Proceedings of 1993 International Joint Conference on Neural Networks

A Stochastic Network with Rotor Neurons

Lucas Parra and Gustavo Deco
Siemens AG, Corporate Research and Development, ZFE ST SN 41
Otto-Hahn-Ring 6, 8000 Munich 83, Germany

Abstract:

We define a new network structure to realize a continuous version of the Boltzmann Machine (BM). Based on Mean Field (MF) theory for continuous and multidimensional elements named Rotors, introduced by Gislén and Peterson we derive the corresponding MF learning algorithm. Simulations demonstrate the learning capability of this network for continuous mappings.

1 Introduction

The classical BM is a well known approach to stochastic neural networks [1]. It has been designed to generalize the original recurrent Hopfield model to a system with hidden units, which can build an internal representation of the desired mapping task. It has been used mainly for pattern completion, encoding problems etc. The classic BM suffers from two basic disadvantages. First it is only able to carry out binary mappings because the model is based on binary spin states; second the learning process is very time-consuming because each single recall requires a complete annealing process. The second disadvantage is reduced by the MF theory, were the stochastic annealing process is approximated by a fast deterministic algorithm [2]. For the remaining continuity problem we present a solution using a generalization of the Spin MF theory to continuous multidimensional elements, which are commonly called Rotors [3].

2 Model description

The proposed BM consists of stochastic multidimensional real valued unit vectors $\boldsymbol{S}_i \in R^d; |\boldsymbol{S}_i| = 1; i = 1 \ldots n$. They can be understood as Rotors which were introduced by Gislén and Peterson in a very general manner [3]. They considered the task of minimizing an energy function $E(\boldsymbol{S}_1 \ldots \boldsymbol{S}_n)$ using MF equations in an annealing process. They introduced new mean field variables $\mathbf{U}_i$ and $\mathbf{V}_i$ not constrained to unity and a temperature T. In the zero temperature limit the variables $\mathbf{V}_i$ can be understood as the thermal average of the Rotor states, $\boldsymbol{V}_i = \langle \boldsymbol{S}_i \rangle$. We allow for two neuron interaction independent in each direction and define therefore the following energy function,

$$E = -\frac{1}{2}\sum_{ijkl} S_{ik} W_{ikjl} S_{jl} = -\frac{1}{2}\sum_{ij} \boldsymbol{S}_i \cdot \boldsymbol{W}_{ij} \cdot \boldsymbol{S}_j \tag{2.1}$$

In the first notation the indexes $i, j = 1 \ldots n$ enumerate the Rotors, and the indexes $k, l = 1 \ldots d$ denote the different dimensions. In this case the cited MF equations read,

$$V_i = \frac{U_i}{|U_i|} F(|U_i|) \qquad U_i = -\frac{1}{T} \sum_j W_{ij} \cdot V_j \tag{2.2}$$

F is a kind of sigmoidal function and is defined by using the modified Bessel functions I_m,

$$F(u) = \log I_{(d-2)/2}(u) - \frac{d-2}{2} \log(u) \tag{2.3}$$

The iterative update of equation (2.2) defines the dynamics of the system. In one dimension these equations reduces to the original MF equations of the discrete BM [2]. To guarantee convergence of the dynamics we demand the usual symmetry conditions in i, j and simultaneously in k, l, i.e. $W_{ikjl} = W_{jlik}$. The equations (2.2) describes fixed-points of the energy function (2.1). The system relax towards these fixed-points while one gradually decrease the temperature. We proved convergence of this dynamics in the case of $2\|W\|/T < 1$. We also showed that there exists a Liapunov function for the corresponding continuous time partial differential equation analogous to the one presented by Hopfield [4].

3 Learning rule

As usual the units are divided into visible and hidden units. The hidden units have no connection to the outside world. The visible units sometimes can be separated further into input and output units. During a free phase the system relaxes according to (2.2)-(2.3) with inputs kept at fixed values. In a clamped phase the output units are fixed as well. The learning rule is a gradient descent minimizing the relative entropy H. In contrast to the binary state space in the original BM in the derivation one has to consider a continuous space, hence the traces change as,

$$\sum_{\{S_i\}} \rightarrow \int \prod_i^n dS_i \tag{3.1}$$

This leads to the following gradient expression,

$$\Delta W_{ikjl} = -\eta \frac{\partial H}{\partial W_{ikjl}} = \eta \beta \left[\overline{\langle S_{ik} S_{jl} \rangle}_{clamped} - \overline{\langle S_{ik} S_{jl} \rangle}_{free} \right] \tag{3.2}$$

where the brackets denote the thermal mean and the bar denotes the average over the training patterns. With the same approximation as in [2],

$$\langle S_i S_j \rangle \approx \langle S_i \rangle \langle S_j \rangle \tag{3.3}$$

we can write the MF learning rule as,

$$\Delta W_{ikjl} = \eta \beta \left[\overline{(V_{ik} V_{jl})}_{clamped} - (V_{ik} V_{jl})_{free} \right] \tag{3.4}$$

4 Simulations

The first aim of the simulation was to prove the practicability of the proposed BM. In some preliminary experiments we confirmed that the MF equations (2.2) converge for every temperature in a few updates cycles, even with connections which are not symmetric in k, l. We use the same annealing process as in the original MF learning. At high temperature the Rotor values move around the origin of their state space. By decreasing the temperature the norm of the Rotors increases until some freezing temperature is reached were $|V_i| \approx 1$ and the values stay fixed. We observed in the experiments that the freezing temperature is correlated with the connection strengths ($\|W\|/T_{freeze}$ is of order one). This gives some guideline to select the temperature schedule. Starting near above the freezing temperature one decrease it slowly. In our experiments we start at temperature 1.0 and decrease it with factor 0.85 until we reach 0.001. To implement a continuous mapping we have to use at least two dimensions. Because of the normalization condition we need (d+1)-dimensional units to code d-dimensional signals. In the first preliminary experiments we used two-dimensional units. To verify the learning algorithm we tested as a discrete mapping the XOR problem. With similar parameters we got the same result as in the original work of Peterson [2]. We also checked the capacity of learning simple continuous mapping like the one-dimensional sinus function (see FIGURE 1). This is the principal result of these paper. The net was trained with 20 sample points it and solved the task nearly perfect (0.9% error).

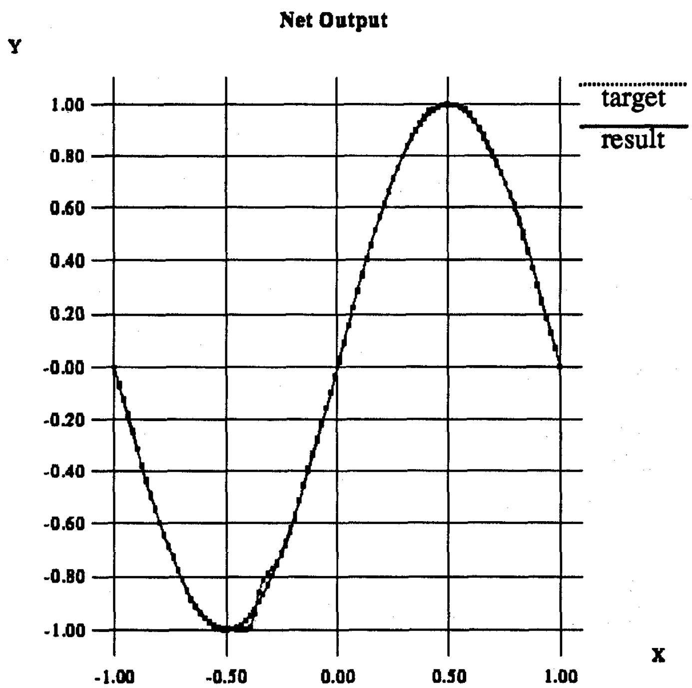

FIGURE 1 : Continuous BM with 10 two-dimensional units: 8 hidden, one input and one output unit. Right: X and Y denote one dimension of the input and output unit respectively.

We were expecting from the system to find the proper energy landscape, where the attraction points are continuously connected. On the contrary, we observe a tendency to perform discontinuous mappings. The location of the discontinuities is very sensitive with respect to the connection strengths. This explains the strong peaks of the learning process in FIGURE 2. In order to compensate these peaks we used the learning constant schedule suggested by Silva and Almeida [5]. All in all this kind of recurrent net has the ability to perform both continuous and discontinuous mapping. For this we expect good performance especially in piecewise continuous mapping.

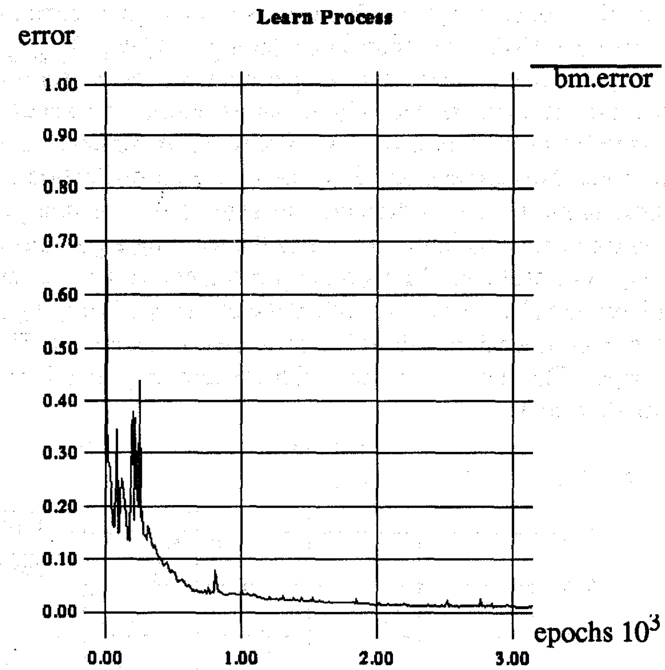

FIGURE 2 : Learning process with 20 sample points o the sin-function.

5 References

[1] D. Ackley, G. Hinton, T. Sejnowski: A Learning Algorithm for Boltzmann Machines. Cognitive Science 9, (1985) 147-169

[2] C. Peterson, J. Anderson: A Mean Field Theory Algorithm for Neural Networks, Complex Systems 1 (1987) 995-1019

[3] L. Gislén, C. Peterson, B. Söderberg: Rotor Neurons: Basic Formalism and Dynamics, Neural Computation 4, (1992) 737-745

[4] J.J. Hopfield: Neurons with Graded Responses Have Collective Computational Properties Like Those of Two-State Neurons. Proceedings of the National Academy of Science, USA 81, (1984) 3088-3092. Reprinted in Anderson and Rosenberg (1988).

[5] F. Silva, L. Almeida: Speeding up Backpropagation. Advanced Neural Computers, Elsevier Science Publischer B.V. (North-Holland) 1990. 151-158