Recurrent Networks as Models of Short Term Memory

Matt Jones (mattj@umich.edu) and Thad A. Polk (tpolk@umich.edu)
Department of Psychology, 525 E. University
Ann Arbor, MI 48109 USA

Abstract

The class of recurrent networks known as attractor networks is known to exhibit behaviors relevant to modeling human memory processes – notably content-addressable memory, storage of repeated inputs as stable patterns (under Hebbian learning), and maintenance of information (as activity) over time. In addition, these networks provide a natural account of the effect of similarity on interference in recall. However when looked at in finer detail there are some ways in which traditional attractor networks fail as models of human short-term memory. In particular, information in human short-term memory decays over time unless it is rehearsed, rather than remaining indefinitely. Also, under Hebbian learning traditional attractor networks have particular trouble learning correlated patterns. Here we investigate some variations on the classic framework which make it more appropriate for modeling human STM. We show (1) how adjusting the threshold of continuously-valued units can lead to networks which maintain activity information temporarily, but decay over time, (2) how noise in learning and/or input leads the similarity structure of the set of stored patterns to be reflected in the distribution of recall errors, and (3) how adding a time-delayed anti-correlative component to the learning rule provides robustness against highly correlated patterns and varying levels of input. These ideas have been incorporated in a model of serial recall that explains many aspects of human behavior on that task, and have also been used in newer simulations that learn temporal properties of the environment.

Introduction

In 1982, John Hopfield introduced the concept of attractor networks – fully and symmetrically connected recurrent networks which he proved to have some remarkable computational properties. These networks, and their extension to continuous-valued units (Hopfield, 1984), have since spawned a considerable amount of research into their potential as associative memory devices.

The main power of an attractor network, in this context, comes from its ability to simultaneously store multiple distributed patterns, termed attactors, in its connection weights. When placed into an arbitrary pattern and allowed to update its activity, the network will settle into that attractor which is most similar to the input. Once settled, the network will maintain its state indefinitely via reverberatory activity among the active units. Furthermore, when a fully recurrent network is trained under a Hebbian learning rule, the patterns on which it was trained tend to become attractors. Thus the network can be said to remember the patterns it has seen in the past.

Together these properties make attractor networks natural candidates for models of human short-term memory. However, there are two important ways in which their behavior differs from that of humans. First there is considerable evidence (e.g., Reitman, 1974) that information in human short-term memory spontaneously decays rather than being held indefinitely. Second, while similarity between stored patterns leads to precisely the type of misclassification errors and interference effects observed in short-term memory, Hebbian learning is particularly bad at learning the highly similar patterns necessary to adequately achieve these effects. In fact, the mechanism which serves to limit the capacity of an attractor network, known as crosstalk, can be formulated entirely in terms of the correlations among the patterns to be stored.

Here we present a modified version of the standard framework of attractor networks, which includes solutions to these problems in the form of a decay mechanism and a learning rule that can handle highly correlated patterns. We then present two demonstrations of the modified framework's potential for modeling human short-term memory: a model of the serial recall task, and preliminary work exploring a network's ability to adaptively adjust its decay time in response to its environment.

A Mechanism for Decay

In a standard attractor network, a stored attractor pattern is a fixed point of the network dynamics because once the network is put into that pattern, the ON units will (on average) all mutually excite each other, serving to maintain their collective activity. But more interesting behavior can emerge when this mutual excitation isn't quite enough to maintain the original activity. With continuous valued units, it's possible for successive updating to lead to slight incremental decreases in the activity levels of each unit, so that over time the information stored as the activation of a particular attractor (or pseudo-attractor) will decay to baseline.

Consider the simple case of a pattern consisting of n units with pairwise connections all of weight w, and assume for the moment (for ease of analysis) that units are updated synchronously. Figure 1 shows the relationship between input and output activities for these units. The sigmoid curve is the neural activation function O(I), giving each individual node's output activity level as a function of its input. The straight line shows the I(O) function – each unit's input (on the horizontal axis) as a function of average output activity of all other units (vertical axis). Using the cobweb diagram approach of dynamic systems theory we can trace the trajectory of the network until it reaches equilibrium. In part (a) we see the network converging on an equilibrium state of high activity, corresponding to the attractor status of the pattern in question.

If, however, the situation is altered slightly by increasing the bias of all units, i.e. shifting the activation function to the right so that more input is required to achieve the same output (part b of Figure 1), then the relationship between the O(I) and I(O) curves can change in a crucial way. When the bias is just slightly above the bifurcation point where the two curves intersect, the equilibrium of the previous example is replaced by a temporary 'bottleneck.' The network eventually settles to a state of minimal activity, but it takes a long time to get there. Because of this we can think of the collection of units as forming a 'pseudo-attractor' which is maintained only temporarily rather than permanently.

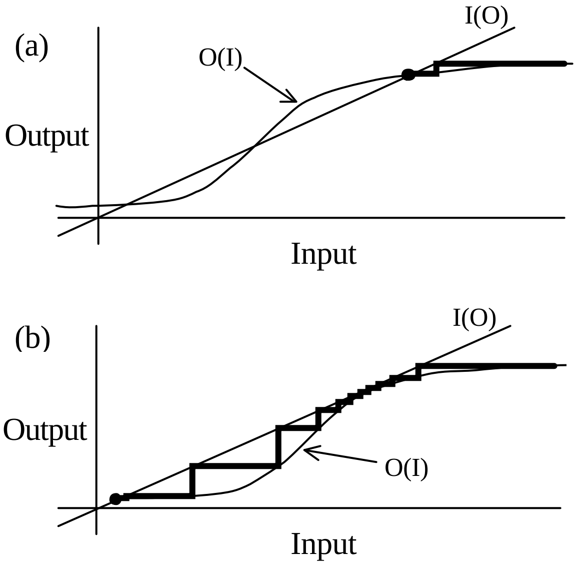

Figure 1. Relationship between inputs and outputs in the network. Evolution of activity can be traced by moving horizontally from a given output level to the I(O) line to give the corresponding input to all nodes, and then moving vertically to the O(I) curve to find the output on the next time step. In (a) we see a standard attractor with a high-activity equilibrium, whereas in (b) the pattern is no longer a true attractor, lingering through a bottleneck but eventually decaying.

The main theoretical problem with this decay mechanism is the hypersensitivity of the decay time to the precise relationship between bias, weights, and number of units. Because the approach relies on being very near the bifurcation between decaying and non-decaying systems, deviations of magnitudes expected in a biological system would be more than enough to drastically alter the decay time or even cross over to the non-decaying regime. Analytically it can be shown that the decay time is asymptotically proportional to $(\alpha-\alpha^*)^{-1/2}$, where α is the ratio of bias to synaptic weight and α^* is the value at the bifurcation point. However changing to asynchronous updating and introducing noise into the dynamics can both be shown to reduce this problem. In the case of noise, the network eventually decays even when below the bifurcation point, as random fluctuations allow it to cross the wall in the energy landscape between the two basins of attraction.

Similarity-based Interference

One phenomenon that appears regularly in memory research is that of interference between similar stored pieces of information. The nature of attractor networks makes them well-suited for modeling this phenomenon, because their classification behavior is based on similarity between input and stored patterns. Under noisy input, or with noisy dynamics, the probability of misclassification (or misrecall) of one item for another has a direct negative relationship to the similarity (Hamming distance) between the two representations.

In modeling phonological confusion data in verbal working memory (Jones & Polk, 2001, in press), we have been able to show (see below) that equating phonological similarity with the degree of overlap among representations in a recurrent network can help to explain these patterns of confusion errors. However the assumption of Hebbian learning in this case is inadequate, as the level of crosstalk generated by the patterns makes them unstable.

In order to counteract the unwanted influence of inactive patterns in making trained patterns unstable, we have added a second, anti-correlative term to the Hebbian updating equation for the weight from unit *j* to unit *i*:

$$\Delta w_{ij} = e_1 \cdot a_j^t a_i^t - e_2 \cdot a_j^t a_i^{t+1}$$

Here the superscript indicates time, with the second term being derived after every unit has updated its activity once.

While the first term (standard Hebbian learning) tends to make the strength of connection between two units approximate the (synchronized) correlation between their activities, the second term deducts based on the 1-step delayed correlation, effectively penalizing synapses for having a consistent effect on their postsynaptic unit. In equilibrium the two terms balance, and the average effect of unit *j* on unit *i* (2nd term) is exactly proportional to the correlation of the two units' activities across the training set (1st term).

When the two learning rate parameters, ε_1 and ε_2, are equal, the algorithm can be interpreted another way. Rearranging terms we get:

$$\Delta w_{ij} = e \cdot a_j d_i \quad \text{with} \quad d_i = a_i^t - a_i^{t+1}$$

The δ-vector can be thought of as an error signal, where the desired value at time *t+1* is a^t. Thus the algorithm is equivalent to 1-step back-propagation through time (Minsky & Papert, 1969; Rumelhart, Hinton, & Williams, 1986), where the network is learning to maintain the same training pattern into which it is placed. Since this task is linearly separable (there are no XOR type problems possible), the general robustness of the back-propagation algorithm can be expected to apply.

Indeed, simple simulations show a clear advantage for the modified rule as compared to the original in terms of learning of correlated patterns, learning under varying levels of input (under the Hebbian rule, networks tend to fall into superpositional patterns when given too much input), and storage capacity for randomly selected sets of patterns.

Application: Model of Serial Recall

The ideas described thus far have been incorporated into an attractor-based model of the serial recall task (Jones & Polk, 2001; in press). The serial recall task is a well-studied test of verbal short term memory, in which a subject is presented with a sequence of items (most commonly words, letters, or digits), and then asked to repeat them back in order. Of the many established phenomena associated with this task, two are especially relevant to this discussion. The *recency effect* (e.g., Crowder, 1972) refers to an advantage in recall probability for items at the end of the list as compared to those immediately before. The *phonological similarity effect* (Conrad & Hull, 1964) is the fact that lists consisting of similar sounding items are associated with poorer recall, and furthermore that when only a subset of the items on a list are similar, errors are predominantly confusions between those particular items (Baddeley, 1968).

Our approach in modeling these and other effects was to connect different attractor-type networks, separately representing item identity and position within a list. The architecture of the model is shown in Figure 2. During presentation of items (and subsequent rehearsal), associations are learned from the Position network to the Item network. At recall, the Position network is sequentially set into each position pattern, and the resulting input to the Item network allows it to converge upon a response.

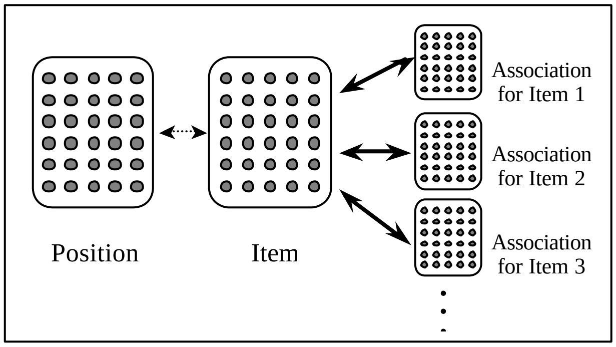

Figure 2. The architecture of the model. Circles represent nodes, rectangles represent individual attractor networks, and arrows represent connections between units in different networks. Nodes within each network are fully interconnected (not shown).

Prior to simulation of the task itself, this Item network is trained under the modified Hebbian algorithm on 12 patterns, 6 of which share a pool of common units (the confusable items), and 6 of which are disjoint (in terms of their ON units) from the first six and from each other. Similarity between the confusable items is expected to lead to interference effects, and specifically misclassification errors at the time of recall.

In addition to the Item network, which is the source of competition between items in response selection, there is a set of Association networks, one for each pattern stored in Item. Each Association network has a single pseudo-attractor with positive associations to the corresponding Item pattern, thus serving as an additional (non-competitive) component to that item's representation. These networks are tuned to decay over slightly less than the amount of time it takes for the system to recall the list. Because presentation takes longer than recall, the Associations for early list items have decayed by the time they are to be recalled, whereas recall of the final item is facilitated by extra input from its Association network.

The model was tested on its ability to simulate data from an experiment by Baddeley (1968, expt. V), which tested subjects on lists of 6 letters. All letters were selected from a pool of 12, six of which were acoustically similar to each other (B,C,D,P,T,V) while the other six were all dissimilar (J,K,L,R,W,Y). Two types of lists were tested: *Even* lists, in which positions 2, 4, and 6 contained items from the similar group, and *Odd* lists, in which similar items appeared in positions 1, 3, and 5. Figure 3 shows the results of this experiment, along with data simulated from the model. In both cases we see selective impairment for the confusable items on all six serial positions (except perhaps the first). Superimposed on this similarity effect is a recency effect: By regrouping the data into separate serial position curves for confusable items and non-confusable items – combining the data from the Even lists for positions 2, 4, and 6 with that from the Odd lists for positions 1, 3, and 5, and vice versa – we see that in both curves, in both the empirical and simulated data, there is a drop in errors on the final position.

From the perspective of our model, the phonological similarity effect is caused by similarity-based interference in the Item network, i.e., an increased probability of misrecall of one similar item for another. The recency effect is due to the decay of information in the Association networks leading to an advantage for the most recently presented item.

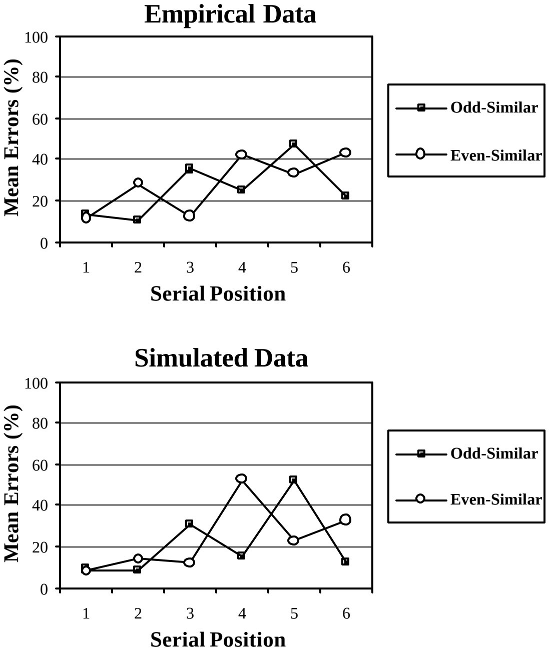

Figure 3: Mean error rates in the phonlogical similarity experiment in both empirical (Baddeley, et. al., 1968) and simulated studies. Data in each graph are divided into condition according to which list positions (evens or odds) contain the phonologically similar items.

Learning of Decay Rates

One further idea that comes out of the framework laid out here is that under the modified Hebbian rule and with the proper feedback, informational decay rates should be adaptable. In this way a system or organism may be able to learn the characteristic time-scale of certain types of information in its environment.

The simplest unsupervised learning paradigm involves a network in continuous contact with its environment, learning associations among the various stimuli to which it is exposed. However it may be more realistic to assume that information about the environment (or at least individual aspects of it) is only intermittently available via sensory channels. In this case one desired function of a recurrent network could be to maintain an accurate representation of the state of the environment even when the sensory channel is closed. In order to do this such a network must be able to learn the temporal dynamics of that aspect of the environment which it is to represent, so that it can mimic those dynamics even when deprived of input.

Consider a simple case with only two possible environmental states – object present and object absent. Furthermore assume that the representation of the object involves all nodes being active (otherwise we can just ignore non-participating nodes), and that absence of the object is represented by all units inactive. When the sensory channel is active, the network receives information about the status of the object and is clamped to the correct representation; however when the channel is closed the network evolves its internal activity freely. Finally assume that learning takes place only when input is available, and not when the network is evolving freely.

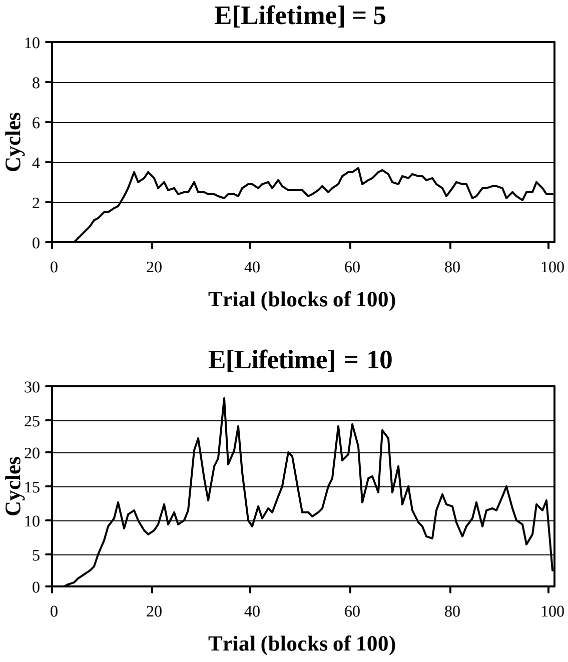

Figure 4: Mean network decay times as a function of trial, for two different average lifetimes of the object in the environment.

To test what would happen in this situation, simulations were run using exponential distributions for both the lifetime of the object and the spacing of contact between the network and its environment. On each trial, the network was presented at time 0 with input representing object present, and then the sensory channel was closed (i.e. the network was isolated from further input). Some time later the network again received information on the status of the environment, depending on whether the object was still present, and learned based on the discrepancy of its prediction (using the modified Hebbian rule).

After each training trial the network was tested, without learning, for its average decay time. The results (see Figure 4) show that the network was indeed able to adjust its decay time in response to feedback from its environment, to a value that roughly approximated the average lifetime of the object.

Conclusions

Attractor networks clearly have a lot to offer cognitive modelers, but the classical framework needs to be extended to capture some of the core phenomena associated with human short-term memory. Here we have discussed two ways in which the standard framework departs from human behavior – indefinite maintenance of activity, and lack of robustness to highly correlated patterns. The proposed mechanism for decay of activity, along with the modification to the Hebbian learning rule, help to address these shortcomings, as demonstrated by our model of the serial recall task. In this model the decay mechanism provides an explanation for the decay of information often simply built in to other models of short-term memory, while the modified learning rule allows the system to store correlated patterns, which in turn provide the basis for similarity-based interference effects.

While it's easy to see the trouble with correlated patterns as a deficiency of the standard model relative to human memory, indefinite storage of information may in fact be one as well (rather than a superiority as it could seem at first glance). Much recent speculation (Schacter, 1999; J.R. Anderson & Schooler, 2000) has centered on the idea that decay of information in short term memory is in fact functional, and there is evidence that the rate of this decay is adaptable to statistics of the environment (R.B. Anderson, 1997; Jones & Sieck, 2001). The final demonstration given here provides one possible explanation of how that adaptation may take place, and may provide a starting point for a theory on the learning of more complex temporal dynamics.

References

Anderson, J. R. & Schooler, L. J. (2000). The adaptive nature of memory. In E. Tulving & F. I. M. Craik (Eds.), *The Oxford handbook of memory* (pp557-570), New York: Oxford University Press

Anderson, R. B., Tweney, R. D., Rivardo, M., & Duncan, S. (1997). Need probability affects retention: A direct demonstration. *Memory & Cognition, 25(6)*, 867-872

Baddeley, A. D. (1968). How does acoustic similarity influence short-term memory. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology, 20A*, 249-264.

Conrad, R. & Hull, A. J. (1964). Information, acoustic confusion and memory span. *British Journal of Psychology, 55*, 429-432.

Crowder, R. G. (1972). Visual and auditory memory. In J. F. Kavanaugh & I. G. Mattingly (Eds.), *Language by ear and by eye.* New York: McGraw-Hill.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*, 2554-2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of 2-state neurons. *Proceedings of the National Academy of Sciences, 81*, 3088-3092.

Jones, M. & Polk, T. A. (2001). An attractor model of serial recall. *Proceedings of the 4th International Conference on Cognitive Modeling*, 121-126.

Jones, M. & Polk, T. A. (in press). An attractor model of serial recall. *Cognitive Systems Research.*

Jones, M. & Sieck, W. (2001). Sampling dependencies and diagnosis learning. *Manuscript in preparation.*

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons*. Cambridge: MIT Press.

Reitman, J. S. (1974). Without surreptitious rehearsal, information in short-term memory decays. *Journal of Verbal Learning & Verbal Behavior, 13(4)*, 365-377

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations* (Chap 8). Cambridge: MIT Press.

Schacter, D. L. (1999). The seven sins of memory: Insights from psychology and cognitive neuroscience. *American Psychologist 54(3)*, 182-203