1

Associative Data Storage and Retrieval in Neural Networks

Günther Palm [1]
Friedrich T. Sommer [2]

ABSTRACT Associative storage and retrieval of binary random patterns in various neural net models with one-step threshold-detection retrieval and local learning rules are the subject of this paper. For different hetero-association and auto-association memory tasks, specified by the properties of the pattern sets to be stored and upper bounds on the retrieval errors, we compare the performance of various models of finite as well as asymptotically infinite size. In infinite models, we consider the case of asymptotically sparse patterns, where the mean activity in a pattern vanishes, and study two asymptotic fidelity requirements: constant error probabilities and vanishing error probabilities.
A signal-to-noise ratio analysis is carried out for one retrieval step where the calculations are comparatively straightforward and easy. As performance measures we propose and evaluate information capacities in bits/synapse which also take into account the important property of fault tolerance. For auto-association we compare one-step and fixed-point retrieval that is analyzed in the literature by methods of statistical mechanics.

1.1 Introduction and Overview

With growing experimental insight in the anatomy of the nervous system as well as the first electrophysiological recordings of nerve cells in the first half of this century, a new theoretical field was opened, namely, the modelling of the experimental findings at one or a few nerve cells, leading to very detailed models of biological neurons [1]. But different from most biological phenomena, where the macroscopic function can be understood by revealing the cellular mechanism, the function of the nervous system as a whole turned out to be constituted by the *collective* behaviour of a very large number of nerve cells and the activity of a large fraction of cells, a

[1] Abteilung Neuroinformatik, Fakultät für Informatik, Universität Ulm, Oberer Eselsberg, D-89081 Ulm, Germany
[2] C. und O. Vogt Institut für Hirnforschung, Universität Düsseldorf, Moorenstr. 5, D-40225 Düsseldorf, Germany

whole activity pattern, had to be considered instead.

The modelling had to drop the biological faithfullness at two points: on the cellular level the models had to be simplified such that a large number of nerve cells could be described and on the macroscopic level the function had to be reduced to simple activity pattern processing like pattern completion, pattern recognition or pattern classification allowing a theoretical description and quantification.

McCulloch and Pitts [2] argued that due to the "all or none" character of nervous activity the neurophysiological findings can be reproduced in models with simple two-state neurons, in particular, in *associative memory models* which exhibit binary activity patterns.

In the fiftees and sixtees small feed-forward neural nets have been suggested for simple control tasks, among them the associative memory [3], [4], or the simple perceptron [5]. All these models employ *one-step retrieval* which means that in one parallel update step the initial or input pattern is transformed to the output pattern. Such models which contain no feed-back loops will be the main subject of this paper.

Little, who introduced the Ising-spin analogy of the neural states [3] [6], opened the door to analyzing the feed-back retrieval process in neural nets with methods of statistical mechanics. The analysis which was develloped during the seventies [7] for lattices of coupled spins with randomly distributed interactions to describe spin glasses could be applied successfully to *fixed-point retrieval* in an associative memory [8][4]. In fixed-point retrieval, the retrieval process is iterated until a stable state is reached. This method has been described in several recent books, e. g. van Hemmen and Kühn [9], Amit [10] and Hertz, Krogh, and Palmer [11].

This paper takes as starting point a larger class of simple processing tasks: the association between members of binary pattern sets. Depending on properties of the randomly generated pattern sets we will characterize different memory tasks (Sect. 1) and concentrate on the question how a neural model has to be designed to yield optimal performance.

We consider feed-forward neural associative memory models with one-step retrieval (Sect. 2). To keep our model as variable as possible, Ising-spin symmetry of the neural states is not assumed and arbitrary local learning rules are admitted to form the synaptic connections. One-step retrieval can be analyzed by elementary probability theory and it is compatible

[3] The two states of a binary neuron are identified with up and down states of a spin particle in the Ising model, the synaptic couplings correspond to the spin-spin interactions.

[4] Pattern completion with fixed-point retrieval in a neural net can be treated like relaxation in a solid, once the storage process has determined the dynamics. The macroscopic observables of the system (corresponding to specific heat, conductivity or magnetization in solids) are then the overlaps to stored patterns, or equivalently the recall errors.

with a larger class of memory tasks, not only pattern completion. On the other hand, as we will discuss, in cases of pattern completion a feed-back retrieval model is preferable. Section 3 contains the detailed signal-to-noise ratio analysis where we have included most of the calculations because the intention of this work is to provide not only results but also the methods.

Another important question concerns the judgement of the performance of different memory models. Unfortunately, in the literature a lot of different measures are used. Instead of staying with the mean retrieval errors obtained from the analysis, we apply elementary information theory to the memory process, leading us to the definition of information capacities which allow to compare models with different memory tasks (Sect. 4).

In Sect. 5 we evaluate these performance measures for the various models. The last section resumes the previous sections and points out the relations to the literature. It compares one-step and fixed-point retrieval, taking advantage of the works based on methods of statistical mechanics. The results of the different approaches, which seem to be quite incoherent at first sight, turn out to be not only comparable but also consistent.

1.1.1 Memory and Representation

A memory process can often be considered as a mapping from one set of events into another set of events; as a trivial example one may think of the problem how to establish a phone line to a friend. To solve the problem one has to map the friends name to his phone number. For the construction of a memory device like a phonebook which helps you with this problem one first has to map or to code the events "the friend's name" and the "his phonenumber" into symbols, in this case strings of letters and numbers, which can be written and read by a user. This mapping will be called the *representation* of the events. The memory device has to store these pairs of strings in some way. It can solve the problem if the representation maps the events into unique data strings. Thus a given set of patterns specify the memory task which a memory device has to solve.

Without loss of generality we focus on binary patterns as data strings. A binary pattern is a string containing only two types of elements, for instance "B" and "W" (for black and white pixels). We will restrict ourselves to such pattern sets where every member has approximately the same *ratio p beween the number of "B" and "W" digits*. We call a pattern *distributed*, if both fractions of pixels have more than one member. Throughout this work we distinguish three different *patterns types*:

- A *singular pattern* has only one "B" digit out of $m-1$ "W" digits, if m is the number of digits in the pattern. A singular pattern is by definition not distributed.
- A *sparse pattern* is distributed but the ratio p between the number of "B" and "W" digits satisfies $p << 0.5$. In the infinite model $m \to \infty$

we will consider the *sparse limit*: $p \to 0$ with $mp \to \infty$ which leads to nontrivial distributed patterns.

- In a *nonsparse pattern* the fraction p between the number of "B" and "W" digits has to be away from zero. In the infinite model: $p = \text{const}$ as $m \to \infty$.

1.1.2 Retrieval from the Memory

The memory device has to store a set of patterns in such a way that a desired pattern can be selectively recalled at the output port. In the *memory retrieval* a desired output pattern is selected by applying a pattern at the input port of the device. We will denote the set of output patterns the *content patterns* $\mathcal{S}^C$.

An input pattern which selects a content pattern will be called its *address pattern* or simply its *address*. The set of address patterns will be denoted with $\mathcal{S}^A$. Thus in the retrieval the memory device has to map from an address pattern to its corresponding content pattern. This map is defined by the set of pairs consisting of address and content pattern:

$$\{(x^1, y^1), ..., (x^M, y^M) : x^k \in \mathcal{S}^A, y^k \in \mathcal{S}^C\}.$$

1.1.3 Fault Tolerance in Addressing

Between two patterns x and $\hat{x}$ the number of different bits $h(x, \hat{x})$ defines a natural distance relation called the *Hamming distance*. Via this distance a whole set of input patterns may specify one desired content pattern uniquely: all patterns $\hat{x}$ with the property $h(\hat{x}, x) < h(\hat{x}, x^k)$ for all $x^k \neq x$ and $x, x^k \in \mathcal{S}^A$. We call a memory retrieval *fault tolerant* if it allows input noise in the sense that many input patterns which have a unique closest address are mapped on the content pattern belonging to this address.

For a set of singular address patterns normally no $\hat{x} \notin \mathcal{S}^A$ has a unique closest address and therefore, fault tolerant retrieval is impossible. Thus fault tolerant retrieval can only be expected, if the address patterns are distributed.

1.1.4 Various Memory Tasks

We call *hetero-association* the general memory task where the set of address patterns $\mathcal{S}^A$ and the set of content patterns $\mathcal{S}^C$ can be chosen arbitrarly.

Below the following special cases of hetero-association will be considered:

- If the address patterns are singular patterns, the memory task is called the *look-up-table task*. Then the singular pixel of an address pattern points into a table of content patterns like the usual access in a look-up-table.

- For singular content patterns we can identify each bit of the content pattern with a class in the set of address patterns. This memory task can be interpreted as *pattern classification* which separates the set of address patterns in disjunct classes. This task (with one-bit content patterns) has been executed by the classical simple perceptron models; see [5].

- *Auto-association* is the case of hetero-association where address and content pattern are identical, therefore also denoted as content addressability. Only for fault-tolerant retrieval the auto-association task makes sense; then the memory performs *pattern completion* from a distorted version $\hat{x}^k$ as input pattern to the errorfree content pattern x^k; see also Forrest and Wallace in [9].

1.1.5 Retrieval Errors

A memory which allows errors in the addressing will perhaps also recall erroneously the wrong content pattern or put at least some errors in the output.

In the retrieval of binary patterns there may occur two types of flip errors in a digit of the output pattern $\hat{y}^k$: A "W" of the content pattern y^k may be turned to a "B" and a "B" in the content pattern y^k may be turned to an "W". Of course, with increasing addressing noise these errors will also increase. But again via the distance relation it is possible that a memory output containing errors in some digits perhaps still specifies the event coded by the original content pattern. A given memory task together with the sets $\mathcal{S}^A$ and $\mathcal{S}^C$ will fix the maximal mean errors which can be tolerated in the retrieval. These upper bounds, which have to be satisfied by the error probabilities, will be called the *fidelity requirement*.

1.2 Neural Associative Memory Models

The typical ingredients of an artificial neural network model are a large number of similiar processor units called *neurons*, which obtain signals through adjustable connections from a large number of input fibres and/or other neurons. In this model the adjustable connections, the *synapses*, connect an input port to each neuron.

The two different types of calculation in the model, the processing of the neural input signal in the retrieval on the one hand and the synaptic adjustment according to the data in the storage phase on the other hand, are separated in time in this model; we distinguish the *storage process* and the *retrieval process*.

To perform the calculations the pixel types "B" and "W" in the input patterns have to be translated into signals which can propagate through

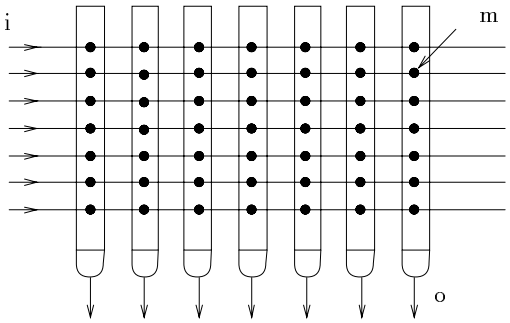

FIGURE 1.1. Schematic view of a neural associative memory. i: retrieval input fibres, o: retrieval output fibres (axons), m: modifiable synaptic connection between neuron and input fibre. The horizontal lines are wires which propagate the input signals to the synapses. Each column represents one neuron. The larger upper section where the synaptic connections access corresponds to the dendritic tree and the lower section the cell body. The arrow pointing below from the cell body corresponds to the axon.

the network. We assign two different values "1" and "a" to the pixel types "B" and "W"; each pattern is identified with an n-vector $x \in \{a,1\}^n$ with $a \in [-1,0]$, we will use synonymously the expressions pattern and $\{a,1\}$ vector. Of course, we are free to exchange "W" and "B" in the assignment; the flip transformation $\mathcal{F}$ applied to all components in the data will not change the memory problem. Here $\mathcal{F}(x_i = W) := B$ and $\mathcal{F}(x_i = B) := W$. Therefore we can always assign the value 1 to the smaller pixel fraction so that
$p = \sharp\{i : x_i = 1\}/(n - \sharp\{i : x_i = 1\}) \leq 0.5$.

Such models have already been proposed and analyzed many years ago; e.g., Uttley [12], Steinbuch [3], Rosenblatt [5], Longuett-Higgins et al [13], Amari [14], Gardner-Medwin [15] and Kohonen [16].

1.2.1 Retrieval Process

In the retrieval phase an address pattern is applied to the input port of the memory. The input signals are propagated via a synaptic connection strenghs matrix $\mathcal{M}_{ij}$ to all neurons. In *one-step retrieval* every neuron j actualizes its state, the *axonal activity* $\tilde{y}_j$, according to this input and the vector $\tilde{y}$ is the retrieval output pattern.

Each neuron has to form the *dendritic potential* d_j, the sum over all its incoming activities

$$d_j := \sum_i \mathcal{M}_{ij} x_i \tag{1.1}$$

and then to determine the new activity value in the neural update equation

$$\tilde{y}_j = f(d_j - \Theta). \tag{1.2}$$

The output signal of a biological neuron are trains of short electric pulses, the neural spikes. It is the *spike rate* and not the amplitude or the duration of a spike which is growing with increasing dendritic potential. This properties have been modelled in the so called *spike coding models*; cf [17, 18, 19, 20]. Here we focus on *rate coding models* where the *neural transferfunction* $f(x)$ describes only the spike rate. In almost all of these models $f(x)$ is a monotonously increasing function. Θ is the *threshold value* which can be adjusted globally for all neurons in each retrieval step.

Models with linear transfer function, as for instance proposed in Kohonen [16] or Anderson [21, 22], lead for large networks to quasi continous valued output patterns.

Binary output patterns are obtained, if the neural transfer function is a two-valued stepfunction: $f(x) = 1$ for $x \geq 0, f(x) = a$ otherwise. The neural state $\tilde{y}_j = 1$ is called *firing* or *active*, $\tilde{y}_j = 0$ *silent* or *passive*. The *retrieval error probabilities* for *on errors* and *off errors* respectively are expressed by conditioned probabilities

$$e_1 := \text{Prob}[\tilde{y}_j^k = a \mid y_j^k = 1] \ , \ e_a := \text{Prob}[\tilde{y}_j^k = 1 \mid y_j^k = a]. \tag{1.3}$$

Such models have been treated from Willshaw et al. [4], Palm [23] and Nadal and Toulouse [24]. In one-step retrieval the output pattern is evaluated from the input pattern after one synchronious parallel calculation of all neurons.

Step-shaped neural transfer functions have also been used in the spin-glass literature on auto-association, e.g. in [25, 8, 26, 27]. These works consider an *iterative retrieval procedure*, where via a feed-back loop the signal flow through the system is iterated until a stationary state, a fixed point, is reached. Such *fixed point retrieval* has been considered for two different ways performing the iteration. In models with parallel update the complete one-step retrieval process is iterated in the manner that the output is fed back as new input; for instance in [6, 15, 28, 29, 30, 31]. In models with sequential random update only one neuron, randomly selected, is updated (1.2) in one iteration step, leading to the new input, which only deviates in one component from the preceding one; see again [25, 8, 26, 27].

The improvement due to iterated retrieval for the pattern completion task obtained in simulations can be observed in Fig. 1.9.

1.2.2 Storage Process

In this process, which is also called the learning process, the synaptic matrix, the storage medium, is formed from the set of patterns to be stored.

During the storage process each pair (x^l, y^l) of patterns to be learned is applied at the in- and output port of the memory. This provides a pre- and postsynaptic value for every synapse $\mathcal{M}_{ij}$.

Learning Rules

For a given pair (x, y) of pre- and postsynaptic activity values the *local synaptic rule* $R(x, y)$ determines explicitely the amount of synaptic connectivity change. For binary patterns there are only four different constellations possible for pre- and postsynaptic activities, viz., $(a, a), (1, a), (a, 1)$, and $(1, 1)$. Thus a synaptic rule is determined by four numbers

$$R = (r_1, r_2, r_3, r_4). \tag{1.4}$$

The following two famous local learning rules will be focused in the subsequent analysis:

- The *Hebb rule* or asymmetrical coincidence rule $H := (0, 0, 0, 1)$ increases the synaptic matrix element for coinciding pre- and postsynaptic firing only. In his 'neurophysiological postulate' Hebb [32] proposed this type of synaptic modification between pairs of firing nervous cells.
- The *agreement rule*, Hopfield rule or symmetrical coincidence rule $A := (1, -1, -1, 1)$ increases the synaptic matrix element for agreeing pre- and postsynaptic states and decreases the synaptic weight for disagreeing states. This rule was used in the original Hopfield model [25].

The above rules are both product rules: $R(x, y) = xy$. For $a = 0$ we obtain the Hebb rule and for $a = -1$ the agreement rule and sometimes, for instance in [33], both are considered as Hebbian learning. We retained the distinction because in the original formulation of his postulate Hebb clearly talks of the influence of synchronously firing neurons on their interconnecting synapse. The psychologist Hebb claimed this postulate to be inspired by physiological and psychological findings while the symmetry between firing and silence in the agreement rule is biologically very implausible.

Storage Procedures

We consider *one-step learning* which means that after one single presentation of every pair the formation of the *synaptic matrix* is finished. Two different types of storage procedures will be examined:

- The *incremental storing procedure*, where the synaptic matrix is given by

$$\mathcal{M} = (\mathcal{M}_{ij}) := \sum_{k=1}^{M} R(x_i^k, y_j^k) \tag{1.5}$$

- The *binary storage procedure*, where the synaptic matrix $\bar{\mathcal{M}}$ is obtained from $\mathcal{M}$ by another highly non-linear operation:

$$\bar{\mathcal{M}}_{ij} := \operatorname{sgn}(\mathcal{M}_{ij}) \tag{1.6}$$

with $\mathrm{sgn}(0) := 0$.

Storage procedures can be strictly local (as in most of the papers cited here) or non-local (as for example in Personnaz et al [34, 35]). Depending on the sign of the average connectivity change, they can be productive, destructive or balancing for the total network connectivity (cf. [36, 37]). Local storage procedures can make use of two (probably the majority) , three (supervised learning with additional teacher signal, e.g. Barto et al [38]) or more terms to compute a synaptic change (compare Palm [36] again). In this paper we concentrate on storage procedures employing strictly local two-term learning rules.

The most common synaptic arrangement in biological neural nets as in the cerebral cortex (and the hippocampus) is the simple dyadic synapse. It connects just two neurons, the presynaptic and the postsynaptic one. Therefore there are just two natural, locally available activity signals: the presynaptic and the postsynaptic activity.

1.2.3 Distributed Storage

One reason of the big come back of systems with neural architecture in the last decade is the fact that in computer science distributed processing turned out more and more to be an indispensible goal. How does the simple memory models introduced in this section display the property of distributed storage ?

For hetero-association local rules store second order correlations between address and content pattern activity; for instance with the Hebb rule each pair of active neurons (x_i^k, y_j^k) affects one synapse $\mathcal{M}_{ij}$.

The storage is called distributed, if the storage of one single pattern pair causes nonlocal changes in the storage medium. More than one element of the synaptic matrix is affected if at least one pattern in the pair is nonsingular, if either the set of address or content patterns contain nonsingular patterns.

Here we define distributed storage in a stricter sense: we require that many matrix elements carry information about more than one pattern pair. In this sense distributed information storage for arbitrary local rules is provided only if both pattern sets, address and content patterns, contain nonsingular and overlapping patterns. Then storage of several pattern pairs will affect the same synapses, so that each entry in the synaptic connectivity matrix $\mathcal{M}$ may contain the superposition of several memory traces, i.e., for most index pairs (i, j) the sum $\sum_k R(x_i^k, y_j^k)$ should have more than one nonzero contribution. Like in holography an accessible content segment (a pattern pair) is written widely spread in the storage medium and different content segments will overlap.

In the case of auto-association local rules store the second-order autocorrelation of the pattern activity; with the Hebb rule each pair of active

neurons in a learning pattern causes a change in one synapse. Distributed storage requires the patterns to be nonsingular and overlapping.

1.3 Analysis of the Retrieval Process

The aim of the present section is the analysis of one-step retrieval in the associative memory after learning, i.e., after the storage process has formed the memory matrix for a given memory task $(\mathcal{S}^A, \mathcal{S}^C)$. In Sect. 1.1.5 and by eq. (1.3) we have introduced the quantities of interest in the analysis of this feed-forward system, viz., the mean retrieval error probabilities in an output pattern for a given input pattern.

We already mentioned in the introduction that different spatial scales can be distinguished in the treatment of neural nets, the microscopic scale of synapses and model neurons and the macroscopic scale of the collective behaviour of all neurons. What we presume about the model is on the microscopic scale (neuron model, learning rules etc.), what we would like to know from a theory is on the macroscopic scale, the collective behaviour of the *whole* set of neurons (retrieval errors). In physics it is quite usual to deal with separable scales, for instance in thermodynamics the nuclear versus the macroscopic scale. Physical mean-field theories which originally have been developed for spin-glasses[5] yield asymptotic results for the retrieval errors[6] in the limit of infinite system size: $m, n \to \infty$ which is often called the *thermodynamic limit* of fixed-point retrieval in the associative memory after learning.

We will consider memory tasks with different mean ratios p between the elements 1 and a in the pattern sets in the finite model and in the thermodynamic limit, i.e., $m \to \infty$. Curiously memory tasks with sparse patterns, as defined in Sect. 1.1.1, will turn out to yield optimal asymptotic performance.

---

[5] Spin-glasses are magnetic solids with two different competing fractions of spin couplings. One fraction favors parallel, the other fraction anti-parallel spin alignment which cause irregular (glass-like) stable spin configurations. The mean-field theory provides values for the mean magnetization as macroscopic order parameter.

[6] The order parameters of a mean-field theory treating neural networks are the M *overlaps* $\{m_l, l = 1, ..., M\}$ where each overlap m_l is defined as the number of common pixels between retrieval output and the content pattern y^l. If we apply a (distorted) address pattern $\tilde{x}^k$ as input pattern, particularly one overlap is important for the retrieval quality, namely the overlap m_k corresponding to the input pattern. The theory provides a mean value $< m_k >$, averaged over a large number of retrieval events which is equivalent to the retrieval error probabilities of Sect. 1.5.

1.3.1 Random Pattern Generation

To apply probability theory for the estimation of mean retrieval error probabilities we have to assume the following properties of the the memory data and of the distortion of the input patterns.

Content and Address Patterns

In the memory tasks we assume the simplest model of the data to be stored, namely sets of randomly generated patterns. The value of each of the n digits in a pattern $x^k \in \mathcal{S}$ is chosen independently with the probability: $p := \text{Prob}[x_i^k = 1]$. A set of randomly generated patterns is fixed by three parameters, the probability p, the dimensionality of a pattern n and the number of patterns M. We will use the following notation for address and content patterns
$\mathcal{S}^A := \mathcal{S}(p, m, M), \mathcal{S}^C := \mathcal{S}(q, n, M)$. For hetero-association the sets $\mathcal{S}^A$ and $\mathcal{S}^C$ will be generated mutually independently.

Input Patterns

The signal detection problem will be treated in three different cases of addressing:

a) A perfect address pattern as input pattern x^k with $n_1 := \sharp\{i : x_i^k = 1\}$ being the number of 1 components.

b) An *ensemble of perfect input patterns*, where now the number of ones in the input pattern n_1 becomes a random variable too. It is a binomially distributed variable and for large m the fraction n_1/m will be close to its expectation value p because of the strong law of large numbers [39]. In the analysis the *average input activity* μ of the ensemble will become an important quantity which, for large m, equals

$$\mu := [n_1 + (m - n_1)a]/m = p + (1 - p)a. \tag{1.7}$$

c) An *ensemble of noisy input patterns* $\hat{\mathcal{S}}^A$, which is generated by a second random generation process from the set of address patterns $\mathcal{S}^A$ used for learning. Here we concentrate on noisy input patterns, where $\hat{x}^k \in \hat{\mathcal{S}}^A$ is a "part" of an address pattern x^k in the following sense: $\text{Prob}\left[\hat{x}_i^k = 0 \mid x_i^k = 0\right] = 1$ and $\text{Prob}\left[\hat{x}_i^k = 1 \mid x_i^k = 1\right] =: p'$. As for the faultless ensemble we describe the input activity for large m by the average input activity of the address ensemble

$$\mu' := pp' + (1 - pp')a \tag{1.8}$$

In the analysis below we will use the prime to indicate the results for the noisy input ensemble.

1.3.2 Site Averaging and Threshold setting

Depending on its dendritic potential (1.1) and the threshold value Θ_j each neuron j "decides" in the update process (1.2) whether it should be active or silent. This can be regarded as a signal detection problem on the random variable d_j which every neuron has to solve.

To find the probabilities for on and off errors in eq. (1.3) we have to consider the neurons seperated in two fractions; the *on-neurons* which should be active in the original content pattern y^k and the *off-neurons* which should not be active. In our model the threshold of each neuron is set to the same value depending only on the total activity of the input pattern. Therefore, it is sufficient to analyze the averaged dendritic potentials in each of the fractions. We will use the notation $d^1 =< d_j >_{j \in \{j : y_j^k = 1\}}$ and $d^a =< d_j >_{j \in \{j : y_j^k = a\}}$. With the assumptions of the last subsection these averaged quantities can be treated as random variables.

Of course, the synapses – randomly generated in the storage process – are "quenched" in the retrieval so that dendritic potentials at different on-sites or off-sites will behave differently. This suggests a memory model where the threshold is adjusted seperately for each neuron, which has been treated in [49] and will be discussed in Sect. 1.6.3.

1.3.3 Binary Storage Procedure

For binary storage, the dendritic potential at neuron j is: $d_j = \sum_i x_i^k \bar{\mathcal{M}}_{ij}$, where the values of the binary Hebb matrix $\bar{\mathcal{M}}$ are distributed on $\{0, 1\}$. The probability that a matrix element is zero can be easily calculated

$$p_0 := \text{Prob}[\mathcal{M}_{ij} = 0] = (1 - pq)^M . \tag{1.9}$$

We discuss the three cases of addressing a) to c) from Sect. 1.3.1 seperately.

a) Given x^k as input pattern the expectation $E(d^1 - d^a) = n_1(1 - p_0)$ is independent of the value a but the variance $\sigma^2(d_j)$ is minimal for $a = 0$. So, optimally we choose $a = 0$. Then we obtain for the dendritic potential at an on neuron $d^1 = n_1$. Thus we maximally can put $\Theta = n_1$ to obtain $e_1 = 0$.

The second error probability is determined from the dendritic potential at an off neuron

$$e_a = \text{Prob}[d^a > \Theta] = \text{Prob}[\prod_{i \in \{i : x_i^k = 1\}} \mathcal{M}_{ij} = 1 \mid y_j^k = 0] \simeq (1 - p_0)^{n_1} . \tag{1.10}$$

b) If we average over an ensemble of perfect patterns, where we adjust the threshold individually for each input to $\Theta = n_1$, then the threshold becomes a random variable too. Now consider the fixed threshold setting $\Theta = En_1$ for all input patterns. For this threshold choice we simply have

to insert the expectation of n_1 into (1.10)

$$e_a \simeq (1-p_0)^{mp}. \tag{1.11}$$

This fixed threshold setting leads to $e_1(E\Theta) > 0$ because of patterns with $n_1 < En_1$ and to $e_a(E\Theta) < Ee_a(\Theta)$ because of the concavity of the function $e_a(\Theta)$. We will use (1.11) as approximation for the retrieval error e_a with the individual threshold adjustment.

c) Finally for noisy addressing we obtain for the same fixed threshold setting $\Theta = p'E(n_1)$

$$e'_{a1} = (e_a)^{p'}. \tag{1.12}$$

Strictly speaking, the above calculation requires independence of the entries $\bar{\mathcal{M}}_{ij}$. Although this is not the case it is shown Appendix 1 that at least for sparse address patterns with $m^{2/3}p \to 0$ the entries $\bar{\mathcal{M}}_{ij}$ become asymptotically independent for large m.

1.3.4 Incremental Storage Procedure

In incremental storage the contribution of each pattern pair is simply summed up in the synaptic weights and we can devide the dendritic potential in two parts: the signal part s, which is the partial sum coming from the storage of the pattern pair (x^k, y^k) and the noise part N, the remaining partial sum which contains no information about y_j^k. From equations (1.1) and (1.5) we obtain

$$\begin{aligned} d_j &= N + s := \sum_i x_i^k \mathcal{M}_{ij} = \sum_i \sum_l x_i^k R(x_i^l, y_j^l) \\ &= \sum_i \sum_l x_i^k R(x_i^l, y_j^l) + \sum_i x_i^k R(x_i^k, y_j^k). \end{aligned}$$

The dendritic potential and its signal part has to be regarded separately at an on neuron ($y_j^k = 1$) and at an off neuron ($y_j^k = a$):

$$s_1 := \sum_i x_i^k R(x_i^k, 1) \quad , \quad s_a := \sum_i x_i^k R(x_i^k, a).$$

We now assume that for the noise parts $E(N_1) = E(N_a)$ holds and that it is the *variance of the noise* $\sigma(N)$, which determines the mean facility to solve the neural detection problem. Inspired by engeneering methods we introduce the *signal-to-noise ratio* as a threshold setting independent retrieval quality measure

$$r := E(s_1 - s_a)/\sigma(N). \tag{1.13}$$

The motivation to do so is quite intuitive: the threshold detection problem can be solved for a lot of neurons for the same value Θ if $E(s_1 - s_a)$ is large and $\sigma(N)$ is low.

The fidelity requirement that e_a and e_1 should be small is equivalent to the corresponding requirement that the signal-to-noise ratio r should be large. How the retrieval errors are balanced between the two possible types of retrieval errors is gouverned by the threshold setting. If both retrieval error probabilities have to be below 0.5, the threshold has to satisfy $Ed^a \leq \Theta \leq Ed^1$, Ed^a being the expectation of the dendritic potential at an off site.
Thus we put $\Theta = Ed^a + \vartheta\sigma(N)r = Ed^1 - (1-\vartheta)\sigma(N)r$ with $\vartheta \in [0.1]$.

For large m the noise term N can be considered as sum of a large number of independent random variables and the central limit theorem holds. Then we can estimate the error probabilities using a normal distribution and get

$$e_1 = \text{Prob}[d^1 - \Theta < 0] \simeq G\left[-E(d^1-\Theta)/\sigma(N)\right] = G\left[-(1-\vartheta)r\right] \quad (1.14)$$

$$e_a = \text{Prob}[d^a - \Theta > 0] \simeq G\left[-\vartheta r\right] \quad (1.15)$$

with the normal or Gaussian distribution $G[x] := (1/\sqrt{2\pi}) \int_{-\infty}^{x} e^{-x^2/2} dx$.

To obtain explicit values for the error probabilities we now have to analyze the signal and noise term in (1.13) for the different ensembles of input patterns and different learning rules (Section 2.2).

For input ensembles we are interested in the mean retrieval errors where for every input the threshold has been set in the optimal way according to the number of active input digits n_1. We insert the signal-to-noise ratio averaged over an input ensemble into (1.14) and consider a fixed threshold setting which is equal for all input patterns. As to binary storage we take this result as an approximation for the individual threshold adjustment which is equivalent to an exchange of the expectations of the pattern average and the input average in the calculation.

Signal-To-Noise Calculation

Again we discern the three cases of addressing described in Sect. 1.3.1.

a) For the faultless address x^k as input the signal is sharply determined as

$$s_1 - s_a = n_1(r_4 - r_3) - (m - n_1)a(r_2 - r_1).$$

The noise decouples into a sum of $(M-1)$ independent contributions corresponding to the storage of the pattern pairs (x^l, y^l) with $l \neq k$. For every pair the input x^k generates a sum of n_1 random variables $R(x,y)$ and of $(m-n_1)$ random variables $aR(x,y)$ at a neuron j. The variable $R(x,y) = R(x_i^l, y_j^l)$ is the four-valued discrete random variable (1.4) with the distribution: $(1-p)(1-q), p(1-q), (1-p)q, pq$.

With $E(R)$ and $\sigma^2(R)$ denoting expectation and variance of $R(x,y)$ a simple (but for $\sigma^2(N)$ tedious) calculation yields

$$E(N) = (M-1)\left[n_1 + (m-n_1)a\right] E(R) \quad (1.16)$$

$$\sigma^2(N) = (M-1)\left\{Q_1\sigma^2(R) + Q_2\,\text{Cov}[R_i R_h]\right\} \quad (1.17)$$

where we have used the abrevations

$$\begin{aligned} Q_1 &:= n_1 + (m - n_1)a^2 \\ Q_2 &:= n_1(n_1 - 1) + 2an_1(m - n_1) + a^2(m - n_1)(m - n_1 - 1) \\ \mathrm{Cov}[R_i R_h] &= q(1-q)\left[p(r_4 - r_3) + (1-p)(r_2 - r_1)\right]^2 \end{aligned}$$

The covariance term: $\mathrm{Cov}[R_i R_h] := \mathrm{Cov}\left[R(x_i^l, y_j^l) R(x_h^l, y_j^l)\right]$ measures the dependency between two contributions in the i-th and h-th place of the column j upon the synaptic matrix.

b) If we average over the ensemble of perfect input patterns we can use again for large m the approximations $n_1/m \simeq (n_1 - 1)/m \simeq (n_1 + 1)/m \simeq p$ and
$(M-1)/m \simeq M/m$ and obtain

$$\begin{aligned} E(s_1 - s_a) &= m\left[p(r_4 - r_3) - (1-p)a(r_2 - r_1)\right] \\ E(N) &= (M-1)m\mu E(R) \end{aligned} \tag{1.18}$$

In equation (1.17) we have to insert

$$Q_1 = m\left[p + (1-p)a^2\right] \quad , \quad Q_2 = m^2\mu^2. \tag{1.19}$$

c) Finally, we consider the ensemble of noisy address patterns. In this case

$$E(s_1' - s_a') = m\left[p(p' + (1-p')a)(r_4 - r_3) - (1-p)a(r_2 - r_1)\right]. \tag{1.20}$$

In the description of the noise we only have to replace in (1.18) and (1.19) p by pp' and μ by μ'.

Signal-to-noise Ratios for Explicit Learning Rules

Regarding (1.17) and (1.18) we observe that the signal-to-noise ratio is the same for the rules R and $bR + c$, where c is an arbitrary and b is a positive number. Two rules that differ only in this way, will be called *essentially identical*. Thus we may denote any rule R as

$$R = (0, r_2, r_3, r_4). \tag{1.21}$$

The following formulae are written more concisely if we introduce instead of r_2, r_3, r_4 the mutually dependent parameters

$$\gamma := r_4 - r_3 - r_2 \quad , \quad \kappa := r_2 + \gamma p \quad , \quad \eta := r_3 + \gamma q.$$

In this notation the variance of the rule becomes

$$\begin{aligned} \sigma^2(R) : &= E(R^2) - (E(R))^2 \\ &= \eta^2 p(1-p) + \kappa^2 q(1-q) + \gamma^2 p(1-p)q(1-q) \end{aligned}$$

In the description of the input ensemble we transform from the parameters p, a to the quantities p, μ, see (1.7).

The signal-to-noise ratio averaged over perfect address patterns b) is then obtained from equation (1.13) as

$$r^2 = (m/M)\frac{\left[\mu\kappa + (1-\mu)p\gamma\right]^2}{\left[p + (\mu-p)^2/(1-p)\right]\sigma^2(R) + mq(1-q)\mu^2\kappa^2}. \qquad (1.22)$$

Averaged over noisy address patterns c) we obtain equivalently

$$r'^2 = (m/M)\frac{\left[\mu'\kappa + (1-\mu)pp'\gamma\right]^2}{\left[pp' + (\mu'-pp')^2/(1-pp')\right]\sigma^2(R) + mq(1-q)\mu'^2\kappa^2} \qquad (1.23)$$

with the definition for μ' taken from (1.8).

Optimal Learning Rule

The expression (1.22) invites to optimize the signal-to-noise ratio in terms of the three parameters γ, κ and η so as to yield the *optimal learning rule* R_0.

The parameter η appears only in $\sigma^2(R)$ in the denominator. We first minimize $\sigma^2(R)$ with $\eta = 0$ and obtain

$$r^2 = (\frac{m}{M})\frac{\left[\mu\kappa + (1-\mu)p\gamma\right]^2}{q(1-q)\left\{\left[p + (\mu-p)^2/(1-p)\right]\left[\kappa^2+\gamma^2 p(1-p)\right] + m\mu^2\kappa^2\right\}}. \qquad (1.24)$$

The (large) factor m in the second term of the denominator in eq. (1.24) makes this term dominating unless at least one of the other factors κ or μ vanishes.

At a first sight we have to distinct two cases which differ with respect to the average activity μ of the input patterns:

- Either μ stays away from zero, then it is optimal to choose $\kappa = 0$ (case 1).
- Or $\mu \to 0$ fast enough to make the second term negligible in the sum of the denominator in eq. (1.24). However, if we insert $\mu = 0$ in (1.24), again $\kappa = 0$ turns out to be the optimal choice (case 2).

Thus both cases leave us with $\kappa = 0$ and $\eta = 0$ and yield the *covariance rule* as general optimal rule

$$R_0 = (pq, -p(1-q), -q(1-p), (1-p)(1-q)). \qquad (1.25)$$

The condition $\mu = 0$ will occur several times in the sequel, and will be referred to as the condition of *zero average input* activity. In particular, for $p = 0.5$ it implies $a = -1$ and for $p \to 0$ this implies $a \to 0$. This condition,

which is equivalent to $a = -p/(1-p)$ or to $p = -a/(1-a)$ fixes the optimal combination between input activity and the model parameter a.

For arbitrary p and a in the input patterns and for arbitrary μ, the optimal signal-to-noise ratio is evaluated by inserting R_0 in eq. (1.24),

$$r_0^2 = (m/M)\frac{(1-\mu)^2 p}{q(1-q)\,[p + (\mu-p)^2/(1-p)]\,(1-p)}. \tag{1.26}$$

Transforming back from μ to a we obtain

$$r_0^2 = (m/M)\frac{p(1-p)(1-a)^2}{[p + (1-p)a^2]\,q(1-q)}. \tag{1.27}$$

Insertion of the zero average input condition $\mu = 0$ in (1.26) yields the optimal signal-to-noise ratio

$$r_0^2 \simeq \frac{m}{Mq(1-q)}. \tag{1.28}$$

Optimizing the signal-to-noise ratio for noisy addresses c), eq. (1.23) leads to the same optimal rule (1.25). Then the signal-to-noise ratio value for perfect addressing is reduced from the noise in the input patterns. For the optimal rule R_0 with $\mu = 0$ it is given by

$$r_0'^2 \simeq \frac{(1-p)p'^2}{p' - 2pp' + p} r_0^2. \tag{1.29}$$

For learning rules with $\kappa \neq 0$ which have a nonzero covariance term only $\mu = 0$ can supress the m^2 term in the variance of the noise. Therefore, $\kappa \neq 0$ and $\mu \neq 0$ lead to vanishing r as $m \to \infty$. A little algebra shows that learning rules with $\mu \neq 0$ and finite γ also yield a vanishing r. In conclusion all suboptimal rules need $\mu = 0$ to achieve a nonvanishing r.

Hebb and Agreement Rule

If we compare the Hebb rule and the agreement rule to the optimal learning rule R_0 we realize, that in general both rules are suboptimal. But nevertheless, for $p = q = 0.5$ the optimal rule becomes equal to the agreement rule: $R_0 = (0.25, -0.25, -0.25, 0.25)$ and for $p, q \to 0$ the Hebb rule is approximated by the optimal rule: $R_0 \to H$.

By equation (1.22) one can compute the signal-to-noise ratio for these rules, the results for $\mu = 0$ you find in Table 1.

As expected, the Hebb rule becomes essentially identical to R_0 for $p, q \to 0$. In the $a = 0$ model, where the parameter a is not adjusted to guarantee $\mu = 0$ we need a stricter sparseness in the address patterns: $mp^2 \to 0$ to provide $\mu \to 0$ fast enough to preserve the essential identity between H and R_0.

By comparing the r^2 values corresponding to the different rules in Table 1 we will derive the performance analysis of Hebb and agreement rule (see Section 5.2 and 5.4) from the analysis of R_0 carried out in this section.

TABLE 1.1. Squared signal-to-noise ratios $r^2(m, M, p, q)$ for $\mu = 0$.

	Optimal rule R_0	Hebb rule H	Agreement rule C
$r^2 =$	$\frac{m}{Mq(1-q)}$	$\frac{m(1-p)}{Mq(1-pq)}$	$\frac{8mp(1-p)}{M\,[p(1-q)+(1-p)q]}$

Summary

With incremental storage procedure the signal-to-noise ratio analysis of one-step threshold-detection retrieval led to the following results:

- If a rule R yields the signal-to-noise ratio r then any rule $bR+c$, with b positive yields the same signal-to-noise ratio. We call these rules *essentially identical*.
- For any rule R the best combination of the parameters p and a is given by the *zero average input condition* $\mu = p + (1-p)a = 0$.
- The maximal signal-to-noise ratio r_0 is always achieved for the covariance rule R_0 (1.25). For increasing μ the value r_0 continously decreases and reaches $r_0 = 0$ at $\mu = 1$.
- Every rule essentially different from R_0 has zero asymptotic signal-to-noise ratio, if the condition $\mu = 0$ is violated.
- The Hebb rule becomes essentially identical to R_0 for memory tasks with $q \to 0$ and $p \to 0$, i.e., for sparse address and content patterns.
- The agreement rule is equal to R_0 for $p = q = 0.5$.
- *Storage of extensively many patterns, i.e., $M/m > 0$ as $m \to \infty$*: In this case R_0 and H achieve asymptotically vanishing errors ($r \to \infty$) for memory tasks with sparse content patterns: $q \to 0$ as $m \to \infty$. The agreement rule A only achieves $r = \text{const}$ as $m \to \infty$.

1.4 Information Theory of the Memory Process

How can the performance of an associative memory model be measured ?

In our notation a given memory task specifies the parameters: p, q, M, p', e_a, e_1. From the signal-to-noise ratio analysis we can determine for randomly generated patterns the maximal number of pattern pairs M^*, for which the required error bounds e_a, e_1 are still satisfied. Then the first idea is to compare the M^* to the number of neurons used in the memory model. This quotient of patterns per neuron $\alpha = M^*/n$ is used in a lot of works

but this measure disregards the parameter q used in the random generation of the content patterns as well as the whole process of addressing.

In the following we use the description of elementary information theory to find performance measures for the memory task and compare them with the size of the storage medium, viz., the number of synaptic connections $n \times m$.

1.4.1 Mean Information Content of Data

Every combination of a memory problem and a coding algorithm will lead to a set of content patterns which exhibit in general very complicated statistical correlations.

For a set of *randomly generated patterns* $\mathcal{S}$ which we have used to carry out the signal-to-noise ratio analysis each digit was chosen independently. The mean information contained in one digit of a pattern is then simply given by the Shannon information [40] for the two alternatives with the probabilities p and $1-p$

$$i(p) := -p \log_2 p - (1-p) \log_2 (1-p)$$

and the *mean information content* in the set of randomly generated content patterns S^C is $I(\mathcal{S}^C) = Mn\ i(q)$ where q is the ratio between 1- and a-components in each content pattern. The *pattern capacity* compares the mean information content of the content patterns with the actual size $m \times n$ of the storage medium and is defined as

$$P(m,n) := \max_M \{I(\mathcal{S}^C)\}/nm = M^* i(q)/m. \tag{1.30}$$

Here M^* equals the maximum number of stored patterns under a given retrieval quality criterion. The definition (1.30) is an adequate measure of how much information can be put in the memory but not at all of how much can be *extracted* during the retrieval. A performance measure should also consider the information loss due to the retrieval errors.

1.4.2 Association Capacity

The memory can be regarded as noisy information channel consisting of two components (see Fig. 2): The channel input is the set of content patterns $\mathcal{S}^C$ and the channel output is the set of recalled content patterns $\tilde{\mathcal{S}}^C$ afflicted with the retrieval errors. The two components correspond to the *storage process* where the sets $\mathcal{S}^A$ and $\mathcal{S}^C$ are transformed into the synaptic matrix and to the retrieval process where the matrix is transformed into a set of memory output patterns $\tilde{\mathcal{S}}^C$. The retrieval error probabilities specify the deviation of $\tilde{\mathcal{S}}^C$ from $\mathcal{S}^C$ and thus the channel capacity.

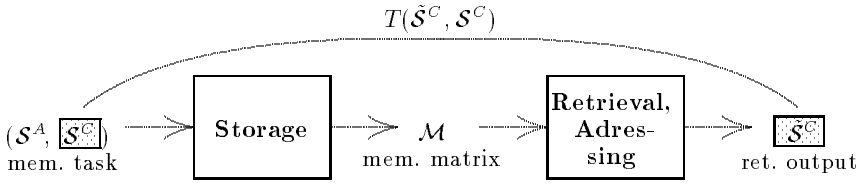

FIGURE 1.2. Output capacity: Information channel of storage and retrieval. (The abbrevations "mem." for memory and "ret." for retrieval have been used.)

The capacity of an information channel is defined as the transinformation that is contained in the output of the channel about the channel's input. The transinformation between $\tilde{\mathcal{S}}^C$ and $\mathcal{S}^C$ can be written

$$T(\tilde{\mathcal{S}}^C, \mathcal{S}^C) = I(\mathcal{S}^C) - I(\mathcal{S}^C \mid \tilde{\mathcal{S}}^C), \tag{1.31}$$

where the *conditional information* $I(\mathcal{S}^C \mid \tilde{\mathcal{S}}^C)$ is subtracted from the information content in $\mathcal{S}^C$. It describes the information necessary to restore the set of perfect content patterns $\mathcal{S}^C$ from the set $\tilde{\mathcal{S}}^C$. For random generation of the data we obtain

$$I(\mathcal{S}^C \mid \tilde{\mathcal{S}}^C)/nm = \frac{M}{m} I(y_i^k \mid \tilde{y}_i^k) \tag{1.32}$$

with the contribution of one digit

$$\begin{aligned} I(y_i^k \mid \tilde{y}_i^k) = & \text{Prob}[\tilde{y}_i^k = 1] i(\text{Prob}[y_i^k = 0 \mid \tilde{y}_i^k = 1]) \\ & + \text{Prob}[\tilde{y}_i^k = 0] i(\text{Prob}[y_i^k = 1 \mid \tilde{y}_i^k = 0]) \\ = & [q(1-e_1) + (1-q)e_a]\, i\left(\frac{(1-q)e_a}{q(1-e_1) + (1-q)e_a}\right) \\ & + [qe_1 + (1-q)(1-e_a)]\, i\left(\frac{qe_1}{qe_1 + (1-q)(1-e_a)}\right). \end{aligned} \tag{1.33}$$

Now we define the *association capacity* as the maximal channel capacity per synapse

$$A(m,n) := \max_M T(\tilde{\mathcal{S}}^C, \mathcal{S}^C)/mn = P(m,n) - \frac{M^*}{m} I(y_i^k \mid \tilde{y}_i^k). \tag{1.34}$$

The capacity of one component of the channel is an upper bound for the capacity of the whole channel: The capacity of the first box in Fig. 2 will be called *storage capacity* (discussed in [41]). The maximal memory capacity that can be achieved for a fixed retrieval procedure (i.e. fixing only the last box in Fig. 2) will be called the *retrieval capacity*.

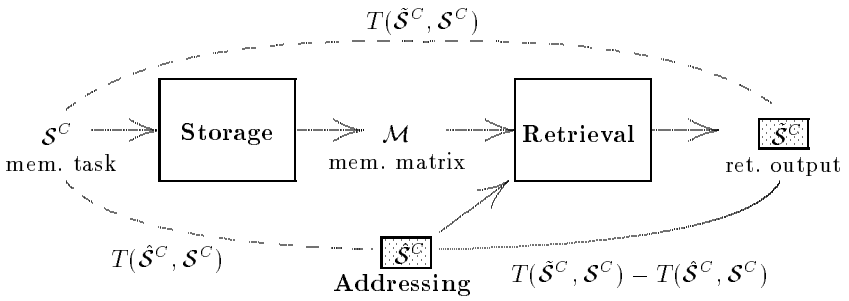

FIGURE 1.3. Completion capacity: Information balance for autoassociation. (The abbrevations "mem." for memory and "ret." for retrieval have been used.)

1.4.3 Including the Addressing Process

The defined association capacity is a quality measure of the retrieved content patterns but the retrieval quality depends on the properties of the input patterns and on the addressing process. Of course, maximal association capacity is obtained for faultless addressing and with growing addressing faults (decreasing probability p') the association capacity A decreases because the number of patterns has to be reduced to satisfy the same retrieval error bounds. To include judgement of addressing fault tolerance for hetero-association we have to observe the dependency $A(p')$.

For auto-association where $\mathcal{S}^A = \mathcal{S}^C$ we will consider the information balance between the information already put into the memories input and the association capacity (see Fig. 3).

This difference gives the amount of information really gained during the retrieval process. We define the *completion capacity* for auto-association as the maximal difference of the transinformation about $\mathcal{S}^C$ contained in the output patterns and contained in the noisy input patterns $\hat{\mathcal{S}}^A$,

$$C(n) := \max_{\hat{\mathcal{S}}^C} \left\{ T(\mathcal{S}^C \mid \tilde{\mathcal{S}}^C) - T(\mathcal{S}^C \mid \hat{\mathcal{S}}^C) \right\} / n^2. \tag{1.35}$$

From (1.31) we obtain

$$\begin{aligned} C(n) &= \max_{\hat{\mathcal{S}}^C} \left\{ I(\mathcal{S}^C \mid \hat{\mathcal{S}}^C) - I(\mathcal{S}^C \mid \tilde{\mathcal{S}}^C) \right\} / n^2 \\ &= \max_{p'} \left\{ M^* \left[I(y_i^k \mid \hat{y}_i^k) - I(y_i^k \mid \tilde{y}_i^k) \right] \right\} / n. \end{aligned} \tag{1.36}$$

In (1.36) we have to insert again the maximum number of stored patterns M^* and the conditioned information to correct the retrieval errors; cf. eq. (1.33). In addition the one-digit contribution of the conditioned information necessary to restore the faultless address patterns $\mathcal{S}^A$ from the noisy input patterns $\hat{\mathcal{S}}^A$ is required. It is given by

$$I(y_i^k \mid \hat{y}_i^k) = (1 - pp')i\left(\frac{p(1-p')}{1-pp'}\right). \tag{1.37}$$

Note that for randomly generated content patterns, i.e., with complete independence of all the pattern components y_i^k, one usually reaches the optimal transinformation rates and thus the formal capacity.

1.4.4 Asymptotic Memory Capacities

In Sect. 3 we have also analyzed the model in the thermodynamic limit, the limit of diverging memory size. For asymptotic values for the capacities in this limit we will not only examine memory tasks where the fidelity requirement remain constant. We will examine the following *asymptotic fidelity requirements* on the retrieval which distinguish asymptotically different ranges of the behaviour of the quantities e_a and e_1 with respect to $q \to 0$ as $m, n \to \infty$:

- The high-fidelity or *hi-fi* requirement: $e_1 \to 0$ and $e_a/q \to 0$. Note that for $q \to 0$ the hi-fi requirement demands for both error types the same behaviour of the ratio between the number of erroneous and correct digits in the output: $d_a \simeq d_1 \to 0$ with the *error ratios* defined by $d_a := e_a/q$ and $d_1 := e_1/(1-q)$.
- The low-fidelity or *lo-fi* requirement: e_1 and e_a stay constant (but small) for $n \to \infty$

With one of these asymptotic retrieval quality criteria the *asymptotic capacities* P, A and C are defined as the limits for $n, m \to \infty$ and $n \to \infty$, respectively.

1.5 Model Performance

1.5.1 Binary Storage

Output capacity

In this memory model the probability $p_0 = \mathrm{Prob}(\bar{\mathcal{M}}_{ij} = 0)$ is decreased, if the number of stored patterns is increased. Since obviously no information could be drawn from a memory matrix with uniform matrix elements we will exclude the cases $p_0 = 1$ and $p_0 = 0$ in the following.

For faultless addressing the maximal number M^* of patterns which can be stored for a given limit on the error probabilities can be calculated by (1.9) and (1.10),

$$M^* = \frac{\ln[p_0]}{\ln[1-pq]} = \frac{\ln\left[1-(e_a)^{1/mp}\right]}{\ln[1-pq]}. \tag{1.38}$$

From (1.34) we obtain for $e_1 = 0$ and $e := e_a << q$ the association capacity

$$A(m,n) \simeq (M^*/m)\left\{i(q) - (1-q)e\log_2[e(1-q)/q]\right\}. \tag{1.39}$$

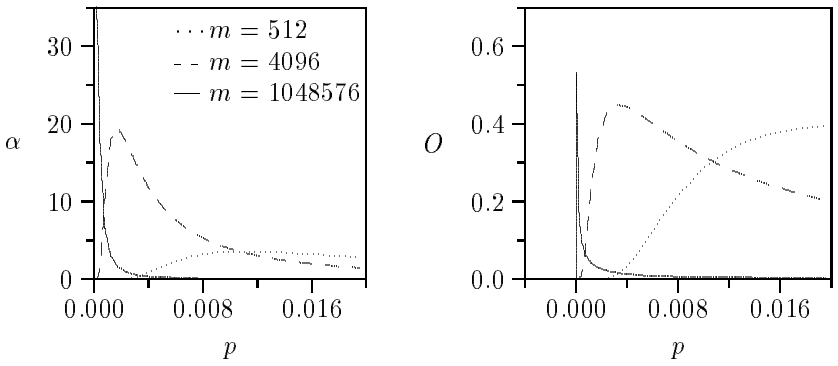

FIGURE 1.4. Binary storage in finite memory sizes: Number of stored patterns α and output capacity A in bits/syn with the lo-fi requirement $d = 0.01$ for $p = q$ and $n = m$.

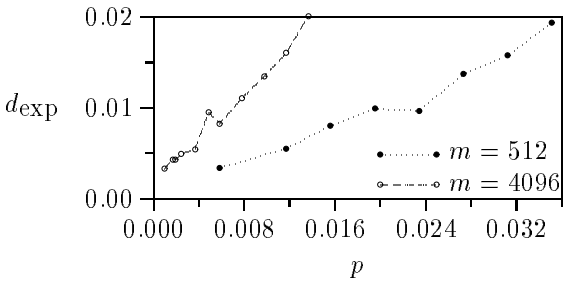

FIGURE 1.5. Retrieval error ratio $d = e_a/k$ of simulations along the α-p curves of Fig. 4 for $d_{\text{theor}} = 0.01$. For low p values, the experimental error is even lower than predicted because we used learning patterns with a non fluctuating activity in the simulations. For higher p values, the theoretic values are too small because in this range the effects of statistical dependence between different matrix elements should not be neclected.

In Fig. 1.4 we have plotted a): $\alpha = M^*/m$ and b): the association capacity against p for $q = p$ and the constant error ratio $d = e_a/p = 0.01$ for three finite memory sizes. Figure 1.5 shows simulation results for the error ratio d. For low p values, the experimental error is even lower than predicted because we used learning patterns with a non-fluctuating activity in the simulations. For higher p values the theoretic values are too small because in this range the effects of statistical dependence between different matrix elements cannot be neclected. Nonvanishing asymptotic association capacity requires $M^*/m > 0$ as $m \to \infty$. In equation (1.38) this can be obtained either for $p_0 \to 0$ which we have already excluded or for $pq \to 0$. In this case we obtain

$$M^* \simeq \frac{\ln[p_0]}{-pq}. \tag{1.40}$$

The hi-fi requirement leads with (1.11) to the following condition on p and q:

$$e_a/q = \exp\left(mp\,\ln[1-p_0] - \ln[q]\right) \to 0. \tag{1.41}$$

In the case $q \to 0$ the requirement (1.41) is satisfied, if we put

$$p = u\frac{-\ln[q]}{m} \tag{1.42}$$

with the positive number $u > -(\ln[1-p_0])^{-1}$. Inserting (1.42) in (1.40) we obtain the inequality

$$M^* < \frac{m\,\ln[p_0]\,\ln[1-p_0]}{-q\,\ln[q]} \tag{1.43}$$

which can be put into (1.39) yielding for $p_0 = 0.5$ and $m \to \infty$ the maximal association capacity: $A \simeq 0.69$ bits/syn.

Note that for auto-association and for hetero-association with $p = q, m = n$ equation (1.42) implies that

$$p \propto \ln[n]/n \tag{1.44}$$

and

$$M^* \propto \left(\frac{n}{\ln[n]}\right)^2. \tag{1.45}$$

The relation (1.45) has already been obtained in [42, 43] for sparse memory patterns with arbitrary learning rules by regarding the space of all possible synaptic interactions; cf. Sect. 1.6.3.

For singular address patterns and arbitrary $q = \text{const}$, however, errorfree retrieval is possible for $M^* \leq m$, which is the combinatorical restriction for nonoverlapping singular patterns. In this case, with (1.39) an association capacity of $A = i(q) \leq 1$ bits/synapse is obtained.

For constant p equation (1.42) demands asymptotically empty content patterns: $q \propto \exp(-mp/u)$, leading to vanishing association capacity.

For singular content patterns the combinatorial restriction $M^* \leq m$ also yields vanishing association capacity.

Fault Tolerance and Completion Capacity

In the case of noisy input patterns (1.12) the hi-fi condition becomes : $e_a/q = \exp\left(mpp'\ln[1-p_0] - \ln[q]\right) \to 0$. Like in the preceeding subsection we obtain the maximal number of patterns by $M'^* = p'M^*$ where M^* is the value for faultless addressing (1.43).

Thus for hetero-association the association capacity exhibits a linear decrease with increasing addressing fault: $A(p') = p'A$.

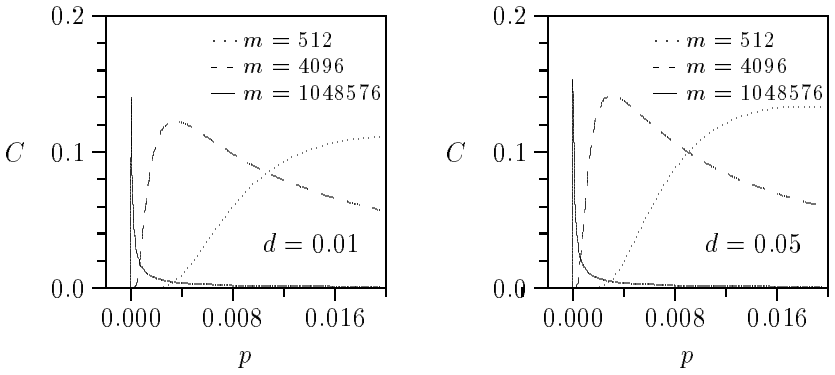

FIGURE 1.6. Binary storage in finite memory sizes: Completion capacity C in bits/syn for two lo-fi values, the maximum has always been achieved for addressation with $p' = 0.5$.

For auto-association with the hi-fi requirement the retrieval error term in the completion capacity (1.36) can be neglected like in the association capacity and we obtain for $p \to 0$

$$\begin{aligned} C &= \max_{p'} \left\{ (M'^*/n)(1-pp')i\left(\frac{p(1-p')}{1-pp'}\right)\right\} \\ &= \max_{p'} \left\{ \frac{\ln[p_0]\ \ln[1-p_0]p'(1-p')}{\ln[2]}\right\} = 0.17 \text{ bits/syn} \end{aligned} \quad (1.46)$$

for $p_0 = 0.5$ and $p' = 0.5$.

In Fig. 1.6 the completion capacity is plotted against p for three finite memory sizes and for the constant error ratios a): $d = e_a/p = 0.01$ and b): $d = 0.05$. The optimum is always obtained for $p' = 0.5$.

1.5.2 Incremental Storage

Output capacity

For faultless addressing, zero average input and the optimal rule R_0, the maximal number of stored patterns for a given signal-to-noise ratio value r is obtained from equation (1.28)

$$M^* = m/(r^2 q(1-q)). \quad (1.47)$$

If the threshold setting provides $e_a/q = e_1/(1-q) =: d$, the association capacity can be computed for small fixed values of the error ratio d from (1.34) and (1.47)

$$A \simeq \frac{i(q) + q(1-q)d\left\{\log_2[qd] + \log_2[(1-q)d]\right\}}{r^2 q(1-q)} \quad (1.48)$$

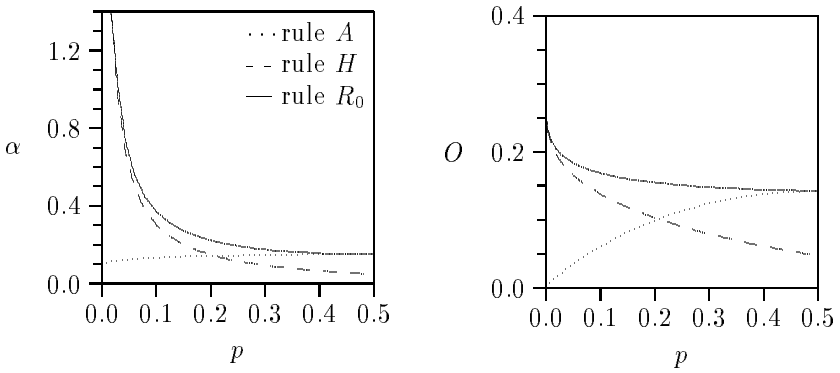

FIGURE 1.7. Model with incremental storage, fullfilled condition of zero average input and $m, n \to \infty$: Number of stored patterns α (left) and asymptotic output capacity A in bits/synsapse (right) for $p = q$ with the lo-fi requirement $d = 0.01$. The optimal rule R_0 is approached by the agreement rule A for $p = 0.5$ and by the Hebb rule for $p \to 0$. For $p \to 0$, the lo-fi output capacity values of optimal and Hebb rule reach but do not exceed the hi-fivalue of $A = 0.72$ bits/synapse (This can only be observed, if the p scale is double logarithmic; see Fig. 5 in [Pa91]).

With substitution of $r = G^{-1}[qd] + G^{-1}[(1-q)d]$ in (1.48) we obtain the association capacity for the rule R_0 for a constant d error ratio, the lo-fi requirement. ($G^{-1}[x]$ is the inverse Gaussian distribution.) In Fig. 1.7 we display the association capacity values for optimal, Hebb and agreement rule, the latter two obtained by comparison of the signal-to-noise ratios in Table 1, Sect. 1.3.4.

The hi-fi requirement can only be obtained for $r \to \infty$ as $m \to \infty$ in (1.47) which is possible either for $M^*/m \to 0$, leading to vanishing association capacity or for $q \to 0$, the case of *sparse* content patterns, which we focus on in the following.

We now choose a diverging signal-to-noise ratio by

$$r = \sqrt{-2\ln[q]}/\vartheta. \tag{1.49}$$

The threshold has to be set asymmetrically: $\vartheta \to 1$ because for sparse patterns $e_a/e_1 \to 0$ is demanded. (This implies $q = \exp[-(\vartheta r)^2/2]$, yielding with Appendix 2: $e_a/q \simeq (\pi r^2/2)^{-1/2} \to 0$. If the threshold ϑ approaches 1 slowly enough that still $(1-\vartheta)r \to \infty$ holds, then also $e_1 \to 0$ is true and the hi-fi requirement is fullfilled.)

With vanishing e/q equation (1.48) simplifies asymptotically to

$$A \geq P + \frac{2e\log_2[e]}{r^2} \simeq P$$

Again the information loss due to retrieval errors can be neclected due to the high fidelity requirement.

Inserting (1.49) in (1.47) we obtain for zero average input and the optimal rule R_0

$$M^* = m/\left(-2q(1-q)\,\ln[q]\right) \tag{1.50}$$

which again can also be found with the Gardner method [42, 43]; cf. Sect. 1.6.3.

With (1.50) and (1.30) we obtain as asymptotic association capacity with the hi-fi requirement: $A = 0.72$ bits/syn.

In contrast to the model with binary storage – where only for sparse content *and* address patterns a positive association capacity has been obtained – with incremental storage an association capacity $A = 0.72$ bits/syn is achieved even for memory tasks with nonsparse address patterns. However, for $\{0,1\}$-neurons we are again restricted to sparse address patterns because for nonsparse address patterns the zero average input condition cannot be satisfied.

With singular address or content patterns which are no interesting cases for associative memory as we will discuss in Sect.1.6.1, incremental and binary storage form the same memory matrix and achieve exactly the same performance; see last part of Sect. 1.5.1.

Fault tolerance and Completion Capacity

For hetero-association with noisy addressing we obtain the association capacity for zero average input and R_0 by using equation (1.29) (remember that $r^2 \propto m/M$)

$$A(p') = \frac{(1-p)p'^2}{p' - 2pp' + p} A. \tag{1.51}$$

For $p = 0.5$ this implies $A(p') = p'^2 A$ and for $p \to 0$ like in the binary case $A(p') = p'A$.

For auto-association with the hi-fi requirement we obtain in a way similar to (1.46)

$$\begin{aligned} C(n) &= \max_{p'} \left\{ \frac{\vartheta^2 p'(1-p')\log_2[p(1-p')]}{2\,\ln[p]} \right\} \\ &\simeq \max_{p'} \left\{ \frac{\vartheta^2 p'(1-p')}{2\ln[2]} \right\} = 0.18 \text{ bits/syn} \end{aligned}$$

Again the maximum is reached for $p' = 0.5$ and $\vartheta \to 1$.

A similar optimization in p' can be carried out for fixed values of p and lo-fi requirement; see Fig. 1.8. In this case the optimum is reached for p' larger than 0.5.

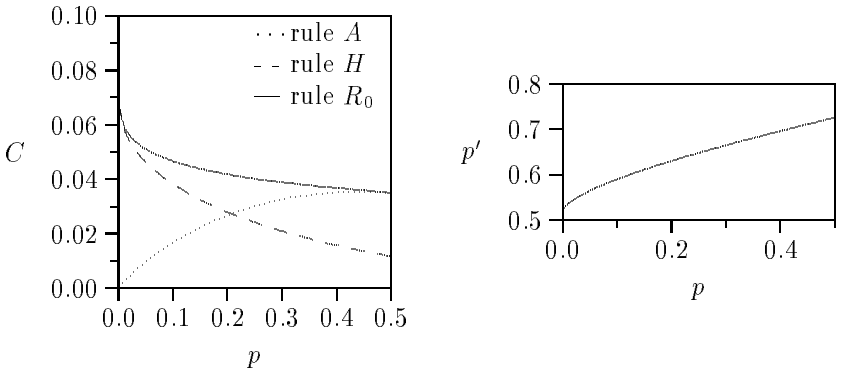

FIGURE 1.8. Incremental storage for $n \to \infty$: Completion capacity in bits/syn with the lo-fi requirement $d = 0.01$. The optimal $\overline{p'}$ in the addressing has been determined numerically (right diagram).

	nonsparse content	sparse content	singular content
nonsparse address	-	incr. R_0	-
sparse address	-	incr. R_0, H bin. H	-
singular address	incr. R_0, H bin. H	-	-

TABLE 1.2. Models which yield $A > 0$ for the hi-fi requirement in different memory tasks. (incr.=incremental storage, bin. = binary storage. For instance: incr.R_0,H denotes the incremental storage model either with optimal rule or with Hebb rule.)

1.6 Discussion

1.6.1 Hetero-association

In applications of associative memory the coding of address and content patterns plays an important role. In Sect. 1.1 we distinguished three types of pattern leading to the memory tasks defined in Sect. 1.4; singular patterns with only a single 1-component, sparse patterns with a low ratio between the numbers of 1- and a-components and nonsparse patterns. To get a general idea Table 2 shows those memory models which achieve association capacity values $A > 0$ under the hi-fi requirement. Note that only Hebb and the optimal learning rule in memory tasks with sparse or singular patterns yield nonvanishing hi-fi association capacity. In the following we shall consider the different types of content patterns subsequently.

	binary	incremental	
	H	H	R_0
nonsparse	-	-	$A = 0.72$
address	-	-	p'^2
sparse	$A = 0.69$	$A = 0.72$	$A = 0.72$
address	p'	p'	p'

TABLE 1.3. Hi-fi association capacity values of the different models for sparse content patterns. As a measure of addressing fault tolerance (cf. Sect. 1.3) in the second line of each cell the reduction factor for faulty addressing is displayed. For instance, with sparse address and content patterns the Hebb rule in the incremental storage yields $A = 0.36$ bits/syn, if in the addressing $p' = 0.5$ is chosen.

Nonsparse Content Patterns

Only in combination with singular address patterns do nonsparse patterns achieve high association capacity. In this case, qualified in Sect. 1.4 as the look-up-table task, all rules achieve $A = 1$. The associative memory works like a RAM device where each of the m content patterns is written into one row of the memory matrix $\bar{\mathcal{M}}$ and, therefore, trivially $A = i(q)$. However, this is no interesting case for associative storage because the storage is not distributed and in the recall no fault tolerance can be obtained: $A(p') = 0$ for $p' < 1$.

Sparse Content Patterns

Combined with sparse or nonsparse address patterns sparse content patterns represent the most important memory task for neural memory models with Hebb or optimal learning rule where high capacity together with associative recall properties is obtained. For optimal association capacity many patterns in the set of sparse learning patterns will overlap. Therefore, in the learning process several pattern pairs affect the same synapse and distributed storage takes place. In Table 3 the hi-fi association capacity values can be compared. For sparse address patterns, Hebb and optimal rule achieve exactly the same performance because with the zero average input condition both rules are essentially identical. Even the binary Hebb rule shows almost the same performance. At a first sight it is striking that binary storage, using only one bit synapses, yields almost the same performance as incremental storage, using synapses that can take many discrete values. This fact becomes understandable, if we consider the mean contributions of all patterns at one synapse by incremental and by binary storage: $E\mathcal{M} = 0.69$ for incremental compared with $E\bar{\mathcal{M}} = 0.5$ for binary storage. In both cases the sparseness requirement prevents the matrix elements from extensive growth; also in incremental storage the vast majority of synapses take only the values 0, 1, and 2.

For nonsparse address patterns only the optimal setup, namely, the rule R_0 in the incremental storage, achieves nonvanishing association capacity. This case is of less importance for applications since implementation is much more difficult (higher computation effort for $a \neq 0$ and the determination of the value of a requires the parameter p of the patterns).

Relaxing the quality criterion does not enhance the association capacity value in the sparse limit. The lo-fi association capacity values, plotted in Fig. 4 and Fig. 7 do not exceed the hi-fi values of Table 3. With the agreement rule finite lo-fi association capacity values can be achieved (see Fig. 7) whereas the hi-fi association capacity always vanishes.

Singular Content Patterns

The neural pattern classifier which responds to a nonsingular input pattern with a single active neuron is often called "grandmother model" or perceptron. Here the information contained in the content patterns is asymptotically vanishing compared to the size of the network: $A = 0$. Again no distributed storage takes place.

1.6.2 Auto-Association

If content and address pattern are identical in order to accomplish pattern completion in the retrieval, we have only to regard the cases of sparse and nonsparse learning patterns.

Asymptotic Results

The amount of information that can be really extracted by pattern completion with high quality is given by the asymptotic hi-fi completion capacity. It always vanishes in case of nonsparse patterns. For one-step retrieval with sparse patterns we have determined $C = 0.18$ and $C = 0.17$ bits/syn for the Hebb rule in incremental and binary storage respectively (Sects. 1.5.1 and 1.5.2).

Using a practically unrealistic fixed-point read-out scheme[7] and the Hebb rule we have found completion capacity values of $C = 0.36$ bits/syn for incremental and $C = 0.35$ bits/syn for binary storage [30, 23]. Thus one would expect the performance of one-step retrieval to be improved by fixed-point retrieval, i.e., starting from a single address pattern and *iterating* the retrieval process until the fixed-point is reached. Asymptotically, however, fixed-point retrieval does not improve the one-step capacity results [44, 45, 46]. It is a consequence of the fullfilled hi-fi condition that already after the first step we get asymptotically vanishing errors for diverging system size.

[7] Fixed points are patterns which remain unchanged during a retrieval step i.e., input and output pattern are identical.

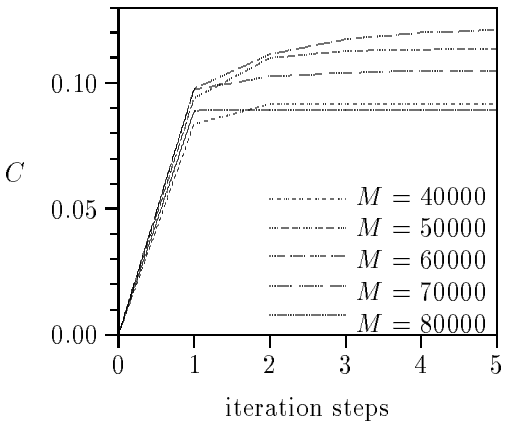

FIGURE 1.9. Completion capacity C in bits/syn for iterative retrieval for address-sation with $p' = 0.5$ which has been achieved in simulations in binary storage with 4096 neurons. Depending on the number of stored patterns M an improvement up to twenty percent (for $M = 60000$) can be obtained after the first step through iteration.

Finite-Size Systems

Although Fig. 1.6 illustrates that the asymptotic capacity bounds are only reached for astronomic memory sizes, even for realistic memory sizes sparse patterns yield better performance than nonsparse patterns. Simulations and analysis have revealed (again cf. [44, 45]) that iterative retrieval methods with an appropriate threshold setting scheme (saying how the threshold has to be aligned during the sequence of retrieval steps), yield superior exploitation of the auto-association storage matrix as compared to one-step retrieval; see Fig. 1.9. For finite systems, fixed-point retrieval does even improve the performance and capacity values above the asymptotic value; e.g. for $n = 4096$ about $C = 0.19$ bits/syn can be obtained.

For a certain application and a given finite memory size, however, we cannot reduce the pattern activity ad libitum by modifying the coding algorithm. Then we may sometimes be faced with $p >> \ln[n]/n$; cf. (1.42). In this case, binary Hebbian storage is ineffective – see Fig. 6 – and incremental storage does not work either.

1.6.3 Relations to other approaches

Hetero-association

The zero average input condition for memory schemes with non-optimal local synaptic rules was first made explicit by Palm [47] but appeared implicitely in some closely related papers. Horner [48] has used it to define the neural off-value a in his model and Nadal and Tolouse [24] have exploited

it (through their condition of 'safely sparse' coding) as a justification for their approximations.

The optimization of the signal-to-noise ratio r carried out by Willshaw and Dayan [37] and independently by Palm [47] has already been suggested – though not carried out – by Hopfield [25]. Also Amit et al [8] have proposed the covariance rule R_0.

The signal-to-noise ratio is a measure of how well threshold detection can be performed in principle, independently of a certain strategy of threshold adjustment. We have examined the model where the threshold assumes the same value Θ for all neurons during one retrieval step and optimized the response behavior depending on the individual input activity. So we could lump together the on- and off- fractions of output neurons and calculate the average signal-to-noise ratio.

In a recent work Willshaw and Dayan [49] have carried out a signal-to-noise analysis using quite similar methods for a different model. In their model the threshold setting Θ_j has been chosen individually for each neuron for the average total activity of input patterns. Thus the signal-to-noise ratio at a single neuron has been optimized for averaged input activity. Due to this difference the results only agree for zero average input activity and in the thermodynamic limit; for the same optimal rule the same signal-to-noise ratio is obtained. In general, their model is not invariant under the addition of an arbitrary constant in the learning rule because for $E(R) \neq 0$ activity fluctuations in an individual input pattern are not compensated by threshold control as in our model.

Most of the results for hetero-association discussed here can be found in the literature in Peretto [50], Nadal and Toulouse [24], Willshaw and Dayan [37] and Palm [47, 51]). Some of our results are numerically identical to those of Nadal and Toulouse who employ different arguments (e.g., approximation of the distribution of the noise term (1.13) by a Poisson distribution). In our framework one could also define a "no fidelity requirement", namely e_a and $e_1 \to 0.5$, which would correspond to the "error-full regime" of Nadal and Toulouse. This leads to the same numerical result $A = 0.46$, which, however, is not very interesting from the engineering point of view since it is worse then what can be achieved with high fidelity. The result for binary storage stems from Willshaw et al [4] for the Hebb rule, and to Hopfield [25] for the agreement rule. A new aspect is the information-theoretical view on the tradeoff between association capacity and fault tolerance.

Auto-association

Auto-association has been treated extensively in the literature; see for example [8, 25, 43, 26, 29]. In two points our treatment differs from most of the papers on auto-association:

- Usually models with fixed-point retrieval (and only with incremental storage) have been considered.
- As the appropriate performance measure for pattern completion we evaluate and compare the completion capacity which takes into account the entire information balance during the retrieval.

With one exception [48, 52] other authors regard (in our terms) the pattern capacity, i.e., the retrieval starts from the perfect pattern as address[8]. Hence, to compare the existing fixed-point results with our one-step retrieval for auto-association we should take the association capacity or pattern capacity results, calculated in Sect. 1.5.2 for hetero-association, in the case $p = q$.

For nonsparse patterns with $p = 0.5$, fixed-point retrieval with the lo-fi requirement stays below one-step retrieval: for the same fidelity of $d = 0.002$ the one-step result for the agreement rule (Fig. 4) is higher than the Hopfield bound for the fixed-point retrieval in [10, p.296]. Here one-step retrieval behaves more smoothly with respect to increasing memory load because the finite retrieval errors after the first step are not further increased by iterated retrieval. If the lo-fi fidelity requirement is succesively weakened, a smooth increase of the one-step association capacity can be observed and no sharp overload breakdown of the capacity (the Hopfield catastrophy) takes place as it is known for fixed-point retrieval at the Hopfield bound α_c [25, 8, 29].

The pattern capacity for the binary agreement rule has been estimated by a comparison of the signal-to-noise ratios for the binary and nonbinary matrix in [25] and has been exactly determined in [26] as $A^b = (2/\pi)A$. For nonsparse learning patterns binary storage is really worse than incremental storage.

Again, as for hetero-association, only for sparse patterns nonzero values for the asymptotic hi-fi capacities can be achieved. For one-step retrieval with $a = 0$ we have found a hi-fi pattern capacity of $P = 0.72$ bits/syn. For fixed-point retrieval, it has been possible to apply the statistical mechanics method to sparse memory patterns; cf. for instance [53, 27]. In [27] just the same value $P = 0.72$ bits/syn has been obtained. By a combinatorial calculation we have also obtained this pattern capacity value for fixed-point retrieval [30]. One-step and fixed-point retrieval yield the same pattern capacity because for sparse patterns the hi-fi condition is satisfied. It guarantees that almost all learned patterns are preserved in the first retrieval step and hence are fixed-points.

[8] To obtain the pattern capacity, it is sufficient to study the properties of the fixed-points as a static problem. Evaluating the completion capacity one has to study how the system state evolves from a noisy input pattern in order to determine the properties of the output pattern with a given address. This is a dynamic problem which is in fact very difficult.

Quite a different way to analyze the storage of sparse and nonsparse patterns through statistical mechanics has been developed by Gardner [42, 43]. In the space of synaptic interactions, she has determined the subspace where every memory pattern is a stable fixed point. For sparse patterns this method yields the same pattern capacity value.

1.6.4 Summary

The main concerns of this paper can be summarized as follows:

- The statistical analysis of a simple feed-forward model with one-step retrieval provides the most elementary treatment of the phenomena of distributed memory and associative storage in neural architecture.
- The asymptotic analytical results are consistent with the literature. For auto-association, most of the cited works consider fixed-point retrieval which allows us to compare one-step with fixed-point retrieval.
- Our information-theoretical approach introduces the capacity definitions as the appropiate performance measures evaluating for the different memory tasks the *information per synapse* which can be stored and recalled. Note that nonvanishing capacity values imply that the information content is proportional to the number of synapses in the model.
- For *local* learning rules *sparse* content patterns turns out to be the best possible case, cf. [54]. High capacity values and distributed storage with fault tolerant retrieval are provided by the Hebb rule and $\{0, 1\}$ neurons. Here the number of stored patterns is much higher than the number of neurons constituting the network. The binary Hebb rule – much easier to implement – yields almost the same performance as the incremental Hebb rule. For auto-association one-step retrieval achieves the same asymptotic capacity values as fixed-point retrieval (for the finite-size model fixed-point retrieval yields higher capacity values). The hi-fi condition can always be fullfilled by sparse content patterns and only by these.

Acknowledgement. We are indepted to F. Schwenker for Fig. 1.9 and for many helpful discussions. We thank J.L. van Hemmen for a critical reading of the manuscript. This work was partially supported by the Bundesministerium für Forschung und Technologie.

Appendix 1

In this section we show for the Hebb rule in binary storage the independence of two different matrix elements. This is required in Sect. 3.2.

Proposition 1 *For the binary storage matrix $\mathcal{M}$ we have as $n \to \infty$*

$$\frac{Prob[\mathcal{M}_{1j}=1 \text{ and } \mathcal{M}_{2j}=1]}{Prob[\mathcal{M}_{1j}=1]Prob[\mathcal{M}_{2j}=1]} \to 1 \text{ and } \frac{Prob[\mathcal{M}_{j1}=1 \text{ and } \mathcal{M}_{j2}=1]}{Prob[\mathcal{M}_{j1}=1]Prob[\mathcal{M}_{j2}=1]} \to 1$$

provided p and $q \to 0$ and $x := Mpq$ stays away from zero for $n \to \infty$.

Proof. $\text{Prob}[\mathcal{M}_{ij} = 1] = 1 - (1-pq)^M$.

$$\begin{aligned}\text{Prob}[\mathcal{M}_{1j}=1 \text{ and } \mathcal{M}_{2j}=1] &= \text{Prob}[(\exists k : x_1^k = x_2^k = 1 \text{ and } y_j^k = 1) \text{or} \\ &\quad (\exists l, m : x_1^l, x_2^l = 0, x_1^m = 0, x_2^m, y_j^l = 1, y_j^m = 1)] \\ &= 1 - (p(E_1) + p(E_2) - p(E_1 \cap E_2)),\end{aligned}$$

where

$$E_1 = [\forall k : \text{not } (x_1^k = x_2^k = 1 \text{ and } y_j^k = 1) \text{ and not } (x_1^k = 1, x_2^k = 0, y_j^k = 1)]$$

and

$$E_2 = [\forall k : \text{not } (x_1^k = x_2^k = 1 \text{ and } y_j^k = 1) \text{ and not } (x_1^k = 0, x_2^k = 1, y_j^k = 1)].$$

Thus $\text{Prob}(E_1) = \text{Prob}(E_2) = (1-pq)^M$ and $\text{Prob}(E_1 \cap E_2) = (1 - q(2p - p^2))^M$.

Therefore we obtain

$$\begin{aligned}&\text{Prob}[\mathcal{M}_{1j} = 1 \text{ and } \mathcal{M}_{2j} = 1] - \text{Prob}[\mathcal{M}_{1j} = 1] \cdot \text{Prob}[\mathcal{M}_{2j} = 1] \\ &= (1-2qp+qp^2)^M - (1-pq)^{2M} = (1-2qp+qp^2)^M - (1-2pq+p^2q^2)^M \\ &= e^{-M(2pq-p^2q)} - e^{-M(2pq-p^2q^2)} = e^{-2pqM}(e^{Mp^2q} - e^{Mp^2q^2}).\end{aligned}$$

Thus we find

$$\frac{\text{Prob}[\mathcal{M}_{1j} = 1 \text{ and } \mathcal{M}_{2j} = 1] - \text{Prob}[\mathcal{M}_{1j} = 1] \cdot \text{Prob}[\mathcal{M}_{2j} = 1]}{\text{Prob}[\mathcal{M}_{1j} = 1] \cdot \text{Prob}[\mathcal{M}_{2j} = 1]} = \frac{e^{-2x}(e^{px} - e^{qpx})}{(1-e^{-x})^2} \to 0$$

since $px \to 0$ and $pqx \to 0$.

This proposition shows the asymptotic pairwise independence of the entries $\mathcal{M}_{ij}$ in the memory matrix $\mathcal{M}$, since entries which are not on the same row or column of the matrix, are independent anyway.

In order to show complete independence one would have to consider arbitrary sets of entries $\mathcal{M}_{ij}$. In this strict sense the entries cannot be independent asymptotically. For example, if one considers all entries in one column of the matrix, then $\text{Prob}[\mathcal{M}_{ij} = 0 \text{ for all } i] = (1-q)^M \approx e^{-Mq}$ which is with (1.9)in general not equal to $p_0^m = (1-pq)^{Mm} \approx e^{-Mmpq}$.

Thus independence can at the best be shown for sets of entries of the matrix $\mathcal{M}$ up to a limited cardinality $L(n)$. The worst case, which is also important for our calculations of storage capacity, is again when all entries are in the same column (or row) of the matrix. This case is treated in the next proposition, which gives only a rough estimate.

Proposition 2

$$\frac{Prob[\mathcal{M}_{ij} = 1 \; for \; i = 1, \ldots, l]}{Prob[\mathcal{M}_{ij} = 1]^l} \to 1 \; for \; n \to \infty$$

as long as $pl^2 \to 0$ *and* $x = Mpq$ *stays away from zero for* $n \to \infty$.

Proof.

$$\begin{aligned}
\text{Prob}[\mathcal{M}_{ij} = 1] &\leq \text{Prob}[\mathcal{M}_{lj} = 1 | \mathcal{M}_{ij} = 1 \text{ for } i = 1, \ldots, l-1] \\
&\leq \text{Prob}[\mathcal{M}_{lj} = 1 | \text{ there are at least } l-1 \text{ pairs } (x^k, y^k) \text{ with } y_j^k = 1] \\
&= 1 - (1-p)^{l-1}(1-pq)^{M-l+1}.
\end{aligned}$$

Therefore

$$\begin{aligned}
0 &\leq \log \frac{p[\mathcal{M}_{ij} = 1 \text{ for } i = 1, \ldots, l]}{p[\mathcal{M}_{ij} = 1]^l} \leq \sum_{i=0}^{l-1} \log \frac{1-(1-p)^i(1-pq)^{M-i}}{1-(1-pq)^M} \\
&= \sum_{i=0}^{l-1} \log \frac{1 - (\frac{1-p}{1-pq})^i p_0}{1-p_0} \leq \sum_{i=0}^{l-1} \log \frac{1-(1-ip)p_0}{1-p_0}, \\
&\text{since} \quad (\frac{1-p}{1-pq})^i \geq (1-p)^i \geq 1 - ip, \\
&\leq \sum_{i=0}^{l-1} ip \frac{p_0}{1-p_0}, \\
&\text{since} \quad \log(1+x) \leq x, \\
&\leq \frac{p \cdot p_0}{1-p_0} \cdot \frac{l^2}{2} \to 0 \text{ for } p \cdot l^2 \to 0, \\
&\text{and if} \quad p_0 = (1-pq)^M \approx e^{-Mpq} = e^{-x} \not\to 1.
\end{aligned}$$

For (1.10) we need the independency of $l = mp$ matrix elements, thus for sparse address patterns with $m^{2/3}p \to 0$ the requirement of Prop. 2 is fullfilled and the independence can be assumed.

Appendix 2

The following estimation of the Gauss integral $G(t)$ is used in Sect. 5.2.

Proposition 3

$$(2\pi t^2)^{-1/2} e^{-t^2/2}(1-t^2) \leq G(-t) = 1 - G(t) \leq (2\pi t^2)^{-1/2} e^{-t^2/2}$$

Proof. Since $x^2 = t^2 + (x-t)^2 + 2t(x-t)$, we have

$$\int_t^\infty e^{-x^2/2} dx = e^{-t^2/2} \int_0^\infty e^{-x^2/2} e^{-xt} dx$$

From this and with $e^{-x^2/2} \leq 1$ we obtain the second inequality directly since $\int_0^\infty e^{-xt} dx = 1/t$ and the first one after partial integration because $\int_0^\infty x e^{-xt} dx = 1/t$.

1.7 References

[1] Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. (Lond.) **117** (1952) 500 - 544

[2] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in neural activity. Bull. of Math. Biophys. **5** (1943)

[3] Steinbuch, K.: Die Lernmatrix. Kybernetik **1** (1961) 36

[4] Willshaw, D.J., Buneman, O.P., Longuet-Higgins, H.C.: Nonholographic associative memory. Nature (London) **222** (1969) 960 - 962

[5] Rosenblatt, F.: Principles of neurodynamics. Spartan Books, New York (1962)

[6] Little, W.A.: The existence of persistent states in the brain. Math. Biosci. **19** (1974) 101 - 120

[7] Kirkpatrick, S., Sherrington, D.: Infinite-ranged models of spin-glasses. Phys. Rev. B **17** (1978) 4384 - 4403

[8] Amit, D.J., Gutfreund, H., Sompolinsky H.: Statistical mechanics of neural networks near saturation. Ann. Phys. **173** (1987) 30 - 67

[9] Domany, E., van Hemmen, J.L., Schulten, K: Models of neural networks. Springer, Berlin (1991)

[10] Amit, D.J.: Modelling brain function. Cambridge University Press (1989)

[11] Hertz, J., Krogh, A. , Palmer, R.G.: Introduction to the theory of neural computation. Addison Wesley, Redwood City, CA (1991)

[12] Uttley, A.M.: Conditional probability machines and conditional reflexes. In: An. Math. Studies **34**, Eds: Shannon, C.E., McCarthy, J., Princeton Univ. Press, Princeton, NJ (1956) 237 - 252

[13] Longuett-Higgins, H.C., Willshaw, D.J., Buneman, O.P.: Theories of associative recall. Q. Rev. Biophys. **3** (1970) 223 - 244

[14] Amari, S.I.: Characteristics of randomly connected threshold-element networks and network systems. Proc. IEEE **59** (1971) 35 - 47

[15] Gardner-Medwin, A.R.: The recall of events through the learning of associations between their parts. Proc. R. Soc. Lond. B. **194** (1976) 375 - 402

[16] Kohonen T.: Associative memory. Springer, Berlin (1977)

[17] Caianiello, E.R.: Outline of a theory of thought processes and thinking machines. J. theor. Biol. **1** (1961) 204 - 225

[18] Holden, A.V.: Models of the stochastic activity of neurons. Springer, Berlin (1976)

[19] Abeles, M.: Local cortical circuits. Springer, Berlin (1982)

[20] Buhmann, J., Schulten, K.: Associative recognition and storage in a model network of physiological neurons. Biol. Cybern. **54** (1986) 319 - 335

[21] Anderson, J.A.: A memory storage model utilizing spatial correlation functions. Kybernetik **5** (1968) 113 - 119

[22] Anderson, J.A.: A simple neural network generating an interactive memory. Math. Biosci. **14** (1972) 197 - 220

[23] Palm, G.: On associative memory. Biol. Cybern. **36** (1980) 19 - 31

[24] Nadal, J.-P., Toulouse, G.: Information storage in sparsely coded memory nets. Network **1** (1990) 61 - 74

[25] Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Sci. **79** (1982) 2554 - 2558

[26] van Hemmen, J.L.: Nonlinear networks near saturation. Phys. Rev. A: Math. Gen. **36** (1987) 1959 - 1962

[27] Tsodyks, M.V., Feigelman, M.V.: The enhanced storage capacity in neural networks with low activity level. Europhys. Lett. **6** (1988) 101 - 105

[28] Amari, S.I.: Statistical neurodynamics of associative memory. Neural Networks **1** (1989) 63 - 73

[29] Fontanari, J.F., Köberle, R.: Information processing in synchronous neural networks. J. Phys. France **49** (1988) 13 - 23

[30] Palm, G., Sommer, F.T.: Information capacity in recurrent McCulloch-Pitts networks with sparsely coded memory states. Network **3** (1992) 1 - 10

[31] Gibson, W.G., Robinson, J.: Statistical analysis of the dynamics of a sparse associative memory. Neural Networks **5** (1992) 645 - 662

[32] Hebb, D.O.: The organization of behavior. Wiley, New York (1949)

[33] Herz, A., Sulzer, B., Kühn, R., van Hemmen, J.L.: The Hebb rule: storing static and dynamic objects in an associative neural network. Europhys. Lett. **7** (1988) 663 - 669; Hebbian learning reconsidered: representation of static and dynamic objects in associative neural nets. Biol. Cybern. **60** (1989) 457 - 467

[34] Personnaz, L., Dreyfus, G., Toulouse, G.: A biologically constrained learning mechanism in networks of formal neurons. J. Stat. Phys. **43** (1986) 411 - 422

[35] Personnaz, L., Guyon, I., Dreyfus, G.: Collective computational properties of neural networks: new learning mechanisms. Phys. Rev. A: Math. Gen. **34** (1986) 4217 - 4228

[36] Palm, G.: Neural assemblies. Springer, Berlin (1982)

[37] Willshaw, D.J., Dayan, P.: Optimal plasticity from matrix memories: what goes up must come down. Neural Comp. **2** (1990) 85 - 93

[38] Barto, A.G., Sutton, R.S., Brouwer, P.S.: Associative search network; a reinforcement learning associative memory. Biol. Cybern. **40** (1981) 201 - 211

[39] Lamperti, J.: Probability. Benjamin, New York (1966)

[40] Shannon, C., Weaver, W.: The mathematical theory of communication. University of Illinois Press, Urbana, Ill. (1949)

[41] Palm, G.: On the information storage capacity of local learning rules. Neural Comp. **4** (1992) 703 - 711

[42] Gardner, E.: Maximum storage capacity in neural networks. Europhys. Lett. **4** (1987) 481 - 485

[43] Gardner, E.: The space of interactions in neural network models. J. Phys. A: Math. Gen. **21** (1988) 257 - 270

[44] Schwenker, F., Sommer, F.T., Palm, G.: Iterative retrieval of sparsely coded patterns in associative memory. Neuronet'93 Prague (1993);

[45] Sommer, F.T.: Theorie neuronaler Assoziativspeicher; Lokales Lernen und iteratives Retrieval von Information. Ph.D. thesis, Düsseldorf (1993)

[46] Palm, G., Schwenker, F., Sommer, F.T.: Associative memory and sparse similarity perserving codes. In: From Statistics to Neural networks: Theory and Pattern Recognition Applications, Ed: Cherkassky, V., Springer NATO ASI Series F, Springer, New York (1993)

[47] Palm, G.: Local learning rules and sparse coding in neural networks. In: Advanced Neural Computers, Ed: Eckmiller, R., Elsevier, Amsterdam (1990) 145 - 150

[48] Horner, H.: Neural networks with low levels of activity: Ising vs. McCulloch-Pitts neurons. Z. Phys. B **75** (1989) 133 - 136

[49] Willshaw, D.J., Dayan, P.: Optimizing synaptic learning rules in linear associative memories. Biol. Cybern. (1991) 253 - 265

[50] Peretto, P.: On learning rules and memory storage abilities. J. Phys. France **49** (1988) 711 - 726

[51] Palm, G.: Memory capacities of local rules for synaptic modification. Concepts in Neuroscience **2** (1991) 97-128

[52] Horner, H., Bormann, D., Frick, M., Kinzelbach, H, Schmidt, A.: Transients and basins of attraction in neural network models. Z. Phys. B **76** (1989) 381 - 398

[53] Buhmann, J., Divko, R., Schulten, K.: Associative memory with high information content. Phys. Rev. A **39** (1989) 2689 - 2692

[54] Palm, G.: Computing with neural networks. Science **235** (1987) 1227 - 1228