Advanced Computational Neuroscience - Week 3
—
Boltzmann Machine

Sam Suidman

March 12, 2022

1 Introduction

Thermodynamics and statistical physics are closely related we know now. One of the pillars of this description is the Boltzmann-Gibbs distribution, which gives the probability that a system is in a certain state and is based on the energy. This distribution can also be used to describe a system of N neurons that are either firing 1 or not 0. The unknown parameters in this model are the weights ω_{ij} and the thresholds θ_i. This is also called the Boltzmann machine. Using this distribution we can calculate the likelihood and minimize this with respect to these parameters using gradient descent. For small N this can be done exact, while for bigger N this is done using mean field (MF) theory with linear response (LR) correction and Metropolis Hastings (MH). We use a salamander retina data set containing the neuron activity (0,1) for each point in time.

2 Theory

Everything starts with the Boltzmann-Gibbs distribution.

$$p(s^\mu|\omega,\theta) = \frac{1}{Z}\exp\left[\frac{1}{2}\sum_{i,j=1}^{N}\omega_{ij}s_i^\mu s_j^\mu + \sum_{i=1}^{N}\theta_i s_i^\mu\right] \tag{1}$$

Where we can recognise the energy of a state as:

$$E(s^\mu) = \frac{1}{2}\sum_{i,j=1}^{N}\omega_{ij}s_i^\mu s_j^\mu + \sum_{i=1}^{N}\theta_i s_i^\mu \tag{2}$$

Where $s^\mu = (s_1^\mu, ..., s_N^\mu)$ with $\mu = 1, ..., P$ the patterns which are the time steps. Z can be calculated by summing over all patterns. This value of Z can be calculated for small N, but not if it is big. Every neuron has 2 states, 0 and 1. This means that there are in total 2^N patterns possible, which grows exponentially and takes to much time to calculate for each iteration in the learning process.

$$Z = \sum_{\mu=1}^{2^N}\exp\left[\frac{1}{2}\sum_{i,j=1}^{N}\omega_{ij}s_i^\mu s_j^\mu + \sum_{i=1}^{N}\theta_i s_i^\mu\right] \tag{3}$$

We want to minimize L, given below, with respect to ω_{ij} and θ_i.

$$L(\omega,\theta) = \frac{1}{P}\sum_{\mu=1}^{P}\log p(s^\mu|\omega,\theta) \propto \frac{1}{2}\sum_{i,j=1}^{N}\omega_{ij}s_i^\mu s_j^\mu + \sum_{i=1}^{N}\theta_i s_i^\mu \tag{4}$$

This leads us to the following 2 Formulas.

$$\frac{\partial L}{\partial \theta_i} = \langle s_i \rangle_c - \langle s_i \rangle = 0 \tag{5}$$

$$\frac{\partial L}{\partial \omega_{ij}} = \langle s_i s_j \rangle_c - \langle s_i s_j \rangle = 0 \tag{6}$$

Here the clamped statistics are given by

$$\langle s_i \rangle_c = \frac{1}{P} \sum_{\mu=1}^{P} s_i^\mu \tag{7}$$

$$\langle s_i s_j \rangle_c = \frac{1}{P} \sum_{\mu=1}^{P} s_i^\mu s_j^\mu \tag{8}$$

Where P are the observed patterns, so in our case the data of the salamanders retina. You can see that this does not depend on the parameters ω and θ at all and thus stay the same during learning. $\langle s_i \rangle$ and $\langle s_i s_j \rangle$, however, do depends on the parameters and on all possible states. They are given by:

$$\langle s_i \rangle = \sum_{\mu=1}^{2^N} s_i^\mu p(s^\mu | \omega, \theta) \tag{9}$$

$$\langle s_i s_j \rangle = \sum_{\mu=1}^{2^N} s_i^\mu s_j^\mu p(s^\mu | \omega, \theta) \tag{10}$$

You can already see the 2^N term here and this is again included in the $p(s^\mu|\omega,\theta)$ term. This means that for big N this is not a feasible method when we want to minimize L as given in Formulas 5 and 6 using gradient descent in an iterative process with learning rate η:

$$\theta_i^{new} = \theta_i^{old} + \eta(\langle s_i \rangle_c - \langle s_i \rangle) \tag{11}$$

$$\omega_{ij}^{new} = \omega_{ij}^{old} + \eta(\langle s_i s_j \rangle_c - \langle s_i s_j \rangle) \tag{12}$$

Luckily there is a way to approximate $\langle s_i \rangle$ and $\langle s_i s_j \rangle$ using mean field theory (MF) and linear response correction (LR). In MF $\langle s_i \rangle$ can be set equal to m_i, which is the solution to the equation

$$\langle s_i \rangle = m_i = \tanh\left[\sum_{j=1}^{N} \omega_{ij} m_j + \theta_i\right] \tag{13}$$

Which is found also via gradient descent:

$$m_i^{new} = m_i^{old} + \eta \left[\tanh(\sum_{j=1}^{N} \omega_{ij} m_j + \theta_i) - m_i\right] \tag{14}$$

This can be used to to estimate via LR:

$$\langle s_i s_j \rangle = \chi_{ij} + m_i m_j \tag{15}$$

with

$$\chi_{ij} = \left[\frac{\delta_{ij}}{1 - m_i^2} - \omega_{ij}\right]^{-1} \tag{16}$$

The MH algorithm works as follows. It starts with a random state s^μ with zeros and ones and calculates the energy as given in equation 2. Then it flips one of the N values and calculates this energy. If this energy difference ΔE is negative this means the energy drops and this state is accepted. If it is not then the state is rejected. In this case a temperature T is chosen as 10%, which means that the rejected new state will be chosen nonetheless with a 10% probability. This is done T_{MH} times for one iteration of equations 11 and 12. $\langle s_i \rangle$ is then the average of all the T_{MH} states.

Then there is also a way to calculate ω_{ij} and θ_i at once without iterations by solving the system of equations given from Formulas 5 and 6. When using $m_i = \langle s_i \rangle = \langle s_i \rangle_c$ and $C_{ij} = \langle s_i s_j \rangle_c - \langle s_i \rangle_c \langle s_j \rangle_c$ the solution is given by:

$$\theta_i = \tanh^{-1}(m_i) - \sum_{j=1}^{N} \omega_{ij} m_j \tag{17}$$

$$\omega_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - (C^{-1})_{ij} \tag{18}$$

Now it can be that C is not invertible, that is why you need to add some extra values ϵ on the diagonal such that $C_{new} = C_{old} + \epsilon 1$.

3 Results

There have been executed 2 main calculations. The exact Boltzmann Machine (BM) and the mean field approximation including linear response correction (LR). For both ways there are chosen $N = 10$ neurons at first. The clamped statistics $\langle s_i \rangle_c$ and $\langle s_i s_j \rangle_c$ are calculated upfront. θ and ω are chosen randomly with the constraint that ω is symmetric.

Now for the case BM case there are $2^{10} = 1024$ possible states which all give another probability $p(s^\mu | \omega, \theta)$. Then equations 9 and 10 are used to find $\langle s_i \rangle$ and $\langle s_i s_j \rangle$. After they are calculated θ and ω are updated according to equation 11 and 12. The change in weights ω and thresholds θ becomes smaller and smaller as is shown in Figure 1.

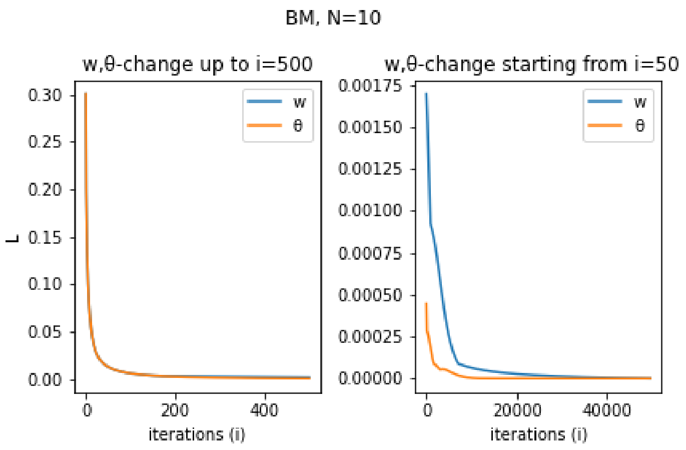

Figure 1: Change in weights ω and thresholds θ over iterations. **Left** Up to iteration 500. **Right** Starting from iteration 500.

The likelihood as given in equation 4 over iterations is shown in Figure 2 for BM and $N = 10$. This likelihood is set such that it ends at $L = 0$. This is an arbitrary choice, because there is only a proportionality equation that can be used. s_{ij}^{μ} is here the $\langle s_i \rangle$ of that iteration

For the MF (and LR) case equation 13 is found via the updating rule of equation14. After this equation 15 is used to update θ and ω via the gradient descent rule of equations 11 and 12. To compare this method with the BM, Figure 3 shows the likelihood L over iterations for the MF method.

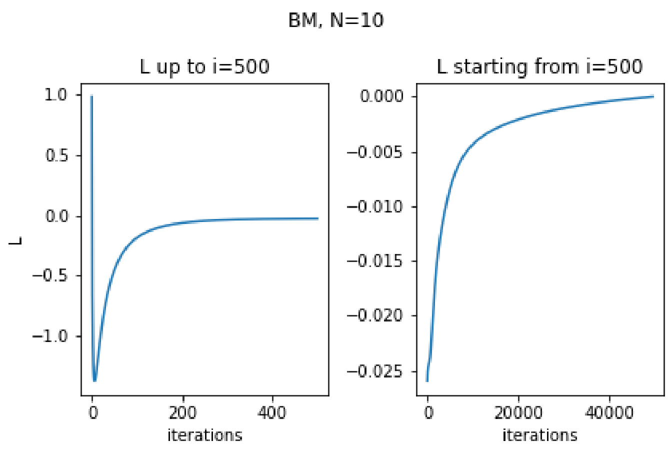

Figure 2: Likelihood over iterations with some constant chosen such that $L(t_{end}) = 0$ for BM and $N = 10$. **Left** Up to iteration 500. **Right** Starting from iteration 500.

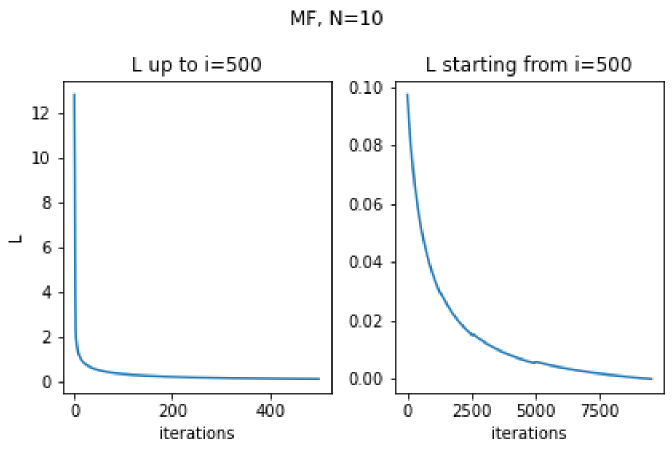

Figure 3: Likelihood over iterations with some constant chosen such that $L(t_{end}) = 0$ for MF and $N = 10$. **Left** Up to iteration 500. **Right** Starting from iteration 500.

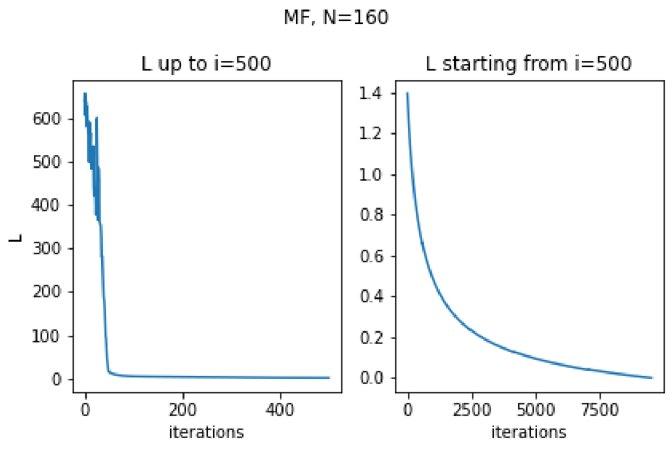

Figure 4: Likelihood over iterations with some constant chosen such that $L(t_{end}) = 0$ for MF and $N = 160$. **Left** Up to iteration 500. **Right** Starting from iteration 500.

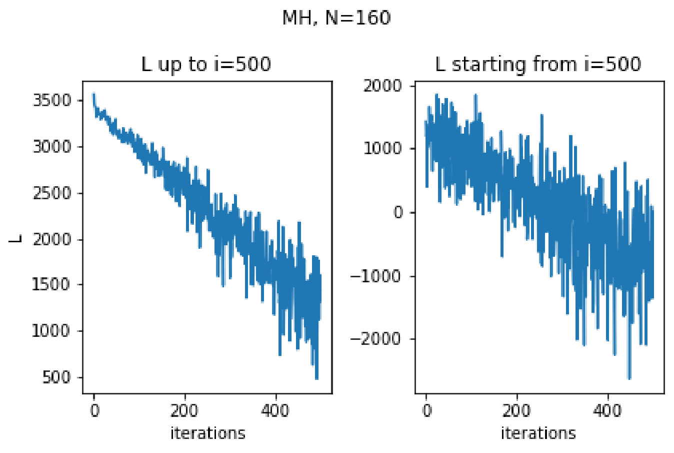

Figure 5: Likelihood over iterations with some constant chosen such that $L(t_{end}) = 0$ for MH, $N = 160$ and $T_{MH} = 1000$. Also there is a temperature T such that in 10% a positive ΔE will nonetheless cause a flip. **Left** Up to iteration 500. **Right** Starting from iteration 500.

Now, for $N = 160$ this is done again for the MF method, but also for MH sampling, where $T_{MH} = 1000$. Plots of the likelihood are shown in Figure 4 and 5. I could not get the MH algorithm to work the right way, so as can be clearly seen, the likelihood does not show any nice convergence, but rather random behaviour.

The last case is where the thresholds θ and weights ω are calculated at once. The likelihood for different ϵ is shown in Figure 6.

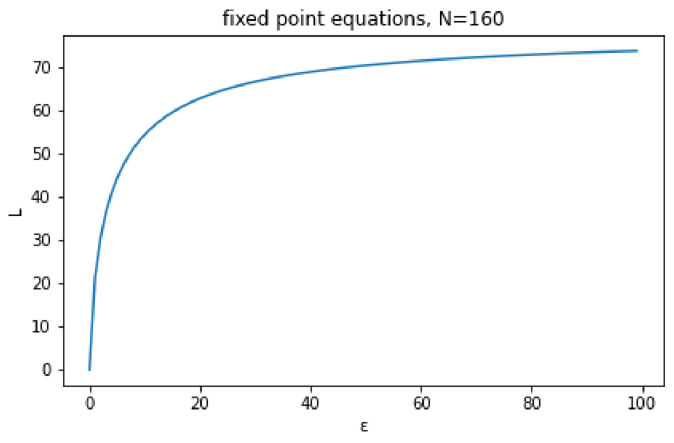

Figure 6: Likelihood with some constant chosen such that $L(\epsilon = 0.000001) = 0$ for the case where ω and θ are calculated at once and $N = 160$.

4 Conclusion

We looked at different methods (BM,MF,MH) and see that for the algorithms that the working algorithms of BM and MF converge. For MH the algorithm I implemented did not work, so also the likelihood did not converge. For the BM algorithm this happens differently than for MF. For BM it drops and then converges upward to a fixed value, for the MF it only drops, but for MF it also works for bigger N. For the fixed point equations we see that the bigger the value of ϵ, the smaller the change in L.

5 Appendix

The code to run this can be found in the attached Jupyter Notebook and python file.