Network 2 (1991) 399–422. Printed in the UK

Neural network models of list learning

Neil Burgess†§, J L Shapiro‡ and M A Moore†

† Department of Theoretical Physics, University of Manchester, Manchester M13 9PL, UK

‡ Department of Computer Science, University of Manchester, Manchester M13 9PL, UK

Received 8 July 1991

Abstract. A neural network model is developed which captures the results of human memory experiments on learning lists of items. The psychological experiments on learning lists are reviewed. Hopfield–Parisi type neural networks are used to model many of the simpler features of order effects in serial recall. The recall of items as a function of their number, their position in the list and their similarity is investigated with simulations. More complex experiments involving different categories of items are modelled using correlated patterns of activity. Insight into how the models work is gained by consideration of the distribution of weights and signal-to-noise ratio arguments.

1. Introduction

Since the resurgence of neural networks, there has been much interest in the macroscopic behaviour of simple mathematical models of memory. The Hopfield model [1] in particular has been widely used to capture some features of human memory, such as content-addressability and error tolerance. Other neural network models have been proposed to model features of human memory [2, 3, 4] as well. The properties of many such models have been extensively studied in their own right, but how good are they as models of human memory?

The answer to this question lies in comparisons between the psychological data on the behaviour of human memory and the behaviour of the models. Although the motivation for constructing neural network models of memory comes from the microscopic level of neurobiology, their utility lies in the similarity of the macroscopic properties they exhibit. As most neural networks are crude idealizations of neurophysiology at best, it seems crucial that they capture the phenomena correctly. A number of researchers have studied neural network models in a variety of psychological domains, including Virasoro [5], Amit *et al* [6], Nadal *et al* [7], Grossberg [2] and many others (see section 3).

One of the simplest and most widely used memory tasks is that of list learning. Data has been collected on performance under an extremely wide range of conditions. In particular there is detailed information on the relative likelihood of errors occurring at different positions in a list. We attempt to model these experiments using Hopfield-type networks and to compare the errors made (as a function of position) with the

§ Present address: Department of Anatomy, University College, London WC1E 6BT, UK.

0954-898X/91/040399+24$03.50

psychological data. We also consider the mechanisms responsible for the behaviour of the model and briefly review some of the explanations of the psychological data.

The task of hearing or seeing and then recalling a list of items involves processing on many levels and by many different systems. However, storage of the internal representations of presented items is one of the most fundamental steps involved in these experiments. This is what we model here. At the present state of knowledge, it is not possible to produce a more complete theory that is even approximately accurate at all levels of complexity. To model high-level brain functions such as memory we must make gross approximations at the neurobiological level. The construction of even very crude models of memory from such simplified building blocks presumes that the macroscopic behaviour shown is not crucially dependent on the microscopic details.

Of course, as more detailed neurobiological information on the relevant structures of the brain become available, models and explanations of psychological phenomena become more constrained. There has been much progress in this direction, see Morris [8].

2. Review of list learning experiments

2.1. Simple (homogeneous) lists

The basic elements of a list learning experiment are as follows. The subject is presented with a list of items, such as words, one by one, and is then asked to recall them either immediately or after doing some other task. Recall can be 'free' in which items can be recalled in any order, or 'serial' in which the items must be recalled in the order of presentation. The types of items used and the way in which they are presented can be varied as can their presentation rate.

Our model will try to reproduce the typical features of a list learning experiment under free recall. There is a wealth of data on different versions of these experiments. One reason for this degree of interest is that the free recall task was taken as providing evidence that human memory consists of at least two separate stores [9] (of which more below). An overview of the subject for non-experts can be found in [10] and [11]. We outline here some of the features of most interest to us.

Figure 1 shows a typical graph of the probability of correctly recalling an item from a particular position in the list. This is called a serial position curve. When recall is immediate, this curve—raised at both ends and flat in the middle—occurs under a very wide range of conditions. This shape is characteristic of the curve irrespective of whether the items are words or nonsense, presented quickly or slowly or even if the subject is drunk or sober. The tendencies for items at the beginning or at the end of a list to be recalled better than the other items are known as primacy and recency respectively.

Recency typically extends over the last two to five items and is unaffected by factors like the rate of presentation or the familiarity of the subject with the items or by the overall length of the list. However, if recall is not immediately after presentation but after some intervening task like simple arithmetic then the recency effect is no longer present, although the remainder of the serial postion curve is unaffected. The rest of the curve, including the primacy region, is affected by the rate of presentation and the number and type of item used. Here recall is better for

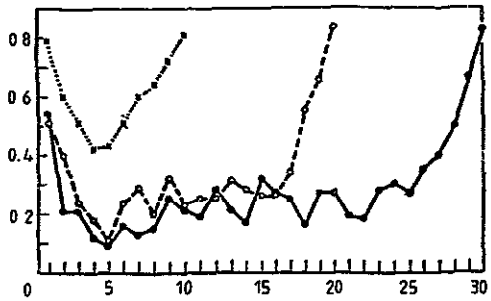

Figure 1. Serial position curves (showing the probability of correct recall versus position in the list) for lists of 10 (full line), 20 (broken line), and 30 (dotted line) words with immediate free recall. Taken from Postman and Phillips [15].

words that are well-known or slowly presented and for shorter lists. However, none of these factors influence recall in the recency region of the curve.

There are many theories in psychology for the explanation of these effects which are dealt with in more detail elsewhere [11,12]. However, the two models to have received the most attention historically are interference theory and the two-system model. A very brief outline of these follows.

In the two-system model proposed by Atkinson and Shiffrin [9] items initially go into a short-term store (STS) from which they are transferred into a permanent long-term store (LTS). A control process of rehearsal can be used to prolong the storage of an item in STS and hence increase the probability of it begin transferred to LTS. The STS has a limited capacity; the first item in is the first one out. Thus the recall of the last few items in the list is boosted by their being in the STS. This offers an explanation of recency. The disappearance of recency when recall is not immediate is explained by subsequent material displacing the last few items from the STS. The explanation of primacy in this model is that rehearsal of the memory trace takes place as soon as it has been presented, so that earlier patterns tend to be rehearsed more. Rehearsal could be conscious or subconscious (see discussions of the 'articulatory loop' [10,11]); it is thought to be cumulative in that previous items are rehearsed whenever a new item is rehearsed.

Interference theory maintains that there is but a single memory store. It holds that forgetting is caused by the memory trace becoming obscured by others rather than simply decaying with time. It was found that the recall of a list of items was impaired by the learning of a second list before recall [13]. This tendency of a later list to interfere with the recall of an earlier one is called retroactive interference or RI. The interference of an earlier list with the recall of a later one is called proactive interference or PI [14]. In both cases the more similar the contents of the lists are the greater the effect is. We will look at this more closely in the next section.

Interference theory can also be used to explain forgetting within a single list of items. Thus, recency is due to RI between items within a list (as there are fewer subsequent items to interfere with the later items). Primacy is due to PI between items within a list (as there are fewer previously learned items to interfere with the earlier items). When recall is not immediate but follows an intervening task, the

RI on the last few items is increased and PI has more time to build up so that the recency disappears [15].

2.2. Lists of different types of items and release from proactive inhibition

In the above discussion, there was no account of the actual content of the lists. The items in the lists were taken as being equivalent. This is appropriate when the items are digits or taken from some other homogeneous list of words. However, a number of effects have been observed when the items can be more or less similar with the other items, either similar in sound or similar in meaning.

An interesting piece of evidence for interference theory is the phenomenon of release from proactive inhibition (RPI) [16]. In this the subject is given a series of trials involving the recall of semantically similar items. PI builds up rapidly during the experiment, causing recall to deteriorate after the first trial. However, when the type of item is changed recall is dramatically improved for the next trial: the recall has been 'released' from the PI by the change to dissimilar items. The effect is more marked when compared to a control in which the type of item is the same within a trial, but is always changed on the subsequent trial. Thus, PI does not have a chance to build up. An example experiment is shown in figure 2. Here each group is given three words to remember per trial. For the control (labelled 4A) the category was changed every trial; for the test group (labelled 4S) the category was changed every seventh trial. The change in type of item could be between words and digits, or between words from different semantic categories (e.g. animals, drinks, plants, etc). The effect was not observed in a 'final free recall' at the end of the experiment of all the items shown [17]. This was interpreted as evidence that RPI is a recall phenomenon rather than a storage phenomenon.

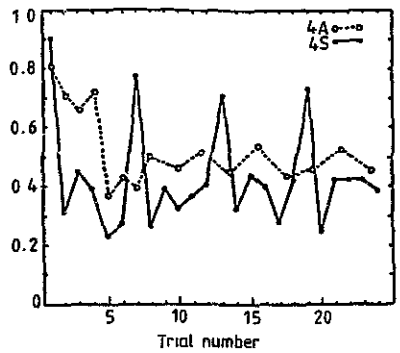

Figure 2. Release from proactive inhibition. In each trial the subjects had to remember three words from a taxonomic category. For group 4A items were taken alternately from four categories. For group 4S the category was changed every seventh trial After Loess [41].

A related phenomenon that is observed when learning different types of items is the Von Restorff [18] effect. This is the observation that a distinct item in an otherwise homogeneous list will be better recalled than the other items. Green [19] showed subsequently that this improved recall was not restricted to an odd one out in a list but occurs, more generally, with the first novel item reached in a list. This

item could be alone or the first item in a block of similar items thus connecting the Von Restorff effect with RPI within a list.

3. Modelling list learning

In this section, we describe our model for serial recall of lists. Our model is based on the Hopfield model, which is a simple model of autoassociative, content-addressable memory. Because the Hopfield model is by its nature a model of memory, it has already been widely used to interpret psychological and neurophysiological memory phenomena. More computationally powerful models, such as the multilayer perceptron, are better suited to more general processing such as language acquisition and vision. However, a small amount of work has been done on forgetting in multilayer perceptrons. Hinton and Plaut [20] looked at the role of decay in weights in learning. Another very detailed model of working memory using multilayer perceptrons was outlined in [21]. The major difference between these models and a Hopfield approach is that the Hopfield model uses a local Hebb rule which makes pattern storage a 'one-shot' event (i.e. the network can store a pattern after a single presentation). Multilayer perceptrons use backpropagation, which learns gradually. The fast learning of the Hebb rule is appropriate for the rapid and short-term storage in list learning. In addition, because the Hopfield model has a limited storage capacity, forgetting in the model described below is due to interference, whereas in Hinton and Plaut's model it is due to decay.

Another approach which has been widely used to model memory is based on a model proposed by Grossberg [2]. Interesting results have been obtained by a number of researchers. For example, a model of the interaction between short and long-term memory in list learning has been developed by Schreter and Pfeifer [4]. A model for learning sequences has been developed by Houghton [3] within this paradigm. The major difference between this approach and the Hopfield approach is that in the latter memory states are distributed across the network, whereas in the former the representations are local.

We now review the main features of the Hopfield model. For a detailed introduction, see Amit [22]. In simplest terms, the Hopfield model is a model of N two-state units which interact with each other via a set of weights. Patterns are stored in the model via Hebbian learning, and the system functions as a content-addressable, autoassociative memory. Pattern storage is a 'one-shot' event; the network stores a pattern after one presentation. However, the Hopfield model has a limited capacity, so it may fail to store a pattern on presentation or forget it later.

The units are completely interconnected by the weights, which are denoted J_{ij}; this connects the ith unit to the jth one. The Hebbian learning rule stores a pattern by increasing J_{ij} if units i and j are the same and decrementing J_{ij} if they are different. The model can store a number of patterns proportional to N. The proportionality constant, α_c, depends upon the relationship between the patterns; for uncorrelated patterns $\alpha_c \approx 0.14$ [23].

If the storage capacity is not exceeded, the model functions as a content-addressable memory. When a noisy version of one of the stored patterns is input, the system relaxes to a pattern very close to the stored one. However, when the number of stored patterns exceeds the strorage capacity, the system relaxes to a degraded version of the stored pattern. The greater the number of stored patterns, the greater

is the degradation. For N very large, this transition is very sharp; the system catastrophically forgets everything when more than $\alpha_c N$ patterns are stored. However, for N not very large, the forgetting is gradual.

A number of researchers have modified the Hopfield model to act as a working memory. In these models, the memory acts as a buffer: it can hold up to $\alpha_c N$ memories, each one stored more or less equally (there is a slight degradation of the pattern which has been in the buffer longest), and when the storage capacity is exceeded, the earliest patterns are forgotten completely. In these models α_c is less than in the Hopfield model, typically $\alpha \approx 0.05$ [24]. These models show perfect recency.

There have been two approaches to achieve this. One was proposed in [25] and [26], where the Hebb learning rule has a different factor for each pattern (possibly modelling changes in the attention [7]) of the subject. In this scheme (referred to as 'marginalist' learning) each new pattern learnt causes a change in the connection weights whose magnitude increases exponentially with the number of patterns learnt. The other approach was suggested by Hopfield [1]. In this approach, the usual Hebb learning rule is used, but the magnitude of the synaptic weights is bounded. When the bound is reached, learning can decrease the magnitude of the weights, but not increase it. This model was studied numerically by Parisi [27] and in [25]. It was solved analytically by van Hemmen *et al* [24]. We refer to this model as the Hopfield–Parisi model. If the parameters are carefully chosen, this model acts like a buffer with recency.

A similar model was proposed by Nadal *et al* [25] as 'learning within bounds' and a modification of the 'marginalist learning' scheme was presented that is equivalent to bounding the weights [7]. (See also [28] and [29] for related work.) A related model was developed by Peretto [30] in which the weights are also bounded but once the weights reach the bound, they stay there forever. This model shows primacy, but no recency. A recent model has been produced by Wong *et al* [31] using weights that have different probabilities of being strengthened or weakened during learning in which primacy and recency can be seen.

In order to reproduce the experiments on list learning, we must introduce mechanisms for primacy and recency, and a way of producing the similarity effects. The simplest way to produce primacy in the model is for the input of a new pattern to cause the magnitude of each weight to be increased in proportion to itself. Thus the learning of each new pattern is accompanied by a reinforcement of the present state of each weight (i.e. a reinforcement of the previously stored patterns) which will tend to cause primacy. The psychological interpretation is of passive cumulative rehearsal or consolidation.

The source of recency we use is that of the Hopfield–Parisi model; the magnitude of the weights is bounded. This seems the most natural source of recency, and clearly makes forgetting of early items an interference phenomena. The alternative is to use a different learning factor for each pattern to produce virtually any serial position curve. Of course putting the required behaviour into a model by hand and producing it again does not seem very productive. We would prefer a model involving a small number of parameters which showed plausible emergent behaviour.

We introduce two parameters to control the relative strengths of these two effects. Let the size of the change of weights due to *new* information be ϵ. The factor by which previously stored patterns are reinforced is γ. If γ equals one, there is no reinforcement of previous patterns and the model is equivalent to the Hopfield–Parisi

model. If γ is bigger than one, there is reinforcement; if γ is less than one, there is actual decay of previous items.

The relative strength of the mechanisms for primacy and recency is conveniently parametrized by the ratio $x^* = \epsilon/(\gamma - 1)$. It is the interplay between the two mechanisms for primacy and recency which gives rise to much of the rich set of behaviour described below.

These two effects are expressed mathematically in the updating rule for the weights. This is the essential equation of the model, and is,

$$J_{ij}(t+1) = f(\gamma J_{ij}(t) + \epsilon \xi_i^{t+1} \xi_j^{t+1})$$
$$f(x) = \begin{cases} x & \text{if } |x| < 1 \\ \operatorname{sgn}(x) & \text{otherwise} \end{cases} \tag{1}$$

and $\gamma > 1$, $\epsilon > 0$.

The similarity effects are modelled by allowing correlations to occur between patterns. This is discussed further in section 4.2.

The basic experiment we are modelling is as follows: the subject is ready for the experiment at $t = 0$. At each time step ($t = 1, 2, \ldots$) the subject is presented with an item (e.g. they hear a word). At time $t = p$, the experiment ends and the subject is asked to recall the items (in any order). Note that p is the number of patterns presented to the subject. After several such trials a serial position curve can be plotted showing the average fraction of items correctly recalled versus their position in the list.

The model is a fully interconnected network of N units S_i taking values ± 1. The weights of the connections J_{ij} between units i and j were initially zero (i.e. starting with a so-called *tabula rasa*; see the discussion in section 5).

Presentation of the list of p items was modelled by setting the units S_i to be successively equal to patterns activity ξ_i^ν for $\nu = 1, \ldots p$, that is, setting:

$$S_i(t=1) = \xi_i^1 \qquad S_i(t=\nu) = \xi_i^\nu \qquad i = 1, \ldots, N \qquad \nu = 1, \ldots p.$$

These patterns are 'learnt' by updating the weights according to equation (1).

Recall of a pattern is modelled purely by testing how well stored it is. The units are set to a noisy version of the pattern (the amount of noise being the fraction of units that have changed sign; we take this to be 0.2) and then relax sequentially under the dynamics:

$$S_i = \operatorname{sgn}\left(\sum_{j=1}^{N} J_{ij} S_j\right) \tag{2}$$

until a stationary state S^* is reached for which equation (2) is true for all i. The recall overlap

$$m^\nu = \frac{1}{N} \sum_{i=1}^{N} S_i^* \xi_i^\nu$$

is interpreted as a measure of how well pattern ν is remembered. All 'recall overlaps' m^ν subsequently shown have been averaged using ten different noisy versions of the

pattern to be recalled. This of course assumes much higher-level processing by other systems. The bare pattern completion/recognition task on the internal representation ξ_i^ν (measured by m^ν) might represent a basic step in recall. We do not, however, have a detailed model for the mechanism of recall itself, only storage (although such a mechanism is probably essential to understand all of the phenomena, as we discuss in the final section).

Here we will study the behaviour of the model in two ways. We will do numerical simulations to determine the types of behaviour shown by the network. In section 5 we will characterize the behaviour of the model in terms of its parameters by considering the distribution of the weights using simple arguments.

4. The serial position curves

4.1. Simple lists

In this section we show the serial position curves for our model with various parameter values (from numerical simulations) and explain them descriptively. A more detailed analysis is given in section 5, and in reference [32].

There are three free parameters in the model, ϵ, γ, and N. The first controls the change in the weights when a new pattern is stored. Thus, it is associated with recency. The second controls the amount which old patterns are reinforced, and is associated with primacy. The third parameter is N, the number of nodes in the network. This parameter controls the capacity of the network. The relative importance of ϵ and γ is expressed in an additional parameter

$$x^* = \epsilon/(\gamma - 1).$$

There are essentially two causes of failure to recall a pattern in this model. Since the memories are stored in the weights, the model will fail to recall a pattern when the weights are not sufficiently strongly correlated with that pattern. How the strength of storage depends upon order within a list is controlled by the parameters ϵ and γ. These two parameters can be adjusted so as to make either the patterns at the beginning of the list or at the end of the list most influence the weights.

The second mechanism for forgetting is interference with other stored patterns. This is very well known in Hopfield-type models and is the cause of the finite storage capacity of the models. The importance of interference is effectively controlled by N, the size of the network. For example, when the patterns are equally and optimally stored in the weights, interference causes all patterns to be forgotten when more than $2N$ patterns are stored [33]. In the large-N limit, this leads to catastrophic forgetting, i.e. the transition from from good recall to total forgetting is abrupt. When N is finite the forgetting is gradual. The storage capacity, the maximum number of patterns which the model can store, typically scales with N. It is expressed in terms of α_c which is defined to be the maximum number of patterns stored divided by N, p_c/N. This capacity depends upon the storage rule and correlations between the patterns. In our system, it will be determined via simulations.

In order to understand the relative strength of storage of the patterns in the weights, consider the distribution of weights $P(J_{ij})$. Initially all the weights are set to zero. As each pattern is input the weights are updated according to equation (1). The initial distribution $P(J_{ij})$ is a spike at zero, this broadens as patterns are learnt.

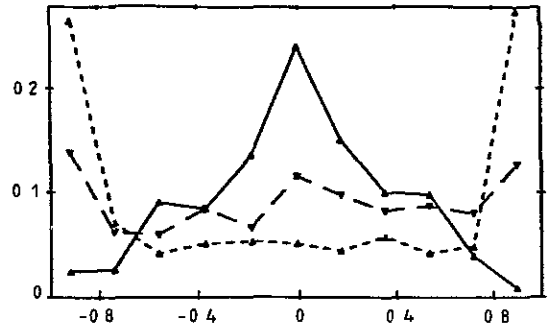

Figure 3. The distribution $P(J_{ij})$ (averaged over 10 simulations for 90 weights) after p patterns have been learnt, where $p = 10$ for the full line, 20 for the broken line and 30 for the dotted line. $\gamma = 1.05$, $\epsilon = 0.1$, $x^* = 2.0$, $p(1) = 24.8$ and $T_1 = 6.7$

The broadening continues until a significant fraction of weights reach ± 1 which they cannot exceed.

The effect of $\gamma > 1$ in equation (1) is to make $P(J_{ij})$ biased towards the extremes ± 1. Thus, after sufficient patterns have been presented $P(J_{ij})$ begins to build up at ± 1. Figure 3 shows a numerical simulation of the distribution of weights after 10, 20 and 30 patterns have been learnt.

Primacy can occur when $x^* \leqslant 1$. In this case, after a sufficient number of patterns have been stored, new patterns have little effect on the weights. This is a consequence of the dynamics of equation (1). Once the magnitude of a weight $|J_{ij}|$ exceeds x^*, the magnitude of that weight can never decrease; its sign has been determined by the patterns stored before it reached $\pm x^*$. Thus, x^* is a point of no return for the weights. After a considerable fraction of the weights reach this value, no new patterns can be stored and primacy occurs.

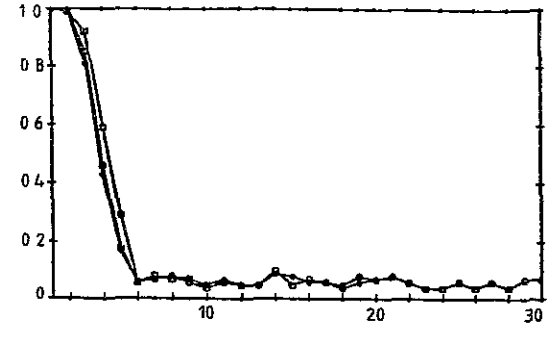

Figure 4. Serial position curves for 10, 20 and 30 patterns showing permanent primacy. All serial position curves are averaged over 10 simulations In each simulation recall is averaged over 10 noised-up versions of the pattern in which spins are flipped with a probability of 0.2. $N = 100$, $\gamma = 1.25$, $\epsilon = 0.2$, $x^* = 0.8$, and $p(x^*) = 5.2$.

The number of patterns which can be stored depends upon ϵ and γ in a way which is discussed in section 5.

A typical serial position curve for $x^* \leqslant 1$ is shown in figure 4. It shows permanent primacy (or 'imprinting'). In fact, there is some storage of recent patterns so long as the magnitudes of the weights are less than one. In the long-time limit, only the first n, say, patterns are remembered. For shorter times, more than that number can be recalled.

Recency can occur when $x^* > 1$. In this case, the sign of a weight can always be changed by subsequent patterns. However, due to the fact that the weights are bounded at ± 1, the weights become decorrelated with the earlier stored patterns. This is because recent patterns can cause the weights to change sign. In this regime, the model shows recency, as in the Hopfield–Parisi model. A typical curve is shown in figure 5.

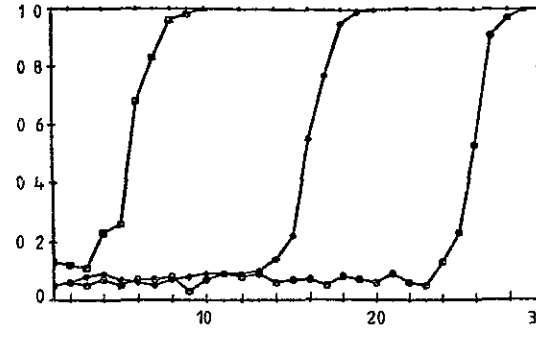

Figure 5. Serial position curves for 10, 20 and 30 patterns showing recency The recency part of the curve is flat due to saturating at perfect recall although the later patterns are stored more strongly in the weights $N = 100$, $\gamma = 1.05$, $\epsilon = 0.45$, $x^* = 9.0$, $p(1) = 4.2$ and $T_1 = 2.3$.

If N is arbitrarily large while the parameters ϵ and γ are fixed, then the behaviour of the model is set solely by x^*: $x^* < 1$ yields primacy; $x^* > 1$ gives recency. In this limit interference is negligible (unless N diverges and γ and ϵ scale with N in a suitable way; see section 5). However, if N is sufficiently small, interference can lead to catastrophic forgetting, and all three behaviours are possible depending upon the parameters chosen. In this case, patterns are lost when more than the critical capacity p_c ($\approx 0.05N$ from simulations) is input to the system. When fewer patterns are input, primacy or recency is shown as before.

When N is finite and for intermediate times (i.e. intermediate numbers of patterns) the model can show primacy and recency together. This behaviour is most pronounced near the boundary between primacy and recency. The patterns affected least by this forgetting are those stored by the most weights. These tend to be the patterns at the beginning of the list and the patterns at the end of the list. Thus the serial position curve is raised towards the ends, i.e. showing primacy and recency see figure 6 (and compare with figure 1).

There is primacy in the serial position curves when the number of patterns stored is not large. When the first few patterns are presented a large proportion of the

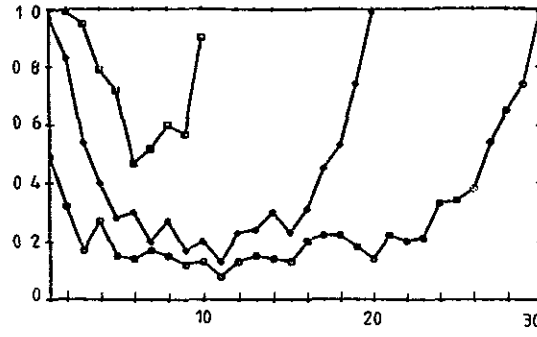

Figure 6. Serial position curves for 10, 20 and 30 patterns showing primacy and recency, with the amount of primacy decaying. $N = 100$, $\gamma = 1.14$, $\epsilon = 0\ 3$, $x^* = 2.14$, $p(1) = 5\ 6$ and $T_1 = 4\ 6$.

weights are nearly zero; these more strongly influence the sign of the weights than later patterns. As the number of patterns stored increases, the recall of the earlier patterns (i.e. the primacy) deteriorates as more weights change sign. The recall of the most recent patterns (i.e. the recency) is not affected by the list length. See figure 5.

The duration of the primacy and recency portions of the curves can be adjusted more or less independently via the parameters. Thus the relative amounts of primacy and recency are not determined by the model.

Since time does not appear explicitly in the model, the disappearance of recency when there is a task before recall can be modelled by the presentation of irrelevant patterns. It is clear from figure 7 that this will model the lack of recency in experiments in which the subject is given a task between hearing the list and recalling it. However, it should be noted that this would be accompanied by a slightly greater decrease in primacy than is observed in human data. This mechanism might be interpreted as the task displacing items from a limited buffer rather than as a build-up of proactive inhibition.

4.2. Lists of different types of item and correlated patterns

In this section, we describe a model in which the items in the lists can be similar or dissimilar to the other items in the lists. We will use this to model the experiments in which there is proactive interference between similar items in lists, but no interference between dissimilar items. These experiments were described in section 2.2, and include the so-called release from protective inhibition RPI effect and the Von Restoff effect.

In these experiments, the items to be recalled come from classes; items in the same class are similar, those from different classes are dissimilar. For example, some of the items could be numbers, others could be the names of flowers, and so forth. To model similarity of item, we use correlated patterns, whereas dissimilar patterns are uncorrelated. This assumes that the internal representations of similar items will themselves be similar. The storage of correlated patterns by Hopfield networks has received much attention recently: modifications of the Hebb rule have been proposed that store correlated patterns more efficiently and storage capacities have

been calculated in the large-network limit [34,35]. Here we want to model release from PI experiments with a finite size network and a finite number of patterns.

Each category or class of patterns is defined by an 'ancestor' (or 'prototype') pattern. The ancestor patterns are unbiased and uncorrelated. The patterns within each class have correlation r with their class ancestor and correlation $c = r^2$ with each other. Patterns from different classes are uncorrelated. (See section 5 for more details.)

Proactive inhibition is observed when a subject undergoes many successive trials (learning *and recalling* a short list) and the type item used is changed every few trials (see figure 2). To model the experiment in figure 2 we presented the network with 24 trials. Each trial consisted of learning and recalling three correlated patterns. The class from which the correlated patterns were drawn was changed every seventh trial Thus four different classes of 18 correlated patterns were used. *The connection weights were not reset to zero between lists.* The control experiment in figure 2 was also modelled: changing the class of pattern every trial.

The change of the weights is still governed by equation (1). However, we took $\gamma = 1$ (i.e. the straightforward Hopfield–Parisi model) to model these experiments because the reasons for putting $\gamma > 1$ in the model of learning a single list (to model rehearsal or consolidation see the discussion in section 6) do not apply to patterns from previous trials (i.e. those that have already been recalled). We were interested in modelling the overall change in performance between trials rather than the details of each trial. (To exactly reproduce the serial position curves within each trial we would probably need some reinforcement of the storage of patterns in a list but only for the duration of that trial.)

In figure 7 the full line shows the results of the simulation of this experiment, the broken line shows the simulation of a control experiment in which the category is changed every trial. The recall of a trial was taken to be the average recall of the three items in a trial.

We modelled the Von Restorff effect by learning a list of ten patterns from one correlation class and one uncorrelated pattern. Recall occurred after all eleven patterns had been presented. The recall overlap for the uncorrelated pattern was (unsurprisingly) better than for the correlated patterns for cases that we tried.

The recall performance of the network is best for the first trial and successively worse for later ones. There is new primacy each time a new class of correlated patterns is presented. This behaviour does not occur in the control experiment or in a final recall test of all the patterns used (this showed only recency). These results are consistent with the psychological data, see figure 2. It is interesting to note that the absence of RPI in a final free recall of items is interpreted as evidence that RPI is a recall phenomenon. Our experiment also showed this, but caused by strength of storage only, as our model has no seperate mechanism of recall.

Essentially the main cause of interference with the storage of a pattern comes from other patterns in the same category. Thus recall performance deteriorates within a class, and at the beginning of a set of trials from a new class recall is relatively good again. The ancestor pattern from each class becomes more strongly stored with the presentation of each pattern from that class (even though it is never presented itself). Eventually the ancestor pattern becomes so strongly stored that it dominates all the individual patterns. The basins of attraction for the individual patterns in a class are slowly lost in the growing basin of attraction of the ancestor pattern. Finally all that can be remembered are the common features of a class and none of the detail.

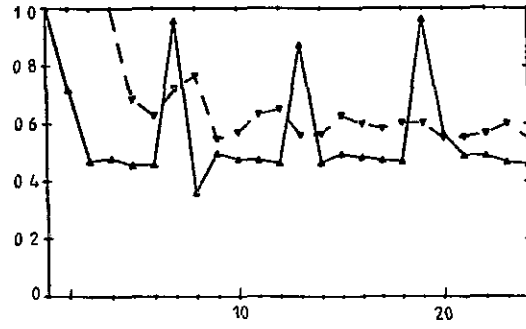

Figure 7. The average recall overlap of the three patterns in each trial. The category from which patterns were taken was changed every seventh trial (RPI experiment, full line) or every trial (control experiment, broken line) Patterns within a category have correlation 0.2, $\epsilon = 0\ 2$, $N = 700$, recall noise was 0 15

The fact that that these models remember the common aspects of a set of correlated patterns, and, in fact, eventually remember the common aspect *at the expense of the patterns which it has actually seen* is common to many Hebbian learning rules. This has lead many investigators to invent other rules which are more efficient at storing correlated patterns [33,34,35]. However, we note that the same effect has been observed in humans, see for example Posner and Keele [36], in which a prototype visual dot pattern was as well recognized as patterns actually seen. In addition, the deterioration of the storage of individual patterns in a class compared to the storage of the common feature of the class has been used to model prosopagnosia [5], which is a deficit in recognizing individuals while recognition of categories is unimpaired.

5. Mathematical analysis

In this section, we present a simple analysis which allows prediction of the behaviour of the model as a function of the parameters. In principle, the techniques of replicas [6] could be used to calculate the serial position curves described in previous sections. However, much of the curve comes from finite size effects, which makes such a calculation difficult. Instead, we rely on simple arguments to explain how the behaviour of the model depends on the various parameters. The effects of γ and ϵ are found from the distribution of the weights $P(J_{i_j})$ and correlations of the weights with stored patterns. Storage capacity is found from signal-to-noise analysis or simply from the the numerical simulations themselves.

5.1. Simple lists

It is useful to define four parameters to characterize the behaviour of the network:

1. $x^* = \epsilon/(\gamma - 1)$, a measure of the relative strength of the mechanisms for primacy and recency.
2. $p(1)$, the number of patterns stored before a significant fraction of weights reach the values ± 1 from their initial value of zero.

3. $p(x^*)$, the number of patterns stored before a significant fraction of weights reach the values $\pm x^*$ (this is only defined for $x^* \leqslant 1$).
4. p_{fp}, the first passage 'time', i.e. the average number of patterns stored before the value of a weight changes from -1 to $+1$ (this is only defined for $x^* > 1$).

These are, of course, functions of the three independent parameters γ, ϵ, and N.

First, consider the case $x^* \leqslant 1$. It is easy to see from equation (1) that when the magnitude of a weight exceeds x^*, the magnitude of that weight cannot decrease. The sign of that weight is determined at that point. Thus, we argue that the amount of pattern storage is set by $p(x^*)$ so long as the storage capacity is not exceeded. An intensive number of patterns will be stored when $p_c \gg p(x^*)$, and only the first $p(x^*)$ patterns will be stored in the long-time limit. Time is measured by the number of patterns input; in this case long time means long compared to $p(1)$. If the number of patterns input is less than $p(x^*)$, all patterns will be stored; if the number of patterns input is between $p(x^*)$ and $p(1)$, then some primacy will be present. Obviously, if $p_c \ll p(x^*)$, then the system will behave like the overloaded Hopfield model and forget catastrophically. *Primacy with an extensive number of patterns stored is also* possible. The storage capacity has been determined in numerical simulations. The value was found to be close to that of the Hopfield–Parisi model, $\alpha_c \approx 0.05$.

We estimate the quantity $p(x^*)$ by finding the p such that the second moment of the weights is equal to x^{*2}. It is straightforward to see (ignoring the effects of the bounds at ± 1) that

$$p(x^*) = \frac{\ln[2\gamma/(\gamma - 1)]}{2\ln(\gamma)}.$$

This result is derived in the appendix. This equation determines the boundary between the primacy regime and the forgetting regime. Extensive pattern storage is possible when γ scales like

$$\gamma \sim 1 + \frac{\ln(N)}{\alpha N}$$

and ϵ goes to zero to make x^* less than 1.

Likewise, $p(1)$ is estimated by setting the second moment to one. This yields,

$$p(1) = \frac{\ln[1 + (\gamma^2 - 1)/\epsilon^2]}{2\ln(\gamma)}.$$

These two parameters determine the number of patterns stored in the primacy regime. When the number of patterns seen by the system is much larger than $p(1)$ most of the weights are pegged at the values ± 1. The number of patterns stored is $p(x^*)$ and no additional patterns can be stored. For example, in figure 4 about five patterns are stored and $p(x^*)$ is 5.3. When the number of patterns is between $p(x^*)$ and $p(1)$, many of the weights are between x^* and 1 and there can be some storage of new items. In this case the number of patterns stored exceeds $p(x^*)$. If the number of patterns input is smaller than $p(x^*)$, then all patterns are stored. Of course, when the number of patterns input exceeds the storage capacity all patterns are lost.

For $x^* > 1$ the sign of a weight J_{ij} can always be changed by subsequent patterns. However, once a weight has reached a value near to ± 1 it remains near

to ± 1 for several time steps. The average number of patterns learnt before a weight J_{ij} changes sign from -1 to $+1$ is the first passage time, p_{fp}.

The first passage time gives an estimate of how long the system takes to forget. As a finite fraction of the weights have changed sign within this time, the weights have become decorrelated with patterns input more than time p_{fp} ago. To put it another way, the first passage time gives us an indication of the relative amounts of time the value of a weight spends near to zero compared to ± 1. Since a pattern can influence those weights which are near zero, this timescale gives an indication of how much a pattern will influence the weights.

If we accept the above explanation of forgetting, then the first passage time as a function of the parameters determines the boundary between the recency and catastrophic forgetting regimes. When $p_{\mathrm{fp}} \ll p_{\mathrm{c}}$ the last p_{fp} patterns will be stored; when $p_{\mathrm{fp}} \gg p_{\mathrm{c}}$ no patterns will be stored after many patterns have been input. The precise evaluation of the first passage time in equation (1) is not trivial. It is tempting to approximate the equation by the Langevin equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\frac{\mathrm{d}V(x)}{\mathrm{d}x} + \eta(t) \tag{3}$$

and use the well known Arrhenius result [37] for a particle escaping over a barrier potential $V(x) = -\frac{1}{2}(\gamma - 1)x^2$. However, this is not correct due to the fact that the noise is bounded. The Arrhenius result assumes Gaussian noise which can surmount any barrier if one waits long enough. The bounded noise of equation (1) can only push a particle over barriers smaller than a certain height. This is why if a weight exceeds x^*, it cannot escape. A consequence of this is that the first passage time diverges as x^* tends to one from above. (It is clear that the first passage time must diverge at $x^* = 1$ because in that case when a weight gets to ± 1 it sticks).

For $x^* \gg 1$ the Arrhenius result is a good approximation to p_{fp}. The Arrhenius equation gives the time for a particle to surmount a barrier due to thermal activation as proportional to $\exp(\Delta E/kT)$ where ΔE is the height of the barrier, k is Boltzmann's constant, and T is the temperature. This can be derived from equation (3) if the second moment of the noise η is proportional to kT. (See section 1.6 of [38] for derivation of the Arrhenius equation from an equation of this form.) For our case, this gives a first passage time proportional to

$$p_{\mathrm{fp}} \propto \exp(1/(\epsilon x^*)).$$

As $x^* \to 1$, we can compute the divergence term by expanding in the possible noise configurations which take the particle over the barrier. The leading-order configuration is one which takes p_1 steps in the same direction, where p_1 is the number of steps required to get from 1 to 0:

$$p_1 = \frac{\ln[x^*/(x^* - 1)]}{\ln(\gamma)}.$$

Thus, the leading contribution to the divergence is

$$T_1 = \left(\frac{x^*}{x^* - 1}\right)^{\ln 2/\ln \gamma}.$$

This is an upper bound for p_{fp}, as in this amount of time, the escape is bound to occur.

There is a minimal process to surmount the barrier, which is to get to the point where it takes more than one step to reach ± 1. This point is $\pm(1-\epsilon)/\gamma$. From this we find a lower bound for the average first passage time:

$$T_2 = \left(\frac{(\epsilon-1)/\gamma + x^*}{x^*-1}\right)^{\ln 2/\ln\gamma}.$$

In simulations, the second approximation gave better results; of course they exhibit the same divergence.

In this regime, $p(1)$ indicates the number of patterns which can be stored at the beginning of the list with little diminution. The first passage time determines the amount of patterns stored in recency when the number of patterns input is large compared to $p(1)$. For example, figure 5 shows storage of about four patterns in the recency parts of the curve. The approximation T_2 predicts 2.3 patterns recalled. Again, this assumes that the storage capacity is not exceeded, otherwise no patterns will be recalled.

It is in the region between the primacy and recency regimes that both can occur in the same experiment. This area occurs for x^* very to 1 and p_{fp} very close to p_c. Figure 8 may help to illustrate this. This picture shows the three regimes defined by what is recalled after a large number of patterns have been input. In this limit, either only the first patterns can be recalled, the last patterns can be recalled, or no patterns can be recalled. Because N is finite, the boundaries are not sharp; they sharpen as N grows. The boundary lines are determined by $p(x^*)/\alpha_c$ and p_{fp}/α_c. Figure 8 shows a plot of these parameters as a function of x^*. They are actually functions of γ and ϵ, but to simplify the picture they are plotted here against the single variable by keeping γ fixed. Thus, the precise shape of the curves will vary with γ; these are examples. The point marked with a star indicates a set of parameters which exhibit both primacy and recency.

5.2. Correlated patterns and inhomogeneous lists

In section 4 we used the learning of correlated patterns to model experiments involving different types of items. We took $\gamma = 1$, consequently an approximate analysis of the distribution of weights is relatively simple. We can understand the main features of the model in terms of signal-to-noise ratio arguments.

If the network is put into one of the memory states, say pattern $\{\xi_i^q\}$ then the local field on site i in the network after t patterns have been learnt in total is:

$$h_i(t) = \sum_{j=1, j\neq i}^{N} I_{ij}(t)\xi_j^q.$$

Since the unit at S_i obeys the dynamics $S_i(t+1) = \mathrm{sgn}(h_i(t))$ the 'signal' from pattern ξ_i^q for the state of the network to remain at pattern ξ_i^q is $S_q(t) = \langle h_i(t)\xi_i^q\rangle_i$. The 'noise' in the signal is

$$\Delta_q(t) = \left(\langle (h_i(t)\xi_i^q)^2\rangle_i - S_q^2(t)\right)^{1/2}.$$

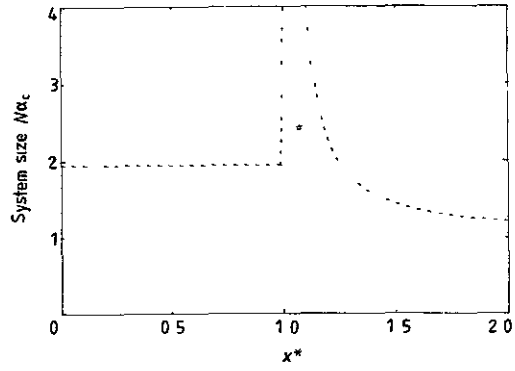

Figure 8. Schematic representation of the behaviour of the model against the parameters x^* and the network size $N\alpha_c$. The regions show what is recalled after a very large number of patterns have been seen. In the primacy region, only the earliest patterns are recalled; in the recency region, only the latest patterns are recalled; in the region labelled catastrophic forgetting, none of the patterns can be recalled. The point marked with a star is an example of parameters which show primacy and recency together. The broken lines do not represent sharp boundaries.

The ratio $R_q(t) = S_q(t)/\Delta_q(t)$ of the signal to the noise indicates the stability of the stored pattern $\{\xi_i^q\}$.

For the Hopfield model (i.e. if the weights were unbounded, corresponding to the small-ϵ limit in our model) we can calculate the signal-to-noise ratio directly.

We will do this for the RPI experiment described in section 4.2.

As described in section 4.2 the patterns learnt: $\{\xi_i^{\nu\mu}, \quad i = 1,\ldots,N, \nu = 1,\ldots,n_c, \mu = 1,\ldots n_p\}$ are from correlation classes of n_p patterns generated from n_c unbiased ancestor patterns: $\{\xi_i^\nu, i = 1,\ldots,N, \nu = 1,\ldots,n_c\}$. They have correlations: $\langle \xi_i^{\nu\mu}\xi_i^{\nu'\mu'}\rangle_i = \delta_{\nu\nu'}(c + (1-c)\delta_{\mu\mu'})$. We will calculate the signal-to-noise ratio $R_{sq}(t)$ for pattern ξ_i^{sq} at the moment that it is recalled.

In the RPI experiment recall of the previous three patterns occurred after the learning of every third pattern (i.e. at the end of each 'trial' of three patterns). The correlation class of each pattern is changed every n_p patterns. Suppose that ξ_i^{sq} is recalled at time t (notice that t must be a multiple of three). The t patterns learnt at time t include $(s-1)n_p$ patterns from different correlation classes and $t-(s-1)n_p = t_s$ patterns from the same class as ξ_i^{sq} ($t_s = q, q+1$ or $q+2$). We will denote the sum over the t learnt patterns $\{\xi_i^{\nu\mu}\}$, $(\nu,\mu) = (1,1),\ldots,(s,q)$, by

$$\sum_{\nu,\mu}{}' = \sum_{\nu=1}^{s-1}\sum_{\mu=1}^{n_p} + \delta_{\nu,s}\sum_{\mu=1}^{t_s}.$$

Then the signal from pattern ξ_i^{sq} is given by

$$\begin{aligned} S_{sq}(t) &= \langle h_i(t)\xi_i^{sq}\rangle_i \\ &= \epsilon\left\langle \sum_{j\neq i}\sum_{\nu,\mu}{}' \xi_i^{\nu\mu}\xi_j^{\nu\mu}\xi_j^{sq}\xi_i^{sq}\right\rangle_i \\ &= \epsilon(N-1)(1+(t_s-1)c^2) \end{aligned}$$

and the square of the noise is given by

$$\begin{aligned}\Delta^2_{sq}(t) &= \langle (h_i(t)\xi_i^{sq})^2\rangle_i - S^2_{sq}(t)\\ &= \epsilon^2\Big\langle \sum_{j,j'\neq i}{\sum_{\nu,\mu}}'{\sum_{\nu',\mu'}}' \xi_i^{\nu\mu}\xi_i^{\nu'\mu'}\xi_j^{\nu\mu}\xi_{j'}^{\nu'\mu'}\xi_j^{sq}\xi_{j'}^{sq}\Big\rangle_i - S^2_{sq}(t)\\ &= \epsilon^2{\sum_{\nu,\mu}}'{\sum_{\nu',\mu'}}'\Big(\sum_{\substack{j,j'\neq i\\ j\neq j'}}{}' \langle \xi_i^{\nu\mu}\xi_i^{\nu'\mu'}\rangle\langle \xi_j^{\nu\mu}\xi_j^{sq}\rangle\langle \xi_{j'}^{\nu'\mu'}\xi_{j'}^{sq}\rangle\\ &\quad + \sum_{j\neq i}\langle \xi_i^{\nu\mu}\xi_i^{\nu'\mu'}\rangle\langle \xi_j^{\nu\mu}\xi_j^{sq}\rangle\Big) - S^2_{sq}(t)\\ &= \epsilon^2(N-1)\Big\{(N-2)[1+3(t_s-1)c^2+(t_s-1)(t_s-2)c^3]\\ &\quad +(s-1)(n_p+n_p(n_p-1)c^2)\\ &\quad +[t_s+t_s(t_s-1)c^2]\Big\} - \epsilon^2(N-1)^2[1+(t_s-1)c^2]^2.\end{aligned}$$

Thus

$$\begin{aligned}R_{sq}(t) = \sqrt{N-1}[1+(t_s-1)c^2]\{&(t_s-1)[c^2+(t_s-2)c^3-(t_s-1)c^4](N-1)\\ &+(t_s-1)[1+(t_s-3)c^2-(t_s-2)c^3]+(s-1)[n_p+n_p(n_p-1)c^2]\}^{-1/2}.\end{aligned}$$

The signal-to-noise ratio $R_s(t)$ of an ancestor pattern ξ_i^r can be calculated similarly as:

$$R_s(t) = \frac{\sqrt{N-1}\,t_s c}{[t_s c(1-c)(N-1)+t_s(1-c)+(s-1)(n_p+n_p(n_p-1)c^2)]^{1/2}}.$$

In the control experiment the correlation class from which the patterns are taken is changed every trial. Thus the signal-to-noise ratio in the control experiment $R^c_{sq}(t)$ is given by:

$$\begin{aligned}R^c_{sq}(t) = \sqrt{N-1}[1+(t_s-1)c^2]\Big(&(t_s-1)[c^2+(t_s-2)c^3-(t_s-1)c^4](N-1)\\ &+(t_s-1)[1+(t_s-3)c^2-(t_s-2)c^3]+\sum_{\substack{\nu=1\\ \nu\neq s}}^{n_c}[t_\nu+t_\nu(t_\nu-1)c^2]\Big)^{-1/2}\end{aligned}$$

where t_ν is the number of patterns learnt from the correlation class of ξ^ν.

Figure 9 shows the signal-to-noise ratio for each trial in the RPI experiment at the time of its recall. The signal-to-noise ratio is the same for the three patterns in a trial (as t_s is the same for each trial). Each trial is represented by one point to facilitate comparison between figures 8 and 9. The full line shows the signal-to-noise ratio $R_{sq}(t)$ for each pattern in the experiment (i.e. $\{\xi_i^{sq}, s=1,\ldots,4, q=1,\ldots,18\}$ *at the time it was recalled.* The broken line shows the signal-to-noise ratio $R_r(t)$ for the relevant ancestor pattern at each recall. The dotted line shows $R^c_{sq}(t)$ for

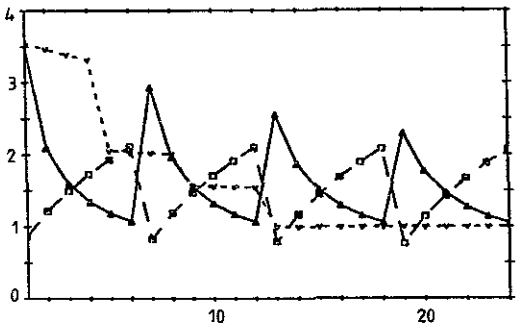

Figure 9. The signal-to-noise ratio for the patterns (full line) and ancestor pattern (broken line) in a trial The correlation class from which patterns were taken was changed every seventh trial (every trial for the control experiment—dotted line) Patterns within a class have correlation 0 20. $N = 700$.

the patterns in the control experiment—where the correlation class of the pattern is changed after every trial.

We can see that for the first trial (i.e. the first three patterns) the signal-to-noise ratio for a pattern is quite high; this will be well recalled. For the second trial the signal-to-noise ratio for a pattern is relatively low (although higher than for an ancestor pattern). By the third trial the ancestor pattern is as well stored as the individual patterns.

Looking at the simulations of the model in an RPI experiment (figure 8) we see that the recall overlap is good during the first trial from a class, much worse for the second trial and slightly better again for all subsequent trials from that class. However, the recall overlap for trials three to six in a class is $\approx \sqrt{c}\times$ (the overlap of patterns with their ancestor) indicating that the final state in each recall is the ancestor pattern (which we have verified). Thus the above analysis does seem to paint the correct qualitative picture for the behaviour of the model (one obvious difference is that the signal to noise ratio at the end of the experiment is the same for all patterns—it does not show the recency observed in simulations which is due to using bounded weights). It also highlights the need for a more explicit model of recall which can interpret the recall overlaps in a neural network in terms of a serial position curve (per cent correct versus temporal position).

The signal-to-noise ratio for patterns in a Von Restorff effect experiment (one pattern from one class in a list of patterns all from another class) also indicate that the one distinct pattern will be better recalled than the others. This was also seen in simulations of the model (see section 4.2).

When the learning rule of section 4.2 is used (i.e. using bounded weights) it is not so easy to calculate the signal to noise ratio. The weights $J_{i,j}(t)$ (and also the quantity $J_{i,j}(t)\xi_i^{sq}\xi_j^{sq}$) perform random walks on $[1,1]$, and the distribution $P(J_{i,j}(t))$ can be calculated† (see [29] for $P(J_{i,j})$ for uncorrelated patterns). However, for many

† The random walks of different $J_{i,j}$ will be correlated—taking any one $J_{i,j}(t)$ to represent $\langle J_{i,j}(t)\rangle_{i,j}$ is only valid in the limit of asymmetric 'dilution' (elimination of connections) where these correlations are negligible.

classes of correlated patterns the expressions get too cumbersome to be of use in calculating $\langle J_{i,j}(t)\xi_i^{sq}\xi_j^{sq}\rangle_s$ and $\langle (J_{i,j}(t)\xi_i^{sq}\xi_j^{sq})^2\rangle_s$.

6. Evaluation and discussion

The model of simple list learning in this paper produces serial recall curves which show four kinds of behaviour:

(a) permanent primacy or 'imprinting' for $x^* < 1$;
(b) primacy which disappears as the number of patterns increases for $x^* > 1$ and $p_{fp} \gg 0.05N$;
(c) primacy and recency where the primacy disappears as the number of patterns increases but the recency is unaffected for $x^* > 1$ and $p_{fp} \sim 0.05N$;
(d) recency only for $x^* > 1$ and $p_{fp} \ll 0.05N$.

The primacy in the model is a result of two things: having $\gamma > 1$ and setting the weights to be zero before the start of the experiment (i.e. starting with *tabula rasa*.) Recency is a natural consequence of using bounded weights as it is in the Hopfield–Parisi model. It is the interplay of these two features that makes the behaviour of the model so flexible.

The plausibility of beginning learning with *tabula rasa* has been questioned, see the discussion in [39]. The start of a learning experiment must be represented by some change in the model. Setting the initial state of the weights to be zero presumes some decay mechanism on a scale longer than the duration of the experiment. The presence of primacy seems to indicate such a decay mechanism although an alternative is indicated by the RPI experiment. The beginning of an experiment could be marked by a change in the type of input patterns.

The initial state of the weights also raises the question of whether the network should have any previous knowledge built in. To start with *tabula rasa* implies that either the subject's vocabulary is stored in a different system (affecting the pre- and post-processing assumed in our model) or that we are modelling the learning of previously unseen items. Learning experiments in lists of gobbledegook also show the usual bowed serial position curve [40].

6.1. Free recall and delayed free recall

Comparison of figures 1 and 7 shows that our model is consistent with the typical serial position curves from free recall experiments in psychology. Performance of a task before recall (which causes the disappearance of recency) is modelled by the network learning redundant patterns. However, this model is not entirely accurate because learning redundant patterns at the end of the presentation of the list also decreases performance (slightly) on the earlier part of the list. In the psychological data only the recency part of the serial position curve shows noticeable deterioration of recall.

Using redundant learning to model the disappearance of recency is more reminiscent of a two-system model explanation (where the most recent items are held in a short-term buffer) than an interference theory explanation. In our model the recency part of a serial position curve has a fixed capacity from which items are displaced; this is a possible interpretation of the idea of a short-term buffer in the two system model.

In more complex delay experiments we could not successfully model pauses by redundant activity, as we could for delayed free recall, see section 4.1. Since the model does not include time explicitly it cannot be used to model trace decay theories of memory or the effect of changing presentation rates. The modelling of presentation rates, delays between items and explicit rehearsal phenomena will require an active mechanism for rehearsal (taking $\gamma > 1$ to achieve primacy could be considered as passive cumulative rehearsal).

6.2. Inhomogeneous lists, RPI, and categorizaton

We attempted to model the learning of lists composed of different groups of items by using correlated patterns of activity. The results (section 4.2) show good agreement with the data on release from proactive inhibition experiments (and also for the von Restorff effect). The approximate signal-to-noise analysis in section 5 gives an adequate qualitative description of the behaviour of the model. That the network behaves in such a similar way to the experimental data suggests (i) that internal repesentations of presented items could be stored in an analogous way in the brain, and (ii) that internal representations of similar items are themselves similar. However, this model raises many questions of the pre- and post-processing of internal representations that we have assumed. We will address some of these below.

It should be noticed that learning patterns of activity from scratch when they are presented corresponds to the learning of previously unseen items rather than to the recognition of known items. The more complicated psychological experiments involving different types of item show the strongest effects where groups of words are differentiated by their meaning. Two similar (i.e. correlated) patterns of activity in our model correspond to items with similar internal representations. We have assumed only that similar items will have similar internal representations. We cannot address the question: in which way were the items similar?

The network often relaxes to the ancestor pattern of a category during recall. The interpretation of this purely in terms of a release from proactive inhibition experiments is difficult. We would like the output of the model to be actual patterns or mistakes, not categories (otherwise we must assume even more about the post-processing in recall: for example, we might assume that a pattern is picked at random from the category). This could perhaps be achieved with another layer of processing as described in [34].

A related problem is that of comparing the 'recall overlap' in the network with psychological data which shows the percentage of correctly recalled items. The likelihood of correctly recalling an item must depend on the how well its internal representation has been stored. However, an explicit model of recall, based on current psychological knowledge of recall, would be more satisfactory. In this respect the model of Sternberg fast scanning [6] is particularly instructive, although a different mechanism is required for the experiments described here.

What we have considered in this paper is a model of the crude features of list learning, ignoring the mechanisms for input and output of data and concentrating only on the storage of the internal representations of items. It is an interference model in that forgetting occurs due to the learning of other items. The mechanism for primacy ($\gamma > 1$ in equation (1)) is an implementation of passive cumulative rehearsal and the recency part of the serial position curves acts like a short-term buffer.

However, these models differ from the conceptual models of psychology in that their neurobiological implementation is conceivable. They are on a lower level of abstraction.

We feel that more detailed comparison of the mechanisms responsible for the behaviour of all the models mentioned in this paper with the conceptual models proposed in psychology would be profitable. It would be particularly interesting if models of memory got to the stage where more detailed implementation of psychological theories was possible.

Acknowledgments

We should like to thank Dr Graham Hitch for many useful discussions about psychology, and for critically reading this manuscript. We also thank Alan Bray for discussions concerning first passage time. JS was a postdoctoral research assistant supported by SERC during the time that much of this work was carried out. This financial support is gratefully acknowledged.

Appendix

In this appendix, the estimate for $p(x^*)$ which was used in the discussion in section 5.1 is derived. This is done by determining the number of patterns required to make the average second moment of the weights equal to x^{*2}.

The dynamical equation for a typical weight, J, ignoring the effect of the bound, is

$$J(t+1) = \gamma J(t) + \epsilon\eta(t).$$

This equation is similar to equation (1), except the effect of the bound is ignored, so there is no function f in the above. For simplicity, the indices i, j have been dropped and η is a random variable which takes values ± 1. The pattern stored at time t determines η.

The solution to this equation is

$$J(t) = \epsilon \sum_{i=0}^{t-1} \gamma^i \eta(t-i-1) + \gamma^t J(0).$$

Assuming *tabula rasa*, $J(0) = 0$, the average of $J(t)$ is zero, and the second moment is given by

$$\langle J(t)^2 \rangle = \epsilon^2 \sum_{i=0}^{t-1} \gamma^{2i}.$$

The sum is a simple geometric progression; the result is

$$\langle J(t)^2 \rangle = \epsilon^2 \frac{\gamma^{2t} - 1}{\gamma^2 - 1}.$$

The estimate for $p(x^*)$ is found by finding the number of patterns p which makes the second moment of J equal to x^*, i.e. solve for p in

$$\langle J(p)^2 \rangle = x^{*2}.$$

The solution is

$$p = \frac{\ln[2\gamma/(\gamma - 1)]}{2\ln(\gamma)}.$$

This is the estimate of $p(x^*)$ used.

References

[1] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc Natl. Acad. Sci. USA* **79** 2554–8

[2] Grossberg S 1969 On the serial learning of lists *Math. Biosci.* **4** 201–53

[3] Houghton G 1989 The problem of serial order: A neural network model of sequence learning and recall *Paper delivered at 2nd European Workshop on Language Generation* (Edinburgh 6–8 April 1989)

[4] Schreter Z and Pfeifer R 1989 Short-term memory/long-term memory interactions in connectionist simulations of psychological experiments on list learning *Neural Networks from Models to Applications* ed L Personnaz and G Dreyfus (Paris IDSET)

[5] Virasoro M A Categorization in neural networks and prosopagnosia *Preprint*

[6] Amit D J, Sagi D and Usher M 1990 Architecture of attractor neural networks performing cognitive fast scanning *Network* **1** 189–216

[7] Nadal J-P, Toulouse G, Changeux J P and Dehaene S 1987 Neural networks: learning and forgetting *Computer Simulations in Brain Science* ed R M J Cotterill (Cambridge: Cambridge University Press)

[8] Morris R G M (ed) 1989 *Parallel Distributed Processing: Implications for Psychology and Neurobiology*

[9] Atkinson R C and Shiffrin R M 1971 The control of short-term memory *Sci. Am.* **225** 82–90

[10] Baddeley A D 1986 *Your Memory A Users Guide* (London: Penguin)

[11] Baddeley A D 1976 *The Psychology of Memory* (New York: Basic Books)

[12] Hitch G J and Burgess N 1991 in preparation

[13] Slamecka N J 1960 Retroactive inhibition of connected discourse as a function of practise level *J. Exp. Psychol.* **59** 104–8

[14] Underwood B J 1957 Interference and forgetting *Psychol. Rev.* **64** 49–60

[15] Postkan L and Phillips L W 1965 Short-term temporal changes in free recall *J. Exp. Psychol.* **17** 135

[16] Wickens D D, Born D G and Allen C K 1963 Proactive inhibition and item similarity in short-term memory *J. Verbal Learning Verbal Behav* **2** 440–5

[17] Watkins O C 1975 The origin of the build-up of proactive inhibition effect *Paper presented at 1975 meeting of Experimental Psychological Society (London)*

[18] von Restorff H 1933 Uber die wirkung von bereichsbildungen im spurenfeld *Psychologisch Forschung* **18** 299–342

[19] Green R T 1958 Surprise, isolation and structural change as factors affecting recall of temporal series *British J. Psychol.* **49** 21–30

[20] Hinton and Plaut 1987 Using fast weights to deblur old memories *Proc Ninth Annual Conference of Cognitive Science Society* (Seattle WA, July 1987)

[21] Schneider W and Detweiler M 1988 A connectionist/control architecture for working memory *Psychology of Learning and Motivation* vol 21

[22] Amit D J 1989 *Modeling Brain Function: the World of Attractor Neural Networks* (Cambridge: Cambridge University Press)

[23] Gutfreund H, Amit D J and Sompolinsky H 1985 Storing infinite numbers of patterns in a spin-glass model of neural networks *Phys. Rev. Lett* **55** 1530–3

[24] van Hemmen J L, Keller G and Kuhn R 1988 Forgetful memories *Europhys. Lett.* **5** 663–8

[25] Nadal J-P, Toulouse G, Changeux J P and Dehaene S 1986 Networks of formal neurons and memory palimpsests *Europhys Lett.* **1** 535
[26] Mézard M, Nadal J-P and Toulouse G 1986 Solvable models of working memories *J Physique* **47** 1457–62
[27] Parisi G 1986 A memory which forgets *J. Phys A. Math. Gen.* **19** L617
[28] Geszti T and Pazmandi F 1987 Learning within bounds and dream sleep *J. Phys. A· Math. Gen.* **20** L1299–303
[29] Derrida B and Nadal J-P 1987 Learning and forgetting in asymmetric, diluted neural networks *J. Stat. Phys.* **49** 993–1009
[30] Peretto P 1987 *Computer Simulations in Brain Science* ed R M J Cotterill (Cambridge. Cambridge University Press)
[31] Wong K Y M, Kahn P E and Sherrington D 1991 A neural network model of working memory exhibiting primary and recency *J. Phys. A: Math. Gen.* **24** 1119–35
[32] Burgess N, Moore M A and Shapiro J L 1989 Human-like forgetting in neural network models of memory *Neural Networks and Spin Glasses* ed W K Theumann and R Koberle (Singapore World Scientific)
[33] Gardner E 1987 Maximum storage capacity in neural networks *Europhys. Lett.* **4** 481–5
[34] Gutfreund H 1988 Neural networks with hierarchically correlated patterns *Phys Rev* A **37** 570
[35] Bacci S, Mato G and Parga N 1989 The organization of metastable states in a neural network with hierarchical patterns *Int. J Neural Systems* **1** 69–76
[36] Posner M I and Keele S W 1968 On the genesis of abstract ideas *J. Exp Psychol.* **77** 353–63
[37] van Kampen N G 1981 *Stochastic Processes in Physics and Chemistry* (Amsterdam. North-Holland)
[38] Serra R, Andretta M, Zanarini G and Compiani M 1986 *Physics of Complex Systems* (Oxford Pergamon)
[39] Toulouse G, Dahaene S and Changeux J P 1986 Spin glass models of learning by selection *Proc Natl. Acad. Sci. USA* **83** 1695–8
[40] Raffel G 1936 Two determinants of the effects of primacy *Am. J Psychol.* **48** 654–7
[41] Loess H 1968 Short-term memory and item similarity *J. Verbal Learning Verbal Behav* **7** 80