theta^T x$$

$$= -\sum_{k<j} x_k w_{kj} x_j - \sum_k \theta_k x_k$$

  - From the entropy maximization

$$\max_{P(x)} - \sum_x P(x) \ln P(x)$$

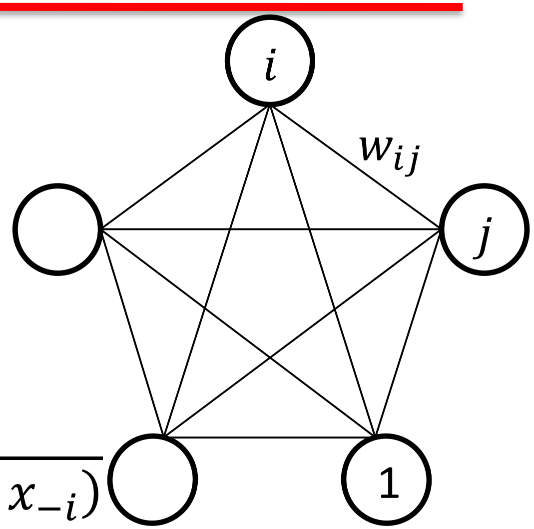
$$s.t \sum_x P(x) = 1, \alpha = \sum_x P(x) E(x)$$

  - Z is the partition function that ensures $\sum_x P(x) = 1$

$$Z = \sum_x \exp\left(-E(x)\right)$$

Boltzmann Machine

$$x^{(t+1)} = \sigma(Wx^{(t)})$$



- Probability distribution on binary vectors x

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

- $E(x) = -\sum_{k<j} x_k w_{kj} x_j - \sum_k \theta_k x_k$

- Gibbs Sampling

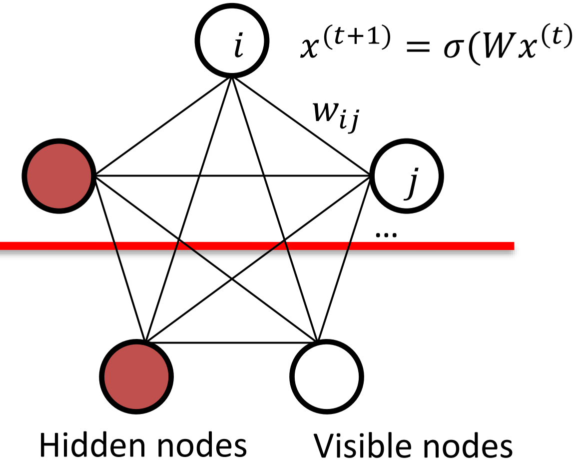
$$P(x_i = 1 | x_{-i}) = \frac{P(x_i = 1,\ x_{-i})}{P(x_i = 1,\ x_{-i}) + P(x_i = 0,\ x_{-i})}$$

$$= \frac{exp\big(-E(x_i = 1,\ x_{-i})\big)}{exp\big(-E(x_i = 1,\ x_{-i})\big) + exp\big(-E(x_i = 0,\ x_{-i})\big)}$$

$$= \frac{1}{1 + exp\big(-E(x_i = 0,\ x_{-i}) + E(x_i = 1,\ x_{-i})\big)}$$

$$= \frac{1}{1 + \exp(-\sum_{j \neq i} w_{ij} x_j - \theta_i)} = \sigma\Big(\sum_{j \neq i} w_{ij} x_j + \theta_i\Big)$$

Restricted Boltzmann Machine



- Variant of Boltzmann Machine
- Restrict the connectivity to make **learning easier**
  - There is a hidden layer and visible layer
  - No hidden-to-hidden or visible-to-visible connections
  - Hidden units extends the class of distributions that can be modeled

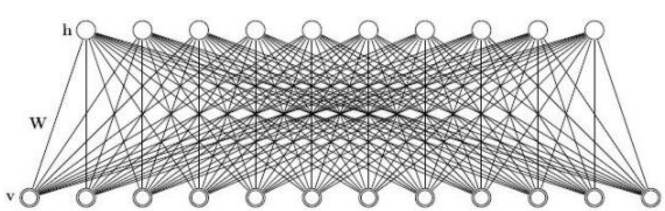
Bias of RBM

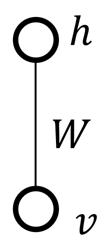
- Energy function

$$E(v,h) = -\sum_{\forall i,j} v_i w_{ij} h_j - \sum_i b_i^v v_i - \sum_j b_j^h h_j = -v^T W h - v^T b^v - h^T b^h$$

  - Vectors h, v are of dimension $J \times 1$ and $I \times 1$
  - W is of dimension $I \times J$

- $p(v) = \sum_h p(h)p(v|h)$





$h = \sigma(Wv)$

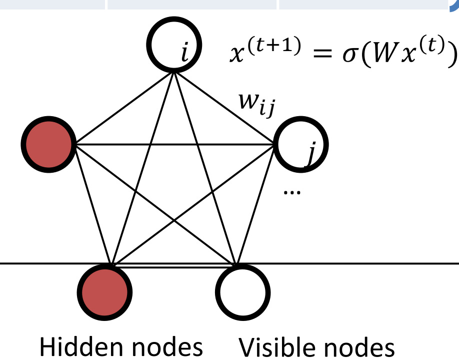
Modeling binary data

- Given a training set of binary vectors, fit a model that will assign a probability to other binary vectors

Name	Harry Potter	Avatar	LOTR3	Gladiator	Titanic	Glitter
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Prefer SF/fantasy

Prefer Oscar winner



Restricted Boltzmann Machine

- Marginal distribution $P(v)$

$$P(v) = \sum_h P(h)P(v|h) = \sum_h P(v,h) = \frac{\sum_h \exp\big(-E(v,h)\big)}{Z}$$

  - $P(v,h)$ is a Boltzmann distribution with energy function $E(v,h)$

  - And $P(v)$ is a Boltzmann distribution with a *energy* $F(v)$

$$P(v) = \frac{\exp\big(-F(v)\big)}{Z}$$

$$F(v) = -\ln \sum_h \exp\big(-E(v,h)\big)$$

  - the energy $F(v)$ cannot be represented as a quadratic form in v (Why?)

RBM Learning

- Maximize the product of probabilities assigned to training set V

$$\arg\max_{W} \prod_{v \in \mathrm{V}} P(v)$$

- Or equivalently, maximize the sum of log probability of V:

$$\arg\max_{W} \sum_{v \in \mathrm{V}} \ln P(v)$$

- The model is updated after each training token or in batch mode

$$w_{ij} \leftarrow w_{ij} + \alpha \frac{\partial \ln P(v)}{\partial w_{ij}} \Big|_{v=v^1}$$

RBM Learning

$$P(v) = \frac{\exp(-F(v))}{Z}$$

$$F(v) = -\ln\sum_h \exp(-E(v,h))$$

- Stochastic gradient ascent
  - Calculate the gradient of the log likelihood, given a training token v^1

$$\left.\frac{\partial \ln P(v)}{\partial w_{ij}}\right|_{v=v^1} = -\left.\frac{\partial F(v)}{\partial w_{ij}}\right|_{v=v^1} - \frac{\partial \ln Z}{\partial w_{ij}}$$

$$= v_i^1 h_j^1 - \frac{\partial}{\partial w_{ij}} \ln\sum_v \exp(-F(v))$$

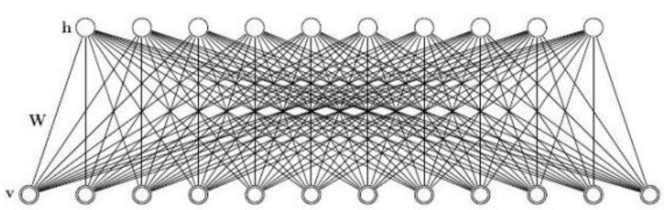
$$= v_i^1 h_j^1 - \frac{1}{\sum_v \exp(-F(v))}\sum_v \exp(-F(v))\frac{\partial F(v)}{\partial w_{ij}}$$

$$= v_i^1 h_j^1 - \frac{1}{Z}\sum_v \exp(-F(v)) v_i h_j$$

$$= v_i^1 h_j^1 - \textstyle\sum_v P(v) v_i h_j$$

Expectation of $v_i h_j$

$$= v_i^1 h_j^1 - \langle v_i h_j \rangle_{model}$$



RBM Learning

- Stochastic gradient ascent

$$F(v) = -\ln \sum_h \exp\big(-E(v,h)\big)$$

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

$$\frac{\partial F(v)}{\partial w_{ij}} = -\frac{\partial}{\partial w_{ij}} \ln \sum_h \exp\big(-E(v,h)\big)$$

$$= -\frac{1}{\cancel{\sum_h} \exp\big(-E(v,h)\big)} \cancel{\sum_h} \exp\big(-E(v,h)\big) \left(-\frac{\partial E(v,h)}{\partial w_{ij}}\right)$$

$$= -v_i h_j \qquad \text{for fixed } v, h$$

RBM Learning

$$\boxed{\left.\frac{\partial \ln P(v)}{\partial w_{ij}}\right|_{v=v^1} = v_i^1 h_j^1 - \langle v_i h_j \rangle_{model}}$$

- If there are K iid training tokens $v^1, \ldots, v^K$

$$\frac{\partial}{\partial w_{ij}} \sum_k \ln P(v^k) = \sum_k \frac{\partial \ln P(v^k)}{\partial w_{ij}}$$
$$= \left(v_i^1 h_j^1 + \cdots + v_i^K h_j^K - K \langle v_i h_j \rangle_{model}\right)$$

  - So that…

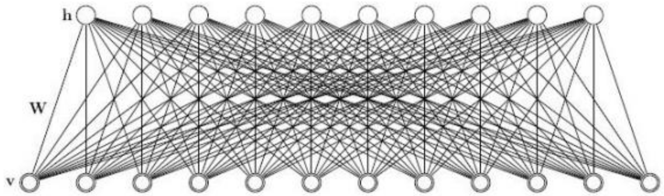
$$\frac{\partial}{\partial w_{ij}} \mathbb{E}_v[\ln P(v)] \approx \frac{\partial}{\partial w_{ij}} \frac{1}{K} \sum_k \ln P(v^k) = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

Data statistics

Model statistics

: unknown

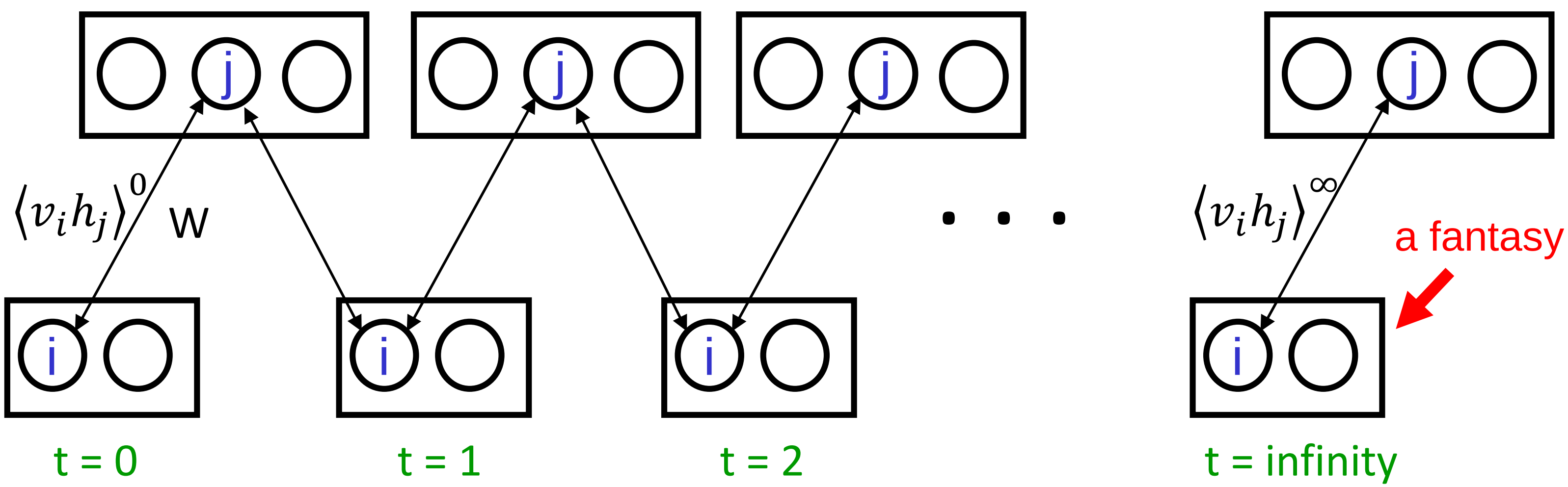
- $\Delta w_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$



Model statistics

- $\langle v_i h_j \rangle_{model}$ can be estimated by using any MCMC algorithm
  - But nobody knows t_{conv} which indicates the step at which $\langle v_i h_j \rangle$ converges

$$h_j \sim \sigma(w_j^T v + c_j)$$



$$v_i \sim \sigma(w_i^T h + c_i)$$

Model statistics

- Contrast Divergence (CD) [Bengio, et al.]: Starting at the given training token $v^{(1)}, h^{(1)}$, run the Markov chain for n steps:
  - $v^{(1)}, h^{(1)} \rightarrow \cdots \rightarrow v^{(n+1)}, h^{(n+1)}$
  - With the edge weight $[w_{ij}]$

- And we can approximate

$$\left.\frac{\partial \ln P(v)}{\partial w_{ij}}\right|_{v=v^1} \approx v_i^{(1)} h_j^{(1)} - v_i^{(n+1)} h_j^{(n+1)}$$

CD-n

- **CD-1** $\rightarrow$ *weight change* $\rightarrow$ **CD-3** $\rightarrow \ldots \rightarrow$ **CD-5** $\rightarrow \ldots \rightarrow$ **CD-7** $\ldots$ **CD-9**

Example of RBM

- Train the RBM using following data (with CD-1)
  - 6 visible units (each movies) with 2 hidden units

Name	Harry Potter	Avatar	LOTR3	Gladiator	Titanic	Glitter
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Prefer SF/fantasy

Prefer Oscar winner

Example of RBM

- And… the network is trained by the following weights:
  - $W = \begin{bmatrix} 4.97 & 2.27 & 4.11 & -4.01 & -5.60 & -2.92 \\ -7.09 & -5.18 & 2.52 & 6.75 & 3.25 & -2.82 \end{bmatrix}$

Name	Harry Potter	Avatar	LOTR3	Gladiator	Titanic	Glitter
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Prefer SF/fantasy

Prefer Oscar winner

  - The **first hidden unit** seems to correspond to the **SF/fantasy** , and the **second hidden unit** seems to correspond to the **Oscar winners** movies
  - If the RBM is presented to a new user, George, who has $[0,0,0,1,1,0]$ as his preferences, then It turns the second hidden unit on

Persistent CD

- A set of samples $v^1, \ldots v^K$ is drawn(observed) from the model distribution
  - The set is maintained and updated whenever the model is updated
  - K Markov chains are run in parallel and, on every update, several steps of Gibbs sampling are performed in each chain
  - The model statistics are derived by averaging over the samples:

$$\langle v_i h_j \rangle_{model} = \frac{1}{K} \sum_k v_i^{k,(n+1)} \, h_j^{k,(n+1)}$$

- Persistent CD generally works better than CD

Interim Summary

- Boltzmann machines try to model a realistic brain learning mechanism (unsupervised model).
- Boltzmann machines and Restricted Boltzmann machines are based on the energy model
- Undirected Graph model such as Markov random field
- The RBM is the simple type of Boltzmann machine, and it can be easily learned
  - We use the Contrastive Divergence (CD) to train the RBM
- Persistent Contrastive Divergence is the improved version of CD, and it lesson the problem that CD does not guarantee the fast convergence