Pergamon

Neural Networks, Vol. 8, No. 3, pp. 375–385, 1995

0893-6080/95 $9.50 + .00

0893-6080(94)00074-3

*CONTRIBUTED ARTICLE*

# Continuous Boltzmann Machine With Rotor Neurons

LUCAS PARRA[1,2] AND GUSTAVO DECO[2]

[1]Ludwig–Maximilian-Universität, Institut for Medical Optics and [2]Siemens AG, Munich, Germany

(*Received* 10 *September* 1993; *revised and accepted* 1 *July* 1994)

**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," we derive the corresponding MF learning algorithm. Simulations demonstrate the learning capability of this network for continuous and piecewise continuous mappings. The rotor neurons are specially suited for cyclic problems of arbitrary dimension.*

**Keywords**—Boltzmann machine, Mean field theory, continuous mapping, Rotor neurons.

## 1. INTRODUCTION

The classical Boltzmann machine (BM) is a well-known approach to stochastic neural networks (Ackley, Hinton, & Sejnowski, 1985). 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-learn step requires the calculation of the mean values of the stochastic state variables. Thus, one has to evaluate the trace over the entire state space, or to perform some Monte Carlo Simulation. This second disadvantage is reduced by the mean field (MF) theory, where the stochastic annealing process is approximated by a fast deterministic algorithm (Peterson & Anderson, 1987). 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" (Gislén, Peterson, & Södeberg, 1992). Similar units called phasors were presented and studied by Noest (1988). These elements were complex valued and thus restricted to the two dimensions of the complex space and used in a Hopfield-like net structure without hidden elements. Similarly to this approach with complex-valued connection strengths, Mozer et al. (1992) included also two-dimensional hidden neurons. We present in this paper the multidimensional generalization of this model. The units we are going to use are called rotors because they can take values on a multidimensional unit sphere. They naturally will be suited for cyclic problems or problems where directions in two, three, or more dimensions come in.

In Section 2 we introduce the classic BM and rotor neurons. We present our multidimensional continuous model in Section 3. In Section 4 we report, among other simulations, results on piecewise continuous mappings. More detailed calculations on the convergence properties of the model are shown in the Appendix.

Acknowledgements: We want to specially thank Stefan Mießbach for numerous contributions in proving the convergence properties of the net dynamic and for advice concerning the "cornered rat" example. We are very grateful to Ingrid Gabler for supplying the experimental data for this same example.

Request for reprints should be sent to Gustavo Deco, Siemens AG, ZFE ST SN 41, Otto-Hahn-Ring 6, 81730 München, Germany.

## 2. UNDERLYING THEORY

### 2.1. The Classic BM

The BM consists of binary stochastic units $S_i \in \{-1, 1\}, i = 1 \ldots n$. They may be connected together with symmetric connection strengths $w_{ij} = w_{ji}$. The units are divided into visible and hidden units. The hidden

units have no connection to the outside world. The visible units sometimes are further separated into input and output units. We label the possible states of the hidden and visible units by $h$ and $v$, respectively. We use the same energy function as in the original Hopfield model,

$$E_{vh} = -\frac{1}{2}\sum_{ij} W_{ij} S_i^{vh} S_j^{vh}. \tag{1}$$

The name Boltzmann machine comes from the fact that the stationary distribution of state variables should be given by the Boltzmann–Gibbs distribution,

$$P_{vh} = e^{-\beta E_{vh}}/Z \tag{2}$$

$$Z = \sum_{vh} e^{-\beta E_{vh}} \tag{3}$$

where $Z$ is the partition function of the system and $\beta = 1/T$ is the inverse temperature. The index $vh$ expresses that the sum is to be performed over the entire state space. To guarantee a Boltzmann distribution of the state variables, one should insist on the principle of detailed balance for the transition probabilities $W(vh \rightarrow v'h')$ from a state $vh$ to a different state $v'h'$. In fact, this principle is satisfied by the so-called Glauber dynamic

$$W(S_i \rightarrow -S_i) = \frac{1}{2}\left(1 - S_i \tanh\left(\beta \sum_j w_{ij} S_j\right)\right). \tag{4}$$

Here we want to mention that to write this expression in this simple form the symmetry of the connection strengths has been used. The system is updated many times according to eqn (4) until it converges to a stationary distribution. That is one reason why the nondeterministic version of the BM is considered to be very slow.

2.1.1. *Annealing.* The most interesting mathematical property of the Boltzmann distribution is that at low temperatures it favors the states with low energy. Independent of the initial conditions and the path, the system is very likely to be found at the low-energy states. In the limit of zero temperature the probability to be in the global minimum is 1. The obstacle is that at low temperature the stationary distribution is reached very slowly in comparison to high temperatures. Therefore, it is common to use the annealing procedure, starting at a high temperature and decreasing it with some appropriate temperature schedule.

2.1.2. *Boltzmann Learning.* The basic idea of the BM is to introduce constraints by keeping some of the visible units fixed and letting the rest of the system relax to a state of low energy. The learning process should find the proper energy function where the patterns to be learned are represented by the most probable states at low temperature. In other words, the learning should adapt the connections $w_{ij}$ to give the visible units some desired probability distribution. In contrast to the Hopfield model, the system may use some internal representation of the visible patterns in the hidden units. The learning rule is a gradient descent method. The cross entropy is used as cost function. In this case it measures the difference between some desired probability distribution $R_v$ and the actual distribution $P_v$ of the visible units

$$H = \sum_v R_v \log \frac{R_v}{P_v}. \tag{5}$$

Here the probability distribution of the visible units, irrespective of the state $h$ of the hidden units, is given by

$$P_v = \sum_h P_{vh}. \tag{6}$$

The resulting learning rule

$$\Delta w_{ij} = -\eta \frac{\partial H}{\partial w_{ij}} = \eta\beta[\overline{\langle S_i S_j\rangle}_{\text{clamped}} - \langle S_i S_j\rangle_{\text{free}}] \tag{7}$$

involves only the mean values of the correlation of the state variables. The first term is the thermal mean value averaged over all presented patterns while keeping the visible units clamped. The second term is the thermal mean of the completely free running system. These two expressions can be understood as a Hebb and an anti-Hebb term. But calculating the accurate thermal average is even more time consuming than reaching the equilibrium distribution, so here the mean field approximation comes into consideration.

## 2.2. Mean Field Approximation

The aim of the MF approximation is to replace the time consuming stochastic procedure by a deterministic algorithm to calculate the desired mean values. There are different ways of deriving the corresponding equations. In the MF approximation itself, it is assumed that

$$\left\langle \tanh\left(\beta \sum_j w_{ij} S_j\right)\right\rangle \approx \tanh\left(\beta \sum_j w_{ij}\langle S_j\rangle\right). \tag{8}$$

That way one arrives at a set of equations for the mean values of the state variables,

$$\langle S_i \rangle = \tanh\left( \beta \sum_j w_{ij} \langle S_j \rangle \right). \tag{9}$$

The solution of this equations can be accomplished by iterative updating. In fact, these equations are the steady state solution of the corresponding partial differential equation of first order.

$$\frac{d}{dt} \langle S_i \rangle = -\langle S_i \rangle + \tanh\left( \beta \sum_j w_{ij} \langle S_j \rangle \right). \tag{10}$$

The iterative updating can be regarded as a discrete integration of this equation with time scale one. For this equation, Hopfield gives a Liapunov function that guarantees the convergence of the solution by demanding the connection strengths to be symmetric. A different way to derive eqn (9) is known as the saddle-point approximation. We will return to it in the next section. While applying the learning rule (7), one further approximation is used. The order of correlation and thermal averaging is changed,

$$\langle S_i S_j \rangle \approx \langle S_i \rangle \langle S_j \rangle. \tag{11}$$

At the end the resulting MF equations give a deterministic algorithm implementing the basic concept of global search of minimum energy in the recall phase.

### 2.3. Rotors

There are different approaches to generalizing the MF formalism to continuous and multidimensional units. Peterson (1987) first considered the case of real-valued but multidimensional units called "rotors" for a general energy function. Noest (1988) studied the Hopfield net in the case of imaginary-valued elements called "phasors" that were thus restricted to the two-dimensional complex space. Mozer et al. (1992 and related works) applied phasors in network models including hidden units. We want to generalize the deterministic discrete BM to the continuous multidimensional case using this time Peterson rotors with a quadratic energy function. We also focus on rotors because one may be interested quite naturally in three- or even higher-dimensional directional units. Peterson and Anderson (1987) introduced rotors in a very general manner. They are defined as multidimensional continuous unit vectors

$$\mathbf{S}_i \in \Re^d; \; |\mathbf{S}_i| = 1; \; i = 1 \ldots n. \tag{12}$$

He considers the task of minimizing an energy function $E(\mathbf{S}_1 \ldots \mathbf{S}_n)$. The starting point for the MF theory is the so-called partition function. The corresponding function $Z$ for rotors is defined as

$$Z = \int e^{-\beta E(\mathbf{S}_1 \ldots \mathbf{S}_n)} d\mathbf{S}_1 \ldots d\mathbf{S}_n. \tag{13}$$

The integration here is to be performed over the $n$ $d$-dimensional unit spheres. To realize the desired approximation, first introduce new mean field variables $\mathbf{U}_i$ and $\mathbf{V}_i$ and evaluate the integrals in $S_i$. The new mean field variables are not restricted to the unit sphere. That way the state space $|\mathbf{S}_i| = 1$ is replaced by a virtual space. In this virtual space the states are not restricted anymore to the unit sphere. On these states of course a different effective potential applies. It appears in the exponent and is different from the original energy for nonzero temperature,

$$Z \propto \int e^{-\beta E_{\text{eff}}} d\mathbf{V}_1 \ldots d\mathbf{V}_n d\mathbf{U}_1 \ldots d\mathbf{U}_n \tag{14}$$

$$E_{\text{eff}} = E(\mathbf{V}_1 \ldots \mathbf{V}_n) - T \sum_i \mathbf{V}_i \cdot \mathbf{U}_i + T \sum_i G(|\mathbf{U}_i|). \tag{15}$$

$G$ is defined by using the modified Bessel functions $I_m$:

$$G(u) = \log I_{(d-2)/2}(u) - \frac{d-2}{2} \log(u). \tag{16}$$

The variables $\mathbf{V}_i^0$ at the saddle point of the effective potential can be understood as the mean of $\mathbf{S}_i$. In the $T \to 0$ limit (see Appendix)

$$\langle \mathbf{S}_i \rangle_{T \to 0} = \mathbf{V}_i^0. \tag{17}$$

The saddle points $\mathbf{V}_i^0$ and $\mathbf{U}_i^0$ are given by the equations

$$\mathbf{U}_i^0 = -\frac{1}{T} \nabla_{\mathbf{V}_i} E(\mathbf{V}_1^0 \ldots \mathbf{V}_n^0) \tag{18}$$

$$\mathbf{V}_i^0 = \frac{\mathbf{U}_i^0}{|\mathbf{U}_i^0|} F(|\mathbf{U}_i^0|) \equiv \mathbf{f}(\mathbf{U}_i^0). \tag{19}$$

$F$ is the derivative of $G$ and has sigmoid shape. These equations are the corresponding generalization of the MF eqn (9) for the continuous multidimensional case.

## 3. CONTINUOUS ROTOR BM

We use rotors as neurons in the BM to facilitate continuous valued mapping. That way we get multidimensional units and we have to expand the formulation of the BM to the multidimensional case. We have to define a proper energy function for the relaxation dynamic and to revise the derivation of the BM learning rule.

### 3.1. Model Description

The structure is analogous to the original BM with rotor neurons. We allow independent neuron interaction in each direction,

$$E = -\frac{1}{2}\sum_{ijkl} S_{ik} W_{ikjl} S_{jl} = -\frac{1}{2}\sum_{ij} \mathbf{S}_i \cdot \mathbf{W}_{ij} \cdot \mathbf{S}_j. \quad (20)$$

The indexes $i, j = 1 \ldots n$ enumerate the neurons and $k, l = 1 \ldots d$ the different dimensions in the first notation. The second notation is used to simplify expressions. To guarantee convergence of the dynamics we demand symmetry in $i, j$ and simultaneously in $k, l$, that is, $W_{ikjl} = W_{jlik}$. Equations (18) and (19) now read:

$$\mathbf{U}_i = -\frac{1}{T}\sum_j \mathbf{W}_{ij} \cdot \mathbf{V}_j \quad (21)$$

$$\mathbf{V}_i = \mathbf{f}(\mathbf{U}_i). \quad (22)$$

This leads to the MF equations for the continuous rotor BM. In one dimension they reduce to the original MF eqn (9) of the discrete BM. Once again, these equations can be viewed as the fixed-point of the corresponding partial differential equation of first order, similar to eqn (10). In the Appendix we present these continuous time equations and a Liapunov function that guarantees the convergence to the fixed-point eqns (21) and (22). The convergence of the iterative algorithm that solves this MF equation can be proved for finite temperatures and bounded connection strength ($2\|W\|/Td < 1$). Nonetheless, in our simulations the system converges in a few iteration steps whatever connection strength or temperature we choose, even with $W_{ikjl}$ not symmetric in $k, l$.

### 3.2. Learning Rule

We want to expand the derivation of the BM to the multidimensional continuous case. Basically, we have to substitute the trace over the binary state space in the definitions (2), (3), (5), and (6) by a trace over the continuous space,

$$\Sigma \to \int \prod_i d\mathbf{S}_i \quad (23)$$

whereby the space $\{\mathbf{S}_i^h\}$ of the hidden units, the space $\{\mathbf{S}_i^v\}$ of the visible units, and the conjunct space of all units $\{\mathbf{S}_i^{hv}\}$ have to be properly considered. This leads to the following definitions:

$$P\{\mathbf{S}_i^{hv}\} = e^{-\beta E\{S_i^{hv}\}}/Z \quad (24)$$

$$P\{\mathbf{S}_i^v\} = \int \prod_i d\mathbf{S}_i^h P\{\mathbf{S}_i^{hv}\} \quad (25)$$

$$H = \int \prod_i d\mathbf{S}_i^v R\{\mathbf{S}_i^v\} \log R\{\mathbf{S}_i^v\}/P\{\mathbf{S}_i^v\}. \quad (26)$$

The partition function $Z$ is given by eqn (13). After some calculations we obtain the gradient of the relative entropy $H$ with respect to the connection strengths,

$$\Delta W_{ikjl} = -\eta \frac{\partial H}{\partial W_{ikjl}} = \eta \int \prod_i d\mathbf{S}_i^v R\{\mathbf{S}_i^v\}/P\{\mathbf{S}_i^v\} \frac{\partial}{\partial W_{ikjl}} P\{\mathbf{S}_i^v\} \quad (27)$$

$$\frac{\partial}{\partial W_{ikjl}} P\{\mathbf{S}_i^v\} = \beta \int \prod_i d\mathbf{S}_i^h (S_{ik}^{hv} S_{jl}^{hv} P\{\mathbf{S}_i^{hv}\} - \langle S_{ik} S_{jl}\rangle P\{\mathbf{S}_i^v\}). \quad (28)$$

The brackets denote the thermal average defined here as

$$\langle\ \rangle_{\text{free}} = \int \prod_i d\mathbf{S}_i^{hv} P\{\mathbf{S}_i^{hv}\}. \quad (29)$$

This thermal average is named "free" because the trace is performed over all visible and hidden units states and there is no constraint on any unit. This is in contrast to the "clamped" thermal average where the visible units are kept fixed:

$$\langle\ \rangle_{\text{clamped}} = \int \prod_i d\mathbf{S}_i^{hv} P\{\mathbf{S}_i^{hv}\}/P\{\mathbf{S}_i^v\} = \int \prod_i d\mathbf{S}_i^{hv} P\{\mathbf{S}_i^h | \mathbf{S}_i^v\}. \quad (30)$$

Inserting eqn (28) in eqn (27) and using eqns (29) and (30) we get an analog learning rule

$$\Delta W_{ikjl} = \eta\beta[\overline{\langle S_{ik} S_{jl}\rangle}_{\text{clamped}} - \langle S_{ik} S_{jl}\rangle_{\text{free}}]. \quad (31)$$

The bar denotes the average over the desired distribution $R$ (i.e., the average over the learning patterns). Again, with the approximation analog to eqn (11) and equality (17), we write the MF learning rule,

$$\Delta W_{ikjl} = \eta\beta[\overline{(V_{ik} V_{jl})}_{\text{clamped}} - (V_{ik} V_{jl})_{\text{free}}]. \quad (32)$$

Because we are going to apply the continuous BM to the function approximation task, we will use the obvious modification done by Hopfield (1987) where the visible units are further separated into input and output

units. The input units lead to an external field and need not be restricted to the normalization condition (12). Nevertheless, we will be using, for the sake of simplicity in some cases, normalized inputs.

## 4. SIMULATIONS

The first aim of the simulation was to prove the feasibility of the proposed BM. In some preliminary experiments we confirmed that eqns (21) and (22) converge for every temperature in a few update cycles, even with connections that are not symmetric in $k, l$. At high temperature the rotor values are moving around the origin of their state space. While decreasing the temperature the norm of the rotors increases until some freezing temperature is reached. There $|\mathbf{V}_i| \approx 1$ and the values remain fixed. We observed in the experiments that the freezing temperature is correlated with the connection strengths ($\|W\|/T_{\text{freeze}}$ is of order one). This gives us a guideline for selecting the temperature schedule. Starting just above the freezing point we decrease the temperature slowly. In our experiments we start at temperature 1.0 and decrease it with factor 0.85 until we reach 0.001.

### 4.1. Coding

We have to use at least a two-dimensional rotor to implement a continuous mapping. Given the normalization condition (12) for the output and hidden units, we need one $(d + 1)$-dimensional units to code $d$-dimensional signals at the output. For any $n$-dimensional signal one is free to chose between $n$ two-dimensional unit or a single $(n + 1)$-dimensional rotor. We are free in selecting the sign of the coordinate that is used for the normalization. We will choose in all our experiments a positive sign. This is, of course, arbitrary and makes no difference either to the desired mapping itself, or to the ability of the net to approximate it. We point out that the formulation of the continuous rotor BM can be readily rewritten for a combination of rotors with different dimensionality.

We are going to see in Section 4.3 that three-dimensional units are better suited for a mapping in the three-dimensional space, rather than two two-dimensional units that represent the two angles of polar coordinates in the three-dimensional space.

In 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 came to the same result as in the original work of Peterson (1987). Furthermore, the capacity of learning simple continuous mapping like the one-dimensional sine function has been checked. The net trained with 20 sample points and solved the task nearly perfectly (0.9% error).

### 4.2. Piecewise Continuous Mapping

We want to explain and demonstrate the inherent ability of the system to perform piecewise continuous maps. Discontinuity occurs when small changes in the input values lead to drastic changes of the output towards which the system relaxes. Remember that because there exists a Liapunov function for the relaxation at a fixed temperature, the fixed-points of the MF equations can be understood as the bottoms of the valleys of the energy function. The fixed-point iteration to calculate these solutions is equivalent to making a gradient descent on that energy landscape with the step size 1.0. (see Appendix). The inputs to the net are kept fixed during the relaxation. They can be understood as a constant external field that parameterizes the energy surface. Thus, the energy surface depends continuously on the inputs. Starting always with the same output and hidden configuration, the relaxation always will relax to the same fixed-point. If a small change in the input occurs, the relaxation point is likely to change also in a small amount, unless the energy surface is changed by the different input in such a way that the same starting point of the relaxation occurs to lie in another basin of attraction. In that case, the fixed-point, towards which the system converges, may differ considerably from the one corresponding to slightly different input. Thus, we expect the system to perform piecewise continuous mappings. And, in fact, in simulations we observe a tendency to perform piecewise continuous mappings. The location of the discontinuity can even be trained, as demonstrated in the example in Figure 1. An arbitrary discontinuous one-dimensional mapping was trained [here we used as target function $f(x) = \text{sig}(x)\exp(-|x|)$]. The location of the discontinuity is very sensitive with respect to the connection strengths. This explains the peaks of the learning process in Figure 2. To compensate for these peaks we used the learning constant schedule suggested by Silva and Almeida (1990). All in all, this kind of recurrent net has the ability to perform both continuous and discontinuous mappings. Therefore, we expect good performance, especially in piecewise continuous mapping.

### 4.3. "Cornered Rat" Control Problem

To test ability of performing piecewise continuous mappings, we use training data from a differential game called "cornered rat" proposed by Breakwell (1977). We do not pretend to solve the mathematical problem itself. We just use the numerical results as training data for our net. The game consists in the following: Imagine a rat trying to evade a cat by running to one of two holes in a rectangular field. More generally, the rat tries to maximize its lifetime. In other words, if the rat knows that it won't reach one of the two holes, it runs

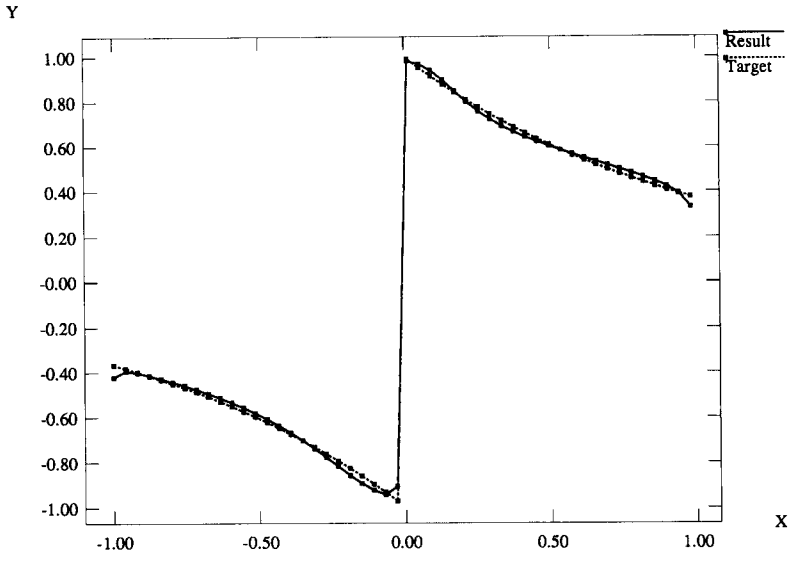


**FIGURE 1. A rotor BM with five two-dimensional units (three hidden, one input, and one output unit) was trained to learn this piecewise continuous map. *X* and *Y* denote one dimension of the input and output unit, respectively. The second dimension is used for the normalization condition (12).**

in the direction where the cat will need the maximum time to reach it. (This will be one of the corners.) Assuming now that both the cat and the rat choose in an optimal way, it is possible to show that both cat and rat will run on straight lines. Depending on its actual position, the rat will choose various directions (Figure 3.). We need a control signal (angle) for the rat according to its position. This has been calculated (in Gabler et al., 1993) by the so called "survival set algorithm" and is found to be a piecewise discontinuous mapping

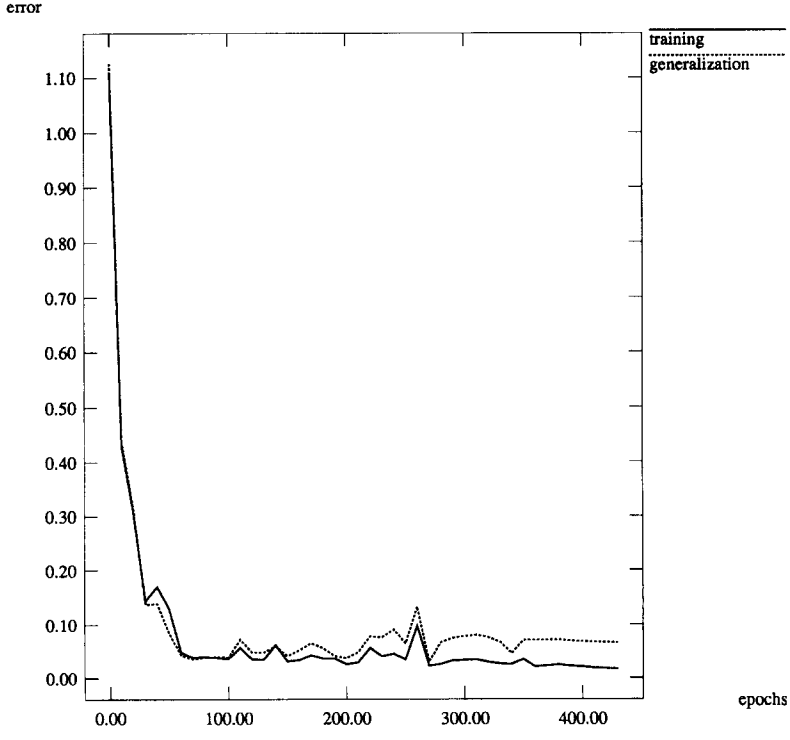


**FIGURE 2. Learning process of the rotor BM for the function of Figure 1. It was trained with 50 sample points and reached an error of 2.0%.**

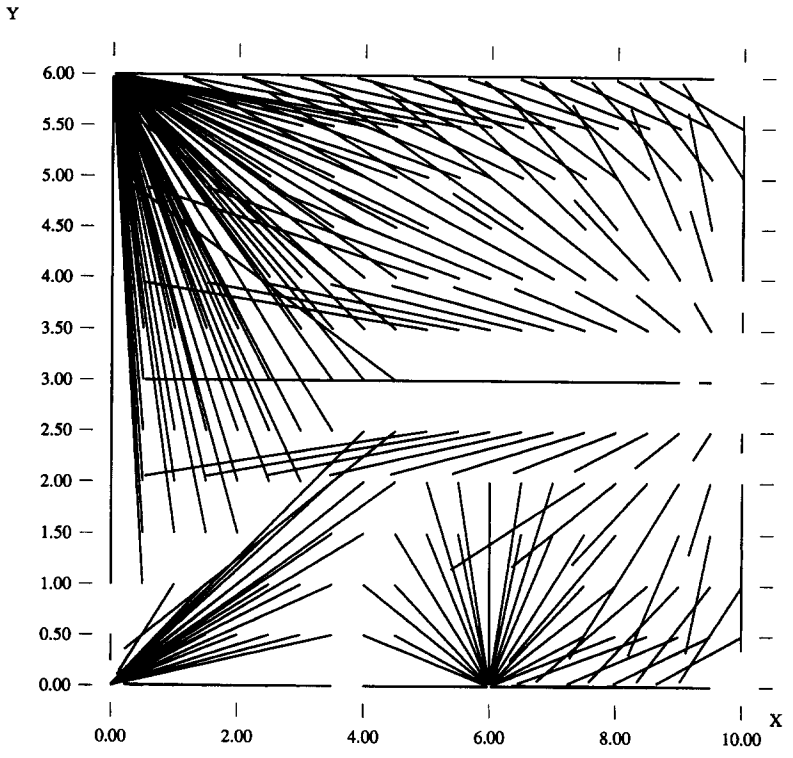


**FIGURE 3. Gabler et al. (1993) calculated traces of the rat. Starting from different locations the rat tries to evade the cat sitting at start position (10, 3). The rat holes are located at (0, 6) and (6, 0).**

from two dimensions like the one shown in Figure 4. Although this optimal control problem can be solved numerically (as done by Gabler et al., 1993), it is computationally intensive. It would be useful to generate the desired mapping first with a few sampling points and to learn this piecewise discontinuous mapping with the proposed BM. We mention that the real task is the calculation of both control signals for the rat and the cat. We limited ourselves to the rat control signal and a single cat position.

We used four hidden two-dimensional rotors, two input rotors for the position coordinates of the rat, and one output for the angle. The range of input coordinates $\{0.0 \ldots 10.0, 0.0 \ldots 6.0\}$ was scaled to $(-1.0 \ldots 1.0, -1.0 \ldots 1.0)$. The second dimension of each input unit was used to fulfill the normalization condition (12). The output unit indicates directly the target angle. As training data we used the $13 \times 21$ sample points showed in Figure 4. After 5000 epochs the error de-

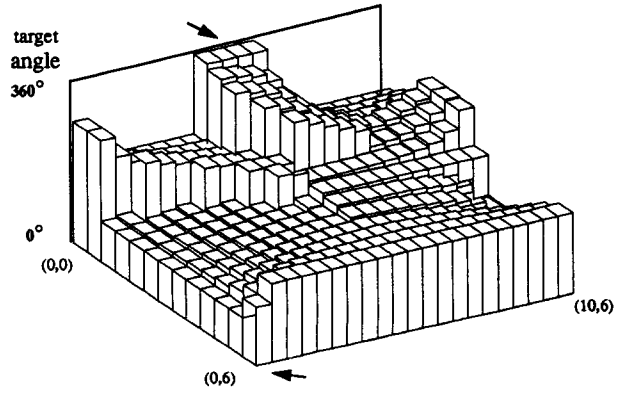


**FIGURE 4. Gabler et al. (1993) calculated optimal control directions of the rat as a mapping from two dimension to one. Height represents the angle of the rat traces. Arrows indicate rat holes.**

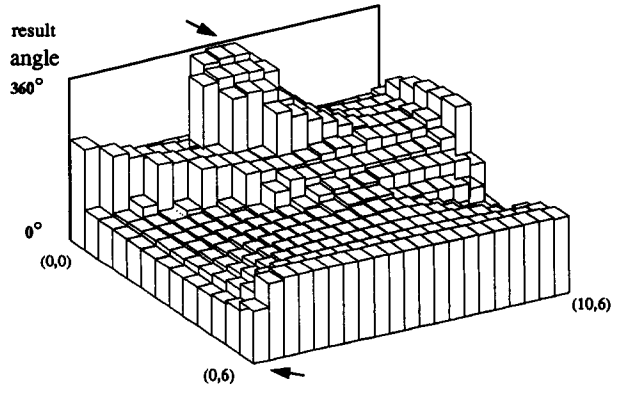


**FIGURE 5. Mapping produced by the continuous BM after 5000 learning epochs.**

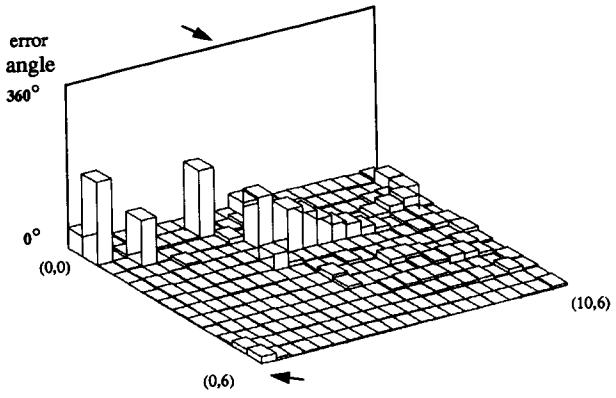


**FIGURE 6. The remaining average angular error continuous BM is 5.48°.**

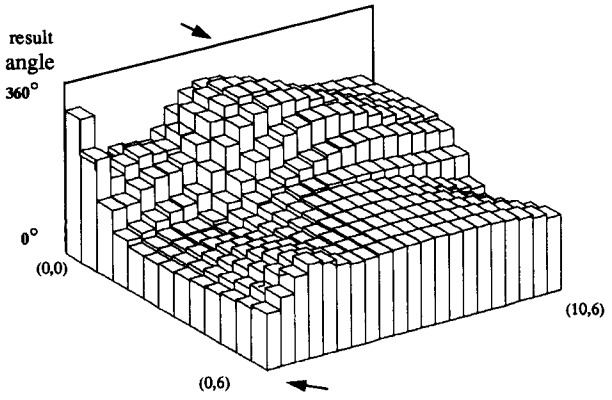


**FIGURE 7. Mapping produced by a MLP with seven hidden neurons after 6000 epochs of learning.**

creased down to 0.1, corresponding to an angle error of 5.48° (see Figure 6). Comparing the target mapping in Figure 4 to the net result after training in Figure 5, the correspondence of the discontinuity is obvious. In comparison, a multilayer perceptron (MLP) with seven hidden neurons and sinusoidal output reached (after 6000 epochs of on-line gradient descent without momentum term) an average error of 16.28°. Based on the same data we also tried a feedforward net with radial basis functions using "partitioning to one" (Moody & Darken, 1989) and a linear output. We got the best result of 9.19° average error with eight hidden neurons by initializing the weights with $K$-means-clustering and $K$-nearest-neighbor algorithm following Moody and Darken (1990) and Duda and Hart (1973). Comparing Figure 5 and Figure 7, the difficulty of the MLP to produce discontinuity can be easily recognized, whereas the BM reproduced the desired edges quite well.

### 4.4. Three-Dimensional Mapping

To verify the advantage of the multidimensional model with respect to the two-dimensional units, we want to show a three-dimensional example. Imagine now the analog control problem of the "cornered rat" in three dimensions. Analogous to the two holes in the rectangle for the rat, there are now for a "dove" two corners to escape in a three-dimensional cube. The corners are located at $(-1, -1, -1)$ and $(+1, +1, +1)$. Unlike the "cornered rat" example, this is an artificial problem where we are only interested in the different performances of three- and two-dimensional rotors. Therefore, we choose a simple separation surface. The dove's target control angle points to the corner, which is less distant from the actual position of the dove.

The nets are trained to learn the two angles of the polar coordinates representation of the target angle. A two-dimensional rotor net uses for this two output units. A three-dimensional rotor net needs only one output unit. The three input dimensions ("dove" position) were coded in the case of the two-dimensional net in three input units and in the case of the three-dimensional net in one three-dimensional input unit. The normalization condition for the inputs was not considered. (Remember that the input units in the original BM do not have to be binary values.) In the two-dimensional net the orthogonal coordinates at the three input units were set to zero. Of course, this leads to a slightly different topology. But experiments with normalized input units showed no substantially different performance. The number of hidden rotor neurons was selected to give the same number of learning parameters (about 80). In the three-dimensional rotor net we used four hidden units and in the two-dimensional we used five hidden units. The training and generalization sets contained 100 points each. The error decreased down to about 8.5° and to 15° for the three- and for the two-dimensional cases, respectively. The generalization er-

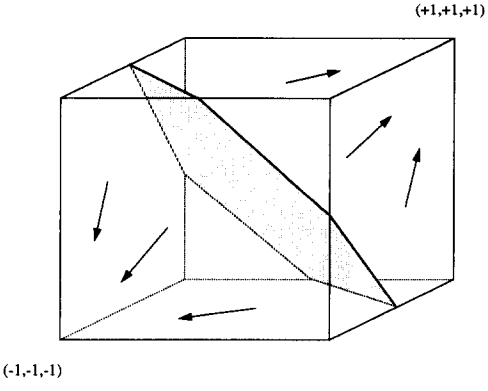


**FIGURE 8. The target directions to be learned point to two opposite corners of a cube. The separation (shadowed) surface is defined by the points with equal distance to the corners.**

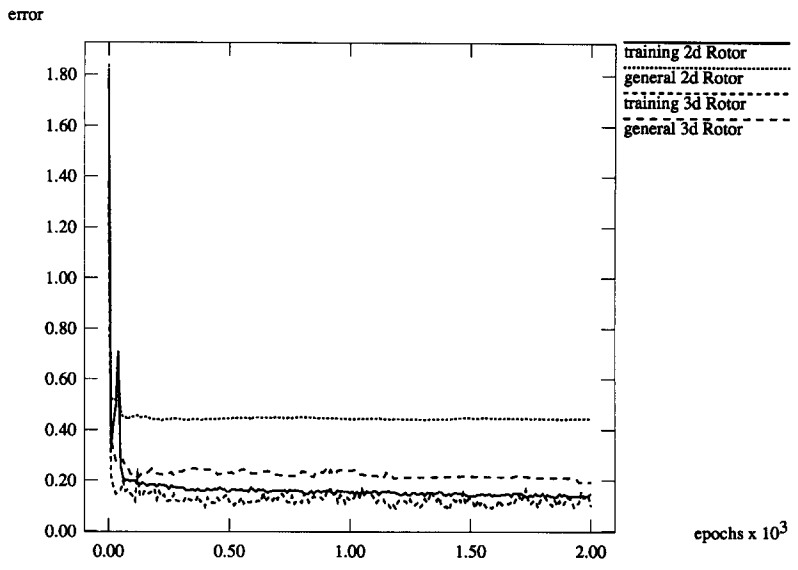


**FIGURE 9. Training and generalization error for a three and a two dimensional rotor net.**

ror differs considerably (14° and 41°) (see Figure 9). This suggests that the three-dimensional rotors capture the desired relation much better because the inherent structure fits this problem. Thus, whenever a direction in an $n$-dimensional ($n > 2$) space is searched, we expect that an $n$-dimensional rotor net will solve the mapping task better than any net with two-dimensional units.

## 5. CONCLUSION

The purpose of this work was the formulation of a continuous version of the classical BM. Therefore, we used the MF theory for rotor neurons. We demonstrated analytically some convergence properties of the resulting rotor dynamic, and derived the appropriate mean MF learning algorithm in analogy to the original BM. This way, we also expanded the models of two-dimensional (or complex-valued) units to arbitrary dimension. We illustrated the convergence of learning in some numerical experiments. We demonstrated the ability to perform piecewise continuous mappings, which represents a difficult task for nonrecursive networks. The multidimensional extension was found advantageous compared to two-dimensional units in the case of mapping into three-dimensional directions.

## REFERENCES

Ackley, D., Hinton, G., & Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science,* **9,** 147–169.

Breakwell, J. V. (1977). Zero-sum differential games with piecewise continuous trajectories. *Lecture Notes of Control and Information Science,* **3.**

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York: Wiley Interscience.

Gabler, T., Miesbach, S., Breitner, H. M., & Pesch, H. J. (1993). Synthesis of optimal strategies for differential games by neural networks. Schwerpunktprogramm der Deutschen Forschungsgemeinschaft: Anwendungsbezogene Optimierung und Steuerung, Report No. 468.

Gislén, L., Peterson, C., & Södeberg, B. (1992). Rotor neurons: Basic formalism and dynamics. *Neural Computation,* **4,** 737–745.

Hopfield, J. (1987). Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks. Proceedings of the National Academy of Sciences, USA **84,** 8429–8433.

Moody, J., & Darken, C. (1989). Fast Learning in Network of locally-tuned Processing Units. *Neural Computation,* **1**(2), 281–294.

Moody, J., & Darken, C. (1990). Fast adaptive K-means clustering: Some empirical result. *International Joint Conference on Neural Networks,* San Diego.

Mozer, M. C., Zemel, R. S., Behrmann, M., & Williams, C. K. I. (1992). Learning to segment images using dynamic feature binding. *Neural Computation,* **4,** 650–665.

Noest, A. (1988). Associative memory in sparse phasor neural networks. *Europhysics Letters,* **6**(6), 469–474.

Peterson, C., & Anderson, J. (1987). A mean field theory algorithm for neural networks. *Complex Systems,* **1,** 995–1019.

Schürmann, B. (1989). Stability and adaption in artificial neural systems. *Physical Review A,* **40**(5), 2681–2688.

Silva, F., & Almeida, L. (1990). Speeding up backpropagation. In *Advanced neural computers* (pp. 151–158). North-Holland: Elsevier Science Publischer B.V.

## APPENDIX

### Saddle-Point and MF Variables

We demonstrate first why the variables $\mathbf{V}_i$ can be understood as the mean of $\mathbf{S}_i$. Using the definition of the thermal average and performing the same procedure that leads to eqn (14), we write

$$\langle \mathbf{S}_i \rangle_{T\to 0} = \lim_{\beta\to\infty} \frac{1}{Z} \int \prod_a d\mathbf{S}_a \mathbf{S}_i e^{-\beta E(\mathbf{S}_1 \ldots \mathbf{S}_n)}$$

$$= \lim_{\beta\to\infty} \frac{\int \prod_a d\mathbf{V}_a \prod_b d\mathbf{U}_b \mathbf{V}_i e^{-\beta E_{\text{eff}}}}{\int \prod_a d\mathbf{V}_a \prod_b d\mathbf{U}_b e^{-\beta E_{\text{eff}}}}. \quad \text{(A.1)}$$

Now the exponent is expanded around the saddle points $\mathbf{U}_i^0$ and $\mathbf{V}_i^0$ of the effective energy. The zero order of the expansion is a constant in the integral and appears both in the nominator and denominator; thus, it cancels. The first order is zero because we expand around the saddle point where $\nabla E_{\text{eff}} = 0$. The second order gives rise to a multidimensional Gaussian where the variance scales with $\sqrt{1/\beta}$. In the limit $\beta \to \infty$, this leads to Dirac delta functions,

$$\langle \mathbf{S}_i \rangle_{T\to 0} = \frac{\int \prod_a d\mathbf{V}_a \delta(\mathbf{V}_a - \mathbf{V}_a^0) \prod_b d\mathbf{U}_b \delta(\mathbf{U}_b - \mathbf{U}_b^0)\mathbf{V}_i}{\int \prod_a d\mathbf{V}_a \delta(\mathbf{V}_a - \mathbf{V}_a^0) \prod_b d\mathbf{U}_b \delta(\mathbf{U}_b - \mathbf{U}_b^0)} = \mathbf{V}_i^0. \tag{A.2}$$

We neglect higher-order terms. Let us point out that in the $n \to \infty$ limit, a similar argument holds. For that it has to be verified that the exponent scales with $n$ for large $n$ as it does in the present case with $\beta$.

## Liapunov Function

Now we show some convergence properties of the proposed MF equation following the same lines as Hopfield. Equations (21) and (22) obviously give the fixed-point of the following partial differential equation:

$$\frac{d\mathbf{U}_i}{dt} = -\mathbf{U}_i + \frac{1}{T}\sum_j \mathbf{W}_{ij} \cdot \mathbf{f}(\mathbf{U}_j). \tag{A.3}$$

To demonstrate that the dynamic converges to this fixed-point, we have to show that there exists some Liapunov function

$$L = -\frac{1}{2T}\sum_{ij} \mathbf{V}_i \cdot \mathbf{W}_{ij} \cdot \mathbf{V}_j + \sum_i \int_0^{\mathbf{V}_i} \mathbf{f}^{-1}(\mathbf{V}) \cdot d\mathbf{V} \tag{A.4}$$

where $\mathbf{f}^{-1} = \mathbf{V}/|\mathbf{V}| F^{-1}(|\mathbf{V}|)$ exists, because $F' > 0$ everywhere. The path integrals can be performed over an arbitrary curve because the integrand has zero rotation. To verify that $L$ is a Liapunov function, we have to show that its time derivative is negative:

$$\frac{dL}{dt} = -\sum_i \frac{d\mathbf{V}_i}{dt} \cdot \left( \frac{1}{T}\sum_j \mathbf{W}_{ij} \cdot \mathbf{V}_j - \mathbf{f}^{-1}(\mathbf{V}_i) \right) = -\sum_i \frac{d\mathbf{V}_i}{dt} \cdot \frac{d\mathbf{U}_i}{dt} = -\sum_{ij} \frac{d\mathbf{V}_i}{dt} \cdot \nabla_{\mathbf{v}_j} \mathbf{f}_i^{-1} \cdot \frac{d\mathbf{V}_j}{dt} < 0. \tag{A.5}$$

The inequality is valid if $\nabla_{\mathbf{v}_j} \mathbf{f}^{-1}(\mathbf{V}_i)$ is positive definite. Note that in eqn (A.5) the symmetry of the connection strengths was needed. With the weaker condition of detailed balance the Liapunov characteristic was proved for the original Hopfield model by Schürmann (1989).

To prove that the forth-rank tensor is positive definite we use the theorem of Gershgorin. To avoid misunderstanding, we use the complete index notation. We abbreviate $V_i = |\mathbf{V}_i|$ and $G = F^{-1}(|\mathbf{V}_i|)$:

$$h_{jl}^{ik} \equiv \frac{\partial f_{ik}^{-1}}{\partial V_{jl}} = \delta_{jl}^{ik} \frac{G}{V_i} - \frac{V_{ik} V_{il} \delta_j^l}{V_i^2} \left( -\frac{G}{V_i} + G' \right). \tag{A.6}$$

First we show that the diagonal elements $h_{ik}^{ik}$ are positive:

$$h_{ik}^{ik} = \frac{G}{V_i} + \frac{V_{ik}^2}{V_i^2} \left( -\frac{G}{V_i} + G' \right) = \frac{G}{V_i} \left( 1 - \frac{V_{ik}^2}{V_i^2} \right) + \frac{V_{ik}^2}{V_i^2} G' \geq 0. \tag{A.7}$$

This holds because $G$ and $G'$ are positive for positive arguments. Furthermore, we have to show that

$$h_{ik}^{ik} > \sum_{jl \neq ik} |h_{jl}^{ik}|. \tag{A.8}$$

The right side can be rewritten as

$$\sum_{jl \neq ik} \left| \frac{V_{ik} V_{il} \delta_j^l}{V_i^2} \left( -\frac{G}{V_i} + G' \right) \right| = \sum_{l \neq k} \left| \frac{V_{ik} V_{il}}{V_i^2} \left( -\frac{G}{V_i} + G' \right) \right| = \frac{|V_{il}|}{V_i^2} \left| -\frac{G}{V_i} + G' \right| \sum_{l \neq k} |V_{il}|. \tag{A.9}$$

Considering eqns (A.7) and (A.9), we have to show

$$\frac{G}{V_i} - \frac{V_{ik}^2}{V_i^2} \left( -\frac{G}{V_i} + G' \right) - \frac{|V_{ik}|}{V_i^2} \left| -\frac{G}{V_i} + G' \right| \sum_{l \neq k} |V_{il}| > 0, \tag{A.10}$$

and

$$|V_{ik}| \left( -\frac{G}{V_i} + G' \right) - \left| -\frac{G}{V_i} + G' \right| \sum_{l \neq k} |V_{il}| < G \frac{V_i}{|V_{ik}|}. \tag{A.11}$$

As

$$-\frac{G}{V_i} + G' < 0 \tag{A.12}$$

holds, eqn (A.11) is equivalent to

$$\left( \frac{G}{V_i} - G' \right) \sum_l |V_{il}| < G \frac{V_i}{|V_{ik}|}. \tag{A.13}$$

Because for any vector $|V_k| \leq \Sigma_l |V_l| \leq |\mathbf{V}|$ holds, it is enough to verify

$$\left( \frac{G}{V_i} - G' \right) V_i < G. \tag{A.14}$$

This last holds by definition because

$$0 < G' V_i. \tag{A.15}$$

## Convergence of the Dynamic

We show now the local convergence of the dynamic defined by the MF eqns (21) and (22):

$$\mathbf{V}_i(t+1) = \mathbf{f}\left( -\frac{1}{T}\sum_j \mathbf{W}_{ij} \cdot \mathbf{V}_j(t) \right). \tag{A.16}$$

We remark that this iterative update algorithm for solving eqns (21) and (22) can also be viewed as discrete time integration of eqn (35) with time scale $\Delta t = 1$. Under conditions of finite temperature and properly bounded connection weights, the local convergence of the algorithm can be shown explicitly. According to the Banach fixed-point theorem, local convergence is guaranteed if

$$\left\|\frac{\partial f_{ik}}{\partial V_{jl}}\right\| < 1 \tag{A.17}$$

$$\left\|\frac{\partial f_{ik}}{\partial V_{jl}}\right\| = \left\|\sum_{nm}\frac{\partial f_{ik}\partial U_{nm}}{\partial U_{nm}\partial V_{jl}}\right\| = \left\|\nabla_{\mathbf{V}}\mathbf{f}\cdot\frac{\mathbf{W}}{\mathbf{T}}\right\|. \tag{A.18}$$

The forth-rank tensor $\nabla_{\mathbf{V}}\mathbf{f} \equiv g$ can be written as eqn (A.6) substituting $\mathbf{V}$ by $\mathbf{U}$ and $G$ by $F$. Because the corresponding conditions (A.12) and (A.15) still hold, the inequalities (A.7) and (A.8) are also valid for this tensor. It is thus positive definite, with positive eigenvalues

$$\lambda_{ik} \in \left[\left(g^{ik}_{ik} + \sum_{jl\neq ik}|g^{ik}_{jl}|\right), \left(g^{ik}_{ik} - \sum_{jl\neq ik}|g^{ik}_{jl}|\right)\right]. \tag{A.19}$$

Now we can bound the norm as

$$\|\nabla_{\mathbf{V}}\mathbf{f}\| = \max_{ik}|\lambda_{ik}| \leq \max_{ik}\left(g^{ik}_{ik} + \sum_{jl\neq ik}|g^{ik}_{jl}|\right)$$

$$< \max_{ik}(2g^{ik}_{ik}) \tag{A.20}$$

$$= 2\max_{ik}\left(\frac{F}{U_i} + \frac{U_{ik}^2}{U_i^2}\left(-\frac{F}{U_i} + F'\right)\right)$$

$$\leq 2\max_{i}\left(\frac{F}{U_i} + \left(-\frac{F}{U_i} + F'\right)\right) = \frac{2}{d} \tag{A.21}$$

where $1/d$ is the maximal slope of $F$ at the zero point. At the end we get from eqn (A.18) condition for the local convergence:

$$\left\|\frac{\partial f_{ik}}{\partial V_{jl}}\right\| = \left\|\nabla_{\mathbf{V}}\mathbf{f}\cdot\frac{\mathbf{W}}{\mathbf{T}}\right\| \leq \|\nabla_{\mathbf{V}}\mathbf{f}\|\left\|\frac{\mathbf{W}}{\mathbf{T}}\right\| < \frac{2}{dT}\|\mathbf{W}\| < 1. \tag{A.22}$$