THE ADATRON LEARNING ALGORITHM WITH CONSTRAINTS

by

P. Nachbar, T. Füssl, J. A. Nossek, and L. O. Chua

Memorandum No. UCB/ERL M92/71

10 July 1992

THE ADATRON LEARNING ALGORITHM WITH CONSTRAINTS

by

P. Nachbar, T. Füssl, J. A. Nossek, and L. O. Chua

Memorandum No. UCB/ERL M92/71

10 July 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

This work is supported in part by the Office of Naval Research under grant N00014-89-J-1402 and the National Science Foundation under grant MIP-8912639.

THE ADATRON LEARNING ALGORITHM WITH CONSTRAINTS

by

P. Nachbar, T. Füssl, J. A. Nossek, and L. O. Chua

Memorandum No. UCB/ERL M92/71

10 July 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

This work is supported in part by the Office of Naval Research under grant N00014-89-J-1402 and the National Science Foundation under grant MIP-8912639.

The AdaTron learning algorithm with constraints

P. Nachbar*, T. Füssl, J. A. Nossek
Institute for Network Theory and Circuit Design
Technical University of Munich, FRG

L. O. Chua
Electronics Research Laboratory, College of Engineering
University of California, Berkeley

July 10, 1992

Abstract

The AdaTron algorithm is able to find the so called perceptron of optimal stability. Although if one imposes constraints on the weights or considers gray scale patterns it is sensible to distinguish different kinds of robustness (stability). We show that in any case the AdaTron algorithm is able to find a perceptron (or attractor network) of optimal robustness, just taking actual constraints into account and apply these results to Cellular Neural Networks (CNN).

1 Introduction

Over the years various algorithm have been developed to solve the so called perceptron, problem e.g. Perceptron algorithm [20, 17], Adaline [24, 25], MinOver [13] and AdaTron [2]. For an excellent review on Perceptron like algorithms we refer the reader to Biehl, Anlauf and Kinzel [4]. Their importance is not only due to the perceptron problem itself, but they are also applicable to the learning problem of attractor networks, provided the transient of the neural network is specified explicitly.

The AdaTron (*Ada*ptative Percep*tron*) algorithm is a very interesting one since it combines fast convergence and yields the so called perceptron of optimal stability [9]. In this paper our main concern is to extend the AdaTron algorithm to the case, when not all the parameters of the weight vector are actually independent. This kind of situation arises rather frequently for attractor networks, if one imposes some symmetry constraints on the weights. This is for example the case for the Hopfield model [12], Little model [14], the Cellular Neural Network (CNN) [6] and the Discrete-Time Cellular Neural Network (DTCNN) [11].

Moreover we are going to relate the notion of stability to the one of robustness [8, 22] in the weight and parameter space which is most important for actual circuit design. It turns out that all three notions coincide, if one considers binary patterns and does not impose symmetry constraints on the weights. In the last section we apply this theory to the CNN learning

*Supported by a grant from the Ernst von Siemens foundation.

problem with rotationally invariant r-neighbourhood templates on a square grid [18] and give some preliminary simulation results.

2 The AdaTron algorithm with symmetry constraints

The perceptron learning problem can be viewed as the problem of solving a set of affine inequalities [20, 17]. It is always possible to reduce this to a linear standard form: given P vectors $\xi^\nu \in R^n, \nu = 1, \cdots, P$ we are looking for a weight vector $W \in R^n$ such that $W^t\xi^\nu > 0$ for $1 \leq \nu \leq P$, where the superscript t denotes the transpose operator. The set of solutions, if not empty, is an unbounded open convex set. More precisely: if W_1 and W_2 are solutions so is λW_1 and $\mu W_1 + (1-\mu)W_2$ for each $\lambda > 0$ and $0 \leq \mu \leq 1$. Following Anlauf and Biehl [2] let us define the field strength of the pattern ξ^ν by $E^\nu := W^t\xi^\nu$. We can now define the stability of a solution W of the perceptron problem [9].
Definition:

$$\Delta(W) := \frac{1}{||W||}\min_\nu E^\nu := \frac{1}{||W||}\min_\nu W^t\xi^\nu \tag{1}$$

is called the stability of the weight vector W, if it solves the perceptron problem given by the patterns $\xi^\nu, \nu = 1, \cdots, P$.

This definition is reasonable, since it is just the minimal euclidean distance from the pattern to the plane determined by the weight vector W.
Definition: If the perceptron problem is solvable, the solution W^* of the following optimization problem is called the perceptron of optimal stability. [9]

$$\Delta := \max_W \Delta(W) \text{ subject to } W^2 = 1\,. \tag{2}$$

It can be shown [4, 2] that this problem is equivalent to

$$\min W^2 \text{ subject to } E^\nu \geq 1 \text{ for } 1 \leq \nu \leq P\,.$$

Hence the problem is reduced to a quadratic convex programming problem. Since, if the problem is solvable, the solution is unique, we can conclude that the perceptron of optimal stability is uniquely defined.

For our following application of the AdaTron algorithm to the CNNs and the DTCNNs an extension of this theorem is necessary, since one usually has to impose some symmetry constraints on the template coefficients [26, 15, 11].
Definition: A weight vector $W \in R^n$ satisfies the linear symmetry constraint $S: R^m \longrightarrow R^n$ iff there is a vector $V \in R^m$ such that $W = SV$.

The perceptron problem with a (linear) symmetry constraint is now defined in an obvious way.

Theorem 1 *Let $S: R^m \longrightarrow R^n$ be a linear map such that S^tS is non-singular. If the perceptron problem with a symmetry constraint S is solvable, the following is equivalent:*

- *Finding the perceptron W^* of optimal stability with symmetry S i.e. the solution of*

$$\max_W \left(\min_\nu E^\nu\right) \quad \textit{subject to} \quad W^2 = 1 \textit{ and }$$
$$\exists V \in R^m : W = SV$$

- *Solving the convex quadratic programming problem given by*

$$\min V^2 \text{ subject to } V^t\zeta^\nu \geq 1 \quad \nu = 1, \cdots, P$$

where $\zeta^\nu := D^{-0.5}U^tS^t\xi^\nu$ *and* U *is an orthogonal matrix such that* $U^t(S^tS)U = D$*; where* D *is a diagonal matrix.*

If V^* *is a solution of the second problem the solution of the initial problem* W^* *is given by* $W^* = \lambda \cdot SUD^{-0.5}V^*$ *for an appropriate* $\lambda > 0$ *to ensure the normalization.*

Observe that since there always exist a particular realization of the linear symmetry S', namely SU, such that S'^tS' is diagonal, it is usually useful to look for such a parametrization right away. Although for some applications of this theorem to circuit design (see corollary 2) this isn't appropriate. For the examples worked out in section 4 we choose the linear map S such that S^tS is diagonal right away.

Proof: Notice that by assumption (S^tS) is symmetrical and positive defined, so all the above notions are well definite. Let us define $F(W,\xi) := \min_\nu W^t\xi^\nu$ and analogously $F(V,\zeta) := \min_\nu V^t\zeta^\nu$. For brevity sake let us further define $A := SUD^{-0.5}$ and since $UD^{-0.5}$ is non-singular we have to prove that the following is equivalent:

I $\min F(W,\xi)$ subject to $W^2 = 1$ and $\exists V : W = AV$.

II $\min V^2$ subject to $F(V,\zeta) \geq 1$.

Using this notation the ζ^ν are given by $A^t\xi^\nu$ and one easily deduces that $A^tA = \mathbf{1}_m$ where $\mathbf{1}_m$ is the $m \times m$ identity matrix. From this we get two properties which we will use frequently without any further notice: $||AV|| = ||V||$ and $F(V,\zeta) = F(AV,\xi)$. The same applies for the homogeneity of F with respect to the first factor, i.e. if $F(W,\xi) > 0$ then $F(\lambda W,\xi) = \lambda F(W,\xi)$ for each $\lambda > 0$. The analogue is true for $F(V,\zeta)$.

Let us first prove that (I) implies (II). Let $W^* = A\tilde{V}$ be a solution of (I). Since S^tS is non-singular, the vector $\tilde{V}$ is uniquely defined and we claim that

$$V^* := \frac{\tilde{V}}{F(\tilde{V},\zeta)}$$

is a solution of (II). By assumption $0 < F(A\tilde{V},\xi) = F(\tilde{V},\zeta)$ and using the homogeneity we see that $F(V^*,\zeta) = 1$. Hence V^* satisfies the constraints. Now let V be arbitrary such that it satisfies the constraints of (II). We have

$$\frac{1}{||V||} \leq \frac{F(V,\zeta)}{||V||} = F(\frac{AV}{||AV||},\xi) \leq F(W^*,\xi) \ ,$$

since W^* is a solution of (I). Carrying on we get

$$\frac{1}{||V||} \leq \frac{F(A\tilde{V},\xi)}{||W^*||} = \frac{F(\tilde{V},\zeta)}{||\tilde{V}||} = \frac{1}{||V^*||}$$

in view of the definition of V^*. This completes the proof of the first part. Now let V^* be a solution of (II) and we claim that

$$W^* := \frac{AV^*}{||V^*||} = \frac{AV^*}{||AV^*||}$$

is a solution of (I). By construction W^* satisfies the constraints and it is also related to V^* as claimed in the theorem. Since $1 \leq F(V^*, \zeta) = F(AV^*, \xi)$ implies that

$$F(W^*, \xi) = \frac{F(V^*, \zeta)}{||V^*||} > 0 \tag{3}$$

and again we can again use the homogeneity of F. Let W be arbitrary such that it satisfies the constraints of (I) and let V be the unique vector such that $W = AV$. Since

$$F(\frac{V}{F(W,\xi)}, \zeta) = \frac{F(V,\zeta)}{F(AV,\xi)} = 1$$

and since V^* is a solution of (II), we have the relation

$$\frac{||V||}{F(W,\xi)} \geq ||V^*|| \ .$$

This relation allows us to complete the proof:

$$\frac{1}{F(W,\xi)} = \frac{||W||}{F(W,\xi)} = \frac{||V||}{F(W,\xi)} \geq ||V^*|| \geq \frac{||V^*||}{F(V^*,\zeta)} = \frac{1}{F(W^*,\xi)}$$

where the last equality follows from Eq. (3). □

Reconsidering the definition of the perceptron of optimal stability (Eq. (2)), one sees that only the constraint $W^2 = 1$ is necessary to ensure the uniqueness of the solution (if the perceptron problem is solvable). Therefore, it is possible to give a slightly different version of theorem 1.

Corollary 1 *With the notation and assumptions of theorem 1 the following are equivalent:*

I $\max_W \min_\nu (W^t \xi^\nu)/||W||$ *subject to* $\exists V \in R^m : W = SV$.

II $\min V^2$ *subject to* $V^t \zeta^\nu \geq 1$ *for all* ν.

Let V^* be a solution of (II) then for arbitrary $\lambda > 0$ $W^* = \lambda S U D^{-0.5} V^*$ is a solution of (I). The norm $|| \cdot ||$ is the standard euclidean norm of R^n.

To avoid unnecessary confusions, note that the two problems are different. However, given a solution of either of them, one finds a solution of the other. We omit the proof since it is straight forward using theorem 1. We would like to note that given a perceptron problem with symmetry constraint S we can use the above idea to map the pattern ξ^ν to $\zeta^\nu := S^t \xi^\nu$. We then end up with an ordinary perceptron problem for the pattern ζ^ν involving only the actually free variables. If V is a solution of this problem we recover a solution of the original problem W by $W = SV$. It was pointed out to us by T. Kozek that the original problem is generally a more computer intensive min-max problem, compared to the convex quadratic programming problem we ended up with. For possible future extensions of this theorem we have kept the proof abstract.

The AdaTron algorithm offers one possibility for solving the above quadratic programming problem. The calculation and numerical experiments in [2, 4, 3] indicate that this algorithm is very efficient. Let us now sketch the basic underlying ideas. First of all since any component of the weight vector W orthogonal to all patterns ξ^ν would just increase the length of W without modifying the field strengths E^ν, the weight vector W has to be of the form

$$W = \frac{1}{n} \sum_{\nu=1}^{P} x_\nu \xi^\nu \quad , \quad \xi^\nu \in R^n \ . \tag{4}$$

The x_ν are called the embedding strength of the patterns ξ^ν. Furthermore since the optimal solution W^* lies inside the cone generated by the ξ^ν, it follows from Farkas' Lemma [19] that all the x_ν can be chosen positive. The algorithm starts with arbitrary positive x_ν and a corresponding weight vector W defined by Eq. (4). Then the weights and the embedding strengths are updated in a serial or in a parallel manner, via the rule

$$W^{n+1} = W^n + \delta x_\nu \xi^\nu \quad \text{and} \quad x_\nu^{n+1} = x_\nu^n + \delta x_\nu \ , \text{ or}$$
$$W^{n+1} = W^n + \sum_\nu \delta x_\nu \xi^\nu \quad \text{and} \quad x_\nu^{n+1} = x_\nu^n + \delta x_\nu$$

with δx_ν given by:

$$\delta x_\nu := \max(-x^\nu, \gamma(1 - E^\nu)) \ .$$

Hence, the Adatron algorithm can be viewed as an adaptive perceptron algorithm which takes into account the information furnished by Farkas' Lemma. For the serial version convergence was proved for $0 < \gamma < 2$. Going through the proof in [4, 2], one notices that the proof includes a more general formulation

$$0 < \gamma < \frac{2n}{\max_\nu \xi^{\nu t} \xi^\nu} \quad \xi^\nu \in R^n \ . \tag{5}$$

which suite our purpose. Since the parallel version essentially performs a gradient descent, γ has to be sufficiently small in order to ensure convergence. For more details and proofs of convergence we refer the reader to the original papers of Anlauf, Biehl and Kinzel [2, 3, 4]. Our numerical studies with binary patterns showed that one gets satisfactory results for $\gamma = 1.3$ for the serial version, and $\gamma = 0.3$ for the parallel one. If the problem is solvable the algorithm converges and all δx_ν are zero. Even if the problem is not solvable, the δx_ν will not be zero but the weight vector will not change further. This is mainly due to the fact the the algorithm minimizes the convex function $\sum_\nu (1 - E^\nu)^2 \theta(1 - E^\nu)$ over the convex set $x_\nu \geq 0$ [4, 3].

Finally we are going to relate the notion of stability to the notion of robustness as used in statistical design by norm-body inscription [8, 22]. The robustness can be defined for arbitrary norms on vector spaces. We will give the definition only for the euclidean norm, since it is the only one we will consider. For a more thorough treatment of this concept as applied to neural networks, we refer the reader to the paper of Seiler, Schuler and Nossek [23].

Definition: For any solution of the perceptron problem W with symmetry constraints

- the (relative) robustness in weight space $r_w(W)$ is defined as the solution of the following optimization problem

$$\max r \ \text{ subject to } \ \forall \Delta W : ||\Delta W|| = r||W|| \text{ implies } (W + \Delta W)^t \xi^\nu \geq 0 \ .$$

- the relative robustness in pattern space $r_p(W)$ is defined as the solution of the following optimization problem

$$\max r \ \text{ subject to } \ \forall \Delta \xi^\nu : ||\Delta \xi^\nu|| = r||\xi^\nu|| \text{ implies } W^t(\xi^\nu + \Delta \xi^\nu) \geq 0 \ .$$

- the absolute robustness in pattern space $R_p(W)$ is defined as the solution of the following optimization problem

$$\max R \text{ subject to } \forall \Delta\xi^\nu : ||\Delta\xi^\nu|| = R \text{ implies } W^t(\xi^\nu + \Delta\xi^\nu) \geq 0 \,.$$

We will usually omit the term relative when referring to the robustness in weight space, since it is the only one of interest for our purposes. The following Lemma provides us with a more explicit formulation of robustness and links it to the previously defined stability (Eq. (1)).

Lemma 1 *For any solution of the perceptron problem W with constraints*

- *the absolute robustness in pattern space (or optimal stability, see Eq.(1)) is given by:*

$$R_p(W) = \frac{1}{||W||} \min_\nu W^t \xi^\nu \;.$$

- *the relative robustness in pattern and weight space is given by:*

$$r_p(W) = r_w(W) = \frac{1}{||W||} \min_\nu \frac{W^t\xi^\nu}{||\xi^\nu||} \;.$$

Proof: We will just prove the claim for the robustness in weight space, since the proofs are almost identical. Let us first check that the above defined $r_w(W)$ satisfies the constraints given by the definition of robustness. Let ΔW be arbitrary such that $||\Delta W|| = \min(W^t\xi^\nu/||\xi^\nu||)$. For any pattern ξ^ν we obtain:

$$(W + \Delta W)^t \xi^\nu = W^t\xi^\nu + ||\Delta W|| \frac{\Delta W^t \xi^\nu}{||\Delta W||} \geq W^t\xi^\nu - ||\Delta W|| \, ||\xi^\nu|| \geq W^t\xi^\nu - \frac{W^t\xi^\nu}{||\xi^\nu||} ||\xi^\nu|| = 0 \,.$$

Let us denote by r^* the solution of the defining optimization problem and we have just shown that $r_w(W) \leq r^*$. We have yet to prove the opposite. To this end let ξ^μ be the pattern for which $W^t\xi^\nu/||\xi^\nu||$ is minimal. Let us define $\Delta W := -r^*||W||\xi^\mu/||\xi^\mu||$ and check the defining property.

$$(W + \Delta W)^t \xi^\mu = W^t\xi^\mu - r^* \frac{||W||}{||\xi^\mu||} ||\xi^\mu||^2 \overset{!}{\geq} 0$$

which shows that $r^* \leq r_w(W)$, and thereby completing the proof. □

We used the notion W and ξ and were referring to the weight and pattern space respectively. The analogous definitions can be given for the parameters V and the "new" pattern ζ; in this case the lemma remains valid. We will only use the notion of (relative) robustness in the parameter space $r_{pa}(V)$. It seems that the other notions are of minor importance. Combining the above lemma with the theorem we get the following corollary.

Corollary 2 *Within the notation of theorem 1 set $\zeta_1^\nu := \alpha D^{-0.5} U^t S^t \xi^\nu$, $\zeta_2^\nu := \zeta_1^\nu/||\xi^\nu||$ and $\zeta_3^\nu := \alpha S^t\xi^\nu/||S^t\xi^\nu||$ for arbitrary $\alpha > 0$. Let V_j^* for $j = 1,2,3$ be the solution of*

$$\min V^2 \text{ subject to } V^t\zeta_j^\nu \geq 1 \;,\; \nu = 1, \cdots, P.$$

Then within an arbitrary scale factor the perceptron W^* of

- *optimal absolute robustness in pattern space (or optimal stability see Eq. (2)) is given by* $W^* = SUD^{-0.5}V_1^*$.
- *optimal robustness in weight space is given by* $W^* = SUD^{-0.5}V_2^*$.
- *optimal robustness in parameter space is given by* $W^* = SV_3^*$.

Proof: Due to lemma 1 the perceptron of optimal robustness in pattern space is given by the solution of

$$\max_W \frac{1}{||W||} \min_\nu W^t \xi^\nu \text{ subject to } \exists V \in R^m \, : \, W = SV. \tag{6}$$

This in turn is, due to corollary 1, equivalent to

$$\min V^2 \text{ subject to } \frac{1}{\alpha} V^t \zeta^\nu \geq 1 \, . \tag{7}$$

A solution of Eq. 6 is given by $W^* = \lambda SUD^{-0.5}V_1^*$ if V_1^* is a solution of Eq. (7) for arbitrary $\lambda > 0$. It is easily verified that Eq. (7) is equivalent to

$$\min V^2 \text{ subject to } V^t \zeta^\nu \geq 1 \, . \tag{8}$$

The solution V_1^* and the solution V_2^* of Eq. (8), are related by $V_1^* = \alpha V_2^*$, which proves the claim. The proofs of the remaining claims are almost identical and we will omit them. □

We have just proved, that we are free to introduce an overall scale factor for the "new" pattern ζ_j^ν ($j = 1,2,3$). Hence the perceptron of optimal robustness in weight space and optimal robustness (relative or absolute) in pattern space coincide for pattern ξ^ν of equal length (e.g. binary patterns). But there are gray scale image processing tasks [7, 10] for which Neural Networks have proved useful and we have to make a choice. For the problem of circuit design the robustness in weight space is probably the most important one. In order to be able to employ the AdaTron algorithm, we have yet to establish an upper bound for $\zeta_2^{\nu t} \zeta_2^\nu$ (Eq. (5)). Depending on the details of the realization it might be that we are mainly interested in the robustness in parameter space. By an appropriate scaling of the ζ_3^ν ($||\zeta_3^\nu|| = \sqrt{m}$ m the dimension of parameter space) we can guarantee, independent of the dimension, that γ (see Eq. (5)) can be chosen according to $0 < \gamma < 2$. If we are interested in processing gray scale images and are really simulating the network, it might prove useful to design a network with the greatest absolute robustness in the pattern space (figure 1). In any case the AdaTron algorithm is able to solve the problem by just taking the actual constraints into account.

3 Application to attractor networks

Let us first of all consider spin-glass like models [1, 16] with a (zero-temperature) parallel dynamics [14, 1, 11] and arbitrary symmetry constraints on the weight matrix $W \in R^{n \times n}$ i.e.

$$S(t+1) = \text{sgn}(WS(t)) \;\; t \in N_0, S \in \{-1,+1\}^n, \exists V \in R^m : \, W = SV$$

Our goal is to prescribe transients given by pairs of patterns (ρ^ν, ξ^ν), i.e. we are looking for a weight matrix such that $\rho^\nu = \text{sgn}(W\xi^\nu)$ which in turn is equivalent to $\rho_i^\nu (W\xi^\nu)_i > 0$ for all patterns ξ^ν and sites i. This isn't yet quite the perceptron problem we were considering. Hower by extending the patterns ξ^ν and the matrix W we have;

$$\begin{aligned} \tilde{\xi}^{\nu i\,t} &:= (0,\cdots,0,\rho_i^\nu \xi^{\nu t},0,\cdots,0) \in R^{n\cdot n} \text{ (at the } i\text{-th place) and} \\ \tilde{W}^t &:= (W_1^t,\cdots,W_n^t) \in R^{n\cdot n}\ ,\ W_i \text{ the (column) vector of the } i\text{-th row of } W. \end{aligned}$$

The above problem is now in exactly the form discussed in the last section (with (νi) considered as the pattern index). Again by choosing the proper normalization of the patterns $\tilde{\xi}^{\nu i}$ one can define the appropriate optimization problem for the different notions of robustness. Since the absolute robustness in pattern space is the most convenient one to choose, we will use it thereafter. With the above notation the attractor network of optimal absolute robustness in the pattern space is given by solving:

$$\max_{\tilde{W}} \min_{\nu i} \tilde{W}^t \tilde{\xi}^{\nu i} \text{ subject to } \tilde{W}^2 = 1 \text{ and } \tilde{W} = SV$$

for some V. Actually this could have been written as well in the original matrix notation. But now it has exactly the form used in the last section and the notations are consistent. Notice that the lengths of the patterns $\tilde{\xi}^{\nu i}$ of the constrained perceptron problem are given by: $||\tilde{\xi}^{\nu i}||^2 = \rho_i^{\nu 2}||\xi^\nu||^2$. They are needed for the formulation of optimal robustness in the weight space (corollary 2). Due to theorem 1, assuming S^tS is non-singular, the above problem is equivalent to

$$\min V^2 \text{ subject to } V^t\zeta^{\nu i} \geq 1\ ,\ \forall \nu, i \tag{9}$$

where the new patterns are computed according to $\zeta^{\nu i} = D^{-0.5}U^tS^t\tilde{\xi}^{\nu i}$, and the solution of the original problem is given by $\tilde{W}^* = \lambda \cdot SUD^{-0.5}V^*$. Actually these equation can be simplified if one takes into account the special structure of the extended patterns $\tilde{\xi}^{\nu i}$. Writing down the constraints in matrix notation

$$\tilde{W}^t = (W_1^t,\cdots,W_n^t) = V^t(S^{1t},\cdots,S^{nt}) = V^tS^t$$

we immediately deduce that $S^tS = \sum_i(S^{it}S^i)$. The S^i are the symmetry constraints of the weight vector W_i of the i-th neuron. Using this notation the new patterns are given by $\zeta^{\nu i} = D^{-0.5}U^tS^{it}\rho_i^\nu\xi^\nu$ and the optimal weights by $W_i^* = \lambda \cdot S^iUD^{-0.5}V^*$. The orthogonal transformation U is such that $U^t\sum(S^{it}S^i)U = D$ is diagonal. Considering the robustness in the parameter space we can even drop the factor $D^{-0.5}U^t$ and $UD^{-0.5}$ in the preceding equations. Further the ζ still have to be scaled according to which kind of robustness we are interested in (see corollary 2). In any case we have to solve a perceptron problem of reduced dimension $m = \dim V$.

For some constraints the dimension of the parameter space can be quite large, which is for example the case for the Hopfield or Little model [12, 14] with a symmetric weight matrix W. If we consider constrains of the form $W_i = S_iV_i$, where the parameters can be different for each neuron, the learning problem can be solved in parallel. We used lower indices to distinguish this case from the one previously considered. This situation arises for example if we give up the translation invariance of the templates, yielding a generalized DTCNN (or CNN). The symmetry matrix S now has a diagonal block form with entries S_i. The problem then basically decouples for each site i, and we have to consider n perceptron problems of reduced dimensions $m_i = \dim V_i$.

Before we work out explicitly an example we would like to discuss to what extend this learning approach is applicable to neural networks other than those previously mentioned. If we just prescribe the fixed points of an neural network, i.e. $\rho^\nu = \xi^\nu$ this is valid in a strict sense for spin-glass like neural networks with a zero-temperature Monte-Carlo dynamics [1, 16], for time continuous neural networks with a piece-wise linear output function [6], or for more generally sigmoid output function with saturation. Even if the output function is just a sigmoid (e.g. tanh) it seems that if the weights are scaled properly we can still proceed as indicated. It is worthwhile remarking that, at least for binary patterns, we should exclude selfcouplings [4]. Briefly, if the number of patterns to be stored is sufficiently large, then the weight vectors $W_i = (0, \cdots, 1, \cdots, 0)$ are the optimal ones. But since they stabilize any pattern, no information is stored. Let us finally briefly consider the case when a transient is explicitly prescribed. Our experiences with CNN [26, 21] indicate, that for the time continuous networks mentioned above, this learning approach is still applicable. Although to our knowledge a proof of this has not yet been given.

4 Application to rotationally invariant CNNs

We are now going to apply the above theory to the problem of finding the most stable rotationally invariant r-neighbourhood [6, 18] templates and bias current for a CNN, or DTCNN on a square grid. By rotationally invariant we mean that any rotation of $k \cdot 90°$ with $k \in N$ around the center of the template and any reflection along the diagonals, as well as horizontal or vertical lines through the center, leaves the template invariant (Eq. (12)). These kind of templates have proved useful for image processing tasks [26, 15, 11], as for example the one considered at the end of this section, which are independent of the image's orientation.

The learning problem for CNNs and DTCNNs is given by a triple of patterns $(\rho^\nu, \xi^\nu, \sigma^\nu)$ where ρ^ν is the desired output, ξ^ν is the initial state, and σ^ν is the input pattern. In a yet abstract notation, not taking into account the grid structure, the CNN of optimal robustness in pattern space is obtained by solving

$$\max_{A,B,I} \min_{k,\nu} \rho_k^\nu (A\xi^\nu + B\sigma^\nu + I)_k \text{ subject to } A^2 + B^2 + I^2 = 1, \text{ and } A = S_1 V^1;\ B = S_2 V^2;\ I = S_3 V^3$$

where we have yet to specify the symmetries S_j. A is the feedback matrix, B the control matrix and I is the bias current (vector) [6]. The important point is that the constraints are separate for each entity. Using the notation of the last section let us define

$$\tilde{W}^T := (A_1^t, \cdots, A_n^t, B_1^t, \cdots, B_n^t, I_1^t, \cdots, I_n^t)$$

and the parameters V as $V^t := (V_1^t, V_2^t, V_3^t)$. The symmetry matrix is then block diagonal with entries S_j for $j = 1, 2, 3$. The square of the lengths of the so called extended patterns is given by $\rho_k^{\nu 2} (\xi^{\nu 2} + \sigma^{\nu 2} + 1)$. As noted (see the previous section) they are needed for the formulation of optimal robustness in weight space. Hence, the original problem is equivalent to

$$\min(V_1^2 + V_2^2 + V_3^2) \text{ subject to } V_1^t \zeta_1^{\nu i} + V_2^t \zeta_2^{\nu i} + V_3^t \zeta_3^{\nu i} \geq 1\ ,\ \forall \nu, i\ , \tag{10}$$

where $\zeta_j^{\nu i} = \rho_i^\nu D_j^{-0.5} U_j^t S_j{}^t \xi^\nu$ for $j = 1, 2, 3$, and U_j is the orthogonal matrix which diagonalizes $S_j{}^t S_j$ to D_j. In turn the optimal "weights" are given by $A^* = \lambda S_1 U_1 D^{-0.5} V_1^*$ $(\lambda > 0)$ and analogously for B and I. Again, a proper normalization of the pattern according to corollary 2 is necessary and we can drop the factor $U_j D_j^{-0.5}$ when considering the robustness in parameter space. Note that each symmetry S_j is given by $S_j^t = (S_j^{1t}, \cdots, S_j^{nt})$, where S_j^i is the rotational symmetry of each site i (see the last section).

With this at hand let us make the notions more explicit so we can define the symmetries. Let each pattern of the above triple $(\rho^\nu, \xi^\nu, \sigma^\nu)$ be an $(m+2r) \times (n+2r)$ matrix α_{kl}^ν for $-1+r \le k \le m+r$ and $-1+r \le l \le n+r$. With m and n we denote the size of the CNN lattice, and by r the size of the neighbourhood. The field strength at site (i,j) of pattern ν is given by

$$E_{ij}^\nu := \rho_{ij}^\nu \sum_{\substack{-1+r\le k\le m+r \\ -1+r\le l\le n+r}} (A_{kl}^{ij}\xi_{kl}^\nu + B_{kl}^{ij}\sigma_{kl}^\nu) + I^{ij} \text{ for } 1 \le i \le m, 1 \le j \le n \,.$$

A^{ij} is the feedback operator, B^{ij} the control operator and I^{ij} the bias current [6] of cell (ij) yet without any symmetry constraints.

Let us first of all consider the symmetries S_1 and S_2 in the above notation. They certainly are equal and from now on we will just write S for either of them. In order to be able to specify the rotational symmetry S^{ij} for each site (ij) let us define

$$E_{st} := \begin{cases} 1 & \text{in the } s\text{-th row and } t\text{-th column} \\ 0 & \text{otherwise} \end{cases}$$

for $0 \le |s|, |t| \le r$ as a basis of the vector space of r-neighbourhood templates. For the space of rotationally invariant templates we choose as a basis the matrices

$$B_{pq} := \frac{1}{\sqrt{|\{p,q\}|}} \sum_{\substack{-r\le s,t\le r \\ \{s,t\}=\{p,q\}}} E_{st} \text{ for } 0 \le p \le q \le r \tag{11}$$

where $\{s,t\} = \{p,q\}$ iff $|s| = p$ and $|t| = q$, or $|s| = q$ and $|t| = p$. And $|\{p,q\}|$ denotes the size of the set $\{(s,t) \in Z^2 : \{s,t\} = \{p,q\}\}$ which can be 1,4 or 8. Hence, any rotationally invariant template can be written as

$$T = \sum_{0\le p\le q\le r} a^{pq} B_{pq} \,.$$

An explicit example for a 2-neighbourhood template of this kind is provided below

$$\begin{pmatrix} a^{22}/2 & a^{12}/\sqrt{8} & a^{02}/2 & a^{12}/\sqrt{8} & a^{22}/2 \\ a^{12}/\sqrt{8} & a^{11}/2 & a^{01}/2 & a^{11}/2 & a^{12}/\sqrt{8} \\ a^{02}/2 & a^{01}/2 & a^{00} & a^{01}/2 & a^{02}/2 \\ a^{12}/\sqrt{8} & a^{11}/2 & a^{01}/2 & a^{11}/2 & a^{12}/\sqrt{8} \\ a^{22}/2 & a^{12}/\sqrt{8} & a^{02}/2 & a^{12}/\sqrt{8} & a^{22}/2 \end{pmatrix} \,. \tag{12}$$

As already noted, the explicit parametrization used is important if we considers robustness in the parameter space. The above choice might not be the most practical one, but it simplifies the computation since, by construction, we have

$$\sum_{\substack{s,t \\ \{s,t\}=\{p,q\}}} T_{st}^2 = a^{pq2} \text{ for } 0 \le p \le q \le r \,. \tag{13}$$

We are now in a position to define the rotational symmetry S^{ij} $(1 \leq i \leq m\,;\, 1 \leq j \leq n)$

$$W^{ij}_{kl} := (S^{ij}T)_{kl} := \begin{cases} T_{k-il-j} & \text{if } |k-i| \leq r \text{ and } |l-j| \leq r \\ 0 & \text{otherwise} \end{cases}$$

for $-1+r \leq k \leq m+r\,;\; -1+r \leq l \leq n+r$ and T as defined in Eq. (4). Let us compute

$$\begin{aligned} T^t \textstyle\sum_{1\leq i,j}(S^{ij^t}S^{ij})T &= \textstyle\sum_{1\leq i,j}(S^{ij}T)^t(S^{ij}T) &= \textstyle\sum_{1\leq i,j}\sum_{-1+r\leq k,l}(S^{ij}T)^2_{kl} \\ = \textstyle\sum_{-1+r\leq k,l}\sum_{\substack{s,|s|\leq r\\ t,|t|\leq r}} T^2_{st} &= (m+2r)(n+2r)\textstyle\sum_{0\leq p\leq q\leq r} a^{pq^2} \end{aligned}$$

where we have used Eq.(13) to show that $\sum S^{ij^t}S^{ij}$ is $(m+2r)(n+2r)$ times the identity matrix. By a slight abuse of notation we still used the transpose notation for the template T, and for the symmetry S. This transpose notation does not refer to the matrix structure of the template but solely considers it as a vector. This certainly could have also been achieved by an arbitrary enumeration of the coefficients. Still we feel, since we are only using the defining properties of the transposition of vectors and linear maps and the relationship between the (euclidean) scalar product and the transposition, that this notation is well suited and will not cause errors.

The symmetry matrix S_3, which ensures the cell independence of the bias current is given by: $I^{ij} = (S^{ij}_3 I) := I$ for all i and j. We can immediately deduce that $\sum_{ij}(S^{ij^t}_3 S^{ij}_3)$ is equal to $m \cdot n$ times the identity matrix. Hence we can choose the identity matrix for each orthogonal transformation U_j $(j = 1,2,3)$. Since we are free to choose an overall scale factor for the patterns (corollary 2), we choose $\sqrt{(m+2r)(n+2r)}$ which just cancels the factor provided by $D_1^{-0.5}$ and $D_2^{-0.5}$. We are left with the task to compute the "new" pattern. For the "new" pattern $\zeta^{\nu ij}$ generated by the input pattern ξ^ν we get:

$$\begin{aligned} T^t\zeta^{\nu ij} &= \rho^\nu_{ij}T^t(S^{ij^t}\xi^\nu) &= \rho^\nu_{ij}(S^{ij}T)^t\xi^\nu &= \rho^\nu_{ij}\textstyle\sum_{-r+1\leq k,l}(S^{ij}T)_{kl}\xi^\nu_{kl} \\ &= \rho^\nu_{ij}\textstyle\sum_{\substack{s,|s|\leq r\\ t,|t|\leq r}} T_{st}\xi^\nu_{s+it+j} &\overset{!}{=} \textstyle\sum_{0\leq s\leq t\leq r} a^{st}\zeta^{\nu ij}_{st} \end{aligned}$$

Hence the "new" patterns $\zeta^{\nu ij}$ (actually $\zeta^{\nu ij}_1$ would be more consistent with Eq. (10)) are given by

$$\zeta^{\nu ij}_{st} = \frac{\rho^\nu_{ij}}{\sqrt{|\{s,t\}|}} \sum_{\substack{-r\leq p,q\leq r\\ \{p,q\}=\{s,t\}}} \xi^\nu_{i+pj+q} \quad \text{for } 0 \leq s \leq t \leq r\,,\; 1 \leq i \leq m\,,\; 1 \leq j \leq n\,. \qquad (14)$$

The "new" input patterns $\epsilon^{\nu ij}$ are computed in the same manner by replacing ξ by σ in Eq. (14). With the above definition of S^3 the "new" patterns for the bias current, temporarily named $\alpha^{\nu ij}$, are given by:

$$I\alpha^{\nu ij} = I^t\alpha^{\nu ij} = \sqrt{\frac{(m+2r)(n+2r)}{m\cdot n}}\,I^t(S^{3ij^t}\rho^\nu_{ij}1) = \sqrt{\frac{(m+2r)(n+2r)}{m\cdot n}}\,I\rho^\nu_{ij}\;.$$

Putting the pieces together and considering that some applications of CNN involve gray scale images, we obtain.

Lemma 2 *Let $M := r^2 + 3r + r$ be the dimension of the parameter space. In order to obtain the rotationally invariant r-neighbourhood CNN on a square lattice with*

- *the optimal absolute robustness in the pattern space, let*

$$F_{\nu ij} = 1 \;\; ; \;\; G = \sqrt{\frac{(m+2r)(n+2r)}{m \cdot n}} \; ; \; \gamma_{\max} = \frac{2M}{M-1+G^2} \, .$$

- *the optimal (relative) robustness in the weight space, let*

$$F_{\nu ij} = \frac{\sqrt{\tilde{F}}}{\sqrt{(\rho_{ij}^{\nu})^2\{||\xi^{\nu}||^2 + ||\sigma^{\nu}||^2 + 1\}}} \; ; \; G = \sqrt{\frac{(m+2r)(n+2r)}{m \cdot n}} \; ; \; \gamma_{\max} = \frac{2M}{M-1+G^2}$$

with $\tilde{F}$ *defined as* $\tilde{F} := \min_{\nu}\left(\{||\xi^{\nu}||^2 + ||\sigma^{\nu}||^2 + 1\}\min_{ij}(\rho_{ij}^{\nu})^2\right)$, *and* $||\xi^{\nu}||^2 = \sum \xi_{kl}^{\nu\,2}$ *where the sum is taken over all* $-1+r \le k \le m+r$ *and* $-1+r \le l \le n+r$.

- *the optimal (relative) robustness in the parameter space (for the parametrization of the templates as defined below), let*

$$F_{\nu ij} = \frac{\sqrt{M}}{\sqrt{||\zeta^{\nu ij}||^2 + ||\epsilon^{\nu ij}||^2 + G^2}} \; ; \; G = 1 \; ; \; \gamma_{\max} = 2$$

with $||\zeta^{\nu ij}||^2 = \sum \zeta_{pq}^{\nu ij\,2}$ *where the sum is taken over all* $0 \le p \le q \le r$.

And solve the following convex quadratic optimization problem:

$$\min \left[\sum_{0 \le p \le q \le r} (a^{pq\,2} + b^{pq\,2}) + I^2\right] \quad \text{subject to} \quad F_{\nu ij}\left[\sum_{0 \le p \le q \le r} (a^{pq}\zeta_{pq}^{\nu ij} + b^{pq}\epsilon_{pq}^{\nu ij}) + IG\rho_{ij}^{\nu}\right] \ge 1$$
$$\forall \nu, 1 \le i \le m, 1 \le j \le n \, .$$

The serial AdaTron algorithm yields a solution for $0 < \gamma < \gamma_{\max}$ *in all cases if* $\xi_{ij}^{\nu}, \rho_{ij}^{\nu}$ *and* σ_{ij}^{ν} *are from the interval* $[-1,1]$ *for all* ν, i, j. *The* $\zeta^{\nu ij}$ *(resp.,* $\epsilon^{\nu ij}$*) are the "new" initial (resp., input) patterns defined by Eq. (14). Within an arbitrary overall scale factor, the feedback template* a*, the control template* b*, and bias current* I *are given by:*

$$a = \sum_{0 \le p \le q \le r} a^{*pq} B_{pq} \; ; \; b = \sum_{0 \le p \le q \le r} b^{*pq} B_{pq} \;\; \text{and} \;\; I = G \cdot I^* \, .$$

The matrices B_{pq} *are defined in Eq. (11) and* a^{*pq}, b^{*pq} *and* I^* *are the solutions of the above optimization problem.*

In most practical cases $m, n >> r$ and we can set $G = 1$, which simplifies the formulas considerably. Furthermore often (e.g. binary images) the lengths of the so called extended patterns $(\rho_{ij}^{\nu})^2\{||\zeta^{\nu}||^2 + ||\sigma^{\nu}||^2 + 1\}$ are equal for all ν, i, j and the first two cases coincide. As already mentioned, the above formulas only apply for a CNN on a square grid. Reconsidering the calculation we see that the vector structure of the templates was the only one crucially used. We just employed the matrix notation to avoid an explicit numbering of the parameters used. So it should be possible to derive similar equations for CNNs on any regular grid. Even

the dimension of the grid seems to be of minor importance.
Proof: Let us first of all compute the norm of the "new" patterns $\zeta^{\nu ij}$:

$$||\zeta^{\nu ij}||^2 = \sum_{0\le s,t\le r} \zeta_{st}^{\nu ij\,2} = \sum_{0\le s,t\le r} \frac{\rho_{ij}^{\nu\,2}}{|\{s,t\}|}\left[\sum_{\substack{-r\le p,q\le r\\ \{p,q\}=\{s,t\}}} \xi_{i+pj+q}^{\nu}\right]^2 \le \frac{1}{2}(r+1)(r+2)\,,$$

The last inequality is due to the assumption, that each $\xi_{kl}^{\nu}, \rho_{kl}^{\nu} \in [-1,1]$. Hence $||\zeta^{\nu ij}||^2 + ||\epsilon^{\nu ij}||^2 + G^2 \le M - 1 + G^2$ where M is the dimension of the parameter space as defined above. The factors $F_{\nu ij}$ are chosen in accordance with corollary 2 within an arbitrary scale factor. We have yet to ensure the proper range of γ according to Eq. (5). Since $F_{\nu ij} \le 1$ for the first two cases and it follows from the above calculation, that $M-1+G^2$ is an upper bound on the lengths of the "new" patterns (including the factor $F_{\nu ij}$). This ensures the range for γ as claimed. In the third case the numerator is chosen so that the square of the lengths of the "new" patterns is equal to M, the dimension of the parameter space. Hence the range for γ is as claimed (Eq. (5)). We have shown that in any case the orthogonal transformations U_j can be taken as the identity matrix. The "weights" are calculated according to Eq. (10) and the remarks thereafter. For the first two cases the templates and bias currents are given by $\sqrt{(m+2r)(n+2r)}D_j^{-0.5}$ times the appropriate entity. Since D_1 and D_2 are equal to $(m+2r)(n+2r)$ times the identity matrix and D_3 is given by $m \cdot n$ times the identity matrix, the result follows. In the third case no adjustments are necessary and G is set to one (corollary 2 and Eq. (10)).

4.1 Simulation results

For our preliminary simulations we used a 16×16 grid with a 1-neighbourhood. We determined the optimal templates and bias currents for some classical binary image processing tasks, such as the edge detector, and the convex corner detector [5]. We will refer to the pattern values with indices higher than 16 or lower than 0 as border pattern values. We were experimenting with border pattern values equal to 0 or -1, the value of the background. For larger grids we expect, that the effect of these zeros on the template values decreases, which might result in a slight change of the template coefficients compared to our results. For both tasks the input pattern σ^{ν} were equal to the initial pattern ξ^{ν} for the transients we trained. Since we also want to ensure that the desired output is a fixed point, we trained each output pattern ρ^{ν} with the original input pattern σ^{ν} to be a fixpoint. So for each training pattern (see figures 2–3) we obtained two patterns to be learned: $(\rho^{\nu}, \sigma^{\nu}, \sigma^{\nu})$ and $(\rho^{\nu}, \rho^{\nu}, \sigma^{\nu})$. The template values and bias currents are collected in table 1 for border pattern values equal to -1 and 0, and the optimal robustness in the weight and the parameter spaces. Since the patterns have equal length the notions of absolute robustness in pattern space and robustness in weight space coincide. The templates are recovered as shown in Eq. (12) not taking the coefficients into account.

Actually we used more patterns than the above mentioned to train the network. The patterns in figures 2–3 were just the first ones. In order to obtain templates that are not only optimal with respect to the training patterns, but also to the task itself we have employed the following procedure. We started out with one training pattern and kept adding new patterns while monitoring the optimal template values (figure 4). Since the optimal values changed by less than 0.1 percent, after 4 patterns were presented we are confident that the template values are optimal with respect to the task. As already noted the template values still can be scaled arbitrarily. The values given in table 1 are chosen, so that the embedding strength $E^{\nu ij}$ of each pattern ξ^{ν} at site (i,j) is greater than or equal to two. We think that this value is appropriate for CNNs since it is well inside the region of saturation. We made several simulations with trained as well as untrained patterns, and in all cases the CNN performed the desired task with this scaling.

5 Conclusion

We have shown that if one considers gray scale patterns, or imposes constraints on the weights, it is sensible to distinguish three different kind of robustness; namely, the absolute robustness in the pattern space (also called optimal stability), the robustness in the weight space, and the robustness in the parameter space. Depending on the application, any of these notions can be the appropriate choice (see the remarks after corollary 2). If the patterns are binary (or more general of equal length) the first two notions coincide. In any case, the problem of finding the most robust weights, just taking actual constraints into account, is equivalent to a convex quadratic programming problem. This problem can be solved very efficiently by the AdaTron algorithm, which shows an exponential decay of errors. Even if not solvable, the algorithm terminates and finds a reasonable partial solution [2, 4]. We have applied this algorithm to two classical binary image processing task for CNNs and have calculated the optimal template coefficients and bias currents under various assumptions.

Actually if we consider the problem of learning transients of an attractor network, the above approach is rigorous only for spin-glass like networks with a (zero-temperature) parallel dynamics [14, 11, 1]. However our experience with CNN [26, 21, 23] and our simulations indicate that it is probably also applicable to time continuous networks with an output function that exhibits saturation. This point certainly deserves a deeper investigation. In any case one of our crucial assumptions in the statement of the learning problem is that the transients are prescribed explicitly, step by step. This obviously is a drawback since different possibilities exist, while still performing the same overall task, and not all of them admit a solution.

Acknowledgement

One of us, P. Nachbar, would like to thank Dr. J.K. Anlauf for many valuable discussions.

References

[1] D.J. Amit, H. Gutfreund, and H. Sompolinsky. *Phys.Rev.*, A(32):1007, 1985.

[2] J.K. Anlauf and M. Biehl. *Europhys. Lett.*, (10):687, 1989.

[3] J.K. Anlauf and M. Biehl. Properties of an adaptive Perceptron algorithm. In R. Eckmiller, G. Hartmann, and G. Hauske, editors, *Parallel Processing in Neural Systems and Computers*. Elsevier, 1990.

[4] M. Biehl, J.K Anlauf, and W. Kinzel. Perceptron learning by constrained optimization: The AdaTron algorithm. In *Neurodynamics 90*, 1990.

[5] L.O. Chua and Lin Yang. Cellular Neural Networks: Applications. *IEEE Tr. CAS*, 35(10):1273, 1988.

[6] L.O. Chua and Lin Yang. Cellular Neural Networks: Theory. *IEEE Tr. CAS*, 35(10):1257, 1988.

[7] K.R. Crounse, T. Roska, and L.O. Chua. Image halftoning with Cellular Neural Networks. Technical Report UCB/ERL M91/106, University of California, Berkeley, November 1991.

[8] S.W. Director and G.D. Hachtel. The simplicial approximation approach to design centering. *IEEE Tr. CAS*, (24):363, 1977.

[9] E. Gardner. *J.Phys.*, A(21):257, 1988.

[10] H. Harrer and J.A.Nossek. Multilayer Discrete-Time Cellular Neural Networks using time-variant Templates. Technical Report TUM-LNS-TR-92-1, TU Muinch, 1992.

[11] H. Harrer and J.A. Nossek. Discrete-Time Cellular Neural Networks. *Int. J. Circuit Theory Appl.*, July 1992. to be published.

[12] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554, 1982. Biophysics.

[13] W. Krauth and M. Mezard. *J.Phys.*, A(20):L745, 1987.

[14] W.A. Little. *Math.Biosci.*, (19):101, 1974.

[15] T. Matsumoto, L.O. Chua, and R. Furukawa. Cnn cloning Template: Hole-filler. *IEEE Tr. CAS*, (37):635, 1990.

[16] M. Mezard, G. Parisi, and M.A. Virasoro. *Spin Glass Theory and Beyond.* World Scientific, Singapore, 1987.

[17] M. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, first edition, 1969.

[18] J.A. Nossek, G. Seiler, T. Roska, and L.O. Chua. Cellular Neural Networks: Theory and circuit design. *Int. J. Circuit Theory Appl.*, 1992. to be published.

[19] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization.* Prentice-Hall, 1982.

[20] F. Rosenblatt. *Principles of Neurodynamics.* Spartan, New York, 1962.

[21] S. Schwarz and W. Mathis. A design algorithm for Cellular Neural Networks. In *Proc. of the Micro-Neuro 91*, Muinch.

[22] G. Seiler and A.J. Schuler. Some notes on statistical design by norm-body inscription. Technical Report TUM-LNS-TR-92-2, TU Muinch, 1992.

[23] G. Seiler, A.J. Schuler, and J.A. Nossek. Design of robust Cellular Neural Networks. Technical Report TUM-LNS-TR-91-13, TU Muinch, 1991.

[24] B. Widrow and M.E. Hoff. *WESCON Convention Record*, volume 4. 1960.

[25] B. Widrow and M.A. Lehr. 30 years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. *Proc. of the IEEE*, 78(9):1415, 1990.

[26] F. Zou, S. Schwarz, and J.A. Nossek. Cellular Neural Network design using a learning algorithm. In *Proc. of the First IEEE International Workshop on Cellular Neural Networks and their Application*, Budapest, 1990. IEEE.

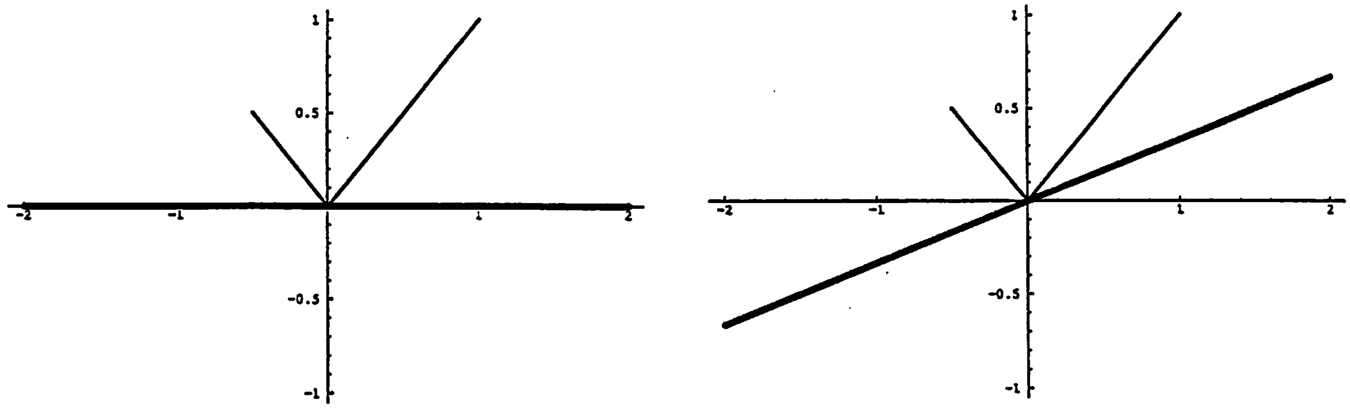

Figure 1: Relative and absolute robustness in pattern space

	border pattern value −1		border pattern value 0	
	weight sp.	param. sp.	weight sp.	param. sp.
	edge detector			
a_{00}	4	5.52967	4	5.53504
a_{01}	-2	-2.72245	-2	-2.72429
a_{11}	0	-0.134687	0	-0.13947
b_{00}	4	5.33429	4	5.33771
b_{01}	-2	-3.11322	-2	-3.11895
b_{11}	0	0.565003	0	0.570453
I	-2	-3.31133	-2	-3.31734
	convex corner detector			
a_{00}	7	5.81904	7	5.82013
a_{01}	-4	-3.13878	-4	-3.13962
a_{11}	-2	-1.47737	-2	-1.47757
b_{00}	7	5.78939	7	5.7905
b_{01}	-4	-3.16843	-4	-3.16925
b_{11}	-2	-1.52184	-2	-1.52203
I	-12	-10.2994	-12	-10.3013

Table 1: Optimal template values

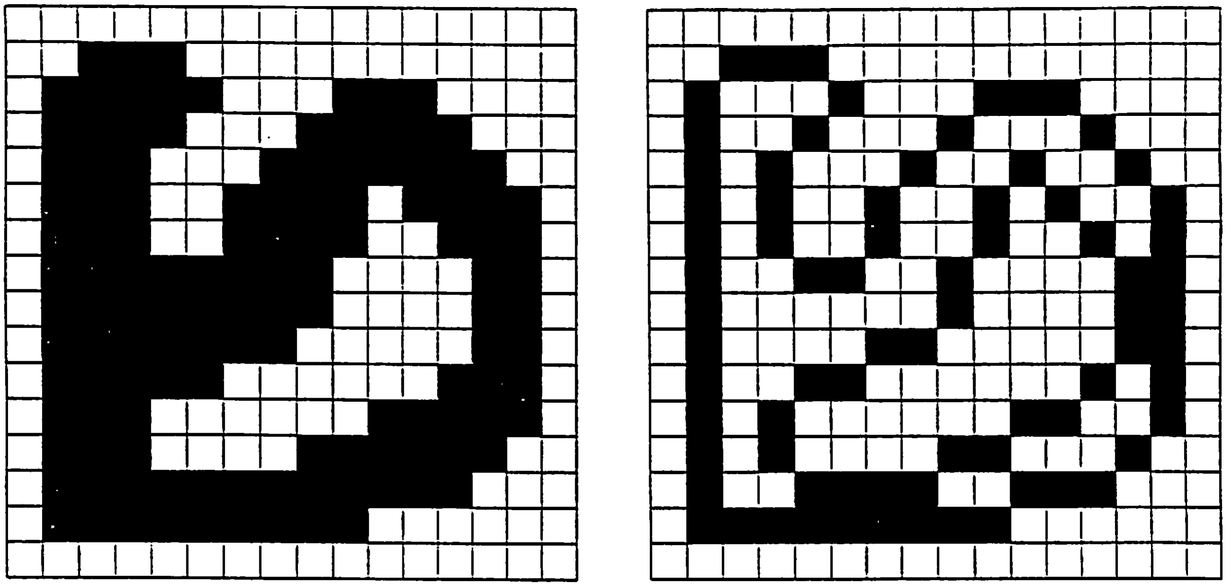

Figure 2: Training pattern for the edge detector (input pattern, output pattern)

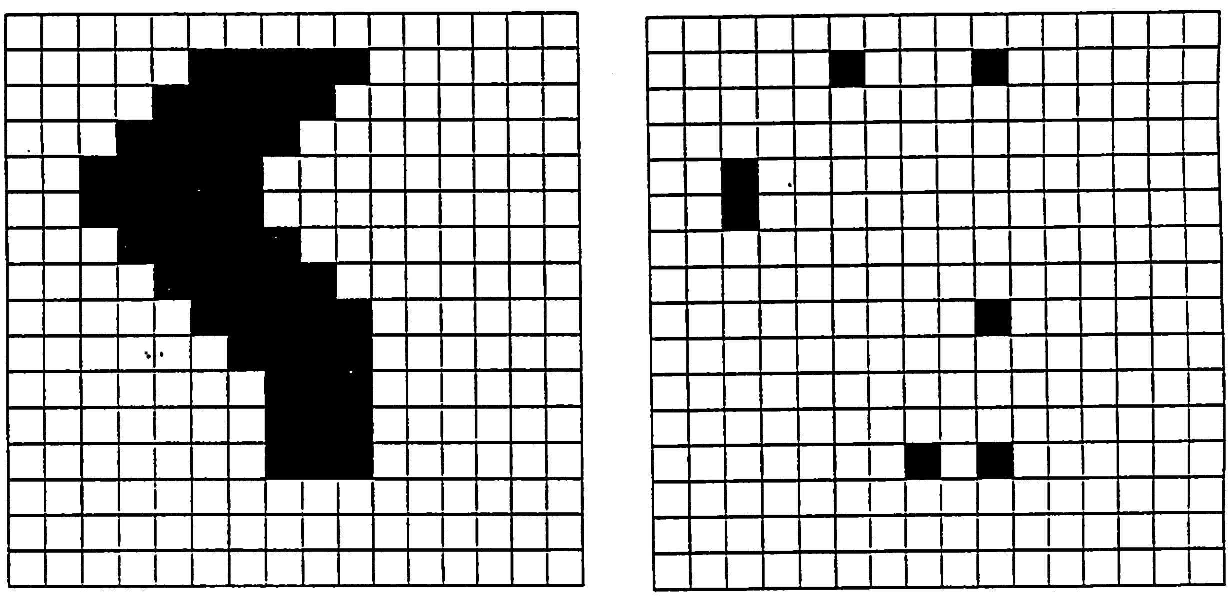

Figure 3: Training pattern for the convex corner detector (input pattern, output pattern)

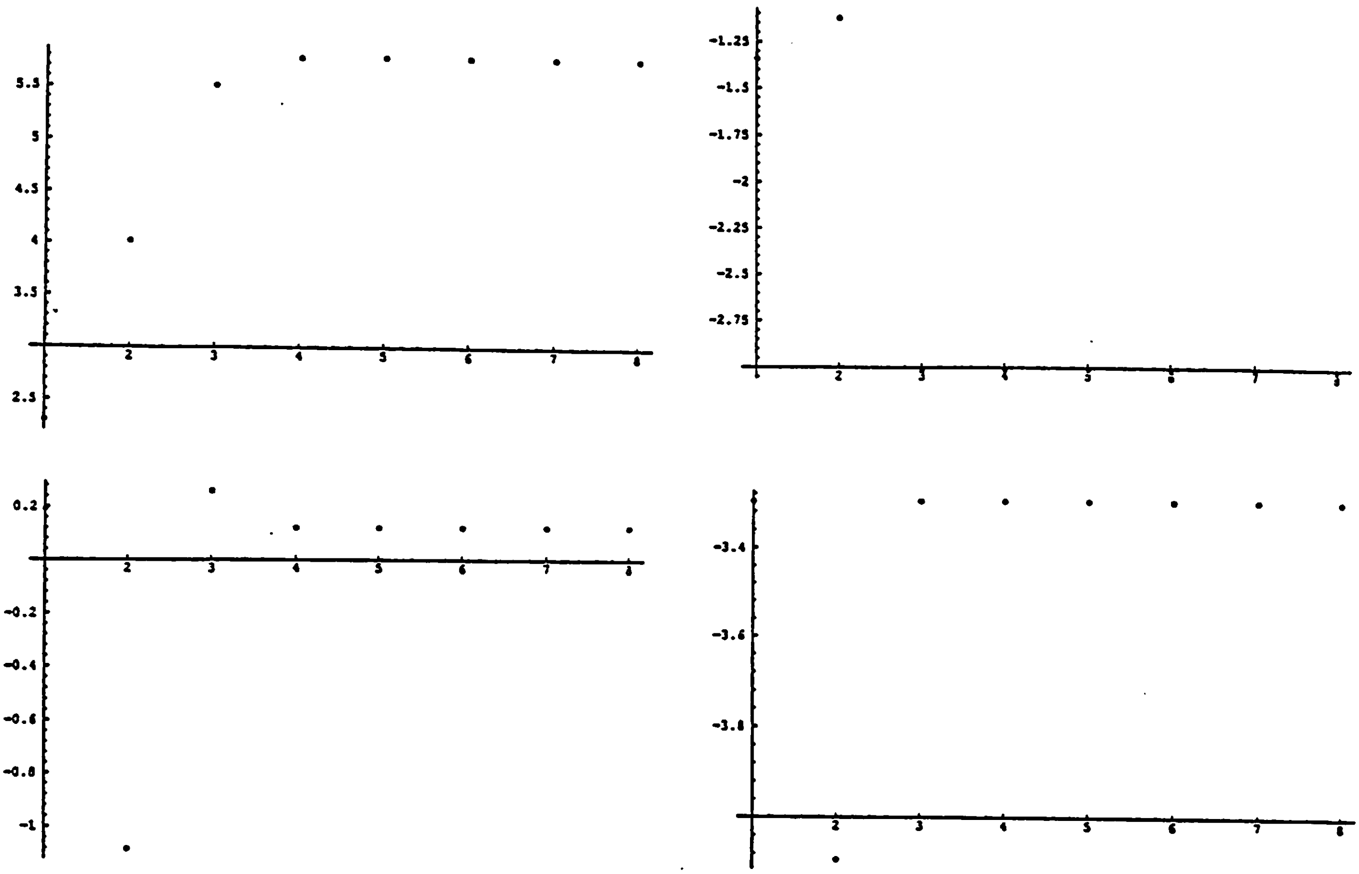

Figure 4: The optimal template values as a function of the number of training images