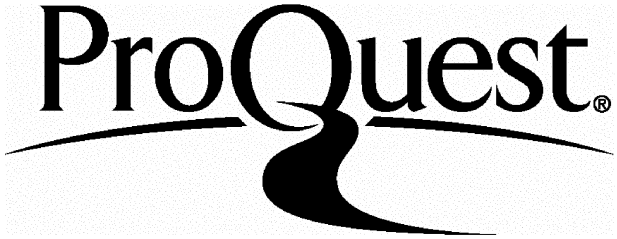
Performance analysis of a ~~partially~~ connected ~~recurrent~~ associative ~~net~~

Hajime Hirase

Doctor of Philosophy
University College London
1996



Abstract

Associative networks have long been regarded as a biologically plausible mechanism for memory storage and retrieval. When the input to the associative network includes erroneous information or the network is incompletely connected, the performance critically depends on the thresholding strategy. Applying a low threshold increases the probability that spurious cell will fire. On the other hand, a high threshold decreases the probability that correct cells will fire. This thesis is concerned with finding improved thresholding strategies in partially connected recurrent associative networks that have binary weights.

In the case of simple feedforward associative networks, the choice of optimal threshold can be analytically computed on the basis of statistical expectations. However, when the network is recurrently connected, such a mathematical analysis of the network behaviour becomes rather complex. Using the formalism proposed by Gibson and Robinson (1992), optimal thresholding sequences are identified for a partially connected recurrent associative network. Surprisingly, the value of the threshold is found to be proportional to the current activation of the network. Based on this result, it is concluded that a linear thresholding strategy (where the level of global threshold is computed linearly from the current activation of the network) is near optimal as far as storage capacity (but not retrieval speed) is concerned. This result is validated in a simulation study. Also, the coefficients of the linear threshold equeation were found to be independent of the memory loading.

The linear thresholding strategy is a global thresholding strategy in that every cell undergoes the same threshold. It is argued that a cell-specific thresholding, in which each cell receives a different level of threshold, produces a better recall performance. A new method to implement local threshold using interneuron learning is proposed. In this paradigm, real-valued synapses are introduced to the inhibitory interneuron that effectively set the threshold level of each principal cell. By varying the inhibitory synaptic efficacy, local thresholding is achieved. It is demonstrated that a simple Hebbian mechanism in the projection from the interneuron can significantly improve the performance of multi-step recall.

Finally, implications for a biological form of associative memory are discussed especially in relation to the hippocampus, which is thought to be involved in the process of intermediate memory and in consolidation.

Declaration

I have composed this thesis myself and it reports original research that has been conducted by myself unless otherwise indicated.

Acknowledgements

I thank my parents for providing a financial support to continue my study. I thank Michael Recce for his enthusiastic and patient supervision, as well as corrections of my English. Also, I would like to acknowledge the members of the neural computation and robotics group, with whom I had many fruitful discussions with academic atmosphere. The members include Giorgio Grasso, Ken Harris, David Jack, Adrienne James, Charles King, Asaur Rahman, John Taylor (alphabetical order), not to mention Michael Recce who leads the group.

Finally, financial support from the ORS scholarship scheme is gracefully acknowledged.

Contents

Chapter 1

Introduction

1.1 Introduction

One of the most fundamental and important functions of the brain is the ability to memorise and recall events. Memory tasks are identified in various ways. For example, we take it for granted that we can remember the face of a person whom we met several hours ago by just listening to his voice in a telephone conversation. Also, from the year a film was made and name of an actor, one can recall the title of the film. Even more, one can probably recall the story line, the scenes, the cinema, etc. A small amount of information can lead to the recall of a substantial amount of information. In both of these examples, the memory is retrieved by associating an event (input cue) to other events stored in the brain.

Conventional computers treat memory processes rather differently. Computer memory is composed of many storage units[1] that are capable of storing a binary vector of a fixed dimension (usually 16, 32 or 64). Each of the storage units has a unique location, called the *address*.

In computer memory, there is no semantic relation between the address of

[1] These units are often called *memory cells* or simply, *cells*. However, to avoid confusion with biological cells or artificial neurons, the term "cell" is not used for computer memory cells.

a memory location and its contents. One aspect of human memory seems to be triggered by specifying a small part of the *contents* of an stored event. This type of memory is called *content-addressable* or *associative*. As memory ability is indispensable in human intelligence, it is important to try to understand the underlying neuronal mechanism to process memory in the brain, in order to build an artificial device with capabilities similar to humans.

Generally, one of two approaches is taken in modelling neural systems. One approach is to model the physiological properties of each neuron as precisely as possible. This approach aims to build a model that faithfully reproduces cellular phenomena that can be measured in experiments, such as electroencephalogram (EEG) recordings or intracellular recordings of membrane potential. A typical example is Hodgkin and Huxley's formalism on spike form transduction (Hodgkin and Huxley, 1952a-d). While this approach may help us understand what a single cell is doing in great detail, it is difficult to use this method to see the functional significance of an ensemble of cells. Precise modelling requires the specification of so much of the complexity of the dynamics that it is almost impossible to simulate a reasonably sized network of the cells using ordinary computers. Also, it is still unclear how much of this complexity is relevant to the function of neuronal networks in the brain. However, with recent advances in computer technology, progress is being made in network scale modelling of detailed neuronal properties (e.g. Traub and Miles 1991).

The alternative approach is to use a relatively simple model for the neuron and study the behaviour of a large network of neurons. This approach is based upon the idea that the computational characteristics of the brain arise from the properties of large networks of neurons, rather the detailed features of

single cells. When a single cell model is simple, it is possible both to simulate and mathematically analyse the behaviour of the model. Recent advances in both neuroscience and computer science have made it possible to put forward reasonably plausible models of biological associative memory. One of those is *the associative network* (Willshaw, Buneman, and Longuet-Higgins 1969).

1.2 Associative networks

Associative networks are very diverse in both structure and function. In this section, I describe the fundamental structure of the general class of associative networks studied in this thesis. More technical or historical issues are described later in this chapter.

1.2.1 Network components

An associative network is a collection of simple computational devices (i.e. artificial neurons or simply, cells) connected in a well defined way. A cell receives many synaptic connections from other cells and provides an output that projects and makes synaptic contact with other cells in the network. If a connection exists from a cell to another cell, there is only one point of connection. In other words, one cell sends its output to another cell by a single point of synaptic contact. (See figure 1.1.)

Each connection has a weight, a signal transmission efficacy, which is modifiable. In this case, the synapses are binary — they are either *effective* (pass the output of the cell through the synapse) or *ineffective* (does not transmit the output to the contacted cell). Cells operate in two modes — at any point in time, they can be either active or inactive. When the cell is active, the cell generates an output of value one that is then conveyed to the cells connected with effective synapses. When the cell is inactive, connected cells are

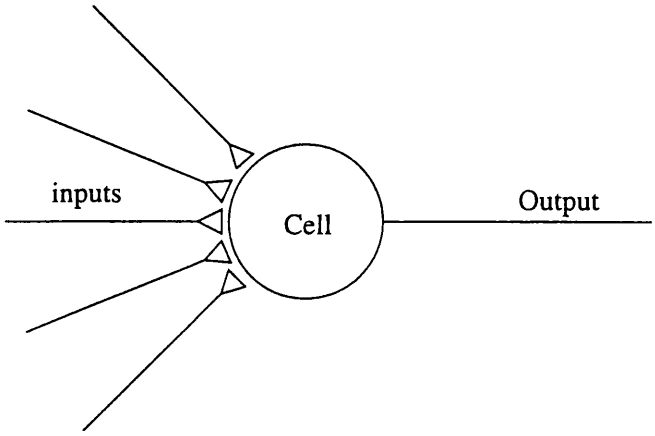


Figure 1.1: A schematic diagram of a cell in the associative network

not affected as the cell produces no output.

At each point in time a subset of the cells in the network is active. Each of the effective inputs from active cells to the cell are summed up to represent the internal potential of the cell. This value is called the *activation* of the cell or the *net input* to the cell. The output of the cell is determined by the value of the activation. In other words, the output is a function of the activation. This function is called the *transfer function.* In the associative networks studied in this thesis, the transfer function is the binary threshold function. That is, when activation is below the threshold value, the output becomes zero and when activation is above threshold the output becomes one (i.e. the cell becomes active).

The cell type described so far is referred to as a *principal cell*, for the reason described in the next subsection. The threshold of the cell is determined by another class of cell called the inhibitory interneuron. In the models described in this work, only one inhibitory interneuron is present in the network. The interneuron receives effective connections from every principal cell and in turn projects back to each principal cells with a fixed synaptic weight. Unlike the

principal cell, the output of the interneuron is graded (not binary). The input from the interneuron is subtracted from the activation of a principal cell and if this result is greater than zero, the principal cell becomes active. Therefore, the interneuron can be regarded as a device that measures the overall activity of the network and in turn applies a global threshold (i.e. every principal cell will have the same level of threshold) to the network of principal cells.

1.2.2 Network functions

As its name implies, an associative network is capable of storing and recalling memory *events*. Events are coded by the pattern of activation of the cells. For example, when there are N cells that can be used to participate in the storage of events, each event is represented as an N-dimensional binary vector (a string of length N consisting of zero and one). Those events consist of input events and output events, that are paired to make an association. The event coding cells are the principal cells.

The associative network operates in two modes — a learning mode and a recall mode. During the learning mode, each of the input events and the corresponding output events are simultaneously presented to the network. Then, the efficacies of the synapses are modified according to a *synaptic learning rule*. It is this process that associates the input event to the output event. It is important to note that synaptic efficacy modification is the only change made to the network during the learning process.

The learning rule used in the associative network is quite simple. Principal cell synapses are initially set to be ineffective. During the learning process, all of the synapses between active presynaptic principal cells and active postsynaptic principal cells are permanently made effective. This synaptic learning rule is sometimes referred to as the *clipped Hebbian rule*. Unlike some other

neural networks (such as backpropagation (Werbos 1974; Rumelhart et al. 1986) or reinforcement learning networks (Sutton 1984)), associative network learning occurs from one presentation of a memory pattern.

In recall mode, the network converts a partial or corrupted input into the corresponding stored pattern, through the operation of the globally applied threshold.

At any point in time there are two types of cells active. The cells that are a subset of the event being cued for recall are called *correct cells*, and the cells that are not part of this event are called *spurious cells*. Associative networks are useful because they can recall a stored pattern from a small subset of the pattern, and the performance is not greatly degraded if this subset includes a small number of spurious cells. In this way, associative memory is robust when confronted with the type of noise that is expected in biology.

When input events differ from output events the network is said to be *heteroassociative* and when input events are exactly the same as output events the network is *autoassociative*. Heteroassociative memory can be regarded as a "pattern converter" as in the input pattern is converted to the corresponding output pattern by the system. In contrast, autoassociative memory functions as a "pattern recognition device" in that a learned memory event is recalled from a corrupted version of the event (i.e. pattern completion). This thesis is concerned with autoassociative networks.

1.2.3 Network layout

There are two classes of principal cells that are used to code events for heteroassociative networks; one class of cells is used to handle input events and the other to handle output events. In heteroassociative networks the two classes of cells are organised in two layers of cells. A typical model of this

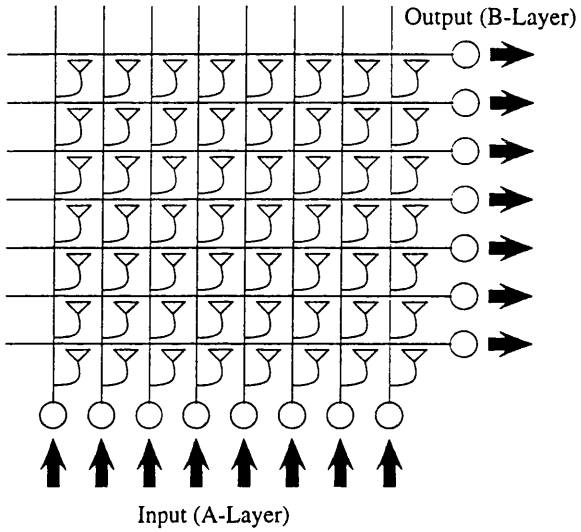


Figure 1.2: A heteroassociative network proposed by Willshaw *et al.* (1969) — the network consists of two layers of cells. The input layer (A layer) is exclusively for processing input events and the output layer (B layer) is exclusively for output event. Each output cell is contacted by every input cell hence establishing 100% connectivity within the network. All of the synapses are binary (0 or 1) and the output of the cell is binary. After a learning of associations takes place (not shown in the diagram), on the presentation of a learned input event, the associated output event is reproduced as a result of taking an appropriate global threshold.

sort is the Willshaw network (Willshaw et al. 1969) (see figure 1.2 for further explanation).

Autoassociative memory can be implemented by presenting the same events to both input and output layers of a heteroassociative network. In this way, the size of the output layer and the input layers are the same (figure 1.3 A). Alternatively, the input layer and the output layer can be unified into one layer since they both handle the same events. The synaptic connection among cells can be made in a recurrent (feed back) fashion so that cells in the network are interconnected to each other. The latter structure will be studied in this thesis as it has some advantages. First, it reduces the number of cells needed to form a network. Second and more importantly, while the former model only allows single stepped memory retrieval (i.e. thresholding is done only once), in the latter model results retrieved from the network can be reapplied to the network to do further recall. The first type of recall is referred to as *simple recall* and the latter as *multi-step* or *progressive recall.* Let dash ($'$) represent a degree of signal corruption. In multi-step recall, it is possible to recover a learned event A from a very corrupted (A'''') by gradually completing the pattern step by step ($A'''' \longrightarrow A''' \longrightarrow A' \longrightarrow A.$), whereas in simple recall A''' could be recovered only to A''.

While the Willshaw network (figure 1.2) is fully connected, the associative network studied in this thesis is partially connected. In fully connected networks, each principal cell makes synaptic contacts to all of the principal cells in the network (Gardner-Medwin 1976). In the partially connected networks studied in this thesis, each principal cell makes a synaptic contact to a fixed number (R) of randomly selected cells in the network. The connectivity of the network is then expressed as R/N^2 , where N is the total number of the

[2]To be preciese, the connectivity is $R/(N-1)$ as there is no self-feedback connection.

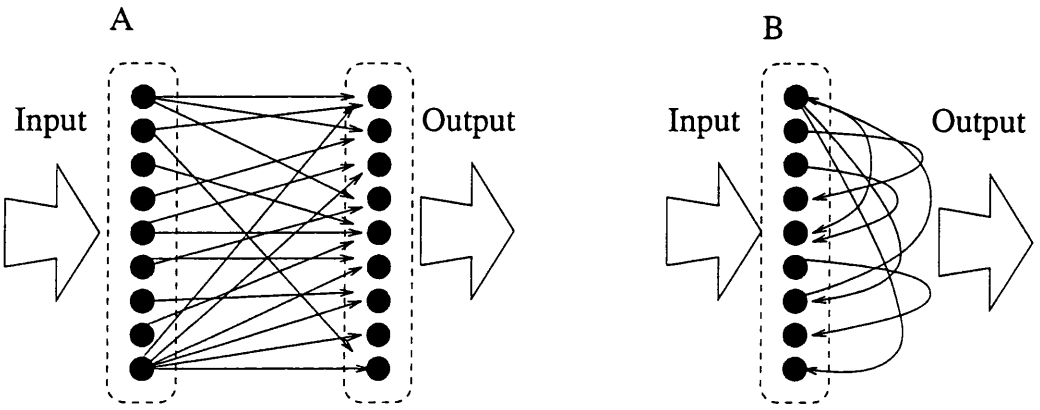


Figure 1.3: Two methods to implement autoassociative memory : **A** Two layer feedforward model — There are input layer and output layer dedicated to input events and output events, respectively. Since it is an autoassociative memory, the input event and the output event are identical in learning mode. **B** One layer feedback model — input layer and output layer are unified. See text for more description.

principal cells in the network.

1.2.4 Summary

In summary, the auto-associative network to be analysed in this thesis is recurrently and partially connected with binary (zero or one) synapses (figure 1.4). The principal cell produces binary (zero or one) output as a result of global threshold (T) applied to the network. Each principal cell makes a synaptic contact to R randomly selected cells. Initially, all connection weights are set to be ineffective (zero). During the learning phase, M binary patterns of size N are presented to the network. Each pattern includes W active (i.e. one) and $N - W$ inactive components. Each component of a pattern corresponds to the state of the principal neuron. In the presence of a pattern, synaptic modification occurs using the clipped Hebbian rule.

For large N, the connectivity can be approximated to R/N. This approximation is assumed for the rest of the thesis.

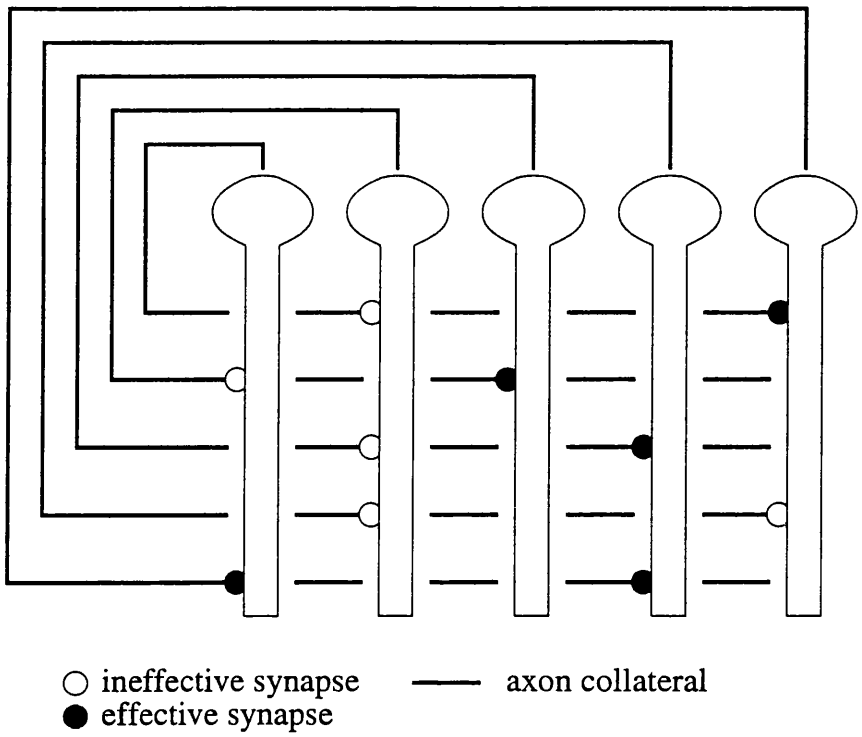


Figure 1.4: A diagram of a partially connected recurrent autoassociative network.

In the recall phase, a small portion of active components in a pattern is presented. The network then tries to complete the rest of the pattern through the multi-step (progressive) recall process, in which the newly activated pattern as a result of global thresholding is then reapplied to the network in each recall step. The mathematical symbols used in this thesis are summarised in appendix A.

1.3 Historical aspects

Several neural network models of associative memory have been proposed and the performances of these network models have been studied. These models can be categorised into the following three types:

1. Feedforward model- This is the original form of heteroassociative memory in which there are two layers of neurons, one for input events and one for output events. Cells in the input layer project to cells in the output layer, but there is no projection from the output layer to the input layer. Since there are only feed forward projections in the network, the recall process occurs in a single threshold step.

2. Symmetrical feedback model - This is a form of autoassociative memory in which cells are organised in one layer. The network is usually fully connected. The network connection is characterised by the symmetrical connection in which the synaptic weight from cell i to cell j is the same as the synaptic weight from cell j to cell i. This type of model has become popular since its dynamics is mathematically well defined.

3. Asymmetrical feedback model - This is another form of autoassociative memory in which cells are organised in one layer. The network studied in this thesis belongs to this category. The network is not necessarily fully connected and the synaptic connections are not symmetrical. This type of model has gained some popularity as the network structure resembles some biological aspects of brain structure including the CA3 region of the hippocampus.

In this section, historical aspects of the neural network models that function associative memory are reviewed according to the above categories.

1.3.1 Feedforward model

The associative network was developed to model the mechanisms of distributed memory (both storing and recall) using a relatively simple neuronal network. The heteroassociative network was originally developed by Steinbuch (1961),

and later rediscovered from the mechanism of holographic memory by Willshaw et al. (1969). As described in figure 1.2, it consists of binary threshold neurons organised in two layers with full connectivity. In the learning mode, synaptic modification occur using the clipped Hebbian rule described in the previous section. In the recall mode, upon the presentation of a learned pattern to the input layer, the associated pattern is produced in the output layer.

The feedforward model with real valued synapses was put forward by Kohonen (1972). In this model, so called correlational learning has been employed as the synaptic modification rule. Starting from the initial value of zero, the synaptic weight from cell i to j is changed according to the following rule.

$$w_{ji} = c \sum_{k=1}^{M} x_j^k x_i^k$$

where c is a constant, M is the number of memory events learned by the matrix, x_j^k denotes jth component of kth learned memory events.

The capacity analysis has been studied for the feedforward models. There is a general agreement that for the fully connected Willshaw network, the theoretical capacity of the network is $\sim ln2 \simeq 0.693$ bits per synapse (Willshaw et al. 1969; Palm 1988; Nadal and Toulouse 1990), which is quite close to that of the real-valued synapse, $1/2ln2 \simeq 0.721$ (Gardner 1987). However, in order to achieve the optimal capacity, very sparse encoding has to be used. Nadal and Toulouse (1990) calculated the capacity of real valued synapses with binary event encoding and additive Hebbian learning, which is a special case of the correlational rule with c=1, to be $1/\pi ln2 \simeq 0.459$. Optimal synaptic learning rules for these kinds of network have been studied by Palm (1988) and Dayan and Willshaw (1991).

The feedforward model with partial connectivity has been studied by Buckingham (1991) and by Buckingham and Willshaw (1993). They demonstrated

that the value selected for the global threshold critically affects the performance of retrieval in partially connected networks. They have shown that local thresholding (i.e. a threshold value that differs from cell to cell) based on the ratio of effective inputs to total inputs results in higher recall performance than global threshold. Such a thresholding method is called the division threshold and was suggested earlier by Marr (1971). No evidence has yet been found that this method is used in real biology. Based on the statistics of the synaptic activity, Buckingham and Willshaw (1993) proposed a thresholding strategy that efficiently separates correct cells from spurious cells. However, this strategy requires mathematical manipulations that are not likely be found in real neurons. Recently, a thresholding strategy for a partially connected model was proposed by Graham and Willshaw (1995). The key point of this strategy is that input to the cell is transformed with a function that takes account of the unit usage (i.e. the number of times the cell has participated in learning) and the total number of active input cells.

1.3.2 Symmetrical feedback model

Ever since a symmetrical feedback model was introduced by Hopfield (1982), the analysis and simulation study of the model have been popular among physicists and mathematicians. The network is comprised of binary neurons that have +1 or -1 output. The cells are fully connected to each other with real valued synapses. The initial model (Hopfield 1982) demonstrated that when the cells are connected with symmetric weights , the activity of the network converges to an attractor state through time. It has been demonstrated that the correlational synaptic learning rule makes the learned events act as attractor states of the network. Cells are updated in a asynchronous manner so that only one cell is updated at each time step. A symmetrical feedback

model with synchronous update (all cells are updated at each time step) has also been proposed before (Little 1974; Little and Shaw 1975).

The network can function as an autoassociative memory since the presentation of a slightly distorted learned event can lead to the recall of the complete event through time. Similar properties have been proposed for neurons with a graded response (sigmoid input-output relation) with continuous time (Hopfield 1984).

The analysis of the network is refined by constructing the Lyapunov function that represents the energy of the network (Hopfield 1982; Cohen and Grossberg 1983). Attractor states are the local minima of the Lyapunov function. It has been demonstrated that asynchronous update, which is used in the Hopfield model (Hopfield 1982)) guarantees convergence to an attractor state and synchronous update can result in an attractor state or a limit cycle consisting of two states (e.g. Hopfield 1982; Amit 1989; Marcus and Westervelt 1989; Marcus, Waugh, and Westervelt 1990). The dynamics of the symmetrical feedback model resembles that of Ising spin glass model. Hence the network is referred to as "spin glass model".

In these networks, it has been generally found that the number of uncorrelated patterns that can be stored in a network is proportional to the number of synapses per neuron in the network. The proportionality constant is calculated to be $0.14 \sim 0.16$ for memory events consisting of half active (+1) and half inactive (-1) components (Amit, Gutfreund, and Sompolinsky 1985; Gardner 1987; Amari and Maginu 1988; Amit 1989).

Although the analysis of the feedback symmetrical model is well established, the main criticisms of the model come from biologists. First, there is no known neuron that based on the input to the cell, can produce both an in-

hibitory signal and an excitatory signal. Secondly, in many if not all biological systems, synaptic connections are asymmetrical and the connectivity is very sparse.

1.3.3 Asymmetrical feedback model

Since most of fully connected autoassociative models are also symmetrically connected, asymmetrical models are usually partially and randomly connected feedback models. Neuronal networks with asymmetrical feedback connections are observed in biology, such as the CA3 region of the hippocampus (Lorente de Nó 1934) and the piriform cortex of the olfactory system (Haberly and Price 1978a,b).

Partially connected recurrent networks have become popular among theorists and modellers since Marr (1971) suggested that this type of neuronal network might exist in the hippocampus and function as an autoassociative memory. Marr's model of memory consisted of three layers of neurons; P1, P2 and P3 (figure 1.5). Cells in P1 provide feedforward projections to the P2 layer and the P2 layer in turn projects to the P3 layer. The neurons in the P3 layer are recurrently and partially connected and can function as an autoassociative memory. In Marr's model the P1 and P2 layers represent cortical cells and the P3 layer represents the hippocampus. The model has recently been reviewed and reassessed by Willshaw and Buckingham (1990) and it has been demonstrated that the model could in fact function as an associative memory, despite the fact that the original model proposed by Marr does not crucially depend on the use of recurrent collaterals.

The network described in section 1.2 was first proposed by Gardner-Medwin (1976), which is an explicit model of the P3 layer in Marr's model. The network consists of N binary principal neurons and an inhibitory neuron. Each

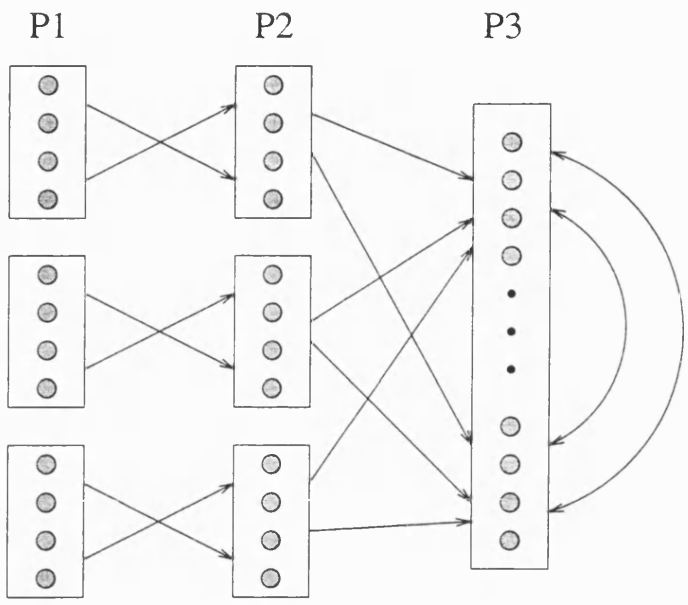


Figure 1.5: The three-layer model analysed by Marr(1971) [redrawn from Willshaw and Buckingham (1990)].

of the principal neurons sends a collateral to R randomly selected other principal neurons. The synaptic connections have binary weights modified with the clipped Hebbian rule. The interneuron controls the value of the global threshold T. The network is assessed with *seeded recall*, in which a small portion (typically 10% of the correct cells) of a learned event is presented and maintained to be active while the network completes the rest of the pattern. Gardner-Medwin demonstrated that progressive recall results in better performance than single stepped simple recall, and also that precise control of the threshold is essential to achieve successful progressive recall (Gardner-Medwin 1976). This threshold control is discussed in more detail later in this chapter (section 1.4.7).

Gibson and Robinson (1992) formalised the dynamics of progressive recall and derived a set of progressive recall equations. These equations are used

to statistically estimate the expected number of active cells from the previous sequence of network states. The progressive recall equations were verified with the simulation results of incompletely connected networks (50% connectivity) (Gibson and Robinson 1992). This analysis is reviewed in appendix B. The theory put forwarded by Gibson and Robinson has also been applied to the CA3 region of the hippocampus with a biologically realistic connectivity pattern (Bennett et al. 1994). They reported that the realistic inhomogeneous connectivity pattern slightly improves the storage capacity, but the result has not yet been verified through simulation.

Sparsely connected networks containing neurons with graded response (as opposed to binary), representing the firing frequency of a neuron, have been studied in detail by Treves and Rolls (Rolls and Treves 1990; Treves 1990; Treves and Rolls 1991,1992,1994). In this work, the neuron is a threshold-linear device, i.e. the output of the neuron is defined as follows:

$$\begin{aligned} V &= g(h-T), h > T \\ &= 0, \text{ otherwise} \end{aligned}$$

where h is the total input to the cell, T is the threshold to the cell, and g is a positive constant. They defined a measure of coding sparseness as:

$$a = \frac{\langle r \rangle^2}{\langle r^2 \rangle}$$

where r denotes the firing rate of each cell in the stored patterns and $< \cdot >$ is the expectation value.

The maximal number of stored patterns per recurrent synapses for each cell is found to be proportional to $\frac{1}{aln(1/a)}$, with a proportionality constant roughly in the range of 0.2 to 0.3. Since the activity of a single neuron can take a positive real value, the formalism can be used to compare different

radix encodings. In fact, they have compared binary encoding with ternary and continuous encodings. The capacity analysis revealed that information storage per cell is higher for binary encodings by a factor of 1.5, when sparse coding is assumed.

Recently, Minai and Levy have discussed thresholding strategies in sparsely connected recurrent networks (Minai and Levy 1994), but their objective is to stabilise the total amount of activity of the networks and the work did not treat the network as an associative memory device. Instead, the work concludes that it is possible to maintain a sparse activity in the sparsely connected network if an appropriate threshold is applied.

Formal capacity analysis for fully connected recurrent networks with McCulloch-Pitts neuron (zero or one output, real valued synapse) was studied by Palm and Sommer (1992). Their calculation revealed that the upper bound capacity limit of networks with clipped Hebbian learning is 0.693 bits/synapse when some active spurious cells could be included (the critical pattern capacity) and 0.347 when absolutely no spurious cell is included (information capacity). For the additive Hebbian learning, the critical pattern capacity is 0.347 and 0.173, respectively. However, the capacity analyses (including the work by Treves and Rolls) are based on the stability of the network activity when the complete memory event and therefore the capacity is different from that of performing seeded recall.

1.4 Key issues

This section is concerned with the issues that are central to the understanding of associative networks. Included in the key issues are the measure of performance, the measure of capacity, the encoding scheme, the capacity of various

1.4.1 Measure of performance

In order to make a comparison between any two associative networks with different configurations, it is essential to define a measure that quantifies the performance of the networks. In addition to Buckingham's (1991), measures of performance are compared and discussed in this section.

The simplest way that the performance of a network can be measured is to define the number of memory events that can be stored and recalled. Loading of the network can be assessed with error free recall (i.e. the recalled result exactly matches the target event) or allowing some error in the recalled result. In the latter case, it is necessary to understand how to measure the way in which the size of the error changes.

The error level is the measure that quantifies how much the recalled result differs from the target event. The simplest measure of error can be computed by counting the number of elements in the recalled events that do not match the target event, i.e. the Hamming distance. The Hamming distance is a simple and intuitive way of measuring the degree of discrepancy between two patterns. However, one drawback is that false negatives and false positives are treated equally, that is, the inconsistency for correct cells carries the same weight as the inconsistency for spurious cells. This is acceptable when the number of correct cells is the same as the number of spurious cells (i.e. $W = N/2$). However, when the number of correct cells are disproportionate (i.e. sparse coding), this is not a desirable property.

For example, suppose the number of principal cells is 1000 ($N = 1000$) and the number of active cells in each event is 100 ($W = 100$), and the recalled result contains 50 correct cells and 50 spurious cells. The Hamming distance between the recalled events and the desired output will be 100, which is the

same as the Hamming distance between the complete event and the state that have no active cells at all.

Two alternative measures have been proposed: For a recalled result $\mathbf{x} = [x_i]$ and the desired target event $\mathbf{y} = [y_i]$, the degree of *overlap* is defined:

$$overlap = \frac{(\mathbf{x} - \overline{\mathbf{x}}) \cdot (\mathbf{y} - \overline{\mathbf{y}})}{\|\mathbf{x} - \overline{\mathbf{x}}\| \, \|\mathbf{y} - \overline{\mathbf{y}}\|}$$

where $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ denote vectors whose elements are the average activity of $\mathbf{x}$ and $\mathbf{y}$, respectively, i.e. $\overline{x}_i = \sum \mathbf{x}_i/N$ and $\overline{y}_i = \sum \mathbf{y}_i/N$, for $1 \leq i \leq N$.

In other words, the overlap is defined as the cosine of the angle between $\mathbf{x} - \overline{\mathbf{x}}$ and $\mathbf{y} - \overline{\mathbf{y}}$. This measure (with some variations) has been used in several studies (e.g. Gibson and Robinson 1992; Bennett et al. 1994; Graham and Willshaw 1995).

An alternative measure of error is based on information theory and was proposed by Gardner-Medwin (1989). According to information theory, when a binary random variable X has a probability p of becoming one, the uncertainty of the random variable can be defined in terms of Shannon entropy (Shannon 1948), i.e.:

$$H(p) = -plog_2p - qlog_2q \quad \text{(bits), where q=1-p}$$

Specifying the content of X correctly implies that the uncertainty ($H(p)$) is removed. Hence this is equivalent to gaining information amounting to $H(p)$ bits.

Applying this concept, each element in an event has the uncertainty of $H(W/N)$ when investigated independently. $H(W/N)$ bits of information will be required to successfully specify an element in the event. There are N of those elements. Therefore the amount of information gained by specifying the

whole event from scratch (I_0)is:

$$I_0 = NH\,(W/N)$$

When a recalled result contains w active cells of which c are correct cells and s are spurious cells, the amount of information required to correct the inconsistency (I_0)is:

$$I_c = wH(s/w) + (N-w)H\left(\frac{w-c}{N-w}\right)$$

The amount of information in the pattern that has been recalled by the network is $I_0 - I_c$. The *quality of recall* (Q) is defined to be the ratio of the amount information in the recalled result to that in the uncorrupted event that was initially stored in the memory:

$$Q = \frac{I_0 - I_c}{I_0}$$

Both of these measures take the range [0,1] and become 1 when the recalled result exactly matches the learned event. The measures can be computed by calculating how different types of error in the recalled pattern are treated. Comparisons between these two measures are shown in four graphs in figure 1.6 with $N = 1000$ and $W = 100$.

In figure 1.6A, both measures are plotted for recalled events that have a varied number of correct cells, but have a fixed Hamming distance of 30 from the stored memory. The overlap measure produces a flat and smooth curve taking values between [0.81,0.86]. The quality measure is a more concave curve rising as the number of correct cells increases. This means that the overlap measure gives (almost) the same values for events which have the same Hamming distance. The quality measure, in contrast, assigns a higher value for those events that have higher number of correct cells.

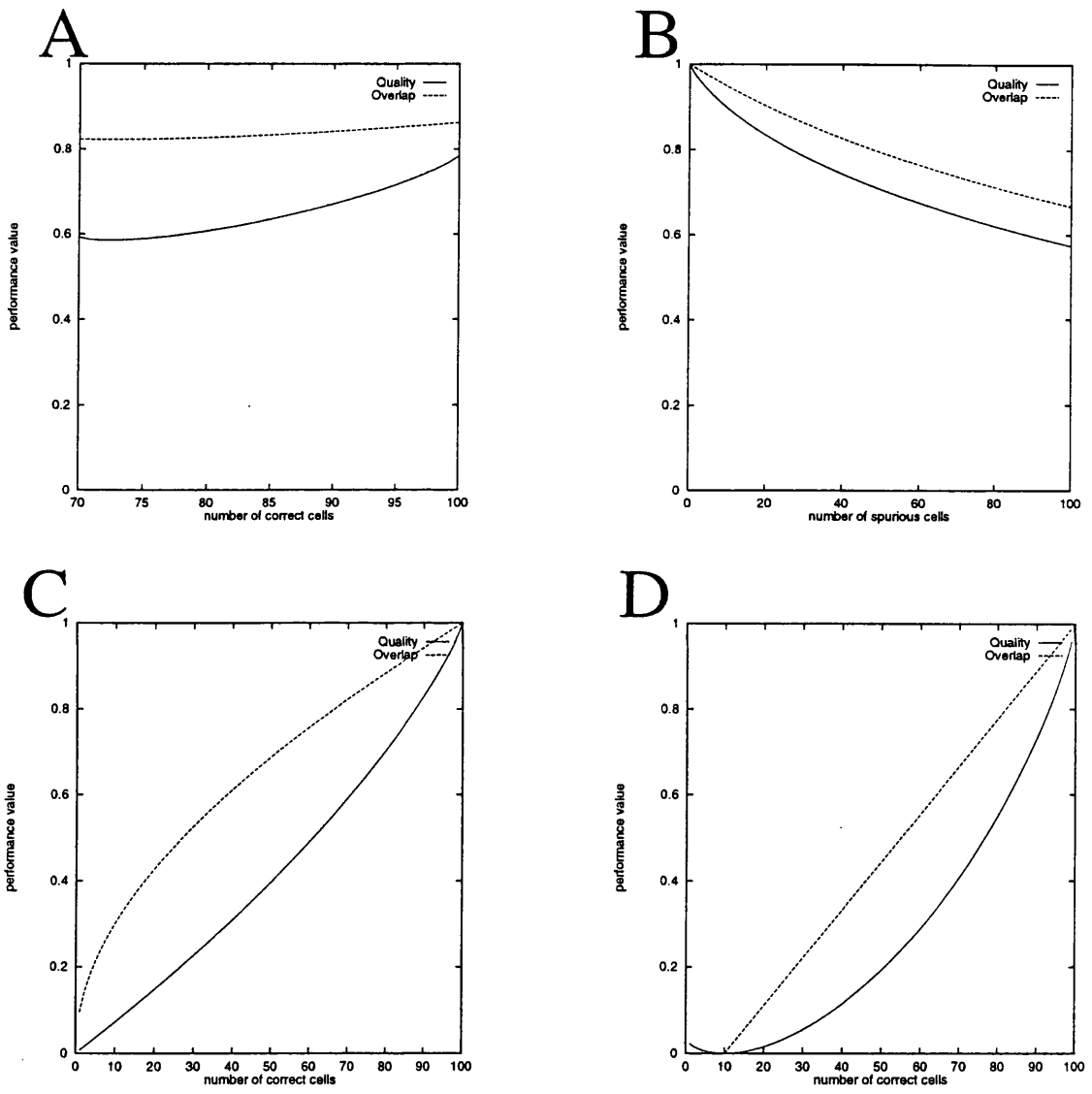


Figure 1.6: Comparisons between overlap and quality measures. Various comparisons are made with the configuration $N = 1000$ and $W = 100$. **A**: The Hamming distance from the stored event is always kept to be 30. **B**: Adding spurious cells to a perfect stored event. **C**: Removing correct cells from a perfect recall. **D**: The number of active cells (active correct cell + active spurioous cells) are kept to be 100.

In figure 1.6B the effect of adding spurious cells to a perfect event is shown. In this plot, both the quality measure and the overlap measure show a similar tendency; they drop smoothly as the number of spurious cells is increased, which is the desired property of the measure. The quality measure drops slightly steeper than the overlap measure, indicating that the quality measure is more sensitive to the presence of spurious cells.

Similarly, figure 1.6C shows the effect of adding correct cells to a null event (no active cells). Here, the quality measure is almost linear in the number of correct cells. The overlap measure, on the other hand, is a smooth convex curve. It can be interpreted that the quality measure is (almost) uniformly sensitive to the number of correct cells while the overlap measure is more sensitive when there are fewer correct cells.

Finally, in figure 1.6D, the two measures of quality are plotted for recalled events that have the same number of active cells (W), but the fraction of the cells that are correct cells is varied. In this case, the overlap measure is a linear function with slope 1/90 and intercept -1/9. The quality measure is a concave curve, which has the effect of emphasizing the presence of correct cells.

Overall the quality measure, based on information theory, fits more closely to the requirements for measuring the recall performance. For this reason it was selected as a way of quantifying the recall performance in this thesis.

1.4.2 Measure of capacity

When considering a memory system of any sort, the capacity is one of the most central issues, and associative memory networks are not an exception. Capacities of associative networks have been discussed since the emergence of the models. This section summarises the results of studies of the capacity of associative networks.

As Palm and Sommer (1992) suggested, there are a few ways to assess the capacity of an associative network. The simplest is to count the number of stored (and recallable) events. This method is meaningful when comparing the capacity of the networks that differ by only one feature, e.g. two structurally identical associative networks that have different thresholding strategies or two structurally and operationally identical associative networks with different activity ratio (W/N). However, when comparing networks that differ in more than one feature (e.g. two networks with different size and activity ratio), the measure becomes suddenly pointless as it is hard to systematically identify which component contributed to the change in the count.

The shortcoming of using the number of stored events as a measure of capacity is that it does not consider the inherent nature of coding. Intuitively, more sparse (W/N is small) events can be stored than dense events. However, it is also intuitive that a sparse event contains less information. This suggest that the information content of an event should be incorporated in the measure used to evaluate the capacity.

As mentioned in the previous section, it is possible to calculate the information content of an event (I_0). If M_{max} is used to denote the maximum number of events that could be stored in an associative network, $M_{max}I_0$ is *the total stored information* and this can be used as a capacity measure.

This measure of the total stored information can be used to compare the capacities of associative networks of the same size, but it is not convenient for comparisons involving networks of different sizes. A measure that overcomes this inconvenience is the *stored information per cell* ($= M_{max}I_0/N$). However, it is important to note that storage of information occurs because there is a change in synaptic efficacy. That is to say that cells are not the source

of storage, but rather synapses are. Therefore, it is reasonable to assess the capacity of the network by calculating the information stored per synapse. Information per synapse ($= \frac{M_{max} I_0}{NR}$) is also useful for comparing networks that have different connectivities as the measure is concerned with how much information can a synapse store and therefore topology of the network can be ignored.

It is useful to note that for a binary synapse, information per synapse represents the *storage efficiency* as well. Since a binary synapse can have only two states (i.e. on or off), the maximum information it can store is one bit. The efficiency of a synapse is the information stored per synapse divided by the maximum information it can store (=1 bit).

In this thesis, the capacity of associative networks will be evaluated with this measure of the information stored per synapse measure.

1.4.3 Measure of loading

The fraction of synapses that have changed from zero to one is a useful general measure of how near an associative memory is to its capacity limit. This is often approximated by the expectation value of the probability that a synapse has been modified (ρ). After a network consisting of N cells has learned M events, each containing W active cells, ρ can be calculated as follows (see Appendix B for more detail.)

$$\rho = 1 - (1 - W^2/N^2)^M$$

For feedforward networks, the maximum capacity is achieved when ρ is 0.5 (Willshaw et al. 1969; Nadal and Toulouse 1990). For a partially connected recurrent network, the recall performance degrades rapidly as ρ approaches close to 0.5 (Gardner-Medwin 1976). However there is no guarantee that an

associative memory network will operate up to this apparent limit.

1.4.4 Event encoding

In the previous section, the significance of sparseness of coding was briefly mentioned. Amari and Maginu (1988) studied fully connected networks that learn randomly generated binary patterns. They presented a mathematical interpretation of the dynamics of the neural activities observed in computer simulations. It was stated that memory retrieval is achieved when the cell activity monotonically approaches the pattern being recalled. By identifying initiating patterns that do not approach to the basin of attraction, both absolute and relative capacity was calculated . Later, this study was extended to show that the network behaves far more efficiently when the learned patterns are sparsely encoded (Amari 1989).

The advantage of sparse encoding has also been discussed previously by Willshaw et al. (1969) and Palm (1980, 1981). Treves and Rolls assessed the capacity of autoassociative memories for different learning rules and architectures and concluded that sparseness of information encoding has a definite advantage (Rolls and Treves 1990; Treves and Rolls 1991)

1.4.5 Recall Paradigm

As briefly stated in section 1.2, there are two distinct types of recall process in autoassociative networks called simple recall and progressive recall. The idea of progressive recall was proposed by Marr (1970), and Gardner-Medwin (1976) demonstrated that progressive recall often results in superior performance in comparison to simple recall.

As mentioned in section 1.2, the paradigm used to assess the performance of associative memory networks in this research is *seeded recall.* In seeded recall,

there are two ways in which the initial cue is treated. The first of these is called *seed-on* recall, in which the initial cue remains persistently active. The other paradigm is called *seed-off* recall, and the recall cue is presented only during the initial step of the recall. Aside from seeded (seed-on or seed-off) recall, there are other types of tasks that can be assessed using an autoassociative memory. One of these is a *recognition* task. In this task, cells representing the complete event together with a number of spurious cells are activated and the network is used to remove the spurious cells without losing the correct cells.

This is somewhat arbitrary as there hardly is any experimental work that shows how much information is used as a cue to lead to a successful recall of memory. Marr (1971) chose to activate 10% of relevant cells of a learned event to initiate recall.

1.4.6 Connectivity

Connectivity is an important issue for at least two reasons. First, it is important to note that biological systems are not fully connected, rather, the connectivity is very sparse (e.g. Ishizuka, Weber, and Amaral 1990). As we are trying to model a function of the brain, it is reasonable consider how associative memory could be achieved in incompletely connected networks. Second, when implementing associative memory in electronic hardware (e.g. a VLSI implementation of an associative network), connections between simulated neurons occupy a large fraction of the chip area and limit the size of the network.

Following Graham and Willshaw (1995), a *fully connected* network is defined as a network where every input cell has a synaptic contact to every output cell. In the case of recurrent networks, "fully connected" implies that a cell makes a synaptic contact to every cell but itself. In a *partially connected*

network, some of the connections between the cells are not established. Often the connectivity of a partially connected network is around 50%. *Sparsely connected* networks are a subset of partially connected networks, but the connectivity is generally much lower. The connectivity of sparsely connected networks is usually 10% or lower.

In some cases when precisely modelling specific neuronal structures, the probability of connection is not uniform throughout the network. Bennett et al. (1994) varies connection probability according to the cell's relative spatial location. As a result they reported a slight capacity increase.

1.4.7 Thresholding Strategy

Thresholding is one of the most crucial operations in both simple recall and progressive recall of partially connected associative networks. For example, if the threshold value is set too high the activity of spurious cells will be inhibited, but the activity of the correct cells might also be surpressed. The opposite applies as well. One might set the threshold level low in order to activate many correct cells, and end up with too many active spurious cells.

In fully connected Willshaw nets without spurious cells in the cues, the thresholding strategy is simple. If the number of active cells in an input event is constant, the threshold should be chosen to be this number, and the recall will be optimal. In this case each correct output cell is guaranteed to receive an input activation that is exactly equal to the number of active input cells. This strategy is also called the winner-take-all (WTA) strategy (Willshaw 1971).

Marr (1971) stated that the neuronal system could employ two types of thresholding scheme. In *subtractive thresholding*, the activation of the cell (where the physiological analogue is the membrane potential) has to exceed a threshold value in order to be active. *Division thresholding* is a scheme where

the ratio of the effective synaptic input and the total synaptic input (before multiplied by the synaptic weights) of a cell has to exceed a certain threshold value in order for the cell to be active. Marr argued that in order to achieve a successful recall, these two thresholding schemes must be used together. That is, a cell has to exceed both the subtractive threshold and division threshold.

Buckingham (1991) and Buckingham & Willshaw (1993) calculated the threshold that minimised the probability of spurious cell firing and maximised the probability of correct cell firing on a cell by cell basis for partially connected Willshaw nets. The thresholding strategy (guess-s), involves complex mathematical computations and relies on detailed knowledge of network parameters. For this reason, it is hard to imagine that a biological system that employs this thresholding strategy.

While thresholding methods and strategies for single step recall have been investigated intensively, thresholding strategies for progressive recall have received less attention. The thresholding strategy employed by Gardner-Medwin (1976) is an orthodox one. Given the dendritic input probability distributions for both the correct cells and the spurious cells, the threshold value is chosen to be as small as possible, while keeping the expected value of spurious firing less than one. [3]

Progressive recall in the Gardner-Medwin type network was further studied by Read and his colleagues (Read, Nenov, and Halgren 1994). They experimented with the network by modifying it in several ways. First, they introduced a constant baseline (tonic) threshold together with the conventional activity based threshold. This had little effect on the recall process, but they argued that it is biologically more plausible. They also modified the recall

[3]Buckingham (1991) points out that the probability distribution Gardner-Medwin used is erroneous.

process to one in which the seed cells (or cue) are only presented at the initial stage of the recall. This approach was taken in order to use memory cues that included spurious cells. If the seed cells remain active throughout the recall, the noisy component will also remain active and as a result the recall may fail or contain more errors. Finally, they introduced a delayed component in the inhibitory mechanism. The strength of inhibition depends not only on the current network activity but also on the activity of prior cycles. They mentioned that the delayed inhibition had the effect of damping the network activity to null, and hence resetting the recall process.

A similar network, although fully connected, was studied by Lansner and Ekeberg with different thresholding strategies (Lansner and Ekeberg 1985). In the first thresholding strategy, the inactive cell that has received the strongest effective inputs is recruited to be active. In the case that there is more than one cell that receives the highest effective input, a cell is chosen randomly from this set. Cells that are already active remain active in the next timestep. In the second strategy, a cell is recruited at a time as explained in the first strategy. However, instead of automatically maintaining the previously active cells in the network, these prior active cells must have sufficient effective inputs to meet each of the threshold limits throughout the progressive recall process.

1.5 Concluding notes

In this chapter, I have summarised the background information of associative networks in various aspects. Having done so, there are many problems yet to be addressed. For example, associative networks assume synchronous parallel update in discrete time steps. This might not be the case in the biological systems. It is quite possible that the action potentials responsible for causing

a neuron to fire arrive at different times and the timing of spike arrival might play an important role. Another unsolved question is whether the biological neurons are binary coded or instead are coded with a larger number of values, using frequency coding (number of spikes per unit time). All of these changes to the associative memory networks might result in a higher memory storage capacity.

Among those unsolved (or not yet posed) questions, thresholding strategy in progressive recall, the method to choose an appropriate sequence of threshold values ($T(t)$) to achieve a good recall is the central theme of this thesis. As mentioned earlier in this chapter, choice of threshold values critically affects the retrieval performance in multi-step recall of partially connected recurrent networks. One of the main issues addressed in this thesis is identifying the optimal thresholding sequences in the partially connected recurrent associative network so as to increase the storage capacity of the network. When the number of recallable pattern approaches to the maximum, the thresholding path is said to be optimal. Once the optimal thresholding sequence is identified, the algorithm to produce the threshold must be sought. It is well worth comparing the algorithm with possible mechanisms of the biology of the brain and see how descriptive a simple model can be. Also, once the optimal thresholding strategy is obtained, the recall performance using the strategy can be taken as a benchmark to compare with the models with extra level of complexity.

The rest of the thesis is organised as follows. In chapter 2 and 3 the methodology of identifying optimal threshold sequences is explored. Based on this methodology optimal thresholding sequences are identified in chapter 4. Also in this chapter, an efficient (and simple) thresholding strategy is proposed by analysing the identified thresholding sequences. Chapter 5 is

more concerned with the question "what causes unsuccessful recall?". As a result of answering to the question, an efficient associative network architecture and learning paradigm is proposed and analysed. Finally, how those models might relate to the real biology of the brain is discussed in chapter 6.

Chapter 2

In search of optimal threshold sequences

2.1 Introduction

In the previous chapter, I have described the critical role that choice of threshold plays in associative memory recall. The thresholding sequence is particularly important in recurrent autoassociative memory, as the result of setting a threshold affects the network activation of the next time step, during which further recall is established. Therefore, it is of natural interest to find a threshold sequence that leads to successful recall in an associative memory that is at or near its capacity limit.

At any one time in the recall process, some cells in the network are active. As described in the previous chapter and also in appendix A and B, the cell activity is expressed in a binary vector $\mathbf{x}$. As the recall progresses, the components of $\mathbf{x}$ continually change through time. The content of $\mathbf{x}$ at a particular time t is $\mathbf{x}(t)$, which is the *state* of network at time t. "State" is an appropriate word to use because a state describes the momentary activity of the network, that is, it is possible to reproduce recall steps if it is possible to reproduce the states associated with them. In other words, while the configuration of the network (N, R, M, and W) describes the hardware specification

of the network, a state describes the instantaneous activity of the network. Furthermore, the sequence of states describes the dynamics of the network.

This chapter focuses on the methodology for finding optimal threshold sequences using the state as the fundamental data structure of the algorithm. The word *optimal* is used in the sense that it achieves the recall with highest quality value (as defined in the previous chapter).

2.2 State tree

When w cells are active at a time, there are w different threshold values that could be applied, as a cell can receive at most w effective dendritic inputs. The probability distribution of the dendritic input is described in the previous chapter. This implies that there will be w possible different states for the next time step of the recall, depending on the threshold applied at the current time step. These states will again have many associated next states that depend on the threshold applied to the network at the next time step, and so on.

A *tree* structure is a collection of nodes and edges. A *node* is a simple object that contains some information. Two nodes can be associated with an *edge*. Every node is connected to some other nodes via edges. One node in the tree is denominated as the *root*. In a tree, there is only one combination of edges (*path*) that connects to the root and a node (this is the crucial definition of tree structure). Therefore, starting from the root node, it is possible to define the *depth* or *level* of a node as the number of edges passed to reach to the node from the root.

Starting from a single state (i.e. at the initiation of a recall), a diverse number of states can be formed in a cascade manner. This can conveniently be represented using a tree structure by abstracting possible states at time t

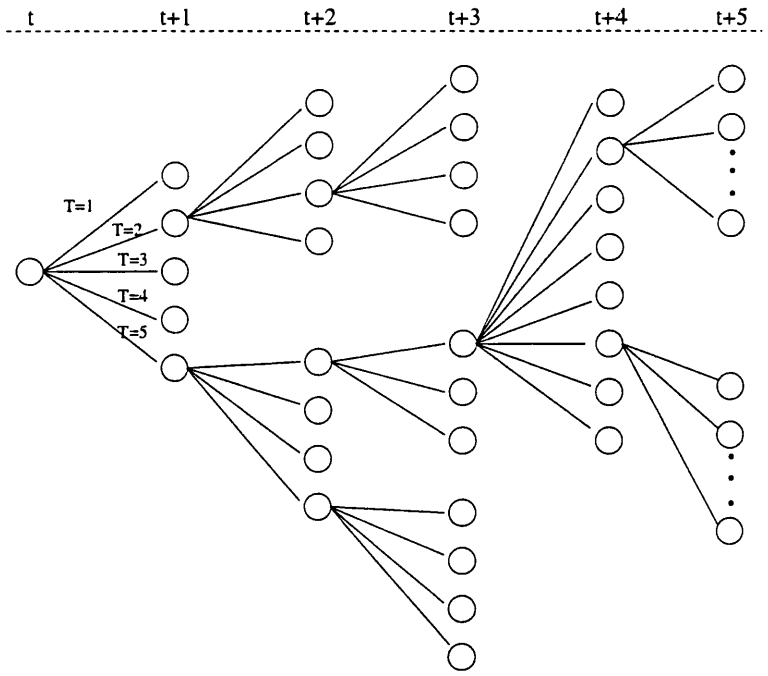


Figure 2.1: A schematic diagram of a state tree: Transition of the network states can be represented using a tree structure. Circles and lines represent nodes and edges, respectively. Each branch corresponds to a distinct threshold applied at the corresponding depth

to nodes of level t (figure 2.1). The initial input to start a recall becomes the root of the tree. Depending on the number of possible values of threshold, subsequent nodes are formed for each of the associated states. In this way, the edge also contains information, namely the value of the threshold applied to the current state to achieve the next state.

Those nodes represent the possible states of the network during recall of a single arbitrary stored memory. These states represented in a tree structures are referred to as the *state tree.*

2.3 The algorithm

In the previous section, it has been emphasised that network states and their relations can effectively be represented using a tree structure. If the recall process is constrained to occur within a finite time limit n, it is possible to describe all the possible states using the state tree, with a maximum depth n. Given the root and a state in the state tree, the recall sequence to reach the state from the initiation of the recall is simply the path between the root and the state. The combination of the edges that constitute the path is the threshold sequence applied to the network to achieve the recall.

A state tree containing a perfect recall has at least one node whose state exactly matches the learned event. This type of node is referred to as an *ideal node.* If an ideal node exists in the tree, the optimal threshold sequence is identified by finding the path between the root node and the ideal node.

It is possible that the tree does not contain an ideal node. For example in a pathological case, a cell may receive no collateral connections, since the connections are established in a probabilistic manner. If such a cell takes part in forming an event, the cell can never be activated via the feedback input. A less stringent criterion is to determine if a state has good rather than perfect recall. The criterion is based on the quality measure (Q) described in the previous chapter. If the state in a node achieves a higher quality than a defined border quality, that node is labelled as a *goal node.* There may be a number of nodes that are labelled as goal nodes.

To summarise, a node in the state tree contains a state of the network and the quality of the state with respect to the event being recalled. An edge contains the threshold value applied to the current state to achieve the next connected state. An ideal node contains exactly the same state as the stored

state. The goal nodes contain states that are sufficiently close to the stored state as measured by the quality metric. Since the threshold sequence to a state is simply the path between the root node and the state, an algorithm to search for threshold sequences to achieve successful recall searches for paths between the root state and the goal states.

2.3.1 Selective inclusion

Suppose we have a state tree that describes all of the possible states up to step n. Suppose also that the recall starts from a cue containing w_0 active cells. Now, there will be w_0 possible nodes connected to the root node. Each of these nodes hosts a different state that has w_1^1, w_1^2, ..., $w_1^{w_0}$ active cells, respectively. As mentioned before, these nodes generate branches according to their state. It is easy to see that the number of nodes and edges in the tree expands exponentially. After several time steps (depth), the state tree becomes so huge that any searching algorithm over the state tree becomes intractable, not to mention the resource limitations of a finite computer system (such as computational power and memory storage).

In order for a searching algorithm to function, the size of the state tree must reduced. One way to restrict the size of tree is to put a limit on the maximum depth, or number of levels. This is equivalent to saying that if a successful recall can occur, it has to occur within a certain defined number of threshold steps. The tree may still grow in a exponential manner, but at least the maximum possible number of nodes within a tree can be calculated by limiting the depth of the tree. Hence the searching algorithm becomes partially tractable.

Another way to restrict the size is to limit the number of edges a node can have to states in the next level of the tree. This is called *selective inclusion.*

At a first glance, this type of selective inclusion seems to be a radical approach to cut down the size of the tree, perhaps so radical that a branch containing some useful nodes (such as goal nodes or the path to goal nodes) may also be excluded. The risk can be reduced by carefully choosing the nodes that are included.

Some evidence in support of selective inclusion of a tree can be seen by looking closely at the network activation during the recall process. Figure 2.2 shows the dendritic input distribution for a partially connected network with half of the correct cells active and no active spurious cells. The total active input (w) is 100, so there are 100 different values of the global threshold. However, as demonstrated in the graph, none of the cells receive more than 20 effective inputs. Therefore, nodes generated as a result of applying threshold greater than 20 can be pruned. On the other hand, choosing a threshold less than seven results in an enormous number of spurious cells, which is likely to lead to failure in the rest of the recall process. Therefore nodes generated as a result of applying a threshold smaller than seven could be pruned, as well.

In fact, pruning can automatically be done on the basis of the quality of the state. That is, if the quality of a generated state is high enough to possibly lead to a successful recall, the state is maintained as a node in the state tree. Otherwise, the state is pruned. All the states mentioned in the previous paragraph should have very low (i.e. near zero) quality, so they are pruned. In the implementation, each node of the search tree is restricted to have five nodes of top quality values. This restriction has to be imposed as the size of the tree expands exponentially with respect to the number of child nodes per node. This issue is discussed further in section 2.4.3

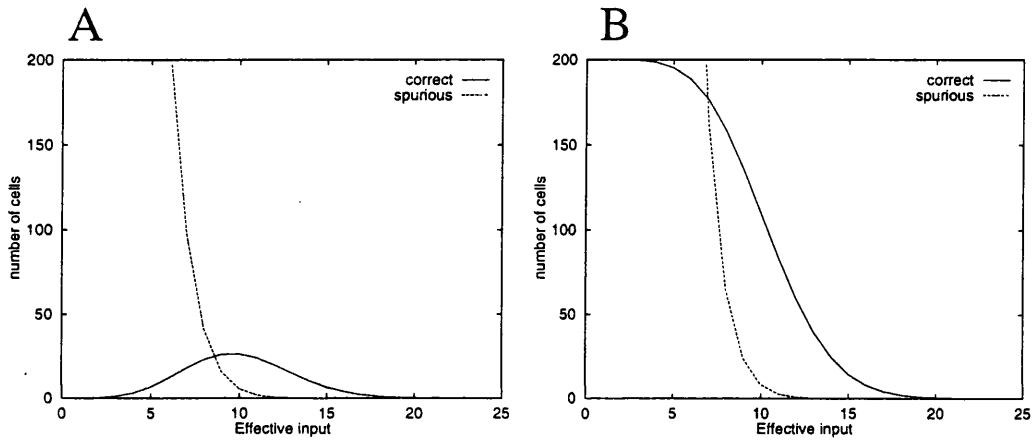


Figure 2.2: Dendritic distributions of a partially connected network. ($N = 5000$, $R = 500$, $M = 200$, $W = 200$, $c = 100$, $s = 0$)**A** Dendritic input (density) distribution depicts how many cells there are that have a certain number of effective inputs **B** Dendritic input mass distribution depicts how many cells there are that have received at least a certain number of effective inputs. In other words, the dendritic input mass distribution is the integral of the dendritic input (density) distribution with respect to the effective input. The distributions are calculated using Buckingham's corrected formulae. (Buckingham, 1991)

2.3.2 Searching and Backtracking

Once the state tree is generated, the task is to search for the goal nodes. Even for a pruned tree, the search space can be very large. A method is required to systematically and efficiently conduct this search.

There are at least two well established methods for searching in a tree. They are called *depth first* search and *breadth first* search. A depth first search travels around the tree from the root node in favour of depth. That is, whenever, there is a connected node deeper than the current node, the algorithm chooses to visit the node. When there are a large number of nodes descending from the current node, the algorithm completes the search over the resulting branch from one node to another. When there is no descending node, the algorithm checks if the node matches to the search criterion. When the node does not match the search criterion, it continues to search by travelling a level upward and visits the next unvisited descending node, and so on (see figure 2.3 A). The depth first search algorithm can conveniently described with the following pseudo-code.

```
depth_first_search(search_item)
{
  if (dps(root_node,search_item)=='found') {
    return('found')
  } else {
    return('not found')
  }
}
dps(node,search_item)
{
  if exist_subnode(node)
    foreach subnode in (node.subnodes) {
      if (dps(subnode,search_item)=='found')
        return('found')
    }
  if contains(node,search_item) {
    return('found!');
```

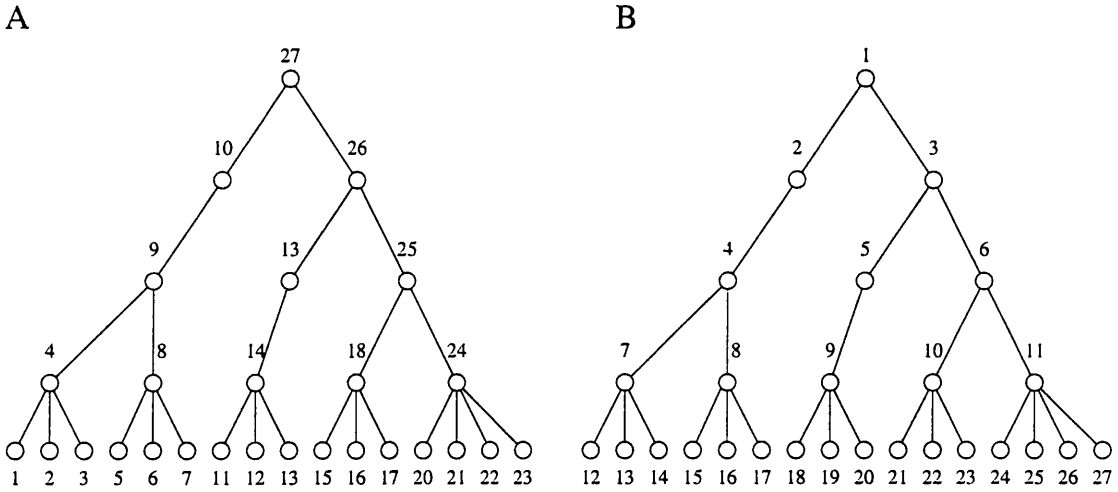


Figure 2.3: Two methods for searching within a tree: **A** Depth first method visits nodes in favour of depth. **B** Breadth first method scans nodes horizontally level by level. The numbers on the nodes show sequence of the search.

```
  }
}
```

The breadth first method scans nodes horizontally level by level (figure 2.3 B). Starting from a root node, it visits all of the nodes immediately below the root node. Upon the completion of the traversal, it then proceeds to scan all of the nodes in level 2, and so on.

Depth first search is useful when the tree is organised in such a way that good nodes tend to exist in a deep left corner, while the breadth first method is more efficient if the good nodes tend to occur in the upper part of the tree. Successful recall occurs once the state achieves a certain quality and the nodes that contain states that satisfy this criterion are called the goal nodes. Therefore, when we wish to know how soon a successful recall can occur, breadth first search is more appropriate.

Once goal nodes are found, the path has to be identified in order to know

the values of thresholds the network has experienced to achieve the state during the process of the recall. This is done by using a technique called *backtracking*. The crucial property of a tree structure is that there exists only one path between the root tree and a node guarantees that there exists only one edge from the current node to the preceding (i.e. one level lower) node. Backtracking is the technique that seeks the edge connecting to the preceding node recursively until the root node is reached. By collecting these edges and reversing the sequence order, the path can be found. Backtracking can easily implemented if each node in the tree includes a pointer that directs to the preceding node that connects to the node.

2.4 Implementation

The state tree is generated by simulating a sparsely and recurrently connected network. In this section, more specific implementation issues about generating the state tree are discussed.

2.4.1 Model description

Sparsely and recurrently connected binary associative networks has been simulated to test the effectiveness of the algorithm and data structure ($N = 5000$, $R = 500$, $W = 200$). On presentation of a subset of the active cells in a learned event, the network should recover the rest of the cells by means of progressive recall. The network is similar to the original network proposed by Gardner-Medwin (1976) in that the network is sparsely connected, and the initial input is presented throughout the progressive recall process.

The technique described in this chapter was tested with two different network loadings. The number of stored patterns (M) was chosen so that ρ (the expected proportion of modified synapses, see chapter 1 and appendix B for

more detail) is 0.2 ($M = 140$) in one case and 0.3 ($M = 223$) in the other.

2.4.2 Tree generation

Pruning of the tree is based on the quality of a node. Pruning is done for each node so that a node branches to five nodes that have the highest quality values. It is more computationally efficient and economical in storage to prune while building the tree rather than building the complete tree structure and then pruning it. This process is described later in this section.

Having fixed the number of descending nodes per node, the total number of nodes in the tree is $\sum_{i=0}^{L} 5^i$, where L is the maximum depth of the tree. This means that the storage resource requirement grows with the order of $O(5^L)$. Computational time required to generate the whole tree grows with the same order. With this in mind, the maximum depth is set to be 6 so that the tree contains 19,531 nodes.

Nodes are formed in a depth first order, where a node is defined to be completely formed when all of the five descending nodes are formed or it is in the maximum level. Beginning from a root node, on presentation of initially active cells, all possible thresholds are tested and the quality values are calculated. Then, the results are sorted in the order of quality value so that five nodes with top quality values can be selected. Among the five nodes, the one with the highest quality value is chosen to expand the tree a level deeper. The process is recursively continued until the maximum depth of the tree is reached. Once the maximum depth of the tree is reached (after creating node b in figure 2.4), the second best node (node c in figure 2.4) will be made from the node above (node c in figure 2.4). Since the second best node is in the level of maximum depth, it cannot be expanded any further. The third, fourth and fifth nodes (node, d,e and f in figure 2.4, respectively) will be generated

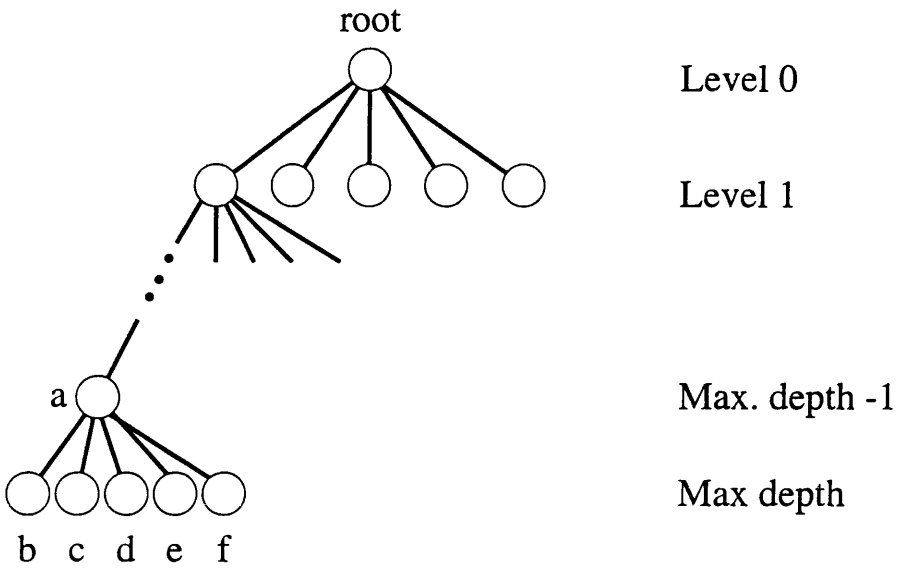


Figure 2.4: Generating a state tree (see text for the explanation).

in the same manner.

Once the leftmost five nodes of the maximum depth are formed, the leftmost node of depth immediately above the maximum depth (node a in figure 2.4) is completely formed. The tree will be expanded for the branches with the same origin as node a. The process will be recursively continued until the whole tree is constructed.

This process is more concisely described by the following pseudo code.

```
generate_tree(state,current_depth)
{
  if (current_depth+1==max_depth)
  {
    nodes = best5nodes(states)
  }
  else {
    nodes = best5nodes(states)
    for (i=0;i<5;i++) {
      generate_tree(nodes[i].state,depth+1)
    }
  }
}
```

Searching for the good nodes is done at the same time as the tree is being generated. This is accomplished by maintaining a global list of nodes that contain highest quality values. Suppose the list is for containing best 10 nodes in the tree. Whenever a node is created during construction of the tree, the node is assessed for its quality. When the quality is higher than a node stored in the list, the node with the lowest quality value in the list will be substituted with the newly created node. When the tree is completed, the list contains the global top 10 quality nodes. Backtracking is then implemented over those nodes to seek the threshold sequences.

2.4.3 Resource requirements

All computers have a limited set of resources. A very large quantity of computer memory would be required to apply the method to a realistic evaluation of a large network. Here, the memory resources required for this task are briefly considered.

Since the activity of each modelled neuron is binary, the state of the neuron requires only one bit of storage. The activity of a network is essentially the collection of the cell activities and it can be stored in a binary array of $N/8$ bytes (1 byte is 8 bits), and the same amount of information is required to hold the next state of the network after a threshold is applied.

As an event is an instance of network activity, $N/8$ bytes are also required to represent an event. The network learns M events, so $MN/8$ bytes are required to store the learned events. Similarly, the connectivity matrix and weight matrix can be stored using $N^2/8$ bytes each. If we assume that all the learning occurs in one stage, the connectivity matrix is not really needed once the weight matrix is generated. Even more efficiently, weight matrix can be written on top of the connectivity matrix so to save the space in the computer

memory.

In total, a naive calculation of major resource requirement for the network is $MN/8 + N^2/8$ bytes. For the network specified in section 2.4.1 (M=223), the resource requirement is approximately 3.3 Mbytes which is small enough to be accommodated in a modern computer. Note that the quantity grows with the order of $O(N^2)$.

The above is a naive calculation of resource requirement. When sparse encoding is assumed, it is more economical to store the indices to the active cell than storing the whole activity of the network in a binary array. $log_2 N$ bits are required to represent N different numbers. If it is possible to specify the absolutely maximum amount of active cells at any time in the network as w_{max}, the network activity is more economically stored if the following condition is satisfied:

$$w_{max} log_2 N < N$$

In practice, modern computer architecture is optimised for memory allocation of a byte, 2 bytes (word), and 4 bytes (long words). To index a cell in a network with $N < 2^{16} (= 65,536)$, 2 bytes are used per index. Each event has a fixed size of active cells, namely W. If the indexing scheme is employed to represent events in the computer memory, it uses $2MW$ bytes.

The same indexing method can be used for the connectivity matrix and the weight matrix, when the connectivity in the network is sparse. Instead of storing a series of binary arrays, if indices to connected cells may be stored for the connectivity matrix is N. In this case, the resource requirement is $2RN$ bytes.

The amount of memory space occupied using the indexing method is $2MW + 2RN$. For the specified network, the resource requirement is about 5.1 Mbytes.

When R is determined as a proportion of N (e.g. 10% connectivity $\equiv$ $R = 0.1N$) the quantity grows with $O(N^2)$. However, the indexing method will become more economical if sparse encoding (small W) and more significantly, sparse connectivity (small R) is realised.

As stated in section 2.4.2, the number of nodes in a depth L state tree is $\sum_{i=0}^{L} 5^i$. Since each node contains a state that occupies $N/8$ bytes, a six depth tree occupies at least 12.2 Mbytes. Physically, the computer has 64Mbytes of memory of which about 50Mbytes is available for general use and the virtual memory is about twice this size. This means that a six depth tree can be constructed confortably while a seven depth tree (which requires at least 61 Mbytes) can fit the system, although the speed of processing will tremendously slow down due to use of virtual memory.

2.5 Results

The state tree with the maximum depth 6 (starting from level 0) for the network was generated and was searched for goal nodes.

Figure 2.5 shows the extracted optimal threshold strategy and the resulting network activity for the network with ρ=0.2. The number of correct cells steadily increased throughout the recall while the number of spurious cell increased at the first step of the recall and then gradually decreased. The final quality value is approximately 0.89.

Figure 2.6 shows the same plot as in figure 2.5 except that the network loading is increased ($\rho = 0.3$). Here, we see that 6 steps are not enough to complete the recall. Spurious cells are boosted as a result of the first thresholding and then filtered out in the second instance of thresholding. At the third state of recall, the level of threshold is decreased so to increase the

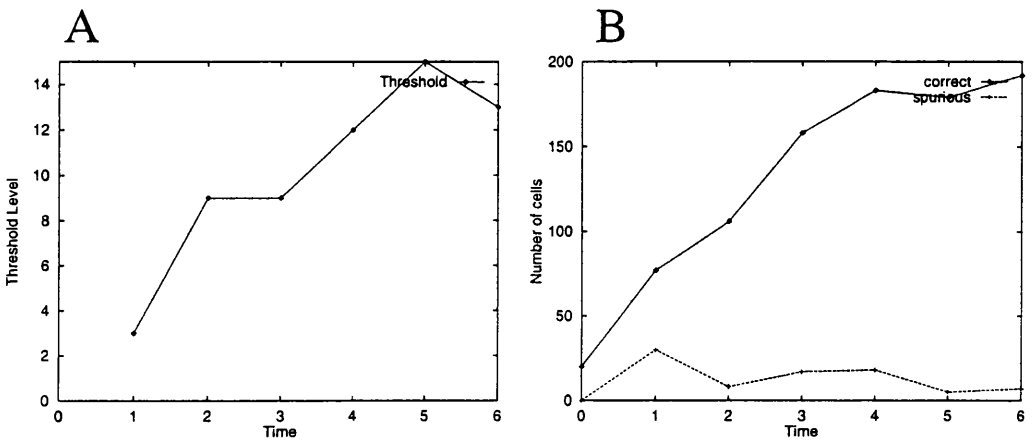


Figure 2.5: Identified optimal threshold sequence (**A**) and the associated activity of the network (**B**). The network configuration is $N = 5000$, $R = 500$, $W = 200$, and M=140 so that ρ=0.2. The network is activated with the initial seed containing 20 correct cells and 0 spurious cells.

number of correct cells while boosting some spurious cells. The fourth and fifth thresholds seem to have the effect of filtering out spurious cells while sustaining the activity of correct cells. Finally at the 6th stage of recall, the threshold is "forced" to be set lower to increase the correct cell activity in spite of the increase of spurious cell, to improve the quality of the recall. The final quality value is approximately 0.41.

One could argue that the reason that successful recall could not be achieved in the network with $\rho = 0.3$ because the loading is beyond the maximum capacity (i.e. overloading). To test the proposition, a depth seven tree was generated for the network, compensating the most of the machine's memory and virtual memory, not to mention the computational time (> two days). If the final quality (at seventh time step of recall) increases, recall may be possible if more time steps are allowed. However, searching through the state tree of such depth would not be not feasible because of the resource constraint.

The result of search over the seven depth state tree is demonstrated in figure 2.7. As observed in figure 2.5 and 2.6, spurious cells are boosted up in

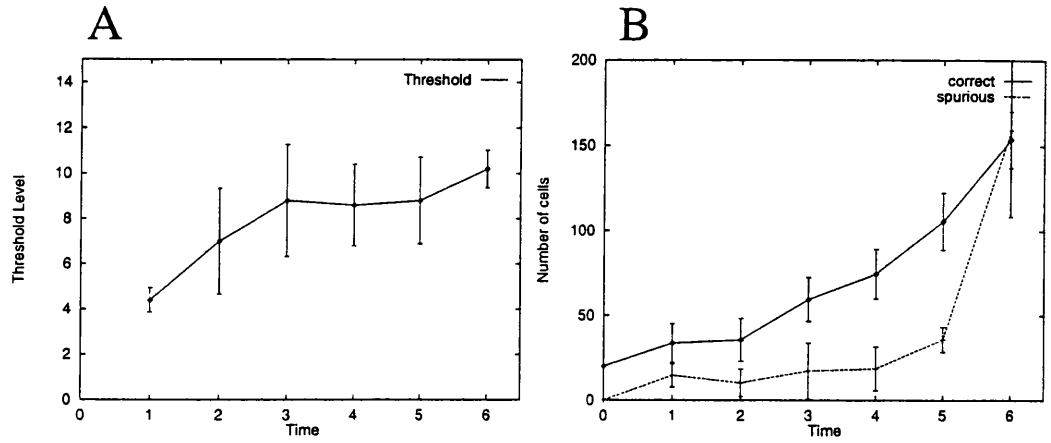


Figure 2.6: Identified optimal threshold sequence (**A**) and the associated activity of the network (**B**). The network and the initial cue configuration and the same as in the previous figure except M is now 223 ($\rho = 0.3$). The large number of active spurious cells in **B** is not due to failure of recall. The plots are the average of 5 tree based searches.

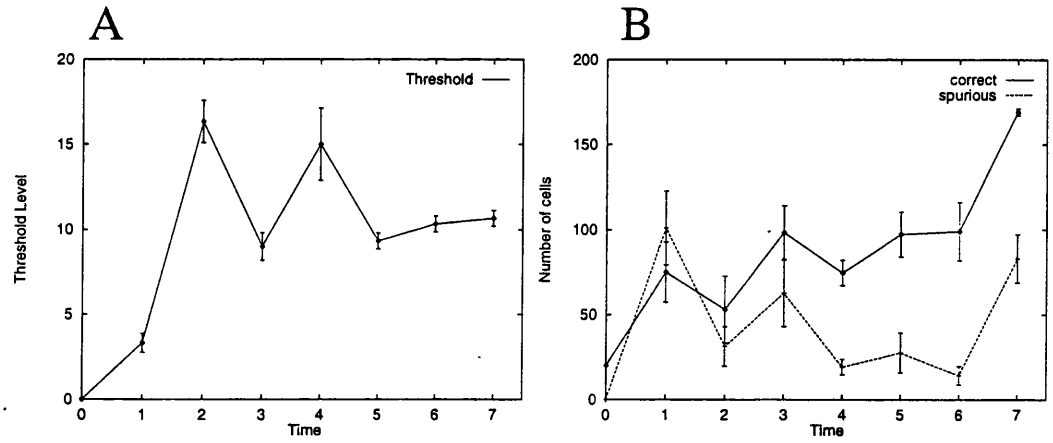


Figure 2.7: Optimal threshold sequence identified in a depth seven tree for the network with $\rho = 0.3$ (The rest of the configuration is the same as the previous figure) (**A**) and the associated activity of the network (**B**). Note the vertical scale change for the threshold sequence graph. This result is the average of 3 tree based searches

the initial stage of recall and are filtered out in the later stages of recall. The final quality value is approximately 0.60, which is a significant improvement from that of 6 depth tree.

2.6 Discussion

The quality of recall in randomly and sparsely connected associative memories is highly dependent on the method in which a global threshold is applied. Progressive recall can be used to increase the recall performance, but the method for determining the optimal threshold sequence for progressive recall has not yet been addressed. In this chapter, it is suggested that dynamical states of a recall can be empirically stored in a tree structure called the state tree. Once this is accomplished, optimal threshold sequences are able to be extracted from the state tree by finding the path between the final node and the root node using the backtracking method.

The thresholding strategy in Gardner-Medwin (1976) is based on the statistical distributions for active cells so that a threshold is chosen in such a way that the threshold minimises the activity of spurious cell activity while maximising the activity of correct cells. In an ideal situation, this is done by choosing the lowest threshold that allows no spurious cells to become active while the number of active correct cells is increased.

The optimal threshold sequences extracted from state trees does not quite match Gardner-Medwin's approach. Instead of always minimising spurious cell activity, it seems to boost correct cell activity while also boosting spurious cell activity. The spurious cell activity is then gradually reduced by raising the value of the threshold. This makes sense because the quality value, which is a measure quantifying how close a state is to the recalled event using informa-

information theory, is higher for the state with many correct cells and some spurious cells than the one with fewer correct cells and no spurious cell. However, a thresholding strategy based on the information quality of the next step would not be a great idea, as one has to look ahead all the possible next states and evaluate the quality of the state for each of them.

First of all, testing all the possible threshold values is computationally expensive. Aside from calculating the dendritic input for each cell and thresholding it accordingly for each tested threshold, it requires a buffer to store the value of the threshold and the quality value that produced the best quality. Calculating the quality value and comparing it to the one in the buffer each time requires an enormous amount of computation.

Another point is that evaluating quality value requires explicit knowledge of which of the cells are correct cells and which are not. This means that the system that calculates the quality value must know what is stored in the memory very precisely. However, if a system knows what is stored in the memory, there is no need of having memory as the system *per se* is already the perfect memory system. Hence, such a system is not expected to exist in biology. So this approach has not brought us closer toward finding a biologically possible meachanism for progressive recall.

One might argue that the boost-and-filter-out threshold sequence is observed because there are only five candidates of state available from a given state and the five candidates all have some amount of spurious cell activity as they are chosen according to the quality value. This seems to be a valid argument. However, when 20 cells are presented at the initiation of the recall (mean input to a correct cell is 2 and the standard deviation is 1.34) surely a node containing no active spurious cell must be one of the five candidates and

the node must be expanded further. If a branch produced a better state with better quality, it would be reported in the search procedure.

From a computational cost point of view, it is relatively inexpensive to increase the complexity of the global threshold unit, in exchange for an improvement in recall performance.

In comparison between an associative memory device and a network of real neurons (e.g. region CA3 in the hippocampus) it is interesting to note that the models lack the complexity of the interneurons present in the central nervous system. Real networks of interneurons may improve the recall performance of the principal cells by producing a more complex strategy for progressive recall.

Summary

In this chapter, a method to identify optimal threshold sequences was suggested and discussed. Optimal threshold sequences have been identified for a sparsely connected recurrent associative network.

Notes Part of this chapter has been published in Hirase and Recce (1995). After completing this study, several improvements became evident. For example, only the root node is required to store a state as the subsequent nodes' states can be determined by applying the corresponding threshold sequence, which would have saved a vast amount of computer memory. However, these potential improvements were not evaluated since the theory based method of seeking optimal threshold sequences (described in the next two chapters) proved to be much more efficient.

Chapter 3

Table-based search for thresholds, using a theoretical estimate

3.1 Introduction

In the previous chapter, evidence was presented to suggest that threshold sequences that result in higher quality states differ from the strategy proposed by Gardner-Medwin (1976). Instead it appears to be beneficial to have spurious cells active during the initial stages of progressive recall. State tree generation is an entirely empirical approach for identifying optimal threshold sequences. The threshold sequence extracted from the state tree is only valid for the particular recall and cannot be generalised. Statistical approximation of optimal threshold sequences might be possible. However, in order to obtain a reliable estimation, a number of threshold sequences has to be obtained. As mentioned in the previous chapter, generating a depth six state tree is computationally very intensive and a depth seven tree is the absolute maximum that can be evaluated with the computational facility in the laboratory. Accumulating sufficient statistics from state trees would require an enormous amount of computational resources and time.

Another approach is to theoretically estimate the dynamics of the net-

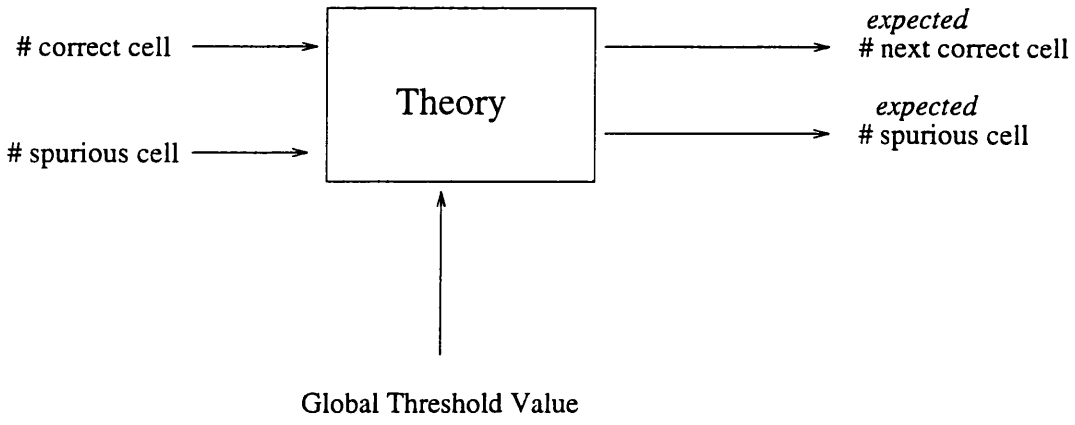


Figure 3.1: A schematic diagram for a model of theoretical estimation of network activity. Given the current status of the network (i.e. the number of active correct cells and the number of active spurious cells) and a global threshold value to be applied to the network, the theory predicts the expected number of correct cells and spurious cells.

work activity. If the current status of the network is known, it is possible to calculate the expected number of active correct cells and spurious cells and therefore compute the expected performance of the recall (figure 3.1). The theory applied for this process was developed by Gardner-Medwin (1976) and later corrected by Buckingham (1991). Using the theoretical estimates of recall performance, one can search for the (expected) optimal threshold sequence. These theoretical estimates are verified by comparing them to results obtained from a sumulation.

There are disadvantages to using a theoretical estimate instead of direct simulation. In the first place the domain of applicability of the theory may not fully overlap with the biological system. Also the theory may not be accurate enough to predict the results from the simulations.

3.2 Theoretical Background

To recapitulate, given c active correct cells and s active spurious cells, totalling $w = c + s$ active cells, Gardner-Medwin (1976) estimated that the probability that any one correct cell receives exactly r effective inputs is:

$$P(d_{corr} = r) = \binom{c}{r} \left(\frac{R}{N}\right)^{r} \left(1 - \frac{R}{N}\right)^{c-r} = \mathcal{B}\left(\frac{R}{N}, c, r\right) \tag{3.1}$$

Further, he approximates the probability using a Poisson distribution, which is reasonable to numerically compute the probability.

Similarly, the probability that a spurious cell receives exactly r effective inputs was proposed to be:

$$P(d_{spur} = r) = \mathcal{B}\left(\rho\frac{R}{N}, c, r\right) \tag{3.2}$$

where ρ is the proportion of modified synapses. ρ can be calculated as:

$$\rho = 1 - \left(1 - \frac{W^2}{N}\right)^{M} \tag{3.3}$$

(For more detailed explanation, see Appendix B)

Further, when $c(t)$ correct cells and $s(t)$ spurious cells are active at time t, the expected number of active correct cells and active spurious cells at the next time step when threshold $T(t+1)$ is applied are:

$$\langle c(t+1)\rangle = WP(d_{corr} > T(t+1)) \tag{3.4}$$

$$\langle s(t+1)\rangle = (N - W)P(d_{spur} > T(t+1)) \tag{3.5}$$

Performance analysis using the above formalism was carried out by Gardner-Medwin (1976) (See figure 3.3).

The above analysis does not accurately model the behaviour of the network, because, as pointed out by Buckingham (Buckingham 1991), it relies on

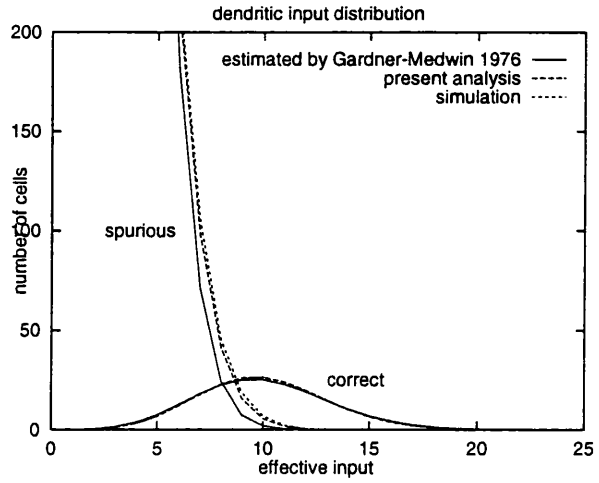


Figure 3.2: Dendritic input distribution ($N = 5000$,$R = 500$,$M = 200$,$W = 200$,$c = 100$,$s = 0$) The new analysis closely matches with the simulated result. The prior theory underestimates the dendritic input of the spurious cells. (After Buckingham(1991), using different parameter sets.)

the assumption that every principal cell participates in the same number of events (hence, use of the constant ρ is justified). In reality, unit usage of a principal cell varies from cell to cell and the proportion of modified synapses to the cell depends on the level of unit usage. In fact, unit usage is binomially distributed with the principal probability W/N. Therefore, distributions involving ρ must be modified to acquire more accurate estimates. The modified version of formula 3.2 is:

$$P(\mathbf{D}_{spur} = r) = \sum_{m=1}^{M} P_e(m)\mathcal{B}\left(\rho(m)\frac{R}{N}, c, r\right) \tag{3.6}$$

where P_e is the probability that a principal cell participates in m events and $\rho(m)$ is the probability that a synapse impinging onto the principal cell with unit usage m is modified to be effective. They are $\mathcal{B}(W/N, M, m)$ and $1-(1-W/N)^m$, respectively.

The result is that the new probability distribution looks more wide-spread

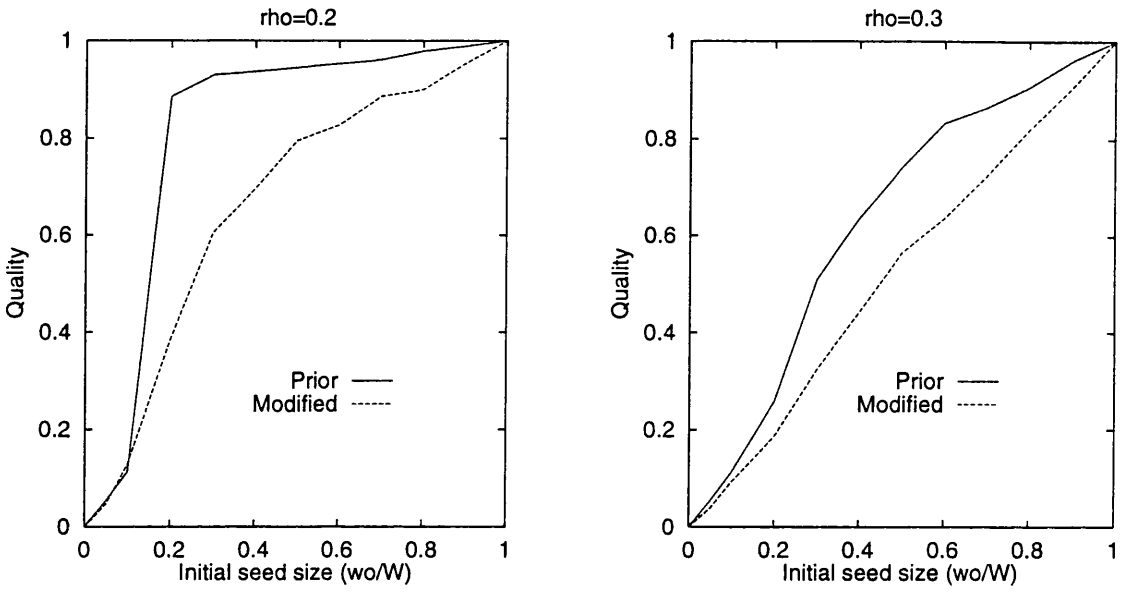


Figure 3.3: Analysis of the expected value of recall performance (Reproduced and modified from Gardner-Medwin (1976). These graphs compare the performance estimated by Gardner-Medwin (1976) [Prior] with the new analysis of the prior strategy based on Buckingham (1991) [Modified]. Note that the prior estimates were overestimates. (See figure 3.7.)

and has a longer tail, although the mean of the distribution is the same. The impact,on the original theory of the corrections to the probability of r effective inputs can be seen in figure 3.2, which compares the total dendritic input expected for correct and spurious cells in an example network configuration. In this figure the prior and present theories both predict the same correct cell distribution as that measured in a simulation of the network. However the prior model underestimates the spurious cell dendritic input, while the present model matches well with the simulated network.

Using the corrected formalism, the capacity analysis is reassessed in figure 3.3. The horizontal axis of the graph is the size of initial seed pattern which contains only correct cells. In the initial model (Gardner-Medwin 1976), the vertical axis was percentile match of correct cells. In this thesis, the previously

defined quality measure is used instead.

The increase in dendritic input for spurious cells due to the corrected dendritic distribution has a significant impact on the expected performance of progressive recall, as shown in Figure 3.3. In this figure the change in expected quality is shown for one configuration of a network. Note the decrease in the expected performance of the prior threshold strategy, that results from the incorrect estimation of the dendritic input in spurious cells.

3.3 Table-based search

It is now possible to evaluate the expected numbers of next active correct cell and spurious cell, given the current state of the network. Moreover, using the formalism defined in the previous section, it is possible to calculate the expected state of the network at time t, if the initial input to the network and the threshold sequence are given. A *table-based search* algorithm is developed to look for optimal network performance from the theoretical calculations of expected performances.

The search is carried out by constructing a two dimensional table (the search table) that stores the shortest sequence of thresholds required to arrive at a particular number of correct and spurious cells. The number of rows and columns in the search table are set by the range of possible numbers of correct and spurious cells. The number of correct cells ranges from zero to W and the range of spurious cells is from zero to $2W$ (See figure 3.4). A larger number of spurious cells is likely to lead to unsuccessful recall.

The table-based search algorithm reads and writes data from the table. Entries of the table are indexed by *cor* and *spur*, corresponding to the number of correct cells and the number of spurious cells, respectively. For conve-

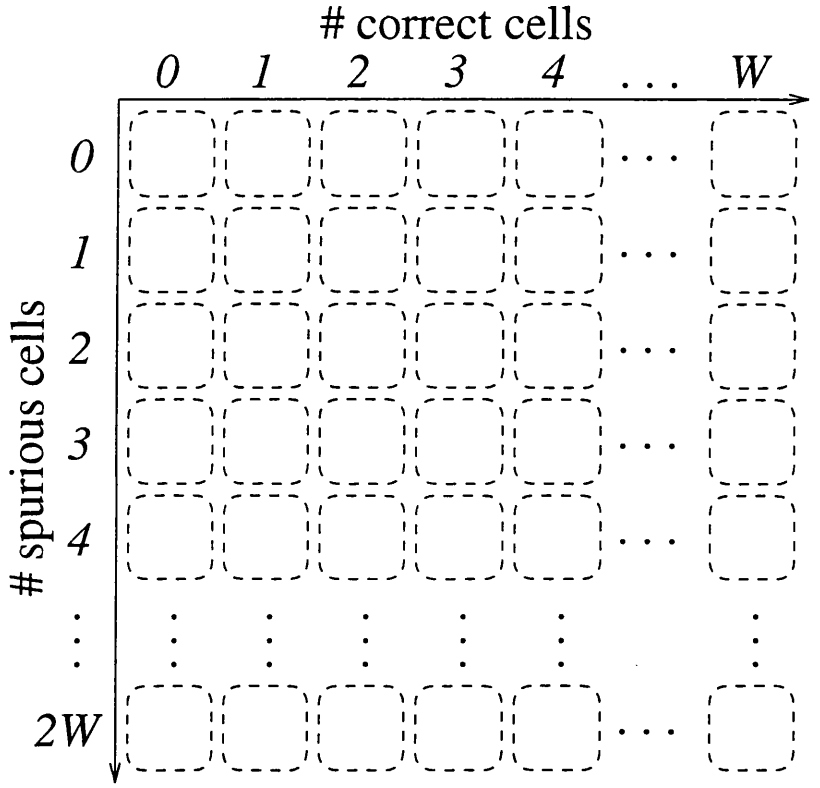


Figure 3.4: A schematic diagram of the search space in table-based search. For each number of correct cells and spurious cells, there is an entity (slot) to hold attributes listed in table 3.1 Each entity in the table is initialised to zero (dashed rounded box). Hence there is no link with the entity at the start of the algorithm.

correct	The number of correct cells in the previous time step. This value is used as an index to point to the previous entry in the table.
spurious	The number of spurious cells in the previous time step. This value is used as an index to point to the previous entry in the table.
threshold	Threshold value applied in the previous time step to obtain the current configuration (i.e. *cor* and *spur*).
stage	The time step (integer value) at which the entry was made into the table.
x	Given the previous configuration, x is the probability that a correct cell fires with the specified *threshold*.
y	Given the previous configuration, y is the probability that a spurious cell fires with the specified *threshold*.

Table 3.1: table caption

nience, the entry with *cor* correct cells and *spur* spurious cells is represented as (*cor*,*spur*). Each entry of the table contains the data shown in table 3.1.

The essential aim of the algorithm is to identify the shortest path between a state (goal state) and the initial state. This is accomplished by linking the entry in the search table that corresponds to the state and the initial state via the entries that represents the intermediate recall states of the network. Starting from the initial state, table entries are completed for the expected results of applicable thresholds. The details are as follows.

Each table entry is initialised to zero (figure 3.4). At the first time step each possible threshold value and the resulting (expected) network status are computed and stored in the table with a time stamp of *one*. Only integer threshold values are tested since the net input to a neuron is necessarily an integer. If two thresholds result in the same number of active correct and spurious cells then the first threshold value is kept in the table.

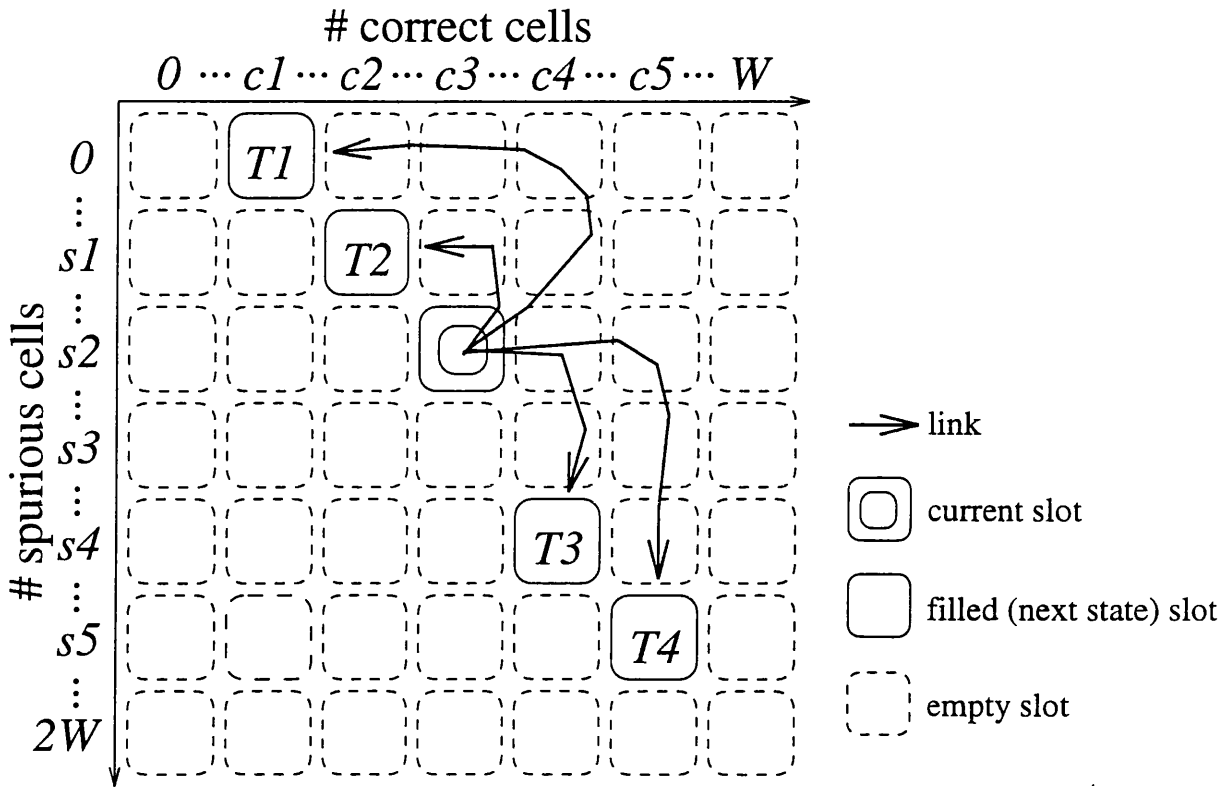


Figure 3.5: The search table is partially filled with the current state and the resulting possible next states. See text for detail.

During the second time step the changes in the number of active correct and spurious cells are calculated for each table entry with a time stamp equal to *one* (current state). Note that the time stamp clearly labels the set of table entries that are possible at the prior time step. If a threshold applied at the nth step results in a number of correct and spurious cells that is already in the table then the table entry is not changed.

In the third time step, exactly the same process is carried out except that the current states are the ones that have the time stamp of *two* and the newly calculated states have time stamp *three*, and so on.

For example, in figure 3.5, a current state has $c3$ correct cells and $s2$ spu-

rious cells with time stamp t. Among possible values of the threshold, four thresholds $T1...4$ produced the next expected states that have not yet been arrived at previously. The four states, $[(c1, 0), (c2, s1), (c4, s4), (c5, s5)]$, are filled with relevant information described in table 3.1. They all have the same time stamp, namely, $t + 1$ and are linked to the current state using *correct* and *spurious* attributes of the entry. This process is carried out until fifteenth time step.

The optimal threshold sequence is found by identifying the table entry that has the highest quality value. The associated threshold sequence can be obtained by back-tracking the *correct* and *spurious* value in the table entries (the technique is explained in chapter 2). In order to shorten this search process we did not evaluate threshold sequences that are guaranteed to result in lower quality recall. More specifically, with each number of spurious cells (row) in the table we only evaluated the thresholds for the entry which had the largest number of correct cells, or equivalently we evaluated at most one starting point per row of the table.

3.4 Results

The table search algorithm is implemented for the network as described in chapter 2. The storage space required to maintain the search table is $W \times 2W \times sizeof(entry)$ bytes. The size of each entry in the table is 20 bytes, when each attribute occupies 4 bytes. The number of active cells in each stored pattern (W) of the tested network is 200. Therefore, the total amount of storage occupied by the search table is 1.6 Mbytes, which is managable.

Figure 3.6 compares the expected value of recall performance using the thresholding strategy proposed by Gardner-Medwin with that found using

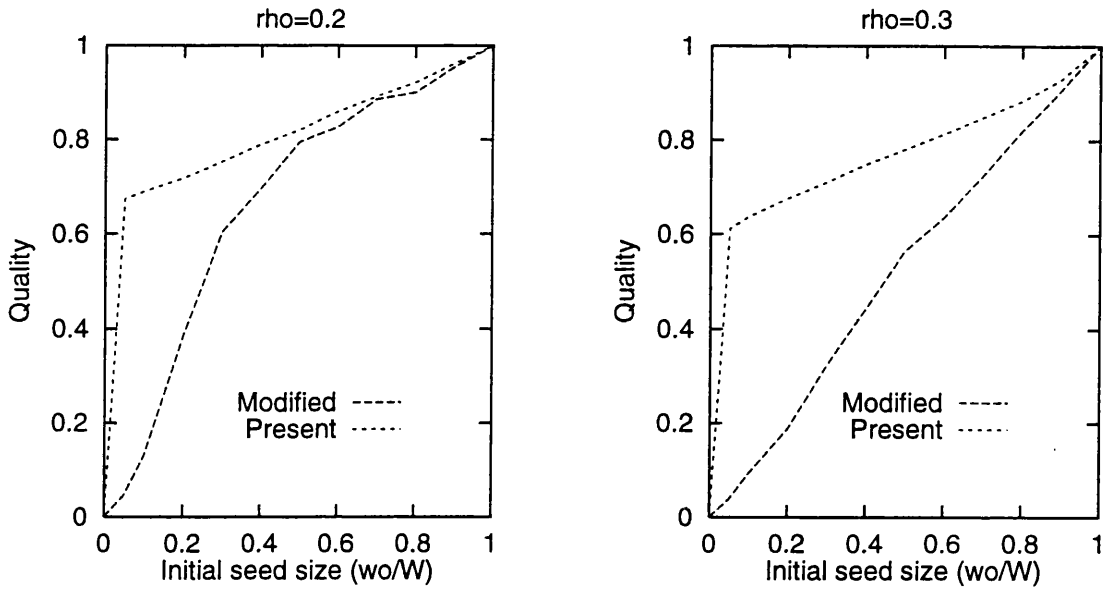


Figure 3.6: Analysis of the expected value of recall performance (figure 3.2) is superimposed with the optimal thresholds identified by table search algorithm (Present). The plot shows that there is a sequence of thresholds that results in far higher recall quality. In order to make a direct comparison, the exhaustive search is restricted to contain no spurious cells in the final stage of recall.

the thresholds from the table search method. This figure shows that the table based search results in higher recall quality than Gardner-Medwin's thresholding strategy (Gardner-Medwin 1976), suggesting that there are better threshold sequences. The recall performance predicted by the table based search is consistently higher than Gardner-Medwin's thresholding strategy, but this is particularly true if the recall is initiated using a seed that is a small fraction of the stored pattern.

In order to test the performance predicted by the present model, we searched through threshold sequences with a simulated network. It is of course impossible to evaluate all threshold sequences with a full simulation of the network. Instead, at each step in the recall process with a simulated network the five threshold settings with highest quality are remembered. All possible threshold

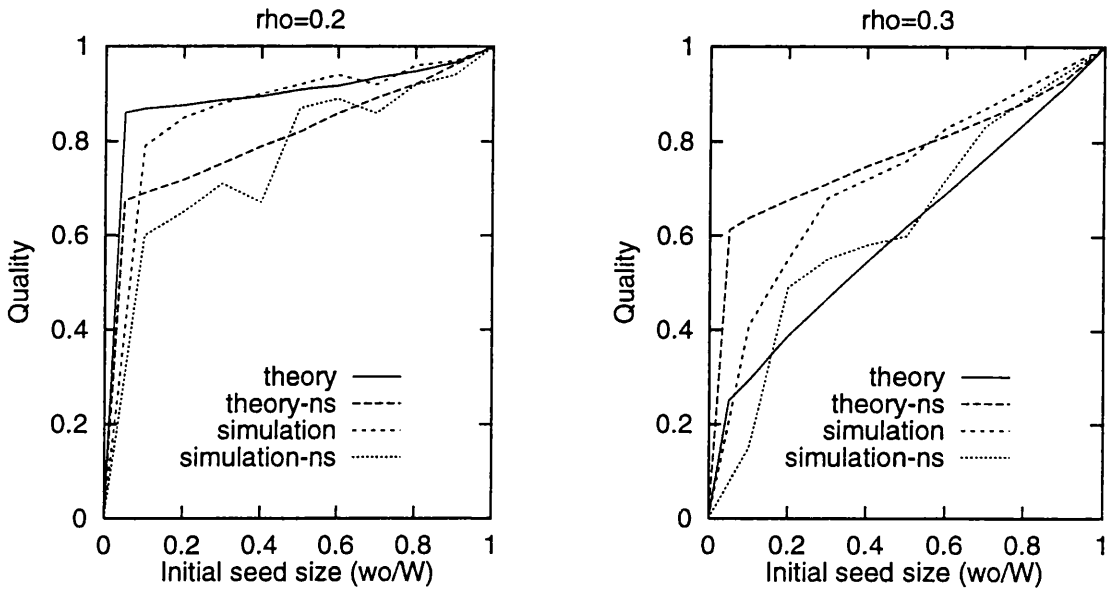


Figure 3.7: Comparison between expected performance and simulation results. $N = 5000, R = 500, W = 200$. In this graph the results of the search for optimal threshold sequence, based on theory, are compared with simulation results. In one case (theory-ns and simulation-ns) the comparison is between recall sequences which end with no spurious cells active, and in the other case (theory and simulation) the end point with best final quality is selected.

settings are tested for each of these five candidates, and the best five at the next step are retained. The number of progressive recall steps evaluated is also limited to five, resulting in $\sum_{i=0}^{5} i^5$ possible end points. The result of this simulation matches well with the theoretical prediction. It is shown, compared to the theoretical prediction, in Figure 3.7.

3.5 Concluding remarks and summary

In this study, it is shown that the prior theory overestimated the performance of the prior progressive recall strategy. Also, it is shown, with both an extension to the theory and by computer simulation of networks, that a better threshold sequence exists. In both cases, the optimal thresholding strategy included several steps with many active spurious cells.

Methods to identify optimal threshold sequences have been proposed and implemented in this chapter and the previous chapter.

Notes Part of this chapter has been published in Hirase and Recce (1995).

Chapter 4

Evaluation of the linear thresholding strategy

4.1 Introduction

In the previous chapters, two methods have been proposed to search for the optimal behaviour of the network. They are the tree based search and the table based search. The tree based search method has been applied to store the simulated states of an associative network, whereas the table based search is designed to derive the optimal thresholding strategy using the (theoretical) expected value dynamics. The basic method of the table based search was described in the previous chapter. Although the optimal performance obtained using the table search algorithm was compared with the simulated results, the thresholding strategy used to achieve this theoretical result has not yet been discussed. This chapter focuses on the obtained threshold sequence from the table search method.

4.2 Method — table based search

The result reported in the previous chapter was obtained using the formalism developed by Buckingham (1991). In chapter 1, it was shown that Buckingham's formalism is valid only for feedforward networks. This is because the

formalism does not take the temporal correlate between the current state and the next state into account (see appendix B for more detail). When c correct cells and s spurious cells are active in the network, the formalism proposed by Buckingham assumes that they are randomly chosen correct cells and randomly chosen spurious cells. This is correct for the first time step, where seed cells are randomly chosen. However, from the second step onwards, the active cells are not completely random. Rather they are chosen by means of the active inputs and the threshold value, i.e. active cells at the current state are prone to be active in the next state as well. Gibson and Robinson (1992) have published a set of formulae to estimate the expected dynamics of the recurrent recall that correctly takes the temporal correlates into account.

In order to acquire a more accurate estimate of the optimal performance of the associative networks and the threshold sequences associated with them, Gibson and Robinson's formalism must be incorporated into the table search algorithm.

Structurally, the partially connected recurrent autoassociative network proposed by Gardner-Medwin (1976) and that studied by Gibson and Robinson (1991) are identical. However, one crucial difference between the two network models exists in the recall paradigm. In Gardner-Medwin's model, the initially active cells (the seed pattern) remain active throughout the recall, i.e. their activity is supported by the external input to the network. In contrast, Gibson and Robinson assumed that the seed cells are transient, i.e. the seed cell is externally activated only in the beginning of the recall and their activity after the first time step depends on the effective input arising from the internal connections.

Unlike the network studied by Gardner-Medwin (1976), the seed cell is ex-

trinsically applied only once at the initiation of a recall, so that the dynamics of the network activation becomes compatible with the progressive recall equation proposed by Gibson and Robinson (1991). Another notable property of a transiently applied seed is that it allows inclusion of spurious cells in the seed pattern. Since the seed pattern is not persistently active, spurious cells in the seed cells will be removed if an appropriate threshold (or series of thresholds) is applied. Spurious cells receive less support than correct cells. It is in this spirit that an optimal threshold sequence for a transiently applied seed recall is assessed using Gibson and Robinson's formalism in this chapter.

4.3 Results

4.3.1 Search for the best thresholding strategy

The table-based search technique was first used to determine the maximum number of events that can be stored in an example network ($N = 6000, R = 3000, W = 150$). The recall performance of the modeled network was tested with increasing amounts of loading ($M = 600$ to 1200, steps of 200). A search for a threshold sequence was performed with each loading level of the memory. The maximum number of time steps in the threshold sequence was 15, which was found to be sufficient for successful progressive recall. Figure 4.1 shows the threshold sequences that were found using this search technique, with threshold values plotted against the time step of the progressive recall. As the number of stored patterns increases, there is a a corresponding increase in the number of steps required by the recall process.

The threshold sequences plotted in Figure 4.1 are not necessarily the unique set of values that result in the highest quality of recall. At lower levels of memory loading (smaller M) there are many threshold sequences which achieve

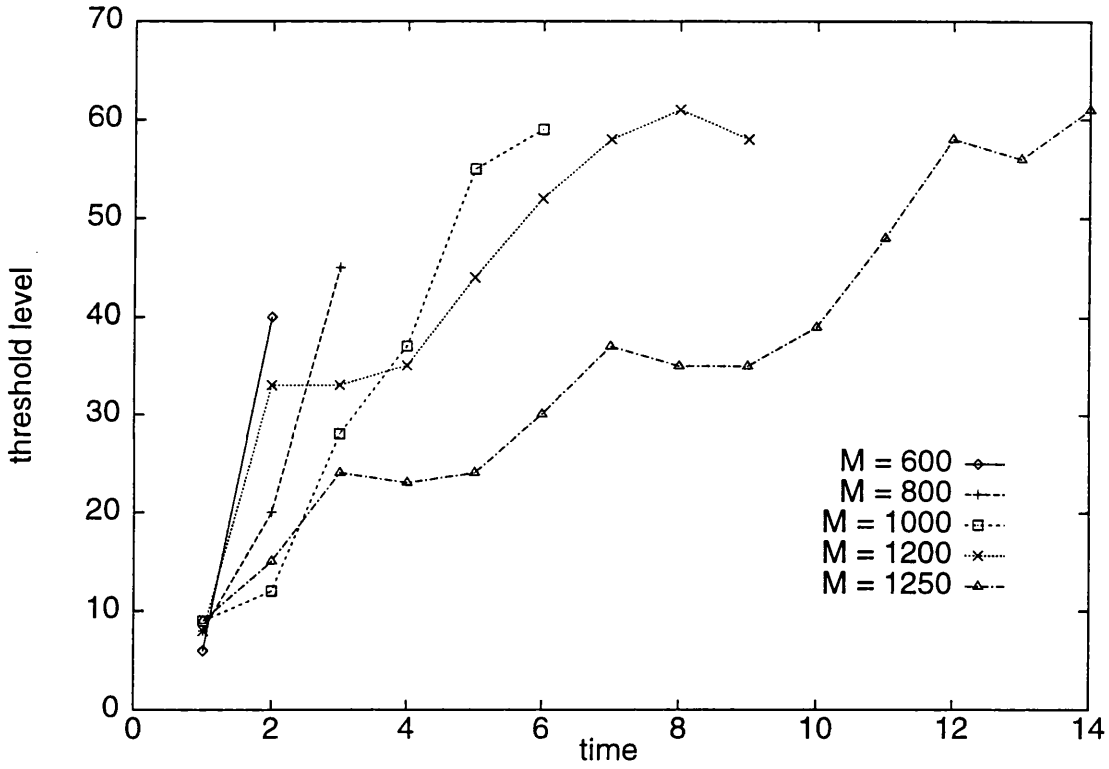


Figure 4.1: Results of the table-based search for an example network. The optimal threshold sequence is plotted for $N = 6000, R = 3000, W = 150, c_0 = 15$ with $M = 600$, 800, 1000, 1200, and 1250.

the same quality. The table only stores one of many ways in which a particular number of correct and spurious cells can be reached. Nearer the capacity limit there may be only one sequence that achieves the highest recall quality. We evaluated the range of threshold sequences that lead to the same quality by searching both from the highest to the lowest, and from the lowest to the highest threshold value at each time step. The two search strategies resulted in different selected threshold sequences, which are most different with smaller values of M.

The table-based search technique was used to find a theoretical prediction for the maximum capacity of the network. If the table-based search fails then there is no global threshold sequence by which a stored event can be reconstructed. Table 4.1 shows how the maximum capacity of the network varies with changes in the thresholding strategy. The number of events and the number of bits of storage per neuron are shown for modeled networks with a range of different event sizes ($W = 100, 150$, and 200). For the examples shown, the best recall sequence provides 14 to 42 times more information capacity than that achieved by the basic form of simple recall, described by Gardner-Medwin (1976).

There appears to some structure between the level of optimal threshold obtained from the progressive recall equation and the time step shown in figure 4.1. However, it is difficult to fit a mathematical function to describe the relationship between the two variables.

In a linear threshold strategy the value of the threshold is proportional to the total number of active cells in the network. The threshold (T) at time t of the recall is defined as $T(t) = \alpha w_{t-1} + \gamma$. In order to compare the results of the table-based search with a linear threshold strategy we re-plotted the

Thresholding Strategy	W100		W150		W200	
	events	(bits/syn)	events	(bits/syn)	events	(bits/syn)
Table based search	2300	(0.098)	1260	(0.0708)	790	(0.0555)
Simple recall	160	(0.00653)	50	(0.00284)	20	(0.00140)
Best linear without offset	1430	(0.0583)	830	(0.0447)	570	(0.0401)
Best linear with offset	2300	(0.0938)	1250	(0.0703)	790	(0.0555)
Best linear with offset (Simulation)	1900	(0.0775)	950	(0.0534)	600	(0.0422)

Table 4.1: Capacity analysis predicted by the theory and found with simulation. The table shows the maximum number of events that can be stored and recalled with an 85% quality given different thresholding strategies for $W = 100, 150$ and 200. Also included is the amount of information (measured in bits) stored per principal cell. ($N = 6000$,$R = 3000$

optimal threshold sequences shown in Figure 4.1 as a function of the network activation (resulting from the prior threshold) instead of the time step (Figure 4.2). Plotted in this manner the optimal threshold sequences appear to be essentially linear and have a small positive offset (y-intercept). The fact that a linear threshold performs so well cannot easily be derived from the theory (see appendix).

In order to model the role of an interneuron, Gibson and Robinson proposed a linear threshold strategy but did not pay much attention to the offset (y-intercept $=\gamma$) of the linear function. This thresholding strategy appears to be a good choice and has a much higher capacity, but does not reach the maximum capacity predicted by the table-based evaluation of the theory.

In the initial analysis there were no spurious cells in the seed pattern. We reran the table based search with spurious added cells to the initial patterns, while keeping the information content of the seed pattern fixed. That is, the

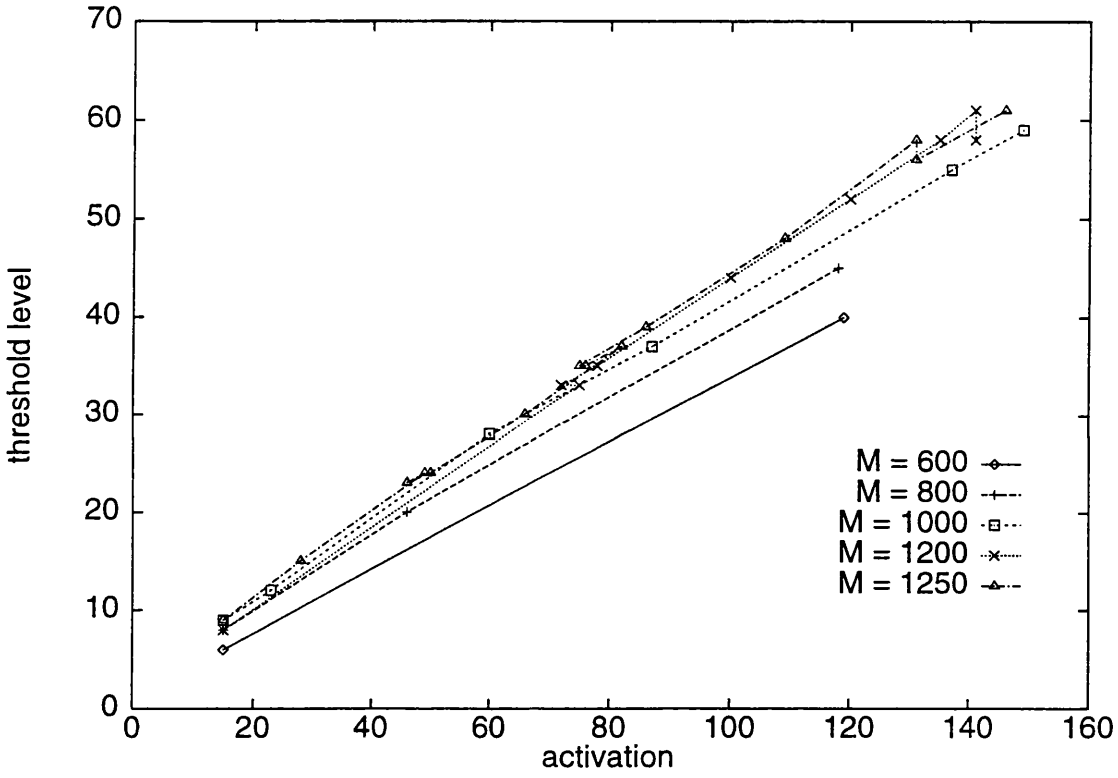


Figure 4.2: Progressive recall threshold sequences are plotted against the activation of the network. Note that these are the same data previously shown in Figure 4.1.

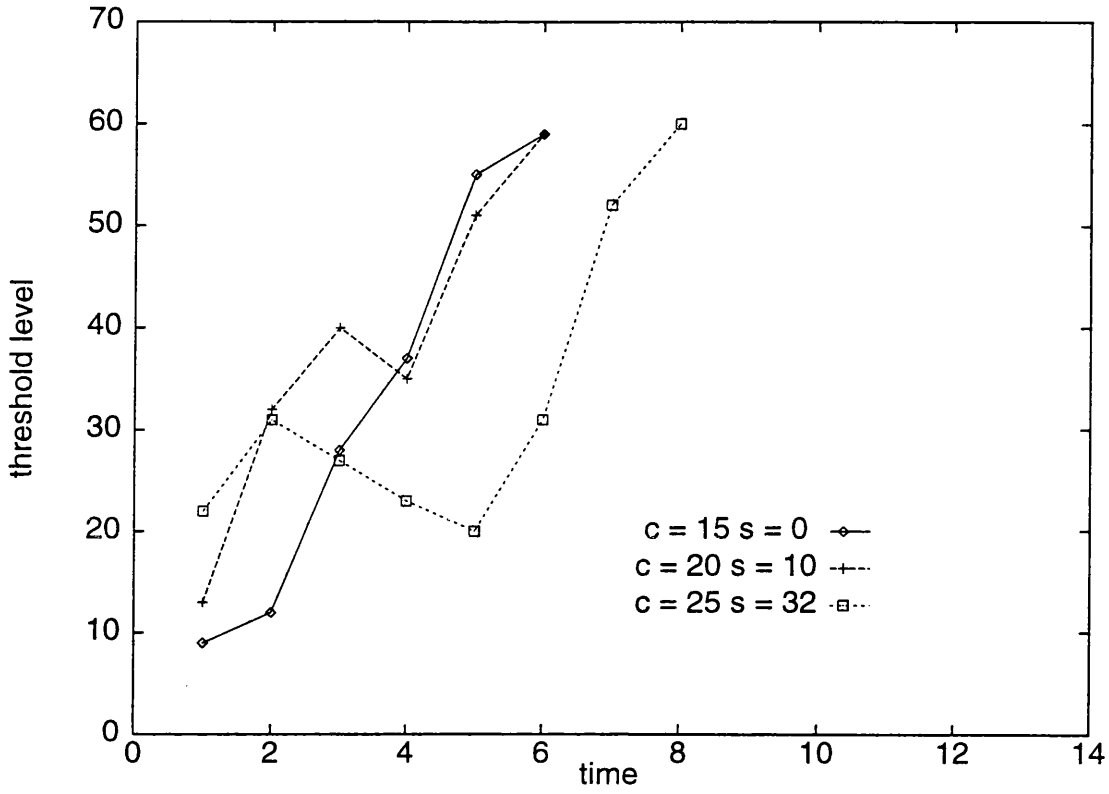


Figure 4.3: Results of the table-based search for an example network using initial patterns with spurious cells. The optimal threshold sequence is plotted for $N = 6000, R = 3000, W = 150, M = 1000$ with $(c_0, s_0) = (15,0)$, $(20,10)$ and $(25,32)$.

seed patterns with spurious cells are chosen in such a way that I_c, the amount of information required to correct the inconsistency with the complete event (see section 1.4.1 of chapter 1 for mathematical detail), is the same as that of the seed pattern without spurious cells.

Figure 4.3 shows the threshold sequences found by the table based search with three different amounts of spurious input in the seed pattern $[(c_0, s_0) = (15,0),(20,10),(25,32)]$. In each of these cases, as shown in figure 4.4, the optimal threshold sequence is a linear function of the network activation. This would have been difficult to predict since the activation is in the best case half spurious cells. If the number of spurious cells in the seed pattern is increased

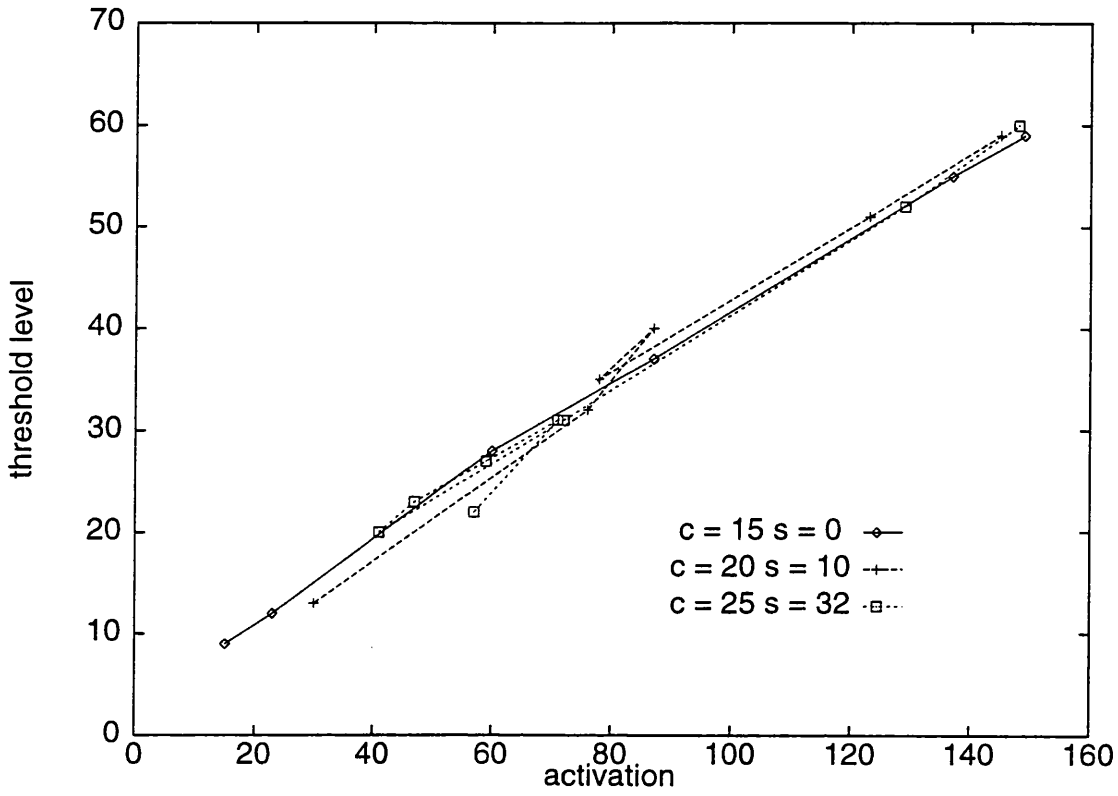


Figure 4.4: Progressive recall threshold sequences are plotted against the activation of the network using initial patterns with spurious cells. Note that these are the same data previously shown in Figure 4.3

much further, recall is not possible. Further, the slope and intercept of the optimal threshold formed with spurious cells in the seed pattern is almost identical to that found with the uncorrupted seed pattern.

4.3.2 Search for the best linear strategy

In their analysis of progressive recall, Gibson and Robinson (1992) evaluated a linear threshold strategy with no offset. They suggested that a positive offset value could be added to the linear threshold to damp oscillations, but they do not expect this modification to have a significant effect on the memory capacity of the network. Using Gibson and Robinson's formalism to predict the network activation, we evaluated a range of values for the slope and offset of the linear

threshold function. For each value of slope (α in [0,1.0] by 0.02) and intercept (γ in [0,10] by 0.1) the threshold sequence was followed for exactly fifteen time steps. The final quality of the recall process for each value of α and γ is shown in the sequence of three-dimensional plots in Figure 4.5. The four plots show the functional region of the associative network as the loading is increased ($M = 600, 800, 1000$ and 1200). Note that as the number of patterns increases beyond 800 the network fails to recall if the offset (γ) is zero. In many cases the associative memory reaches the final activation before the completion of the fixed length number of steps in the recall sequence. However if the memory does not stabilise then after further steps the quality will decrease and will not be considered successful recall in this analysis.

The results presented in Figure 4.5 show that the range of α and γ corresponding to successful recall changes with increased load (M). Since this region has a single peak the region corresponding to recall above a fixed quality corresponds to a single closed contour in a two dimensional plot of α against γ. Figure 4.6 shows how the ($\geq$ 85% quality) contour line changes with increasing loading of the network. The size of the region containing linear thresholds that result in high quality recall decreases with increasing M, but the functional region at large M is largely contained in the functional region at lower values of M. As shown in Table 4.1, with the addition of an offset ($\alpha = 0.42, \gamma = 2.8$, for $W = 150$) the theoretical performance of the linear thresholding strategy is essentially the same as with the optimal threshold sequence. This implies that, based on the theory, no arbitrary sequence of thresholds, even if it is matched to the parameters of the network, performs significantly better than a linear threshold.

A further question is the way in which the optimal values of α and γ vary

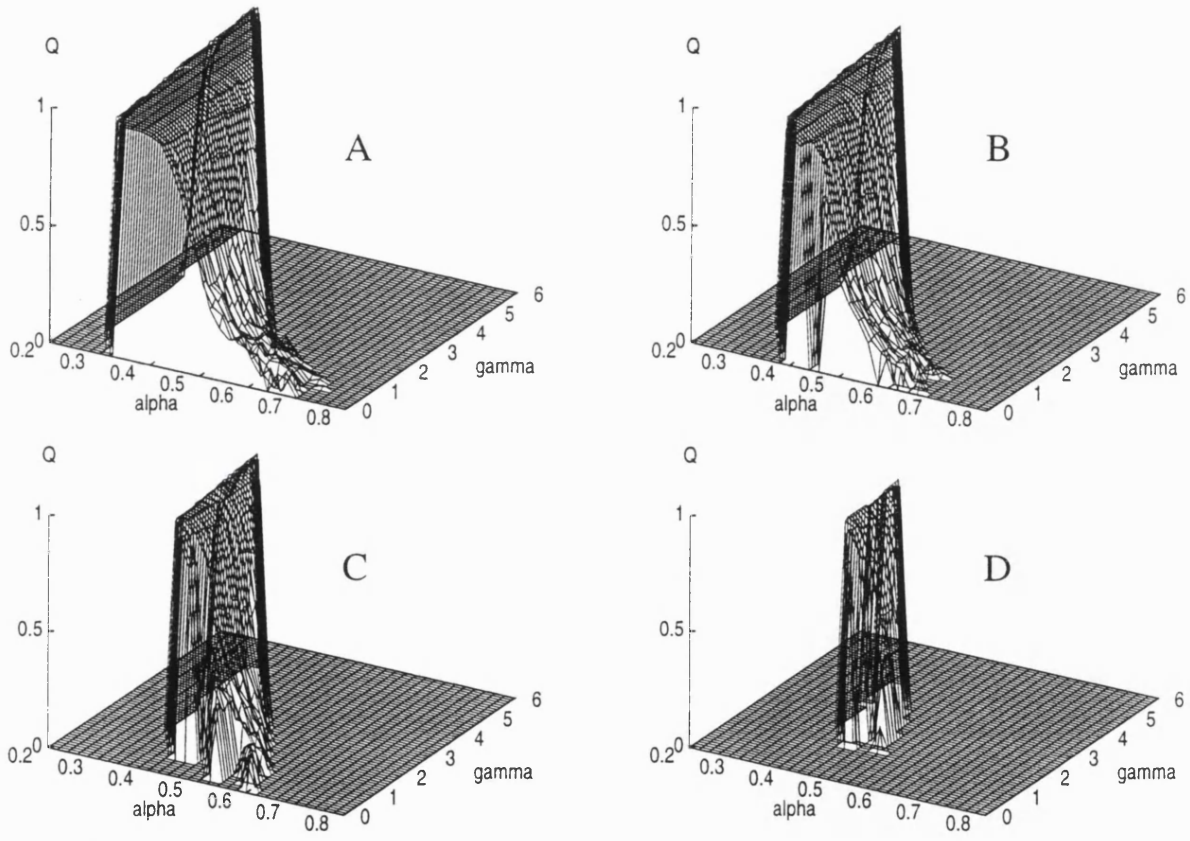


Figure 4.5: Quality of recall at 15th step is theoretically calculated and plotted for ranges of α and γ ($N = 6000$, $R = 3000$, $W = 150$). Each of the three dimensional plots corresponds to a different number of stored patterns; A:$M = 600$, B:$M = 800$, C:$M = 1000$, D:$M = 1200$.

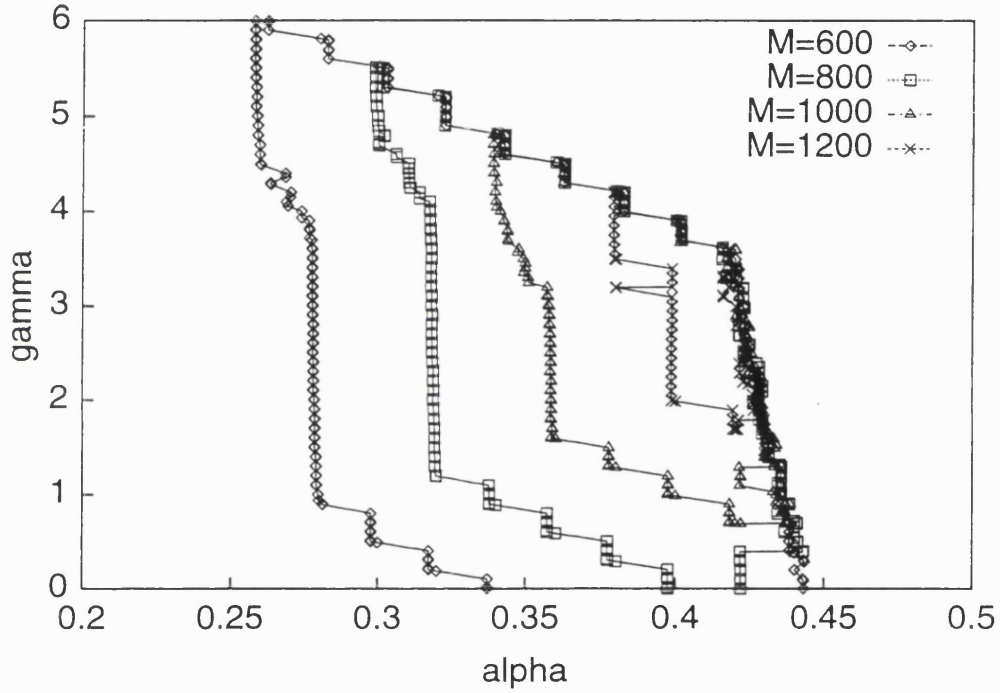


Figure 4.6: Four superimposed contour plots of the functional linear thresholds ($\geq$ 85% recall quality) for different levels of memory loading ($M = 600, 800, 1000$, and 1200. The linear thresholds are defined by the slope (α and offset (γ). The network has the same configuration as in figure 4.5.

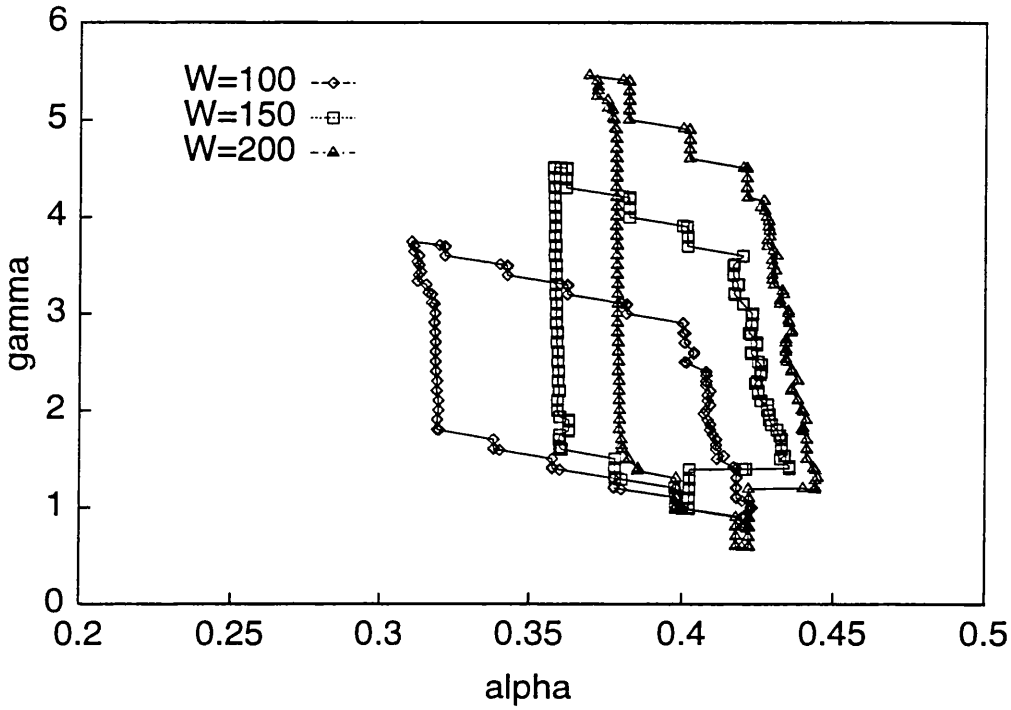


Figure 4.7: These contour plots of functional linear thresholds ($\geq$ 85% recall quality) for different activation sizes (W) in the same network configuration ($N = 6000, R = 3000$). Since the storage capacity varies with W, the three contours have different corresponding loading levels ($M = 1850$ for $W = 100$; $M = 1050$ for $W = 150$; $M = 650$ for $W = 200$). These loading level are selected so that the area of the region with the contour is a constant (100 block, $\delta\alpha = 0.02$, $\delta\gamma = 0.1$).

with other changes to the network. In order to obtain an initial measure, the size of the theoretically expected functional region ($\geq$ 85% quality) of the associative memory was measured for event sizes (W) of 100, 150, and 200. Since the capacity changes with event size the functional region was measured with a different level of loading for each value of W. The loading level was selected so that the functional region has a constant area of one hundred blocks (each block is 0.02 α by 0.1 γ). This set of contour lines are shown in Figure 4.7. Over this range of event size the functional region moves but still overlaps.

4.3.3 Simulation Results

Gibson and Robinson (1992) compare their theory of progressive recall with simulation results at both low and high levels of memory loading. In the low loading case the network had stored much less than its capacity and the high loading case the simulated network is beyond its capacity limit. Of course it is most difficult to match the performance of the theory to the simulation near the limit of the network's capacity.

In order to test the predictions of the theory we ran simulations of the three network configurations listed in Table 2. We evaluated 100 different linear thresholds for each example network (e.g. for $W = 150$, as in Figure 4.6, α in [0.2,0.47], by 0.03; γ in [0,6], by 0.7), and with increasing levels of loading. The thresholds selected for this test correspond to the high capacity region found using the evaluation of the theory.

Figure 4.8 shows the performance of the simulated networks using the linear model, parameterised by α and γ with four different levels of loading. Each of the points in this figure represents the average performance of 10 simulation experiments. Note that the shape and best values for α and γ predicted by the theory correspond well to the region of good performance in the simulations. However, the overall storage capacity of the simulated networks is lower (see Table 2) than theoretically expected. The maximum number of patterns that could be stored and recalled using the simulation corresponds more closely with the point at which the theory suggests that there is a functional area of 100 blocks as shown in Figure 4.7. In part the difference in performance between the simulation and theory is a result of the averaging of simulation results. For instance, if there are eight out of ten perfect recalls and two complete failures, the average is below the pass limit (85% quality).

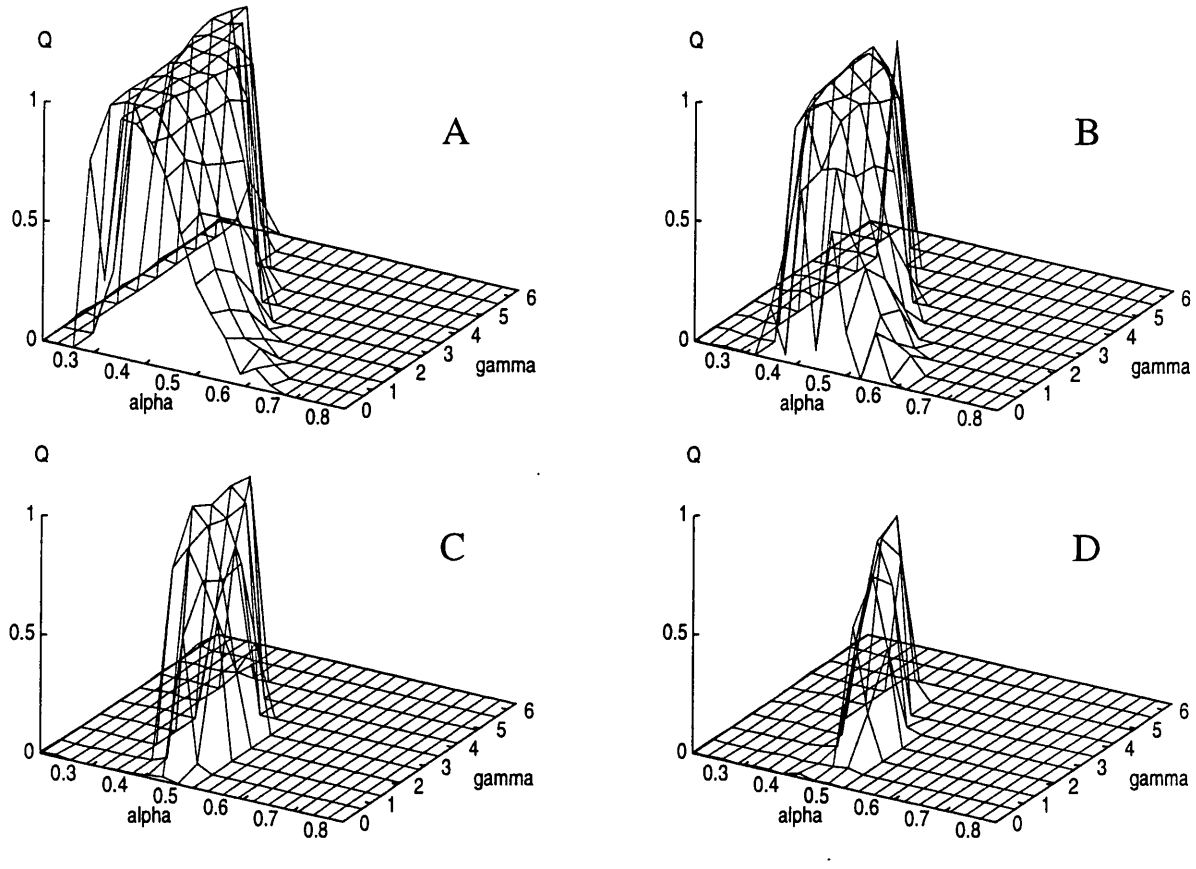


Figure 4.8: Four three dimensional plots of the quality of recall found in the simulation experiments ($N = 6000$, $R = 3000$, $W150$). The qualities found in 10 recall trials are averaged at the 15th time step and plotted for ranges of α and γ (A:$M = 500$, B:$M = 700$, C:$M = 900$, D:$M = 950$).

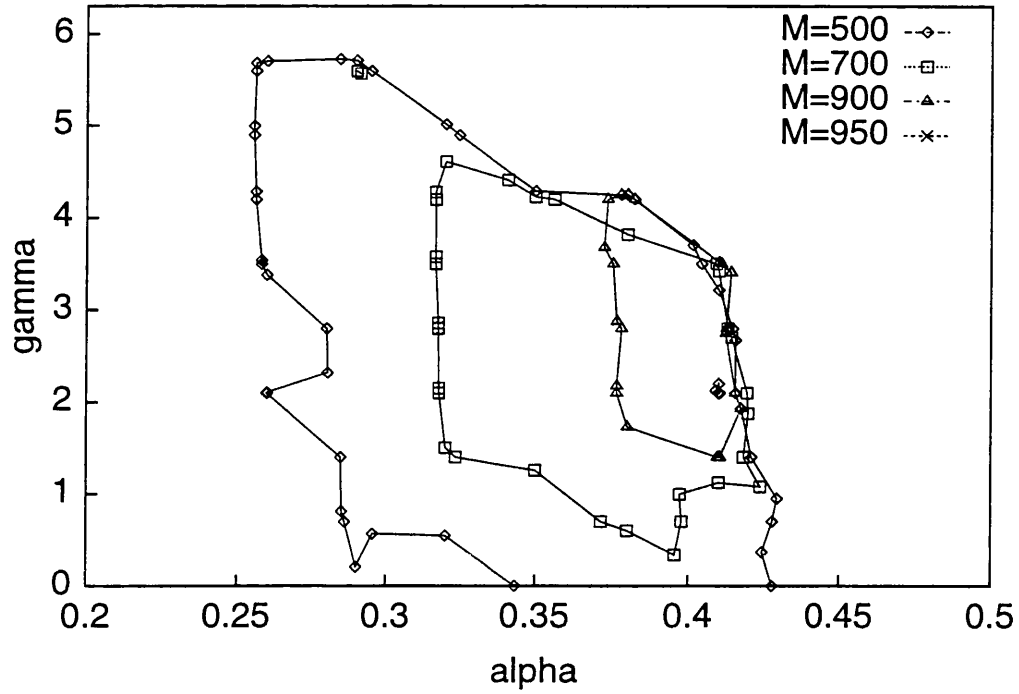


Figure 4.9: Four superimposed contour plots of the functional linear thresholds ($\geq$ 85% recall quality) for different levels of memory loading (M = 500, 700, 900, and 950. The linear thresholds are defined by the slope (α and offset (γ). The network configuration is the same as that of figure 4.8.

In a similar manner to that described previously, the contour lines corresponding to an $\geq$ 85% quality level were extracted from the simulation results, and are plotted in Figure 4.9. Note that the shape of the contour regions has been well predicted by the theory, and that the functional region at high loading is included in the functional regions at lower memory loading. As before, this implies that a single value of the slope and offset for the linear threshold works independent of loading.

4.4 Discussion

In this study the table-based search method is used to evaluate the performance of all possible progressive recall strategies in a simple form of associative memory. In this type of biologically plausible associative memory the best performance is achieved with sparse coding of information and with simple synaptic interactions that do not require a wide range of weight values or weights that change of sign. The results presented in this chapter, which are consistent with those previously reported by Amari (1989) and Treves and Rolls (1991), show that there is an increase in information capacity as the encoding of stored events becomes more sparse. The memory capacity (in bits) resulting from the best linear threshold strategy is twice as high with $W = 100$ as it is with $W = 200$.

The basic finding in this chapter is that the optimal thresholding sequence for progressive recall in this type of auto-associative memory is a linear function of the network activation. Unlike reported in Gibson and Robinson (1992) or Bennett et al. (1994), the recall capability of the network depends strongly on the choice of the offset value in the linear function, γ. In particular, it is important to have γ greater than zero for networks that are near the capacity limit.

In most of the present analysis the seed pattern contained only correct cells. We also applied the table-based search technique to a recall sequence in which the seed pattern included spurious cells. In the example cases presented here the resulting optimal threshold sequence was also a linear function, with approximately the same slope and intercept, as that found with an uncorrupted seed pattern. However, if the number of spurious cells in the seed pattern is increased further then the total activation in the network will result in a global

threshold that will de-activate all of the principal cells in the network.

The thresholding strategy found using the table-based search differs from earlier progressive recall strategies (Gardner-Medwin 1976; Lansner and Ekeberg 1985), in that it allows the activation of spurious cells. In the threshold sequences found using this method there is a consistent period, during the early states of the recall, in which many spurious cells are active. Other suggested thresholding strategies use cell-specific features (e.g. winner-take-all interaction; selection of the cell with the single highest activation; knowledge of the un-weighted as well as the weighted activation input) in order to increase the storage capacity of the network. In this chapter we have shown that substantial capacity increase can be achieved using a form of neural interaction which does not require this additional complexity.

The results obtained in this study are based on the use of the theory developed by Gibson and Robinson (1992). The predictions calculated using the theory match well to the results found from simulations. This match between theory and simulation makes it possible to efficiently analyse variations in the recall process. However, the theory needs to be extended so that a similar type of analysis can be performed on variations of the associative memory networks that are interesting either for an expected higher storage capacity or more biologically realistic properties. For example, these data have been analysed using example networks that have fifty percent connectivity. If significantly lower (biologically reasonable) connectivity levels are used the activation of the principal cell can no longer be described using a Gaussian distribution (Buckingham 1991) and the model is no longer a good indicator of the performance expected from the simulations. Furthermore, the model required that the extrinsic seed pattern is only briefly presented (during the first time step

). If this pattern is presented throughout the recall then the storage capacity should be higher. It would also be useful to extend the theory to cope with asynchronous recall, which might increase the storage capacity.

It is difficult to compare the performance of different methods for constructing associative memories. However, perhaps one measure is the number of bits of information that can be stored in each synapse. The number of synapses in the network is clearly one of the most biologically expensive features, and at least from the silicon circuit point of view it is more expensive to have a large number of different states in each synapse. Using this measure the associative memory described here compares favourably with a Hopfield network (Hopfield 1982) which has been shown to have a maximum capacity of 0.14 bits per synapse. The simulation results reported here have achieved 0.08 bits per synapse, using two state synapses instead of the large range of states required to encode real-valued synapses.

In the present theoretical and simulation-based evaluation of example networks, we found that the constants (α and γ) that define the best thresholding strategy do not vary with the number of patterns stored in the memory. This makes the implementation of this type of memory much simpler. In a biological implementation of this memory, both of these constants correspond to properties of the interneuron. For example, each of the input weights to the interneuron could be α/N and the baseline activity rate of the interneuron might be γ. It has long been suggested that the CA3 region in the hippocampus can be modelled using a sparsely and recurrently connected network of principal cells (Marr 1971). This type of memory model may eventually lead to a better understanding of biological neural network function.

Notes Part of this chapter has been reported in Hirase and Recce (1996, in press). I have been responsible for conducting the research including the algorithms for the table based search, writing the text, producing the graphs, and the programming. The linear threshold idea was a result of combined effort and discussions with Michael Recce.

Chapter 5

Interneuron Learning

5.1 Introduction

Up until this point, I have been concentrating on partially connected recurrent associative networks with binary synapses. It has been discussed that in these networks, determining the appropriate threshold value is critical to achieve successful recall. A thresholding strategy that determines the appropriate value of the global threshold must be worked out in order to make an efficient recall.

In the last chapter, the search for optimal threshold sequences using the table-search method revealed that the linear thresholding strategy achieves near-optimal recall performance. In this strategy, the value of the threshold is determined as a linear function of the number of currently active cells in the network. Using this strategy, the capacity of the network is measured both theoretically and in simulation. Having said the thresholding strategy is near-optimal, the result can be taken as a benchmark to measure improvements that occur when modification are made to the network.

In this chapter, a new concept called *interneuron learning* is introduced. The interneuron learning concept provides a local threshold for each cell rather than a global threshold by allowing multiple levels of synaptic efficacy in synap-

tic projection from the interneuron to pyramidal cells. To start with, the need for a local threshold is discussed in the next section. Then, the interneuron synaptic modification rule is studied. Following that, simulation results of a network with interneuron learning are compared with that of an associative memory network using the linear thresholding strategy.

5.2 Overlap problem

The main issue addressed so far in this thesis is the thresholding strategy that maximises the number of retrievable stored events (the capacity). When more events are stored beyond the capacity limit, recall fails catastrophically. The mechanism of recall failure has not yet been addressed in this thesis. This section gives an intuitive understanding of the reason that recall is more likely to fail with higher network loading.

Suppose five memory events, as in figure 5.1, have been learned by the associative network. As depicted in figure 5.1, cells experience different numbers of events during the learning. For instance, cell a does not participate in any events while cell b participates in all of the five events. In fact, the number of events a cell experiences during learning is the unit usage (as defined in section 3.2 of chapter 3). Furthermore, when random patterns are stored, the unit usages of the cells have the binomial distribution $\mathcal{B}\left(\frac{W}{N}, M\right)$.

When learned events do not have any active cells in common, the events are *orthogonal* to each other, otherwise, they are said to have an *overlap in representation.* It is not difficult to see that as the network is loaded with memory events, the probability that the learned events remain orthogonal to each other becomes extremely low, as the active components of the memory event is selected in a random manner.

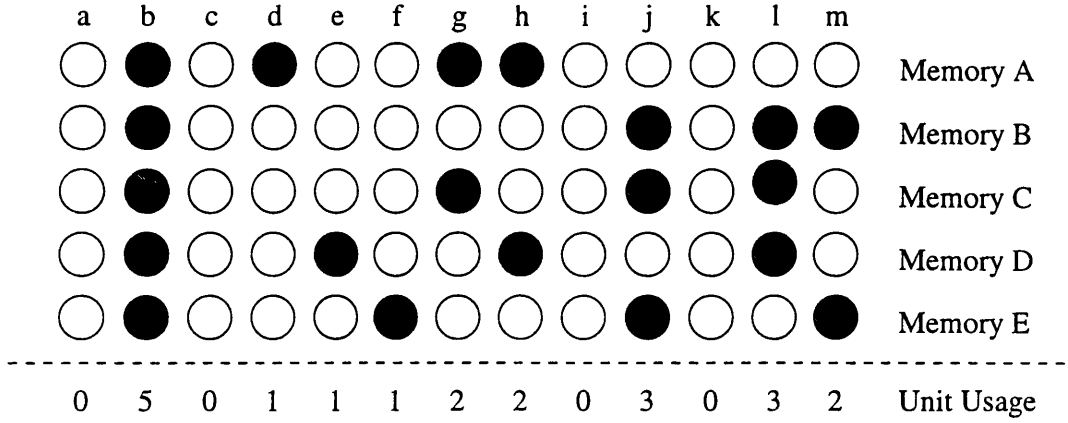


Figure 5.1: Overlaps in memory presentations — filled circle represents the active cells in a memory event and unfilled (white) circle represents the inactive components in the memory event. 5 memory events are shown using 13 cells of which 4 cells are active in each of the event. ($N = 13$, $M = 5$, and $W = 4$). The cells are labelled from a to m and the events are labelled from A to E. In the bottom of the figure is the unit usage for each cell.

Gardner-Medwin distinguished between *common cells* that participate in a disproportionate number of events and *distinct cells* that are in fewer events (Gardner-Medwin 1989; Gardner-Medwin and Kaul 1995). In other words, common cells are the cells with high unit usage values and distinct cells are the ones with low unit usage. These terms are defined rather vaguely, but it is sufficient to understand that as the network becomes heavily loaded, cells tend to be more common and less distinct since there will be many overlaps among the memory events.

Now, what are the significance of commonness or distinctiveness of the cells in memory recall? Suppose all active cells are the common cells at a moment during a recall. For instance, imagine recalling the memory event B with the memory configuration in figure 5.1 and active cells are b and j. Although both of the cells belong to event B, they are not specific to the event, i.e. they also belong to events C and E. This means that the likelihood the active cells

promote the other correct cells (l and m) is the same as the likelihood the active cells promote the cells which belong to event C or D. This is bad news for the recall process because the complete recovery of the event is impossible in this case.

The opposite is also logical. When the active cells consists of only the distinct correct cells, the event is likely to be successfully recovered as the output from active cells projects to the recalling event with high specificity. It is significant to note that distinct cells provide higher information content than common cells in information theory.

In short, the distinct cells provide more informative inputs to the other correct cells to build the event. On the other hand, the common cells, when activated, are prone to promote spurious cell activation.

A natural strategy to achieve a successful progressive recall is to promote the distinct cells at the initial stages of the recall process and inhibit common cells until later steps in the recall process. In this way, after a few steps of the recall, active cells becomes highly specific to the recalling event. Once a cell assembly with distinct correct cells is established, it is relatively easy to recover the rest of the cells in the stored event.

One way to implement such a recall strategy to introduce a more complex learning rule for the synapses between the principal cells. For example, one can reduce the synaptic efficacy between common cells, and enhance the synaptic efficacy between distinct cells (Crick and Mitchison 1983; Hopfield et al. 1983; Gardner-Medwin 1989; Gardner-Medwin 1991). Of course, this would require real valued synapses for all of the synapses arising from the principal cells.

Alternatively, it is possible to not change the principal cells' synapses and to introduce multiple levels of synaptic strength in the projection from an

inhibitory interneuron to the principal cells. In this model, the synaptic weight from the interneuron depends on the unit usage of the cell, so to inhibit the cells with high unit usage more strongly than the cells with lower unit usage.

5.3 Model and analysis

As described in the previous chapters, the network consists of N principal cells and one interneuron. The principal cells make R recurrent connections to other randomly selected cells. The example net studied extensively in this chapter has 50% synaptic connectivity (i.e. $R = N/2$). The network learns M events, each containing W randomly selected, active principal cells. The recurrent synapses are initially set to zero (i.e. ineffective), and are modified by the clipped Hebbian rule.

The interneuron receives a fixed weight (value of one) synaptic connection from every principal cell. Hence its activity is proportional to the number of active principal cells. The interneuron in turn projects to each principal cell and forms a inhibitory synapse (see figure 5.2). The activity of the interneuron essentially determines the level of threshold for the principal cells as they are binary McCulloch-Pitts neurons with a step-function threshold at zero, i.e. the behaviour of principal cell i is determined by $\Theta[I_i + \sum_{j=1}^{N} w_{ij}]$, where $\Theta[\cdot]$ is the Heaviside function and I_i is the inhibitory effect received from the interneuron (I_i is a negative quantity).

Global threshold model

This is the conventional partially connected recurrent associative network model studied so far in this thesis. Without interneuron learning the inhibitory synapses to the principal cells have equal, fixed synaptic weights. Therefore, each principal cell applies the same sequence of threshold levels. Previously I

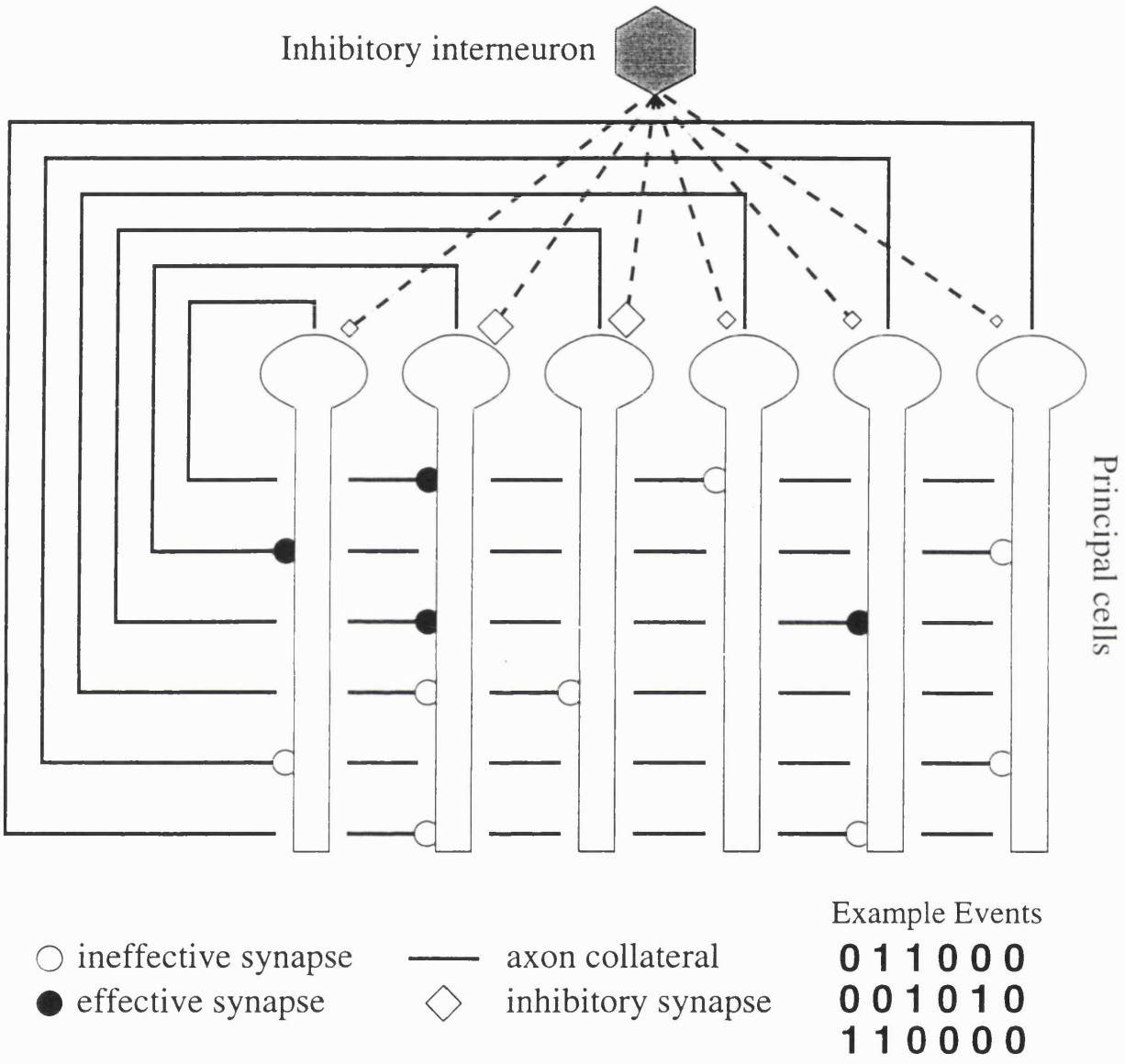


Figure 5.2: A schematic diagram of the associative memory network ($N = 6$, $R = 2$, $M = 3$, $W = 2$). Each of the principal cells sends its output via the collaterals (thick line). Synaptic contacts are made from the collaterals to other (randomly selected) principal cells. Some of the synapses are effective (black circle) as a result of learning the example events. The interneuron has an inhibitory connection to every principal cell. In the interneuron learning model, the interneuron synaptic weights are changed according to the target cell's unit usage. Inputs to the inhibitory cell are omitted in this figure.

tested all possible thresholding sequences and found that the best threshold sequence can be approximated by a linear function of the number of active cells in the network. In this case the inhibitory input to each principal cell is given by $-I_i(t) = \alpha * w(t) + \gamma$, where $w(t)$ is the number of total active cells at time (t) and α and γ are constants. Furthermore, the coefficients α and γ do not depend on the network loading (M) (for the example network, $\alpha = 0.41$ and $\gamma = 2.1$). The best thresholding strategy is the same linear function, when the seed pattern contains a small number of spurious cells (chapter 4).

Interneuron learning model

In this model, the interneuron has a fixed weight (value of one) synaptic connection from every principal cell. Hence its activity is proportional to the number of active principal cells. The interneuron also projects to each principal cell and this synaptic weight increases each time that the principal cell is part of an event, and thereby co-active with the interneuron. Synaptic weights do not change during recall.

For a principal cell, the modified interneuron input is analogous to a cell-specific threshold function, that varies with the number of events the cell experienced during learning (*unit usage*=m). We first find this cell-specific threshold function and then define a method for changing the synaptic weights of the interneuron projection. By extending prior work(Buckingham and Willshaw 1993) to fit the present model, the probability ρ that a recurrent projection to a principal cell becomes effective is:

$$\rho(m) = 1 - \left(1 - \frac{W}{N}\right)^m$$

The probability that a principal cell participates in one event is W/N, so the probability P_e that a principal cell experiences m events out of M events

is binomially distributed with W/N, i.e. $P_e = \mathcal{B}(W/N, M, m)$. Based on this distribution, the probability that a spurious cell with m unit usage receives exactly r effective inputs given w active correct cells is:

$$P(\mathbf{D}_{spur} = r|m) = \mathcal{B}\left(\rho(m)\frac{R}{N}, w, r\right)$$

I choose the threshold $T(m)$ to keep the probability of spurious firing small (i.e. $T(m)$ is the minimum value that satisfies the following condition):

$$1 - \sum_{i=0}^{T(m)} P(\mathbf{D}_{spur} = i|m) < 0.0001$$

Here, the spurious cell firing probability is used as the key condition for the learning rule for interneuron synapses. The primary reason for setting the probability of spurious cell activity to be essentially zero, is that it makes the remainder of the analysis simpler. In the prior chapters, the objective was to find the rule for changing thresholds that resulted in the highest capacity, while here a simple learning rule is sufficient to test the concept of interneuron learning. When spurious cells are inhibited to the level of null activity it is likely that stable recall is achieved. Once the idea of interneuron learning is validated, a more complex and efficient learning rule can be developed.

In Figure 5.3A the calculated threshold levels are plotted, for a range of different activations, as a function of the unit usage for an example network ($N = 6000; W = 150$). Since the threshold (T) function is linearly dependent on unit usage (Figure 5.3A) it is approximated to be $\alpha(w)m + \beta(w)$, where the slope α and y-intercept β are both functions of the network activation w. These slopes and intercepts are plotted for a range of activation levels in Figure 5.3B, and both have a linear dependence on activation ($\alpha(w) = Aw + B; \beta(w) = Cw + D$). By substitution:

$$T_i = (Am + C)w + Bm + D$$

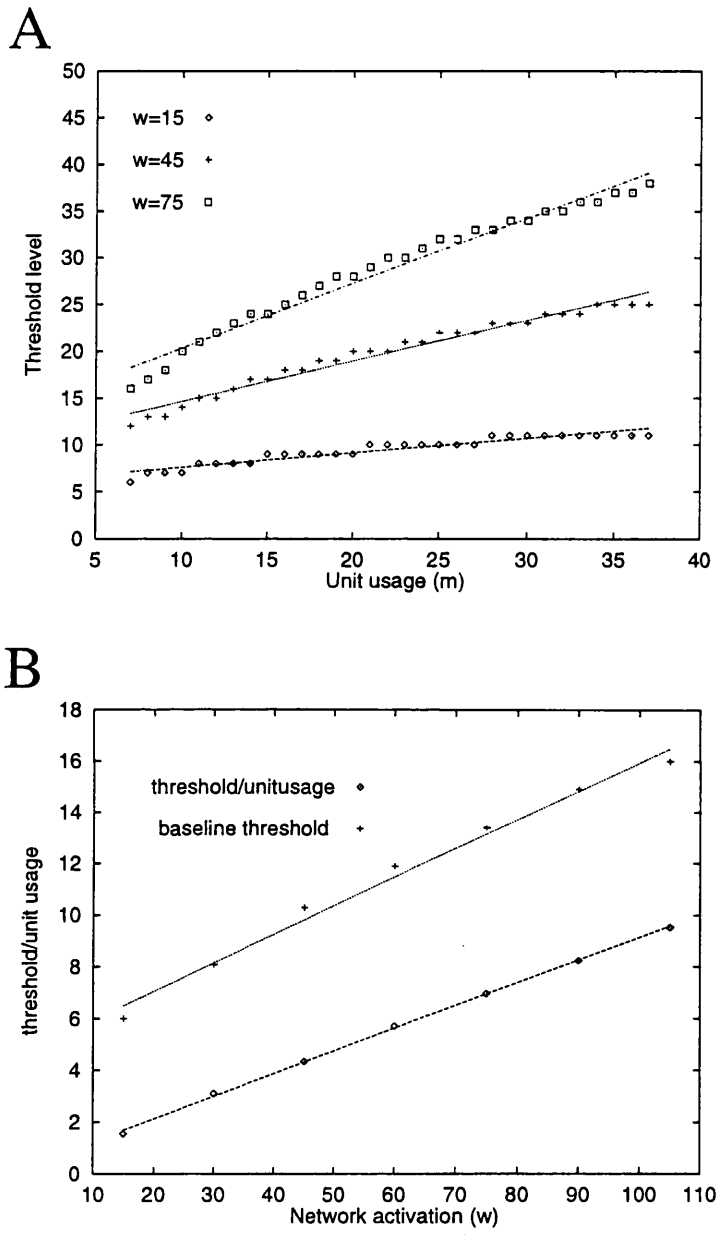


Figure 5.3: A: Minimum theoretical threshold levels to inhibit spurious firing are plotted against the unit usage of the cell. The threshold level changes according to the current activity of the network (w). w consists of correct cells only. The plots indicate that the three levels of network activation (w=15,45, and 75) can be approximated to a linear function. The network configuration is $N = 6000, W = 150$. B: The slopes and intercepts of the lines in figure 1A are plotted against the activation level of the network (w). A least square linear fit to the data is included in the plots.

The output of a principal cell is the step function (Θ) of the recurrent inputs minus the threshold. Therefore D is the independent offset threshold of the cell, B depends only on the prior unit usage of the cell (not the current input), and $(Am + C)w$ is input from the interneuron. In this interpretation, C is the initial value of each synaptic projection from the interneuron to a principal cell and with each stored event involving that principal cell the synaptic weight increases by A. In the example plotted in the figure $A = 8.8 \times 10^{-3}$, $B = 3.8 \times 10^{-2}$, $C = 0.11$, and $D = 4.4$.

In order to estimate the relative contributions of each term in the threshold function, I compute the expectation value of m ($\langle m \rangle = WM/N$). Near the capacity limit $M = 1400$, $\langle m \rangle = 35$, $w = W = 150$, $A\langle m \rangle w = 46.2$, $Cw = 16.5$, $B\langle m \rangle = 1.3$ and $D = 4.4$. For all values of activation (after application of the 10% seed pattern) and for all values of $\langle m \rangle$ the $B\langle m \rangle$ term is smallest. Furthermore, B does not reflect inputs from interneuron plasticity, so in the following simulations B is set to zero.

5.4 Simulation Results

To test the effect of interneuron plasticity I evaluated the storage capacity of an example network ($N = 6000$, $R = 3000$, $W = 150$).

The interneuron projection starts with a synaptic weight of $C = 0.11$ to each principal cell and increments it by $A = 3.8 \times 10^{-2}$ each time that the principal cell is active in a stored event. The example network is assessed with a pattern completion task (seeded recall). The network is initially activated with 10% of correct cells of a randomly chosen learned event (and no spurious cells). This initial cue is extrinsically applied only once at the initial stage of recall. Through the process of multi-step recall using the recurrent connections, the

event is recovered in successful recall.

As shown in Figure 5.4B, without modifying the interneuron synapses, performance of this network deteriorates after 950 patterns have been stored. Figure 5.4 A shows the average performance of this network as a function of the parameter D after 1400 patterns have been stored. The value of D has been varied since part of the calculated offset B was set to zero. Figure 5.4B compares the best performance achieved without interneuron learning to the capacity with the best value of D from figure 5.4A.

Both with and without interneuron learning the process tends to result either in *stable recall*, where the activity level stays in a reasonable range (i.e. $W \pm 50\%$) during the final steps of recall or *failed-recall*, where the activation level is either much too high with thousands of spurious cells, or almost zero. Figure 5.5 reconfirms this observation. The figure shows a histogram of the distribution of recall quality found in the simulations. The results from all levels of loading are included in this distribution. The bimodality nature of the distributions indicates that recalls result in either stable recall or total failure.

In order to find the recall failure mechanism, the probability of the occurrences of stable recall (success probability) and the average quality of recall for the stable recalls with various loadings of the network are investigated (figure 5.6).

The probability of successful recall without interneuron learning drops steeply after $M = 900$ patterns have been stored, while with interneuron learning, the probability of successful recall drops gradually after $M = 1250$. The proportion of the recurrent excitatory synapses that have been modified, $\rho = 1 - (1 - W^2/N^2)^M$ is 0.43 when 900 patterns have been stored and 0.54

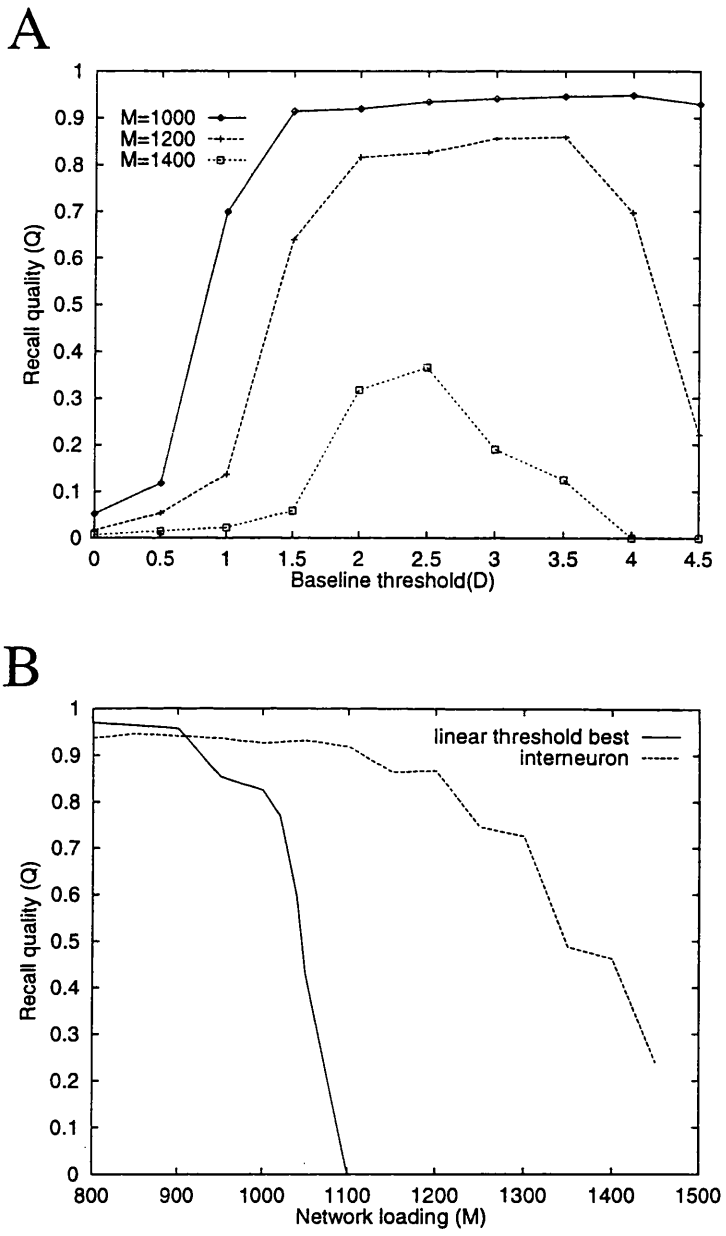


Figure 5.4: Simulation results for the example network (N=6000, W=150) A: Averaged recall quality is plotted against the value of baseline threshold (D in the text). The network configuration is the same as in figure 1. B: Performance comparison between global thresholding and interneuron learning. Average recall quality is plotted against loading of the network (number of learned events.) The network with interneuron learning has 50% more capacity.

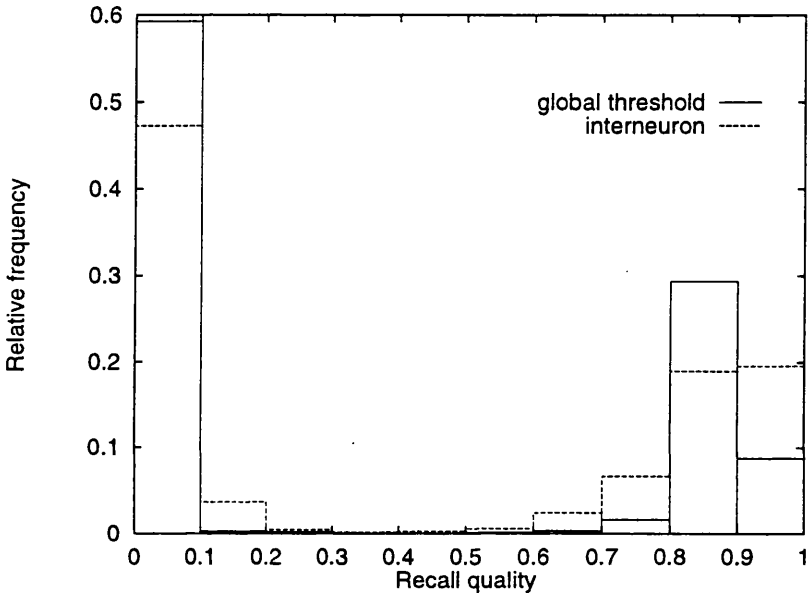


Figure 5.5: A histogram of the recalled quality: The final recall quality is a bimodal distribution indicating that the results tend to be either good stable recall or failed-recall. The quality in failed-recall is essentially zero, whereas stable recall has quality above 0.6 in most cases. The results from all levels of loading are included in this distribution. The solid lines are the distribution without interneuron learning, and the dashed lines are with interneuron learning. Note that distributions are bimodal for both global threshold model and interneuron learning model.

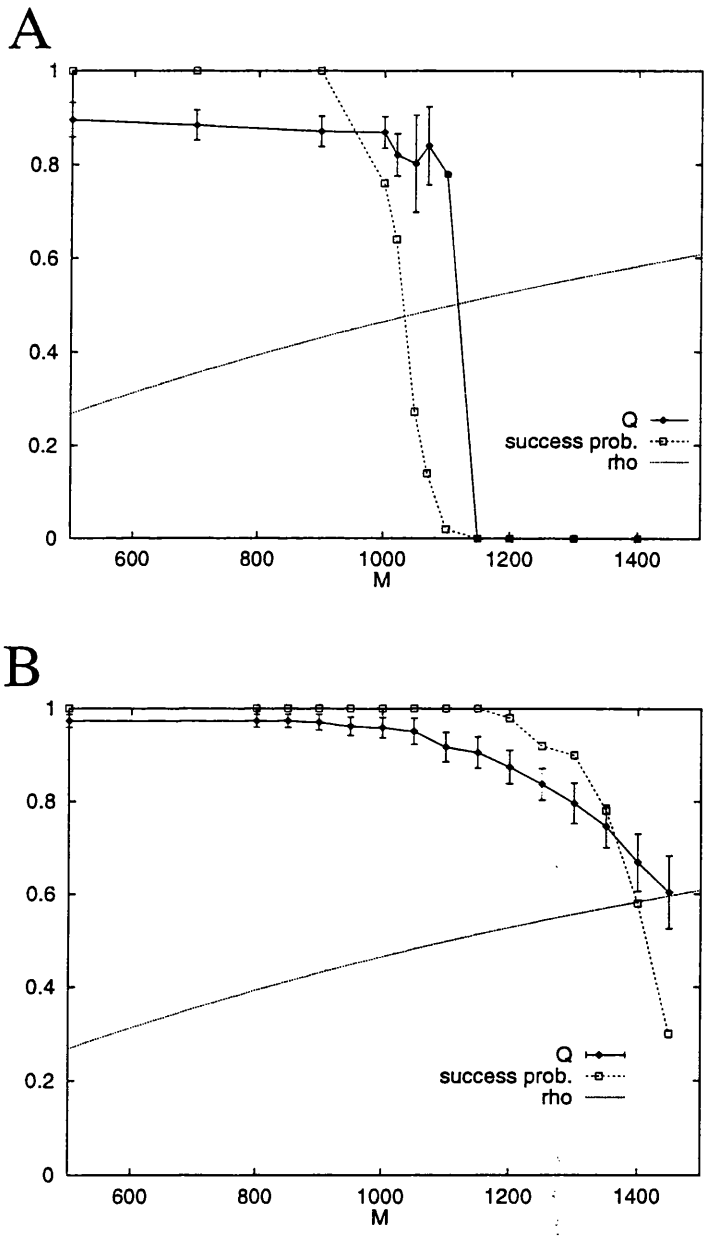


Figure 5.6: Performances of associative memories: Each point in the graph represents average of 250 simulations with 5 different weight matrices. Also plotted is ρ, the proportion of the modified principal cell synapses. **A** global thresholding model, **B** interneuron learning model.

when 1250 patterns have been stored. In a previous study (Gardner-Medwin 1976) it was estimated that this auto-associative memory would fail when 50% of the recurrent synapses have been modified. In the simulations that did not include interneuron learning, the recall quality is maintained at a constant level (c.a. 0.9) in the range with a high probability probability of successful recall, but drops rapidly. On the other hand with interneuron learning the probability of successful recall remains high for significantly longer, and the recall quality drops smoothly as the network loading is increased. The gradual decrease in quality is a direct result of the increase in the number of common cells which develop thresholds that are sufficiently high that they are never active. Furthermore, the suppression of the common cells decreases the effective size of the seed pattern, which eventually leads to a drop in the probability of successful recall.

The recall performance of the network is slightly reduced when the initial seed input includes spurious cells (see figure 5.7). However, the spurious cells only reduced the probability of successful recall and did not affect the quality of the recalled patterns.

5.5 Discussion

In this chapter, I have described how Hebbian plasticity, in the projection from a single interneuron to a sparse network of principal cells, can reduce the effect of overlap on the performance of an auto-associative memory. In the present analysis I found a threshold for individual cells that depend on the unit usage of the cell. An implication of this threshold is that the best values of the constants (A, C, D) will not depend on M. Therefore the constants depend only on the topology of the network and on W, parameters that may be set

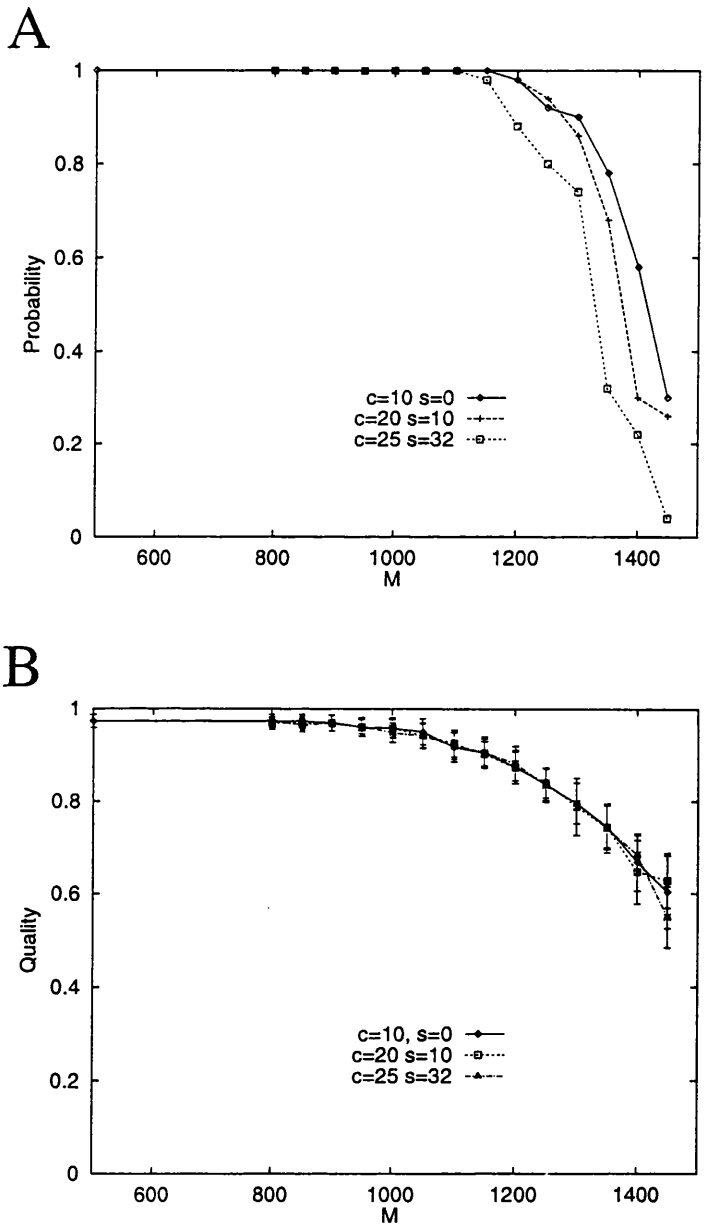


Figure 5.7: Performance of associative memory with interneuron learning, and with corrupted seed input. Results are presented from three configurations of initial seed input (correct,spurious) = {(15,0),(20,10),(25,32)}. Each of these seed inputs provides the same amount of information about the stored pattern to the associative network. Both the probability of successful recall (A), and the quality of the recalled patterns (B) are plotted.

genetically.

The interneuron learning model, described in this chapter, results in a significant increase in the storage capacity of an auto-associative memory, without adding complexity to the principal neurons or requiring multi-valued synapses in the more numerous ($N \times R$) feedback connections between principal cells. After all, when a neural network circuit is implemented in a VLSI, synaptic contacts take up the most of the chip area. If all of the principal cells' synapses are multi-valued, the VLSI circuit becomes considerably more complicated and requires much more space. The interneuron learning model introduces N multi-valued synapses in total which is far less numerous than introducing multi-valued synapses in the principal cell network that contains RN synapses.

Rather than applying a conventional analysis, and only examining the averaged quality of recall, I separated this measure into the probability of successful recall and the average recall of the successful recalled patterns. Using these independent measures I demonstrated that for both global thresholding and interneuron learning networks, the quality of recalled events is sustained at a nearly constant level when a stable recall occurs. Furthermore, as the network load (the number of learned events) is increased, the cause for degradation of recall performance is a decrease in the probability of stable recall. In this case the network fails when there is insufficient positive feedback within a cell assembly.

In physiological investigations of systems of neurons that perform putative memory functions, interneurons have received much less attention than principal (pyramidal) cells. While there is good evidence for synaptic plasticity among pyramidal cell networks, plasticity in the synaptic projections

from interneurons has not been intensively studied, although there appears to be some evidence for interneuron synaptic plasticity in recent studies (see the next chapter for more detail.) This is partly due to the fact that interneurons are much less numerous, and therefore it is more difficult to record from them. The improvements found in the networks described here would predict that when these studies are eventually performed that the projection from the interneuron will be modifiable, will have many different possible synaptic weights, and will be modified by the Hebb rule. Discussion on the biological aspects of the associative memory model are presented in the next chapter.

Notes Part of this result was presented in Computational Neuroscience meeting 1996a (CNS*96) in Boston, MA.

Chapter 6

Associative networks and the hippocampus

6.1 Introduction

The hippocampus is an elongated, C-shaped structure with the septotemporal axis running ventrocaudally from the septal nuclei. It can be divided into four cortical regions; the dentate gyrus, the hippocampus proper (which can be further subdivided into CA1-CA4), the subicular complex, and the entorhinal cortex. Inputs to the entorhinal cortex come from essentially all association areas in the cerebral cortex. Therefore, the hippocampus receives highly processed multimodal information. Primarily, sensory input arrives from the entorhinal cortex onto the granule cells via the so-called perforant pathway. In general, the granule cells makes axo-dendritic connections to the excitatory CA3 pyramidal cells via the mossy fibres as well as to some inhibitory basket cells. The CA3 pyramidal cells send collaterals (the Schaffer collaterals) to the CA1 pyramidal cells, but they also project back to the CA3 pyramidal cells (the associational connections). The CA1 pyramidal cells project back onto the entorhinal cortex as well as the subiculum. The subicular cells project onto the entorhinal cortex (summarised in Brown and Zador (1990)). The hippocampus, therefore, is located in the end stream of the sensory processing

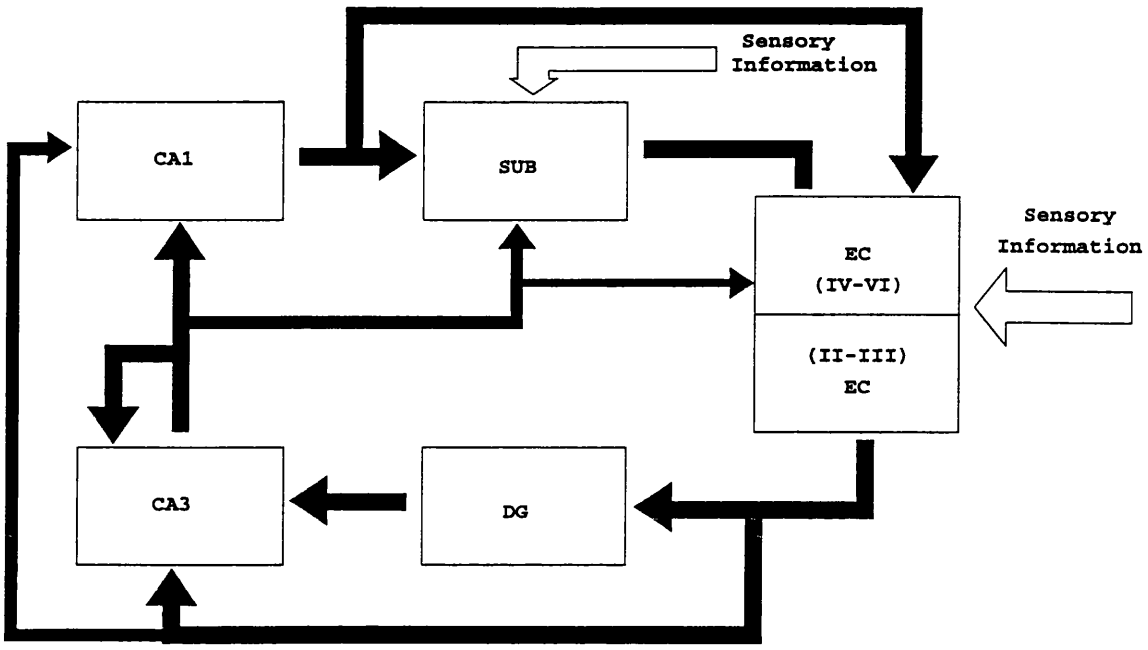


Figure 6.1: Basic excitatory connections of the hippocampus. (Adopted from Brown and Zador (1990))

and is in a very strategic position that is able to process polymodal information (see figure 6.1 for a function block diagram). One of the brain functions that require such high quality information is memory formation.

The notion that the hippocampus functions as an associative memory device was first theoretically proposed and analysed by Marr (1971). Since then, several computational models have been proposed to describe how the hippocampus might function as an associative memory. In particular, the CA3 region of the hippocampus has attracted the largest part of the attention in these models due to the recurrent connectivity within the structure (Gardner-Medwin 1976; McNaughton and Morris 1987; McNaughton and Nadel 1990; Buckingham 1991; Treves and Rolls 1992; Bennett et al. 1994). The biological justifications for basing the autoassociative network in the CA3 region have been discussed in great detail in this prior work. Here, I briefly summarise the

biological justifications for the CA3 autoassociative network reported previously, together with recent evidence to support the existence of such a network.

Since the observation of profound amnesia following bilateral removal of the hippocampus and nearby structures in patient H.M. (Scoville and Milner 1957), it has become apparent that the hippocampal formation plays a crucial role in learning and memory. Notably, there is also little doubt that the hippocampus is necessary for the development of long term memory (Squire 1992). Furthermore, the hippocampus has been shown to be necessary to process spatial memory and consequentially, to guide navigation in rodents (O'Keefe and Nadel 1978). Evidence for spatial memory processing in the hippocampus is found also in primates (Ono et al. 1991). Although whether the function of the hippocampus is exclusive to spatial information processing is still under dispute (Eichenbaum and Otto 1992), the spatial memory idea gives the most intuitive understanding and testable memory paradigm of the hippocampus as a memory device. It is in this spirit that part of this chapter considers encoding spatial information (i.e. place) as memory events for the associative network.

However, the associative memory system analysed in the previous chapters was selected for its mathematical simplicity and not because it incorporates all of the known anatomical, pharmacological, and physiological properties of hippocampal neurons. The interest in this model depends on a consistency between the properties of the model and these observed in experimental neuroscience. Therefore, the aim of this chapter is to show the correspondence between the assumptions and observed properties of the model with the known biology, and then to make plausible predictions from the model. For this reason, most of the discussion is concentrated on the neuronal circuit of

the hippocampus rather than a mere general discussion of the functions of the hippocampus. There is a large volume of literature that reviews the general aspects of hippocampal functions (for example, Brown and Zador (1990), Squire (1992), or Jarrard (1993), to name a few) .

In order to discuss these issues quantitatively comparison is drawn between the model and the extensively studied anatomy and physiology of the rat hippocampus.

6.2 Assumptions of the model

In this section the basic assumptions of the model are described. The assumptions of the model can be subdivided into three categories; A) compositional issues, B) operational issues, and C) coding issues. Each assumption is indexed according to these categories (e.g. A1, B3, ... etc.). The biological justifications for the assumptions will be explained in the next sections. At the ends of the paragraphs, the relevant assumptions will be referenced to their indices in square (e.g. [A1]) brackets.

A. Compositional issues

These assumptions concern wiring a hardware of the model and are mostly related to the anatomy of the hippocampus.

A1 There are two classes of cells; excitatory principal cells and inhibitory interneurons. The principal cells are more numerous than inhibitory cells. Every principal cell must receive connections from the inhibitory cell to have a varying threshold effect depending on the activity of the interneuron. The connections established by the inhibitory interneuron must be diffuse so that the threshold is globally applied.

A2 The principal cells in the network are recurrently connected. Furthermore, the connection pattern among the principal cell network is sparse and random.

A3 Specific external input to the network must exist in order to supply (or rather teach) memory events to the network. There are strong external inputs to the principal cells to teach memory events to the network and to initiate recall.

B. Operational issues

These assumptions concern the function of a single cell and the controls necessary to operate the associative network. These issues are mostly related to physiology of the hippocampal cells.

B1 The synaptic interactions of principal cells within the recurrent collaterals operate in an *all or none* fashion. Equivalently, synaptic transmission can be approximated to be binary and is not graded. Principal cell synapses have two operational states, namely, the effective state and the ineffective state. Also, the recurrent connections must be sufficiently powerful to keep cells active during the cycles of a recall.

B2 The network operates in two distinct modes; learning mode and recall mode. In the learning mode, synaptic modification (i.e. from ineffective state to effective state) occurs when pre- and post-synaptic cells are both active.

B3 During the recall mode, progressive recall is achieved in discrete time steps. In order for this to happen, principal cells that are being recalled must fire synchronously to construct the cell assembly by activating each other.

B4 Each principal cell's threshold is modified by input from inhibitory interneurons. The response of the inhibitory interneurons to an input is faster than that of the principal cells so that the inhibitory interneuron can sample the instantaneous activity of the principal cells and in turn provide the appropriate inhibitory signal to the principal cell network. This also requires that unlike the principal cells, inhibitory interneurons have a graded response.

B5 Synapses from inhibitory interneurons to principal cells might be modifiable in order to increase the storage capacity.

C. Coding issues

The previously stated two issues are the "hardware" requirements of the biological autoassociative memory. The assumptions addressed addressed here are the "software" requirements in that they are concerned about what goes into the network. These issues are related to physiological and behavioural studies of the hippocampus.

C1 The number of active cells (W) in an event is fixed and relatively small in comparison with to the total number of cells (N).

C2 Each memory event is coded independently in order to avoid overlap in the representation of memory events.

6.3 Anatomical aspects

In this section, anatomy of the hippocampus, with a special attention to the CA3 region, is discussed in some detail in order to support the assumptions stated in the previous sections.

6.3.1 Principal cells

In each of the areas of the hippocampus proper, the majority (more than 90%) of the cells are excitatory cells (Misgeld and Frotscher 1986; Ishizuka et al. 1990). It is estimated that in the Sprague-Dawley rat there are 200,000 entorhinal layer 2 cortical cells projecting to the dentate gyrus, 1,000,000 dentate granule cells, 420,000 pyramidal cells in CA1, 330,000 in CA3 (and CA2), and 128,000 in the subiculum (summarised in Amaral et al. (1990)) [1]. [A1]

According to Amaral et al. (1990), each dentate granule cell contacts approximately 14 pyramidal cells in CA3, assuming that each presynaptic expansion terminates on only one pyramidal cell. If mossy fibre contacts on the CA3 pyramidal neuron are homogeneously distributed, each CA3 pyramidal cell is expected to be innervated by approximately 46 granule cells. The number of synapses innervating a neuron is usually estimated from measurements of the dendritic length and the density of spines or synapses along dendritic branches. The connectivity among the CA3 pyramidal cells is estimated to be roughly 2% and the connectivity of the projection from the CA3 pyramidal cells to the CA1 pyramidal cells via the Schaffer collateral is also roughly 2%, if the spine density of 1 spine/μm is assumed (Amaral, Ishizuka, and Claiborne 1990) [A2,A3]

Recent anatomical methods of intracellular labelling and computer reconstruction have been employed to determine the quantitative pattern of neuronal innervation in the hippocampus. Using this method, it has become clear that hippocampal pyramidal neurons have rich three-dimensional longitudinal connections (Amaral and Witter 1989; Ishizuka et al. 1990; Tamamaki and Nojyo 1991), as opposed to the previously proposed lamellar hypothesis (Andersn

[1] Note that this number is for one hemisphere.

et al. 1971). Within the CA3 region, proximally located (near dentate gyrus) the CA3 cells tend to innervate only the proximal portion of CA3, whereas mid and distal portions of CA3 project throughout the transverse extent of CA3 (Ishizuka et al. 1990). The widespread connection pattern among CA3 pyramidal cells implicates that the whole CA3 substrate can act as a single network. In fact, the idea that CA3 is the "longitudinal associational bundle" was suggested first by Lorente de Nó (1934). Expanding this view, it has been suggested that there could be a single bilateral CA3 network (Treves and Rolls 1994) since the CA3 receives recurrent connection from the contralateral CA3 with the same probability as the ipsilateral CA3 pyramidal cells. [A1,A2]

From the recurrent associative memory point of view, it is advantageous to have uniformly distributed random connections among CA3 pyramidal cells. First, by keeping the connections random, memory events can have a wide range of patterns. It is easy to think of the opposite extreme case. If one group of cells (group A) in the network receives few synaptic contacts from other cells in the network and another group of cells (group B) are heavily innervated, it is very difficult for the network to learn an memory event consisting of cells in both of the groups. For instance, in the recall process, while the cells in group B can be easily be recruited, the cells in group B can hardly be activated because of the scarcity of effective inputs. It then implies that only the cells which receive heavier projections can be used as coding cells. If all cells are equally innervated, all cells can be used as coding cells which will increase the number of possible memory events.

Second, It is easier to theoretically analyse networks with homogeneous connectivity since a statistical estimate of the next network state can be made reliably using expectation value analyses. Therefore, modulation of the net-

work activity, such as the application of a global threshold can be modelled easily using the statistical dynamics.

6.3.2 Interneurons

Since almost all models of the CA3 recurrent associative network assume that interneuron activity determines the level of the threshold for principal cells, it is important to study the anatomy and physiology of interneurons in the hippocampus. There are several types of interneurons identified in the hippocampus. These types of interneurons have recently been reviewed by Freund and Buzsáki (1996).

Interneurons can be classified 1) morphologically by studying the dendritic and axonal arborisation as well as the location of the cell bodies location of cell bodies, 2) neurochemically by studying the neurotransmitter, neuropeptides and calcium binding proteins inside the cell, or 3) physiologically by measuring the membrane potential of the cell.

Most of the hippocampal interneurons are immunoreactive for GABA which is the major inhibitory transmitter of the hippocampus. In guinea pig, there are about 33,000 inhibitory interneurons in CA3, which is about 10% of the pyramidal cell number in the region (Misgeld and Frotscher 1986). A similar number is calculated by Woodson *et al.* (1989). Inhibitory interneurons can be identified by GABA, glutamic acid decarboxylase (GAD), or parvalbumin immunostaining. Among those immunostains, parvalbumin is known to stain most of the dendritic arborisation. 20% of all GABAergic neurons are parvalbumin-immunoreactive (Kosaka et al. 1987). [A1]

As far as the morphology of hippocampal interneurons are concerned, chandelier cells and basket cells are the strong candidates for the threshold managing interneurons since 1) they are able to sample network activity level from

a large subpopulation of the pyramidal cells, 2) and in turn exert a strong inhibitory effect. Hippocampal interneurons' physiological characteristics will be described in the next section.

One might argue that the fact that only one interneuron was present in the model is an oversimplification. This simplification was made in order to to make the simulation and the mathematical analysis simpler. The existence of a multiple number of interneurons does not contradict the model, because the network of interneurons may still be sampling the local activity and returning the appropriate inhibitory input, based on the local activity level. Since the cellular activity in CA3 is considered to be distributed, sampling with a large number of interneurons increases the confidence of the measure of overall network activity. Such a suggestion has been made elsewhere (Bennett et al. 1994; Minai and Levy 1994) [A2]

6.4 Physiological aspects

6.4.1 Synaptic modification

The efficacy of hippocampal principal cell synapses have been demonstrated to change in the paradigm of long-term potentiation (LTP) (Bliss and Lomo 1973; Bliss and Gardner-Medwin 1973). LTP is a persistent increase in synaptic efficacy that can be rapidly induced, and it is induced when the presynaptic neuron fires action potentials and the postsynaptic dendrite is sufficiently depolarised (Brown, Kairiss, and Keenan 1990). LTP was originally observed in the dentate gyrus (mossy fibre projections to CA3) (Bliss and Lomo 1973; Bliss and Gardner-Medwin 1973), and later Hebbian type LTP was found in the Schaffer collateral synapses (Kelso et al. 1986; Wigstrom et al. 1986; Malinow and Miller 1986). Schaffer collateral LTP is mediated by NMDA glu-

tamate receptors (Bliss and Collingridge 1983). Injecting NMDA blocker into the rat hippocampus impairs spatial learning ability (Davis et al. 1992).

Considering the morphological, physiological, and neurochemical similarity between the CA1 pyramidal cell and the CA3 pyramidal cell, it is very likely that LTP is induced among CA3 pyramidal cells. There is evidence that LTP occurs in the recurrent collaterals of the CA3 network when there is presynaptic activity concurrently with strong postsynaptic depolarisation (Miles 1988). [B2]

6.4.2 Synaptic transmission

Graded synapses, as opposed to binary synapses, have been preferred to be more biologically realistic in many neural network models (for a review of synaptic learning models, see, for example (Brown et al. 1990)). The criticism directed towards binary synapses is largely that they are an oversimplification of the biological system. This is partly because different excitatory postsynaptic potentials have been observed in different synapses of pyramidal cells in the hippocampus. However, there is no evidence that a single synapse can have more than two levels of efficacy.

On the other hand, as described in chapter 1, theoretical studies have shown that binary synapses can exhibit near-optimal information storage efficiency (Willshaw et al. 1969; Palm 1988; Nadal and Toulouse 1990). If binary synapses have the greatest storage efficiency, then why would the process of evolution produce non-binary synapses.

Another basis of the claim that real-valued synapses are more biologically plausible is that the synaptic weight is not concerned with the levels of conductance, but with the probability of releasing vesicles that contain neurotransmitter (quanta). According to this view, highly potentiated synapses have

higher probability of releasing the neurotransmitter when activated and as the synapse becomes less potentiated, the probability of releasing the neurotransmitter becomes lower (Lisman 1994).

Stevens and Wang (1995) have reported that synaptic transmission is rather probabilistic when a single spike is produced in a presynaptic hippocampal pyramidal cell. However, when two consecutive spikes are presented to the pyramidal cell, the synaptic transmission is ensured in the second spike (the probability being 0.89). As hippocampal cells fire in bursts (i.e. complex spike, consisting of 3-10 spikes) during the theta activity (O'Keefe and Nadel 1978), the excitatory postsynaptic potential (EPSP) is almost certainly transfered, which argues against models based on the probability of synaptic conduction (e.g. Sejnowski (1977)).

Recently, Markram and Tsodyks (1996) have demonstrated in neocortical pyramidal cells that when burst activity is present in the presynaptic cell, temporal weight renormalisation occurs at the synapse so that the synaptic conduction becomes less effective for subsequent spikes. This means that when cells signal information in bursts, the first few spikes conducts most of the information and the rest of the spikes are just to ensure the synaptic transmission.

With these experimental results, it is not justified to discard the idea that binary synapses are a biologically plausible mechanism of synaptic function.

6.4.3 Synchronisation

Aside from the input from the entorhinal cortex, there is another important input to the hippocampus that originates from the medial septum and the diagonal band of Broca (Raisman 1966; Meibach and Siegel 1977). Although the projections terminate throughout the hippocampus, they preferentially

terminate in CA3 and the dentate gyrus. The connection between CA3 and the medial septum is reciprocal in that CA3 cells also project to the medial septum (Raisman 1966). The input includes cholinergic and GABAergic fibers which are considered to play an important role in modulating the hippocampal function, especially the CA3 area (Stewart and Fox 1990). Interestingly, the GABAergic fibres terminate preferentially on inhibitory interneurons (Freund and Antal 1988).

The septal input is critically important to the hippocampal function for at least three reasons 1) it mediates the theta rhythm as the pacemaker (for a review,Stewart and Fox (1990)), without which the recall of spatial memory in the hippocampus cannot function (Mizumori et al. 1989) 2) it secretes GABA to the hippocampus during theta to probably disinhibit the interneurons, as the GABAergic projection preferentially terminates onto inhibitory interneurons (Freund and Antal 1988), and 3) cholinergic projection from the septum to the hippocampus are likely to be important as acetylcholine (ACh) has been suggested to play an important role in learning associations in recurrent circuits (Hasselmo 1993; Hasselmo 1995).

As far as the hippocampal EEG activity is concerned, the hippocampus operates in two distinct modes: the theta mode in which sinusoidal theta rhythm (6-12 Hz) is present in the EEG, and large irregular activity (LIA) in which sharp wave of a large amplitude is observed in random intervals. In awake rats, theta activity (Vanderwolf 1969) is present when the animal is engaged in locomotion (such as walking, running) and LIA occurs during immobility (O'Keefe and Nadel 1978). Theta activity also occurs during REM sleep.

Recently, Cobb *et al.* (Cobb et al. 1995) have reported that firing of

GABAergic interneurons can cause synchronised theta period oscillations in pyramidal cells. Further, they demonstrated that a single interneuron can drive this modulation to about 1,000 pyramidal cells. This is strong evidence for synchronised activity in the hippocampus. Bragin *et al.* (Bragin et al. 1995) reported that gamma activity (40-100Hz) is present within the theta activity. Gamma activity is present only during theta and disappears during LIA. The cause of the oscillation is unknown, however, they hypothesise that the gamma oscillation is generated by entorhinal or hilar input to the inhibitory interneuron network in the hippocampus. Recently, Whittington *et al.* (Whittington et al. 1995) demonstrated that 40-Hz membrane oscillation in pyramidal cells is induced by inhibitory neurons that have $GABA_A$ synapses, when metabotropic glutamate receptors are activated. These findings lead to a thought that while pyramidal cells code memory events, it is the interneurons that control the behaviour of the network. To be more precise, since a GABAergic interneuron innervates approximately 3000 pyramidal cells (estimated from Schwartzkroin and Kundel (1985) and Boss et al. (1985)), it is likely that the interneurons, or a network of interneurons mediate the synchronisation of pyramidal cell firing.

Additional evidence for synchronisation in the hippocampal network comes from a behavioural study. O'Keefe and Recce (1993) found a temporal relation ship between the theta oscillation in the hippocampus and the activity of place cells. They reported that frequency of the occurrences of the complex spikes in place cells (pyramidal cells) is slightly shorter than that of the theta wave. This means that the timing of the place cell firing and the corresponding phase of the theta wave consistently shifts backward as the animal runs through the place field. Since the cells that have the same place field have similar firing

properties, those cells are regarded to be synchronised with the phase of the theta oscillation. The temporal firing pattern provides more information for the spatial location of the animal than can be found with the firing rate alone. This result has recently been reconfirmed by Skaggs et al. (1996).

6.4.4 Thresholding

Excitatory synaptic transmission from CA3 pyramidal cells occurs in 10 to 15 ms, while synaptic transmission from CA3 pyramidal cells to inhibitory interneurons occur in a much shorter time scale (2-3 ms). The inhibitory interneuron can in turn give quick (3-5ms) response back to pyramidal cells. (Miles 1990a; Miles and Wong 1987). This provides a physiological basis of threshold control by inhibitory interneurons. [B4]

6.4.5 Synaptic plasticity in interneurons

It has long been assumed and believed that the synaptic efficacy of inhibitory synapses does not change. However, recent studies in the visual cortex suggests that long-term modification of synaptic efficacy occurs in cortical inhibitory synapses (Marty and Llano 1995). Inhibitory synaptic modification was first demonstrated by Korn et al. (1992) in glycinergic synapses projecting onto fish Mauthner cells. LTP in a GABAergic cell has also recently been observed in the rat visual cortex. Repetitive stimulation of an inhibitory pathway onto layer V cells led to a prolonged potentiation (> 1) of the IPSPs and weaker stimuli led to a potentiation that is transient within 30 minuites (Komatsu and Iwakiri 1993). The synaptic modification was reported to be stronger in developing animals than in adults (Komatsu 1994). There is partial evidence that GABAergic cells in the hippocampus undergo a LTP like phenomenon (Otis et al. 1994).

Miles (1990b) reported that the average amplitude of CA3 inhibitory post-synaptic potentials (IPSP) initiated by single action potentials at different synapses varied over a range of ninefold. The variation is close to the variation of the number of terminals made by a basket cell to a pyramidal cells (3 to 35 terminals). However, the same study also reported that IPSPs elicited by the same inhibitory cell in several pyramidal cells have similar amplitude. This might suggest a possibility of synaptic modification in inhibitory interneurons, although the paradigm is not what is suggested in the previous chapter.

6.4.6 Activity level

One to three hundred synchronously active excitatory synapses seemed to be necessary for spike generation in CA1 pyramidal cells (Sayer et al. 1989; Andersen 1990). Assuming that CA3 pyramidal cells exhibits similar physiology[2] and a uniform 5% connectivity occurs in the CA3 network, the number indicates that 2,000 to 6,000 cells in average are necessary to recruit a correct cell. To ensure 95% ($z \simeq 1.65$) of the cells receive 300 effective inputs:

$$\frac{R}{N}w - 1.65\sqrt{w\frac{R}{N}\left(1-\frac{R}{N}\right)} = 300, \text{where } \frac{R}{N} = 0.05$$

must be satisfied. Solving this equation, w is found to be roughly 6200. This gives a coding ratio (W/N) of 6200/330000 $\simeq$ 2%. So, the minimum stable cell assembly is around this figure. Obviously the effect of inhibitory input is not considered in this figure. In reality, the pyramidal cells in CA3 are known to receive powerful synaptic inhibition. If the inhibitory effect is considered, the coding ratio will be higher than 2%. [C1]

[2]the number of excitatory synapses required for spike generation may be slightly higher for CA3 pyramidal cells as the cell body is larger.

6.5 Prediction from the model

Having discussed the biological plausibility of the model in previous sections, there are several predictions that can be made from the model.

First, the fact that the linear thresholding strategy achieves progressive recall in a highly efficient manner predicts that such a thresholding strategy may be realised in biology. That is, the inhibitory effect from the hippocampal interneurons to the pyramidal cells is a linear function of the total number of active pyramidal cells. The finding that the coefficients of the linear thresholding equation does not depend much on the memory loading suggests that the physiological characteristics of the interneurons could be preconfigured, perhaps at a genetic level.

Second, as figure 4.1 shows, as the network are loaded more intensively, the time required to complete recall increases. The model predicts that this applies to the biological recurrent associative network, too. If each recall step takes a theta cycle (say 10Hz), it will take 1-2 seconds to achieve a recall for a fully loaded network. Also, it is possible that the hippocampal reccurrent network keeps a constant level of memory loading and therefore the time required to complete a recall is preserved. Important and frequently used memory events may be consolidated to the neocortex for long-term memory storage, and rarely used memory events may be forgotten in order to make a space for new memory to be stored.

Third, the results in chapter 5 encourages the idea that inhibitory interneurons also participate in memory encoding. The interneuron learning model also predicts that synaptic plasticity of interneuron occurs in a graded fashion rather as opposed to the on-or-off synapses, which are assumed in pyramidal cells. As described in previous sections, synaptic plasticity of interneurons

have been reported recently and it is expected that more investigation will be made in this area in the near future.

6.6 Implications from the model

Behavioural experiments have demonstrated that hippocampal cells take part in spatial memory encoding (O'Keefe and Nadel 1978; O'Keefe and Speakman 1987; Muller et al. 1987; Muller and Kubie 1987). A place cell fires when the animal passes a specific place in an environment (O'Keefe and Dostrovsky 1971). This can be regarded as evidence hippocampal cells encode spatial memory of *place*. If the neuronal circuit in CA3 operates as autoassociative memory, it is reasonable to think that an event (cell assembly), rather than a single cell, encodes a spatial location.

A typical place cell has a circular receptive field in the environment (*place field*) of roughly constant size with the peak response point in the center of the field. However, occasionally cells fire in more than two places in the same environment. Such a cell is referred to as the *multi-field place cell.* There are also multi-environment place cells that operate as place cells in more than one environment.

Multi-field place cells have been argued to be observed because of bad separation of spikes, i.e. a single electrode may be recording from several nearby cells. Although Multi-electrode techniques improved spike separation multi-field place cells are still observed although the number of occurrence has decreased.

I would argue the possibility of overlap among memory events as a cause of multi-field place cells. That is, when a cell is involved in learning two memory events that encodes different places (i.e. overlap), the cell fires in both of the

places. In this way, existence of multi-field place cells and multi-environment place cells are justified. Furthermore, this hypothesis predicts that as the rat learns more spatial experiences, occurrence of multi-field place cells or multi-environment place cell will increase. Naive calculation shows the following: If, coding ratio of 2% and the network is reasonably loaded ($\rho = 0.3$), the average number of events a cell experience $\langle m \rangle$(i.e. the expectation value of unit usage) is

$$\begin{aligned} \rho &= 1 - \left[1 - \left(\frac{W}{N}\right)^2\right]^M \\ M &= \frac{log(1-\rho)}{log\left[1 - \left(\frac{W}{N}\right)^2\right]} \\ \overline{m} &= M\frac{W}{N} \simeq 17.8 \end{aligned}$$

The above formulae exhibit an interesting relationship between the coding ratio (W/N) and the average unit usage ($\overline{m}$), that for a fixed ρ, average unit usage decreases monotonically with coding ratio. So, for a fixed ρ, a cell becomes less specific to encoding events as the coding becomes sparse.

The coding ratio problem gives an interesting insight to place cell recording experiments because it might help to predict the probability of finding place cells per tetrode (multi-channel electrode) penetration. Since place cell coding is known to be not topographic (i.e. neighbouring cells do not represent neighbouring places in the environment), memory events of the places are considered to be independently coded. Therefore, when the spatial distribution of place cells (i.e. how many cells are needed to represent an environment of, say size 1m×1m) are known and the coding ratio can be estimated, the probability of finding place cell can also be predicted. Conversely, if the probability of finding a place cell per electrode penetration can statistically be measured, the coding ratio may be able to be estimated. In order to do this calculation,

one need to know how many cells can be recorded per tetrode penetration. However, research for this subject has not been systematically done to date.

6.7 Future perspectives

Having studied thresholding strategies for a recurrently and sparsely connected binary associative memory network, there are several themes yet to be addressed.

First, the capacity of the network can theoretically be found only by calculating the Gibson and Robinson progressive recall equation with exhaustive search for optimal threshold. As in chapter 4, this threshold sequence turned out to be linear to the momental activity of the network. However, to find the slope and intercept of the linear function, the exhaustive search for threshold sequence must be pursued. In order to be able to easily estimate the capacity of a recurrently connected binary associative memory network, the method of finding the slope and intercept of the linear threshold function has to be generalised. Another approach to find the capacity of a network with progressive recall is to measure (or theoretically calculate) how many correct cells (or how much information) can be retrieved from a seed pattern by a single cell. If no information gain is possible from a seed cell using any threshold level, it is very likely that the network is loaded beyond its capacity. Perhaps this method is easier to generalise for a larger class of recurrent network models.

Second, interneuron learning strategy is not quite perfectly formalised, and the learning rule proposed in chapter 5 is rather empirically fitted. An immediate goal of the interneuron learning study will be to formulate the optimal learning rule for the inhibitory synapses by formalising the interneuron network dynamics.

Third, having learnt that the unit usages of active cells matters much to a successful recall, it will be interesting to think of a new quality metric based upon the unit usages of the active cells. Using this measure, it is possible to evaluate the quality of seed cell and the likeliness of a successful recall using the seed.

Finally, with these issues in mind, we are ready to add extra levels of complexity to the recurrent associative memory network in order to incorporate more biological plausibility.

Appendix A

Nomenclature

Operations and functions

$P(X)$	the probability of occurrence of X
$\langle x \rangle$	the expected value of a variable x
$\mathcal{B}(p,n,m)$	the binomial probability for m occurrences of the event out of n trials where each event has the probability of p to happen, i.e. $\frac{n!}{m!(n-m)!}p^m(1-p)^{n-m}$.
$\mathcal{N}(\mu,\sigma^2)$	Normal distribution with the mean $= \mu$ and the variance $= \sigma^2$
$\Phi(z)$	The standard normal distribution function: $\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} exp(-u^2/2)du$
$\phi(z)$	The standard normal density function: $\phi(z) = \frac{1}{\sqrt{2\pi}}exp(-u^2/2)$

Constants and variables

N	the total number of principal cells in the network
R	the number of synaptic connections made by each cell
M	the number of events experienced by the network
W	the number of cells active during an event

$c(t)$ the total number of correct cells at time t

$s(t)$ the total number of spurious cells at time t

$w(t)$ the total number of active cells at time t $(= c_t + s_t)$

$T(t)$ the threshold value of the principal cells at time t

α the slope for the linear thresholding strategy.

γ the offset for the linear thresholding strategy.

ρ the expected proportion of the modifiable synapses which have been rendered effective

corr the index to a correct cell.

spur the index to a spurious cell.

Vectors and matrices:

Both vectors and matrices are printed in bold. In general, capital letter symbols like $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ denote matrices and small letter symbols like $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ denote vectors.

$\mathbf{C}$ the connectivity matrix

$\mathbf{d}(t)$ the net inputs to the principal cells at time t

$\mathbf{W}$ the matrix of learned connections

$\mathbf{x}(t)$ the activity of the network at time t

$\mathbf{Z}$ the collection of learned events

Appendix B

Theoretical interpretations

The purpose of this appendix is to mathematically describe and analyse the recurrent autoassociative network described in chapter 1. The analysis is mainly based upon works by Gardner-Medwin (1976), Buckingham & Willshaw (1992, 1993), and Gibson & Robinson (1992) who recently formalised dynamics of progressive recall.

B.1 Basic Notations

As described in chapter 1 and in appendix A, the following terms are defined. The associative network contains N principal cells, each of which sends collaterals to R other principal cells. The network learns M events consisting of W active components. $\mathbf{Z}_{N\times M} = [z_{ij}]$ is a collection of the learned events and $\mathbf{z}_i$ represents the column vector of ith learned event. Further, we define two $N \times N$ matrices; the connectivity matrix $\mathbf{C} = [c_{ij}]$ and the matrix of learned connections (the weight matrix), $\mathbf{W} = [w_{ij}]$.

$c_{ij} = 1$ when there is a connection (regardless the effectiveness of the connection) from cell j to cell i, otherwise, $\mathbf{C}_{ij} = 0$. $\mathbf{C}_{ii} = 0$ (i.e. a cell does not project onto itself) and $\sum_{i=1}^{N} c_{ij} = R$ for all j by definition.

$\mathbf{W}$ is similar to $\mathbf{C}$ in that wherever c_{ij} is 0, w_{ij} is also 0. w_{ij} becomes 1 where there is a connection (i.e. $c_{ij} = 1$) **and** the synapse is modified to be

effective as a result of the learning process. Formally,

$$w_{ij} = c_{ij} \bigvee_{k=1}^{M} z_{ki} z_{kj}$$

Activity of the network at time t is described by a N-dimensional binary vector $\mathbf{x}(t) = [x_i]$ with $x_j(t) = 1$ meaning cell j is active at time t and $x_i(t) = 0$, otherwise (the activity of a principal cell is binary). When $\mathbf{z}_n$ is recalled, $w(t)$, the total number of active cells at time t is defined as:

$$w(t) = \sum_{i=1}^{N} x_i(t)$$

Similarly, $c(t)$, the total number of active correct cells is:

$$c(t) = \sum_{i=1}^{N} z_{ni} x_i(t) = \mathbf{z} \cdot \mathbf{x}(t)$$

$s(t)$, the total number of active spurious cells is therefore $s(t) = \sum_{i=0}^{N-1} \widetilde{z_{ni}} x_i(t)$ or rather $s(t) = w(t) - c(t)$. ($\tilde{a}$ represents the complement of a binary expression a, i.e. $\tilde{a} = 1, a = 0$ and $\tilde{a} = 0, a = 1$.) Note that $w(t)$ and $c(t)$ shall not be confused with w_{ij} and c_{ij}.

The net input to cell (sometimes referred to as the dendritic input in other literatures) at time t is denoted to a N-dimensional vector to $\mathbf{d}(t)$, i.e. $d_i(t)$ is the amount of the net input to cell i. To illustrate the dynamics of the associative memory,

$$\mathbf{d}(t) = \mathbf{W}\mathbf{x}(t)$$

The threshold $T(t)$ is globally applied to the network at time t.

$$x_i(t+1) = \Theta[d_i(t) - T(t+1)]$$

where $\Theta[\cdot]$ is the Heaviside function ($\Theta[x] = 1, x > 0$ and $\Theta[x] = 0$, otherwise).

Finally, *corr* and *spur* are are used as the index to a correct cell and the index to a spurious cell, respectively. For example, $P(d_{corr}(t) = 3)$ indicates the probability that the net input to a correct cell is 3.

Mathematical symbols and notations used in this thesis are summarised in appendix A.

B.2 Foundations

Original analysis by Gardner-Medwin (Gardner-Medwin 1976) and his thresholding strategy is summarised in this section. He estimated the expected number of active cells at the next stage of recall, given the level of current activity of the network and a threshold. Based on the analysis, he defined a thresholding strategy that leads to a successful recall.

When a single event is learned, the probability that any one synapse in the network satisfies the modification conditions is:

$$P(w_{ij} = 1 | c_{ij} = 1, M = 1) \simeq \frac{W^2}{N^2} = \frac{W(W-1)}{N(N-1)}$$

It follows that when the network has learned M randomly generated events, the (expected) probability that a synapse becomes effective (ρ) is $1 - (1 - W^2/N^2)^M$. The quantity can be approximated as follows by taking the first term of the Taylor's expansion.

$$\begin{aligned} \langle \rho \rangle = 1 - (1 - W^2/N^2)^M &\approx 1 - exp(-MW^2/N^2), \quad W/N \ll 1 \\ &\approx MW^2/N^2, \quad MW^2/N^2 \ll 1 \end{aligned}$$

When c correct cells are active (and there is no active spurious cell), the probability that a correct cell receives exactly r effective inputs is just a matter

of connection probability. This is a binomial probability[1], i.e.:

$$P(d_{corr} = r) = \binom{c}{r} \left(\frac{R}{N}\right)^r \left(1 - \frac{R}{N}\right)^{c-r} = \mathcal{B}\left(\frac{R}{N}, c, r\right)$$

Similarly, the probability that a spurious cell receives exactly r effective inputs was proposed to be:

$$P(d_{spur} = r) = \mathcal{B}\left(\rho\frac{R}{N}, c, r\right)$$

Note that $P(d_{corr} = r)$ and $P(d_{spur} = r)$ are referred to as *dendritic sum distribution* for correct cells and dendritic sum distribution of spurious cells, respectively. These distributions are used to compute the expected number of active cells after a threshold is applied to the network.

When threshold T is applied, cells with net input greater than T becomes active. The probability that a correct cell receives greater than T effective inputs is:

$$P(d_{corr} > T) = 1 - \sum_{r=0}^{T} P(d_{corr} = r) = 1 - \sum_{r=0}^{T} \mathcal{B}\left(\frac{R}{N}, c, r\right)$$

The probability that a spurious cell receives greater than T effective inputs is

$$P(d_{spur} > T) = 1 - \sum_{r=0}^{T} P(d_{spur} = r) = 1 - \sum_{r=0}^{T} \mathcal{B}\left(\rho\frac{R}{N}, c, r\right)$$

Adapting the above formalism, when only $c(t)$ correct cells are active at time t and the threshold $T(t+1)$ is applied to the network, the expected numbers of active correct cells and spurious cells at the next time step are

$$\langle c(t+1)\rangle = WP(d_{corr} > T(t+1))$$

$$\langle s(t+1)\rangle = (N - W)P(d_{spur} > T(t+1))$$

[1]In Gardner-Medwin(1976), the probability is approximated to the Poisson distribution for computational reasons. R/N represents the connectivity of the network. To be preciese, the connectivity is $R/(N-1)$ as there is no self-feedback connection. For large N the connectivity can be approximated to R/N. This approximation is assumed throughout the appendix.

, respectively, where $T(t+1)$ indicates the global threshold applied to the network.

The thresholding strategy suggested in Gardner-Medwin (Gardner-Medwin 1976) was to choose the lowest threshold that will inhibit the spurious cell activity. i.e. $T(t+1)$ is such that $\langle s(t+1)\rangle = 0$ and $(N-W)P(d_{spur}(t) > T(t+1)-1) > 0$. More about the thresholding strategy is also discussed in chapter 6.

B.3 On distribution of modified synapses

While working on two-layered hetero-associative networks, Buckingham noticed that the dendritic sum distribution (occurrence v.s. net input) for spurious cells did not match the previously believed binomial distribution of ρ and the connectivity. Rather, he noticed that the dendritic sum distribution for spurious cells depends not only on the connectivity of the network but also the unit usage (the number of events the cell is involved during the learning stage) of the cell. This is because there is a correlation between w_{ij} and w_{ik}. (In the previous section, the correlation is ignored and the probability of modified synapse is uniformly treated as $\frac{R}{N}\rho$.)

In fact, the notion that the w_{ij} and w_{ik} are not independently distributed has earlier been mentioned by Nadal and Toulouse (Nadal and Toulouse 1990). While Nadal and Toulouse estimated the probability distribution of the modified synapses using a normal distribution, Buckingham calculated more precisely defined distribution, i.e. using sum of binomial distributions. The following is adapted from Buckingham's (Buckingham 1991) work to fit the model studied in this thesis.

It is important to know that cells experience different number of events

during learning. *Unit usage* (m) is the number of times the principal cell participates in the learned events. When a principal cell has been involved in learning m events (i.e. there are m events that have 1 for the principal cell), the probability that a synapse contacting to the principal cell becomes effective is:

$$\rho(m) = 1 - \left(1 - \frac{W}{N}\right)^m$$

The probability that a principal cell participates in one event is W/N, so the probability that a principal cell experiences m events out of M events (P_e) is binomially distributed with W/N:

$$P_e = \mathcal{B}\left(\frac{W}{N}, M, m\right)$$

The new dendritic input distribution can be computed for $P(\mathbf{D}_{corr} = r)$ by combining the above two distributions. Given c active correct cells, the probability that any one spurious cell receives r ($0 < r < c$) effective inputs is therefore:

$$P(\mathbf{D}_{spur} = r) = \sum_{m=1}^{M} P_e(m)\mathcal{B}\left(\rho(m)\frac{R}{N}, c, r\right)$$

B.4 Progressive recall equations

Gibson and Robinson(Gibson and Robinson 1992) calculated the statistical dynamics of activity of the network for progressive recall. They are also aware of the fact that w_{ij} and w_{ik} are not independently distributed. Their method of calculating the statistical dynamics, /em progressive recall equation is reviewed as below (slightly modified from the original derivation to fit the conventions defined sor far).

They approximated the dendritic sum distribution using the normal distributions. Given w active cells which consists of c correct cells and s spurious

cells:

$$P(d_{corr}) = \mathcal{N}(\mu_{corr}, \sigma^2_{corr})$$
$$P(d_{spur}) = \mathcal{N}(\mu_{spur}, \sigma^2_{spur})$$

where $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2.

μ_{corr} is the mean number of dendritic input received by a correct cell. Likewise, μ_{spur} is the mean number of dendritic input received by a spurious cell. They are:

$$\mu_{corr} = c\frac{R}{N} + s\rho\frac{R}{N}, \quad \mu_{spur} = w\rho\frac{R}{N}$$

where $w = c + s$

$c\sigma^2_{corr}$ is the variance of the dendritic input distribution to a correct cell and $c\sigma^2_{spur}$ is the variance of the dendritic input to a spurious cell. This can be calculated by treating each of the incoming synaptic input as a random variable so that the dendritic input is the sum of the random variables.

For a correct cell, the probability distribution for receiving an effective input from another active correct cell is equivallent to a Bernoulli trial with mean $\frac{R}{N}$ and variance $\frac{R}{N}\left(1 - \frac{R}{N}\right)$. Similarly, the probability distribution for receiving an effective input from another active spurious cell is a Bernoulli trial with mean $\rho\frac{R}{N}$ and variance $\rho\frac{R}{N}\left(1 - \rho\frac{R}{N}\right)$. As discussed in the previous section, the covariance for the random variables w_{ij} and w_{ik} must also be taken into consideration. The covariance is denoted as $cov(w_{ij}, w_{ik})$ for convenience. $c\sigma^2_{corr}$ is calculated as:

$$\sigma^2_{corr} = c\frac{R}{N}\left(1 - \frac{R}{N}\right) + s\rho\frac{R}{N}\left(1 - \rho\frac{R}{N}\right) + s(s-1)cov(w_{ij}, w_{ik})$$

Similarly, σ_{spur} is the variance of the sum of the random variable $w_{i*}*$, where * indicates the indecies to the active cell (there are w of them).

$$\sigma^2_{spur} = w\rho\frac{R}{N}\left(1 - \rho\frac{R}{N}\right) + w(w-1)cov(w_{ij}, w_{ik})$$

$cov(w_{ij}, w_{ik}$, the covariance between w_{ij} and w_{ik} must be calculated. Elaborative work by Gibson and Robinson (Gibson and Robinson 1992) demonstrated that the covariance can be computed as follows.

$$= \left(\frac{R}{N}\right)^2 \left[1 - 2\left(\frac{W}{N}\right)^2 + \left(\frac{W}{N}\right)^3\right]^M - \left[1 - \left(\frac{W}{N}\right)^2\right]^{2M}$$

Consequentially, given threshold value $T(t+1)$, the expected activities of correct cells and spurious cell after the threshold is applied are:

$$\begin{aligned} \langle c(t+1)\rangle &= \Phi\left(\frac{\mu_c - T(t+1)}{\sigma_c}\right) \\ \langle s(t+1)\rangle &= \Phi\left(\frac{\mu_s - T(t+1)}{\sigma_s}\right) \end{aligned}$$

where $\Phi(\cdot)$ is the normal distribution function (for the definition, see appendix A).

The above is so called the level 1 equations. (The level 0 equation is the above without the consideration of the covariance among the modified synaptic connections — this is essentially the same as Gardner-Medwin's analysis (Gardner-Medwin 1976).) The level 1 analysis is equivalent to approximating Buckingham's dendritic sum distribution to a normal distribution. For simple recall (single step recall) the formalism applies as c correct cells and s spurious cells are arbitrarily chosen. In progressive recall, the level 1 equation predicts the network behaviour when the network loading is low. (i.e. small W and M). For high network loadings, it fails to predict the behaviour of the network.

As Gibson and Robinson (Gibson and Robinson 1992) inspected, this is because there is (yet again) a correlation between pre-threshold network activity ($\mathbf{X}(t)$) and post-threshold network activity ($\mathbf{X}(t+1)$). They have worked out the level 2 equation which incorporates the correlation between $\mathbf{X}(t)$ and $\mathbf{X}(t+1)$. It turns out that this can be written with 4 coupled non-linear difference equations.

If we define $x(t)$ as the expected probability of a correct cell become active at time t and $y(t)$ as the expected probability of a spurious cell become active at time t (i.e. $x(t) = \langle c(t) \rangle / W$ and $x(t) = \langle s(t) \rangle / (N - W)$), the level 2 equations are summarised as follows:

$$x(t+1) = \Phi\left(\frac{\mu_{corr}(t) - T(t+1)}{\sigma_{corr}(t)}\right), \qquad y(t+1) = \Phi\left(\frac{\mu_{spur}(t) - T(t+1)}{\sigma_{spur}(t)}\right)$$

$$x'(t+1) = \Phi\left(\frac{\mu'_{corr}(t) - T(t+1)}{\sigma'_{corr}(t)}\right), \qquad y'(t+1) = \Phi\left(\frac{\mu'_{spur}(t) - T(t+1)}{\sigma'_{spur}(t)}\right)$$

where $x(t)'$, $y(t)'$, and associated variables with ' (dash) are subsidiary variables that need be computed to find x_{t+1} and y_{t+1}. $x'(t)$ and $y'(t)$ can explicitly be written as:

$$x'(t) = \frac{\langle c(t) | w_{corr,spur} = 1 \rangle}{W}, \quad y'(t) = \frac{\langle s(t) | w_{corr,spur} = 1 \rangle}{N - W}$$

Now, $\mu_{corr}(t)$,$\mu_{spur}(t)$,$\sigma_{corr}(t)$, and $\sigma_{spur}(t)$ have to be defined. Gibson and Robinson calculates that they are:

$$\mu_{corr}(t) = \frac{WR}{N^2} x(t) + \left(1 - \frac{W}{N}\right) \frac{R}{N} \rho y'(t)$$

$$\mu_{spur}(t) = \frac{WR}{N^2} x'(t) + \left(1 - \frac{W}{N}\right) \frac{R}{N} \rho y'(t)$$

$\mu'_{corr}(t)$ and $mu'_{spur}(t)$ can be calculated by replacing ρ by ρ' where ρ' is defined below.

$$[n\sigma_1(t)]^2 = nac(1-c)x_t + n(1-a)c\rho y'_t \left(1 - c\rho \frac{y'_t}{y_t}\right) + n^2(1-a)^2 c^2 \xi (y'_t)^2$$

$$\begin{aligned}[n\sigma_n(t)]^2 \;=\; & nac\rho x'_t \left(1 - c\rho \frac{x'_t}{x_t}\right) + n(1-a)c\rho y'_t \left(1 - c\rho \frac{y'_t}{y_t}\right) + \\ & n^2 a^2 c^2 \xi (x'_t)^2 + 2n^2 a(a-a)c^2 \xi x'_t y'_t + n^2 + (1-a)^2 c^2 \xi (y'_t)^2\end{aligned}$$

Let us define a function β_k which operates on a and M as:

$$\beta_k = [1 - a(1 - (1-a)^k)]^M$$

Then, ρ' and ξ' can compactly expressed as:

$$\rho' = \frac{1 - 2\beta_1 + \beta_2}{1 - \beta_1}$$

$$\xi' = \frac{1 - 3\beta_1 + 3\beta_2 - \beta_3}{1 - \beta_1} - \rho'^2$$

Bibliography

Amaral, D. G., N. Ishizuka, and B. Claiborne (1990). Neurons, numbers, and the hippocampal network. *Progress in Brain Research 83*, 1–11.

Amaral, D. G. and M. P. Witter (1989). Three-dimensional organisation of the hippocampal formation: a review of anatomical data. *Neuroscience 31*, 571–591.

Amari, S. (1989). Characteristics of sparsely encoded associative memory. *Neural Networks 2*, 451–457.

Amari, S. and K. Maginu (1988). Statistical neurodynamics of assoative memory. *Neural Networks 1*, 63–77.

Amit, D. J. (1989). *Modelling brain function: the world of attractor neural networks*. UK: Cambridge University Press.

Amit, D. J., H. Gutfreund, and H. Sompolinsky (1985). Storing infinite number of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett. 55*, 1530.

Andersn, P., V. P. Bliss, and K. K. Skrede (1971). Lamellar organization of hippocampal excitatory pathways. *Expl. Brain Res. 13*, 222–238.

Anersen, P. (1990). Synaptic integration in hippocampal CA1 pyramids. *Progress in Brain Research 83*, 215–222.

Bennett, M. R., W. G. Gibson, and J. Robinson (1994). Dynamics of

the CA3 pyramidal neuron autoassociative memory network in the hippocampus. *Phil. Trans. R. Soc. Lond. B 343*, 167–187.

Bliss, T. V. P. and G. L. Collingridge (1983). A synaptic model of memory: long-term potentiation in the hippocampus. *Nature (Lond) 361*, 31–39.

Bliss, T. V. P. and A. R. Gardner-Medwin (1973). Long-lasting potentiation of synaptic transmission in the dentate area dentate area of the anesthetized rabbit following simulation of the perforant path. *J. Physiol. (Lond.) 232*, 357–374.

Bliss, T. V. P. and T. Lomo (1973). Long-lasting potentiation of synaptic transmission in the dentate area dentate area of the anesthetized rabbit following simulation of the perforant path. *J. Physiol. (Lond.) 232*, 331–356.

Boss, B. D., G. M. Peterson, and W. M. Cowan (1985). On the numbers of neurons in the dentate gyrus of the rat. *Brain Research 388*, 144–150.

Bragin, A., G. Jandó, Z. Nádasdy, J. K. Hetke, K. Wise, and G. Buzsáki (1995). Gamma (40-100 hz) oscillation in the hippocampus of the behaving rat. *J. Neurophysiol. 15*, 47–60.

Brown, T. H., E. W. Kairiss, and C. L. Keenan (1990). Hebbian synapses: biophysical mechanisms and algorithms. *Annu. Rev. Neurosci. 13*, 475–511.

Brown, T. H. and A. M. Zador (1990). Hippocampus. In G. M. Shepherd (Ed.), *The Synaptic Organization of the Brain*, pp. 346–388. Oxford University Press. 3rd edition.

Buckingham, J. T. (1991). *Delicate nets, faint recollections: a study of partially connected associative network memories.* Ph. D. thesis, University

of Edinburgh.

Buckingham, J. T. and D. J. Willshaw (1992). Performance characteristics of the associative net. *Network: Computation in Neural Systems 3*, 407–414.

Buckingham, J. T. and D. J. Willshaw (1993). On setting unit thresholds in an incompletely connected associative net. *Network: Computation in Neural Systems 4*, 441–459.

Cobb, S. R., E. H. Buhl, K. Halasy, O. Paulsen, and P. Somogyi (1995). Synchronization of neuronal activity in hippocampus by individual GABAergic interneurons. *Nature (Lond) 378*, 75–78.

Cohen, M. A. and S. Grossberg (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics SMC-13*, 815–826.

Crick, F. and G. Mitchison (1983). The function of dream sleep. *Nature (Lond) 304*, 111–114.

Davis, S., S. P. Butcher, and R. G. M. Morris (1992). The NMDA receptor antagonist D-2-amino-5-phosphonopentanoate (D-AP5) impairs spatial-learning and LTP invivo at intracerebral concentrations comparable to those that block LTP invitro. *J. Neurosci. 12*, 21–34.

Dayan, P. and D. J. Willshaw (1991). Optimising synaptic learning rules in linear associative memories. *Biological Cybernetics 65*, 253–265.

Eichenbaum, H. and T. Otto (1992). The hippocampus — what does it do? *Behav. Neural. Biol. 57*, 2–36.

Freund, T. F. and M. Antal (1988). GABA-containing neurons in the sptum

control inhibitory interneurons in the hippocampus. *Nature (Lond) 336*, 170–173.

Freund, T. F. and G. Buzsáki (1996). Interneurons of the hippocampus. *Hippocampus 6*, 347–470.

Gardner, E. (1987). Maximum storage capacity in neural networks. *Europhysics Letters 4*, 481–485.

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

Gardner-Medwin, A. R. (1989). Doubly modifiable synapses: a model of short and long term auto-associative memory. *Proc. R. Soc. Lond. B 238*, 137–154.

Gardner-Medwin, A. R. (1991). Possible strategies for using sleep to improve episodic memory in the face of overlap. In J. G. Taylor and C. L. T. Mannion (Eds.), *Theory and applications of neural networks*, pp. 129–138. London,UK: Springer-Verlag.

Gardner-Medwin, A. R. and S. Kaul (1995). Possible mechanisms for reducing memory confusion during sleep. *Behavioural Brain Research 69*, 167–175.

Gibson, W. G. and J. Robinson (1992). Statistical analysis of the dynamics of a sparse associative memory. *Neural Networks 5*, 645–661.

Graham, B. and D. Willshaw (1995). Improving recall from an associative memory. *Biological Cybernetics 72*, 337–346.

Haberly, L. B. and J. L. Price (1978a). Association and commissural fiber systems of the olfactory cortex of the rat, i. Systems originating in the piriform corte4x and adjacent areas. *J. Comp. Neurol. 178*, 711–740.

Haberly, L. B. and J. L. Price (1978b). Association and commissural fiber systems of the olfactory cortex of the rat, ii. Systems originating in the olfactory peduncle. *J. Comp. Neurol. 181*, 781–808.

Hasselmo, M. (1993). Acetylcholine and learning in a cortical associative memory. *Neural Computation 5*, 32–44.

Hasselmo, M. (1995). Neuromodulation and cortical function: Modeling the physiological basis of behavior. *Behav. Brain Res. 67*, 1–27.

Hirase, H. and M. Recce (1995). Performance analysis of progressive recall in partially connected recurrent networks. In *ICANN'95 Proceedings of International Conference on Artificial Neural Networks*, Paris,France, pp. 509–514.

Hirase, H. and M. Recce (1996a). Interneuron plasticity in associative networks (submitted). In *CNS'96 Computational neuroscience meeting*, MA.

Hirase, H. and M. Recce (1996b). A search for the optimal thresholding sequence in an associative memory. *Network: Computation in Neural Systems*, in press.

Hodgkin, A. L. and A. F. Huxley (1952a). Currents carried by sodium and pottasium ions through the membrane of the giant axon of loligo. *J. Physiol. Lond. 116*, 449–472.

Hodgkin, A. L. and A. F. Huxley (1952b). The components of membrane conductance in the giant axon of loligo. *J. Physiol. Lond. 116*, 473–496.

Hodgkin, A. L. and A. F. Huxley (1952c). The dual effect of membrane potential on sodium conductance in the giant axon of loligo. *J. Physiol. Lond. 116*, 497–506.

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

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. 79*, 6871–6875.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two state neurons. *Proc. Nat. Acad. Sci. 81*, 3088–3092.

Hopfield, J. J., D. I. Feinstein, and R. G. Palmer (1983). 'unlearning' has a stabilising effect in collective memories. *Nature (Lond) 304*, 158–159.

Ishizuka, N., J. Weber, and D. G. Amaral (1990). Organization of intrahippocampal projections originating from CA3 pyramidal cells in the rat. *J. Comp. Neurol. 295*, 580–623.

Jarrard, L. E. (1993). On the role of the hippocampus in learning and memory in the rat. *Behavioral and neural biology 60*, 9–26.

Kelso, S. R., A. H. Ganong, and T. H. Brown (1986). Hebbian synapses in the hippocampus. *Proc. Nat. Acad. Sci. 83*, 5326–5331.

Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers C-21*, 353–359.

Komatsu, Y. (1994). Age-dependent long-term potentiation of inhibitory synaptic transmission in rat visual cortex. *J. Neurosci. 14*, 6488–6499.

Komatsu, Y. and M. Iwakiri (1993). Long-term modification of inhibitory synaptic transmission in developing visual cortex. *Neuroreport 4*, 907–910.

Korn, H., Y. Oda, and D. S. Faber (1992). Long-term potentiation of inhibitory circuits and synapses in the central nervous system. *Proc. Nat. Acad. Sci. 89*, 440–443.

Kosaka, T., H. Katsumaru, K. Hama, J. Y. Wu, and C. W. Heizmann (1987). GABAergic neurons containing the Ca2+-binding protein parvalbumin in the rat hippocampus and dentate gyrus. *Brain Res. 419*, 119–130.

Lansner, A. and O. Ekeberg (1985). Reliability and speed of recall in an associative network. *IEEE Trans. PAMI 7*, 490–498.

Lisman, J. (1994). The CaM kinase II hypothesis for the storage of synaptic memory. *Trends in Neuroscience 17*, 406–412.

Little, W. A. (1974). The existence of persistent states in the brain. *Mathematical Biosciences 19*, 101–120.

Little, W. A. and G. L. Shaw (1975). A statistical theory of short and long term memory. *Behavioural Biology 14*, 115–133.

Lorente de Nó, R. (1934). Studies of the structure of the cerebral cortex: II. continuation of the study of the ammonic system. *J. Psychol. Neurol. 46*, 113–177.

Malinow, R. and J. P. Miller (1986). Postsynaptic hyperpolarization during conditioning reversibly blocks induction of long-term-potentiation. *Nature (Lond) 320*, 529–530.

Marcus, C. M., F. R. Waugh, and R. M. Westervelt (1990). Associative memory in an analog iterated-map neural network. *Physical Review A 41*, 3355–3364.

Marcus, C. M. and R. M. Westervelt (1989). Dynamics of iterated-map neural networks. *Physical Review A 40*, 501–504.

Markram, H. and M. Tsodyks (1996). Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature (Lond) 382*, 807–810.

Marr, D. (1970). A theory for cerebral neocortex. *Proc. R. Soc. Lond. B 176*, 161–234.

Marr, D. (1971). Simple memory: a theory for archicortex. *Phil. Trans. R. Soc. Lond. B 176*, 23–81.

Marty, A. and I. Llano (1995). Modulation of inhibitory synapses in the mammalian brain. *Curr. Opinion Neurobiol. 5*, 335–341.

McNaughton, B. L. and R. G. M. Morris (1987). Hippocampal synaptic enhancement and information storage within a distributed memory system. *Trends in Neuroscience 10*, 408–415.

McNaughton, B. L. and L. Nadel (1990). Hebb-marr networks and the neurobiological representation of action in space. In M. A. Gluck and D. E. Rumelhart (Eds.), *Neuroscience and Connectionist Theory*, pp. 1–64. Lawrence Erlbaum Associates.

Meibach, R. C. and A. Siegel (1977). Efferent connections of the septal area in the rat: an analysis utilizing retrograde and anterograde transport methods. *Brain. Res. 119*, 1–20.

Miles, R. (1988). Plasticity of recurrent excitatory synapses between CA3 hippocampal pyramidal cells. *Soc. for Neurosci. Abstr. 14*, 19.

Miles, R. (1990a). Synaptic excitation of inhibitory cells by single CA3 hippocampal cells of the guinea-pig in vitro. *J. Physiol. 428*, 61–77.

Miles, R. (1990b). Variation in strength of inhibitory synapses in the CA3 region of guinea-pig hippocampus in vitro. *J. Physiol. 431*, 659–676.

Miles, R. and R. K. S. Wong (1987). Inhibitory control of local excitatory circuits in the guinea-pig hippocampus. *J. Physiol. 388*, 611–629.

Minai, A. A. and W. B. Levy (1994). Setting the activity level in sparse random networks. *Neural Computation 6*, 85–99.

Misgeld, U. and M. Frotscher (1986). Post-synaptic-GABAergic inhibition of non-pyramidal neurones in the guineapig hippocampus. *Neuroscience 19*, 193–206.

Mizumori, S. J. Y., B. L. McNaughton, and C. A. Barnes (1989). A comparison of supramammillary and medial septal influences on hippocampal field potentials and single-unit activity. *J. Neurophysiol. 61*, 15–31.

Muller, R. U. and J. L. Kubie (1987). The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *J. Neurosci. 7*, 1951–1968.

Muller, R. U., J. L. Kubie, and J. B. J. Rank (1987). Spatial firing patterns of hippocampal complex-spike cells in a fixed environment. *J. Neurosci. 7*, 1935–1950.

Nadal, J.-P. and G. Toulouse (1990). Information storage in sparsely coded memory nets. *Network: Computation in Neural Systems 297*, 681–683.

O'Keefe, J. and J. Dostrovsky (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research 34*, 171–175.

O'Keefe, J. and L. Nadel (1978). *The Hippocampus as a Cognitive Map.* Oxford, UK: Clarendon Press.

O'Keefe, J. and M. L. Recce (1993). Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus 3*, 317–330.

O'Keefe, J. and A. Speakman (1987). Single unit activity in the rat hippocampus during a spatial memory task. *Exp. Brain Res. 68*, 1–27.

Ono, T., K. Nakamura, M. Fukuda, and R. Tamura (1991). Place recognition responses of neurons in monkey hippocampus. *Neurosci. Lett. 121*, 194–198.

Otis, T. S., Y. De Koninck, and I. Mody (1994). Lasting potentiation of inhibition is associated with increased number of γ-aminobutyric acid type a receptors activated during miniature inhibitory postsynaptic currents. *Proc. Nat. Acad. Sci. 91*, 7698–7702.

Palm, G. (1980). On associative memory. *Biological Cybernetics 36*, 646–658.

Palm, G. (1981). On storage capacity of an associative memory with randomly distributed storage elements. *Biological Cybernetics 39*, 125–127.

Palm, G. (1988). Local synaptic rules with maximal information storage capacity. In H. Haken (Ed.), *Neural and Synergetic Computers*, pp. 100–110. Berlin: Springer-Verlag.

Palm, G. and F. T. Sommer (1992). Information capacity in recurrent McCulloch-Pitts networks with sparsely coded memory states. *Network: Computation in Neural Systems 3*, 177–186.

Raisman, G. (1966). The connecxions of the septum. *Brain 89*, 317–348.

Read, W., V. I. Nenov, and E. Halgren (1994). Role of inhibition in memory retrieval by hippocampal area CA3. *Neuroscience and Biobehavioral Reviews 18*, 55–68.

Rolls, E. T. and A. Treves (1990). The relative advantages of sparse versus distributed encoding for associative neuronal networks in the brain.

Network: Computation in Neural Systems 1, 407–421.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature (Lond) 323*, 533–536.

Sayer, R., S. Redman, and P. Andersen (1989). Amplitude fluctuations in small EPSPs recorded from CA1 pyramidal cells in the guinea pig hippocampal slice. *J. Neurosci. 9*, 840–850.

Schwartzkroin, P. A. and D. D. Kundel (1985). Morphology of identified interneurons in the CA1 regions of guinea pig hippocampus. *J. Comp. Neurol. 232*, 205–218.

Scoville, W. B. and B. Milner (1957). Loss of recent memory after bilateral hippocampal lesions. *J. Neurol. Neurosurg. Psychiatry 20*, 11–21.

Sejnowski, T. J. (1977). Strong covarience with nonlinearly interacting neurons. *J. Math. Biol. 69*, 303–321.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal 27*, 379–423,623–656.

Skaggs, W. E., B. L. McNaughton, and M. A. Wilson (1996). Theta-phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus 6*, 149–172.

Squire, L. R. (1992). Memory and the hippocampus: a synthesis from findings with rats, monkeys and humans. *Psych. Rev. 99*, 195–231.

Steinbuch, K. (1961). Die Lernmatrix. *Kybernetik 1*, 36–45.

Stevens, C. F. and Y. Wang (1995). Facilitation and depression at single central synapses. *Neuron 14*, 795–802.

Stewart, M. and S. E. Fox (1990). Do septal neurons pace the hippocampal theta rhythm? *Trends in Neuroscience 13*, 163–168.

Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning.* Ph. D. thesis, University of Massachusetts, MA.

Tamamaki, N. and Y. Nojyo (1991). Crossing fiber arrays in the rat hippocampus as demonstrated by three-dimensional reconstruction. *J. Comp. Neurol. 303*, 435–442.

Traub, R. D. and R. Miles (1991). *Neuronal networks of the hippocampus.* UK: Cambridge University Press.

Treves, A. (1991a). Dilution and sparse coding in threshold-linear nets. *Journal of Physics A24*, 327–335.

Treves, A. and E. T. Rolls (1991). What determines the capacity of autoassoative memories in the brain? *Network: Computation in Neural Systems 2*, 371–397.

Treves, A. and E. T. Rolls (1992). Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network. *Hippocampus 2*, 189–200.

Treves, A. and E. T. Rolls (1994). Computational analysis of the role of the hippocampus in memory. *Hippocampus 4*, 374–391.

Vanderwolf, C. H. (1969). Hippocampal electrical activity and voluntary movement in the rat. *Electroencephalogr. Clin. Neurophysiol. 26*, 407–418.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Ph. D. thesis, Harvard University.

Whittington, M. A., R. D. Traub, and J. G. R. Jefferys (1995). Synchronized oscillations in interneuron networks driven by metabotropic glutamate receptor activation. *Nature (Lond) 373*, 612–615.

Wigstrom, H., B. Gustafsson, and Y.-Y. Huang (1986). Hippocampal long-lasting potentiation is induced byh pairing single afferent volleys with intracellularly injected depolarizing current pulses. *Acta Physiol. Scand. 126*, 317–319.

Willshaw, D. J. (1971). *Models of Distributed Associative Memory.* Ph. D. thesis, University of Edinburgh.

Willshaw, D. J. and J. T. Buckingham (1990). An assessment of marr's theory of the hippocampus as a temporary memory store. *Phil. Trans. R. Soc. Lond. B 329*, 205–215.

Willshaw, D. J., O. P. Buneman, and H. Longuet-Higgins (1969). Non-holographic associative memory. *Nature (Lond) 222*, 960–962.

Woodson, W., L. Nitecka, and Y. Ben-Ari (1989). Organization of the rat hippocampal formation: a quantative immunocytochemical study. *J. Comp. Neurol. 280*, 254–271.