HD-THEP-00-02
WUE-ITP-2000-007

Learning structured data from unspecific reinforcement

Michael Biehl[1], Reimer Kühn[2], Ion-Olimpiu Stamatescu[2,3]

[1]Institut für Theoretische Physik, Universität Würzburg, Am Hubland
D-97074 Würzburg, Germany
[2]Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16 und 19
D-69120 Heidelberg, Germany
[3]Forschungstätte der Evangelischen Studiengemeinschaft, Schmeilweg 5,
D-69118 Heidelberg, Germany

March 22, 2001

Abstract

We show that a straightforward extension of a simple learning model based on the Hebb rule, the previously introduced Association-Reinforcement-Hebb-Rule, can cope with "delayed", unspecific reinforcement also in the case of structured data and lead to perfect generalization.

1 Introduction

Learning from unspecific reinforcement may be essential in various contexts, both natural and artificial, where, typically, the results of particular actions add to a final consequence which only is valuated. The freedom residing in each step is not (or only partially) controlled directly and the learner must cope with the necessity of improving its performance only from information concerning the final success of a complex series of actions. This is a recognized difficult kind of problems ("class III" problems in the classification of Hertz et al. [1] – see, e. g. [2], [3] for evolved AI algorithms), which, however, may be of vital significance in natural or simulated life situations.

It is therefore important to find out whether there are *simple and robust procedures* for such problems, *which might have developed under natural conditions and which may be basic also for artificial learning rules.* In previous works [4],[5] we have introduced an "Association-Reinforcement" learning model based on the following conception:

1. For each given input (external situation) the agent answers with an action (operation) depending solely on the input and on its instantaneous internal (cognitive) structure, and simultaneously strengthens (in its internal structure) the *blind* association between this particular input and action.

2. At the end of a series of actions (path) the final success is judged. Then the associations "situation - operation" which have been involved on this path are re-weighted equally (in the internal structure), depending only on the final success: *unspecific* reinforcement.

In [5] we studied an implementation of this model to a classification problem for perceptrons, the *Association-Reinforcement-Hebb-rule*. This implementation is mathematically tractable by well developed methods and allows, under circumstances, to obtain exact results. Our analysis showed some amazing properties:

a) Despite the fact that feedback on the learner's performance enters its learning dynamics only in an *unspecific* way in that it cannot be associated with single identifiable correct or incorrect associations, convergence of the AR-Hebb-algorithm in the sense of *asymptotically perfect generalization* can be proven to occur under rather general conditions.

b) For given initial conditions, this convergence depends on the learning parameters characterizing the 2 steps described above; in particular none of these steps can be completely inhibited. Alternatively, for given algorithm parameters convergence may depend on initial conditions.

In detail the dynamics of this algorithm was found to be very complex and interesting, being controlled by fixed points in the pre-asymptotic regime, and having a continuous set of asymptotic convergence laws. These results could easily be extended to the more realistic case where in the second step the unspecific reinforcement is randomly applied to only part of the associations achieved in the first step (the agent does not recall everything it has done on the trial) [5]. Further extensions concern the question of structured data and of multi-layer perceptrons.

Structured data represent a more involved classification problem and it is known that for $L = 1$ (*specific* reinforcement in our terminology) when teacher and data vector are not fully aligned (or exactly uncorrelated) the usual Hebb rule does not lead to convergence of the student vector onto that of the teacher, while the perceptron algorithm does [6]. The AR-Hebb-rule has an intrinsic parameter λ and for $L = 1$ it interpolates between the perceptron algorithm (for $\lambda = 0$) and the usual Hebb-rule (for $\lambda = 1/2$), which are both known to converge. For $L > 1$ the AR-Hebb-rule does not converge for $\lambda = 0$ [5]. The situation is succintly described in Table 1.

	$L = 1$ specific reinforcement	$L > 1$ unspecific reinforcement
unstructured data	$\lambda \geq 0$	$\lambda > 0$ [5]
structured data	$\lambda = 0$ [6]	this study

Table 1: *Convergence condition for the AR-Hebb-rule.*

It is therefore a non-trivial question whether the unspecific reinforcement problem ($L > 1$) can be solved for structured data and in particular, whether some immediate extension of the AR-Hebb-rule can be shown to converge in this case. This is, however, an important question if we want to argue that the AR-learning algorithm is a basic process with a certain natural basis, since structured data is the generic situation. It is therefore this question which we shall address in this paper. In a future publication we shall treat the problem of the committee machine as a first step to multi-layer perceptrons.

In section 2 we shall describe the learning model in the general setting and in section 3 we shall discuss its convergence properties providing numerical and analytical results. Thereby we shall briefly recall the non-structured data case and then concentrate on the general, structured data case. Section 4 contains the conclusions.

2 Learning rule for perceptrons under unspecific reinforcement

We consider one layer perceptrons with Ising or real number units s_i, real weights (synapses) J_i and one Ising output unit:

$$s = \mathrm{sign}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N} J_i\, s_i\right) \tag{1}$$

Here N is the number of input nodes, and we put no explicit thresholds. The network (student) is presented with series of patterns $s_i = \xi_i^{(q,l)}$, $q = 1, ..., Q$, $l = 1, ..., L$ to which it answers with $s^{(q,l)}$. A training period consists of the successive presentation of L patterns. The answers are compared with the corresponding answers $t^{(q,l)}$ of a teacher with pre-given weights B_i and the average error made by the student over one training period is calculated:

$$e_q = \frac{1}{2L}\sum_{l=1}^{L} |t^{(q,l)} - s^{(q,l)}|. \tag{2}$$

The training algorithm consists of two parts:

I. - a "blind" Hebb-type *association* at each presentation of a pattern:

$$J_i^{(q,l+1)} = J_i^{(q,l)} + \frac{a_1}{\sqrt{N}}\, s^{(q,l)}\, \xi_i^{(q,l)}; \tag{3}$$

II. - an "unspecific" but graded *reinforcement* proportional to the average error e_q (2), also Hebbian, at the end of each training period,

$$J_i^{(q+1,1)} = J_i^{(q,L+1)} - \frac{a_2}{\sqrt{N}}\, e_q \sum_{l=1}^{L} r_l\, s^{(q,l)}\, \xi_i^{(q,l)}. \tag{4}$$

where r_l is a dichotomic random variable:

$$r_l = \begin{cases} 1 & \text{with probability} \quad \rho \\ 0 & \text{with probability} \quad 1-\rho \end{cases} . \tag{5}$$

Because of these 2 steps we called this algorithm "association/reinforcement(AR)-Hebb-rule". The relevant parameter is the ratio $\lambda = a_1/a_2$. We are interested in the behavior with the number of iterations q of the generalization error $\epsilon_g(q)$:

$$\epsilon_g(q) = \frac{1}{\pi}\arccos\left(\frac{\boldsymbol{J}\cdot\boldsymbol{B}}{|\boldsymbol{J}|\;|\boldsymbol{B}|}\right), \tag{6}$$

in particular we shall test whether the behavior of $\epsilon_g(q)$ follows a power law at large q:

$$\epsilon_g(q) \simeq \text{const } q^{-p} \ . \tag{7}$$

The training patterns $\{\xi_i^{(q,l)}\}$ are generated randomly from the following distribution:

$$\begin{aligned} P(\boldsymbol{\xi}) &= \frac{1}{2}\sum_{\sigma=\pm 1} P(\boldsymbol{\xi}|\sigma) \\ P(\boldsymbol{\xi}|\sigma) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}(\xi_i - m\sigma C_i)^2} \end{aligned} \tag{8}$$

and we take:

$$\boldsymbol{C}^2 = \boldsymbol{B}^2 = N, \quad \boldsymbol{C}\cdot\boldsymbol{B} = \eta N \tag{9}$$

with fixed, given m, η. Notice the following features:

a) During training the student only uses its own associations $\boldsymbol{\xi}^{(q,l)} \leftrightarrow s^{(q,l)}$ and the average error e_q which does not refer specifically to the particular steps l.

b) Since the answers $s^{(q,l)}$ are made on the basis of the instantaneous weight values $\boldsymbol{J}^{(q,l)}$ which change at each step according to eq. (3), the series of answers form a correlated sequence with each step depending on the previous one.[1] Therefore e_q measures in fact the performance of a "path", an interdependent set of decisions.

c) In contrast with the case studied in [5] the patterns can now have a structure. This introduces essential differences to the previous situation, as we shall see in the next section.

d) We explicitly account for imperfect recall at the reinforcement step by the parameter ρ (5). This introduces a supplementary, biologically motivated randomness which, as already suggested in [5], does not appear to introduce qualitative changes in the results, however (see section 3).

e) For $L = 1$ (and $\rho = 1$) the algorithm reproduces the usual "perceptron rule" (for $a_1 = 0$, i.e. $\lambda = 0$) or to the usual "unsupervised Hebb rule" (for $a_2 = 2a_1$, i.e. $\lambda = 1/2$) for on-line learning, for which the corresponding asymptotic behavior is known [8], [9], [6].

To study the learning behaviour we use Monte Carlo simulation and coarse grained analysis. The latter is provided by combining the *blind association* (3) during a learning period of L

[1] In the "thermodynamic" limit $L/N \to 0$, which is relevant for the coarse grained analysis in the next section, this aspect is lost - however, for L/N small this aspect is significant. This can be observed especially well in more "realistic" problems like the one described in [7], where a small neural network using the AR-Hebb rule controls a "robot" finding its way on a board with obstacles, the only feedback being the time to arrive at destination. In that case the "answers" of the robot are actual "actions", since it does not only change its internal structure (synapses) but it also takes corresponding steps on the board. In the case of the perceptron the answers are followed by "actions" only in the sense that the network changes its synapses accordingly.

elementary steps and the graded *unspecific reinforcement* (4) at the end of each learning period into one coarse grained step

$$J_i^{(q+1,1)} = J_i^{(q,1)} + \frac{1}{\sqrt{N}}(a_1 - a_2 e_q) \sum_{l=1}^{L} \mathrm{sign}\left(\frac{1}{\sqrt{N}} \sum_{j=1}^{N} J_j^{(q,l)}\, \xi_j^{(q,l)} \right) \xi_i^{(q,l)}, \tag{10}$$

$$e_q = \frac{1}{2L} \sum_{l=1}^{L} \left| \mathrm{sign}\left(\frac{1}{\sqrt{N}} \sum_{k=1}^{N} J_k^{(q,l)}\, \xi_k^{(q,l)} \right) - \mathrm{sign}\left(\frac{1}{\sqrt{N}} \sum_{k=1}^{N} B_k\, \xi_k^{(q,l)} \right) \right|. \tag{11}$$

For simplicity we take for the time being $r_l = 1$, i. e. $\rho = 1$ in eq. (5). We use

$$\alpha = qL/N, \quad \lambda = a_1/a_2 \tag{12}$$

and rescale everything with a_2, which means that we can take without loss of generality $a_2 = 1$ in (10),(11). We define the overlaps:

$$\mathcal{R}(\alpha) = \frac{1}{N} \boldsymbol{B} \cdot \boldsymbol{J}^{(q,l)} \quad , \mathcal{Q}(\alpha) = \frac{1}{N} [\boldsymbol{J}^{(q,l)}]^2 \quad , \mathcal{D}(\alpha) = \frac{1}{N} \boldsymbol{C} \cdot \boldsymbol{J}^{(q,l)} \quad . \tag{13}$$

Note that in the "thermodynamic limit" $L/N \to 0$ the overlaps are self-averaging and we can neglect the dependence of $\mathcal{R}$, $\mathcal{D}$ and $\mathcal{Q}$ on l. We shall follow standard procedures [9], [1], [10], [11]. Treating α as a continuous variable we obtain the coarse grained equations:

$$\frac{d\mathcal{R}}{d\alpha} = \left(\lambda - \frac{1}{2}\right) A_{JT} + \frac{1}{2L} A_{TT} + \frac{1}{2}\left(1 - \frac{1}{L}\right) S_{JT} A_{JT}, \tag{14}$$

$$\frac{d\mathcal{D}}{d\alpha} = \left(\lambda - \frac{1}{2}\right) A_{JC} + \frac{1}{2L} A_{TC} + \frac{1}{2}\left(1 - \frac{1}{L}\right) S_{JT} A_{JC}, \tag{15}$$

$$\begin{aligned} \frac{d\sqrt{\mathcal{Q}}}{d\alpha} &= \left(\lambda - \frac{1}{2}\right) A_{JJ} + \frac{1}{2L} A_{TJ} + \frac{1}{2}\left(1 - \frac{1}{L}\right) S_{JT} A_{JJ} \\ &+ \frac{1}{2\sqrt{\mathcal{Q}}} \left[\left(\lambda - \frac{1}{2}\right) S_{JT} + \frac{1}{4}\left(1 - \frac{1}{L}\right) S_{JT}^2 + \left(\lambda - \frac{1}{2}\right)^2 + \frac{1}{4L} \right], \end{aligned} \tag{16}$$

where the expectation values $A_{.}, S_{.}$ are given in Appendix (section 5.1). These equations describe the flow of the three quantities $\mathcal{R}$, $\mathcal{D}$ and $\mathcal{Q}$ with α and involve the data/teacher parameters m and η and the learning parameter λ. In terms of $\mathcal{R}$ and $\mathcal{Q}$, the generalization error is given by

$$\epsilon_g = \frac{1}{\pi} \arccos\left(\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}} \right) \quad . \tag{17}$$

3 Convergence behaviour of the AR-Hebb algorithm

3.1 Non-structured data

The case of non-structured data – $m = 0$ in eq. (36)-(43) – has been treated in [5], here we briefly recall some of the results for the later comparison with the structured data case.

Monte Carlo simulations indicate that in spite of the partial information contained in the unspecific reinforcement perfect generalization is achieved by the AR-Hebb algorithm and it depends on the learning parameters – see [5]. This intriguing behaviour is elucidated by the coarse grained analysis. In this case eqs. (14)-(16) reduce to two equations (for $\mathcal{R}$ and $\mathcal{Q}$) which have as general asymptotic solutions

$$\epsilon_g^2 \simeq \frac{1}{2\pi(\frac{1}{\lambda L}-1)}\,\alpha^{-1} + \tilde{c}_1 \alpha^{-\frac{1}{\lambda L}} \qquad \text{for } \lambda \neq \frac{1}{L}, \tag{18}$$

$$\epsilon_g^2 \simeq \left(\frac{1}{2\pi}\ln\alpha + \tilde{c}_2\right)\alpha^{-1} \qquad \text{for } \lambda = \frac{1}{L}, \tag{19}$$

$$\mathcal{Q} \simeq \frac{2}{\pi}\lambda^2\,\alpha^2 \tag{20}$$

at large α. We see that for $\lambda < \frac{1}{L}$ we obtain asymptotically perfect generalization, the dominant term exhibiting the usual power -1/2 , while for $\lambda > \frac{1}{L}$ the second term in (18) dominates and ensures again perfect generalization but with a different power law, $-1/(2\lambda L)$. For $\lambda = \frac{1}{L}$ we obtain logarithmic corrections – see eq. (19). Notice that these results hold also for $L = 1$ where one re-obtains the asymptotic behavior found in [8]. One can generally see that for $\lambda = 0$ one cannot have perfect generalization for $L > 1$ [5].

This learning algorithm is further characterized by highly interesting pre-asymptotics, dominated by two stationarity conditions, one for the self-overlap, $d\mathcal{Q}/d\alpha = 0$, and one for the generalization error $d\epsilon_g/d\alpha = 0$. For suitable values of the network parameters, the two stationarity conditions may simultaneously be satisfied, leading to fixed points of the learning dynamics, one of them stable and of poor generalization, the other with one attractive and one repulsive direction. Correspondingly, the flow is divided by a separatrix defined by a critical $\lambda_c(\mathcal{Q}_0)$ into trajectories leading to convergence according to the asymptotic behaviour (18)-(20) for $\lambda > \lambda_c(\mathcal{Q}_0)$, or to poor generalization otherwise.

The salient features of these results for the case of non-structured data are the convergence of the AR-Hebb-algorithm in the sense of *asymptotically perfect generalization* with a power law depending on the learning parameters L and λ and the existence of a minimal value $\lambda_c(\mathcal{Q}_0)$, fixed by the pre-asymptotic structure and below which the system is driven toward complete confusion. Notice also that the best convergence is achieved for λ just above λ_c. One last point concerns the recalling parameter ρ, eqs. (4),(5). A rough first quantitative characterization of this modification would be that it leads to an effective rescaling of the parameter λ, viz. $\lambda \to \lambda/\rho$, leading to a corresponding reduction of critical λ's by approximately a factor ρ. This is well supported by numerical simulations and we conclude that the algorithm is stable against this supplementary element of indeterminism.

3.2 Structured data

Numerical simulations indicate that for $m \neq 0$ and $0 < |\eta| < 1$ the behaviour of the algorithm for all ρ is more involved: generically, no convergence is found in this case for fixed values of the learning parameters. This agrees with the expectations, since, on the one hand the situation found at $L = 1$, $\rho = 1$ for structured data [8] could be expected to hold the more so for $L > 1$, namely that Hebb updating leads to a nonzero asymptotic generalization error. On the other hand, the situation found before for non-structured data should hold also for structured data, namely that for $L > 1$ the perceptron rule ($\lambda = 0$) (which for $L = 1$ was shown to work also in the structured data case [8]) does not lead to convergence. See Table 1.

In fact one can make a general argument that for fixed λ the AR-Hebb rule does not lead to perfect generalization for generically structured data. To obtain good generalization requires $\mathcal{R}/\sqrt{\mathcal{Q}} \to 1$ and $\mathcal{D}/\sqrt{\mathcal{Q}} \to \eta$, from which one may obtain the necessary dominant scaling (with $\mathcal{Q}$) of the various integrals appearing in (36)-(43), namely

$$\begin{aligned}
A_{JT} &\simeq \kappa \ , \\
A_{TJ} &\simeq \sqrt{Q}\,\kappa + o(\sqrt{Q}) \ , \\
A_{JJ} &\simeq \sqrt{Q}\,\kappa + o(\sqrt{Q}) \ , \\
A_{JC} &\simeq \kappa \ , \\
A_{TC} &\simeq \kappa \ , \\
A_{TT} &\simeq \kappa \ , \\
S_{JT} &\simeq 1 \ ,
\end{aligned}$$

with

$$\kappa = m\,\eta\,\varphi(m\,\eta) + \sqrt{\frac{2}{\pi}}\,\mathrm{e}^{-m^2\,\eta^2/2} \ . \tag{21}$$

This in turn would lead to the following asymptotic expressions for the flow equations (14)-(16) (at fixed λ)

$$\begin{aligned}
\frac{d\mathcal{R}}{d\alpha} &\simeq \kappa\,\lambda, \\
\frac{d\mathcal{D}}{d\alpha} &\simeq \kappa\,\lambda, \\
\frac{d\mathcal{Q}}{d\alpha} &\simeq \sqrt{\mathcal{Q}}\,\kappa\,\lambda + \lambda^2 \ .
\end{aligned}$$

The solution at large α would be $R \simeq \kappa\,\lambda\,\alpha + R_0$ and $\mathcal{D} \simeq \kappa\,\lambda\,\alpha + D_0$, while $\mathcal{Q}$ is asymptotically given through the implicit equation

$$\sqrt{\mathcal{Q}} \simeq \frac{1}{2}\,\kappa\,\lambda\,\alpha + \frac{\lambda}{\kappa}\,\ln(\sqrt{\mathcal{Q}}\,\kappa + \lambda) + \frac{1}{2}\,\kappa\,\lambda\,\kappa_0 \ . \tag{22}$$

Here R_0, D_0, and κ_0 are integration constants. Hence, asymptotically, $\sqrt{\mathcal{Q}} \sim \frac{1}{2}\,\kappa\,\lambda\,\alpha$ which is *incompatible* with the requirement of good generalization $\mathcal{R}/\sqrt{\mathcal{Q}} \to 1$. Thus the algorithm will *not* converge, if λ is kept fixed.

The question arises, however, whether a simple extension of the algorithm may not overcome the Odyssean dilemma hinted at in the beginning of this section. We hence suggest to tune the parameter λ such that it is large enough at small α to overcome the pre-asymptotic conditions and it tends to zero at large α in order to approach asymptotically the perceptron rule. As can be seen on Figs. 1, 2 this procedure is successful.

Since the situation is now much more complicated we shall not try to solve the general asymptotic problem, as we did in the case of non-structured data, but we shall limit ourselves

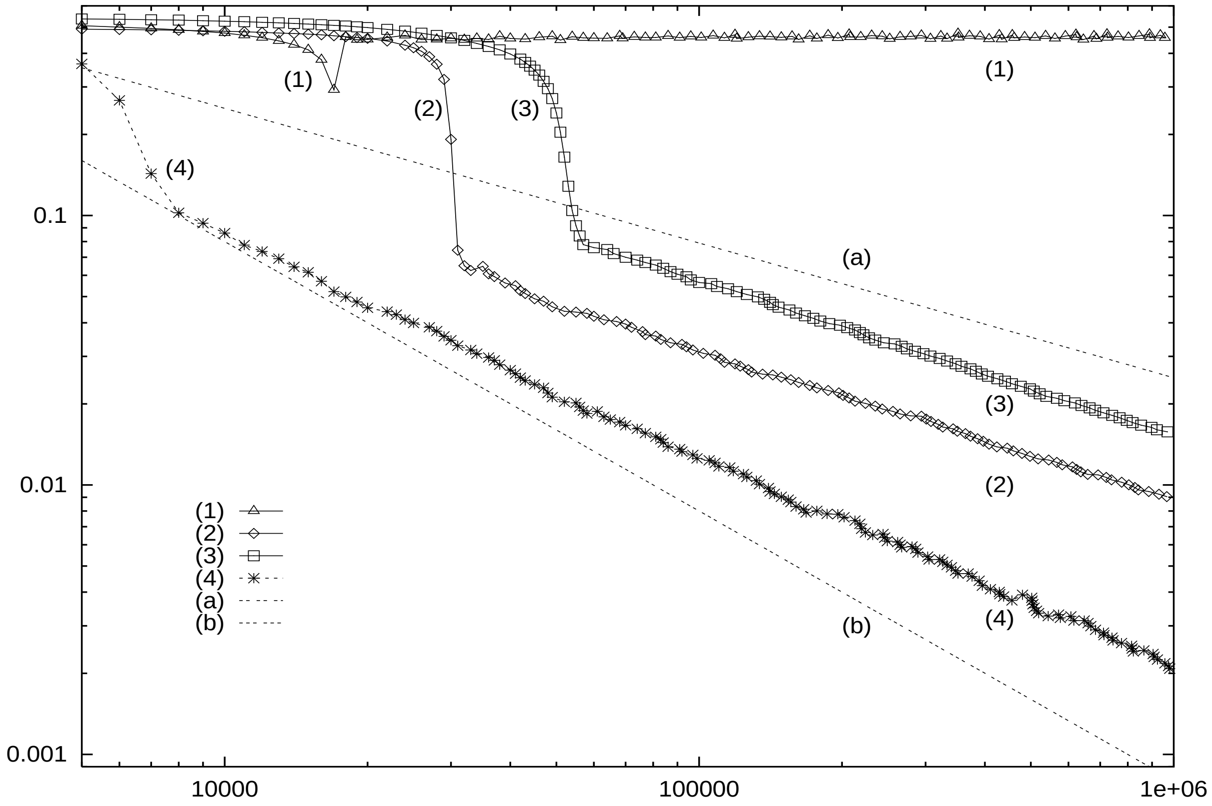

Figure 1: *Structured data, unspecific reinforcement eqs. (3-5), numerical results. Tuning dependence: generalization error* ϵ_g *vs* $q = \alpha N/L$ *for* $L = 10$, $N = 100$, *starting point* $\sqrt{\mathcal{Q}(0)} = 100$, *data/teacher parameters* $m = 1$, $\eta = 0.28$ *and recall probability* $\rho = 1$, *for two* λ*-tuning procedures.* (**1**) $\lambda = 4./\sqrt{\alpha}$; (**2**) $\lambda = 6./\sqrt{\alpha}$; (**3**) $\lambda = 10./\sqrt{\alpha}$; (**4**) $\lambda = 0.5\, e(\alpha; 1)$, *eq.(35).* (**a**),(**b**) *illustrative power laws* ($1/\sqrt{\alpha}$, $1/\alpha$, *respectively). Note the change in behaviour between* $\lambda_0 = 4$ (**1**) *and* $\lambda_0 = 6$, 10 (**2**), (**3**)*: in the first case there is no learning, in the other cases learning obtains. The asymptotic regime sets in earlier for* $\lambda = \lambda_0\, e(\alpha; 1)$ *tuning than for* $\lambda = \lambda_0/\sqrt{\alpha}$ *tuning - compare* (**4**) *with* (**2**), (**3**). *Both types of tuning show thresholds in* λ_0.

to prove that robust solutions exist. To do so, it is useful to introduce

$$x \equiv \pi\epsilon_g = \arccos\left(\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}}\right), \tag{23}$$

$$y \equiv \arccos\left(\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\right), \tag{24}$$

$$z \equiv \arccos\eta\ , \tag{25}$$

for which the geometric constraint

$$\sin\frac{y}{2} - \sin\frac{z}{2} = \omega\ \sin\frac{x}{2}, \quad |\omega| \leq 1 \tag{26}$$

should be noted. We can replace the variable y by ω. For a power-law tuning of the parameter λ according to

$$\lambda = \lambda_0\,\alpha^{-r}, \tag{27}$$

an ansatz of the form

$$\mathcal{Q} = c^2\,\alpha^{2q}, \tag{28}$$

$$\epsilon_g = a\,\alpha^{-p}, \tag{29}$$

$$\omega = b\,\alpha^{-s}\ , \tag{30}$$

with $p \simeq q \simeq r > s \geq 0$, if inserted into the flow equations (14)-(16), gives rise to following asymptotic equations.

$$2\sqrt{\mathcal{Q}}\,\frac{d\epsilon_g}{d\alpha} \simeq A_{11}\,\lambda + \frac{A_{12}}{\sqrt{\mathcal{Q}}} + A_2\,\epsilon_g, \tag{31}$$

$$2\sqrt{\mathcal{Q}}\sin\frac{z}{2}\,\epsilon_g\,\frac{d\omega}{d\alpha} \simeq B_{11}\,\lambda + \frac{B_{12}}{\sqrt{\mathcal{Q}}} + B_2\,\epsilon_g, \tag{32}$$

$$\frac{d\sqrt{\mathcal{Q}}}{d\alpha} \simeq C_0\,\lambda + C_1\,\epsilon_g, \tag{33}$$

Here the coefficients $A_\gamma, B_\gamma, C_\gamma$ are function of m, z, L and of ω (the explicit expressions are given in Appendix, section 5.2).

It is easy to see that an asymptotic solution can exist for:

$$p = q = r = 1/2, \quad s = 0, \tag{34}$$

which is therefore compatible with the assumptions used to derive the asymptotic equations (31)-(33). Then a, b, c are obtained as function of m, η, L for given λ_0, with some restrictions on the latter (notice that the coefficients $A_\gamma, B_\gamma, C_\gamma$ depend nonlinearly on ω, hence on b). For illustration, we show in Fig. 3 the values of a, b and c as function of λ_0 for $L = 10$, $m = 1$ and two values of the data-teacher overlap η. Notice that there is no asymptotic solution for λ_0 below $\simeq 0.2$. This is a new feature, compared to the unstructured data case: we find already in the asymptotics a lower bound on λ, namely $\lambda > \lambda_{0,c}^{asympt.}/\sqrt{\alpha}$.

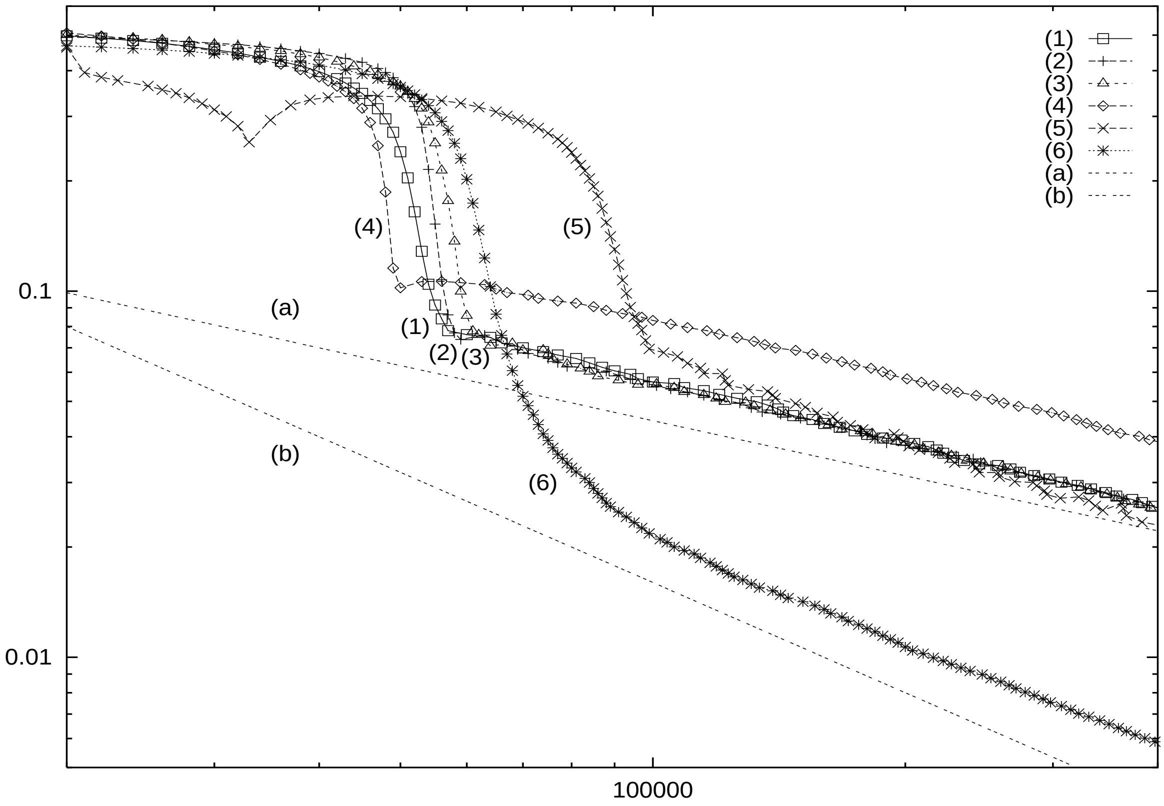

Figure 2: *Structured data, unspecific reinforcement eqs. (3-5), numerical results. Parameter dependence: generalization error* ϵ_g *vs* $q = \alpha N/L$ *for* $L = 10$, $N = 100$, *starting point* $\sqrt{\mathcal{Q}(0)} = 100$, *various data/teacher parameters* m, η, *recall probabilities* ρ *and variable types.* (**1**) $m = 1$, $\eta = 0.28$, $\rho = 1$, $\lambda = 10./\sqrt{\alpha}$; (**2**) $m = 1$, $\eta = 0.28$, $\rho = 1$, $\lambda = 10./\sqrt{\alpha}$, (Ising); (**3**) $m = 1$, $\eta = 0.28$, $\rho = 0.5$, $\lambda = 5./\sqrt{\alpha}$; (**4**) $m = 1$, $\eta = 0.6$, $\rho = 1$, $\lambda = 10./\sqrt{\alpha}$. (**5**) $m = 4$, $\eta = 0.28$, $\rho = 1$, $\lambda = 10./\sqrt{\alpha}$; (**6**) $m = 1$, $\eta = 0.$, $\rho = 1$, $\lambda = 10./\sqrt{\alpha}$; (**a**), (**b**) *illustrative power laws (*$1/\sqrt{\alpha}$, $1/\alpha$, *respectively). Note the robustness of learning (for* λ_0 *large enough) against change of variable types -* (**1**),(**2**) *- , recall behaviour (after the rescaling* $\lambda_0 \to w\,\lambda_0$*) -* (**1**),(**3**) *- and variation of the data parameters -* (**1**),(**4**),(**5**). *Notice that the* $\eta = 0$ *case* (**6**) *is essentially equivalent to the unstructured data case, which converges also for fixed* λ.

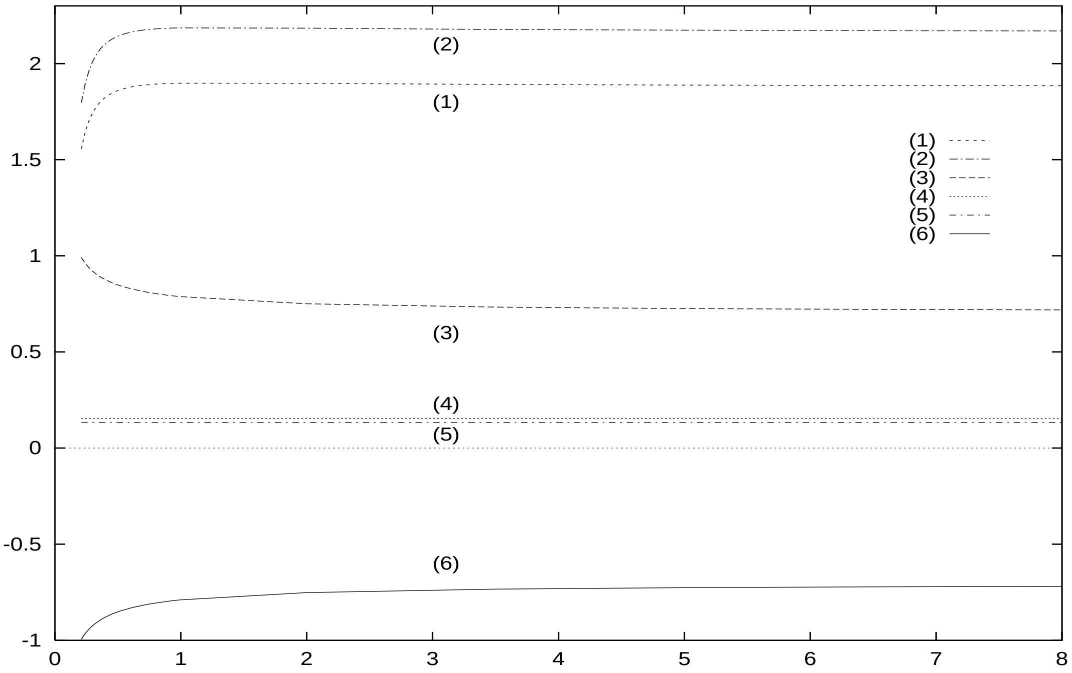

Figure 3: *Coefficients of asymptotic solutions eqs. (28)-(34) for $\lambda = \lambda_0/\sqrt{\alpha}$-tuning, for $L = 10$, $m = 1$ and two values of the overlap, η as functions of λ_0. There are generally two solutions: a, $\pm b$, c with b practically independent on η.* **(1),(2)***: a/λ_0 for $\eta = 0.28, 0.6$, respectively.* **(3),(6)***: $\pm b$.* **(4),(5)***: c for $\eta = 0.28, 0.6$, respectively. Note that there is no solution for $\lambda_0 < \lambda_{0,c}^{asympt.} \sim 0.2$.*

In Fig. 4 we show the solution of the full equations (14)-(16) for $\lambda \sim 1/\sqrt{\alpha}$-scaling – compare also with Figs. 1, 2 – which can be seen to approach the asymptotic solution (27)-(30),(34). The solutions are robust in the sense that for all m, η, L there exists a large region of λ_0 leading to convergence according to (34).

We have thus shown that the simple decrease of λ as $1/\sqrt{\alpha}$ provides convergence to asymptotic perfect generalization with the power $-1/2$. Alternatively, one can decrease λ as $1/\sqrt{Q}$, or as $e(\alpha; q_0)$, where

$$e(\alpha; q_0) = \frac{1}{q(\alpha) - q_0 + 1} \sum_{q=q_0}^{q(\alpha)=\alpha N/L} e_q \tag{35}$$

using the running "observed error" e_q (2) (this is in a sense the most natural choice, since the student only uses the observed error rate e_q to become increasingly sensitive to the feedback).

Convergence to perfect generalization is also observed for the slower $\lambda \sim \alpha^{-r}$-scaling with $r < 1/2$, albeit with correspondingly reduced convergence rates. This demonstrates the robustness of the general set up. We have not systematically studied the dependence of the convergence rates on r, however.

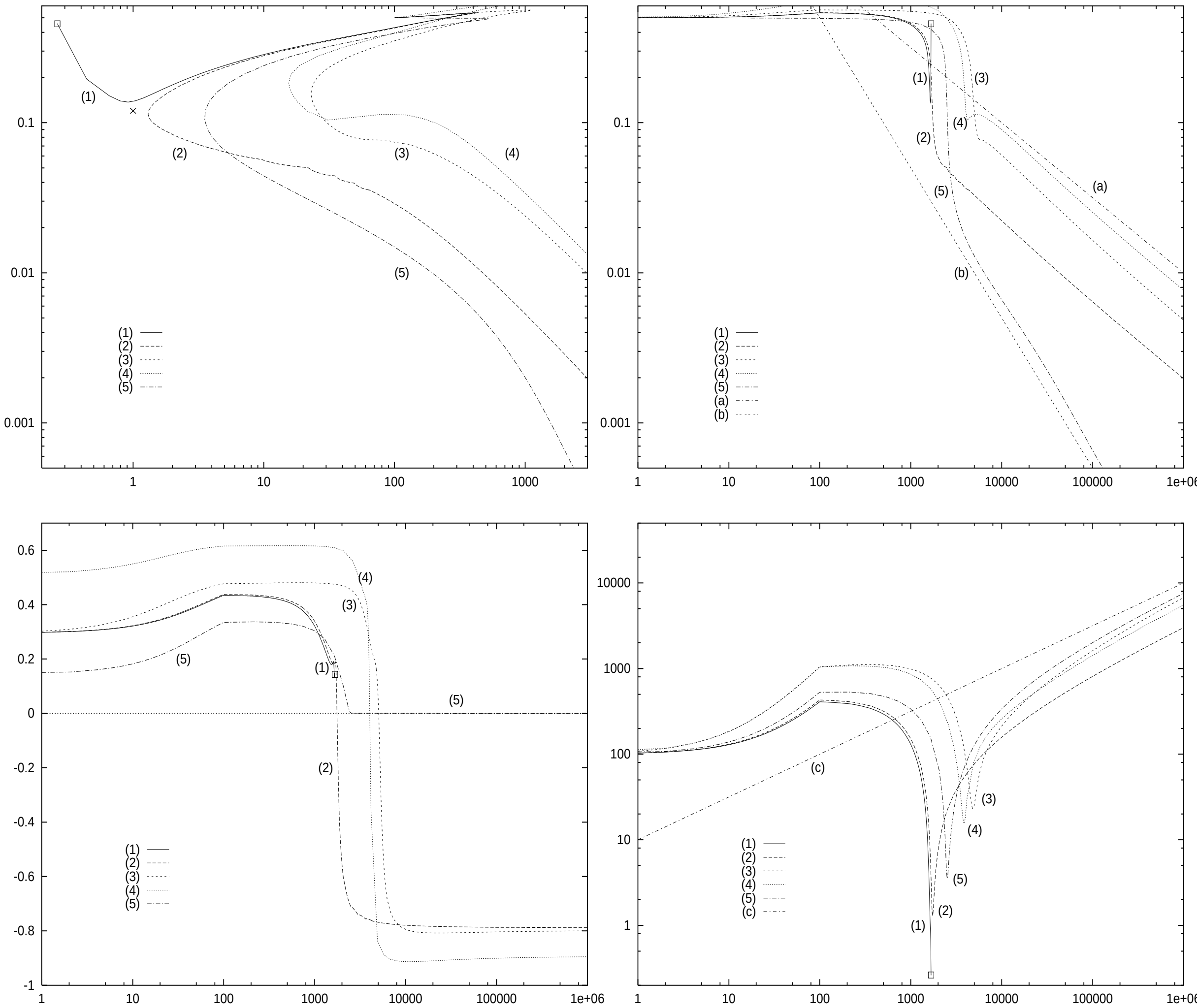

Figure 4: *Solution of the flow equations (14)-(16) for $\lambda = \lambda_0/\sqrt{\alpha}$-tuning, for $L = 10$, starting point $\sqrt{\mathcal{Q}(0)} = 100$, $m = 1$ and various overlaps η: flow with α in the ϵ_g-$\sqrt{\mathcal{Q}}$ plane (upper left), ϵ_g vs α (upper right), ω vs α (lower left) and $\sqrt{\mathcal{Q}}$ vs α (lower right).* **(1)** $\eta = 0.28$, $\lambda = 3.9/\sqrt{\alpha}$; **(2)** $\eta = 0.28$, $\lambda = 4.1/\sqrt{\alpha}$; **(3)** $\eta = 0.28$, $\lambda = 10./\sqrt{\alpha}$; **(4)** $\eta = 0.6$, $\lambda = 10./\sqrt{\alpha}$; **(5)** $\eta = 0.$, $\lambda = 5./\sqrt{\alpha}$; **(a)**, **(b)**, **(c)** *illustrative power laws ($1/\sqrt{\alpha}$, $1/\alpha$ and $\sqrt{\alpha}$, respectively). Note the change in behaviour between $\lambda_0 = 3.9$* **(1)** *and $\lambda_0 = 4.1$* **(2)** *and higher* **(3)**, *compare with Figs. 1, 2. We indicate by a square the position of the attractive fixed point of poor generalization and by a cross (on the upper left plot) the position of the second (attractive/repulsive) fixed point which determines the separatrix. Again, $\eta = 0$* **(5)** *is a special case.*

In all cases the pre-asymptotic regime is very important. Here similar phenomena to the non-structured data case seem to take place: the flow is divided by a separatrix defined by a $\lambda_{0,c}$ – see Fig. 4 (the MC simulation presents the same effect, compare Fig. 1). For $\lambda_0 < \lambda_{0,c}$ the flow runs into the attractive fixed point of poor generalization (on the Fig. 4 we stop the algorithm when the flow approaches the fixed point; from the simulation it can be seen that then the error stays practically 0.5 for ever – see curve (1) on Fig. 1). Because of the complexity of the variable and parameter space we could not obtain a systematic picture of the fixed point structure. Instead we tried to select some generic points in L, m, η, ρ and generic initial conditions, and study numerically the convergence properties of the eqs. (14)-(16) as function of the learning parameter λ. Therefore the following results are only illustrative. In the tables we give the bound $\lambda_{0,c}$ for various parameters for scaling with $1/\sqrt{\alpha}$ – Table 2, and for scaling with $e(\alpha; q_0)$ – Table 3. In the second case an averaging over e_q is needed in order to avoid strong fluctuations, but the amount of averaging is not essential. The dependence on the teacher/data overlap parameter η increases with the anisotropy parameter m as expected, and the flow structure may become more complex, as suggested by the curve (**5**) on Fig. 2. All these dependencies reflect themselves quantitatively in the threshold values $\lambda_{0,c}$, the qualitative picture, however, remains unaffected. For λ_0 above the quoted $\lambda_{0,c}$ learning obtains, but slower with increasing λ. See also Fig. 4.

$\sqrt{Q_0}$	10		100		1000	
η	0.28	0.6	0.28	0.6	0.28	0.6
$\lambda_{0,c}$ $(m = 1)$	3.7(1)	4.5(1)	3.95(5)	5.1(1)	6.1(1)	9.3(1)
$\lambda_{0,c}$ $(m = 4)$	3.9(1)	0.17(1)	6.1(1)	0.19(1)	14.1(1)	0.5(1)

Table 2: *Critical value of λ_0 for $L = 10$, $\rho = 1$, various $\sqrt{Q_0}$ and data parameters m, η. Here $\lambda = \lambda_0/\sqrt{\alpha}$ tuning is used.*

$\sqrt{Q_0}$	10		100		1000	
η	0.28	0.6	0.28	0.6	0.28	0.6
$\lambda_{0,c}$ (**1**)	0.45(1)	0.45(1)	0.31(1)	0.31(1)	0.19(1)	0.23(1)
$\lambda_{0,c}$ (**2**)	0.41(1)	0.41(1)	0.31(5)	0.31(1)	0.19(1)	0.23(1)

Table 3: *Critical value of λ_0 for $L = 10$, $\rho = 1$, $m = 1$, various $\sqrt{Q_0}$ and overlap parameter η. Here $\lambda = \lambda_0\, e(\alpha; q_0)$ tuning is used, with $q_0 = 1$ (**1**) and with $q_0 = \alpha N/L - 10000$ (**2**) – i.e., $e(\alpha)$ is e_q averaged from the beginning or averaged over the last 10000 iteration, respectively.*

4 Summary and Discussion

In the present paper we have investigated the performance of the AR-Hebb-algorithm [5] in the case where the input patterns are structured. The pattern statistics is characterized by the anisotropy vector $m\boldsymbol{C}$ and performance of the learning rule depends on m and on the overlap η between the anisotropy vector and the vector $\boldsymbol{B}$ that defines the rule – apart from the parameters λ, L and ρ which characterize the AR-Hebb-algorithm. We have seen that varying arbitrarily the projection of the data vector onto the teacher vector, as well as varying the strength of the data anisotropy itself does not change the general result: in the first case only slight changes in the threshold and convergence parameters are found, in the second case additional fixed points may appear, complicating the preasymptotic behaviour, however not in an essential way. Since this kind of data structure provides a quite extreme learning problem we do not expect more complex patterns in the data structure to produce new learning behaviour. A systematic test of this conjecture would have gone, however, beyond the scope of this paper.

As for usual Hebb learning, a rescaling of learning parameters is *required* to achieve good generalization for the classification of structured patterns. In this way the algorithm is tuned to approach the perceptron algorithm in the limit $e_q \ll 1$. Notice, however, that the thresholds for λ found even in the asymptotic behaviour for $L > 1$ mean that even in this limit the algorithm remains different from the perceptron one – in contradistinction to the $L = 1$ case.

Given L and ρ the only free parameter of the algorithm is λ, and tuning of λ may proceed in various ways. For instance, one may scale λ either with α, i. e., with the number of input-output pairs presented, or with the self-overlap $\mathcal{Q}$, or with the empirical error-rate e_q.

Our analysis of the asymptotics reveals that the scaling $\lambda \sim \alpha^{-1/2}$, which according to that analysis is equivalent to the scalings $\lambda \sim \mathcal{Q}^{-1/2}$, or $\lambda \sim e(\alpha)$, leads to asymptotically perfect generalization. The behaviour is robust in the sense that the prefactor λ_0 may be varied over a wide range without changing the asymptotic scaling of the generalization error. The tuning (annealing) required to obtain a working algorithm is not fine-tuning. The only requirement for obtaining good generalization is that λ_0 in (27) exceeds a certain minimum value, $\lambda_{0,c}^{asympt.}$. This behaviour is reminiscent of the fact that a minimum value of λ was also required in the case of unstructured data. In that case, however, the reason was entirely related to pre-asymptotic behaviour related to the fixed-point structure of the flow equations, whereas the above analysis pertains to the asymptotic domain.

In what concerns the pre-asymptotic behaviour, we see from the numerical solution of the full flow-equations and from simulations some empirical evidence that a non-trivial fixed point structure governing this behaviour is present also in the case studied here, in analogy to what has been found in [5]. As the present dynamical problem is *three*-dimensional instead of two-dimensional, however, the consequences of this might be suspected to be less severe. For instance, a fixed-point with stable and unstable directions in three dimensions does not necessarily produce a separatrix as in the two-dimensional case. However, the projection onto the ϵ_g-$\sqrt{\mathcal{Q}}$ plane shows a separatrix and hence a $\lambda_{0,c}$, as in the unstructured data case (see Fig. 4), with $\lambda_{0,c} > \lambda_{0,c}^{asympt.}$. This, and the necessity to scale down λ, seems to be a rather general feature.

The detail of the λ tuning does not seem important: essentially λ should not be too small during the preasymptotics (it may even stay constant) and should decrease in the asymptotics as α^{-r} with $0 < r \leq 1/2$ (but with a prefactor bounded from below). Any law which ensures these conditions appears to work, in particular $\lambda = \lambda_0/\sqrt{\alpha}$ or $\lambda = \lambda_0\, e(\alpha; q_0)$ with λ_0 above the thresholds given in the tables 2,3. This holds, of course, also for the unstructured data case, where however the scaling down of λ modifies the asymptotic law leading to faster convergence. The exceptional behaviour observed for $\eta = 0$ illustrates this fact, since in the student-teacher scenario $\eta = 0$ is equivalent to the unstructured case. Finally we find again that the algorithm is stable against noise or a further dilution of the information introduced by taking $\rho < 1$ – see Fig. 2.

As shown in our analysis the concept of learning introduced in section 1 leads to simple and robust procedures for the problem of learning under the rather realistic condition of *unspecific reinforcement*. This is a problem whose solution may be of vital importance in typical "life" situations, whether natural or artificial, and it is important to find out whether natural mechanisms can develop to tackle it. Our model may have this capacity, since it involves two, rather natural, steps: The *blind association* which means in fact "crediting" its own best choice for an action, and the *unspecific reinforcement*, which means taking into account the lesson from the environment. The implementation for perceptron has allowed systematic statistical and analytical results, showing that a good learning behaviour obtains in this way. Learning turns out to be very stable against variations in the parameters, but requires a minimal amount of *blind association* simultaneously with a scaling down of that parameter, e. g. proportionally with the observed error, or inverse proportionally to the square root of the number of iterations. A heuristic argument as to why $\lambda = 0$ does not work for unspecific reinforcement was suggested in our earlier paper. It is roughly as follows: since for $L = 1$ e_q can only be 0 or 1 $\lambda = 0$ means penalty for failure, no change for success, which is the usual perceptron learning rule known to converge. However, for $L > 1$ e_q can take fractional values in the interval $[0, 1]$. In this case $\lambda = 0$ means penalty for all answers which are short of perfect, i.e. even if the pupil is successful in far above 50% of the cases. This implies that the student does not really "learn" but only is confirmed when he already has learned perfectly; all partial failures are treated as complete failure. For $L = 1$ partial failure *is* complete failure, therefore the usual perceptron algorithm provides a specific punishment. For $L > 1$ this is no longer true and we need the blind association to indirectly provide an element of specificity in the punishment. The mathematical realization of this condition is represented by the fixed point structure and the asymptotic behaviour.

Although a full description of the parameter space was not possible, we could show that the qualitative features discussed above hold over a wide range of generic conditions. Away from the thermodynamic limit, i.e. for non-infinitesimal L/N, and even more so for the "realistic" models in [4], [7] it is significant that the blind association cannot be understood as a simple renormalization of the unspecific reinforcement. As can be seen from eqs. (10), (11), the two steps are of different character, since the error e_q which enters the reinforcement step depends on all $\boldsymbol{J}^{(q,l)}$ which are updated at each l-blind association step. The simulations show, however, in all these cases a learning behaviour not unsimilar to that found in the coarse grained limit, in particular the peculiar features of the possiblity of perfect generalization in spite of the unspecific reinforcement, and the necessity of the blind association. This, together with the stability to noise, suggests that the properties observed here may really be of a very general nature.

Acknowledgments: This project was initiated during the Seminar 'Statistical Physics of Neural Networks' in Dresden, March 1999. The authors would like to thank the Max Planck Institut für Physik Komplexer Systeme in Dresden for hospitality and financial support and the participants to the workshop for interesting discussions.

5 Appendix

5.1 Expectations values

The expectations values $A_{\cdot,\cdot}, S_{\cdot}$ in (14)-(16) are:

$$A_{JT} = m\,\eta\,\varphi\left(m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\right) + \sqrt{\frac{2}{\pi}}\,\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}}\,\mathrm{e}^{-\frac{m^2\mathcal{D}^2}{2\mathcal{Q}}} \tag{36}$$

$$A_{TT} = m\,\eta\,\varphi(m\,\eta) + \sqrt{\frac{2}{\pi}}\,\mathrm{e}^{-\frac{m^2\eta^2}{2}} \tag{37}$$

$$S_{JT} = 1 + \varphi\left(m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\right) - \varphi(m\,\eta) - 4\,G\left(\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}}, \frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}, \eta\right) \tag{38}$$

$$A_{JJ} = m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\,\varphi\left(m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\right) + \sqrt{\frac{2}{\pi}}\,\mathrm{e}^{-\frac{m^2\mathcal{D}^2}{2\mathcal{Q}}} \tag{39}$$

$$A_{TJ} = m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\,\varphi(m\,\eta) + \sqrt{\frac{2}{\pi}}\,\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}}\,\mathrm{e}^{-\frac{m^2\eta^2}{2}} \tag{40}$$

$$A_{JC} = m\,\varphi\left(m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\right) + \sqrt{\frac{2}{\pi}}\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}\,\mathrm{e}^{-\frac{m^2\mathcal{D}^2}{2\mathcal{Q}}} \tag{41}$$

$$A_{TC} = m\,\varphi(m\,\eta) + \sqrt{\frac{2}{\pi}}\,\eta\,\mathrm{e}^{-\frac{m^2\eta^2}{2}} \tag{42}$$

where

$$G\left(\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}}, \frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}, \eta\right) = \frac{1}{2}\int_{-\infty}^{m\,\frac{\mathcal{D}}{\sqrt{\mathcal{Q}}}} \frac{dt}{\sqrt{2\pi}}\,\mathrm{e}^{-\frac{1}{2}t^2}\left(1 + \varphi\left(\frac{t\,\frac{\mathcal{R}}{\sqrt{\mathcal{Q}}} - m\,\eta}{\sqrt{1 - \frac{\mathcal{R}^2}{\mathcal{Q}}}}\right)\right) \tag{43}$$

and $\varphi(x) = \mathrm{erf}(x/\sqrt{2})$, with erf the error function.

5.2 Asymptotic coefficients

In terms of the parametrization introduced in (23)-(26) and the combinations

$$u = \frac{1}{2}\,m\cos(z)\,\sqrt{2}\ , \tag{44}$$

$$v = m\sin(\frac{1}{2}\,z)\,\sqrt{2}\ , \tag{45}$$

$$g(v\omega) = v\,\omega\,\mathrm{erf}(v\,\omega)\ , \tag{46}$$

$$f(v\omega) = v\,\omega\,\mathrm{erf}(v\,\omega) + \frac{e^{-v^2\,\omega^2}}{\sqrt{\pi}}\ , \tag{47}$$

the Maple expressions for the coefficients $A_\gamma, B_\gamma, C_\gamma$ in (31)-(33) are given as:

$$A_{11} = -4\,\frac{\sin(\frac{1}{2}\,z)\,\mathrm{erf}(u)\,\omega}{\pi} \tag{48}$$

$$A_{12} = \frac{e^{-u^2} f(v\,\omega)}{\pi\sqrt{\pi}\,L} \tag{49}$$

$$A_2 = \frac{e^{-u^2}\,(-2\,\frac{1+2\,v^2\,\omega^2}{L} + 4\,(1-\frac{1}{L})\,\mathrm{erf}(u)\,f(v\,\omega))}{\sqrt{2\pi}} \tag{50}$$

$$B_{11} = 4\,\frac{m\sin(\frac{1}{2}\,z)^2\,\mathrm{erf}(u)\,(\omega^2-1+s^2)}{\pi} \tag{51}$$

$$B_{12} = -\frac{\sin(\frac{1}{2}\,z)\,e^{-u^2}\,\omega\,f(v\,\omega)}{\pi^{3/2}\,L} \tag{52}$$

$$B_2 = 4\,\frac{m\sin(\frac{1}{2}\,z)^2\,e^{-u^2}\,(\omega^2-1+s^2)\,(\frac{v\,\omega}{L} - (1-\frac{1}{L})\,\mathrm{erf}(u)\,f(v\,\omega))}{\sqrt{\pi}} \tag{53}$$

$$C_0 = \sqrt{2}\,u\,\mathrm{erf}(u) + \frac{\sqrt{2}\,e^{-u^2}}{\sqrt{\pi}} \tag{54}$$

$$C_1 = -(1-\frac{1}{L})\,e^{-u^2}\,f(v\,\omega)\,(\sqrt{\pi}\,m\,\mathrm{erf}(u)\cos(z) + \sqrt{2}\,e^{-u^2}) \tag{55}$$

References

[1] Hertz, J., Krogh, A. and Palmer, R.G. (1991), *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading, Mass.).

[2] Watkins, C.J.C.H. (1989), "Learning from delayed rewards", Ph.D.Thesis, unpublished.

[3] Sutton, R.S. (1988), Machine Learning **3**, 9.

[4] Mlodinow, L. and Stamatescu, I.-O. (1985), Int. J. of Comp. and Inform. Sci, **14**, 201.

[5] Kühn, R. and Stamatescu, I.-O. (1999), J. Phys. A: Math. Gen. **32**, 5479.

[6] Riegler, P., Biehl, M., Solla S.A. and Marangi, C (1996) in Proceedings of the 7th Italian Workshop on Neural Networks 1995, eds. Marinaro, M. and Tagliaferr, R., p. 87, World Scientific (Singapore).

[7] Stamatescu, I.-O. (1998), "Statistical Features in Learning", contribution to LEARNING'98, Madrid, cond-mat/9809135.

[8] Biehl, M. and Riegler, P. (1994), Europhys. Lett. **28**, 525.

[9] Vallet, F. (1989), Europhys. Lett. **8**, 747.

[10] Kinouchi, O. and Caticha, N. (1992), J. Phys. A **25**, 6243.

[11] Biehl, M. and Schwarze, H. (1992), Europhys. Lett. **20**, 733.