TRACE FEATURE MAP: A MODEL OF EPISODIC ASSOCIATIVE MEMORY *†

Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin, Austin, TX 78712-1188
tel. (512) 471-9571, fax (512) 471-8885, email risto@cs.utexas.edu

Abstract

An approach to episodic associative memory is presented, which has several desirable properties as a human memory model. The design is based on topological feature map representation of data. An ordinary feature map is a classifier, mapping an input vector onto a topologically meaningful location on the map. A trace feature map, in addition, creates a memory trace on that location. The traces can be stored episodically in a single presentation, and retrieved with a partial cue. Nearby traces overlap, which results in plausible memory interference behavior. Performance degrades gracefully as the memory is overloaded. More recent traces are easier to recall as are traces that are unique in the memory.

1 Introduction

Neural network models of associative memory have been extensively studied in the last few decades (see e.g. Willshaw et al., 1969; Anderson, 1972; Kohonen, 1972, 1977, 1984; Cooper, 1973; Little and Shaw, 1975; Anderson et al., 1977; Hinton and Anderson, 1981, Hopfield, 1982, 1984; Grossberg, 1983; Knapp and Anderson, 1984; Ackley et al., 1985; McClelland and Rumelhart, 1986; Kanerva, 1988). These models are motivated by the properties of human associative memory, which are very different from those of the usual digital computer memory. Instead of accessing memory with specific addresses, associative memories are content-addressable. Items (binary or gray-scale pattern vectors) can be retrieved with partial or approximate cue patterns. Several patterns can be stored on the same physical hardware, and similar patterns interfere with each other. Performance degrades gracefully when the memory is overloaded.

Many associative memory models were developed as mathematical abstractions of human memory. The goal was to understand the mathematics of associative memory, rather than to build psychologically plausible models. In a cognitive model of associative memory, (1) storage and retrieval should be based on local processing, rather than on a global computation of the weight matrix. (2) It should be possible to store items episodically, in only a single presentation, rather than iterating through all items multiple times. (3) Distributed representations, which are continuously valued, non-orthogonal and not necessarily linearly independent, should be supported, because several interesting cognitive phenomena has been shown to emerge from such representation style (Hinton et al., 1986; van Gelder,

---

*This research was supported in part by an ITA Foundation grant and by fellowships from the Academy of Finland, the Emil Aaltonen Foundation and the Foundation for the Advancement of Technology (Finland) when the author was at UCLA.

†To appear in *Biological Cybernetics*, 1991

1989). Many associative memory models are restricted to binary vectors, and some require the vectors to be linearly independent or orthogonal. (4) Memory interference effects and effects of damage should be psychologically plausible. Many models exhibit undifferentiated decrease in performance, rather than effects of recency, primacy, uniqueness, categorization, proactive and retroactive transfer etc.

Auto-associative matrix memories are the simplest class of associative memory, and theoretically fairly well understood (Kohonen, 1972, 1977, 1984; Hopfield 1982, 1984). In these models, N-dimensional vectors are represented on N units which are fully connected through an $N \times N$ weight matrix. The weights are assigned so that the vectors become attractors for the dynamical system. In some special cases, e.g. with linearly independent binary patterns, it is possible to determine the appropriate weight values "off-line" with a non-local algorithm (Kohonen, 1984). Adding new patterns requires recomputing the weight matrix. The Hopfield network (Hopfield, 1982, 1984) is perhaps the best-known example of this type of associative memory.

There also exist associative memory models where storage and retrieval takes place through local computations. The patterns are stored by cycling through the set of all patterns multiple times and making small adjustments on the network weights at each presentation. Eventually the weights converge to a configuration where the network makes the correct associations for all patterns in the training set, and possibly generalizes to new but similar patterns. Backpropagation networks (Rumelhart et al., 1986) and the Boltzman machine (Hinton et al., 1984; Ackley et al., 1985), which make use of "hidden units" (extra units that are not part of the item representation), and the Brain-State-In-A-Box model (Anderson et al., 1977; Anderson, 1986) are examples of this approach.

Note that all patterns to be stored need to be known in advance in the local learning models. Storing patterns in a single presentation is possible only when the patterns are orthogonal. In this case there is no crosstalk between patterns, and the correct associations can be established in a single weight modification. If the patterns are not orthogonal, each new presentation alters the traces of the previous patterns, and early traces are gradually erased from the memory. Several iterative presentations with very small weight adjustments are necessary for successful learning.

Episodic storage of non-orthogonal patterns is infeasible in the local learning models because they distribute the trace of each item over the whole memory hardware, and crosstalk in general cannot be avoided. The solution is to use different hardware to store different memories. This idea is employed in e.g. Read, Nenov and Halgren's binary auto-associative memory, which is based on the Gardner-Medwin model of the hippocampal formation (Read et al., in press). Each vector is encoded by approximately 10% of the neurons, randomly distributed in the network. Also, in Kanerva's sparse distributed memory for binary vectors (Kanerva, 1988), each vector is stored at those neurons whose addresses are within a certain radius of the input vector. Different neurons participate in storing different vectors.

Trace feature map is a new approach to auto-associative memory which aims at plausible modeling of human memory. The space of possible items to be stored is laid out on a topological feature map (Kohonen, 1984). Each unit on the map stands for a particular input item, i.e. the input space is represented using value-unit encoding (Ballard, 1986). Which items are currently stored in the memory is indicated by the lateral (recurrent) weights of the network.

The main features of the trace feature map model are: (1) The memory traces are created on a spatially localized area of the hardware. (2) The traces are created in a single presentation, without knowledge of future items to be stored and without re-activating the traces of the previous items. (3) Interaction of nearby traces results in plausible memory interference behavior: more recent traces are easier to recall, and unique traces are preserved.

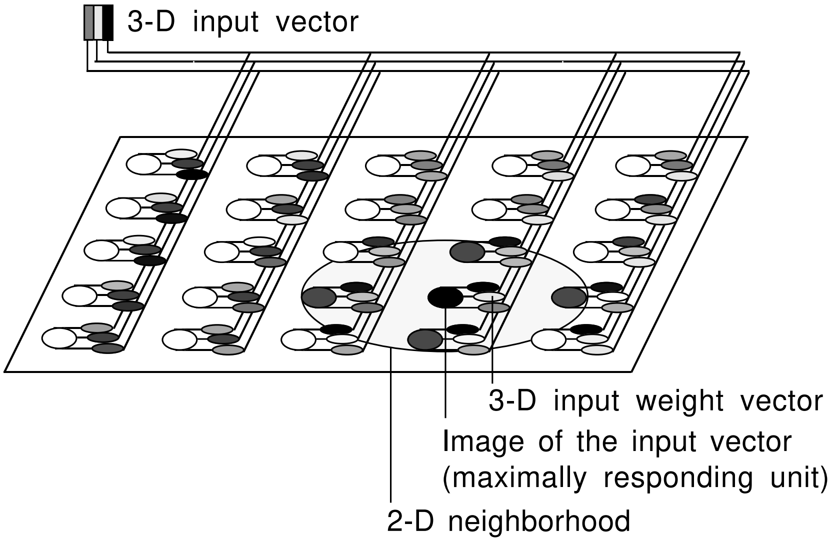

Figure 1: **A topological feature map network.** This network implements a mapping from a 3-dimensional input space onto a 2-dimensional location in the network. The values of the input components, weights and the unit output are indicated by gray-scale coding.

(4) Local damage to the network results in category-specific memory loss. (5) Storage and retrieval takes place through local computations. (6) The input vectors can be continuously valued and linearly dependent. (7) The memory is limited within the space represented by the feature map, i.e. it is a memory of occurrences of familiar items.

In the following, the basic properties of topological feature maps are first reviewed, and the trace feature map architecture is introduced as an extension to ordinary feature maps. The storage and retrieval mechanisms are described, followed by discussion of memory capacity and memory effects. An example application, episodic story memory, is briefly described. Discussion of some of the implications, open issues and directions for future research concludes the article.

2 Topological feature maps

2.1 General mechanisms

A 2-D topological feature map (Kohonen, 1984) implements a topology-preserving mapping from a high-dimensional input space onto a 2-D output space. The map consists of an array of processing units, each with N weight parameters (figure 1). The map takes an N-dimensional vector as its input, and produces a localized pattern of activity as its output. In other words, an input vector is mapped onto a location on the map.

Each processing unit receives the same input vector, and produces one output value. The response is proportional to the similarity of the input vector and the unit's weight vector. The unit with the largest output value constitutes the image of the input vector on the map. The weight vectors are ordered in such a way that the output activity smoothly decreases with the distance from the image unit, forming a localized response.

The weight vectors approximate specific items of the input space in such a way that topological relations are retained. This means roughly that nearby vectors in the input space are mapped onto nearby units on the map. This is a very useful property, since the complex similarity relationships of the high-dimensional input space become visible on the map.

The organization of the map, i.e. the assignment of the weight vectors, is formed in an unsupervised learning process. Input items are randomly drawn from the input distribution and presented to the network one at a time. The map responds to each vector by developing a localized activity pattern. The weight vector of the maximally responding unit and each unit in its neighborhood are changed towards the input vector. These units now produce an even stronger response to the same input. In the process, the weight vectors become better approximations of the input vector distribution and neighboring vectors become more parallel, which over time results in global order.

The size of the weight change neighborhood and the gain of the weight change decrease with time, allowing the map to make finer and finer distinctions between items. Eventually, the distribution of the weight vectors becomes an approximation of the input vector distribution. This means that more weight vectors are allocated to dense areas of the input space, i.e. these areas are magnified (represented to greater detail) on the map. The two dimensions of the map do not necessarily stand for any recognizable features of the input space. The dimensions develop automatically to facilitate best discrimination between input items.

Each adaptation step consists of three computational tasks: (1) computing the initial response of each unit to the external input by measuring the similarity of the input vector and the unit's weight vector, (2) determining the adapting neighborhood by focusing the initial response of the map to the neighborhood of the maximally responding unit, and (3) changing the weights in this neighborhood. The following discussion concentrates on generating and focusing the response, because these processes are central in the operation of trace feature maps. The weight adaptation occurs only during self-organization, which is assumed to have taken place before the operation of trace feature maps. For more details on the self-organization process, see e.g. (Kohonen, 1982ab, 1984, 1990; Ritter and Schulten, 1988; Miikkulainen, 1991).

2.2 Generating and focusing the response

In a biologically plausible implementation of topological feature maps (Kohonen, 1982b; Miikkulainen, 1991) the similarity is measured by a scalar product of the input vector and the weight vector, i.e. by computing a weighted sum of the input components. This approach is cumbersome, because it requires that both the input vectors and the weight vectors are normalized (Miikkulainen, 1991). Normalization can be efficiently abstracted by using Euclidian distance as the similarity measure (Kohonen, 1984). The distance between the input vector and the weight vector is scaled between 0 and 1 and negated so that the value 1 indicates maximum similarity:

$$s_{ij} = 1 - \frac{\|x - m_{ij}\|}{d_{max}}, \tag{1}$$

where x is the external input vector, m_{ij} is the weight vector of unit (i, j) (in a 2-dimensional map), d_{max} is the maximum distance of two vectors in the input space (e.g. $\sqrt{2}$ in the 2-D unit square) and s_{ij} stands for the unit's similarity value. The initial response η_{ij} of unit (i, j) to an external input vector is then

$$\eta_{ij} = \sigma(s_{ij}), \tag{2}$$

where σ is the familiar sigmoid activation function of the type

$$\sigma(z) = \frac{1}{1 + e^{\beta(\delta - z)}}. \tag{3}$$

The parameter β determines the slope of the sigmoid and δ its displacement from the origin. The sigmoid introduces a nonlinearity (a soft threshold) into the reponse, and limits its output within the range $[0, 1]$.

The initial response can be focused through lateral inhibition. Each unit on the feature map receives activation from its neighbors through lateral connection weights. During self-organization, these weights are fixed in the "Mexican hat" pattern, i.e. the connections from the closest units are excitatory and from the units further away are inhibitory. The response of the network evolves over time according to

$$\eta_{ij}(t) = \sigma \left(s_{ij} + \sum_{k,l} \gamma_{kl,ij} \eta_{kl}(t - \Delta t) \right), \tag{4}$$

where $\gamma_{kl,ij}$ is the lateral connection weight on the connection from unit (k, l) to unit (i, j), and $\eta_{kl}(t - \Delta t)$ is the activity of unit (k, l) during the previous time step. The primary effect of lateral inhibition is to sharpen the contrast between the high and low activity areas. If the diameter of the lateral inhibition mask is comparable to the diameter of the initial response of the network, in successive iterations of equation 4 the response becomes more focused around the maximally responding unit. The more lateral inhibition there is compared to lateral excitation, the narrower is the final stable response. Self-organizing weight changes then take place withing the final stable activity pattern.

2.3 Feature maps as memory models

Feature maps have several potentially useful properties for modeling memory. The main reason is that both distributed and localist features (Feldman and Ballard, 1982; Hinton et al., 1986; van Gelder, 1989) are combined in the feature map representation. Distributed representations for the input items are stored in the input weights of the feature map units. In addition, each unit is a localist representation for an input item, and the spatial arrangement of the units corresponds to the topological relations of the items. Some of the properties that result include:

(1) The classification performed by a feature map is based on a large number of parameters (the input weight components), making it very robust. Incomplete and somewhat noisy representations can usually be correctly recognized.

(2) Once an inexact input item is recognized, it is possible to recover its exact representation from the weights of the image unit. In other words, categorical perception can be modeled (Miikkulainen, 1990a).

(3) The map tends to be continuous. It contains many intermediate units which do not stand for any particular input item, but represent combinations of items. This means that in some cases it is possible to recover a blend of two items.

(4) Several items can be active on the map at the same time, i.e. different alternatives can be represented distinctly and in parallel. Associations between different items can be implemented through lateral connections (Kohonen and Mäkisara, 1986). With connections between different maps, many-to-many mappings are possible (Miikkulainen, 1990b). Note that e.g. backpropagation networks can represent ambiguity only by blending the possible alternatives.

(5) Because items are stored in different parts of the map, episodic storage is possible. Storing new traces does not necessarily affect all other traces on the map. Nearby traces, which represent similar items, are more likely to be affected.

(6) Differences between the most frequent input items are magnified spatially in the mapping, i.e. the variations of the most common inputs are more finely discriminated.

(7) The self-organizing process requires no supervision and makes no assumptions about the data. The properties that best distinguish between input items are determined automatically, and may be very different for different kinds of items.

3 Trace feature maps

3.1 Definition

An ordinary feature map is a classifier, mapping an input vector onto a location on the map. A trace feature map, in addition, creates a memory trace on that location. The map remembers that at some point it received an input item that was classified there. The traces can be stored one at a time, as items are read in, and retrieved with a partial cue. In the following, the basic trace feature map mechanism is presented and its properties are illustrated and analyzed using uniformly distributed 2-dimensional input data.

In the biological model of self-organization, lateral connections between units are responsible for neighborhood selection (section 2.2). The response concentrates around the maximally responding unit, and the weight changes occur in this area. Kohonen and Mäkisara (1986) suggested that lateral connections could also be responsible for associations between items. Trace feature maps are based on a similar idea: after the map has become ordered, *the lateral connections are used to store episodic memory traces.*

The output η_{ij} of unit (i,j) on a trace feature map is based on equations 1, 3 and 4:

$$\eta_{ij}(t) = \sigma\left((1-\theta)\left(1-\frac{\|x-m_{ij}\|}{d_{max}}\right)+\theta\sum_{k,l}\gamma_{kl,ij}\eta_{kl}(t-\Delta t)\right), \qquad (5)$$

where (as before) x is the external input vector, m_{ij} is the unit's weight vector, d_{max} is the maximum distance of two vectors in the input space, $\gamma_{kl,ij}$ is the lateral connection weight on the connection from unit (k,l) to unit (i,j), and $\eta_{kl}(t-\Delta t)$ is the output of unit (k,l) during the previous time step. In other words, each unit computes a weighted sum of its lateral activity, adds the activity resulting from the external input, and develops an output activity which is a sigmoid of the sum. The parameter θ determines the balance of external and lateral activation, and it is used to separate the storage and retrieval processes.

During self-organization, $\theta > 0$, and the lateral connections implement lateral inhibition. Excitation decreases gradually while inhibition increases, making the weight change neighborhoods smaller. At the same time, θ also decreases gradually, making the response more sensitive to the external input as the map becomes more ordered (Miikkulainen, 1991). At the limit, the weight vectors form a topological map of the input space, $\theta = 0$ and all lateral connections are inhibitory. This is the ideal initial configuration for the operation of trace feature maps.

However, the trace feature map mechanism is independent of how the map is formed. The map may be produced by a self-organizing process where settling through lateral connections is explicitly modeled, or by a process where settling is replaced by a computational abstraction (which is the usual practical implementation of feature maps, see e.g. Kohonen (1984)). Alternatively, the map may be formed by simply assigning ordered weights to the feature map units to begin with (as is done below).

3.2 Storage mechanism

Let us assume that we have an ordered 2-D feature map of a uniform distribution on the unit square. The input vectors in this case are 2-dimensional, with each component uniformly

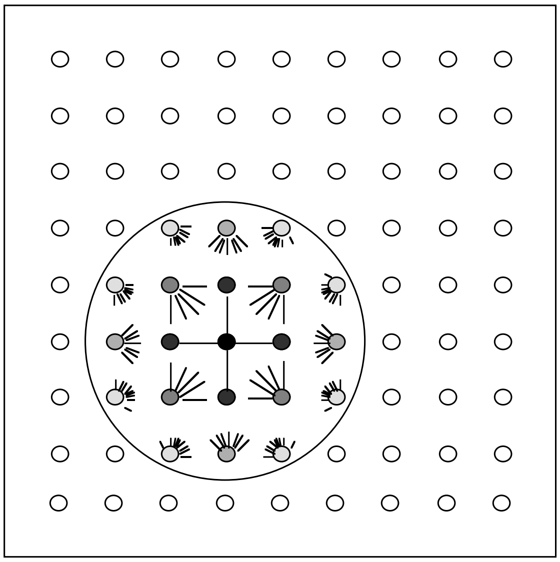

Figure 2: **Creating a trace.** The map lays out a uniform distribution on the unit square. The coordinates of each unit correspond to the two components of its weight vector. Gray-scale coding indicates the responses to the external input [0.4, 0.4], with $\theta = 0.0, \beta = 10.0, \delta = 1.4$. Line segments indicate excitatory lateral weights emanating from each unit. The lateral weight parameters were $\gamma_E = 2.0, \gamma_I = -0.5$. Inhibitory lateral connections are not shown.

distributed within [0, 1]. The weight vectors form a regular grid on the unit square, and each unit is responsible for an approximately equal area of the input space (figure 2).

Let us further assume that the lateral connections of the map are all inhibitory with $\gamma_{kl,ij} = \gamma_I$ (a negative constant). This means that the map is blank, i.e. contains no traces. During storage, $\theta = 0$ so that the response of the map depends only on the external input activity.

When an input vector is presented to this map, it responds by developing a localized symmetric activity "bubble" around the unit whose weight vector is closest to the input vector (figure 2). The diameter and the intensity of the bubble depend on the sigmoid parameters.

A trace is created by modifying the lateral connections of the active units [1]. For each unit in the bubble, a connection to a unit with a higher activity becomes excitatory, while a connection to a unit with a lower activity becomes inhibitory, both proportional to the activity level of the presynaptic unit:

$$\gamma_{ij,kl} = \begin{cases} \gamma_E \, \eta_{ij} & \text{if } \eta_{ij} < \eta_{kl} \text{ or } (i,j) \equiv (k,l) \\ \gamma_I \, \eta_{ij} & \text{if } \eta_{ij} \geq \eta_{kl} \end{cases} \tag{6}$$

where $\gamma_E > 0$ and $\gamma_I < 0$ are the inhibition and excitation strength parameters. The units within the response are now "pointing" towards the unit with the highest activity in the bubble (figure 2).

Note that the trace is created *in a single presentation*[2], and it is not necessary to know what has already been stored in the memory and what additional items need to be stored later.

---

[1] Since $\sigma(z) > 0$, a unit is considered active if its output $\eta > \eta_a$, where η_a is a suitable threshold value, e.g. 0.1

[2] The weight change in reality would be a gradual process. We assume that the input is present long enough so that the weights have time to reach their stable values given in equation 6.

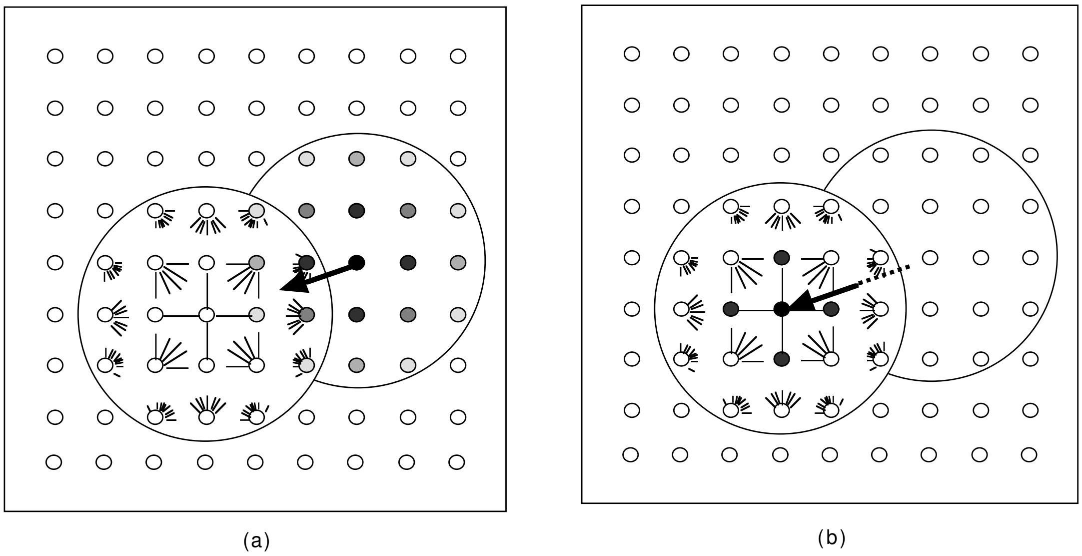

Figure 3: **Retrieval from a trace feature map with a partial cue.** The lateral connections pull the initial response to vector [0.7, 0.5] (a) to the center of the trace [0.4, 0.4] (b). Parameter settings were otherwise the same as in figure 2, except $\theta = 0.5$ during retrieval.

3.3 Retrieval mechanism

During retrieval, θ is positive, e.g. 0.5. The response of the map now depends both on the external input vector and the lateral connections encoding the traces.

A stored vector is retrieved by presenting an approximation of the vector to the map. The initial response is again a localized activity pattern (figure 3a). Because the map is topological, the center of this pattern is likely to be somewhere near the target trace. If the cue vector is close enough to the target, the initial response overlaps with the trace. In the next few settling iterations, the excitatory lateral connections within the trace pull the activity towards the center of the trace. The activity settles around the center, and the external input weights of the unit with the highest activity give the stored vector (figure 3b).

If the cue vector is too far from the target, the initial response does not overlap with the trace. The lateral connections of the units within the initial response are all inhibitory, and in the next step, all activity is turned off. The next step is then again the initial response. In subsequent steps, the activity oscillates between nonactivity and the initial response. This oscillation indicates that there is no appropriate trace in the memory.

Notice that the retrieval process makes no distinction between incomplete, noisy and partly incorrect cues. Any pattern can be used to cue the memory, and if there is a trace close enough to the cue, it will be returned. As a result, minor errors in the cue can be automatically corrected.

3.4 Memory effects and capacity

The trace feature map exhibits interesting memory effects which result from interactions between traces. When two traces are stored close to each other, the later trace steals units from the older one (figure 4). When a cue is mapped onto the general neighborhood of the two traces, the later one is likely to receive more lateral activation, which allows it to inhibit the older trace and eventually turn it off. This results in a recency effect: if several similar items are stored on the same map, the later traces are more likely to be retrieved.

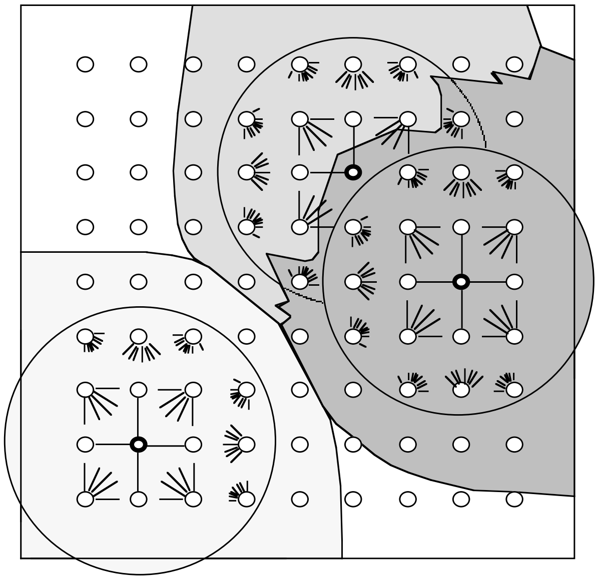

Figure 4: **Interaction of nearby traces.** Traces are indicated by circles as before. The trace on the right has partially obscured an earlier trace. The three shaded areas indicate the attractor basins for each trace. These areas were determined by systematically presenting input vectors from a 200 × 200 grid and observing which trace (if any) was retrieved. The weight space and the input space are superimposed in this figure. For example, any vector from the bottom left area of the input space will retrieve the bottom left trace. The parameter settings were the same as in figures 2 and 3.

As the memory becomes overloaded, older traces become increasingly hard to retrieve, and eventually they may be completely replaced by newer traces. However, this process is selective in that traces in a sparse area of the map are not affected, no matter how old they are. In other words, unique items are preserved.

Figure 4 depicts the attractor basins for each of the three traces stored on a map. The two topmost traces are located in the same area, and the newer of them has partially obscured the older one. The third trace is located in the lower left hand corner by itself and it is not affected by the newer traces.

It is difficult to characterize the memory capacity of trace feature maps. Memory effects are an important part of the model, but they greatly complicate analysis. The diameter of the trace is obviously an important factor. The narrower the traces, the more of them will fit on the map without obscuring each other. On the other hand, they become harder to retrieve because more accurate cues are required.

Instead of attempting to estimate how many traces of a given diameter can be stored on a particular trace feature map, it makes more sense to outline the behavior of the memory as it becomes increasingly loaded with traces. Figure 5 depicts a few such performance descriptors as a function of the number of traces stored in the memory.

There is a 15% chance that a random vector will retrieve a single solitary trace, i.e. on the average, the basin covers 15% of the map in this experiment (the plot labeled “Nearest” in figure 5). As more traces are stored on the same map, the basins become smaller. With six traces, each one has only 7.5% chance of being retrieved with a vector that is nearest to them. At this point, the percentage of “Nearest” begins to level off, indicating that the additional traces are mostly stealing units from the older ones.

Even when there are only two traces on the map, there is an 8% chance that the later one completely wipes out the earlier one (“Lost” plot). The percentage of lost traces grows

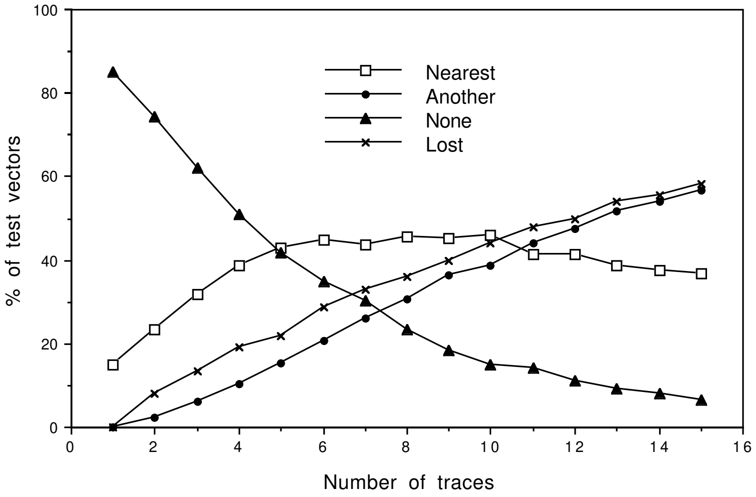

Figure 5: **Performance descriptors with increasing number of traces.** As before, the map consisted of 9 × 9 units and the diameter of the trace was about 0.5. The sigmoid and lateral interaction parameters were $\beta = 15.0, \delta = 1.4, \gamma_E = 0.5, \gamma_I = -0.075$. Each percentage represents an average of 100 trials. In each trial, the traces were uniformly distributed on the unit square. Test vectors were laid out on a 20 × 20 grid that uniformly covered the unit square. The test vectors were presented to the map one at a time, and depending on which trace they retrieved (if any), they were classified as "Nearest", "Another" or "None". The fourth possibility, settling onto a unit that is not a center of any trace, only occurred in about 0.1% of the trials, and was not plotted in the figure. If a trace was not retrieved at all during the trial, it was counted as a "Lost" trace. Typical standard deviations for each descriptor were: Nearest 12, Another 6, None 10, Lost 14.

approximately linearly as more traces are stored on the same map. With eight traces, 1/3 of them are inaccessible. This is surprisingly little, knowing that a single trace not clipped by the boundaries of the map covers about 1/4 of the map in this experiment.

The plot labeled "Another" demonstrates recency effect[3]. The number of test vectors that will retrieve another trace grows approximately linearly with the number of traces stored. As the memory becomes overloaded, the oldest traces are gradually replaced by newer ones. It is necessary to specify more and more accurate cues to retrieve old traces, and eventually they become inaccessible. With eleven traces on the same map, a random cue is as likely to retrieve a later trace than the nearest one (44% vs. 42%). At that point, 47% of the traces have become inaccessible altogether.

The main conclusion from this experiment is that the trace feature map memory degrades very gracefully when it is overloaded. On the other hand, memory effects are possible even under very light load. Statistically the behavior is very predictable, but individual cases vary significantly. The standard deviations of the above percentages were typically greater than 10.

4 Example: Episodic memory for stories

Trace feature maps have been used in episodic memory for DISCERN, a neural network model of script-based story understanding (Miikkulainen, 1990a). DISCERN reads short narratives about stereotypical event sequences (such as visiting a fancy restaurant or trav-

---

[3] When some other trace as the nearest one is retrieved, it is generally because it was stored more recently than the nearest trace.

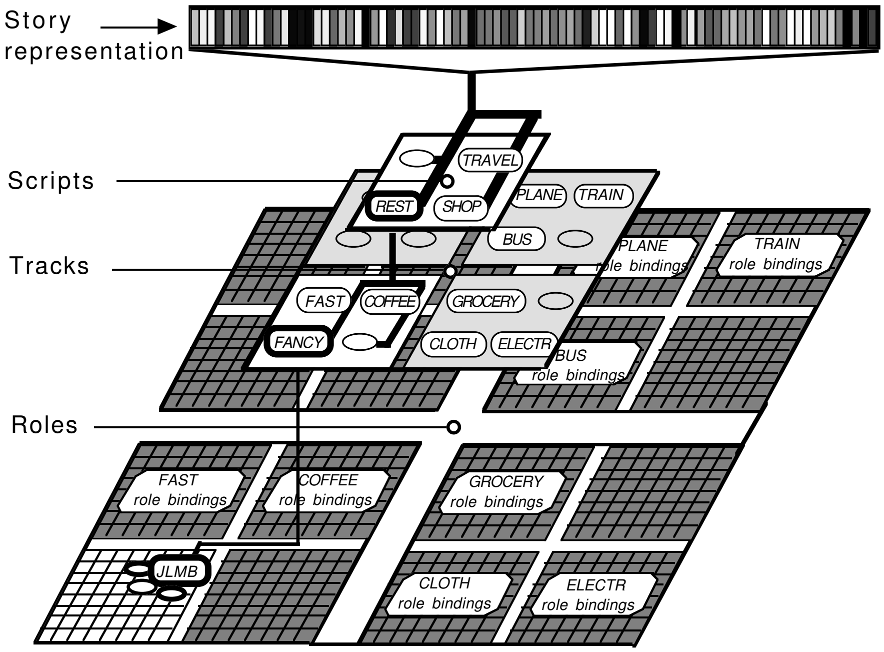

Figure 6: **Episodic memory for script-based stories.** The labels at the top and middle levels indicate the maximally responding unit for stories based on the different scripts and tracks. The labels at the bottom level indicate the role-binding map for each track. This particular input story representation is classified as an instance of the restaurant script and fancy-restaurant track, and the role bindings are found to be customer=`John`, food=`lobster`, restaurant=`MaMaison`, tip=`big` (i.e. `JLMB`). The trace is created around the role binding unit labeled `JLMB` in the fancy-restaurant role-binding map (lower left corner).

eling by airplane), stores them in episodic memory, generates fully expanded paraphrases of the narratives, and answers questions about them.

The episodic memory in DISCERN is a pyramid of feature maps, organized according to the hierarchical taxonomy of script-based stories (figure 6). The highest level of the hierarchy is a single feature map that lays out the different script classes. Beneath each unit of this map there is another feature map that lays out the tracks (established variations) of the script. At the bottom level, the different role bindings of each track are separated.

The script taxonomy is extracted from examples of story representations. The pyramid structure itself is predetermined and fixed, but the maps are self-organized one level at a time from top to bottom. The top- and middle-level maps act as filters, (1) choosing the relevant input items for each lower-level map and (2) compressing the representation of these items to the most relevant components (i.e. components that vary the most) before passing them down to the lower-level map for a more detailed mapping.

The map hierarchy receives a distributed representation of the story as its input and classifies it as an instance of a particular script, track and role binding. The maximally responding units at the script-, track- and role-binding levels provide a unique memory representation for each story. However, the role-binding unit alone uniquely identifies the story, and a trace needs to be created only at the bottom level. The script and the track level are ordinary feature maps, while the role-binding level consists of trace feature maps.

When a representation is stored in the episodic memory, the map hierarchy determines the appropriate role-binding map and the location on that map. The trace feature map mechanism creates a memory trace at that location. A story is retrieved from the memory by giving it a partial story representation as a cue. Unless the cue is highly deficient, the map hierarchy is able to recognize it as an instance of the correct script and the track, and form a partial cue for the role-binding map. The trace feature map then completes the role binding. The complete story representation is retrieved from the weight vectors of the maximally responding units at the script-, track-, and role-binding levels.

In DISCERN, the trace feature maps have to operate in less than ideal conditions. The role-binding maps lay out high-dimensional spaces on a small number of units, and the maps have more structure and less continuity than in the ideal case discussed in previous sections. Similar vectors may sometimes be mapped on different areas of the map. While the basic mechanisms remain the same, they need to be parameterized slightly differently to guarantee robust operation in these conditions (see (Miikkulainen, 1990a) for details; the effect of different parameters on performance are discussed in the appendix).

5 Discussion

5.1 Issues in feature map representation

The trace feature map model makes a number of predictions about human knowledge organization and the mechanisms underlying knowledge processing. Perhaps most important is the idea of laying out memory traces spatially on maps. This is in line with the general idea of brain representation by value-unit encoding, as proposed by Barlow (1972) and Ballard (1986). Several topological maps, including retinotopic, tonotopic and also tactile, motor and spatial maps are known to exist in the central nervous system (Konishi, 1986; Knudsen et al., 1987). It is quite possible that higher-level information is also represented in a similar manner. For example, it has been shown (through simulation), that it is possible to form maps of phonemes (Kohonen et al., 1984) and word semantics (Ritter and Kohonen, 1989; Miikkulainen, 1990b) from simple contextual information. Neurophysiological evidence for such phonotopic and semantotopic maps is still to be found.

The operation of the trace feature map reflects the physical organization of the hardware. We have seen how high-level phenomena such as recency preference for similar memory traces and persistence of unique traces can be explained by the spatial layout of the memory. The model can be lesioned, and local damage results in loss of specific types of traces. Such category-specific impairments have been observed in the semantic memory of aphasic patients (Warrington, 1975; Warrington and Shallice, 1984; Warrington and McCarthy, 1987). It remains to be seen whether similar impairments can also occur in the episodic memory. The model also predicts that it is not possible to selectively loose all traces of a particular time period, provided the traces have been fully consolidated in the episodic memory.

Representation of the input space on feature maps seems to suffer from combinatorial explosion. The number of units on the map determines how many items the system can tell apart. If k is the number of input dimensions and N is the number of different values in each dimension, N^k units are needed to represent the space of all possible combinations. This "N^k problem" is a general limitation of the value-unit encoding approach (Ballard, 1986).

A plausible solution, which also seems to be in use in the value-unit maps in the central nervous system, is to divide complex spaces into several lower dimensional ones, and combine them hierarchically (Ballard, 1986). None of the individual feature maps would ever need to map more than a few dimensions. If there are many dimensions of variation, these could be split into separate maps. The representation would consists of image units in each and every submap, i.e. it would be distributed and compositional. However, it is not clear how multiple items can be represented distinctly in parallel in such an architecture (von der Malsburg, 1987).

A complex feature map representation could thus contain two kinds of hierarchical relations: (1) a higher-level map forms a categorization of its input space and passes the input items down to the appropriate submaps for more accurate mapping of each category (as in the episodic memory of DISCERN and in (Miikkulainen, 1990c)), and (2) the whole input space of the parent map is laid out on the lower level maps, but different dimensions are represented on different maps. The representation of an item in such a hierarchical system would be an AND-OR tree.

Currently, the number of items that can be represented on a single feature map is limited by the number of units. Resolution could be improved by taking the response pattern as a whole into account. Instead of using the input weights of the maximally responding unit as the representation, the weight vectors of all active units could be combined, proportional to their activity. Items between units on the map would be represented as linear combinations of the existing unit weights.

5.2 Limitations of the model

There are two major issues that the model does not address. First, it assumes that the whole space of possible input items is represented on the map. Truly novel items cannot be stored correctly, because the system has no mechanism for representing traces that do not fit the memory organization.

Similar dissociation of novel and familiar experience has been observed in hippocampal amnesia. Amnesic patients often cannot explicitly recall individual episodes, although these episodes cause normal priming effects (Warrington and Weiskrantz, 1974; Tulving and Schacter, 1990). However, priming is limited to pre-existing representations. The patients can form implicit memory traces of already familiar elements, but they cannot form traces that combine previously unrelated semantic elements (Shimamura, 1986; Halgren, 1984).

A possible explanation is that two dissociable memory encoding processes exist. Encoding novel integrative traces requires hippocampal activity, whereas recording occurrences of familiar items is based on other cortical regions (Shimamura, 1986; Halgren, 1990). Trace feature maps can be seen as a model for the latter process.

People often exhibit better recall not only for the most recent experience, but also for the first experience of a particular kind. In the trace feature map framework, the first experience would be coded as a novelty through the hippocampal processes, and therefore it could be more prominent in the memory than a mere map representation. Subsequent experiences would be familiar and recorded only on the (updated) map. Since all traces on the map are equal (except for the recency effect), the primacy effect would be a result of the hippocampal encoding process only. Interestingly, this is again in line with observations on amnesic patients, who exhibit recency effect but no primacy effect (Baddeley, 1982).

The second open issue concerns the "the beginning of experience" in trace feature maps. The model currently makes a sharp distinction between self-organization and operation as memory, although it seems that memory mechanisms should be active during the self-organization also. However, even if traces were stored during self-organization, they would become inaccessible later as the memory organization evolves. Only traces that were created after the organization settled could be reliably recalled. In that sense the two-phase model can be seen as an approximation of the actual memory organization process.

5.3 Extensions

The global parameter θ is used to select between storage and retrieval processes. When $\theta = 0$, the lateral connections of the map are inactive, and the map develops a response according to the external input activity. This response is coded into the lateral connections as the memory trace. When $\theta > 0$, the lateral connections are active, and the map settles into a stored trace. Lateral connections are not modified during retrieval.

Interesting effects would result from relaxing this simple two-mode operation. If the lateral connections were active during storing an item, they would affect the trace that is formed. Items that are novel, i.e. in a sparse area of the map, would be stored as before. Items close to existing traces would be pulled closer to the traces. In effect, the input pattern would be perceived as more similar to the existing traces than it actually is. Contents of the memory would affect how new items are remembered. This kind of interaction would be consistent with proactive interference in human memory (Postman, 1971; Thorndyke and Hayes-Roth, 1979).

On the other hand, it would also be possible to modify the lateral weights during recall. After the activity has settled, the lateral connections could adapt to the final activity pattern. If the initial activity was emphasized in the settling process (with small θ), the final pattern would reflect the initial activity more than the lateral trace connections, and the trace would be changed towards the cue. This would have the effect of cues modifying the memory, which has been shown to happen e.g. in eyewitness testimony with leading questions (Loftus, 1975; Loftus et al., 1978).

Settling times give interesting insight into the recall process. Settling usually takes a few iterations longer when the cue is far away on the map. When there are two traces about equidistant from the cue, convergence can take very long. In other words, retrieval with an ambiguous cue is slower than retrieval with an unambiguous one, which behavior has also been observed in short-term memory recognition tasks (Baddeley, 1976). The settling times provide information that could be used by a high-level monitor process to decide on the validity of the recalled trace.

Usually one of the ambiguous traces wins after a number of iterations, turning the other one off completely. However, it is also possible that both traces remain active in the stable pattern, indicating ambiguous retrieval. Our implementation currently retrieves the one with the strongest activity, but it would also be possible to retrieve an average of the two traces. This strategy would model retroactive memory confusions where a blend of two traces is retrieved, e.g. `blue circle` from `red circle, blue square` (Bower, 1974; Thorndyke and Hayes-Roth, 1979). Another interesting extension would be to allow mutually excitatory traces. Activating one trace would then automatically turn on one or more other traces also. This process can be used to implement heteroassociative recall (Kohonen and Mäkisara, 1986).

Modulating the trace diameter has not been explored at all in the current model. It would be possible to make the early traces more resistant to forgetting by storing them on a wider area. Also, traces at a crowded section of the memory could be made smaller, and more of them would fit in without severe interference. Interestingly, decreasing the diameter of the trace with experience is in line with the self-organizing process, where the neighborhoods are initially large but decrease as the map becomes more ordered. Perhaps traces could be created during self-organization also. Forming traces could be an essential part of the process. The traces would modify the lateral connections so that they better support the current state of self-organization. Combining self-organization with the trace feature map mechanism is a very interesting future research direction.

6 Conclusion

The trace feature map model has many properties that make it attractive as a human associative memory model. The model does not require linear independence or binary representations, and all computations are local. Topological feature map representation makes it possible to store different items on different hardware, which in turn makes episodic storage possible without excessive crosstalk. Where memory interferences occur, they are psychologically plausible. More recent traces are easier to recall as are traces that are unique in the memory. Several other neuropsychological phenomena can also be explained by the model or its extensions. However, how novel memories are encoded and how storing traces can be combined with self-organization remain largely open research questions at this point.

References

Ackley DH, Hinton GE, Sejnowski TJ (1985) A learning algorithm for Boltzmann machines. Cogn Sci 9:147–169

Anderson JA (1972) A simple neural network generating an interactive memory. Math Biosci 14:197–220

Anderson JA (1986) Cognitive capabilities of a parallel system. In: Bienenstock E, Soulie FF, Weisbuch G (eds) Disordered systems and biological organization. Springer, Berlin Heidelberg New York, pp. 209–226

Anderson JA, Silverstein JW, Ritz SA, Jones RS (1977) Distinctive features, categorical perception and probability learning: some applications of a neural model. Psych Rev 84:413–451

Baddeley AD (1976) The psychology of memory. Basic Books, New York

Baddeley AD (1982) Amnesia: a minimal model and an interpretation. In: Cermak LS (ed) Memory and amnesia. Lawrence Erlbaum, Hillsdale

Ballard DH (1986) Cortical connections and parallel processing: structure and function. Behav Brain Sci 9:67–120

Barlow HB (1972) Single units and sensation: a neuron doctrine for perceptual psychology? Perception 1:371–394

Bower GH (1974) Selective facilitation and interference in retention of prose. J Educ Psych 66:1–8

Cooper LN (1973) A possible organization of animal memory and learning. In: Lundquist B, Lundquist S (eds) Proceedings of the Nobel symposium on collective properties of physical systems. Academic Press, New York, pp 252–264

Feldman JA, Ballard DH (1982) Connectionist models and their properties. Cogn Sci 6:205–254

Grossberg S (1983) Studies of mind and brain: neural principles of learning, perception, development, cognition and motor control. Boston studies in the philosophy of science, vol 70. Reidel, Dordrecht, Holland Boston

Halgren E (1984) Human hippocampal and amygdala recording and stimulation: evidence for a neural model of recent memory. In: Butters N, Squire L (eds) The neuropsychology of memory. Guilford, New York, pp. 165–182

Halgren E (1990) Physiological integration of the declarative memory system (unpublished)

Hinton GE, Anderson JA (eds) (1981) Parallel Models of Associative Memory. Lawrence Erlbaum, Hillsdale

Hinton GE, McClelland JL, Rumelhart DE (1986) Distributed representations. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol 1: Foundations. MIT Press, Cambridge, Mass, pp. 77–109

Hinton GE, Sejnowski TJ, Ackley DH (1984) Boltzmann machines: constraint satisfaction networks that learn. Technical Report CMU-CS-84-119, Computer Science Department, Carnegie Mellon University

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Sciences USA, pp 2554–2558

Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. In: Proceedings of the National Academy of Sciences USA, pp 3088–3092

Kanerva P (1988) Sparse distributed memory. MIT Press, Cambridge, Mass

Knapp AG, Anderson JA (1984) Theory of categorization based on distributed memory storage. J Exp Psych: Learning, Memory Cogn 10:616–637

Knudsen EI, du Lac S, Esterly SD (1987) Computational maps in the brain. In: Cowan WM, Shooter EM, Stevens CF, Thompson RF, (eds) Ann Rev Neurosci, pp 41–65. Annual Reviews, Palo Alto, Calif

Kohonen T (1972) Correlation matrix memories. IEEE Trans Comput C-21:353–3590

Kohonen T (1977) Associative memory – a system-theoretical approach. Springer, Berlin Heidelberg New York

Kohonen T (1982a) Analysis of a simple self-organizing process. Biol Cybern 44:135–140

Kohonen T (1982b) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Kohonen T (1984) Self-organization and associative memory. Springer, Berlin Heidelberg New York

Kohonen T (1990) The self-organizing map. Proceedings of the IEEE 78:1464–1480

Kohonen T, Mäkisara K, Saramäki T (1984) Phonotopic maps – insightful representation of phonological features for speech recognition. In: Proceedings of the 6th International Conference on Pattern Recognition. IEEE Computer Society Press, pp 182–185

Kohonen T, Mäkisara K (1986) Representation of sensory information in self-organizing feature maps. In: Denker J (ed) AIP Conference Proceedings 151. American Institute of Physics, New York, pp 271–276

Konishi M (1986) Centrally synthesized maps of sensory space. Trends Neurosci 9:163–168

Little WA, Shaw GL (1975) A statistical theory of short and long term memory. Behav Biol 14:115–133

Loftus EF (1975) Leading questions and the eyewitness report. Cogn Psych 7:560–572

Loftus EF, Miller DG, Burns HJ (1978) Semantic integration of verbal information into a visual memory. J Exp Psych: Hum Learn Memory, 4:19–31

McClelland JL, Rumelhart DE (1986) A distributed model of human learning and memory. In: McClelland JL, Rumelhart DE (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol 2: Psychological and biological models. MIT Press, Cambridge, Mass, pp 170–215

Miikkulainen R (1990a) DISCERN: a distributed artificial neural network model of script processing and memory. PhD thesis, Computer Science Department, University of California, Los Angeles

Miikkulainen R (1990b) A distributed feature map model of the lexicon. In: Proceedings of the Twelfth Annual Cognitive Science Society Conference. Lawrence Erlbaum, Hillsdale, pp. 447–454

Miikkulainen R (1990c) Script recognition with hierarchical feature maps. Conn Sci 2:83–101

Miikkulainen R (1991) Self-organizing process based on lateral inhibition and synaptic resource redistribution. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN-91), Espoo, Finland, pp. 415–420

Postman L (1971) Transfer, interference and forgetting. In: Kling JW, Riggs LA (eds) Experimental psychology, 3rd edn. Holt, Rinehart and Winston, New York, pp 1019–1132

Read W, Nenov VI, Halgren E (in press) Inhibition-controlled retrieval by an autoassociative model of hippocampal area CA3. Hippocampus

Ritter H, Kohonen T (1989) Self-organizing semantic maps. Biol Cybern 61:241–254

Ritter HJ, Schulten KJ (1988) Convergency properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection. Biol Cybern 60:59–71

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol 1: Foundations. MIT Press, Cambridge, Mass, pp. 318–362

Shimamura AP (1986) Priming effects in amnesia: evidence for a dissociable memory function. Q J Exp Psych 38:619–644

Thorndyke PW, Hayes-Roth B (1979) The use of schemata in the acquisition and transfer of knowledge. Cogn Psych 11:82–106

Tulving E, Schacter DL (1990) Priming and human memory systems. Science 247:301–306

van Gelder T (1989) Distributed Representation. PhD thesis, Department of Philosophy, University of Pittsburgh

von der Malsburg C (1987) Synaptic plasticity as basis of brain organization. In: Changeux J-P, Konishi M (eds) The neural and molecular bases of learning. Wiley, New York, pp 411–432

Warrington EK (1975) The selective impairment of semantic memory. Q J Exp Psych 27:635–657

Warrington EK and McCarthy RA (1987) Categories of knowledge: further fractionations and an attempted integration. Brain 110:1273–1296

Warrington EK, Shallice T (1984) Category specific semantic impairments. Brain 107:829–854

Warrington EK, Weiskrantz L (1974) The effect of prior learning on subsequent retention in amnesic patients. Neuropsychologia 12:419–428

Willshaw DJ, Buneman OP, Longuet-Higgins HC (1969) Non-holographic associative memory. Nature 222:960–962

APPENDIX: Tuning the parameters

The operation of trace feature maps is highly sensitive to the sigmoid and lateral weight parameters. Reliability of the settling process, the extent of the trace, the accuracy of the retrieved vector, the distinction between successful retrieval and failure, and the maximum distance for retrieval can be controlled with these parameters.

Ideally, settling should be a continuous and gradual process, where each successive pattern is closer to the final stable pattern. In many cases however, it resembles damped oscillation, where the activity alternates between two patterns that both converge to the same final pattern. In one of these patterns the lateral connections dominate, the other one reflects the external input activity. Oscillatory behavior is more likely to occur when the lateral weight parameters γ_E and γ_I are large. Thus oscillations can be reduced by reducing the absolute values of the weights (or equivalently, the θ-parameter), and also by reducing the ratio $\frac{\gamma_I}{\gamma_E}$.

The diameter of the trace depends on the extent of the final stable activity pattern. The steeper the slope of the sigmoid, the more the sigmoid is displaced towards infinity, and the higher the inhibition, the smaller the bubble. Narrower