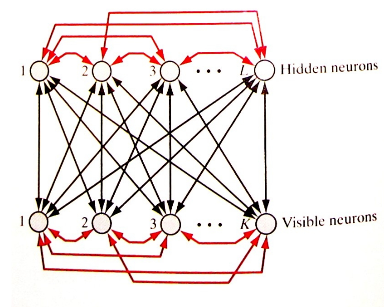
Boltzmann Machine

- CSCE 636 Neural Networks
- Haykin Chapter 11: Stochastic Methods rooted in statistical mechanics.
- Instructor: Yoonsuck Choe

Boltzmann Machine



- Stochastic binary machine: +1 or -1.
- Fully connected symmetric connections: $w_{ij} = w_{ji}$.
- Visible vs. hidden neurons, clamped vs. free-running.
- Goal: Learn weights to model prob. dist of visible units.
- Unsupervised. Pattern completion.

Boltzmann Machine: Energy

- Network state: $\mathbf{x}$ from random variable $\mathbf{X}$.
- $w_{ij} = w_{ji}$ and $w_{ii} = 0$.
- Energy (in analogy to thermodynamics):

$$E(\mathbf{x}) = -\frac{1}{2}\sum_i \sum_{j, i \neq j} w_{ji} x_i x_j$$

Boltzmann Machine: Prob. of a State $\mathbf{x}$

- Probability of a state $\mathbf{x}$ given $E(\mathbf{x})$ follows the *Gibbs distribution*:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z}\exp\left(-\frac{E(\mathbf{x})}{T}\right),$$

  - Z: *partition function* (normalization factor – hard to compute)

$$Z = \sum_{\forall \mathbf{x}} \exp(-E(\mathbf{x})/T)$$

  - T: temperature parameter.
  - Low energy states are exponentially more probable.
- With the above, we can calculate

$$P(X_j = x | \{X_i = x_i\}_{i=1, i\neq j}^{K})$$

  - This can be done without knowing Z.

Boltzmann Machine: $P(X_j = x|\text{the rest})$

- $A : X_j = x$. $B : \{X_i = x_i\}_{i=1,i\neq j}^{K}$ (the rest).

$$\begin{aligned} P(X_j = x|\text{the rest}) &= \frac{P(A,B)}{P(B)} \\ &= \frac{P(A,B)}{\sum_A P(A,B)} = \frac{P(A,B)}{P(A,B)+P(\neg A,B)} \\ &= \frac{1}{1+\exp\left(-\frac{x}{T}\sum_{i,i\neq j} w_{ji}x_i\right)} \\ &= \text{sigmoid}\left(\frac{x}{T}\sum_{i,i\neq j} w_{ji}x_i\right) \end{aligned}$$

- Can compute equilibrium state based on the above.

Boltzmann Machine: Gibbs Sampling

- Initialize $\mathbf{x}^{(0)}$ to a random vector.
- For $j = 1, 2, ..., n$ (generate n samples $\mathbf{x} \sim P(\mathbf{X})$)
  - $x_1^{(j+1)}$ from $p(x_1|x_2^{(j)}, x_3^{(j)}, ..., x_K^{(j)})$
  - $x_2^{(j+1)}$ from $p(x_2|x_1^{(j+1)}, x_3^{(j)}, ..., x_K^{(j)})$
  - $x_3^{(j+1)}$ from $p(x_3|x_1^{(j+1)}, x_2^{(j+1)}, x_4^{(j)} ..., x_K^{(j)})$
  - ...
  - $x_K^{(j+1)}$ from $p(x_K|x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)} ..., x_{K-1}^{(j+1)})$

  $\rightarrow$ One new sample $\mathbf{x}^{(j+1)} \sim P(\mathbf{X})$.
- Simulated annealing used (high T to low T) for faster conv.

Boltzmann Learning Rule (1)

- Probability of activity pattern being *one of* the training patterns (visible unit: subvector $\mathbf{x}_\alpha$; hidden unit: subvector $\mathbf{x}_\beta$), given the weight vector $\mathbf{w}$.

$$P(\mathbf{X}_\alpha = \mathbf{x}_\alpha)$$

- Log-likelihood of the visible units being *any one of* the trainning patterns (assuming they are mutually independent) $\mathcal{T}$:

$$\begin{aligned} L(\mathbf{w}) &= \log \prod_{\mathbf{x}_\alpha \in \mathcal{T}} P(\mathbf{X}_\alpha = \mathbf{x}_\alpha) \\ &= \sum_{\mathbf{x}_\alpha \in \mathcal{T}} \log P(\mathbf{X}_\alpha = \mathbf{x}_\alpha) \end{aligned}$$

- We want to learn $\mathbf{w}$ that **maximizes** $L(\mathbf{w})$.

Boltzmann Learning Rule (2)

- Want to calculate $P(\mathbf{X}_\alpha = \mathbf{x}_\alpha)$ (probability of finding the visible neurons in state $\mathbf{x}_\alpha$ with any $\mathbf{x}_\beta$): use energy function.

$$\begin{aligned} P(\mathbf{X}_\alpha = \mathbf{x}_\alpha) &= \sum_{\mathbf{x}_\beta} P(\mathbf{X}_\alpha = \mathbf{x}_\alpha, \mathbf{X}_\beta = \mathbf{x}_\beta) \\ &= \frac{1}{Z}\sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right) \\ \log P(\mathbf{X}_\alpha = \mathbf{x}_\alpha) &= \log\sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right) - \log Z \\ &= \log\sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right) \\ &\quad - \log\sum_{\mathbf{x}} \exp\left(-\frac{E(\mathbf{x})}{T}\right) \end{aligned}$$

- Note: $Z = \sum_{\mathbf{x}} \exp\left(-\frac{E(\mathbf{x})}{T}\right)$

Boltzmann Learning Rule (3)

- Finally, we get:

$$L(\mathbf{w}) = \sum_{\mathbf{x}_\alpha \in \mathcal{T}} \left(\log \sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right) - \log \sum_{\mathbf{x}} \exp\left(-\frac{E(\mathbf{x})}{T}\right) \right)$$

- Note that $\mathbf{w}$ is involved in:

$$E(\mathbf{x}) = -\frac{1}{2}\sum_i \sum_{j, i \neq j} w_{ji} x_i x_j$$

- Differentiating $L(\mathbf{w})$ wrt w_{ji}, we get:

$$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \frac{1}{T}\sum_{\mathbf{x}_\alpha \in \mathcal{T}} \left(\sum_{\mathbf{x}_\beta} P(\mathbf{X}_\beta = \mathbf{x}_\beta | \mathbf{X}_\alpha = \mathbf{x}_\alpha) x_j x_i - \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) x_j x_i \right)$$

Boltzmann Learning Rule (3-2): Some hints

To derive: $$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \frac{1}{T}\sum_{\mathbf{x}_\alpha \in \mathcal{T}} \left(\sum_{\mathbf{x}_\beta} P(\mathbf{X}_\beta = \mathbf{x}_\beta | \mathbf{X}_\alpha = \mathbf{x}_\alpha) x_j x_i - \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) x_j x_i \right)$$

$$\frac{\partial E(\mathbf{x})}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \left(-\frac{1}{2}\sum_i \sum_{j, i \neq j} w_{ji} x_i x_j \right) = -\frac{1}{2} x_i x_j, \quad i \neq j$$

$$P(\mathbf{X}_\beta = \mathbf{x}_\beta | \mathbf{X}_\alpha = \mathbf{x}_\alpha) = \frac{P(\mathbf{X}_\alpha = \mathbf{x}_\alpha, \mathbf{X}_\beta = \mathbf{x}_\beta)}{P(\mathbf{X}_\alpha = \mathbf{x}_\alpha)} = \frac{\frac{1}{Z}\exp\left(-\frac{E(\mathbf{x})}{T}\right)}{\frac{1}{Z}\sum_{\mathbf{x}_\beta}\exp\left(-\frac{E(\mathbf{x})}{T}\right)}$$

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z}\exp\left(-\frac{E(\mathbf{x})}{T}\right) = \frac{\exp\left(-\frac{E(\mathbf{x})}{T}\right)}{\sum_{\mathbf{x}}\exp\left(-\frac{E(\mathbf{x})}{T}\right)}$$

Boltzmann Learning Rule (4)

- Setting:

$$\rho_{ji}^+ = \sum_{\mathbf{x}_\alpha \in \mathcal{T}} \sum_{\mathbf{x}_\beta} P(\mathbf{X}_\beta = \mathbf{x}_\beta | \mathbf{X}_\alpha = \mathbf{x}_\alpha) x_j x_i$$

$$\rho_{ji}^- = \sum_{\mathbf{x}_\alpha \in \mathcal{T}} \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) x_j x_i$$

- We get:

$$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \frac{1}{T}\left(\rho_{ji}^+ - \rho_{ji}^-\right)$$

- Attempting to maximize $L(\mathbf{w})$, we get:

$$\Delta w_{ji} = \epsilon \frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \eta \left(\rho_{ji}^+ - \rho_{ji}^-\right)$$

where $\eta = \frac{\epsilon}{T}$. This is *gradient ascent*.

Boltzmann Machine Summary

- Theoretically elegant.
- Slow in practice (especially the unclamped phase).