Restricted Boltzman Machines

Léo Tarbouriech and Joseph Touzet

Machnie Learning
M2 PCS

4 mars 2024

1. RBM and EBM
    - Energy Based Models (EBM)
    - Hopfield model and Restricted Boltzman Machines
    - Trainning an RBM
    - Gibbs sampling
2. At/Out of equilibrium Boltzmann machines
    - Training an RBM in and out of equilibrium
    - Out of equilibrium
    - Training and sampling an out of equilibrium RBM
3. RBM as a Langevin Process
    - Learning a Langevin Process

Energy Based Models (EBM)

- Learn correlations between features
- Assume we do not know anything about the correlations
- Two point correlations are sufficients because the model is fully connected
- Recall statistical field theory course –> fully connected lattice $\leftrightarrow$ only gaussian terms matters

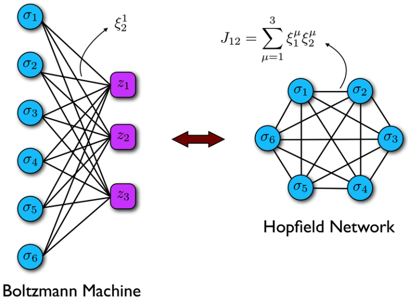
$$
\begin{aligned}
H &= -\sum_{i=1}^{N_{features}} -\frac{1}{2}\sum_{i,j} v_i J_{ij} v_j \\
p(\mathrm{v}|\{a_i, J_{ij}\}) &= \frac{1}{Z}\exp\left(\sum_i a_i v_i + \sum_{ij} v_i J_{ij} v_j\right)
\end{aligned}
$$

- J_{ij} is a fully connected random matrice

RBM and EBM

Hopfield model and Restricted Boltzman Machines

- In an Hopfield model we would have $J_{ij} = W_{i\mu} W_{j\mu}$
- With $W_{i\mu} = \{-1, +1\}$
- This model is know to be able to "remember patterns"
- Hard to train, does not converge appart for very simple situations –> Few patterns
- Slow to sample if the system is very big.



RBM and EBM

Hopfield model and Restricted Boltzman Machines

To make it more easy to train $\rightarrow$ decouple the spins :

- Model on an Acyclic Directed Graph $\rightarrow$ *Potts model*
- Lattent variables

$$
\begin{aligned}
p(v) &= \frac{e^{\sum_i a_i v_i} \prod_\mu \int dh_\mu \exp\left(\frac{-1}{2} \sum_\mu h_\mu^2 - \sum_i v_i W_{i\mu} h_\mu\right)}{Z} && (1)\\
E(v,h) &= \sum_i a_i v_i + \frac{1}{2} \sum_\mu h_\mu^2 + \sum_{i\mu} v_i W_{i\mu} h_\mu && (2)\\
p(v,h) &= \frac{e^{-E(v,h)}}{Z} && (3)
\end{aligned}
$$

RBM and EBM

Training an RBM

$$\begin{aligned}
\mathcal{L} &= \log(p_{model}(v,h)) - \log(p_{data}(v,h)) \quad (4)\\
&= \langle E(v_0,h_1) - E(v_\infty,h_\infty)\rangle\\
\frac{\partial \mathcal{L}}{\partial W_{i\mu}} &= \langle v_i h_\mu\rangle_{model} - \langle v_i h_\mu\rangle_{data} \quad (5)\\
\frac{\partial \mathcal{L}}{\partial a_i} &= \langle v_i\rangle_{model} - \langle v_i\rangle_{data} \quad (6)\\
\frac{\partial \mathcal{L}}{\partial b_i} &= \langle h_i\rangle_{model} - \langle h_i\rangle_{data} \quad (7)
\end{aligned}$$

At/out of equilibrium

Gibbs sampling

Use the metropolis hasting algorithm :

1. Take a configuration radomly (from the data)
2. Pick another configuration randomly $x_1 \rightarrow x_2$
3. Accept the configiration woth rate [1]
$\alpha = min\left(1, \frac{\pi(x_2)q(x_1|x_2)}{\pi(x_1)q(x_2|x_1)}\right)$

At/out of equilibrium

Gibbs sampling

Gibbs sampling $\rightarrow$ marginalise each degree of freedom :

- $\alpha(x_1, x_2|X^-) = min\left(1, \frac{\pi(x_2|x^-)q(x_1|x_2,x^-)}{\pi(x_1|x^-)q(x_2|x_1,x^-)}\right)$
- q can be any distribution that has support on the full phase space and respect detailed balance so $q(x_1|x_2x^-) = \pi(x_1|x^-)$ is satisfactory.

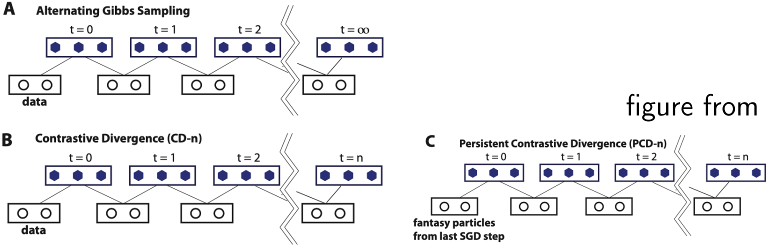


figure from [2]

At/Out of equilibrium Boltzmann machines

Algorithm for equilibrium sampling :

1. Start from an initial point and make many Gibbs steps until equilibrium
2. Repeat for each particle in the batch
3. Compute the gradients on the batch

- Most of the time this procedure is too slow. And we are not sampling the equilibrium distribution.
- Most of the time 1-5 steps of Gibbs sampling are sufficient for each predictions

Training an RBM in and out of equilibrium

Training at equilibrium

Set up :

- "And" data → 111, 100, 010, 000
- Visible layer $N_v = 3$
- hidden layer $N_h = 3$
- Discrete hidden and visible variables in $\{0, 1\}$

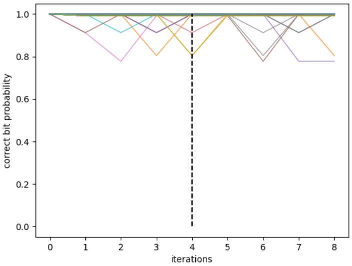


Figure – valid state retention

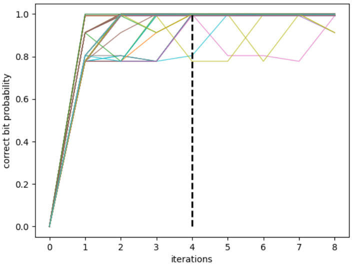


Figure – error state correction

At and out of equilibrium

Trainning and sampling an out of equilibrium RBM - MNIST

Pseudo-continuous RBM from discrete RBM :

- $N_v = 28 \times 28$ discrete visible variables
- discrete hidden layer with $N_h = 500$
- using bit probability as continuous output
- using random sampling as continuous input for training

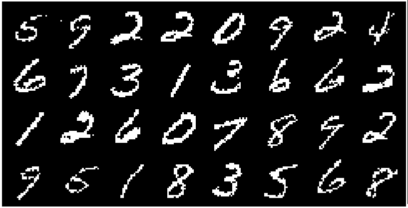


Figure – Real data

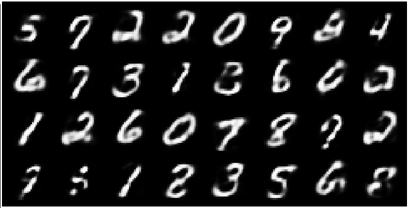


Figure – Generated data (pixel probabilities)

At and out of equilibrium

Trainning and sampling an out of equilibrium RBM - MNIST

- Initialise with random noise
- The Machine tends to generate something that "looks like" the data

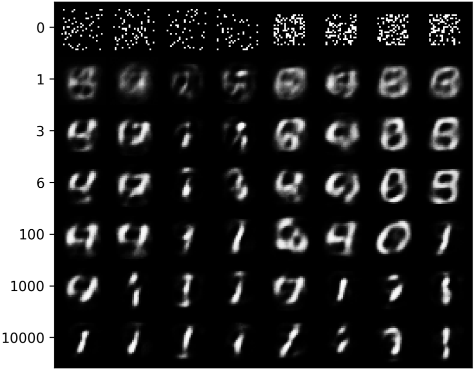
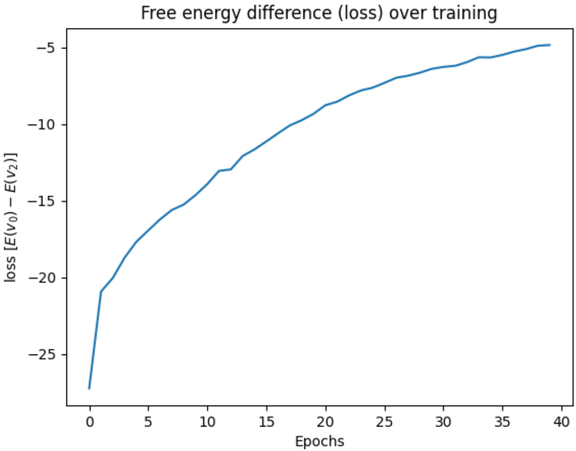


Figure – Generated data (pixel probabilities) - starting from random noise and over iterations

At and out of equilibrium

Trainning and sampling an out of equilibrium RBM - MNIST Fashion

$\Delta_E = E(v_0) - E(v_n) \rightarrow 0$ because v_0 is becoming a local optimum, thus v_0 is a stable point and $\Delta_E = 0$



At and out of equilibrium

Trainning and sampling an out of equilibrium RBM - MNIST Fashion

Real vs generated data (same setup as for MNIST) :

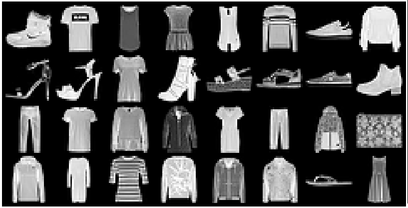


Figure – Real data

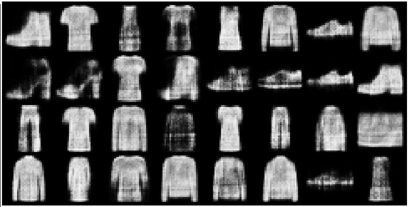


Figure – Generated data (pixel probabilities)

At and out of equilibrium

Trainning and sampling an out of equilibrium RBM - MNIST Fashion

The Boltzmann Machine (as Hopfield) network is able to memorise and retrive patterns. The image that were used for the training becomes attractors for the dynamics of the Boltzmann machine.

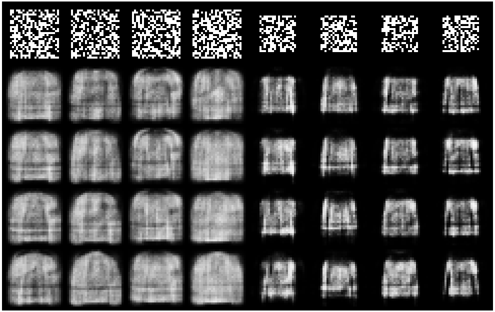


Figure – Generated data (pixel probabilities) - starting from random noise and over iterations

RBM as a langevin process

Mearning as a langevin process

There is some litterature on that : [3], [4], [5].

Theorem

Let be f_θ an observable conjugated with the parameter θ and let us suppose that there exist a set of parameter such that $\nabla\mathcal{L} = 0$ then if we generate sample with the exact same procedure as the training, it represent correctly the statistics of the data.

Theorem

In the same way if $\nabla\mathcal{L} = \epsilon$, if we sample the model in the same procedure as the training, we will have an error of order ϵ on the observable. For a sufficiently small ϵ it is possible to find $k^\dagger$ such that the error on $\langle f_{\theta^\dagger,i}\rangle$ vanishes.

Bibliography

[allowframebreaks]

Flannery Vertterling Teukolsky, Press.
Numerical reciepes in C++, third ed.
Cambridge, 2007.

Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab.
A high-bias, low-variance introduction to machine learning for physicists.
Physics Reports, 810 :1–124, May 2019.

Elisabeth Agoritsas, Giovanni Catania, Aurélien Decelle, and Beatriz Seoane.
Explaining the effects of non-convergent sampling in the training of energy-based models, 2023.

A. Decelle, G. Fissore, and C. Furtlehner.
Thermodynamics of restricted boltzmann machines and related