FYS 4411, APRIL 21, 2022

Neural Network : Boltzmann machines

P1 : $\Psi_T(\vec{R};\vec{\alpha}) = \varphi_{OB}(\vec{r}_1, \vec{r}_2, \ldots \vec{r}_N;\vec{\alpha}) \times \Psi_C(\vec{r}_1 \vec{r}_2 \ldots \vec{r}_N;\vec{\alpha})$

$$\Psi_C(\vec{r}_1, \vec{r}_2, \ldots \vec{r}_N;\vec{\alpha}) = \prod_{i<j} f(|\vec{r}_{ij}|)$$

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$

$$\varphi_{OB} = \prod_{i=1}^{N} e^{-\alpha^2 r_i^2}$$

P2

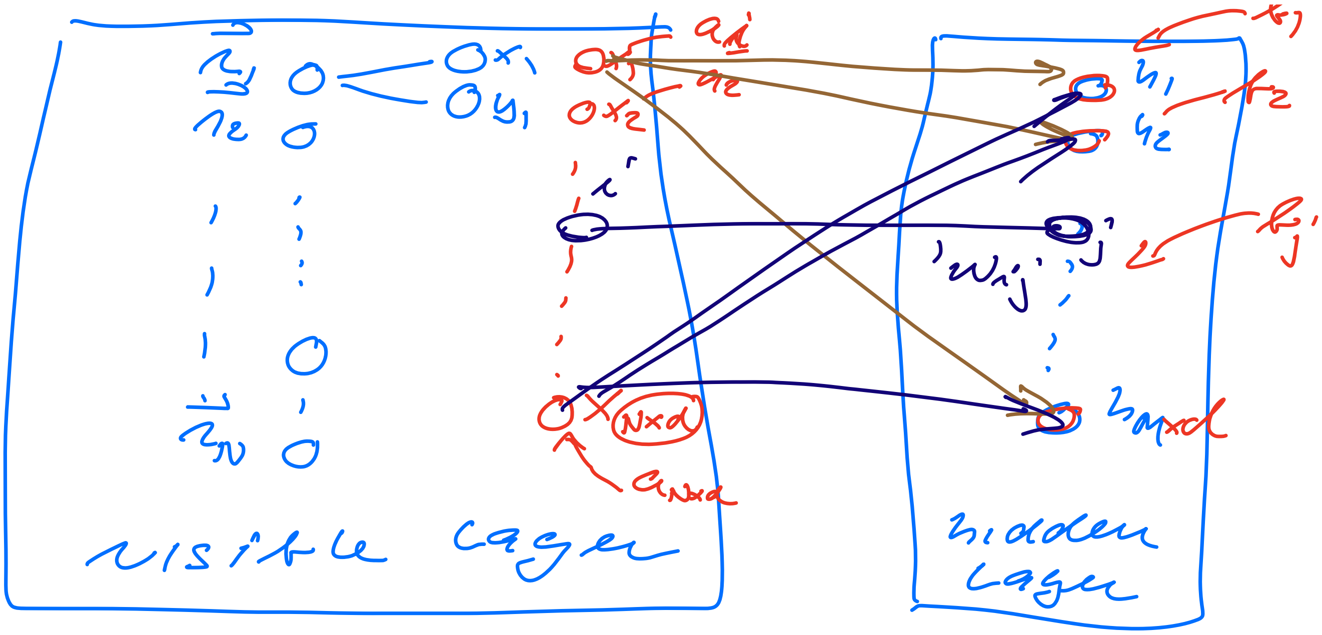
$$\Psi_T(\vec{R};\vec{\alpha}) =$$

$$(\vec{R} = \{\vec{r}_1, \vec{r}_2, \ldots \vec{r}_N\})$$

$$= \frac{1}{Z} e^{-E(\vec{R}, \vec{h};\theta)}$$

$$Z = \int d\vec{r}_1 \int d\vec{r}_2 \ldots \int d\vec{r}_N \int d\vec{h}_1 \ldots \int d\vec{h}_M \, e^{-E(\vec{r}_1 \vec{r}_2, \ldots; h_1 h_2 \ldots; \theta)}$$

Gaussian-Binary E :



our case $N = 2$
we need 4 input
nodes

$$E(\vec{r}_1, \vec{r}_2; \vec{h}_1, \vec{h}_2; \Theta)$$

$$\Theta = \begin{cases} \text{Bias visible } a_i \\ \text{Bias hidden } b_j \\ \text{weights between} \\ \text{hidden and visible} \\ \text{layer } w_{ij} \end{cases}$$

Binary _ Binary

$$E_{BB} = \sum_{i=1}^{N\times d} x_i a_i + \sum_{j=1}^{M\times d} h_j b_j + \sum_{i,j}^{N\times d, M\times d} w_{ij} x_i h_j$$

$x_i = \{0, 1\}$; $h_i = \{0, 1\}$

Gaussian – Binary

$$\sum_{i=1}^{N\times d} x_i a_i \rightarrow \sum_{i=1}^{N\times d} \frac{(x_i - a_i)^2}{\sigma_i^2}$$

E_{GB} in our case.

Two approaches;

(i) $|\Psi_T|^2 = \frac{1}{Z} \sum_{\vec{h}} e^{-E(\vec{x};\vec{h};\Theta)} = RBM$

(ii) $\Psi_T = \_\_\_\_ , \_\_\_\_$

Add more approaches

– with an interaction

$$V(r_{ij}) \propto \frac{1}{r_{ij}}$$

$$\Psi_{CMIN} \propto e^{\gamma r_{ij}}$$

$$|\Psi_T|^2 = RBM \times \left(e^{\gamma r_{ij}}\right)^2$$

$$= RBM \times (\Psi_{CMIN})^2$$

– For fermion (Ψ has to be antisymmetric)

$$|\Psi_T|^2 = |\Psi_{SD}|^2 \times RBM \times (\Psi_{CMIN})^2$$

$$\frac{d\langle E\rangle}{d\Theta} = 2\left[\left\langle \frac{\psi_T'}{\psi_T} E_L \right\rangle - \left\langle \frac{\psi_T'}{\psi_T} \right\rangle \langle E_L \rangle \right]$$

$$\langle \ln \psi_T \rangle$$

with an RBM implemented we can then replace the RBM with a Neural Network

- $\psi_T \sim \text{RBM}$ (EGB)
- $\psi_T \sim \text{RBM} \times \psi_{Cmin}$

use this RBM as input to a neural network

$$N(\vec{R};\Theta)$$

- $\psi_T = N(\vec{R};\Theta)$
- $\psi_T = N(\vec{R};\Theta) \times \psi_{Cmin}$
- $\psi_T = \psi_{HD}(\vec{R}) \times \psi_{Cmin}$

$$\times \ \mathcal{N}(\vec{x}; \Theta)$$