Interactive tandem networks and the sequential learning problem

Robert M. French
Center for Research on Concepts and Cognition
Indiana University, Bloomington, IN 47408
french@cogsci.indiana.edu

Abstract

This paper presents a novel connectionist architecture to handle the "sensitivity-stability" problem and, in particular, an extreme manifestation of the problem, catastrophic interference. This architecture, called an interactive tandem-network (ITN) architecture, consists of two continually interacting networks, one — the LTM network — dynamically storing "prototypes" of the patterns learned, the other — the STM network — being responsible for "short-term" learning of new patterns. Prototypes stored in the LTM network influence hidden-layer representations in the STM network and, conversely, newly learned representations in the STM network gradually modify the more stable LTM prototypes. As prototypes are learned by the LTM network, they are dynamically constrained to maximize mutual orthogonality. This system of tandem networks performs particularly well on the problem of catastrophic interference. It also produces "long-term" representations that are stable in the face of new input and "short-term" representations that remain sensitive to new input. Justification for this type of architecture is similar to that given recently by McClelland, McNaughton, & O'Reilly (1994) in arguing for the necessary complementarity of the hippocampal (short-term memory) and neocortical (long-term memory) systems.

Introduction

The problem of catastrophic interference has been discussed in the connectionist literature since it was first brought to light by McCloskey and Cohen (1989) and Ratcliff (1990). Catastrophic interference is characterized by the abrupt and radical forgetting of previously learned information due to the learning of new information. Not only is catastrophic interference unhuman-like, but it also poses significant problems for potential applications of connectionist networks — a face-recognition program that can correctly identify one hundred faces but forgets them all upon learning five new ones is obviously of little practical use. It is therefore important to attempt to design systems that resist such sudden loss of stored information, while still remaining sensitive to new input.

One way to frame the problem of catastrophic interference is in terms of the well-known "sensitivity-stability" problem (Hebb, 1949). How can a system be designed that is at once sensitive to new input, yet not disrupted by it? As memory devices, lookup tables and backpropagation networks lie at opposite ends of the stability-sensitivity spectrum. The former are certainly stable in the presence of new information but lack the crucial ability to generalize on new input. On the other hand, standard backpropagation networks store information in a highly distributed, interconnected manner, making them extremely sensitive to new input but because new representations will almost invariably overlap with old ones, severe interference is likely to result.

The idea that catastrophic interference and generalization are simply the flip sides of the same coin was suggested by French (1991). By reducing the amount of overlap among internal representations, it was demonstrated that a corresponding decrease in catastrophic interference could be achieved. French argued for the use of *semi-distributed* representations, in other words, internal representations in which information was distributed over a subset of the hidden layer rather than over the entire hidden layer. The idea was that if the active portions of representations could be confined to a subset of the hidden nodes, it would be possible to have representations that had a higher degree of mutual orthogonality than with standard backpropagation. This would prevent the representations from interfering with each other as much and would reduce catastrophic forgetting.

Numerous other techniques have been developed that attempt to modify connectionist networks so as to allow them to retain their ability to generalize while at the same time reducing their tendency to catastrophically forget when learning new information. Not unsurprisingly many of these techniques rely, either implicitly or explicitly, on some form of "semi-distributed" internal representation scheme in order to achieve improved performance catastrophic interference (Krushke, 1993; McRae & Hetherington, 1993; Murre, 1992; French, 1994; Tetewsky, Schultz, & Buckingham, 1994; Sharkey & Sharkey, 1994; etc.).

Theoretical reasons for an interactive tandem-network (ITN) architecture

All of the above systems attempt to solve the problem of catastrophic interference within the context of a single network. There are at least two major problems with this type of approach. These are:

- Single networks do not, in general, differentiate between short-term learning and long-term learning. (Two notable exceptions to this are found in Hinton & Plaut (1987) and Kaplan, Chown, & Sonntag (1991)).
- Overlearning is, in general, not possible in a single-network system. Once an error-driven connectionist network has learned a set of patterns to criterion, no amount of re-presentation of the patterns will "solidify" the learning. Once criterion is reached, further weight changes cease regardless of the number of additional presentations of the input.

In this paper, I will present an interactive tandem network (ITN) architecture comprising a "long-term memory (LTM) network" interacting with a "short-term memory (STM) network." A brief outline of the ITN architecture can be found in French (1994). The present paper will show that the prototypes learned by this system are indeed considerably more resistant to change than the representations learned in the short-term network, which, in turn, are more resistant to change than representations in a standard backpropagation network. I will also show that this type of network performs very well on the problem of catastrophic interference and, yet, remains sensitive to new input. Finally, there is an obvious parallel between this interacting tandem-network architecture and the complementary hippocampal and neocortical sytems described in McClelland, McNaughton, & O'Reilly (1994). These authors justify the complementary nature of these two neural systems is as follows:

> "The sequential acquisition of new data is incompatible with the gradual discovery of structure and can lead to *catastrophic interference* with what has previously been learned. In light of these observations, we suggest that the neocortex may be optimized for the gradual discovery of the shared structure of events and experiences, and that the hippocampal system is there to provide a mechanism for rapid acquistion of new information without interference with previously discovered regularities. After this initial acquistion, the hippocampal system serves as a teacher to the neocortex..."

The justification of the ITN architecture described below is remarkably similar to that given above by McClelland, McNaughton, & O'Reilly. The advantages of this type of interactive tandem-network design are:

- Good performance on catastrophic interference.
- Separate storage of long-term memory structures (prototypes) and recently learned, short-term information.
- Prototype consolidation occurs gradually in the prototype network, being learned from the more ephemeral representations in the short-term network.
- Previously learned prototypes influence (or bias) representations of newly acquired information in the short-term network.

In what follows, I will describe the details of the architecture and describe its performance on the problem of catastrophic interference.

Details of the ITN architecture

The proposed architecture consists of two standard backpropagation networks: the STM-network — for "short-term" learning — and the LTM-network — for storing prototypes of the representations learned by the STM-network. The general idea of prototype learning is as follows. Assume that a number of input patterns I_1, ..., I_n, belong to two separate categories C_1 and C_2 and that the STM-network has learned to correctly associate these ten patterns with their respective categories. Taking an average of the STM-network's hidden-layer representations for all of the input patterns in category C_1 would produce a hidden-layer "prototype", P_1, for that category. Similarly, the average of all STM-network's hidden-layer representations associated with C_2 would produce a prototype P_2. The LTM-network (or prototype-network) then learns to associate the two categories with their respective prototypes, in this case, C_1 with P_1 and C_2 with P_2.

The LTM-network is responsible for three things, discussed in detail below:

- building weighted prototypes
- dynamically separating prototypes ("context-biasing")
- influencing representations of new patterns in STM-network.

Building weighted prototypes

Prototypes are gradually built up based on the representations in the short-term network. This works as follows. Assume that a new input I is presented to the ITN and is to be associated with the category C. Presumably, the LTM-network already has some prototype P associated with the category C (initially, a random vector). When the input I is fed forward through the STM-network, it produces a hidden-layer representation R and an output O. Presumably, there will be an error, ε, between the desired output C, and the actual output O.

The greater this error, the smaller the contribution of R to the already-existing prototype P associated with the category C. The new weighted prototype that is built by the LTM-network is therefore a weighted average of the old prototype and the hidden-layer representation of the new input:

$$P_{new} = (\omega_P P_{old} + \omega_R R)/(\omega_P + \omega_R)$$

where:

P_{new} is the new prototype;

P_{old} is the old prototype;

R is the representation in the STM-network of the new pattern

ω_P is a constant weighting factor that gives the prototype more weight than any single representation from the short-term network;

$\omega_R = (1 - \varepsilon)$ is the "weight" of the STM representation in the new prototype. ε is the error between the desired category C and actual output O of the STM network.

This method of prototype averaging is patterned after Anderson's weighted averaging model of information integration (Anderson, 1981). Before the LTM-network actually learns this new prototype, it must first be "context-biased" (described below) in order to increase its orthogonality with respect to the other prototypes stored in the LTM-network.

Dynamically separating prototypes

In order to reduce catastrophic interference in a simple backpropagation network, French (1994) introduced a technique called context-biasing, designed to produce hidden-layer representations that were both as orthogonal and as distributed as possible. The LTM-network uses this same technique to "context-bias" the prototypes, thereby ensuring that they are both well separated and well distributed.

In order to decrease the mutual overlap of prototypes, the LTM-network has a one time-step memory. On each new Category-Prototype association to be learned, the LTM-network must remember the previous Category–Prototype (C–P) association, in much the same way that an Elman network "remembers" the previous hidden-layer representation that is fed into the current hidden layer (Elman 1989).

When the LTM-network learns the association of the current category, C_t, with the current prototype, P_t, it first computes the average Hamming distance between C_t and the previously seen category, C_{t-1}. Based on this Hamming distance, P_t is modified so as to "separate" it from the previous prototype, P_{t-1}. The greater the Hamming distance between the two category vectors, C_t and C_{t-1}, the greater the separation between the associated prototypes, P_t and P_{t-1}. In other words, the degree of separation will reflect the size of the difference between the categories with which the prototypes are associated. The greater the category disparity, the greater the separation. The precise prototype separation rule is given below. The activation A of each node of the current prototype, $P_{current}$, is modified with respect to Pprevious as follows:

if $A_{current} < A_{previous}$ then

$$A_{biased} = A_{current} - \alpha\beta A_{current}$$

else $A_{biased} = A_{current} + \alpha\beta(1 - A_{current})$

where

α = average Hamming distance between the previous and the current categories;

β = separation coefficient (usually 0.5 or 0.2)

This new context-biased prototype (i.e., with the new "biased" activation values), which includes the weighted addition of the current STM hidden-layer representation, is the one that the prototype network will now learn to associate with the current category.

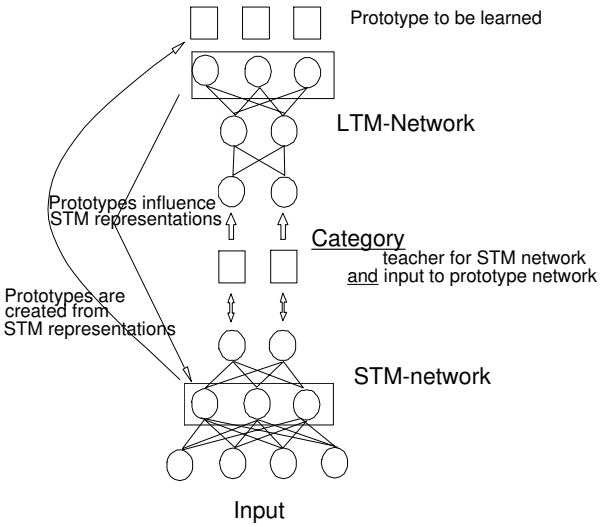

Figure 1: ITN architecture

Representation biasing within the STM-network

In humans, previously learned information stored in long-term memory affects (or "biases") short-term memory representations of new knowledge. The present architecture attempts to simulate the influence of well-learned prototypes on new representations in short-term memory. If a prototype P has a well-learned association in the LTM-network with a category C, when the STM-network is presented with a new input I belonging to C, the STM hidden-layer representation of the I–C association will be "biased" towards P.

This works in the following manner. An input I (to be associated with the category C) is fed forward through the STM-network, producing a hidden-layer representation, $R_{natural}$. The STM output is compared with the desired output C and the error is backpropagated through the STM-network, changing the weights appropriately. At the same time, C is fed forward through the prototype network, producing an output P, the learned prototype associated with C. P then "biases" the activation levels of $R_{natural}$ by shifting them somewhat towards P. This is done as follows. Each node of the representation $R_{natural}$ has an activation level, A_R. The activation level of the corresponding node in the prototype P will be designated A_P. Then, for each node of $R_{natural}$ and its corresponding node in P:

if $A_R < A_P$ then
$A_R^{new} = A_R + \alpha A_{diff}$
else $A_R^{new} = A_R - \alpha A_{diff}$
where:
α is the biasing coefficient (usually 0.5)
$A_{diff} = |A_P - A_R|$

This produces a "biased" representation, R_{biased}. The new set of activations for the biased representation is then "locked into" the input-to-hidden weights by backpropagating from the hidden layer to the input layer an "error signal" consisting of the difference between $R_{natural}$ and R_{biased}. This "locking-in" technique is discussed in (French 1991). It is to be noted that numerous techniques involving dynamically "massaging" the hidden-layer representations in order achieve certain types of representations have been developed (e.g., Kruschke 1989; French 1991; and Murre 1993).

Results

To test this model, I used data from the 1984 Congressional Voting Records (Murphy & Aha, 1992), which gives the voting record and party affiliation (Republican or Democrat) of each member of Congress in 1984. The network was trained to associate 50 different voting patterns with party affiliation. The ITN that was used consisted of two feedforward backpropagation networks: one 16-10-1 STM-network and one 1-4-10 LTM-network for the prototypes. Once the network had learned the 50 initial associations, it was then given a small set of ten "maverick" members of Congress. On six of the sixteen issues the Republican members of this group voted like Democrats, and vice-versa. When the network learned this new associations and was retested on the original set of fifty associations, as expected, it had completely forgotten them. The speed with which the network relearned the original data was used to measure how completely the network had forgotten the original data (Hetherington & Seidenberg, 1989).

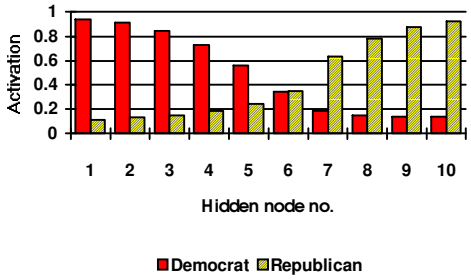

Figure 2. The well-separated, well-distributed STM representations produced by the ITN.

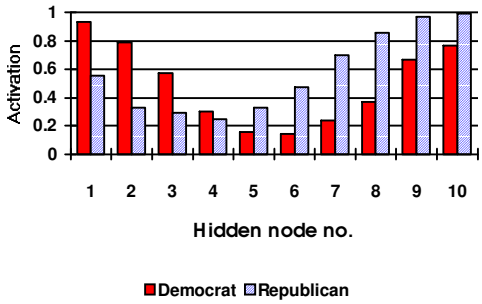

Figure 3. When a standard BP network is run on the same problem, there is far greater representational overlap, and, as a result, worse catastrophic interference

As can be seen in Figure 2, the ITN developed STM representations that were both well distributed across the hidden layer and were also well separated. Figure 3 shows the considerably greater overlap of representations when a standard backpropagation network is used. Figure 4 below shows that the ITN takes half the number of epochs to relearn the data as a standard backpropagation network

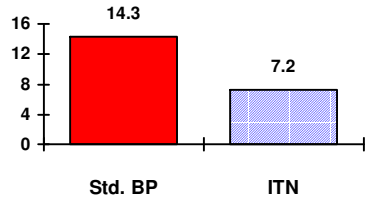

Figure 4. The number of epochs required to relearn the original data for standard BP and ITN.

and has also developed a distinct set of prototypes for the information learned. We will see in the next section that, not only are the LTM prototypes very stable in the presence of disruptive information, but the STM representations remain very sensitive (i.e., exhibit considerable change) to new input

Stability of LTM/Sensitivity of STM

Finally, I will demonstrate the most important advantage of this system, namely that the LTM prototypes are more resistant to change than the representations in STM. The amount of change that the LTM prototype for Democrats undergoes after the ITN has learned the new set of "maverick" inputs is compared to the average amount of change in the corresponding STM representations for the same category. Figure 4 shows the results of 100 runs of the program on the Republicans/Democrats classification problem. Notice that when the ITN learns the new set of maverick members of Congress, the LTM prototypes change only slightly, whereas there is considerable more change in the corresponding STM representations. In other words, the LTM representations (prototypes) remain stable, whereas the STM representations are free to, and indeed do, change quite significantly in the presence of new input. It is also important to note the far greater perturbation of hidden-layer representations in a standard backpropagation network in which the representations cannot benefit from any stabilizing influence of LTM prototypes.

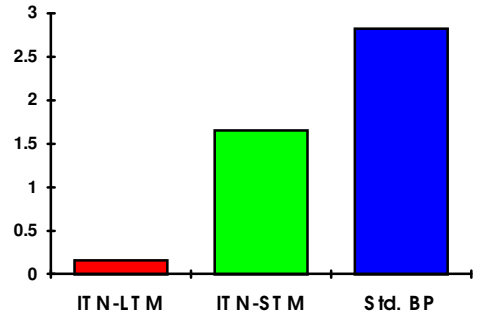

Figure 5. Average disruption (amount of activation change) in LTM prototypes, STM-representations, and BP hidden-layer representations after learning new input.

Conclusions

In an attempt to address the problem of sensitivity and stability in connectionist models of memory, this paper has introduced the concept of an Interactive Tandem-Network (ITN) that consists of two continually interacting backpropagation networks, an LTM network and an STM network. The former stores prototypes that have been gradually built up from representations learned in the STM. These prototypes are distributed and separated in the LTM by a technique called context-biasing (French 1994). The prototypes in LTM also influence new representations in STM. This interactive two-network system is resistant to catastrophic forgetting and produces LTM representations ("prototypes") that are resistant to ephemeral changes in input (guaranteeing stability) and STM representations that are considerably modified by new input (guaranteeing sensitivity to new input). Finally, this system has a natural "real-world" counterpart in the hippocampus/neocortex dichotomy (McClelland, McNaughton, & O'Reilly, 1994). The ITN architecture is a novel attempt to address the important issue of how to build "sensitive yet stable" connectionist models of memory.

Acknowledgments

The author would like to thank Mike Gasser for his contribution to the ideas of this paper.

References

Anderson, N. H. (1981). *Foundations of information integration theory*. New York, NY: Academic Press.

Elman, J. L. (1990) Finding Structure in time. *Cognitive Science*, **14**, 179-211.

French, R. M. (1991) Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. In *Proceedings of the 13th Annual Cognitive Science Society Conference*. Hillsdale, NJ: Lawrence Erlbaum, 173-178.

French, R. M. (1994) Dynamically contraining connectionist networks to produce orthogonal, distributed representations to reduce catastrophic interference. In *Proceedings of the 16th Annual Cognitive Science Society Conference*. Hillsdale, NJ: Lawrence Erlbaum, 335-340.

French, R. M. (1994). Catastrophic forgetting in connectionist networks: Can it be predicted, can it be prevented? In Cowan, J.D., Tesauro,G., & Alspector, J. (eds.) *Advances in Neural Information Processing Systems 6*. San Francisco, CA: Morgan Kauffmann, 1176-1177

Hebb, D. O. (1949). *Organization of Behavior*. New York, N.Y.: Wiley & Sons.

Hetherington, P. A. and Seidenberg, M. S., (1989), Is there 'catastrophic interference' in connectionist networks?, In *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, 26-33.

Hinton, G. E. & Plaut, D. C. (1987) Using Fast Weights to Deblur Old Memories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, 177-186.

Kaplan, S., Chown, E., & Sonntag, M., (1991) Tracing recurrent activity in cognitive elements: The trace mode of temporal dynamics in a cell-assembly. *Connection Science.*

Kruschke, J. K. (1989) Distributed bottlenecks for improved generalization in back-propagation

networks. *International Journal of Neural Networks Research & Applications*, **1**, 187–193.

Kruschke, J. K. (1993) Human Category Learning: Implications for Backpropagation Models. *Connection Science*, Vol. 5, No. 1, 1993.

McClelland, J., McNaughton, B., & O'Reilly, R., Why there are complementary learning systems in the hippocampus and neocortex. CMU Tech Report PDP.CNS.94.1, March 1994.

McCloskey, M. & Cohen, N. J. (1989). "Catastrophic interference in connectionist networks: The sequential learning problem" The Psychology of Learning and Motivation, 24, 109-165.

McRae, K. & Hetherington, P. (1993) Catastrophic interference is eliminated in pretrained networks. In *Proceedings of the 15h Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum. 723-728.

Murphy, P. & Aha, D. (1992). UCI repository of machine learning databases. Maintained at the Dept. of Information and Computer Science, U.C. Irvine, Irvine, CA.

Murre, J. (1992) The effects of pattern presentation on interference in backpropagation networks. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum. 54-59.

Ratcliff, R. (1990) Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions, 97, 285-308.

Sharkey, N. & Sharkey, A., (1994) Understanding Catastrophic Interference in Neural Nets, Computer Science Technical Report CS-94-4, University of Sheffield, Sheffield, England.

Tetewsky, S., Shultz, T., & Buckingham, D. Assessing Interference and Savings in Connectionist Models of Human Recognition Memory, Psychology Department technical report, McGill University, presented at 1994 Meeting of the Psychonomic Society.