Evolutionary Learning Strategies for Cellular Neural Networks

R. Kunz and R. Tetzlaff

Institut für Angewandte Physik, Universität Frankfurt
Robert Mayer-Straße 2–4, 60054 Frankfurt a. M., Germany
phone: +49 69 798 28317, fax: +49 69 798 28865
R.Kunz@rz.uni-frankfurt.de

ABSTRACT: *In this paper a new learning algorithm for Cellular Neural Networks is presented based on evolutionary strategies. The proposed global optimization procedure is discussed in detail and the performance on various parameter determination problems will be shown afterwards.*

1. Introduction

The universal Cellular Neural Networks (CNN) paradigm [1] has been studied in various investigations, which are leading to many different applications, e.g. image processing, by solving nonlinear partial differential equations or by modelling complex natural phenomena. Generally a CNN is an arrangement of coupled cells, where all cells interact only within a local, usually small neighborhood with its neighbors. A CNN can be described by a system of coupled nonlinear differential equations

$$\dot{x}_i = F(x_i, z_i, \vec{u}_j, \vec{y}_j) \tag{1}$$

where the elements of the vectors $\vec{u}_j, \vec{y}_j$ are the inputs and cell outputs $y_i = f(x_i)$ of all neighbor cells $c_j \in N_i(r)$. The set of CNN parameter values is sometimes called a CNN-gene [3]. For some applications this gene can be determined analytically for other problems a parameter training has to be performed. In recent publications [4, 5, 6] different training algorithm were introduced, some of them are limited to a special CNN type, other are endangered by local minima. In this paper we will introduce an alternative method for the determination of the CNN-gene based on an evolutionary process (ELS). [7]. In the following sections, we will first define an evolutionary process in the case of CNN, then we introduce a new learning procedure. Finally the performance of the learning procedure is evaluated for 3 different well studied problems and for a new interesting problem.

2. The evolutionary strategy

A CNN-gene consists of all CNN parameters with a certain task for a given network. In the following definitions evolutionary syntax for CNN is introduced.

Definition 1:
A *feature* is a value or a function describing one concrete interaction between CNN cells or actions on CNN cells at a given position in the network, i.e. the feedback connection of a cell itself.

Definition 2:
A CNN operation consisting of a certain number of features is called an *individual* $\mathcal{I} = \mathcal{I}(a_1, a_2, b_1, b_2, ..., i)$, for example the edge detection operation is defined by $\mathcal{I}^{edge}(2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3)$.

Straightforward, a population P is introduced here.

Definition 3:
A number N of individuals $\mathcal{I}$ sharing the same structure of features are forming a *population* $P = P(\mathcal{I}_1, \mathcal{I}_2, ...\mathcal{I}_N)$. For example $\mathcal{I}_1(1, -1, 0)$ and $\mathcal{I}_2(2, -3, 0)$ are members of the same population, neither $\mathcal{I}_3(2, -3, 0, 1)$ nor $\mathcal{I}_4(-1, -1)$ belong to it.

It is important to clarify that in one population only individuals having the *same* number of features describing the *same* connections are included. Of course all those individuals can and generally do have different behavior in their result. For example the **and** template and the **or** template are members of the same population but produce different results.

Definition 4:
A *generation* $G(P, k)$ consists of a population P at a given epoch of the evolution k; here a evolution is equal to the iteration step of the optimization procedure. For example $G(P, 0)$ describes the initial generation and $G(P, 1)$ its child generation. Every new generation is created from its parent generation with major or minor modifications.

Definition 5:
A desired behavior that an individual should fulfill is called the *environment*. This is usually the optimization target. An individual is stated fit if it well suits its environment, measured by a so called fitness function.

3. Outline of the algorithm

To start a learning procedure, firstly a generation $G(P, 0)$ will be initiated, where all N individuals will be initialized by random numbers. Each individual has the same number M of features. To create such a population, we use different methods: normal distributed or uniform distributed random numbers $\vec{\mu}$ with zero mean and a given but free to choose initial variation or a variation of an equidistant fit with a fixed distribution range where the distance of an individual to their nearest neighbours is equal for each individual.

$$G(P, 0) \supseteq \mathcal{I}_i = \vec{\mu}, \quad 0 \leq i \leq N \tag{2}$$

The function $F(\mathcal{I})$ is used to define the fitness of each individual in the population to the given environment. Here the mean square error (MSE)

$$F(\mathcal{I}) = \frac{1}{Z} \sum_{i=1}^{Z} (\hat{y}_i^k(t) - y_i^k(t))^2 \tag{3}$$

or the absolute mean error (AMSE)

$$F(\mathcal{I}) = \frac{1}{Z} \sum_{i=1}^{Z} (|\hat{y}_i^k(t) - y_i^k(t)|) \tag{4}$$

is considered. y_i^k denotes the output of a cell of the Z cells in the environment. Now the selection starts by creating the next generation based on the actual one. $K < M$ individuals of the population having the best fitness are the parents of the following generation $G(P, k+1)$. Here the children were obtained by a superposition of the parents features with small random numbers $\vec{\mu}$, either normal or uniform distributed with zero mean.

$$\mathcal{I}_i(G(P, k+1)) = \mathcal{I}_j(G(P, k)) + \vec{\mu}, \qquad j \in [1, K], i \in [1, M] \tag{5}$$

Additionally two other aspects were taken into account to create the next generation. One describes the case of mutating or crossing some features of two childs as shown in Fig. 1. The crossing point and the effect of crossing is determined randomly. Furthermore the case of an immigration

$$G(k+1, P) = G(k, P) + \mathcal{I}_k, \qquad \mathcal{I}_k \notin G(k, P) \tag{6}$$

is also considered, where the new individuals $\mathcal{I}_k$ are initialized the same as the first generation. Both aspects are necessary to avoid a fast convergence of a population to a local minimum. The new generation then consists of the parents and children, of immigrants and mutated children. The algorithm continues until either a predefined number of iterations is reached or a desired minimum is found.

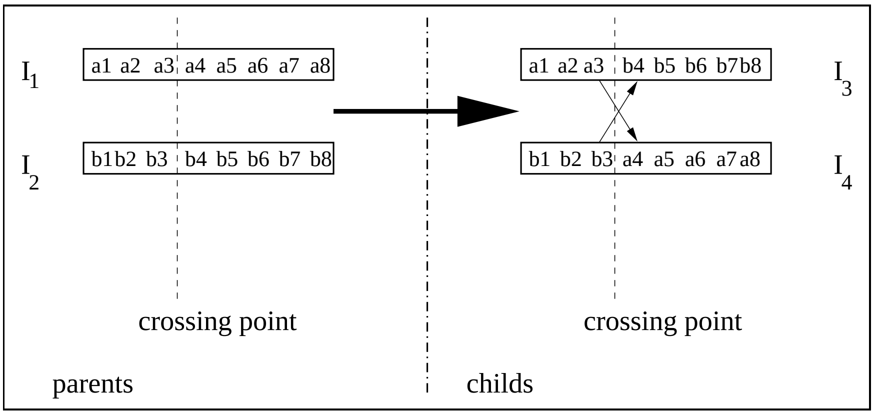

Figure 1: The crossing of features of two parents.

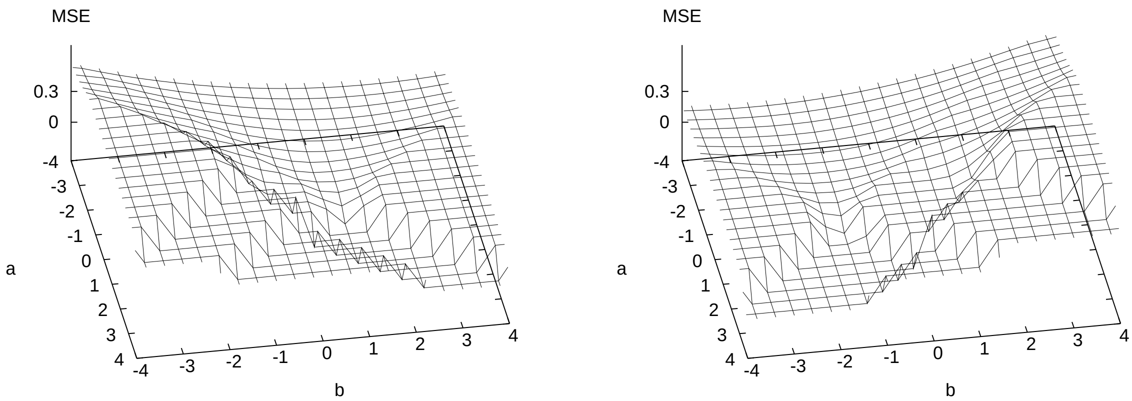

Figure 2: Error surface for the **and** *template at two different bias values of (a) $I = -1$ and (b) $I = -1.2$. a and b denote the two free parameter of this template.*

4. Results

The ELS has been implemented in the universal simulation system SCNN 2000 [8] and its performance has been measured for various cases. In this paper image processing, the minimization of parameter deviations considering tolerances in hardware implementations and a modelling problem will be discussed.

4.1. Image processing

For the training of an image processing template, the well known **and** template [14] was considered. Its individual consists only of three features, which are $\mathcal{I}^{and} = (a = 1, b = 1, I = 1.5)$. The error surfaces for different bias values are shown in Fig. 2.

In this case, the ELS finds the correct location within $k = 56$ generations, with a gradient based algorithm, 600 iterations were necessary to obtain this result. The performance of the parameter training is shown in Fig. 3.

As a further example a training of the **average** template has been performed, leading again to a fast and accurate determination of the parameter values, the MSE is shown in Fig. 4a.

4.2. Minimizing parameter deviations

We have shown [10] that minimizing the effects of parameter variations improves the accuracy of hardware implementations significantly. Here the ELS were used to minimize such effects of deviated

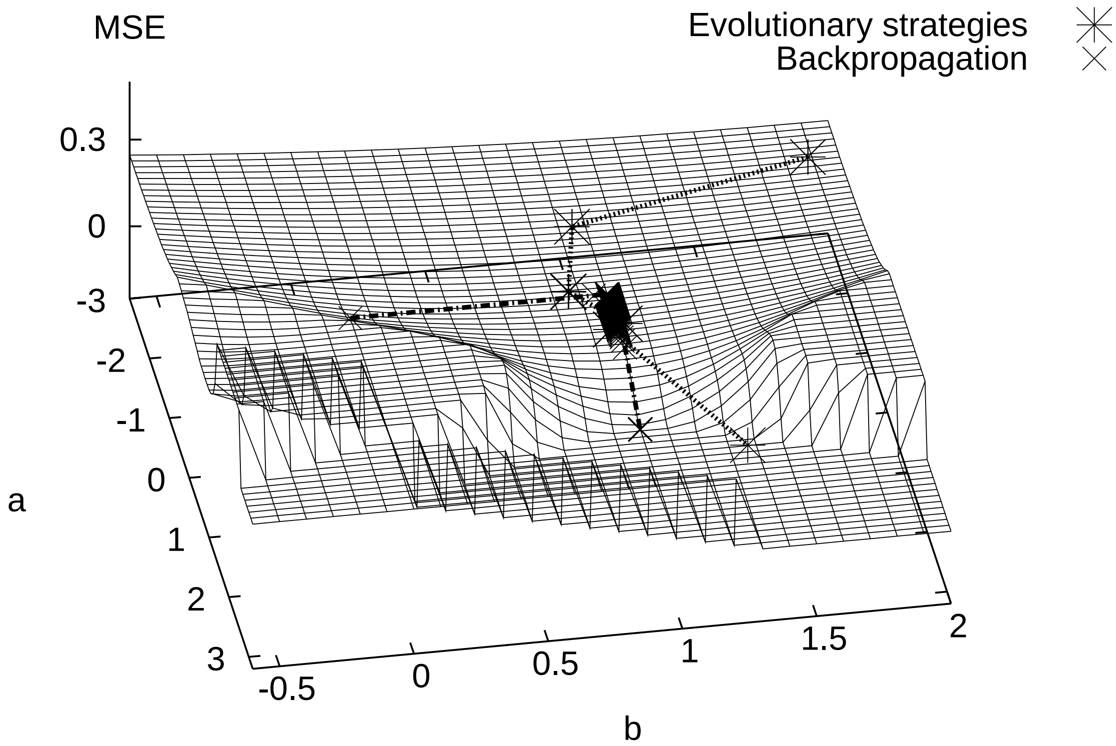

Figure 3: Error surface for the **and** *template and development of the learning process for the evolutionary process and for a standard back-propagation algorithm*

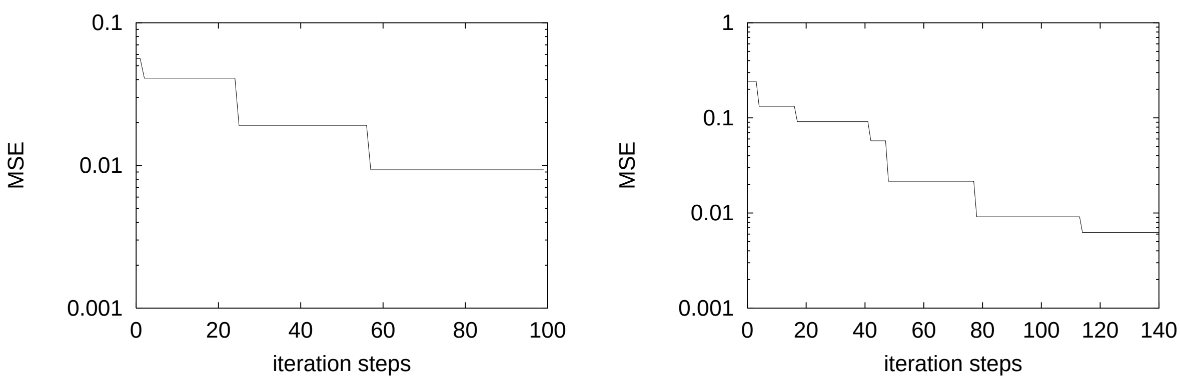

Figure 4: MSE vs. no. of trainingssteps for training of the **average** *template and (b): minimizing the effect of parameter deviations on the* **hlf33** *template.*

Population	Parents	Iteration steps	Cross over	MSE	Polynomial
500	50	65130	0.0	0.000578704	5
500	50	54174	0.05	0.00696738	2
100	50	63917	0.6	0.137478	2
100	50	66696	0.2	0.144889	2
100	50	83443	0.5	0.148365	2
100	50	61138	0.4	0.147439	2

Table 1: Results by applying different ELS leading to approximations $D_{CNN}(k)$.

templates described in [6]. Firstly the parameter of a CNN were modified with uniform distributed random numbers, leading to a translation variant CNN, showing the behavior of hardware realizations. Then the template elements will be re-adjusted with an optimization procedure in order to minimize the difference of the cell outputs to those of a simulated error free network. In previous investigations always gradient based methods have been used. The application of ELS leads in all treated cases to results with the same accuracy. One typical result for a network with 80×80 cells is shown in Fig. 4b. Thereby, with uniform random values, each element of the **hlf33** template was modified with a standard deviation $\mu = 10\%$.

4.3. Analysis of brain electrical activity in epilepsy

Different investigations [11] showed that the spatio-temporal behaviour of brain electrical activity in epilepsy can be characterized by estimates $D_2^*(k)$ of an effective correlation dimension in many cases. Especially, we have shown [12, 13] that the dimension $D_2^*(k)$ can be approximated by $D_{CNN}(k)$ with a high accuracy. $D_{CNN}(k)$ is a function of CNN cell output values. In this paper we have studied the performance of ELS by a determination of CNN leading to aproximations $D_{CNN}(k)$ of $D_2^*(k)$.

$$D_{CNN}(k,m) = H(\{y_j(t_\tau)|j \in [1, \tilde{M}]\}), \qquad \tilde{M} \leq M \tag{7}$$

In this paper an approximation $D_{CNN}(k,m)$ of $D_2^*(k,m)$ with $0 \leq D_2^*(k,m) \leq 10$ is obtained by a determination of the normalized average

$$H = \frac{5}{M} \sum_{j=1}^{M} y_j(t_\tau) + 1, \qquad 0 \leq H \leq 10 \tag{8}$$

of the cell outputs. Thus, by calculating $D_{CNN}(k,m)$ for all signal segments of a recording, a so called CNN dimension profile follows, which is compared to the result obtained with $D_2^*(k)$. Results for a data set of one patient are given in Table 1 and Fig. 5. Hereby the maximum variation μ of the randomly distributed individuals is set to 10, always the piecewise linear output function and Neumann boundary conditions were used. In all treated cases, a feedback template with a neighborhood radius $N = 1$ and a space invariant bias were determined. As Table 1 clearly shows, an increasing number of individuals within a population always yield to better results, which is caused by an improved parameter distribution in the parameter space. Up to 46 parameters were determined throughout these optimizations. Finally, the

5. Conclusion

Throughout various calculations, ELS has to be proven to be powerful learning algorithms. It enables a global search and is especially suited to problems, where local minima of the MSE occur at the cost of speed and/or memory requirements.

6. References

[1] L.O. Chua and L. Yang: " Cellular Neural Networks: Theory and Applications"; IEEE Transactions on Circuits and Systems vol. 35, pp. 1257–1290, 1988.

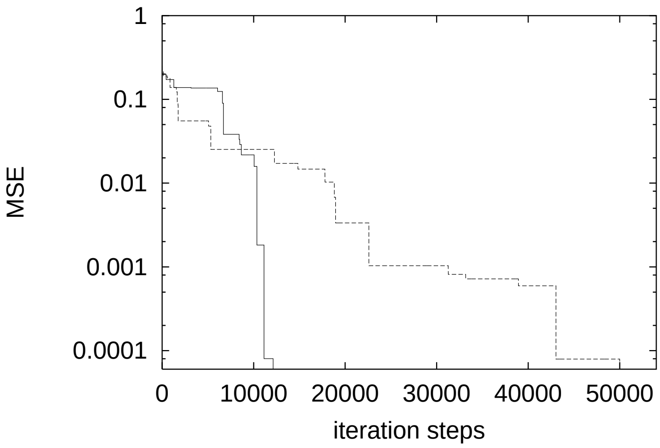

Figure 5: MSE vs. no. of trainingsteps. Determination of a CNN for the analysis of brain electrical activity.

[2] V.Cimagalli and M.Balsi: " Cellular Neural Networks: A Review"; Proc. 6th Italian Workshop on Parallel Architectures and Neural Networks, Vietri sul Mare, Italy,1993.

[3] L. O. Chua: CNN:" A paradigm for complexity"; World Scientific Series on Nonlinear Science, Series A, Vol. 31, 1998.

[4] J.A. Nossek: " Design and Learning with Cellular Neural Networks"; Proc. IEEE CNNA 94, Rome, pp. 137–146,1994.

[5] M. H"anggi and G.S. Moschytz: Attacking the General Classification Problem with CNNs, Proc. ECCTD99, pp. 771-774, Stresa, Italy, 1999.

[6] R.Tetzlaff, R.Kunz and D.Wolf: " Minimizing parameter deviations on Cellular Neural Networks" Proc. ECCTD 97, pp357ff, Budapest, Hungary, 1997.

[7] Charles Darwin " On the Origins of Species by Means of Natural Selection",1856

[8] R. Kunz, R. Tetzlaff and D. Wolf: " SCNN: A Universal Simulator for Cellular Neural Networks"; Proc. IEEE CNNA 96, Sevilla, pp. 255–260, 1996.

[9] F. Puffer, R. Tetzlaff and D. Wolf: " A Learning Algorithm for Solving Nonlinear Partial Differential Equations with Cellular Neural Networks (CNN)"; Proc. URSI ISSSE, San Francisco, pp 501–504, 1995.

[10] R.Tetzlaff, R.Kunz, G.Geis and D.Wolf: " Minimizing the Effects of Tolerance Faults on Hardware Realizations of Cellular Neural Networks"; Proc. CNNA 98, London, pp 254–258, 1998.

[11] K. Lehnertz and C. E. Elger: "Can epileptic seizures be predicted?" Evidence from nonlinear time series analyses of brain electrical activity, Phys. Rev. Lett, Vol. 80, pp. 5019-5022, 1998.

[12] R.Tetzlaff, R.Kunz and D.Wolf: " Analysis of Brain Electrical Activity in Epilepsy with Cellular Neural Networks (CNN)"; Proc. ECCTD 99, Stresa, Italy, 1999.

[13] R. Kunz, R. Tetzlaff and D.Wolf: " Brain electrical activity in epilepsy: Characterization of the spatio-temporal dynamics with Cellular Neural Networks based on a correlation dimension analysis"; Accepted for publication at the ISCAS 2000, Geneve.

[14] T.Roska and L.Kék: " CSL: CNN Software Library"; Version 7, Budapest, 1997.