FYS4411/9411, APRIL 20, 2023

Deep learning

- Discriminative approach (supervised learning) NN, CNN, RNN, ...
- Generative methods unsupervised learning, Boltzmann machines, GANs, variational autoencoders

Boltzmann machines

- Dataset (x, y)
- Model with parameters Θ
- cost/loss/risk.. function

optimize cost function (gradient descent methods) find optimal parameters Θ

Dataset $\underbrace{y \simeq f(x;\Theta)}_{\text{Model}}$

cost function $C(\Theta)$

Example:

mean squared error

$$C(\Theta) = \frac{1}{N}\sum_{i=0}^{N-1}\left(y_i - f(x_i;\Theta)\right)^2$$

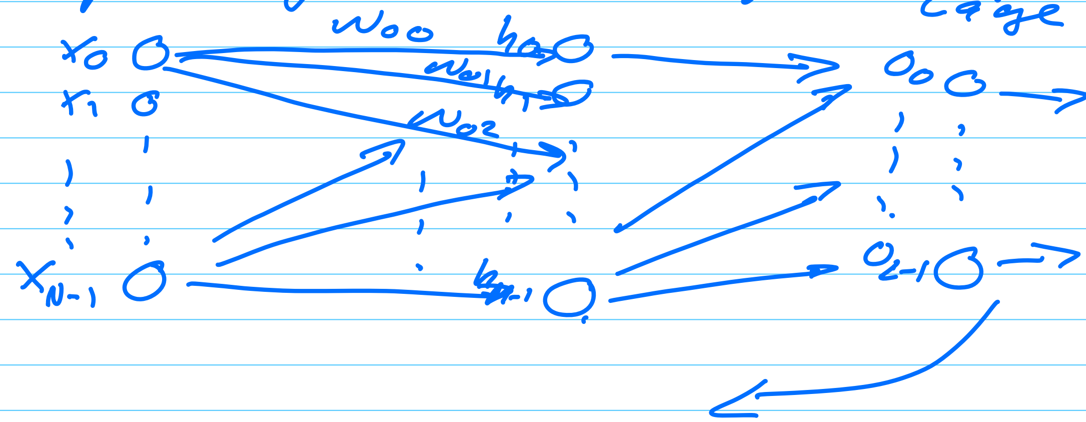
optimization (optimal $\hat{\theta}$)

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^p} C(\theta)$$

$$\frac{\partial C(\theta)}{\partial \theta} = 0$$

Neural network (Model)

input layer hidden layer output layer



$$\text{output} = f(x;\theta)$$

$$\theta = \{W, b\}$$

- Back propagation algo
- Automatic differentiation

<u>Project 2</u>

Markov chain Monte Carlo

Gradient Descent
Metropolis-Hastings
(Gibbs sampling)
Blocking.

Standard NN

$$\psi(\vec{R};\alpha) \Rightarrow \psi(x;\theta)$$

$$= \underbrace{NN(x;\theta)}_{\text{neural network.}}$$

Examples in lectures in python. | C++ not optimal for including ML libraries

Boltzmann machine

$$\psi(\vec{R};\alpha) \Rightarrow \psi(x;\theta)$$

$$= p(x;\theta) = \frac{1}{Z} e^{-\beta E(x;\theta)}$$

$E(x;\theta)$ reflects the way we model the network

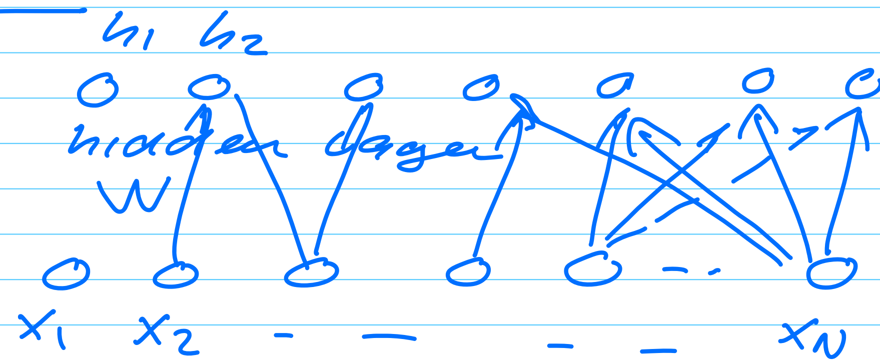
Benefit: we can reuse p1 fully, but need to replace $\psi(\vec{R};\alpha)$ with the Boltzmann distribution

The function to optimize

$$\langle E(\theta)\rangle$$

$$\vec{\nabla}_\theta \langle E(\theta)\rangle = 0$$

RBM



visible layer

$x_i W_{ij} h_j$ $\qquad$ $b_i x_i$

weights $\qquad$ biases

$c_j h_j$ $\qquad$ $\theta = \{b, c, W\}$

biases

$$\psi(x;\theta) \Rightarrow p(x, h; \theta)$$

$$p(x, h; \theta) = \frac{1}{Z} \tilde{p}(x, h; \theta)$$

$$\tilde{p}(x, h; \theta) = e^{-\beta E(x, h; \theta)}$$

$$Z = \sum_{x h} \tilde{p}(x, h; \theta)$$

$E(x, h; \theta)$

- Binary - Binary

$$E(x, h; \theta) = \sum_{i=0}^{N-1} b_i x_i + \sum_{j=0}^{M-1} c_j h_j$$

$$+ \sum_{i j}^{N-1, M-1} x_i W_{ij} h_j$$

we are interested in

$$\psi(x; \theta) = \underbrace{p(x; \theta)}_{\text{marginal distribution}}$$

$$p(x; \theta) = \sum_h p(x, h; \theta) \quad \left(\int_{h \in D} p(x, h; \theta) \, dh \right)$$

Gaussian - Binary

$$E(x, h; \theta) = \sum_{i=0}^{N-1} \frac{(x_i - b_i)^2}{\sigma_i^2}$$

$$+ \sum_{j=0}^{N-1} c_j h_j + \sum_{ij}^{M-1\,N-1} x_i W_{ij} h_j$$

Assume that $p(x;\theta)$

$$= \frac{1}{Z(\theta)} \tilde{p}(x;\theta)$$

$$\tilde{p}(x;\theta) = e^{-\beta E(x;\theta)}$$

$$Z(\theta) = \sum_x \tilde{p}(x;\theta)$$

$$\beta = \frac{1}{k_B T} = 1$$

We want optimize $p(x;\theta)$

optimize $\log p(x;\theta)$

$$\nabla_\theta [\log p(x;\theta)] = 0$$

$$= \nabla_\theta \log \tilde{p}(x;\theta) -$$

$$\nabla_\theta \log Z(\theta)$$

positive phase

negative phase

$$\nabla_\theta \log Z(\theta) = \frac{\nabla_\theta Z(\theta)}{Z}$$

$$= \frac{\nabla_\theta \left[\sum_x \tilde{p}(x;\theta) \right]}{Z}$$

$$= \sum_x \frac{\nabla_\theta \tilde{p}(x;\theta)}{Z}$$

$$\left(\begin{array}{l} \tilde{p}(x;\theta) > 0 \\ \tilde{p}(x) = \exp(\log \tilde{p}(x)) \end{array} \right)$$

$$= \sum_x \frac{\nabla_\theta \exp(\log \tilde{p}(x;\theta))}{Z}$$

$$= \frac{\sum_x \exp(\log \tilde{p}(x;\theta)) \nabla_\theta \log \tilde{p}}{Z}$$

$$= \sum_x \frac{\tilde{p}(x;\theta) \nabla_\theta \log \tilde{p}(x;\theta)}{Z}$$

$$\frac{\tilde{p}(x;\theta)}{Z(\theta)} = p(x;\theta)$$

$$= \sum_x p(x;\theta) \nabla_\theta \log \tilde{p}(x;\theta)$$

$$\left(\mathbb{E}[f^u] = \sum_x p(x;\theta) f^u(x;\theta) \right)$$

$$= \mathbb{E}\left[\nabla_\theta \log \tilde{p}(x;\theta) \right]$$

$$= \nabla_\theta \log Z(\theta)$$