"From Spin Physics over Deep Learning to Social Implementation - the Shortest Route"
Exercises for
Lecture B
From Spinglasses to Neural Networks
H.-G. Matuttis

# Contents



These exercises are for you: Familiarize yourself with the concepts explained in the lecture, play around and see what happens when you make changes of the program.
And better make copy of the original program, in case something goes wrong.

# 1 Statistical mechanics of Boltzmann-Distribution and Thermodynamic weight

Statistical mechanics, wants to obtain distributions for density, probability, energy etc. . If you are not comfortable with the idea of thermal equilibrium, thermodynamic weight, Boltzmann-distribution etc., these exercises are for you.

## 1.1 Picture from the live of a thermal equilibrium

Run the program `BarometricgasEquilibriumFromAllStates.m` and see how in every frame, particles are in a different position, but the average densities distribution (with "thermal fluctuations"=stochastic error bars) stays the same. The energy for that system

is the position energy of the atoms together with the gas pressure, all from the ideal gas equation

$$pV = NRT,$$

with pressure $p$, volume $V$, Avogadro constant $N$, Gas constant $R = 8.3\ldots$ J/(K mol) and Temperature $T$. You see the density distribution of air molecules from height zero to 20 km.
At each timestep, at a particle is randomly taken from one hight (one "state") and is put back at another height or the same height, governed by the "master equation" (which is the Boltzmann distribution). Particle $i$ is removed and re-inserted omewhere else according to the absolute probability $p = \exp(-E_i/k_bT$ (Boltzmann factor) due to position energy $E_i$.
When you look at the listing, you see a variable p, and it is left open whether this corresponds to the pressure or the probability to find the particle, which in the case of the ideal gas equation is the same.

## 1.2 Picture from the live of a thermal equilibrium

Run the program `BarometricgasEquilibriumRelProb_MarkovChainMonteCarlo.m`, which does nearly the same as `BarometricgasEquilibriumFromAllStates.m`, only that the particle is reinserted no according to the "absolute probability" $p = \exp(-E_i/k_bT$ but according to relative probability

$$p = \exp\Big(-\underbrace{\Delta E}_{E_f - E_i}/k_bT\Big)$$

of the initial state $i$ and the final state $f$. This program runs faster for large heights where the densities become very low.

## 1.3 Equilibration

Take one of the barometric gas programs, make a copy, and replace the initialization with something unphysically crazy (all particles in the top or lower 20% of the system height). Run the program an observe the "equilibration process", i.e. how the particles reach the thermal equilibrium after some time.
Generally observe how there are more states / particles below than above: That is the fact that there are more states below than above. For very high temperatures, the densities may look constant.
**Warning! Grave Programming error!!!**
In the MATLAB-programs of todays and yesterdays Ising-simulations and in yesterdays Simulated annealing, there is a typo which causes the stripes: the correct lines read

```
sumspin=ising(ixright(ix),iy)+ising(ixleft(ix),iy)+...
        ising(ix,iyup(iy))+ising(ix,iydo(iy));
```

and **not**

```
sumspin=ising(ixright(ix),iy)+ising(ixleft(ix),iy)+...
        ising(ix,iyup(iy))+ising(ix,iydo(ix));
                                          ^^
```

The second coordinate must always be an x-coordinate! I found this bug only because I used this kernel several times in different programs.

### 1.4 Probability states and energy

Observe in particular that there are more particles in the lower than in the upper states

# 2 Critical State

The critical state is the state of a phase transition, i.e. the temperature where water starts boiling, 100° C. In the 19th century, with steam engines, this was "critical" because the steel for the boilers was not good, so there was always a "critical" danger of the boiler exploding.

## 2.1 Fractality

Google "ordered fractal" and "disordered fractal" and observe how "fractals" are the same shape in each scale, like this here:
`https://www.youtube.com/watch?v=Zh4oVYty61M`
Zooming in will get your closer, but you will see the same shape again.

## 2.2 Fractality of the critical state

At boiling, at the "critical temperature" of 100° C, water bubble in a steam bubble in a water bubble and so on ... Each snapshot of the steam-water-mixture would be a fractal. Use the ising-program from yesterday, make the system size large $100 \times 100$ or more, but it will take time. Set the temperature to the critical temperature to `kbT=2.2691853` and see the fractality evolves: Black stripe in a white stripe in a black stripe .... (depending on how many spins you use), or black cluster in a white cluster in a white cluster ....

## 2.3 Second order phase transition

Watch on youtube how in `https://www.youtube.com/watch?v=Dq7DeuDaCEM` the ice is melting (look at the thermometer column, bit small) while the temperature is not changing. This is typical for a "phase transition of second order": The temperature does not change while the state changes (from fluid to gas, from solid to fluid etc.). The reason is the large heat capacity, which prevents a temperature change.

# 3 Spin Glasses

You can run the Ising spinglas simulations with `IsingSpinglasSimulation2022.m` in directory `IsingSpinglas.m` and
the Sherrington Krikpatrick model with
`GaussSpinglasInfiniteRange2022.m` in directory `GaussSpinglass.m`. You can find the "critical temperatures" (takes some time, better do it at home...): In contrast to the "ordered," "ferromagnetic" Ising model, the low temperature for the (disordered) spin glasses has no magnetic ordering, but the energy, the heat capacity (variance of the energy, divided by the temperature) as well as the susceptibity (variance of the magnetization) behave similar to the Ising model:

You will find a maximum of the susceptibity and the heat capacity at the critical temperature. At the critical temperature, the energy goes from the "high-temperature" value to the "groundstate energy value".
Amazingly, it does not matter whether the interaction $J_{ij}$ is $\pm 1$ as in the Ising Spinglas model order whether $J_{ij}$ is Gaussian distributed as in the Sherrington Kirkpatrick model: The "disorder rules" the system. Though we have different transition temperatures, the basic behavior is the same: At the critical temperature, the energy goes down, the heat capacity and susceptibity have a maximum.

# 4 Power iteration

Play around with the power-iteration in
`PowerIteration`
`PowerIterationEigenvector`
and observe the convergence to eigenvector of the dominant eigenvalue. In case where all eigenvalues are the same, the convergence should be to the closest eigenvector.

# 5 Hopfield Neural Network

## 5.1 Program structure

Look at the initalization of the Hopfield Neural Networks of
`ABCextInputNNfromGaussSpinglasInfiniteRange2022.m` in
`NN_Berkelium34x34NoisyLetters` and
`Main_NewfontABCextInputNN2022.m` in `Blockschrift34x34Handwritten`. See how the the image is read in, converted into a matrix and this matrix is converted with some low level image processing into pieces for the individual letters.
Observe next how the blocks for the lectures are unrolled into vectors, the vectors written in a rectangular matrix and from this, the Hopfield-interaction is formed as an "outer product". Then the diagonal elements are set to zero, and then non-zero eigenvalues are equalized to the largest eigenvalues.

## 5.2 Playing around with Berkelium

This program uses several figure-windows to show you the distribution of the interaction `Jij`, the letters, the original graphics which was used, etc.
For `ABCextInputNNfromGaussSpinglasInfiniteRange2022.m` in
`NN_Berkelium34x34NoisyLetters`:

1. Change the input pattern (number of the letter in line 56, at the moment it is set to 1 for A) and see how different letters are recognized ... or not.

2. Change the noise level by changing `if rand<.34` in line 66. Higher noise level should make make recognition more difficult, convergence to the "correct" pattern should become less likely.

3. Comment out (don't delete) the equalization of the largest eigenvalue in line 46 to line 50: Probably, most of the time, no matter what the input is, the final pattern will be the same. (I did not try that one out).

### 5.3 Playing around with Blockschrift

This program uses several figure-windows to show you the distribution of the interaction `Jij`, the letters, the original graphics which was used, etc. `Main_NewfontABCextInputNN2022.m` in
`Blockschrift34x34Handwritten` has an interface where you can draw letters: Use the mouse to draw in (not out of) the graphics window with the grid, each move must be terminated by a mouseclick, and press **return** to terminate the input.
Dont click other MATLAB graphics windows, else the output will change to that window.
Observe how the Hopfield Neural Network converges to the correct letter - or maybe not.
You can try to different letter shapes, and you can change the input resolution
`nquad2=34`
`nquad1=34`
to e.g. $54 \times 54$. You will see that this does not necessarily improve convergence. Especially and "S" is very difficult to draw.
If you feel confident, replace the single-spin update in the current implementation with a power-iteration where you multiply the current vector `isvec` from the right to the interaction matrix
`isvec_new=J*isvec`
and round the result to
`isvec_new=sign(isvec_new)`
(you can get rid of the double loop with

```
for ixrep=1:lx
    for iyrep=1:ly
```

) and see whether the convergence is better or worse for single spin updates where only one spin is changed at a time.

## 6 Fitting

Play around with the correct fits in
`FittingPolynomialsToNoisyData.m`
and then try overfitting with `OverFittingPolynomialsToNoisyData.m`
or underfitting with `UnderFittingPolynomialsToNoisyData.m`. You can also program higher order polynomials and fit them to higher order. You will see that if you fit a parabola with 3rd order, the leading coefficient $a_3$ for $a_3 x^3$ will be nearly zero.
Never show data in a presentation where you use overfitting, only use the numer of coefficients which are reasonable.