Restricted Boltzmann Machine (RBM) and Deep Belief Network (DBN)

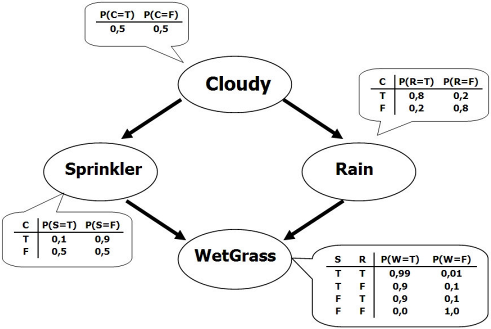
Zhen Li

Department of Mathematics, HKUST

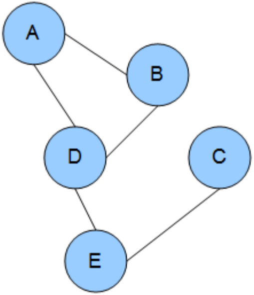
October 28, 2017

Probabilistic Graphical Models

A (probabilistic) graphical model (PGM) is a probabilistic model for which a graph is used to express **dependences (edges)** between **random variables (nodes/units)**.

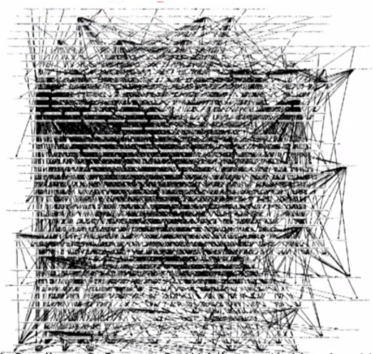


Bayesian network



Markov random field

Figure: Two basic types of PGMs. (copied from internet)



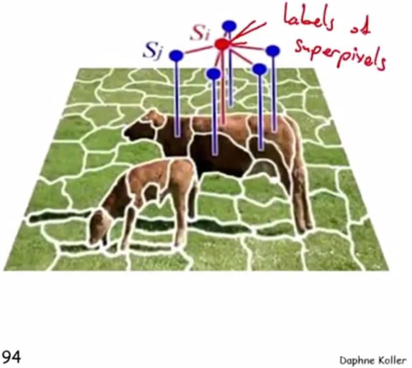


Figure: Applications of PGM. Left: Medical diagnosis system. Right: Image segmentation. (copied from internet)

Boltzmann Machine

A Boltzmann machine is a special type of graphical model. It has two types of units: **visible units** $\{v_i\}_{i=1,m}$ and **hidden units** $\{h_j\}_{j=1,n}$. Their values are called **states**. Each edge is assigned with a real number $\{W_{ij}\}_{i=1,m;j=1,n}$ called **weight**.

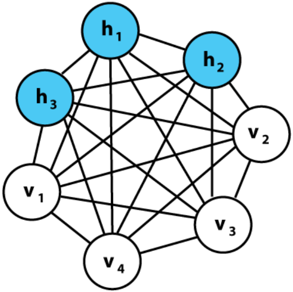


Figure: Boltzmann Machine. (copied from internet)

Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) is a special type of Boltzmann Machine for which the graph is organized in **two layers**: visible layer and hidden layer. There is **no intra-layer connection**.

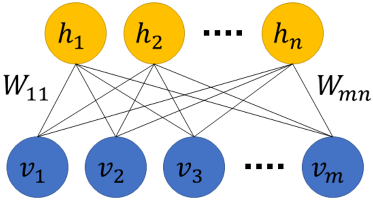


Figure: Restricted Boltzmann Machine.

Question: Why is RBM worthy to know?

- Connection with neural networks. It is an important pretraining method for DNN.
- Historical importance: It initializes the bloom of deep learning (Hinton and Salakhutdinov, 2006).
- Future perspective: It is an important unsupervised learning method.
- Beautiful structure and interpretation, rooted in statistical physics.

We denote visible states by row vector $\mathbf{v} = (v_i)_{i=1,m}$, hidden states by $\mathbf{h} = (h_j)_{j=1,n}$, and weights by matrix $\mathbf{W} = (W_{ij})_{m \times n}$.

The joint distribution of all units is the Boltzmann distribution

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}, \tag{1}$$

where E is the **energy** of the RBM. The normalization coefficient

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{2}$$

is called **partition function**. It is a function of parameters (such as weights $\mathbf{W}$). The sum is taken over all possible visible and hidden states.

The energy of an RBM can be defined in different ways. For example,

- If all the units take binary states, then we usually take

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \mathbf{b}\mathbf{h}^{\mathrm{T}} - \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}. \tag{3}$$

- If the visible units take binary states, but the hidden units take real-valued states, then we can take

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{a}\mathbf{v}^{\mathrm{T}} + \frac{\|\mathbf{h} - \mathbf{b}\|^2}{2} - \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}. \tag{4}$$

Here $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^n$ are called **biases** of visible and hidden units, respectively. They are also parameters of the RBM.

Given the energy function, we can compute the conditional distributions $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$.

For binary states, with energy defined by (3), we have

$$\begin{aligned} P(\mathbf{v}|\mathbf{h}) &\propto P(\mathbf{v},\mathbf{h}) && (5)\\ &\propto \exp\left(\mathbf{a}\mathbf{v}^{\mathrm{T}} + \mathbf{b}\mathbf{h}^{\mathrm{T}} + \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}\right) && (6)\\ &\propto \prod_{i=1}^{m} \exp\left(a_i v_i + v_i \sum_{j=1}^{n} W_{ij} h_j\right). && (7) \end{aligned}$$

On the other hand, the visible units are conditionally independent with each other for given $\mathbf{h}$, thus

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i|\mathbf{h}). \tag{8}$$

Thus for any $1 \leq i \leq m$,

$$P(v_i|\mathbf{h}) \propto \exp\left(a_i v_i + v_i \sum_{j=1}^{n} W_{ij} h_j\right). \tag{9}$$

Since $P(v_i = 1|\mathbf{h}) + P(v_i = 0|\mathbf{h}) = 1$, we have

$$P(v_i = 1|\mathbf{h}) = \frac{\exp\left(a_i + \sum_{j=1}^{n} W_{ij} h_j\right)}{1 + \exp\left(a_i + \sum_{j=1}^{n} W_{ij} h_j\right)} \tag{10}$$

$$= \sigma\left(a_i + \sum_{j=1}^{n} W_{ij} h_j\right). \tag{11}$$

Here σ is the logistic/sigmoid function. Similar for $P(h_j = 1|\mathbf{v})$. This type of units are called **logistic units**.

If the hidden states are real-valued and the energy function is defined by (4), then $P(v_i = 1|\mathbf{h})$ remains the same, but $P(h_j = 1|\mathbf{v})$ is different.

$$\begin{aligned} P(\mathbf{h}|\mathbf{v}) &\propto P(\mathbf{v}, \mathbf{h}) && (12) \\ &\propto \exp\left(\mathbf{a}\mathbf{v}^{\mathrm{T}} - \frac{\|\mathbf{h}-\mathbf{b}\|^2}{2} + \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}\right) && (13) \\ &\propto \prod_{j=1}^{n} \exp\left(-\frac{(h_j - b_j)^2}{2} + h_j \sum_{i=1}^{m} v_i W_{ij}\right). && (14) \end{aligned}$$

On the other hand, the hidden units are conditionally independent with each other for given $\mathbf{v}$, thus

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n} P(h_j|\mathbf{v}). \qquad (15)$$

Thus for any $1 \leq j \leq n$,

$$P(h_j|\mathbf{v}) \propto \exp\left(-\frac{(h_j - b_j)^2}{2} + h_j \sum_{i=1}^{m} v_i W_{ij}\right). \qquad (16)$$

After normalization, we get

$$P(h_j|\mathbf{v}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(h_j - \mu_j)^2}{2}\right), \qquad (17)$$

where

$$\mu_j = b_j + \sum_{i=1}^{m} v_i W_{ij}. \qquad (18)$$

Thus the hidden units are Gaussian whose means μ_j's are linear functions of visible states v_i. This type of hidden units are called **Gaussian units** or **linear units** (with Gaussian noise).

Learn an RBM

RBM is a generative model. Given parameters $(\mathbf{W}, \mathbf{a}, \mathbf{b})$, we can generate data by sampling from the Boltzmann distribution (1). On the other hand, given enough data we want to estimate the parameters $(\mathbf{W}, \mathbf{a}, \mathbf{b})$ of the RBM.

The only data we have are the observed states of visible units $\mathbf{v}$. It's natural to maximize their chance to be observed (**likelihood**). Suppose that $P(\mathbf{v}, \mathbf{h})$ is in the form of Boltzmann distribution (1). Then the marginal distribution of visible units is

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}, \tag{19}$$

where the sum is taken over all possible hidden states.

Denote the dataset of observed visible states as V. Its expected log likelihood is

$$L(\mathbf{W}, \mathbf{a}, \mathbf{b}) = \frac{1}{|V|} \log \prod_{\mathbf{v} \in V} P(\mathbf{v}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} \log P(\mathbf{v}). \tag{20}$$

We can use the method of **gradient ascent** to maximize L:

- Initialize $\mathbf{W}, \mathbf{a}, \mathbf{b}$ and choose a learning rate ϵ.
- Repeat

$$\mathbf{W} \leftarrow \mathbf{W} + \epsilon \frac{1}{|V|} \sum_{\mathbf{v} \in V} \frac{\partial \log P(\mathbf{v})}{\partial \mathbf{W}} \tag{21}$$

$$\mathbf{a} \leftarrow \mathbf{a} + \epsilon \frac{1}{|V|} \sum_{\mathbf{v} \in V} \frac{\partial \log P(\mathbf{v})}{\partial \mathbf{a}} \tag{22}$$

$$\mathbf{b} \leftarrow \mathbf{b} + \epsilon \frac{1}{|V|} \sum_{\mathbf{v} \in V} \frac{\partial \log P(\mathbf{v})}{\partial \mathbf{b}}. \tag{23}$$

Let's calculate the needed derivatives.

$$\frac{\partial P(\mathbf{v})}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \left(\frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \right) \tag{24}$$

$$= \frac{1}{Z} \frac{\partial}{\partial \mathbf{W}} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} + \left(\frac{\partial}{\partial \mathbf{W}} \frac{1}{Z} \right) \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{25}$$

$$= \frac{1}{Z} \sum_{\mathbf{h}} \frac{\partial e^{-E(\mathbf{v},\mathbf{h})}}{\partial \mathbf{W}} - \frac{1}{Z^2} \frac{\partial Z}{\partial \mathbf{W}} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{26}$$

In the case that both visible and hidden units are logistic, the energy E is given by (3). Thus

$$\frac{\partial e^{-E(\mathbf{v},\mathbf{h})}}{\partial \mathbf{W}} = e^{-E(\mathbf{v},\mathbf{h})} \frac{\partial}{\partial \mathbf{W}} \left(\mathbf{a}\mathbf{v}^{\mathrm{T}} + \mathbf{b}\mathbf{h}^{\mathrm{T}} + \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}} \right) \tag{27}$$

$$= e^{-E(\mathbf{v},\mathbf{h})} \mathbf{v}^{\mathrm{T}} \mathbf{h}. \tag{28}$$

Therefore

$$\frac{\partial Z}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} = \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \mathbf{v}'^{\mathrm{T}} \mathbf{h}'. \tag{29}$$

Plug these two results back, we get

$$\begin{aligned} \frac{\partial P(\mathbf{v})}{\partial \mathbf{W}} = & \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \mathbf{v}^{\mathrm{T}} \mathbf{h} && (30) \\ & - \frac{1}{Z^2} \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \mathbf{v}'^{\mathrm{T}} \mathbf{h}' \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} && (31) \\ = & \sum_{\mathbf{h}} P(\mathbf{v},\mathbf{h}) \mathbf{v}^{\mathrm{T}} \mathbf{h} - P(\mathbf{v}) \sum_{\mathbf{v}',\mathbf{h}'} P(\mathbf{v}',\mathbf{h}') \mathbf{v}'^{\mathrm{T}} \mathbf{h}'. && (32) \end{aligned}$$

Thus

$$\frac{\partial \log P(\mathbf{v})}{\partial \mathbf{W}} = \frac{1}{P(\mathbf{v})}\frac{\partial P(\mathbf{v})}{\partial \mathbf{W}} \tag{33}$$

$$= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v})\mathbf{v}^{\mathrm{T}}\mathbf{h} - \sum_{\mathbf{v}',\mathbf{h}'} P(\mathbf{v}',\mathbf{h}')\mathbf{v}'^{\mathrm{T}}\mathbf{h}' \tag{34}$$

$$= \mathbb{E}_{\mathbf{h}|\mathbf{v}}[\mathbf{v}^{\mathrm{T}}\mathbf{h}] - \mathbb{E}[\mathbf{v}^{\mathrm{T}}\mathbf{h}] \tag{35}$$

$$=: \langle \mathbf{v}^{\mathrm{T}}\mathbf{h}\rangle_{data} - \langle \mathbf{v}^{\mathrm{T}}\mathbf{h}\rangle_{model}. \tag{36}$$

Similary, we have

$$\frac{\partial \log P(\mathbf{v})}{\partial \mathbf{a}} = \langle \mathbf{v}\rangle_{data} - \langle \mathbf{v}\rangle_{model}, \tag{37}$$

$$\frac{\partial \log P(\mathbf{v})}{\partial \mathbf{b}} = \langle \mathbf{h}\rangle_{data} - \langle \mathbf{h}\rangle_{model}. \tag{38}$$

Notice that $\langle \mathbf{v}^{\mathrm{T}}\mathbf{h}\rangle_{data}$ is a matrix. Let's consider its entries.

$$
\begin{aligned}
\langle v_i h_j\rangle_{data} &= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) v_i h_j && (39)\\
&= v_i \sum_{h_j} \sum_{\mathbf{h}\backslash h_j} P(h_j|\mathbf{v}) \prod_{k\neq j} P(h_k|\mathbf{v}) h_j && (40)\\
&= v_i \sum_{h_j} h_j P(h_j|\mathbf{v}) \sum_{\mathbf{h}\backslash h_j} \prod_{k\neq j} P(h_k|\mathbf{v}) && (41)\\
&= v_i \sum_{h_j} h_j P(h_j|\mathbf{v}) && (42)\\
&= v_i P(h_j = 1|\mathbf{v}). && (43)
\end{aligned}
$$

Similary, we have

$$
\begin{aligned}
\langle v_i\rangle_{data} &= v_i, && (44)\\
\langle h_j\rangle_{data} &= P(h_j = 1|\mathbf{v}). && (45)
\end{aligned}
$$

The key for the above simplification of $\langle\rangle_{data}$ is the conditional independence of hidden units for given $\mathbf{v}$. This property is absent for the unconditional distribution $P(\mathbf{v}, \mathbf{h})$. So we can not get a simple formula to compute model expectations $\langle\rangle_{model}$ exactly. It can be estimated by **alternating Gibbs sampling** of $(\mathbf{v}, \mathbf{h})$.

Firstly,

- Given an observed visible state $\mathbf{v}^0$, draw $\mathbf{h}^0$ from $P(\mathbf{h}|\mathbf{v}^0)$.

Then repeat the following procedure for several times:

- Draw $\mathbf{v}^l$ from $P(\mathbf{v}|\mathbf{h}^{l-1})$, called a **reconstruction**.
- Draw $\mathbf{h}^l$ from $P(\mathbf{h}|\mathbf{v}^l)$.

Use the last pair $(\mathbf{v}^l, \mathbf{h}^l)$ to estimate model expectations:

$$\langle v_i h_j \rangle_{model} \approx \langle v_i^l h_j^l \rangle_{reconstruction}, \tag{46}$$

$$\langle v_i \rangle_{model} \approx \langle v_i^l \rangle_{reconstruction}, \tag{47}$$

$$\langle h_j \rangle_{model} \approx \langle h_j^l \rangle_{reconstruction}. \tag{48}$$

Monitoring the Learning

It's usually impractical to monitor the likelihood. Instead, two quantities are used: **reconstruction error**

$$\|\mathbf{v}^1 - \mathbf{v}^0\|^2 \tag{49}$$

and **free energy**

$$F(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{50}$$

As training going on, these two quantities should decrease. [1] The computation of reconstruction error is straightforward. Let's derive the formula of free energy for given energy.

[1] Not always!

For binary states, with energy defined by (3), we have

$$F(\mathbf{v}) = -\log \sum_{\mathbf{h}} \exp\left(\mathbf{a}\mathbf{v}^{\mathrm{T}} + \mathbf{b}\mathbf{h}^{\mathrm{T}} + \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}\right) \tag{51}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \sum_{\mathbf{h}} \exp\left(\sum_{j=1}^{n}\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right) h_j\right) \tag{52}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \sum_{\mathbf{h}} \prod_{j=1}^{n} \exp\left(\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right) h_j\right) \tag{53}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \prod_{j=1}^{n} \sum_{h_j} \exp\left(\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right) h_j\right) \tag{54}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \sum_{j=1}^{n} \log\left(1 + \exp\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right)\right). \tag{55}$$

If the hidden states are real-valued and the energy function is defined by (4),

$$F(\mathbf{v}) = -\log \int_{\mathbb{R}} \exp\left(\mathbf{a}\mathbf{v}^{\mathrm{T}} - \frac{\|\mathbf{h}-\mathbf{b}\|^2}{2} + \mathbf{v}\mathbf{W}\mathbf{h}^{\mathrm{T}}\right) \mathrm{d}\mathbf{h} \tag{56}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \int_{\mathbb{R}} \exp\left(\sum_{j=1}^{n}\left(-\frac{(h_j-b_j)^2}{2} + \sum_{i=1}^{m} v_i W_{ij}\right)\right) \mathrm{d}\mathbf{h} \tag{57}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \int_{\mathbb{R}} \prod_{j=1}^{n} \exp\left(-\frac{(h_j-b_j)^2}{2} + \sum_{i=1}^{m} v_i W_{ij}\right) \mathrm{d}\mathbf{h} \tag{58}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \log \prod_{j=1}^{n} \int_{\mathbb{R}} \exp\left(-\frac{(h_j-b_j)^2}{2} + \sum_{i=1}^{m} v_i W_{ij}\right) \mathrm{d}h_j \tag{59}$$

$$F(\mathbf{v}) = -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \sum_{j=1}^{n} \log\left(\sqrt{2\pi}\exp\left(\frac{\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right)^2 - b_j^2}{2}\right)\right) \tag{60}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{n}\left(\left(b_j + \sum_{i=1}^{m} v_i W_{ij}\right)^2 - b_j^2\right) \tag{61}$$

$$= -\mathbf{a}\mathbf{v}^{\mathrm{T}} - \frac{n}{2}\log(2\pi) - \frac{1}{2}\left(\|\mathbf{b} + \mathbf{v}\mathbf{W}\|^2 - \|\mathbf{b}\|^2\right). \tag{62}$$

Practical Issues

In practice, some modifications and tricks are employed to improve efficiency and accuracy.

- It's more efficient to partition V into many mini-batches. Each update only randomly use one of the mini-batches.
- To compute $P(\mathbf{h}|\mathbf{v})$, use the probability of $\mathbf{v}$ instead of $\mathbf{v}$ itself.
- In alternating Gibbs sampling, repeating once already works well.

For more details about practical issues, refer to Hinton and Salakhutdinov (2006), E. Hinton (2010).

Deep Belief Network

A DBN is a multi-layer generalization of RBM, which contains several layers of hidden units. It can be **trained as a stack of RBMs**. A sample of hidden units in one RBM serves as an observation of the visible units in the higher RBM.

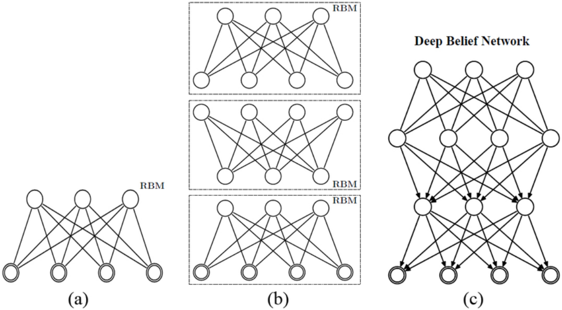


Figure: Restricted Boltzmann Machine (copied from internet).

A DBN can be unrolled into a symmetric DNN called **autoencoder**. The parameters learned from RBMs can be used as initialization of the DNN. Reconstruction by the DBN is equivalent to feeding data through this autoencoder. It is then fine-tuned to improve its performance.

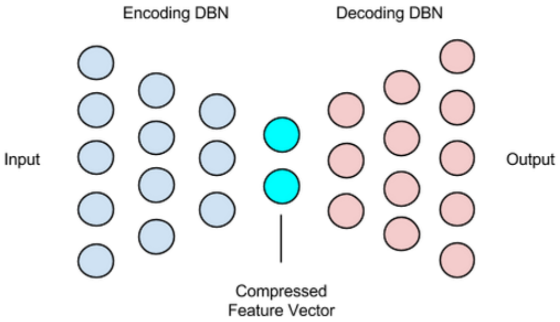


Figure: Autoencoder (copied from internet).

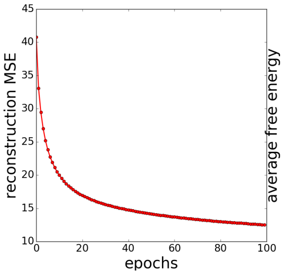
Experiments

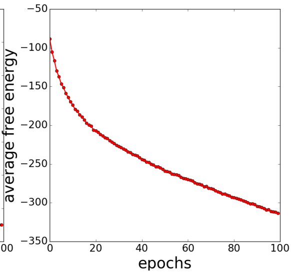
Data: MNIST is a dataset of $m = 28 \times 28 = 784$ grayscale images of handwritten digits $0 \sim 9$ with labels. [2] The values of pixels are within $[0, 1]$. **They are regarded as probabilities of binary states.** The training set V contains $|V| = 60000$ samples. The test set contains 10000 samples.

[2] The dataset is downloaded from Yann LeCun's MNIST database (http://yann.lecun.com/exdb/mnist/).

RBM

- Visible units: $m = 784$
- Hidden units: $n = 100$
- Algrorithm: SGD with mini-batch size 10
- # epochs: 100
- Learning rate: 0.001

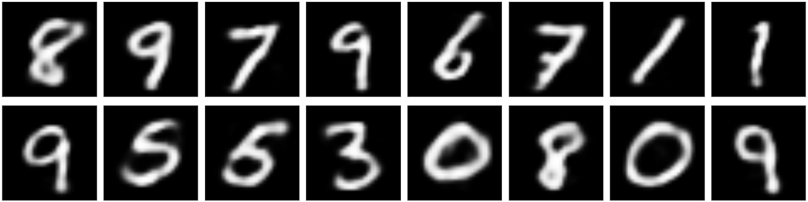




RBM

original

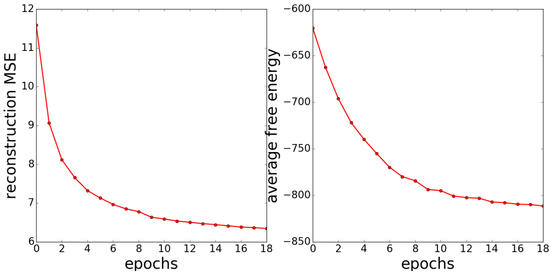
reconstruction



RBM

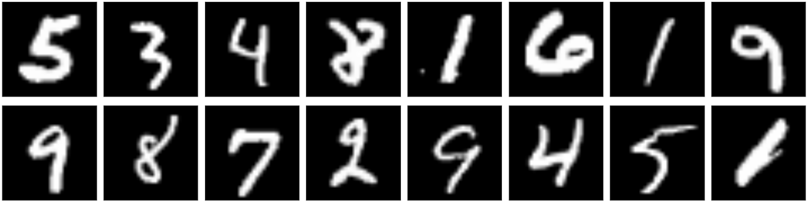
Add more hidden units.

- Hidden units: $n = 1000$
- # epochs: 20
- Learning rate: 0.01

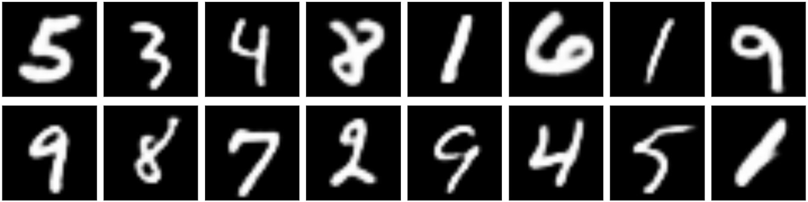


RBM

original

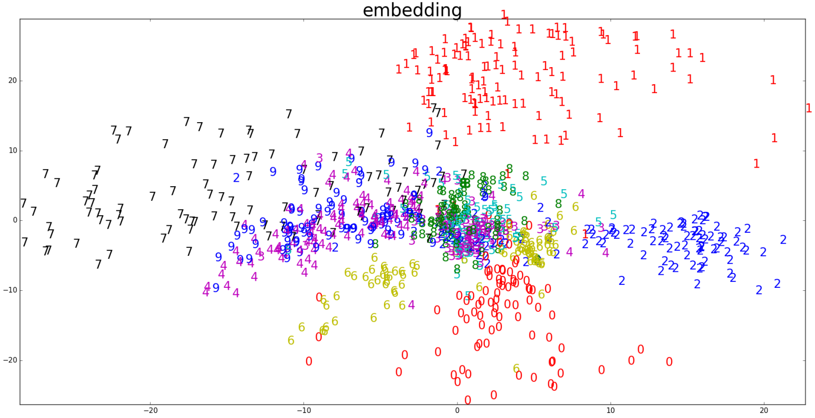


reconstruction

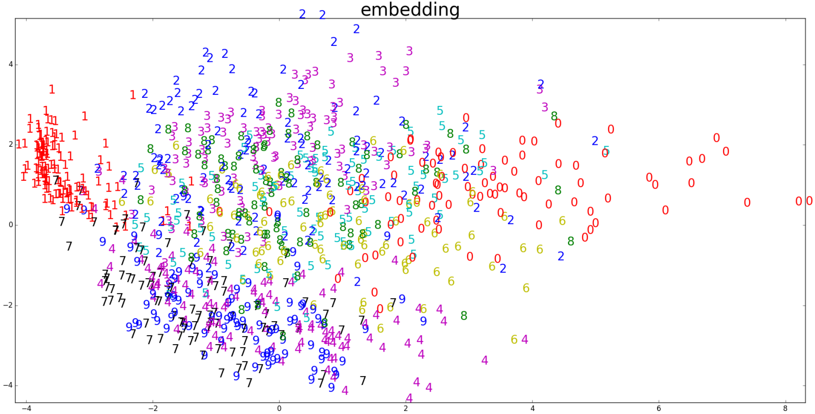


Embedding by DBN

Learn a 2D code through DBN (784-1000-500-250-2):



Compare with 2D embedding of PCA:



Thank you

E. Hinton, G. (2010). A practical guide to training restricted boltzmann machines (version 1). Technical report.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.