From Spin systems to Hopfield Neural Networks

Hans-Georg Matuttis
Department of Mechanical and Intelligent Systems Engineering

Starting from disordered Spinglass interactions with "infinite range", we will construct $\pm J$ "spinglas" interactions so that a prescribed pattern of up- and down-spins can appear as pattern in the groundstate of the spin system, for example numbers or letters , , ... Even when the initial state is totally random, the pattern (or its opposite) will appear when the system is cooled down as ``groundstate" – or the exactly opposite state , , ... In the next step, we will construct "Hopfield Neural Networks" with $\pm J$ interactions so that several uncorrelated (dissimilar) patterns like X, M and B become "degenerate" groundstates of the "Spinglas", i.e. they have all the same energy value. These respective states can be reached by ``cooling down'' the spin system from initial states which close to the respective pattern. In the case of correlated (similar) patterns like E, F, H, more has to be done: They are so close that their combined state / overlap "⊦" will have a lower energy than all other states, so that any input will converge to "⊦". We can prevent this by manipulating the groundstate energy, setting them equal for all inputted patterns. In the case of e.g. 24x24 pixels, we can recover all capital letters of the alphabet if the input was close enough. Nevertheless, the recognition letters from other fonts will fail. This shows the need for a more general approach, where a letter is not represented by only a single pattern, but by a database of possible input patterns for 1, 2, 3 .... This approach is a "deep learning Neural Network" which will be discussed in Part C of the Workshop.

Finally, we will discuss how the number of parameters influences data-modeling: Fitting of parabolas $y = a + bx + cx^2$ will not become better when we increase the number of parameters so that we have $y = a + bx + cx^2 + dx^3 + ex^4$ ...., because d, e, ... should all be zero: If not, they will represent a shape which is not at all a parabola, which is called "overfitting". The same phenomenon can occur with Neural Networks: In principle, for our Hopfield network with $lx \times ly = N$ spins / pixels, we have N^2 interactions, even if we have only 26 input patterns (capital letters) in total. This leads to a very ragged ``energy landscape'' and a rather erratic convergence of inputs which deviate too much from the programmed input patterns. (The power of "Deep learning" with improved convergence compared to Hopfield neural networks results from the possibility to increase the alternatives for each input pattern while simultaneously reducing the total number of parameters.)

Exercises: Basics of linear algebra (outer products, numerical diagonalization) will be practiced to manipulate the "energy function" for given input patterns. There will be an opportunity to play with the interactions and ``hand written'' (via MATLAB's ginput-interface) or "randomized" (via random numbers) input patters to get a firsthand experience of success or failure of the pattern recognition: In particular, convergence patterns which is not letters (and not in the database) is possible. Increasing the number of spins / pixels in the representation will not help. Parameter fitting of parabolas will show how few redundant parameters do not improve the quality of the fit, while many more parameters result in shapes which are not parabolas at all. (This will motivate the following part C on Deep Learning, where an approach with more input data but less free parameters is introduced.)