Fachbereich Informatik der Universität Hamburg

Vogt-Kölln-Str. 30 · D–22527 Hamburg · Germany

University of Hamburg — Computer Science Department

Bericht Nr. 173 · Report No. 173

Local stability learning rules and phase space gardening in neural networks

Norman Hendrich

FBI–HH–B–173/95

Februar 1995

In die Reihe der Berichte des
Fachbereichs Informatik
aufgenommen durch

Accepted for Publication in the
Report Series of the Department
of Computer Science by

Prof. Dr. K. von der Heide and Prof. Dr. K. Lagemann

Abstract

This report studies the dynamical properties of Hopfield-Gardner neural networks, with the object of phase-space gardening.

It introduces the new concept of *local stability* learning rules that specify the stability of each pattern at each neuron individually.

Simulations of the networks show that these learning rules allow to adjust the size and shape the basins of attraction of the stored patterns as desired.

Zusammenfassung

Dieser Bericht untersucht die Möglichkeiten, die dynamischen Eigenschaften von neuronalen Netzwerken des Hopfield-Gardner Typs gezielt zu modifizieren — das sogenannte 'Phase-Space Gardening'. Dazu wird das neue Konzept von *local stability* Lernregeln vorgestellt. Idee dieser Lernregeln ist es, für jedes Neuron und zu lernende Muster die Stabilität einzeln vorzugeben.

Umfangreiche Simulationen zeigen, daß durch die Verwendung dieser Lernregeln tatsächlich die Größe und Gestalt der Einzugsbereiche der Muster in weiten Grenzen frei eingestellt werden können.

Contents

1 Introduction 1

1.1 The Hopfield-Gardner neural network model . . . 2

1.2 Storage capacity and stabilities . . . 3

1.3 Basins of attraction analysis . . . 4

1.4 Scope of this work . . . 6

1.4.1 Local stability learning rules . . . 7

1.4.2 Size of the basins of attraction . . . 7

1.4.3 Shape of the basins of attraction: Phase Space Gardening . . . 8

2 Local stability learning rules 9

2.1 Iterative learning rules . . . 9

2.2 The distribution of stabilites . . . 10

2.3 Local stability learning rules . . . 10

2.4 Local stability learning in the binary couplings network . . . 11

3 Simulation Method 15

3.1 Basic simulation algorithm . . . 15

3.2 Typical simulation data $m_f(m_0)$ . . . 15

3.3 Estimation of the basins of attraction . . . 16

3.4 Random pattern generation . . . 17

3.5 Parallel dynamics . . . 19

4 Phase-space gardening: Size of the basins of attraction 21

4.1 Piece by piece constant distribution of stabilities . . . 21

4.2 Linear distribution of stabilities, spherical model . . . 27

4.3 Linear distribution of stabilities: binary couplings model . . . 30

5 Phase-space gardening: Shaping the basins of attraction 33

5.1 The LR model . . . 33

5.2 The LR simulation strategy . . . 33

5.3 A first LR implementation . . . 37

5.4 Basins of attraction in the LR model . . . 38

6 Summary 44

References 45

List of Figures

1 Storage capacity $\alpha(\kappa)$ as a function of minimum stability in the spherical model . 4
2 Example memory pattern and three test patterns 6
3 Phase-space view of the basins of attraction 8
4 Minover PSG learning algorithm 11
5 Energy minimization learning algorithm [Koehler *et. al.* 89] 13
6 Energy minimization PSG learning algorithm [Hendrich 93] 13
7 Distribution of stabilities $\rho(\kappa)$ in the binary couplings network. 14
8 Basic simulation algorithm 16
9 Mean final overlap m_f vs. initial overlap m_0, finite-size scaling 17
10 Example for the fit of the model (equation 15) with $m_f(m_0)$ data 18
11 Random test pattern generation 19
12 Fast parallel dynamics 20
13 Piece by piece constant and linear distribution of desired stabilities $\rho(\kappa)$ 21
14 Piece by piece constant $\rho(\kappa)$, binary couplings network, $\alpha = 0.10$ 24
15 Piece by piece constant $\rho(\kappa)$, binary couplings network, $\alpha = 0.15$ 25
16 $m_c(\kappa)$ in the binary couplings network, $\alpha = 0.15$, piece by piece constant $\rho(\kappa)$. 26
17 $m_c(\kappa)$ in the binary couplings network, $\alpha = 0.25$, piece by piece constant $\rho(\kappa)$. 26
18 $m_c(\mu)$ and $\kappa(\mu)$, linear distribution of stabilites, spherical model, $\alpha = 0.3$ 28
19 $m_c(\kappa)$ in the spherical model, linear $\rho(\kappa)$ 29
20 $m_c(\mu)$ and $\kappa(\mu)$, linear distribution of stabilites, binary couplings model, $\alpha = 0.15$ 31
21 $m_c(\kappa)$ in the binary couplings network, linear $\rho(\kappa)$ 32
22 Desired Distribution of stabilities in the LR model 34
23 Example pattern and derived LR test patterns 35
24 Estimation of left and right critical initial overlap $m_{c,L}$, $m_{c,R}$ 36
25 Left and right critical initial overlap, $m_{c,L}$ and $m_{c,R}$, and difference Δm_c 36
26 Basins of attraction $m_c(\mu)$ in the LR model, naive learning rule, $\alpha = 0.15$ . . . 37
27 Global distribution of stabilities $\rho(\kappa)$ and examples of the local pattern stabilites $\kappa^\mu(i)$ in the LR model 39
28 Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.15$ $(N = 600)$ 40
29 Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.15$, finite-size scaling 41
30 Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.20$ $(N = 600)$ 42
31 Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.25$ $(N = 600)$ 43

1 Introduction

Hopfield-Gardner neural networks are both simple and very powerful models of associative memory. Because of their analogy to Ising-spin systems, many properties of the networks can be studied using the methods of statistical physics. Probably most important is the replica calculation of the maximum storage capacity by [Gardner 87]. However, the dynamics of the networks is very complex and still out of reach of the theoretical methods.

This report focuses on a study of the basins of attraction in Hopfield-Gardner networks. While previous studies used networks near saturation, where all patterns have basins of attraction of approximately the same size, the object of this report are techniques to adjust size and shape of the basins of attraction for each pattern—techniques also called *phase space gardening* for obvious reasons.

To this end, a new class of *local stability learning rules* is presented. These learning rules allow to set the desired stability of a pattern at each neuron. Extensive simulations of the resulting networks show that the local stability learning rules allow to set the size and shape of the basins of attraction of the memory patterns.

Main contributions of this report are as follows:

- Section 2 introduces the concept of *local stability learning rules*. Two local stability learning rules are presented in detail:

  A local stability learning rule for the integer couplings network (spherical model) is derived as a generalization of the Minover algorithm. This algorithm allows to reach the optimal stabilities for given patterns.

  Also, a new learning algorithm for the binary couplings network is presented. The algorithm combines a greedy optimization (energy-minimization) strategy with the Minover cost-function and can be adapted for local stability learning. It reaches a storage capacity of $\alpha \approx 0.4$ in the binary couplings network.

- The simulation strategy used in this report, based upon [Forrest 88], is explained in section 3.

  A new three parameter scaling law for the $m_f(m_0)$ mean final overlap as a function of initial overlap data is presented. It provides very accurate fits to the $m_f(m_0)$ data and allows to derive the size of the basin of attraction m_c for a given pattern.

- In section 4 the local stability learning rules are used to set the size of the basins of attraction.

  Simulations of both the integer and the binary couplings networks show that the size of the basin of attraction of a pattern correlates with the pattern stability. In the same network, some patterns may have very large basins of attraction, while other patterns have small basins of attraction.

- An attempt to control the shape of the basins of attraction is presented in section 5. A left/right (LR) anisotropy in the stability distributions is used during learning. The simulations show that this simple strategy suffices to install a LR anisotropy in the basins of attraction.

Put together, the results from section 4 and section 5 prove that the local stability learning rules allow to set both the size and the shape of the basins of attraction—they allow full phase space gardening. To the best of our knowledge, this report is the most complete study of the dynamics of Hopfield-Gardner networks so far and it is the first work to study in detail the dependence of the basins of attraction on pattern stabilities.

This introduction presents the basic definitions and terminology used in the later sections. First, section 1.1 describes the original Hopfield memory neural network with Hebb learning rule and explains autoassociative storage. Then the Gardner calculation of the maximum storage capacity is presented. Third, the problems encountered in the theoretical understanding of the dynamics of the networks are sketched in section 1.3.

Finally, section 1.4 summarizes the scope of this work, namely the analysis of local stability learning rules and their use to adjust the size and shape of the basins of attraction of the stored memory patterns.

1.1 The Hopfield-Gardner neural network model

Since the 1982 seminal paper [Hopfield 82], spin-glass neural networks have attracted considerable attention. The networks show collective computational properties which make them a paradigm for fault-tolerant massive-parallel computation.

A Hopfield network consists of a highly interconneted system of N Ising spins (called neurons) $S_i = \pm 1$ with couplings J_{ij} from neuron j to neuron i.

The basic idea is to store $P = \alpha \cdot N$ binary patterns $\xi_i^\mu = \pm 1$, $\mu = 1 \ldots P$, $i = 1 \ldots N$ as the fixed points of the dynamics of the network.

To compute the state S_i at time step $t + 1$ both a parallel dynamics

$$S_i(t+1) = \mathrm{sgn}\Big(||J_{ij}||^{-1} \sum_{j \neq i} J_{ij} S_j(t) \Big) = \mathrm{sgn}\Big(h_i(t) \Big), \tag{1}$$

or serial dynamics (one spin update after another) can be used. A pattern is stored as the fixed point of this dynamics if $\xi_i^\mu = \mathrm{sgn}(||J_{ij}||^{-1} \sum_{j \neq i} J_{ij} \xi_j^\mu)$. The patterns are then memorized by the network in a content-addressable (auto-associative) way: starting from an initial state which partially resembles one of the patterns, the system will rapidly evolve into the nearest attractor—the pattern.

Hopfield proposed to set the synaptic couplings according to the Hebb learning rule [Hopfield 82],

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu . \tag{2}$$

With the Hebb-rule, the local field $h_{i\mu}$ at neuron S_i for pattern ξ_i^μ becomes

$$h_{i\mu} = \frac{1}{N} \sum_{j \neq i} \sum_{\nu=1}^{P} \xi_i^\nu \xi_j^\nu \xi_j^\mu = \frac{1}{N} \Big((N-1)\xi_i^\mu + \sum_{j \neq i} \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu \Big). \tag{3}$$

The first term stabilizes the pattern, because $\xi_i^\mu = \mathrm{sgn}(\xi_i^\mu)$. For uncorrelated patterns (that is, $\sum_j \xi_i^\mu \xi_j^\mu = 0$), the sum $\sum_{j \neq i} \sum_{\nu \neq \mu} \xi_i^\nu \xi_j^\nu \xi_j^\mu$ consists of $P \cdot N$ uncorrelated terms $\pm 1/N$, with mean 0 and variance $\sqrt{P/N}$. Therefore, the patterns will be fixed points of the dynamics as long as $\alpha = P/N < 1$.

From simulations Hopfield found that the patterns are fixed points and attractors of the dynamics as long as $\alpha < 0.15$.

Many aspects of the Hopfield networks are quite well understood by now. For example, [Amit *et. al.* 85] and [Amit *et. al.* 87] were able to calculate the mean-field theory of the Hopfield model for both finite and infinite number of patterns in the replica-approximation.

1.2 Storage capacity and stabilities

Shortly after the publication of the Hopfield model, the search for learning rules with better storage capacity α_c than the Hebb rule started. Given P patterns ξ_i^μ, is there a choice of the J_{ij} so that the patterns are stored?

In 1987, [Gardner 87] succeeded to calculate the optimal storage capacity of attractor networks, applying the replica method to the space of interactions J_{ij}. To ensure finite basins of attractions around the patterns, a positive 'stability' κ is introduced and the synaptic couplings J_{ij} are chosen to fulfil the constraints

$$\kappa_{i\mu} = h_{i\mu} \cdot \xi_i^\mu \geq \kappa > 0. \tag{4}$$

Obviously, all patterns are stored correctly, if (the symbol $\Theta(x)$ denotes the Heavyside function)

$$\prod_{\mu,i} \Theta(\, h_{i\mu} \cdot \xi_i^\mu - \kappa\,) = 1. \tag{5}$$

Gardner was able to calculate the volume of the subspace of all couplings J_{ij} that fulfil the above condition as a function of the stability κ. With the spherical norm $||J_{ij}|| = \sum_j J_{ij}^2 = N$, one has

$$V_T = \frac{\int \prod_{i\neq j} dJ_{ij} \prod_{\mu,i} \Theta(\, h_{i\mu} \cdot \xi_i^\mu - \kappa\,)\, \delta(\, \sum_j J_{ij}^2 - N\,)}{\int \prod_{i\neq j} dJ_{ij} \prod_i \delta(\, \sum_j J_{ij}^2 - N\,)}. \tag{6}$$

If this subspace V_T shrinks to a single point, the network has reached its optimal storage capacity. The very complicated calculation gives (in the replica-symmetric approximation) the storage capacity α as a function the of minimum stability κ as

$$\alpha_c(\kappa) = \left[\frac{1}{\sqrt{2\pi}} \int_{-\kappa}^{\infty} dt\ e^{-t^2/2}\, (t+\kappa)^2 \right]^{-1}, \tag{7}$$

so that $\alpha_c(0) = 2$ [Gardner 87]. This function is sketched in figure 1.

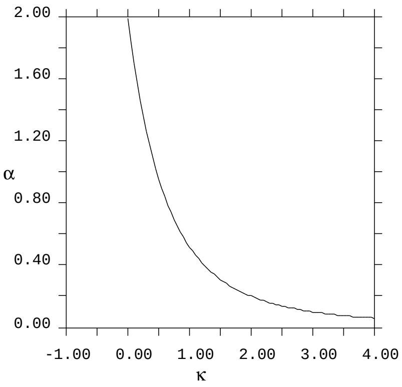


Figure 1: Storage capacity $\alpha(\kappa)$ as a function of minimum stability in the spherical model

A similar calculation is also possible if the couplings are restricted to binary values or integer values with a finite range [Gardner & Derrida 88], [Gutfreund & Stein 90]. The effects of replica breaking, however, are very large. The current best estimation for the optimal storage capacity of the binary couplings model is $\alpha_{c,B} = 0.83$, in excellent agreement with the zero-entropy calculation and simulations [Krauth & Opper 89], [Krauth & Mézard 89].

1.3 Basins of attraction analysis

Despite several efforts, no simple analytical model exists for the description of the dynamical properties of the Hopfield-Gardner networks [Krauth *et. al.* 88], [Kepler & Abbott 88], [Opper *et. al.* 89], [Nardulli & Pasquariello 90]. Some of the difficulties encountered are obvious:

- The Hamiltonian (the matrix of couplings J_{ij}) is known explicitly only for the simple case of the Hopfield model, but not for the better networks with iterative learning.
- Even in the Hopfield model, the calculation of the first-time overlap m_1 of the network state with a pattern after one step of the dynamics is extremely complicated [Bruce *et. al.* 87]. The calculation of m_2, m_3, etc. is (almost) hopeless.
- Several dynamical rules are plausible and have to be studied. For example, simulations show that a dynamical rule with memory-terms [Kanter & Sompolinsky 87] may lead to better recall than the simple parallel or serial dynamics.
- The energy landscape of the networks is very complicated. Spurious attractors, e. g. resulting from a mixture of patterns, were first observed in the Hopfield model and may completely dominate the dynamics. For example, above $\alpha = 0.051$ the spurious states are the global energy minima in the Hopfield model.

As shown in the previous section, the analytical models allow to calculate whether the patterns ξ^μ are stored as fixpoints in the networks. The main objective of an analysis of the dynamics of the networks is a model for the *basins of attraction* of the patterns. That is, what input patterns will be recalled by the network under its dynamics?

Starting from a test pattern ξ' with initial overlap $m_0(\xi', \xi^\mu) = 1/N \sum_i \xi'_i \xi^\mu_i$ with a memory pattern ξ^μ, first the pattern is loaded into the network, so that $S_i(0) = \xi'_i$. Then the dynamics of the network is iterated, to give $S_i(1) = \mathrm{sgn}(h_i(0))$, then $S_i(2) = \mathrm{sgn}(h_i(1))$, $\ldots$ After some steps of the dynamics, the network may reach a fixed point, $S_i(t_f) = S_i(t_f - 1)$. The test pattern is recalled correctly, if $S_i(t_f) = \xi^\mu_i$. However, the network may recall another memory pattern, $S_i(t_f) = \xi^\nu_i$, or evolve into a spurious attractor or cycle.

The *basin of attraction* of memory pattern ξ^μ is the set of all input patterns ξ' which fulfil $\{\xi' \mid S(0) = \xi' \Rightarrow S(t_f) = \xi^\mu\}$. For example, if all test patterns with initial overlap $m_0 > m_c$ with a given pattern ξ^μ are recalled correctly by the network, but test patterns with overlap $m_0 < m_c$ are not recalled, then m_c is the size of the basin of attraction of pattern ξ^μ.

A simple estimation is possible, if the patterns are stored with stability κ, so that $\xi^\mu_i h_{i\mu} > \kappa$. The network will then recognize all test patterns that differ in less than $O(\kappa N^{-1/2})$ positions from pattern ξ^μ in one time step. However, simulations show that the basins of attraction around the memory patterns are much larger, of order $O(1)$.

A typical example of a (very small) memory pattern and three input patterns with different overlap with this memory pattern is shown in figure 2. The pattern ξ (figure 2a) consists of 144 bit positions, with (almost) equal probability of -1 (white) and $+1$ (black). The patterns shown in figure 2b, 2c, and 2d represent patterns ξ' with overlap $m_0 = 0.9$, 0.8 and 0.5.

The size of the basins of attraction in Hopfield-Gardner networks may be much larger than $m_0 = 0.5$. A network could therefore easily recognize even the third test pattern.

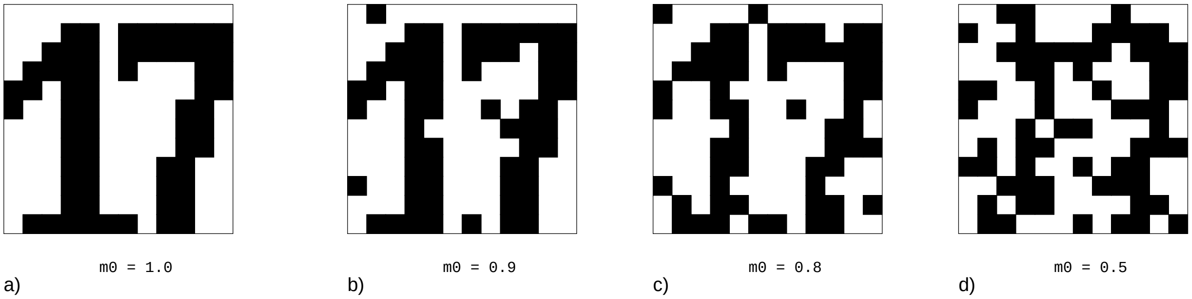


Figure 2: Example memory pattern and three test patterns

Only in special cases it has been possible to derive analytical models for the dynamics of the networks, e. g. in sparse connected Hopfield networks with less than $O(\ln N)$ couplings per neuron [Derrida *et. al.* 87]. The calculation is also possible in the saturated, sparse connected Gardner network, where the basins of attraction are found to be optimal below $\alpha \leq 0.41$—all test patterns with macroscopic overlap $m_0 > 1/\sqrt{N}$ are recalled correctly [Gardner 89a].

A model for the basins of attraction in fully connected, saturated networks was presented in [Kepler & Abbott 88]. From computer simulations they concluded that most patterns are perfectly recalled, if the overlap m_1 between network state and memory pattern ξ^μ after one step of the dynamics is larger than the initial overlap m_0 of the test pattern $\xi^{\mu,r}$. With an elaborate replica calculation [Kepler & Abbott 88] were able to calculate $m_1(m_0)$ and could predict the basins of attraction. However, the model is only approximate and valid only for saturated networks with $\kappa \approx \kappa_c(\alpha)$.

Therefore, so far most studies of the dynamics of the networks used computer simulations [Forrest 88], [Hendrich 91], [Koscielny-Bunde 92]. The simulation strategy used in this report is presented in section 3.

1.4 Scope of this work

Phase-space gardening—the capability to control the size and shape of the basins of attraction around the stored patterns—is very desirable for actual applications of large Hopfield networks as associative memories. This would also make Hopfield networks a much better model for associative recall in the human brain, where obviously some patterns are recalled much better than others.

As a means to phase-space gardening (PSG) this report introduces and studies the concept of *local stability* learning rules, both in the spherical model (unrestricted integer couplings) and the binary couplings model. Extensive simulations of the resulting networks study the size and shape of the basins of attraction around the stored patterns.

The results show that the size of the basins of attraction of a pattern ξ^μ can be accurately controlled by the value of the minimum stability of this pattern—in a wide range from tiny to very large basins of attraction. The shape of the basins of attraction, however, can be adjusted to a much lesser extent.

1.4.1 Local stability learning rules

The original Hopfield model with Hebb learning gives the same stabilities (a Gaussian distribution) for all patterns. On the other hand, iterative learning rules allow to select the minimum stabilities for a given pattern. Since the first simulations of iterative learned networks it has been conjectured that larger stabilities imply larger basins of attraction [Forrest 88].

However, most previous simulations of iterative learned networks used saturated networks with a global stability $\kappa = \kappa_{\max}(\alpha)$ only, and did therefore not study the effect of different pattern stabilities at given storage density α.

This report introduces the concept of *local stability* learning rules, where the desired minimum stability is not a global constant, but rather chosen individually for each pattern and neuron. That is, instead of one global stability, $h_{i\mu}\xi_i^\mu > \kappa$, the local stability learning rules use $P \cdot N$ local stabilities Λ_i^μ with $h_{i\mu}\xi_i^\mu > \Lambda_i^\mu$.

This generalization is possible for both the iterative learning rules used in the spherical model networks and the energy-minimization learning rules proposed for the binary-couplings network. In fact, the generalization of the iterative Minover [Krauth & Mézard 87] learning rule and it's speed-optimized variant [Abbott & Kepler 89a] are straightforward.

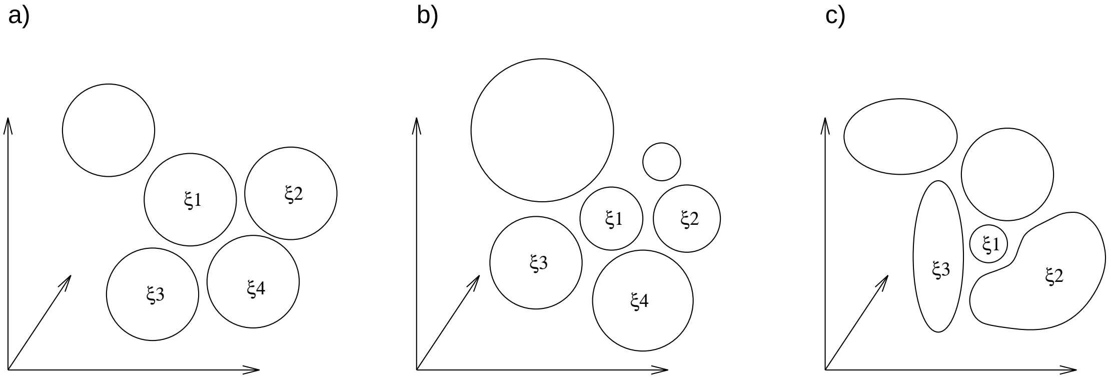
Several initial distributions of the stabilities and the resulting distributions after learning are studied in this paper.

1.4.2 Size of the basins of attraction

The first set of experiments with the local stability learning rules is presented in section 4. The basins of attraction are studied both in the spherical and the binary-couplings model for two typical choices for the distribution of stabilities.

To test whether larger stabilities actually give larger basins of attraction, a piece by piece constant interval function is chosen for the distribution of stabilities $\rho(\kappa)$. That is, the patterns are divided into groups and each group of patterns is assigned a stability Λ_i before learning. The simulations show that the size of the basins of attraction correlates well with the stabilities for all storage densities. In fact, it is possible to install very large basins of attraction for some of the patterns even at relatively high storage ratio, at expense of the size of the basins of attraction of most other patterns.

The second set of simulations presents networks with a linear distribution of stabilities, that is, the stability of pattern ξ^μ is given by $\Lambda_\mu = \mu/P \cdot \Lambda_{\max}$. The simulation data show that it is possible to set the size of the basin of attraction individually for each pattern.



a) All basins of attraction are spherical and of same size (Hopfield, saturated Gardner model) b) Basins of attraction of different size c) Anisotropic basins of attraction (full phase-space gardening)

Figure 3: Phase-space view of the basins of attraction

1.4.3 Shape of the basins of attraction: Phase Space Gardening

The simulations presented in section 5 try to install anisotropic basins of attraction around the patterns.

A simple model is introduced to generate a left/right (LR) asymmetry in the distribution of stabilities of the patterns. A pattern with a high stability Λ_h at the left half of neurons $i = 1 \dots N/2$ but low stability Λ_l at the right half of neurons $i = N/2 + 1 \dots N$ should be recalled better from test patterns with noise at the left half of neurons.

The simulations show that it is possible to shape the basins of attraction using this technique. However, because small stabilites imply small basins of attraction, the left/right asymmetry is not large for either patterns with very small or very large basins of attraction.

Some of the implications of the phase-space gardening will be presented in the summary.

2 Local stability learning rules

This section presents an overview of learning rules for the Hopfield-Gardner networks, based upon the Hebb rule as used in the original Hopfield model. The properties of some iterative learning rules, namely the Minover rule which allows to attain the optimal storage capacity of the Gardner model, are summarized in section 2.1. The distributions of stabilities resulting from Hebb and Minover learning are presented in section 2.2.

Section 2.3 introduces the new concept of *local stability learning rules*, the main contribution of this report. A local stability learning rule for the spherical model is presented as a generalization of the Minover rule. Finally, a local stability learning rule for the binary couplings network based on the energy-minimization rule is given in section 2.4.

2.1 Iterative learning rules

Shortly after the presentation of the original Hopfield model with Hebb learning, several groups proposed iterative learning rules to improve the storage capacity and the recall properties of the networks. All of these learning rules are based upon the iteration of Hebb learning steps, until no further improvements are possible [Diederich & Opper 87].

For the purpose of this report, the Minover algorithm [Krauth & Mézard 87] is of special interest. Given a network with N neurons and a set of P patterns ξ^μ the Minover learning rule iteratively modifies the couplings J_{ij} to give the solution with optimal minimum stability κ. The idea of the algorithm is an iteration of Hebb learning steps. To set the new value of synapse J_{ij}, the patterns are sorted by their stabilities, and the pattern ξ^μ with the lowest stability so far is used for the Hebb learning,

$$J_{ij} \rightarrow J_{ij} + N^{-1} \cdot \Theta(\kappa_{i\mu} - \kappa) \cdot \xi_i^\mu \xi_j^\mu. \tag{8}$$

This is iterated for all neurons and couplings, until all stabilities $\kappa_{i\mu} > \kappa$. A pseudocode description of the Minover algorithm, already modified for local stability learning, is given in figure 4. The original Minover algorithm uses the global stability κ instead of the local stabilities Λ_i^μ.

To improve the convergence, [Abbott & Kepler 89a] [Abbott 90] proposed to scale the individual Hebb steps with the norm $||J_{ij}||$ of the coupling matrix. The optimal learning speed (minimum number of iterations) is reached for

$$J_{ij} \rightarrow J_{ij} + N^{-1} \cdot \Theta(\kappa_{i\mu} - \kappa) \cdot 2(\kappa + \delta - \kappa_{i\mu})||J_{ij}|| \cdot \xi_i^\mu \xi_j^\mu. \tag{9}$$

2.2 The distribution of stabilites

One criterion for the classification of attractor neural networks is the distribution of stabilities $\rho(\Lambda)$ after learning, where $\rho(\Lambda)d\Lambda$ is the normalized fraction of stabilities Λ_i^μ in the interval $[\Lambda, \Lambda + d\Lambda]$ [Abbott & Kepler 89b].

In the Hopfield model a simple analysis of the local fields (see equation 3) gives

$$\rho_H(\Lambda) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\Lambda - \frac{1}{\sqrt{\alpha}}\right)^2\right]. \tag{10}$$

That is, a Gaussian distribution centered around $1/\sqrt{\alpha}$ and variance 1. Because the Gaussian distribution is not zero for $\Lambda < 0$, the patterns are not stored perfectly in the Hopfield model.

In saturated Gardner networks with parameters α, $\kappa_c(\alpha)$ the distribution of stabilities can be calculated as well. One has

$$\rho(\Lambda) = \frac{1}{\sqrt{2\pi}} \exp(-\Lambda^2/2)\Theta(\Lambda - \kappa_c) + \frac{1}{2}\Big(1 + \mathrm{erf}(\kappa_c/\sqrt{2})\Big)\delta(\Lambda - \kappa_c), \tag{11}$$

that is a δ peak at $\Lambda = \kappa_c$ and an exponential tail above $\Lambda \geq \kappa_c$.

2.3 Local stability learning rules

The learning rules as presented above try to install the same stabilities for all patterns, to give saturated networks. Simulations show that this results in networks where all patterns have similar and isotropic basins of attraction [Forrest 88].

Under the assumption that the size of the basin of attraction of a pattern correlates with the pattern stability, it is obvious how to modify the above iterative learning rules.

- If the basin of attraction of pattern ξ^μ should be isotropic but of given size, the pattern is learned until its stabilities $\kappa_{i\mu}$ are all above a preset value Λ_μ, independent of neuron index i. The value of Λ_μ will, however, depend on the storage ratio α and the stabilities chosen for the other patterns.

  For example, to enlarge the basin of attraction of pattern ξ^1 in an otherwise saturated network (N, P), the value $\Lambda_2 \approx \kappa_c(P/N) - \epsilon$ is chosen for patterns ξ^2 to ξ^P, but a larger value $\Lambda_1 > \kappa_c(P/N)$ is chosen for pattern ξ^1.

  Two sets of experiments with this type of learning rule are presented in section 4.

- If full phase-space gardening is desired, all individual stabilities $\kappa_{i\mu}$ of pattern ξ^μ are preset during learning.

  If the pattern ξ^μ should be recalled with large basin of attraction at some group of neurons $S_{i_1,\ldots,i_n}$, the respective desired stabilities $\kappa_{i_1,\mu} \ldots \kappa_{i_n,\mu}$ are set to large values.

  Simulations with this type of learning rule are presented in section 5.

The generalization of the original Minover algorithm into a local stability learning algorithm is straigtforward—the global parameter κ is split into individual values of Λ_i^μ for all patterns and neurons. A pseudocode description of the algorithm is given in figure 4.

```
Minover PSG learning( network $N$, patterns $\xi^\mu$, desired stabilities $\Lambda_i^\mu$):
initialize the couplings $J_{ij}$ to random or Hebb values
for all neurons $S_i$ do (parallel)
    repeat
        calculate all pattern stabilities $\kappa_{i\mu}$
        select the pattern $\xi^\nu$ with lowest stability $\kappa_{i\nu}$
        $J_{ij} \rightarrow J_{ij} + N^{-1} \cdot \Theta(\Lambda_i^\nu - \kappa_{i\nu}) \cdot \xi_i^\nu \xi_j^\nu$
    until (all $\kappa_{i\mu} > \Lambda_i^\mu$)
end for
calculate new norm $||J_{ij}||$
return $J_{ij}$
```

Figure 4: Minover PSG learning algorithm

2.4 Local stability learning in the binary couplings network

The Hopfield-Gardner network with clipped binary couplings $J_{ij} = \pm 1$ is of special theoretical and practical interest. First, the binary couplings network has finite information content in the limit $N \rightarrow \infty$. It stores $0.83 \cdot N^2$ bits using a coupling matrix of N^2 bits—whereas the standard integer-coupled networks have vanishing relative information content (less than $2 \cdot N^2$ bits stored in a coupling matrix of $N^2 \cdot \ln N$ bits). Second, the use of only binary couplings make this model especially attractive for electronic or optical implementation.

Unfortunately, learning is much harder in the binary couplings network. No learning rules (except exhaustive search) are known that allow to reach the critical storage ratio $\alpha_B = 0.83$.

The simple clipped Hebb rule, $J_{ij} = \mathrm{sgn}(\sum_\mu \xi_i^\mu \xi_j^\mu)$ gives a critical storage ratio below $\alpha \approx 0.1$, with large number of bit-errors in the patterns and small basins of attraction. Obviously, algorithms based on iteration of Hebb steps are not possible in the binary couplings network.

However, the learning can be formulated as an optimization problem and standard optimization algorithms may be used. For example, [Koehler *et. al.* 89] defined a cost-function as

$$E = \sum_{i,\mu} (\kappa_{i\mu} - \kappa)^2 \qquad (12)$$

and used a simple gradient descent method to minimize this cost-function by flipping the J_{ij}. A pseudocode description of the algorithm is shown in figure 5. It gives a critical storage ratio of $\alpha_{b,K} \approx 0.4$.

A variant of this algorithm similar to the Minover scheme was used by [Hendrich 92] in a simulation of fault-tolerance in the binary-couplings network. The idea is to use the cost-function

$$E = \sum_{i,\mu} \Theta(\kappa - \kappa_{i\mu}) \tag{13}$$

in combination with the gradient-descent. This algorithm, too, gives $\alpha_{b,H} \approx 0.4$ but is easier to compute and better suited for actual hardware implementations.

Examples of the distribution of stabilities after learning with this algorithm are shown in figure 7.

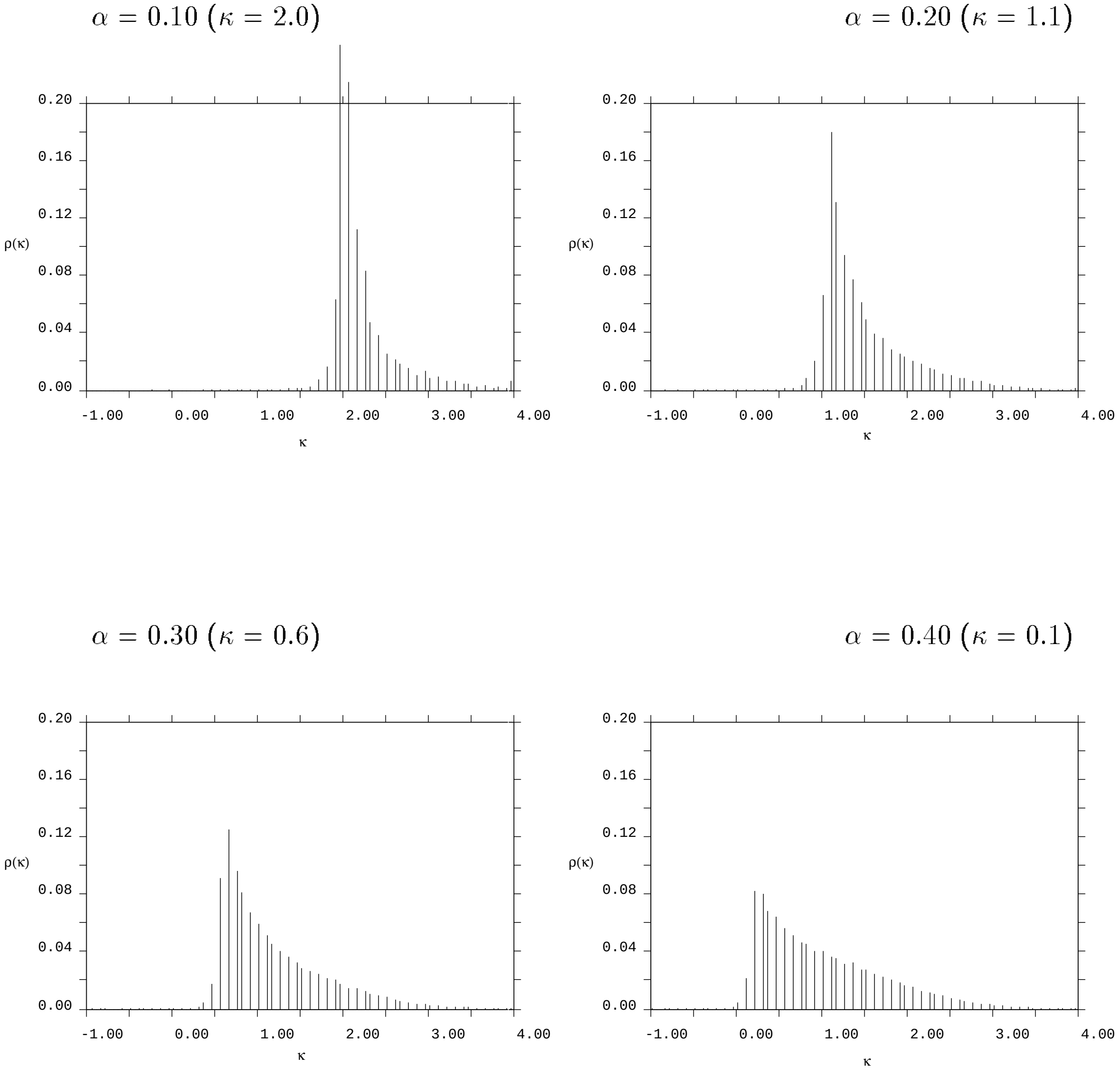
Both of the energy-minimization algorithms can be adapted for phase-space gardening. The idea, again, is to select the individual stabilities a pattern ξ^μ should have at neuron S_i after learning. The modifications to the algorithms are straightforward, see figure 6 for a pseudocode description.

```
Energy minimization learning( network $N$, patterns $\xi^\mu$, stability $\kappa$ ):
initialize the couplings $J_{ij}$ to random values
for all neurons $S_i$ do (parallel)
    calculate initial energy, $E = \sum_{j,\mu}(\kappa_{i\mu} - \kappa)^2$
    repeat
        select a random index $k$
        calculate $E_+ = E(J'_{ik} = J_{ik})$ and $E_- = E(J'_{ik} = -J_{ik})$
        if $(E_- < E_+)$ then set $J_{ik} = -J_{ik}$
    until (no further improvement in $E$)
end for
```

Figure 5: Energy minimization learning algorithm [Koehler *et. al.* 89]

```
Energy minimization PSG learning(
    network $N$, patterns $\xi^\mu$, desired stabilities $\Lambda_i^\mu$ ):
initialize the couplings $J_{ij}$ to random values
for all neurons $S_i$ do (parallel)
    calculate initial energy, $E = \sum_{j,\mu} \Theta(\Lambda_i^\mu - \kappa_{i\mu})$
    repeat
        select a random index $k$
        calculate $E_+ = E(J'_{ik} = J_{ik})$ and $E_- = E(J'_{ik} = -J_{ik})$
        if $(E_- < E_+)$ then set $J_{ik} = -J_{ik}$
    until (no further improvement in $E$)
end for
```

Figure 6: Energy minimization PSG learning algorithm [Hendrich 93]



Four examples of the distribution of stabilities $\rho(\kappa)$ in the binary couplings network after energy-minimization learning ($N = 512$). The diagrams show the fraction of pattern stabilities $\rho(\kappa)\Delta\kappa$ in the interval $[\kappa, \kappa + \Delta\kappa]$.

Figure 7: Distribution of stabilities $\rho(\kappa)$ in the binary couplings network.

3 Simulation Method

The results presented in this paper were obtained using the simulation strategy first introduced by [Forrest 88]. This section describes the simulation method in more detail.

First, the basic idea of the simulations is sketched in section 3.1. Then the typical simulation data, namely the mean final overlap m_f and the fraction of perfectly recalled patterns f_p, is presented. In section 3.3 a new scaling law for the $m_f(m_0)$ data is proposed that allows to determine the size of the basins of attraction. Finally, the algorithms for test pattern generation and parallel dynamics are discussed in section 3.4 and section 3.5.

3.1 Basic simulation algorithm

As discussed above, the analytical description of the complex dynamics of Hopfield-Gardner networks is still out of reach. Therefore, a study of the dynamics of the networks has to recur to computer simulation.

The basic simulation method used in this paper follows the ideas introduced in [Forrest 88]. Given the network size N and storage density α, the network is initialized, a set of random patterns $\xi_i^\mu = \pm 1$, $i = 1 \ldots N$ and $\mu = 1 \ldots P$ $(P = \alpha \cdot N)$ is generated and the desired learning rule (Hebb, iterative, Minover, energy minimization, etc.) is applied.

To study the basins of attraction around a given pattern ξ^μ, random test patterns $\xi^{\mu,r}$ with initial overlap $m_0 = 1/N \sum_j \xi_j^\mu \xi_j^{\mu,r}$ with pattern ξ^μ are generated and iterated to stability under the network dynamics. Then, the interesting statistical data is recorded, including the first step overlap m_1 after one step of the dynamics, the final overlap m_f at the fixpoint, the fraction f_p of perfectly recalled test patterns, etc.

A pseude-code notation of the basic simulation algorithm is shown in figure 8.

In previous simulations of saturated networks, no phase-space gardening was intended. Therefore, all patterns ξ^μ had the same stabilities and the simulations could average over all patterns ξ^μ. To study the effects of phase-space gardening, however, the simulations must record data for each pattern and are therefore computationally very expensive.

3.2 Typical simulation data $m_f(m_0)$

As an example of the typical data recorded in the simulations, figure 9 shows the mean final overlap m_f of the test pattern $\xi_i^{\mu,r}$ with a pattern ξ_i^μ as a function of the initial overlap m_0 for networks of different size.

It is obvious that the transition between bad (almost none) and good (almost perfect) recognition of patterns becomes sharper in larger networks, indicating a phase transition between no and perfect recognition in the limit $N \to \infty$.

```
Basins of attraction simulation( network $N$ ):
select global parameters and learning rule ($N$, $P = \alpha \cdot N$)
for $i = 1$ to #simulations
    create network and set of random patterns
    apply learning rule
    for $\mu = 1$ to $P$
        select pattern $\xi_i^\mu$
        for initial overlap $m_0 = m_{0,1}$ to $m_{0,K}$
            for $j = 1$ to #test-patterns
                generate test-pattern
                iterate to stability
                record statistical data (e.g. $m_f$)
            end for
        end for
        calculate pattern data (e.g. $m_f(m_0)$)
    end for
    calculate network data (e.g. $m_c(\kappa)$)
end for
average over networks
```

Figure 8: Basic simulation algorithm

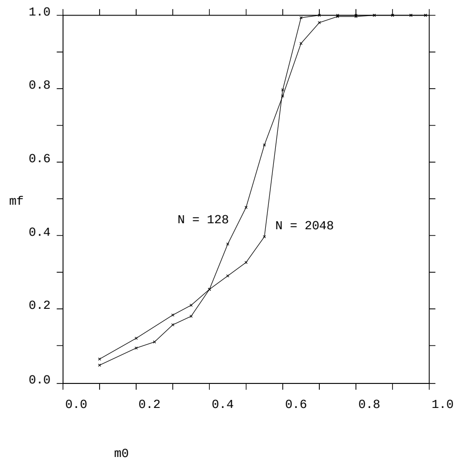
3.3 Estimation of the basins of attraction

Given the simulation data for the final overlap $m_f(m_0)$ as a function of initial overlap at discrete values of m_0 only, there remains the problem to get an accurate estimation for the size of the basins of attraction. In the following the 'critical initial overlap' m_c is used for the size of the basin of attraction. In an infinite sized network, all patterns $\xi_i^{\mu,r}$ with overlap $m_0 > m_c$ are recalled perfertly as ξ^μ and test patterns with overlap $m_0 < m_c$ are not recalled. However, in typical simulations small networks are used and finite size effects have to be studied.

In [Forrest 88] a simple analytical model was proposed for the fraction f_p of perfectly recalled patterns as a function of m_0, critical initial overlap m_c and network size N. The function used was

$$f_p(m_0)/(1 - f_p(m_0)) = a_1 \exp(N\, a_2(m_0 - m_c)) \tag{14}$$

which provided a very accurate fit to the observed simulation data. Examples for the $f_p(m_0)$ simulation data are shown in figure 14 and figure 15 for several memory patterns in the binary couplings model.



Typical data for the mean final overlap m_f as a function of initial test pattern overlap m_0. The diagrams shows binary couplings networks with $\alpha = 0.20$, $N = 128$ and $N = 2048$.
Each data point (cross) is averaged over 1000 random test patterns with overlap m_0 with a randomly selected pattern ξ^μ. The lines show the linear interpolation between the data points.
Note that the transition between perfect ($m_f \approx 1$) and no recognition becomes very sharp in larger networks.

Figure 9: Mean final overlap m_f vs. initial overlap m_0, finite-size scaling

In this report, a similar scaling hypothesis is presented for the function $m_f(m_0)$, namely,

$$m_f(m_0) = \frac{a_1 m_0 + \exp(N\, a_2(m_0 - m_c))}{1 + \exp(N\, a_2(m_0 - m_c))}. \tag{15}$$

Typical fits to the simulation data $m_f(m_0)$ for different patterns with large and tiny basins of attraction are shown in figure 10 for a binary-couplings network with $\alpha = 0.15$ and $N = 512$. From this fit it is possible to get an accurate estimation for m_c for a given pattern.

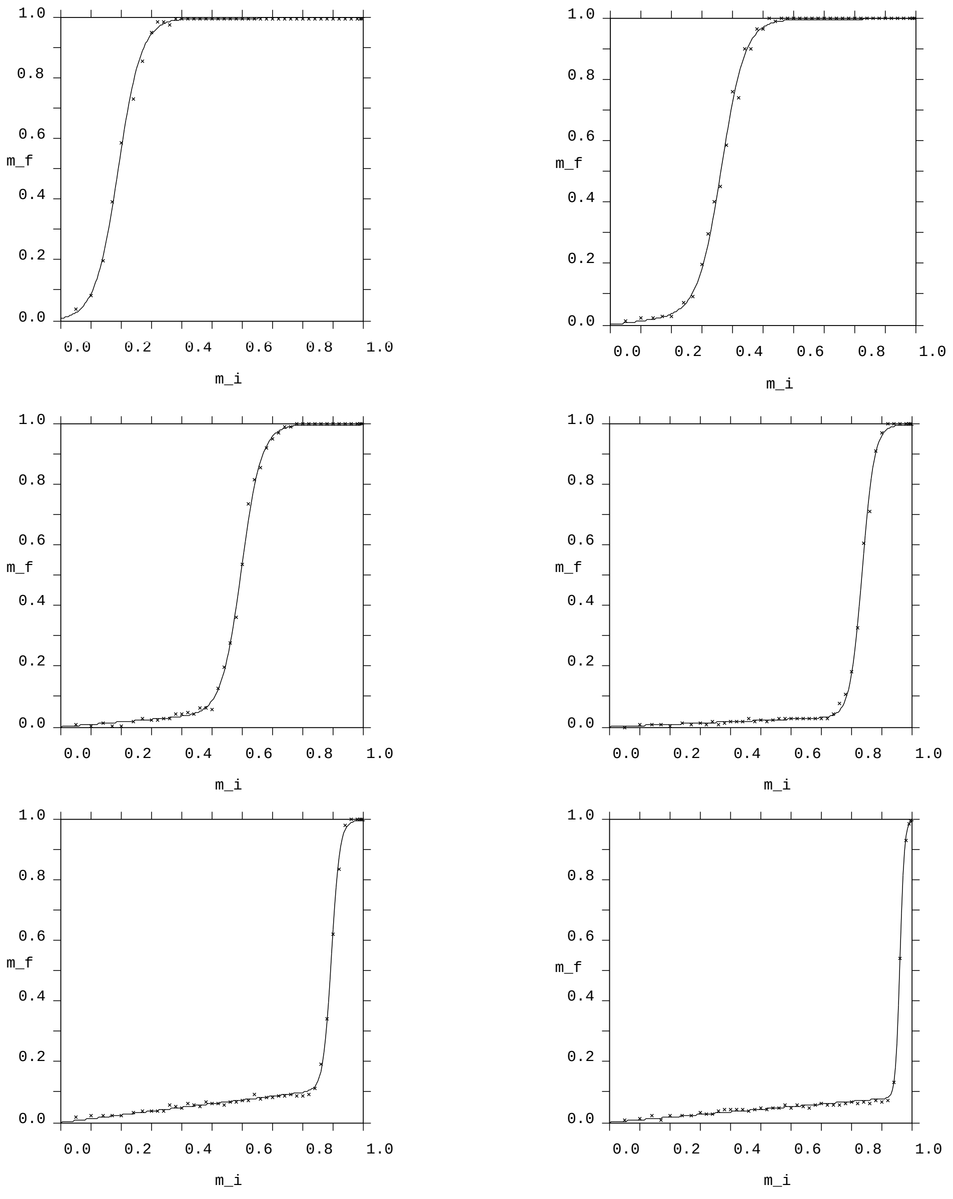
Because the computational effort for the non-linear fit (equation 15) is very high, some of the results presented in sections 4 and 5 use a much simpler linear interpolation to the values $m_f(m_0)$. The value of m_c is taken from $m_f(m_c) = 0.95$.

3.4 Random pattern generation

The basic algorithm for random test pattern generation is simple. Given a pattern ξ^μ and desired initial overlap m_0, N random numbers in the range $[-1, 1]$ are generated. If random number i is less than m_0, then $\xi_i^{\mu,r} = \xi_i^\mu$, else $\xi_i^{\mu,r} = -\xi_i^\mu$.

Note from figure 9 that the transition between almost perfect and no recall as a function of initial pattern overlap m_0 typically is very sharp. This means that it is necessary to generate test patterns with exactly overlap m_0—otherwise the transition in the $m_f(m_0)$ data would 'smear out'.

Therefore, the simple algorithm sketched above cannot be used for the small networks in the simulations, because it leads to finite-size fluctuations of the order $1/\sqrt{N}$. For example, with



Examples for the fit of the model (equation 15) with m_f data from the simulations. The diagrams show $m_f(m_0)$ data (crosses) and the fit (lines) for six different patterns with high (top left) to low (bottom right) minimim stability. The simulations are for a binary couplings network with $\alpha = 0.15$, $N = 512$ and a linear distribution of stabilities, see section 4.2.

Figure 10: Example for the fit of the model (equation 15) with $m_f(m_0)$ data

$N = 1024$ fluctuations of order $1/\sqrt{N}$ correspond to an error of about $\Delta m_0 \approx 0.03$ which is clearly too large.

The solution is to use a better algorithm that gives exactly the desired overlap. In the simulations reported here a two step approach was implemented. First, the simple algorithm is run to get an initial test pattern ξ'. Then the actual overlap between ξ' and ξ^μ is calculated and compared with the desired overlap m_0. If the overlap is too high (low), additional random selected bit positions are flipped (flipped back), until the overlap is correct. The pseudocode is shown in figure 3.4.

```
Random test pattern generation( pattern $\xi^\mu$, overlap $m_0$ ):
for $i = 1$ to $N$ do
    if $(\mathrm{random}(-1,1) < m_0)$ then $\xi'_i = \xi^\mu_i$
    else $\xi'_i = -\xi^\mu_i$
end for
calculate overlap $m'$ between $\xi'$ and $\xi^\mu$
if $(m' = m_0)$ return $\xi'$
while( $m' > m_0$) do
    find index $i$ with $\xi'_i = \xi^\mu_i$
        flip $\xi'_i$, update $m'$
end while
while( $m' < m_0$) do
    find index $i$ with $\xi'_i = -\xi^\mu_i$
        flip $\xi'_i$, update $m'$
end while
return $\xi'$
```

Figure 11: Random test pattern generation

3.5 Parallel dynamics

To improve the simulation speed, a slight optimization of the parallel dynamics is used. An array h_i is used to store the local fields of the neurons. If a new test pattern ξ is loaded into the network, the local fields of all neurons are calculated and stored in this array. For the next steps of the dynamics, this array is updated only whenever a neuron flips its state, see figure 12.

When the test pattern has a large overlap with an attractor state, only a few neurons $F << N$ will flip and the algorithm uses $O(F \cdot N)$ operations instead of $O(N^2)$ for the trivial calculation of the parallel dynamics.

```
Fast parallel dynamics( network $N$, initial pattern $\xi$ )
for $i = 1$ to $N$ calculate the local field $h_i(\xi)$
$t = 0$
repeat
    $t = t + 1$
    for $i = 1$ to $N$ do $S_i(t+1) = \mathrm{sgn}(h_i(t))$
    for $i = 1$ to $N$ do
        if $(S_i(t+1) \neq S_i(t))$ then
            for $k = 1$ to $N$ do $h_k(t+1) = h_k(t) + 2J_{ki}S_i(t+1)$
        end if
    end for
until $(S(t+1) = S(t)$ or $t > t_{\mathrm{max}})$
```

Figure 12: Fast parallel dynamics

4 Phase-space gardening: Size of the basins of attraction

This section studies the basins of attraction for two different models in the spherical as well as the binary couplings network.

The first model uses a piece by piece constant distribution of stabilities. That is, the patterns are divided into groups $\{\xi_1 \ldots \xi_k\}$, $\{\xi_{k+1} \ldots \xi_l\}$, $\{\xi_{l+1} \ldots \xi_m\}$, ...and each group is assigned a different stability during learning. If the pattern stabilities correlate with the size of the basins of attraction, then each group of patterns should have basins of attraction of approximately the same size—and the larger the stabilities the larger the basins of attraction.

The second model uses a linear distribution of stabilites. The desired stability of pattern ξ^μ is $\Lambda^\mu = \Lambda_{\min} + (\mu/P) \cdot (\Lambda_{\max} - \Lambda_{\min})$. Therefore, each pattern ξ^μ should have a slightly larger basin of attraction than pattern $\xi^{\mu-1}$.

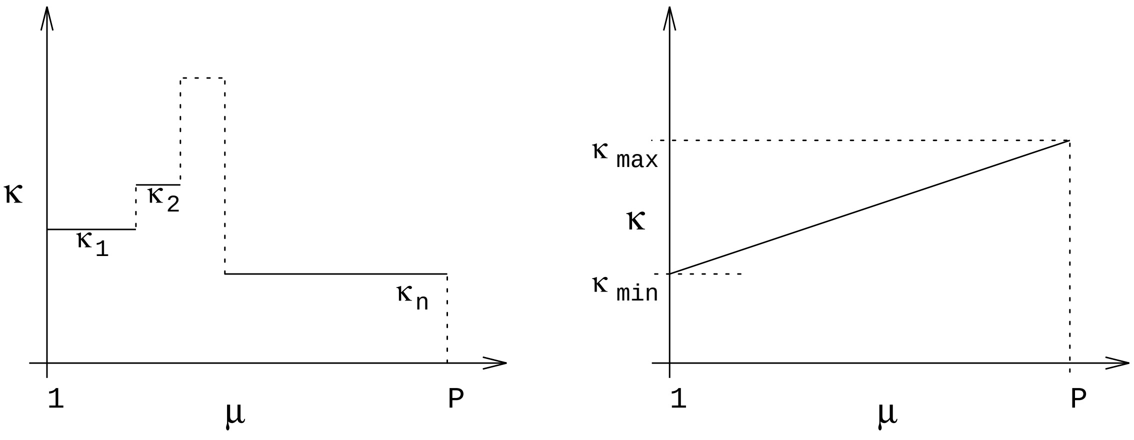


Figure 13: Piece by piece constant and linear distribution of desired stabilities $\rho(\kappa)$

Both example distributions of desired stabilities are sketched in figure 13. The following sections present the simulation results for both models in detail.

4.1 Piece by piece constant distribution of stabilities

The standard learning algorithms give the same global stability to all patterns. Perhaps the most simple generalization is to divide the patterns into groups $\{\xi_1 \ldots \xi_k\}$, $\{\xi_{k+1} \ldots \xi_l\}$, $\{\xi_{l+1} \ldots \xi_m\}$, and to use a learning rule that gives the same stabilities for each pattern of a group but different stabilities for the different groups. With local stability learning rules this is possible with a piece by piece constant function for the desired stabilities Λ_i^μ.

Typical simulation data is shown in figure 14 for a binary couplings network with $\alpha = 0.1$ and $N = 512$. The desired stabilities were set to $\kappa = 2.3$ and $\kappa = 1.8$ for the first 10% and the second 10% of the patterns, while a lower value of $\kappa = 0.9$ was used for the remaining 80% of the patterns.

The resulting distribution of stabilities after learning is shown in figure 14a. It shows the three peaks expected for a superposition of the distributions from the three pattern groups. However, the learning rule did not reach the desired stabilities. The maxima are at $\kappa \approx 1.9$, 1.4 and 0.7.

After learning the basins of attraction were estimated for each pattern group. That is, three simulations were run. First, test patterns were generated and iterated to stability for the patterns with desired stability $\kappa = 2.3$, then for the second ($\kappa = 1.8$), and finally the third pattern group ($\kappa = 0.9$).

The simulation data for the mean final overlap $m_f(m_0)$ and the fraction of perfectly recalled patterns $f_p(m_0)$ are shown in figure 14b and 14c. It is easy to see that the basins of attraction are very large, around $m_c \approx 0.2$ for the patterns with high stability (desired stability $\kappa = 2.3$), about $m_c \approx 0.35$ for the patterns with stability $\kappa = 1.8$, and smaller ($m_c \approx 0.65$) for the remaining patterns with $\kappa = 0.9$.

A similar experiment with five groups of patterns at storage density $\alpha = 0.15$ is shown in figure 15. The desired stabilities were $\kappa = 2.1$, 1.8, 1.6, 1.4 and 1.1 for 10%, 10%, 10%, 10%, and 80% of the patterns. The resulting distribution of stabilities is shown in figure 15a, with the mean final overlap m_f and the fraction of perfectly recalled patterns f_p in figure 15b and 15c. As in the previous experiment, the m_f and f_p data was averaged over all patterns from the corresponding pattern group.

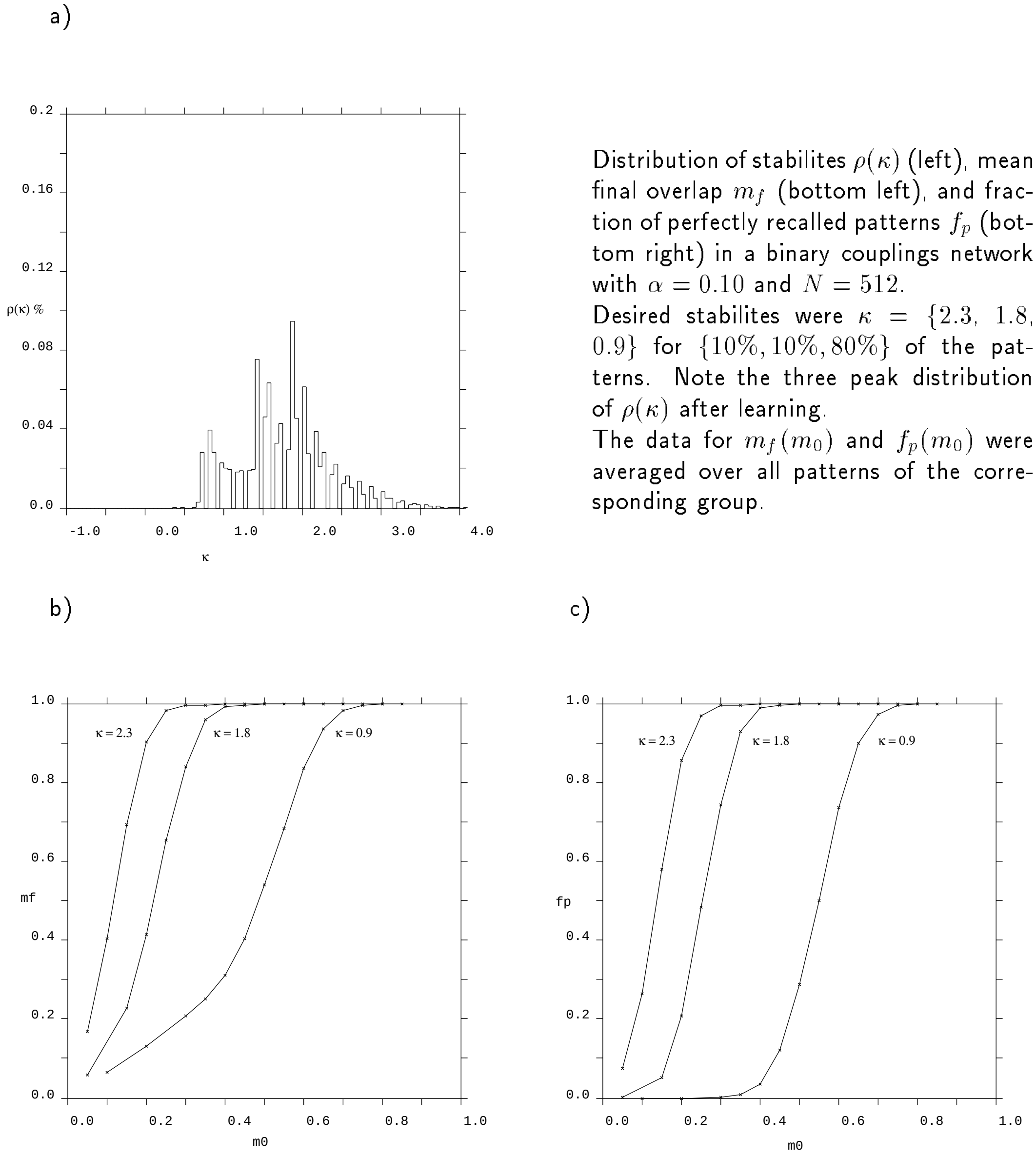
Again, the learning rule allows to control the size of the basin of attraction. The $m_f(m_0)$ data indicate basins of attraction of about $m_c \approx 0.2$, 0.28, 0.38, 0.4 and 0.65 for the different pattern groups.

In order to further test the local stability learning concept, the next simulations do not average over the patterns from each pattern group with the same desired stability. Instead, several patterns are selected at random from the patterns groups and simulated individually. This allows to test whether all patterns with nearly equal stability have basins of attraction of nearly equal size.

The natural way to display these data is the $m_c(\kappa)$ plot shown in figure 16. For each selected pattern, the size m_c of its basin of attraction is determined from the $m_f(m_0)$ data and plotted versus its minimum or mean stability. If the basins of attraction are equal for all patterns with equal stability, then the plot should only show clusters of (m_c, κ) points for the different stability groups.

This is shown in figure 16 for a network with $\alpha = 0.15$ and desired pattern stabilities $\kappa = 2.6$, 2.2, 1.8, 1.4, and 0.8 ($N = 512$). The correlation between the pattern stability and the size of the basin of attraction is obvious. In the same network, some patterns have basins of attraction as large as $m_c = 0.18$ (corresponding to a mean stability of $\kappa_{\rm mean} = 2.1$), while most patterns have $m_c = 0.85$ at $\kappa_{\rm mean} = 1.0$.

The local stability learning rules can also be used for phase-space gardening in the binary couplings networks at higher storage ratio. In figure 17 two examples are shown for $\alpha = 0.25$ and $N = 512$. The left plot shows a network with desired pattern stabilities $\kappa = \{2.5,\ 2.0,\ 1.5,\ 0.8\}$ for 10%, 10%, 10%, and 70% of the patterns. The right plot shows a network with desired pattern stabilities $\kappa = \{2.3,\ 1.8,\ 1.4,\ 0.7\}$. In both simulations, the basins of attraction are small for the patterns with lowest stability ($m_c \approx 0.97$), but much larger (up to $m_c = 0.25$) for the patterns with higher stability.

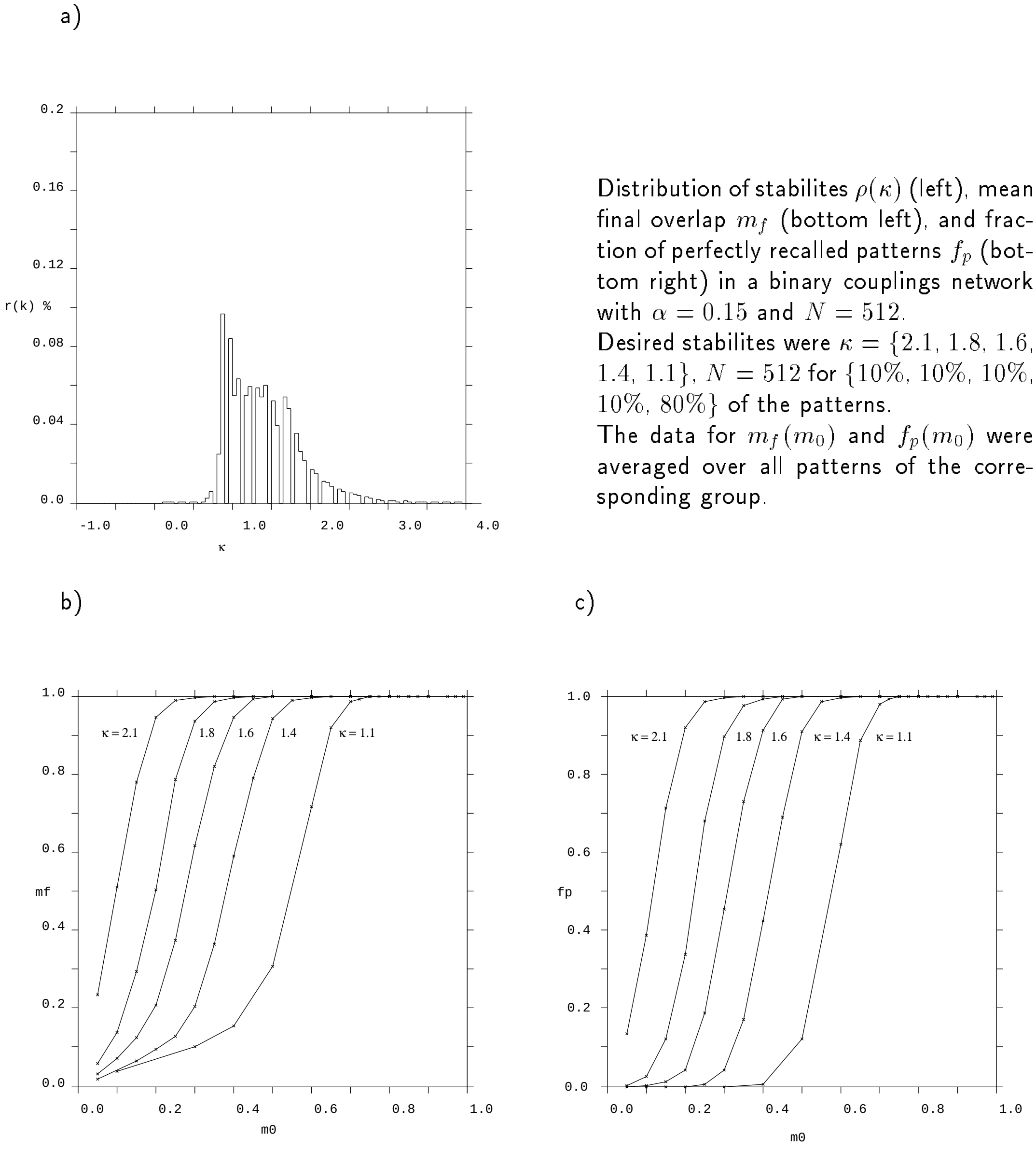


Distribution of stabilites $\rho(\kappa)$ (left), mean final overlap m_f (bottom left), and fraction of perfectly recalled patterns f_p (bottom right) in a binary couplings network with $\alpha = 0.10$ and $N = 512$.

Desired stabilites were $\kappa = \{2.3,\ 1.8,\ 0.9\}$ for $\{10\%, 10\%, 80\%\}$ of the patterns. Note the three peak distribution of $\rho(\kappa)$ after learning.

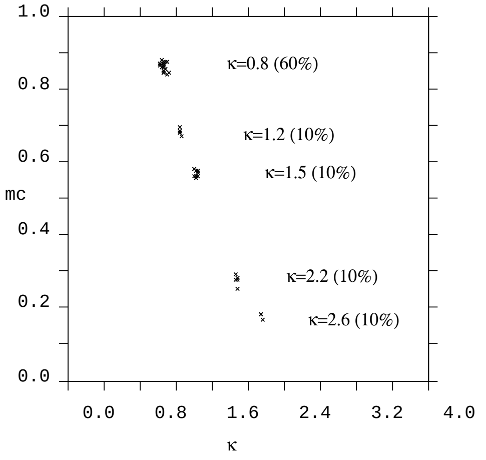
The data for $m_f(m_0)$ and $f_p(m_0)$ were averaged over all patterns of the corresponding group.

Figure 14: Piece by piece constant $\rho(\kappa)$, binary couplings network, $\alpha = 0.10$



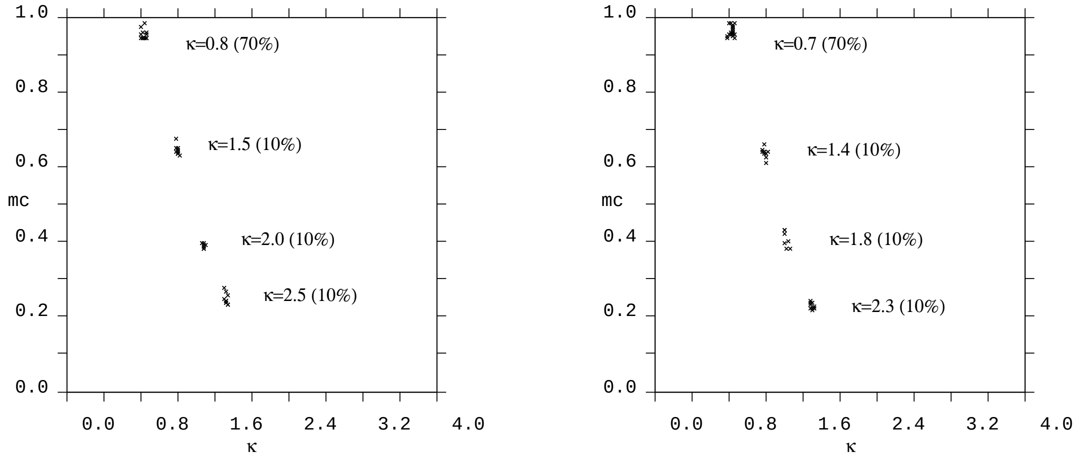
Distribution of stabilites $\rho(\kappa)$ (left), mean final overlap m_f (bottom left), and fraction of perfectly recalled patterns f_p (bottom right) in a binary couplings network with $\alpha = 0.15$ and $N = 512$.
Desired stabilites were $\kappa = \{2.1, 1.8, 1.6, 1.4, 1.1\}$, $N = 512$ for $\{10\%, 10\%, 10\%, 10\%, 80\%\}$ of the patterns.
The data for $m_f(m_0)$ and $f_p(m_0)$ were averaged over all patterns of the corresponding group.

Figure 15: Piece by piece constant $\rho(\kappa)$, binary couplings network, $\alpha = 0.15$



Critical initial overlap m_c (size of the basin of attraction) as a function of mean pattern stability κ_{mean} in a binary couplings network with $\alpha = 0.15$ ($N = 512$). In this simulation m_c was calculated individually for some randomly selected patterns from each stability group.
Desired stabilites were $\kappa = \{2.6, 2.2, 1.8, 1.4, 0.8\}$ for $\{10\%, 10\%, 10\%, 10\%, 80\%\}$ of the patterns.

Figure 16: $m_c(\kappa)$ in the binary couplings network, $\alpha = 0.15$, piece by piece constant $\rho(\kappa)$



Same as figure 16, but for higher storage density $\alpha = 0.25$, $N = 512$.
left: $\kappa = \{2.5, 2.0, 1.5, 0.8\}$, right: $\kappa = \{2.3, 1.8, 1.4, 0.7\}$

Figure 17: $m_c(\kappa)$ in the binary couplings network, $\alpha = 0.25$, piece by piece constant $\rho(\kappa)$

4.2 Linear distribution of stabilities, spherical model

This section presents simulation data for the Hopfield-Gardner network with integer couplings (spherical model) under a linear distribution of pattern stabilities.

That is, the desired stability for pattern ξ^μ is set to $\Lambda_\mu = (\mu/P) \cdot \kappa_{\max}$, and then the modified Minover learning rule (figure 4) is applied. If the correlation between pattern stability and size of the basin of attraction is monotone, then each pattern ξ^μ should have a larger basin of attracton than pattern $\xi^{\mu-1}$. Therefore all patterns have to be simulated in order to determine their basins of attraction.

A typical simulation result at low storage density $\alpha = 0.3$ is shown in figure 18. The bottom diagram gives the minimum and mean pattern stability $\kappa_{\min}$ and κ_{mean} as a function of pattern index after learning. While the modified Minover algorithm could not reach the desired minimum stabilities, $\Lambda_\mu = 3.5 \cdot (\mu/P)$, it did install a linear distribution with approximately $\Lambda_\mu \approx 3.5 \cdot (\mu/P) - 0.8$. Some patterns with low index are not stored at all.

The upper diagram presents the size of the basins of attraction as a function of pattern index. Only patterns with $m_c < 1$ are shown. For patterns that are recognized at all, one has $m_c(\mu) < m_c(\mu - 1)$, the expected behaviour.

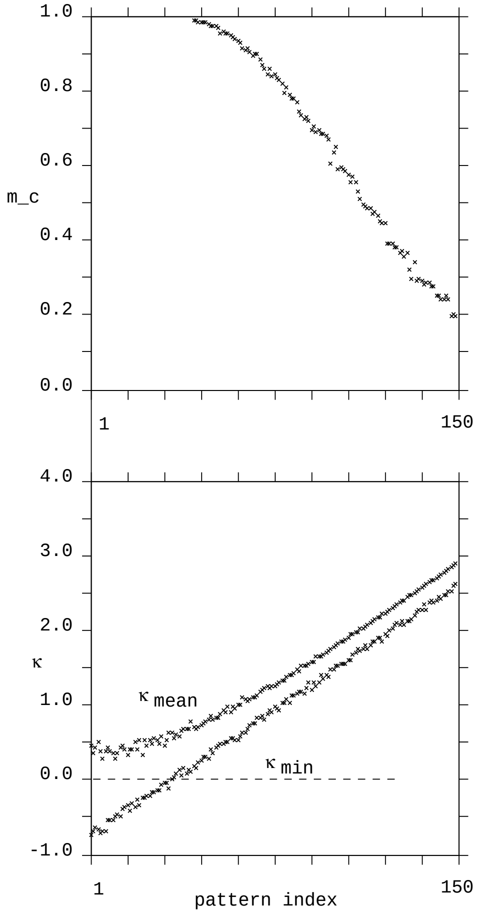
The comparison between the upper and the bottom diagram shows the correlation between pattern minimum stability and size of the basin of attraction. The corresponding plot of m_c versus $\kappa_{\min}$ is shown in figure 19b.

The dependence of m_c on $\kappa_{\min}$ is approximately linear for stabilities $\kappa_{\min} > 0.8$. For minimum stabilities $0 < \kappa_{\min} < 0.8$ the function $m_c(\kappa_{\min})$ is nonlinear, but can be approximated by a low degree polynomial (even a quadratic function suffices). The exact functional dependence is not known. Unfortunately, the compuational effort for an accurate estimation of $m_c(\kappa_{\min}, \alpha)$ is extremely high and out of the scope of this study.

In any case, the results show that the size of basins of attraction of each pattern corresponds to its stability, which in turn can be set during learning with a local stability learning rule.

Similar results for networks with other values of the storage density are shown in figure 19a, 19c, and 19d for $\alpha = 0.2$, 0.5 and 0.7 respectively. Smaller (larger) values of α were not studied in the simulations, because the basins of attraction in these networks are very large (small) anyway.

Note that the dependence on $m_c(\kappa_{\min})$ differs for all four simulations from figure 19, though the overall behaviour is similar.

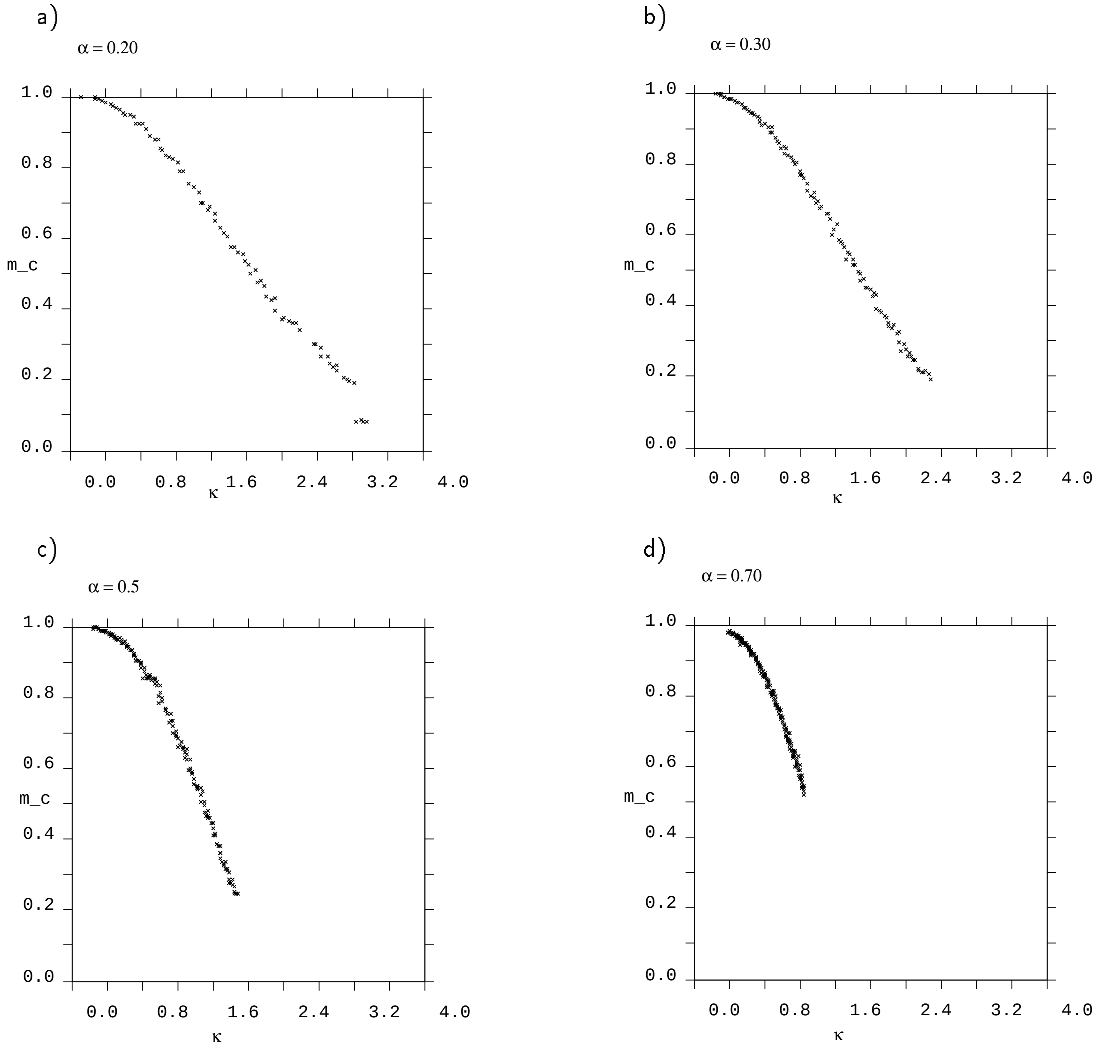


The lower diagram shows the minimum $\kappa_{\rm min}$, and mean $\kappa_{\rm mean}$ pattern stabilites after Minover PSG learning with a linear distribution of desired stabilities, $\Lambda^\mu = 3.5 \cdot \mu/150$ in a network with $N = 512$, $\alpha = 0.3$.

Note that the learning rule did not succeed to install positive minimum stabilites for all patterns, because the desired maximum stability ($\kappa = 3.5$) was too large—some patterns are therefore not stored. However, the learning rule did install a linear distribution of minimum stabilities.

The upper diagram shows the critical initial overlap m_c (the size of the basins of attraction) for each pattern. The patterns with minimum stability less than (or near) zero are not recalled.

Figure 18: $m_c(\mu)$ and $\kappa(\mu)$, linear distribution of stabilites, spherical model, $\alpha = 0.3$



Size of the basins of attraction m_c as a function of pattern stability κ_μ in the spherical model with Minover PSG learning ($N = 512$).
The simulations used a linear distribution of desired stabilities $\Lambda^\mu = (\mu/P) \cdot \kappa_{\max}$ with parameters
$\alpha = 0.20$, $\kappa_{\max} = 4.0$ (top left),
$\alpha = 0.30$, $\kappa_{\max} = 3.5$ (top right),
$\alpha = 0.50$, $\kappa_{\max} = 2.5$ (bottom left),
$\alpha = 0.70$, $\kappa_{\max} = 1.5$ (bottom right) (Only patterns $\mu > 70$ are shown).
Because of their small stabilities, the first few patterns are not recalled.

Figure 19: $m_c(\kappa)$ in the spherical model, linear $\rho(\kappa)$

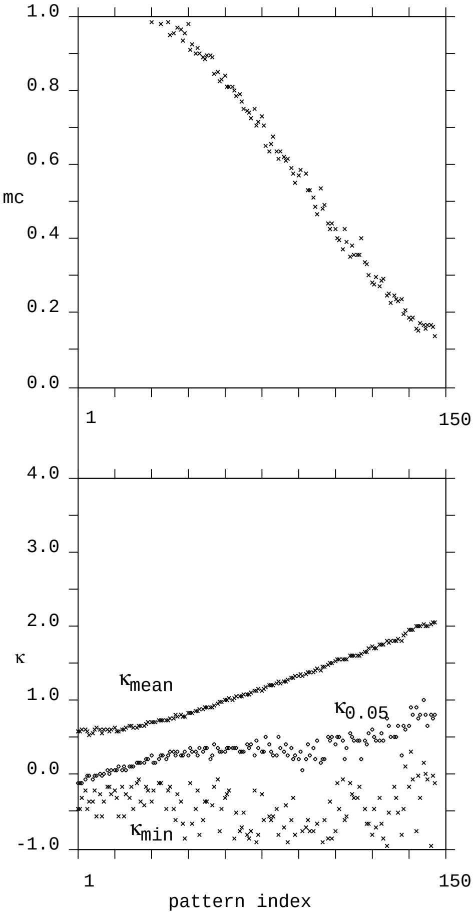
4.3 Linear distribution of stabilities: binary couplings model

Unlike the Minover learning algorithm, the energy-minimization algorithms for the binary couplings networks cannot guarantee the desired values for the minimum stabilities. Rather, the minimum stabilities vary greatly after learning, and some of the stabilities may even be negative—the patterns will be stored with some bit errors. Only the overall distribution of stabilities $\kappa_{i\mu}$ will follow the desired shape.

This is shown in figure 20, where in the bottom diagram the minimum, 5% percentile, and mean pattern stabilities are plotted against the pattern index for a binary couplings network with $\alpha = 0.15$, $N = 1024$ after energy-minimization PSG learning. While the mean stabilities show a smooth behaviour, the minimum stabilities vary greatly (and most are smaller than zero) and even the 5% percentile stabilities are somewhat scattered.

As in figure 18 for the spherical model, the upper diagram from figure 20 plots the size of the basins of attraction m_c versus the pattern index. Interestingly, the plot of $m_c(\mu)$ is rather smooth—indicating a correlation of m_c with κ_{mean} rather than κ_{min}. Again, the patterns with too low stabilities are not recalled by the network. These patterns, with $m_c > 1$, are not shown in the diagram.

As a consequence of the scattering of the values of κ_{min} and even $\kappa_{0.05}$, the following plots of the simulations of the binary couplings model will show m_c as a function of κ_{mean}. In figure 21 four simulations with $\alpha = 0.10$, 0.15, 0.20 and 0.25 are shown.



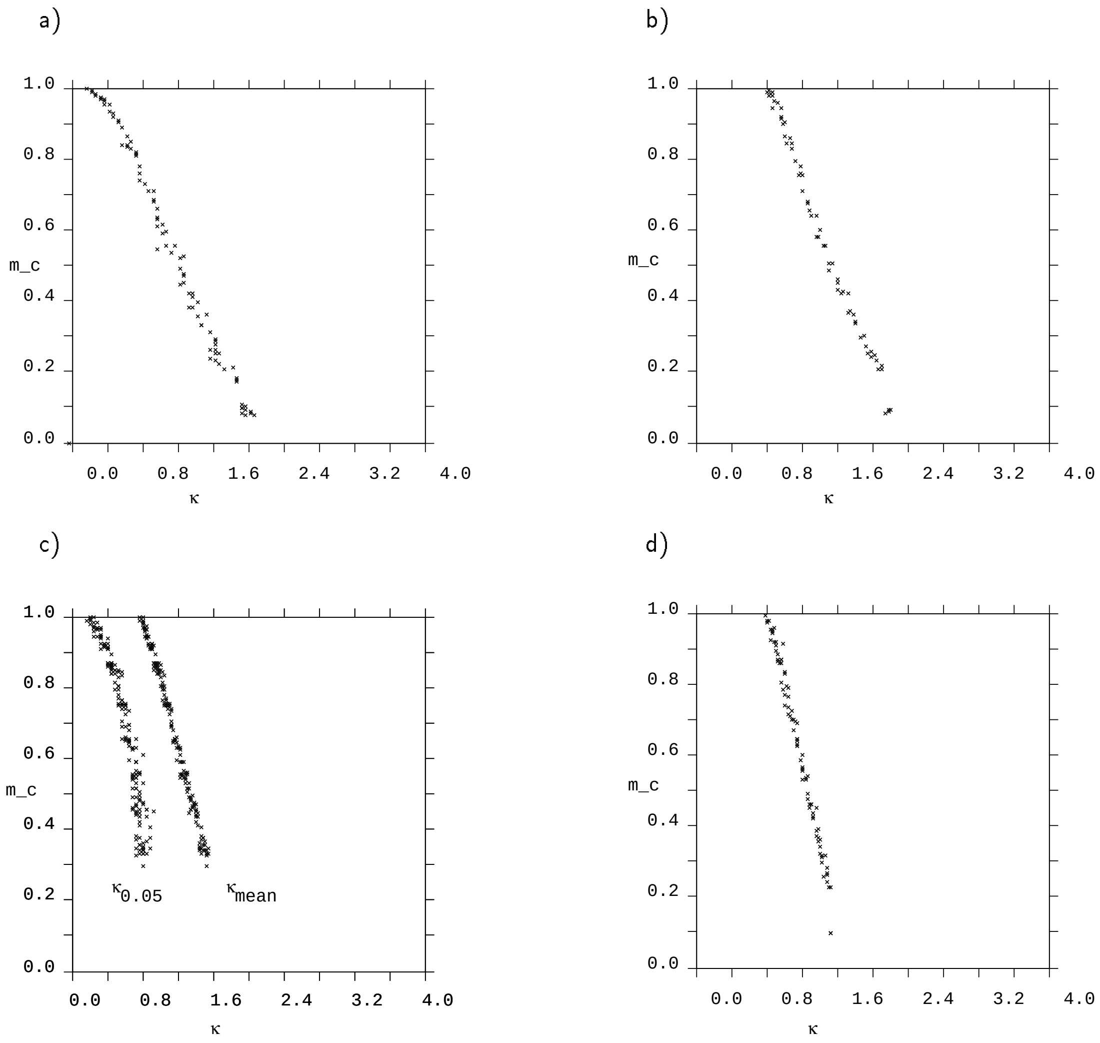
The lower diagram shows the minimum, 5% percentile, and mean pattern stabilites $\kappa_{\min}$ (cross), $\kappa_{0.05}$ (diamond), and κ_{mean} (cross) after energy-minimization PSG learning with a linear distribution of desired stabilities in the binary couplings network, $\rho(\Lambda)(i) = 3.0 \cdot i/150$ in a network with $\alpha = 0.15$, $N = 1024$.

Note that the learning rule (figure 6) cannot guarantee minimum stabilities larger than zero. The distribution $\rho(\kappa)$ always has a little tail of too small stabilities. The diagram shows that the minimum stability is less than zero for almost all patterns.

The upper diagram shows the critical initial overlap m_c (the size of the basins of attraction) for each pattern. Several patterns with too low stabilities are not recalled at all.

For the recalled patterns, however, the size of the basins of attraction depends strongly on the value of the pattern stability. The values of m_c in the same network range from very little basins of attraction $m_c \approx 1.0$ to very large ones $m_c \approx 0.12$.

Figure 20: $m_c(\mu)$ and $\kappa(\mu)$, linear distribution of stabilites, binary couplings model, $\alpha = 0.15$



Size of the basins of attraction m_c as a function of pattern stability $\kappa_{\mu,\{\min,\max\}}$ in the binary couplings model with energy-minimization PSG learning ($N = 512$).
All simulations used a linear distribution of desired stabilities $\Lambda^\mu = (\mu/P) \cdot \kappa_{\max}$ with parameters
$m_c(\kappa_{0.05})$, $\alpha = 0.10$, $\kappa_{\max} = 4.0$ (top left),
$m_c(\kappa_{\text{mean}})$, $\alpha = 0.15$, $\kappa_{\max} = 3.0$ (top right),
$m_c(\kappa_{0.05})$ and $m_c(\kappa_{\text{mean}})$, $\alpha = 0.20$, $\kappa_{\max} = 2.5$ (bottom left),
$m_c(\kappa_{\text{mean}})$, $\alpha = 0.25$, $\kappa_{\max} = 2.0$ (bottom right).
Because of their small stabilities, the first few patterns are not recalled. As an example, the third simulation (bottom left) shows both the $m_c(\kappa_{0.05})$ and the $m_c(\kappa_{\text{mean}})$ data.

Figure 21: $m_c(\kappa)$ in the binary couplings network, linear $\rho(\kappa)$

5 Phase-space gardening: Shaping the basins of attraction

As the results of section 4 show, local stability learning rules allow to set the size of the basins of attraction for each pattern. There remains the question whether the shape of the basins of attraction can be adjusted as well.

To this end, in section 5.1 the very simple LR model is introduced and the corresponding simulation strategy is presented in section 5.2. A first naive implementation using the energy-minimization PSG learning rule is described in section 5.3. Simulation results which show LR anisotropy in the basins of attraction are reported in section 5.4.

5.1 The LR model

Unlike other neural network models, the neuron and pattern values in a Hopfield-Gardner network are binary only. Therefore a neuron either has the correct value (for a given pattern) or not. There are no analog middle values. So what is meant with 'shape' of the basins of attraction?

The goal of the Hopfield network is to recognize the trained patterns from noisy input patterns. To shape the basin of attraction of a pattern ξ^μ therefore means to specify in detail what inputs should be recalled as the pattern ξ^μ. For example, one could wish that some test patterns are recalled, although they contained much noise in some region (say, the first half of bits), but less noise in the remaining bit positions.

This section presents a very simple model for this kind of anisotropy, called the LR model. The idea is to assign different stabilites to the left and right halves of each pattern while learning with a local stability learning rule. That is, the patterns ξ^μ are divided into two groups. The first patterns $\mu = 1, \ldots, P/2$ are assigned the desired stabilities $\kappa_{i\mu} = \{\kappa_L, \kappa_R\}$ and the remaining patterns $\mu = P/2 + 1, \ldots, P$ are assigned the stabilities $\kappa_{i\mu} = \{\kappa_R, \kappa_L\}$. These distributions of stabilities for the first and second halves of patterns are sketched in figure 22.

With $\kappa_L > \kappa_R$ this choice of the distribution of stabilities means that the first half of patterns should be recognized better (from less initial overlap) at the left half of bit positions (neurons) and worse at the right half of neurons—and vice versa.

5.2 The LR simulation strategy

To test whether the basins of attraction show a left/right aniostropy after LR learning, naturally, one has to simulate the network with anisotropic test patterns.

To generate these test patterns, the test pattern generation procedure from figure 3.4 is called twice for each pattern. First it is applied with desired overlap $m_{0,L}$ to the left half of a memory pattern and then with desired overlap $m_{0,R}$ to the right half. This is illustrated in figure 23. The memory pattern from figure 2 and two test patterns with desired overlap $m_0 = \{m_{0,L}, m_{0,R}\} = \{1.0, 0.5\}$ and $m_0 = \{0.5, 1.0\}$ are shown.

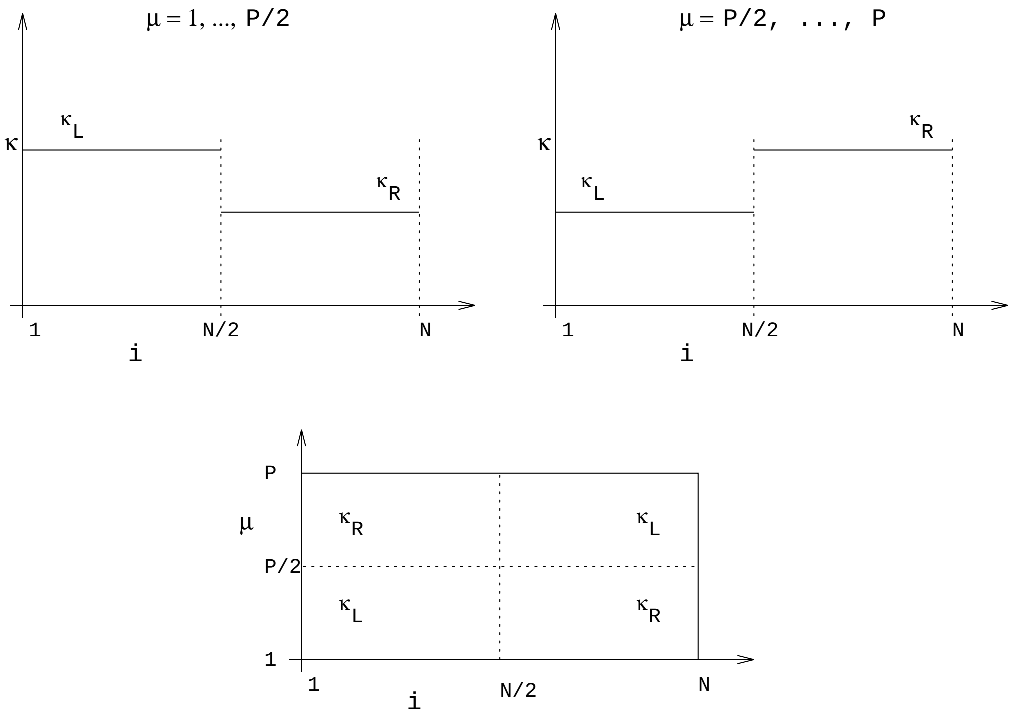


Figure 22: Desired Distribution of stabilities in the LR model

To estimate the basins of attraction of the LR network, one could simulate the network with many different combinations of the values $m_{0,L}$ and $m_{0,R}$. It would then be possible to reconstruct the two-dimensional function of $m_f(m_{0,L}, m_{0,R})$ and from this to estimate the size of the basins of attraction as a function of both $m_{0,L}$ and $m_{0,R}$. However, the computational effort to simulate each pattern (several hundred test patterns per pattern per combination of $m_{0,L}$ and $m_{0,R}$) is by far too large.

The simulations reported here therefore limit the exploration of the $m_f(m_{0,L}, m_{0,R})$ surface. Only two simulations are run per memory pattern. First $m_{0,L}$ is varied, while $m_{0,R} = m_{\text{const}}$ is held constant. Then $m_{0,R}$ is varied while $m_{0,L} = m_{\text{const}}$ is constant. The constant value in each case is chosen so that $m_f(m_0 \approx 1, m_{\text{const}}) \approx 1$, but $m_f(m_0 \approx 0, m_{\text{const}}) \approx 0$.

The two simulations then allow to estimate $m_{c,L}$ and $m_{c,R}$ from the $m_f(m_{0,i}, m_{\text{const}})$ and $m_f(m_{\text{const}}, m_{0,i})$ values for each pattern. This strategy is illustrated in figure 24.

If the LR learning succeeds to install a LR anisotropy in the basins of attraction, then $m_{c,L}$ and $m_{c,R}$ will differ for each pattern. One obvious measure of this anisotropy is the difference $\Delta m_c = m_{c,L} - m_{c,R}$. The value $\Delta m_c(\xi^\mu)$ is positive for a pattern ξ^μ, if the pattern is recalled

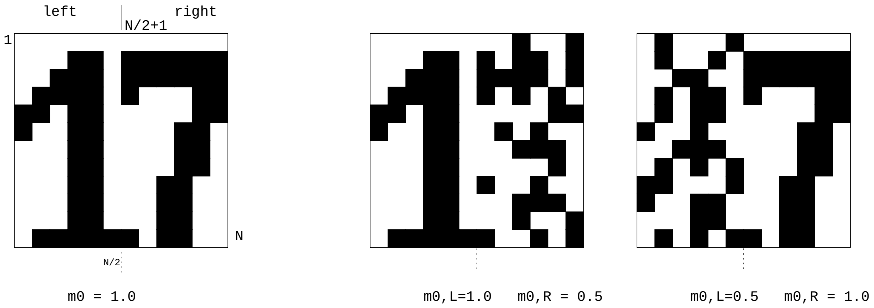


Figure 23: Example pattern and derived LR test patterns

better from test patterns with noisy left half, but recalled worse from test patterns with noisy right half.

The following representation of the simulation data is used in the remainder of this section. The upper part of the diagram shows the values of $m_{c,L}(\xi^\mu)$ (cross) and $m_{c,R}(\xi^\mu)$ (diamond) for each pattern, as a function of pattern index. The lower part of the diagram plots the difference $\Delta m_c(\xi^\mu)$ for each pattern. An example is sketched in figure 25.

Both the absolute size of the basin of attraction of a given pattern, and the relative LR anisotropy are shown in these diagrams. If the LR learning worked well, all patterns in the left half of the diagram (pattern indices $1, \ldots, N/2$) will have positive values of Δm_c, while the patterns in the right half of the diagram should have negative values of Δm_c.

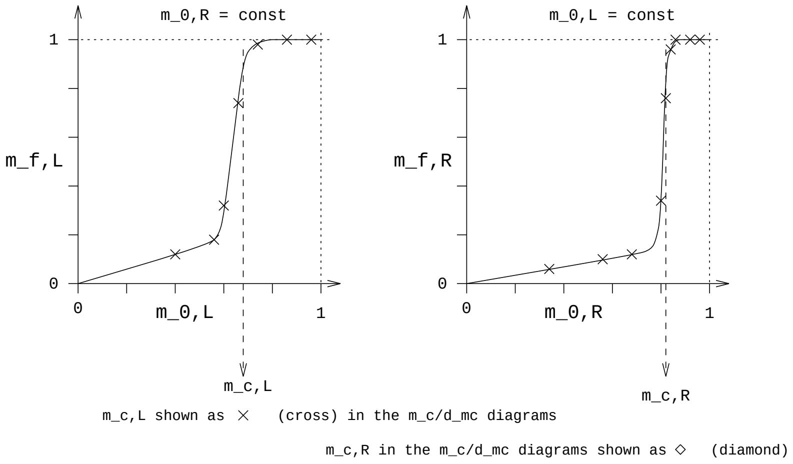


Figure 24: Estimation of left and right critical initial overlap $m_{c,L}$, $m_{c,R}$

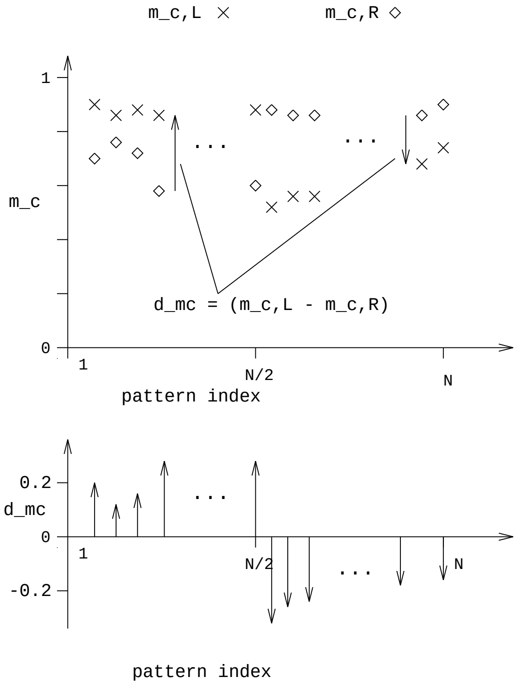
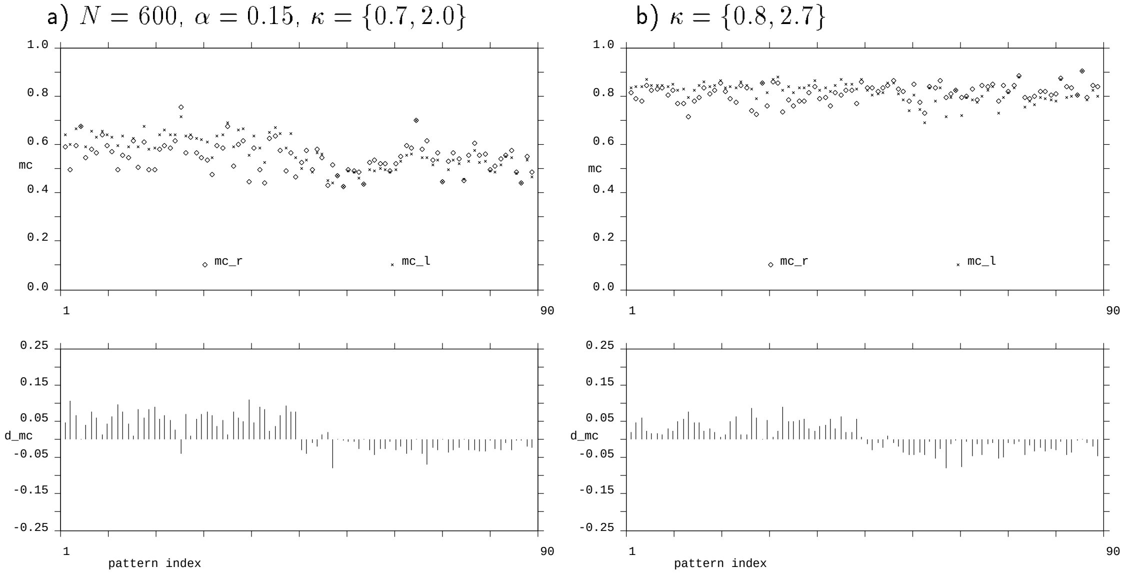


Figure 25: Left and right critical initial overlap, $m_{c,L}$ and $m_{c,R}$, and difference Δm_c

5.3 A first LR implementation

The results of the first run of LR simulations are shown in figure 26. It is easy to see that the learning strategy has let to an LR asymmetry in the recognition in the patterns. However, the LR asymmetry—from the Δm_c plots—is much larger for the first half of patterns $\mu = 1, \ldots, P/2$ than for the remaining patterns.


a) $N = 600$, $\alpha = 0.15$, $\kappa = \{0.7, 2.0\}$

b) $\kappa = \{0.8, 2.7\}$

Figure 26: Basins of attraction $m_c(\mu)$ in the LR model, naive learning rule, $\alpha = 0.15$

Obviously, the learning rule did not work very well for the patterns with $\mu > P/2$. This counter-intuitive result is a consequence of a modification of the energy-minimization PSG learning rule (figure 6) used for the simulations. In this version of the learning algorithm, the index k from the inner loop was simply incremented, rather than chosen randomly. That is, the left half of synapses (J_{ij}, $j = 0, \ldots, N/2$) was set first at each neuron, and the synapses with higher index would more and more depend on the synapses already learnt. Due to the simple gradient descent minimization strategy, the J_{ij} already learnt were not likely to flip. Therefore the LR asymmetry is largest for the synapses set first.

This learning strategy can therefore be used, if a large LR aniostropy is desired for only some of the patterns. The synapses where these patterns should be recognized best are then simply learnt first.

5.4 Basins of attraction in the LR model

This section presents simulation results for the LR model in the binary couplings network. All simulations used the learning algorithm from figure 6, with random selection of index k.

First, an example of the distribution $\rho(\kappa)$ of stabilities after learning with this algorithm and the pattern stability distributions for two randomly chosen patterns are shown in figure 27 for a network with $\alpha = 0.15$ and $N = 600$. The corresponding basins of attraction are shown in figure 28c. The control simulation with $\kappa_L = \kappa_R$ gives an estimate to the size of the finite-size and random fluctuations (figure 28a).

The finite-size scaling of the basins of attraction in the binary couplings network with $\alpha = 0.15$ and $\kappa_L = 0.9$, $\kappa_R = 2.0$ is shown in figure 29 for networks with $N = 300$, $N = 600$, and $N = 1200$. Similar results for storage ratios $\alpha = 0.20$ and $\alpha = 0.25$ are shown in figure 30 and 31.

Storage ratios $\alpha < 0.10$ were not studied in the LR model, because it is very difficult to set small values for the stabilities during learning. Therefore the basins of attraction in these networks are very large for all patterns. Similar, networks with $\alpha > 0.25$ were not studied, because the learning rules fail to set the required large values of the stabilities. Additionally, the basins of attraction are tiny anyway and there is little room for a LR anisotropy.

As the simulations from figures 28 to 31 show, the simple LR strategy allows to set anisotropic basins of attraction in the networks. If no LR asymmetry is used ($\kappa_L = \kappa_R$), the control simulation from figure 28a demonstrates that the basins of attraction are isotropic for all patterns.

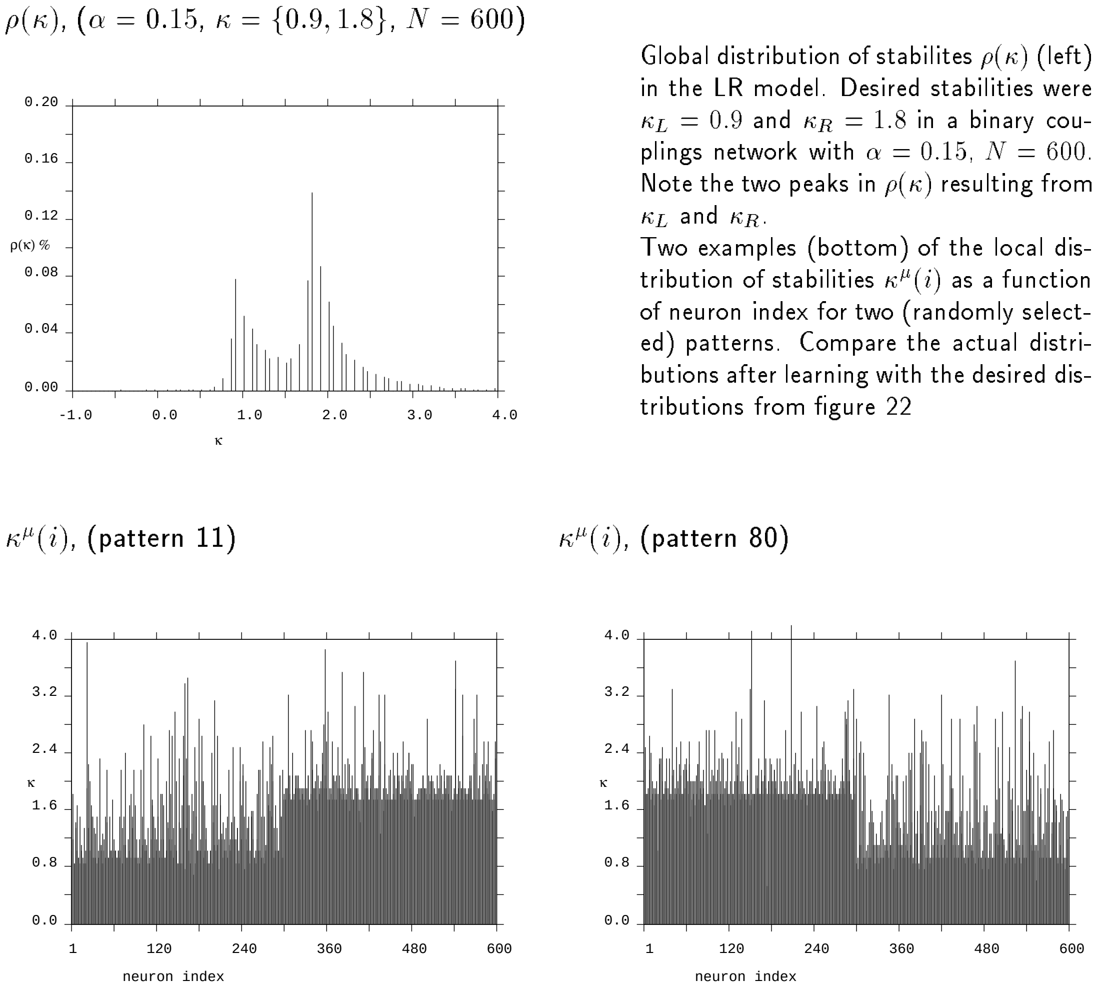
However, the LR simulations indicate that the shape can be controlled to a much lesser extent than the size of the basins of attraction. Recall from the simulations from section 4 that the size of the basins of attraction could be set to values from $m_c = 1 - \epsilon$ to $m_c \approx 0.2$.

The largest Δm_c reached in the simulations is for the network with $\alpha = 0.20$, $\kappa_L = 0.7$ and $\kappa_R = 1.7$ from figure 30d, where $\Delta m_c \approx 0.7$.

Note, too, that the larger the values of Δm_c, the smaller the lower of $m_{c,L}$ and $m_{c,R}$. The learning rule did install anisotropic basins of attraction, but these are smaller than the isotropic basins. Also, the fluctuations of $m_{c,L}$ and $m_{c,R}$ are rather large for the networks studied, see e.g. figure 28c. The finite-size scaling from figure 29 indicates that the fluctuations decrease with increasing network size.

Naturally, a large Δm_c needs large values of κ_R and small values of κ_L. While it is possible to set large stabilities to only a few patterns (as in section 4), the LR simulations need large stabilities for one-half of bit positions of all patterns. This limits the value of κ_R and therefore the amount of LR anisotropy.

A further study of anisotropic basins of attraction might therefore try to install stabilities with large LR asymmetry for only some patterns, with most of the remaining patterns well below the critical pattern stability $\kappa_c(\alpha)$.



$\rho(\kappa)$, ($\alpha = 0.15$, $\kappa = \{0.9, 1.8\}$, $N = 600$)

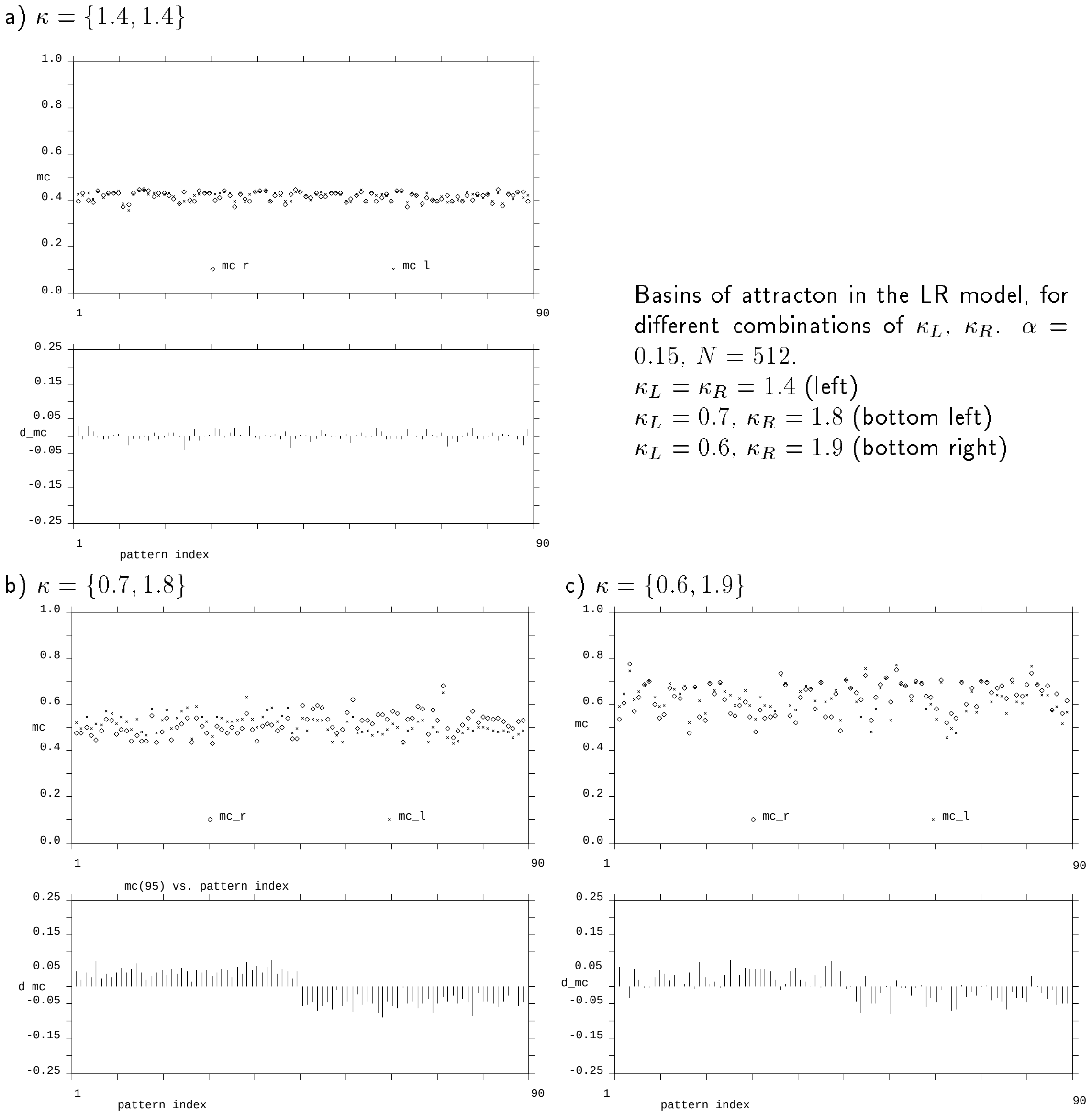
Global distribution of stabilites $\rho(\kappa)$ (left) in the LR model. Desired stabilities were $\kappa_L = 0.9$ and $\kappa_R = 1.8$ in a binary couplings network with $\alpha = 0.15$, $N = 600$. Note the two peaks in $\rho(\kappa)$ resulting from κ_L and κ_R.

Two examples (bottom) of the local distribution of stabilities $\kappa^\mu(i)$ as a function of neuron index for two (randomly selected) patterns. Compare the actual distributions after learning with the desired distributions from figure 22

$\kappa^\mu(i)$, (pattern 11)

$\kappa^\mu(i)$, (pattern 80)

Figure 27: Global distribution of stabilities $\rho(\kappa)$ and examples of the local pattern stabilites $\kappa^\mu(i)$ in the LR model



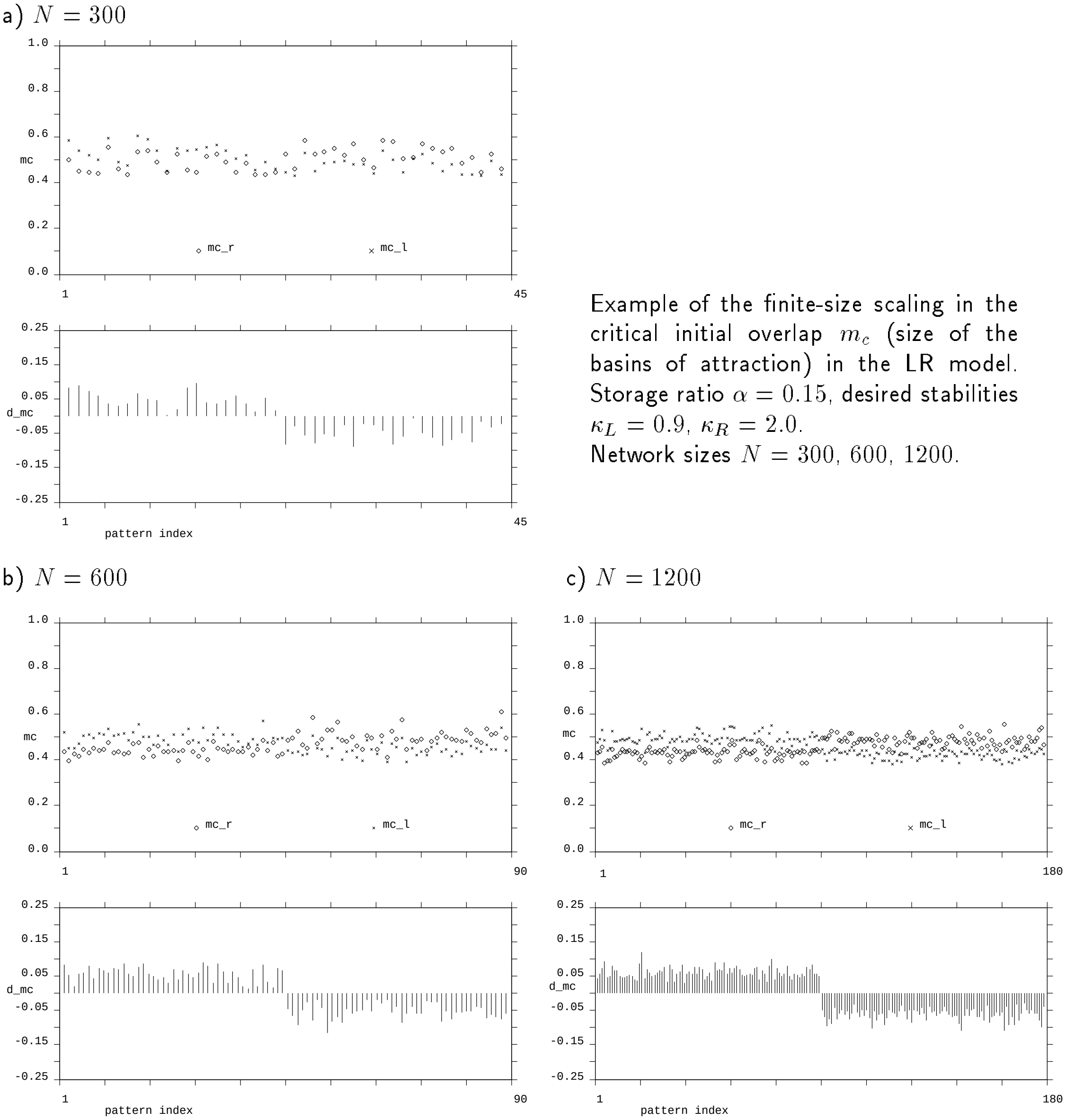
a) $\kappa = \{1.4, 1.4\}$

b) $\kappa = \{0.7, 1.8\}$

c) $\kappa = \{0.6, 1.9\}$

Basins of attracton in the LR model, for different combinations of κ_L, κ_R. $\alpha = 0.15$, $N = 512$.
$\kappa_L = \kappa_R = 1.4$ (left)
$\kappa_L = 0.7$, $\kappa_R = 1.8$ (bottom left)
$\kappa_L = 0.6$, $\kappa_R = 1.9$ (bottom right)

Figure 28: Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.15$ ($N = 600$)

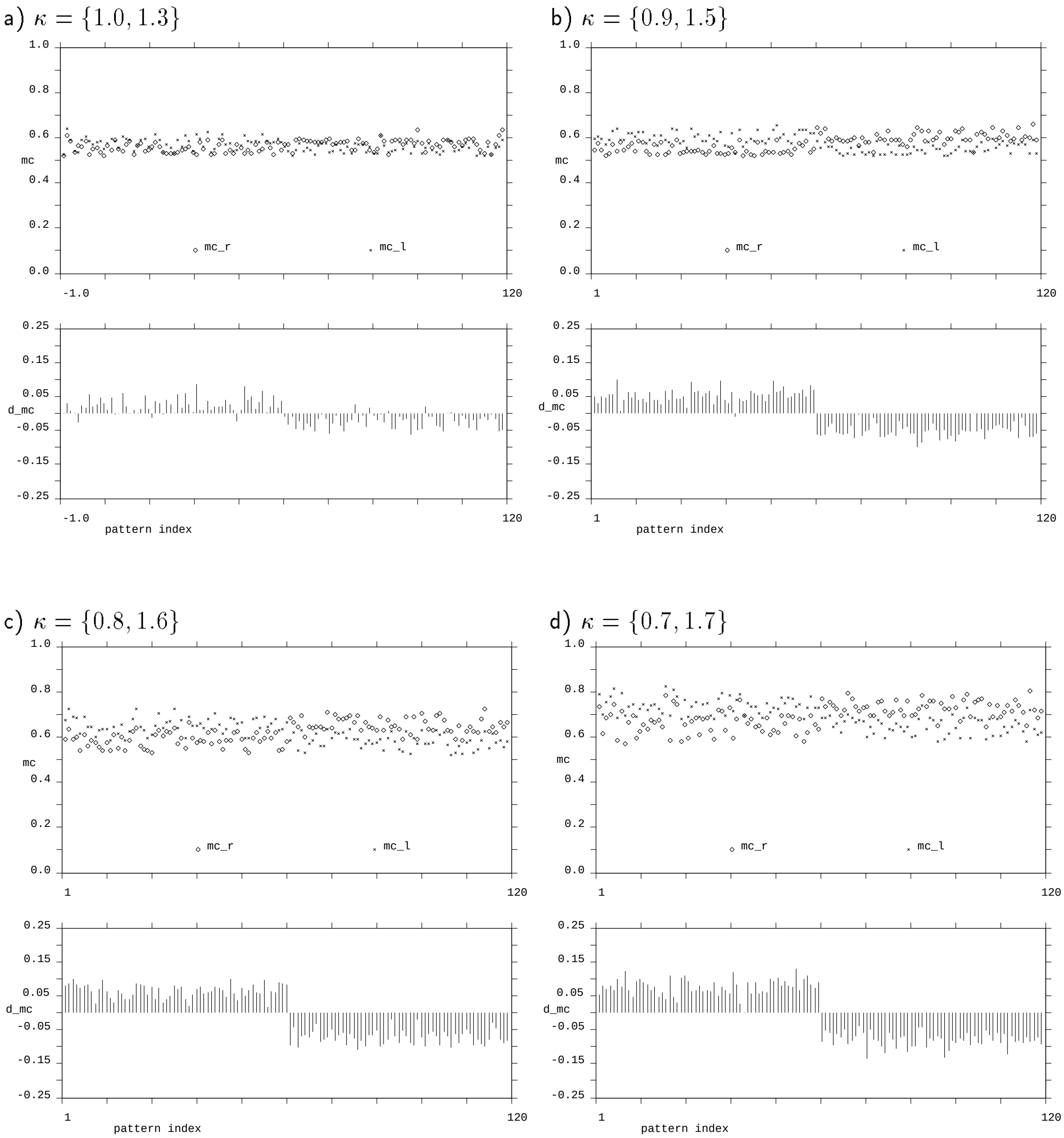

a) $N = 300$

b) $N = 600$

c) $N = 1200$

Example of the finite-size scaling in the critical initial overlap m_c (size of the basins of attraction) in the LR model. Storage ratio $\alpha = 0.15$, desired stabilities $\kappa_L = 0.9$, $\kappa_R = 2.0$.
Network sizes $N = 300$, 600, 1200.

Figure 29: Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.15$, finite-size scaling

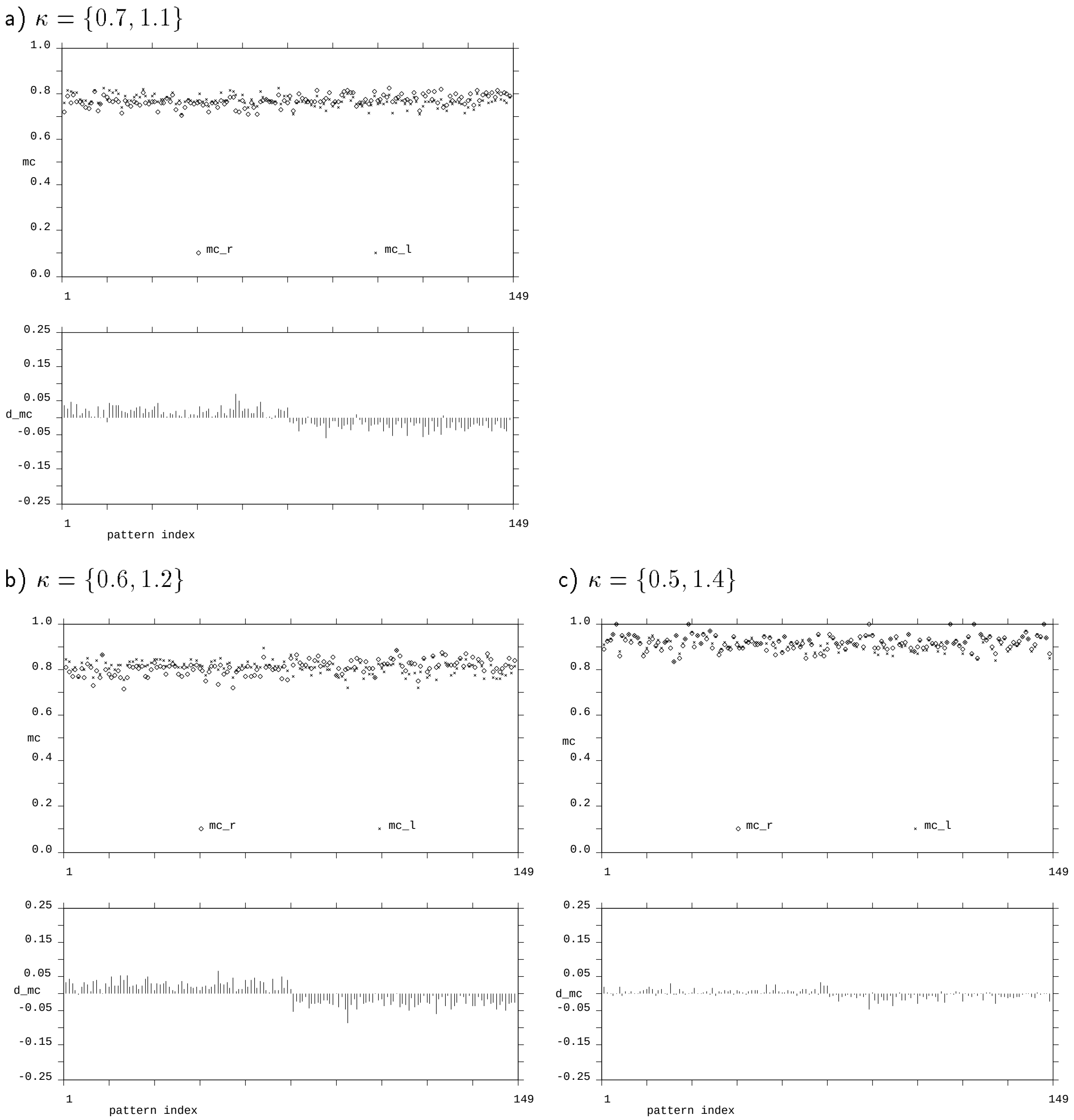

a) $\kappa = \{1.0, 1.3\}$

b) $\kappa = \{0.9, 1.5\}$

c) $\kappa = \{0.8, 1.6\}$

d) $\kappa = \{0.7, 1.7\}$

Figure 30: Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.20$ ($N = 600$)



a) $\kappa = \{0.7, 1.1\}$

b) $\kappa = \{0.6, 1.2\}$

c) $\kappa = \{0.5, 1.4\}$

Figure 31: Basins of attraction $m_c(\mu)$ in the LR model, $\alpha = 0.25$ ($N = 600$)

6 Summary

This report studied the dynamical properties of Hopfield-Gardner attractor neural networks. Based upon the simulations from [Forrest 88] it tried to answer the question whether size and shape of the basins of attraction of the memory patterns depend on the pattern stabilities.

It introduced the concept of 'local stability' learning rules as a means of phase-space gardening. Two local stability learning rules were studied in detail: A straightforward generalization of the Minover algorithm for the Gardner network with integer couplings (spherical model) and a new energy-minimization learning algorithm for the binary couplings network.

The idea of the local stability learning rules is to specify an initial distribution of desired pattern stabilities during the learning phase in order to set the basins of attraction. This was demonstrated for two simple models. The first model used a piece by piece constant distribution of stabilities. The simulation results show that all patterns with the same value of the stabilities indeed have basins of attraction of the same size. Very large basins of attraction (corresponding to low values of $m_c \approx 0.2$) and small basins of attraction (with $m_c \approx 1 - \epsilon$) are possible in the same network.

The second model used a linear distribution of stabilities, $\Lambda^\mu = (\mu/P)\kappa_{\max}$. This results in a network, where the basin of attraction of each pattern ξ^μ is a little larger than the basin of attraction of pattern $\xi^{\mu-1}$. The dependence of $m_c(\kappa)$ is linear for stabilities $\kappa_{i\mu} > 0.8$ and nonlinear below. Unfortunately, it still seems impossible to derive an analytical model for $m_c(\kappa)$.

In both the piece by piece constant and the linear model, the distribution of stabilities does depend on the pattern index μ, but not on the neuron index i. Therefore, the resulting basins of attraction are isotropic.

As a simple way to anisotropic basins of attraction, the LR model was presented in section 5.1. It splits the distribution of stabilities into a left and right half for the first and second half of patterns. The distribution of stabilities then does depend on both pattern index μ and neuron index i. The simulations show that this simple model suffices to install anisotropic basins of attraction.

Furhter work on local stability learning rules may concentrate on correlated pattern storage, where the simple Hebb learning breaks down. Another important application not studied in this report is the storage of temporal sequences.

With the concept of local stability learning rules, this report introduced the means for a 'micro-surgery' of the dynamics of Hopfield/Gardner networks. Both the size and shape of the basins of attraction can be adjusted as desired, within the limits of the critical storage capacity. This makes the networks much more attractive for applications and better candidates for associative recall in brain models.

References

[Abbott & Kepler 89a] L. F. Abbott and T. B. Kepler, Optimal learning in neural network models, *Journal of Physics* A 22, L711–L717 (1989).

[Abbott & Kepler 89b] L. F. Abbott and T. B. Kepler, Universality in the space of interactions for network models, *Journal of Physics* A 22, 2031–2038 (1989).

[Abbott 90] L. F. Abbott, Learning in neural network memories, *Network* 1, 105–122 (1990).

[Amit *et. al.* 85] D. J. Amit, H. Gutfreund and H. Sompolinsky, Spin-glass models of neural networks, *Physical Review A* 32, 1007–1018 (1985).

[Amit *et. al.* 87] D. J. Amit, H. Gutfreund and H. Sompolinsky, Statistical Mechanics of Neural Networks near Saturation, *Annals of Physics* 173, 30–67 (1987).

[Amit *et. al.* 89] D. J. Amit, C. Campbell and K. Y. M. Wong, The interaction space of neural networks with sign-constrained synapses, *Journal of Physics* A 22, 4687–4693 (1989).

[Anderson & Rosenfeld 88] J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing — Foundations of Research*, MIT Press, Cambridge, Mass., 1988

[Binder & Young 86] K. Binder and A. P. Young, Spin Glasses, *Rev. Mod. Phys.* 58-4 (1986).

[Bruce *et. al.* 87] A. D. Bruce, E. J. Gardner and D. J. Wallace, Dynamics and statistical mechanics of the Hopfield-model, *Journal of Physics* A 20, 2909–2934 (1987).

[Crisanti & Sompolinsky 87] A. Crisanti and H. Sompolinsky, Dynamics of spin systems with randomly asymmetric bonds: Langevin dynamics and a spherical model, *Physical Review* A 36, 4922 (1987).

[Derrida *et. al.* 87] B. Derrida, E. Gardner and A. Zippelius, An exactly solvable neural network model, *Europhysics Letters* 4, 481 (1987).

[Diederich & Opper 87] S. Diederich and M. Opper, Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules, *Physical Review Letters* 58, 949–952 (1987).

[Fontanari & Theumann 90] J. F. Fontanari and W. K. Theumann, On the storage of correlated patterns in Hopfield's model, *Journal de Physique (France)* 51, 375–386 (1990).

[Forrest 88] B. M. Forrest, Content-addressability and learning in neural networks, *Journal of Physics* A 21, 245-255 (1988).

[Gardner 86] E. Gardner, Structure of metastable states in the Hopfield model, *Journal of Physics* A 19, L1047–L1052 (1986).

[Gardner 87] E. Gardner, Maximum Storage Capacity in Neural Networks, *Europhysics Letters* 4, 481–485 (1987).

[Gardner 88a] E. Gardner, The space of interactions in neural network models, *Journal of Physics* A 21, 257–270 (1988). (Edinburgh Preprint No. 87/396).

[Gardner & Derrida 88] E. Gardner, B. Derrida, Optimal storage properties of neural network models, *Journal of Physics* A 21, 271–284 (1988).

[Gardner 89a] E. Gardner, Optimal basins of attraction in randomly sparse neural network models, *Journal of Physics* A 22, 1969–1974 (1989).

[Gardner 89c] E. J. Gardner, N. Stroud and D. J. Wallace, Training with noise and the storage of correlated patterns in a neural network model, *Journal of Physics* A 22, 2019–2030 (1989).

[Gutfreund & Stein 90] H. Gutfreund and Y. Stein, Capacity of neural networks with discrete synaptic couplings, *Journal of Physics* A 23, 2613–2630 (1990).

[Hendrich 91] N. Hendrich, Associative memory in damaged neural networks, *Journal of Physics* A 24, 2877–2887 (1991)

[Hendrich 92] N. Hendrich, Associative memory networks, fault-tolerance and coding theory, *Proc. IJCNN-91*, 437–440 (1991)

[Hendrich 93] N. Hendrich, Phase-space gardening in the binary couplings memory network, *Proc. ICANN-93*, 735 (1993)

[Hopfield 82] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci.* 79, 2554–2558 (1982).

[Kanter & Sompolinsky 87] I. Kanter and H. Sompolinsky, Associative recall of memory without errors, *Physical Review* A 35, 380–392 (1987).

[Kepler & Abbott 88] T. B. Kepler and L. F. Abbott, Domains of attraction in neural networks, *Journal de Physique (France)* 49, 1657–1662 (1988).

[Kirkpatrick & Sherrington 78] S. Kirkpatrick and D. Sherrington, Infinite ranged models of spin-glasses, *Phys. Rev. B* 17, 4384 (1978).

[Koehler *et. al.* 89] H. Koehler, S. Diederich, W. Kinzel and M. Opper, Learning Algorithm for a Neural Network with Binary Synapses, *private communication* (Uni Giessen preprint).

[Koscielny-Bunde 90] E. Koscielny-Bunde, Effect of damage in neural networks, *Journal of Statistical Physics* 58, 1257 (1990).

[Koscielny-Bunde 92] E. Koscielny-Bunde, Assoziativspeichernetze mit unvollständiger Vernetzung, *Dissertation*, Universität Hamburg, Fachbereich Informatik (1992).

[Krauth & Mézard 87] W. Krauth and M. Mézard, Learning algorithms with optimal stability in neural networks, *Journal of Physics* A 20, L745–L752 (1987).

[Krauth *et. al.* 88] W. Krauth, J.-P. Nadal and M. Mézard, The roles of stability and symmetry in the dynamics of neural networks, *Journal of Physics* A 21, 2995–3011 (1988).

[Krauth & Opper 89] W. Krauth and M. Opper, Critical storage capacity of the $J = \pm 1$ neural network, *Journal of Physics* A 22, L519–L523 (1989).

[Krauth & Mézard 89] W. Krauth and M. Mézard, Storage capacity of memory networks with binary couplings, *Journal de Physique (France)* 50, 3057–3066 (1989).

[Mézard, Parisi & Virasoro 87] M. Mézard, G. Parisi and M. A. Virasoro, *Spin glass theory and beyond*, World Scientific Lecture Notes in Physics Vol. 9, Singapure 1987.

[Nardulli & Pasquariello 90] G. Nardulli and G. Pasquariello, Domains of attraction of neural networks at finite temperature, *International Neural Network Conference, Paris, 1990*, (Univ. Bari preprint TH/90-66), to appear in *Journal of Physics* A.

[Opper *et. al.* 89] M. Opper, J. Kleinz, H. Koehler and W. Kinzel, Basins of attraction near the critical storage capacity for neural networks with constant stabilities, *Journal of Physics* A 22, L407–L411 (1989).

[Wang *et. al.* 93] T. Wang, X. Zhuang, X. Xing, and X, Xiao, A neuron-weighted learning algorithm and its hardware implementation in associative memories, *IEEE Transactions on Computers* 42, 636–640 (1993)