The Evolution of Minimal Catastrophic Forgetting in Neural Systems

Tebogo Seipone (T.Seipone@cs.bham.ac.uk)
School of Computer Science, The University of Birmingham,
Birmingham, B15 2TT, UK

John A. Bullinaria (J.A.Bullinaria@cs.bham.ac.uk)
School of Computer Science, The University of Birmingham,
Birmingham, B15 2TT, UK

Abstract

It is well known that neural systems can suffer catastrophic forgetting of previously learned patterns when trained on new patterns, and that this renders many cognitive models unrealistic. However, through evolution, humans have arrived at mechanisms which minimize this problem, and so in this paper we aim to show how simulated evolution can be used to generate neural network models with significantly less catastrophic forgetting than traditionally formulated models.

Introduction

It is normal for humans to gradually forget what they have previously learned, particularly during the learning of new information. However, in traditional artificial neural networks, the forgetting is considerably more catastrophic, and this proves to be a serious limitation of such cognitive models (McCloskey & Cohen, 1989; Ratcliff, 1990; French, 1999). Human brains have presumably evolved by natural selection to minimize this problem. The aim of this paper is to show how simulated evolution can be used to minimize the problem in artificial neural networks too.

We shall begin by describing the problem of catastrophic forgetting in more detail, and outline the principal previous approaches to reduce it. We then discuss the possibilities for evolving artificial neural networks, and explain the particular approach that we have adopted for our study. Then, in our largest section, we present a series of simulation results, and compare each of our evolved systems against the baseline of traditionally built systems. We end with some discussion and conclusions.

Catastrophic Forgetting

After a neural network has been trained on one set of patterns, training on a new set can seriously disrupt, or cause loss of, the previously learned patterns. This 'catastrophic forgetting' is a direct result of the stability/ plasticity dilemma which was first investigated by McCloskey & Cohen (1989) and Ratcliff (1990). Since then, various approaches have been studied in an attempt to reduce or eliminate it (French, 1999).

Several approaches were based on the idea that if some, or all, of the previous information is re-learned together with any new information, then the network will not forget the old information. This process is called interleaved learning (Ratcliff, 1990; McClelland, McNaughton & O'Reilly, 1995). The need to store the old patterns, which may be impractical (and also rather implausible for cognitive systems), can be avoided by 'pseudo-rehearsal' which involves creating pseudo-items (that approximate the original items) to learn with the new items (Robins, 1995). Of course, this still requires storage of the pseudo-items, and it is far from obvious how best to create pseudo-items that represent the old items sufficiently well.

Given that the main cause of catastrophic forgetting is interference in the shared weights, many approaches have attempted to reduce that interference. For example, one can restrict the way in which the hidden unit activations are distributed, and hence the connection usage. French (1991) has used activation sharpening algorithms to reduce the hidden unit activation overlaps. The Sharkey & Sharkey (1995) HARM model uses a neural network implementation of a lookup table. This divides the learning task into two sub-tasks; first eliminating the overlap in the input patterns, and then producing appropriate outputs from the hidden nodes. Other approaches have involved allowing two sets of weighted connections between nodes. Hinton & Plaut (1987) used dual-additive weights, with fast weights to learn new patterns and slow weights for long-term storage.

Related approaches have been based on the belief that humans do not suffer from catastrophic forgetting because their brains have evolved two distinct areas to deal with the problem (McClelland, McNaughton & O'Reilly, 1995). The hippocampal system deals with learning new information, whilst the neocortical system slowly consolidates that new information with the old for long-term storage, using some form of interleaved learning. Dual-model architectures consisting of two distinct networks, one for early processing and another for long-term storage of previously learnt information, together with an interfacing mechanism, have been developed to simulate this separation (French, 1997; Ans & Rousset, 1997).

All these approaches have been based on variations of traditional neural networks, with the designers themselves deciding on the architecture, node activation functions, learning algorithms, the various parameter values, and so on. In this paper, we aim to show how evolutionary computation techniques can be used to *evolve* neural systems that suffer less catastrophic interference than traditionally built systems.

Evolving Neural Network Models

Evolutionary algorithms are a class of non-deterministic optimization techniques that apply the basic principles of evolution by natural selection to find high performance solutions. They maintain a population of individuals, each encoding a potential solution to the problem at hand, and use some form of fitness function to determine which solutions to discard and which to keep to form the next generation. Cross-overs and random mutations are applied to the remaining solutions to create new individuals, and the process continues, with increasingly fit populations. This approach has been combined with artificial neural networks in a number of ways to create highly successful learning systems. It has been used to select optimal network topologies, choices of input features, numbers of hidden units, node transfer functions, learning algorithms and parameters, and even the connection weights themselves, with results reported as being superior to traditionally built systems (e.g. Yao, 1999; Bullinaria, 2003).

In this study, the underlying network architecture and learning algorithm are fixed to be Multi-Layer Perceptrons with one hidden layer, trained by gradient descent weight updating (back-propagation) with the Cross Entropy error measure (Bullinaria 2003). The aim is to evolve the various other neural network topology and learning parameters to produce systems that suffer minimal catastrophic forgetting. Evolution is simulated by using populations of individual neural networks, each initialized with random weights from their own innately specified ranges. The initial population has random innate parameters. At each generation, each network is trained on the same set of initial patterns until it has learnt all those patterns (i.e. has all its output activations within a particular tolerance of their target outputs) or until a maximum number of epochs of training is reached. They are then trained on a new set of patterns in the same manner. After the new patterns have been learned, each network is tested again to see how well the original patterns are still remembered, that is, the number of output units 'correct' (within a particular tolerance) over all of the initial patterns. The fittest individuals are naturally those with the highest number remembered. The least fit half of the population is then removed, and each of the remaining individuals randomly select a mating partner to produce one child, thus restoring the population size. The children inherit innate characteristics (i.e. parameter values) from the range spanned by both parents, with random Gaussian mutations added to allow values outside that range (Bullinaria, 2003). For each new generation, a new global random set of training/remembering data is generated, and all the networks are started with new random initial weights according to their innately specified ranges.

The following Section will make these ideas more concrete by specifying our simulations in more detail, and presenting the results from a systematic sequence of experiments that explore the issues involved.

Simulation Results

To begin we need to fix a convenient training set that is small enough for the simulations to run reasonably quickly, yet large enough to be representative. Consistency with earlier work led us to a variation of that used by Hinton & Plaut (1987), namely random associations of 12 bit random binary patterns with 6 bits 'on'. New random data sets of this specification were generated for each generation. Each network was trained on 20 such patterns until the error on each output bit was less than 0.1, or the maximum of 1000 epochs was reached. It was then trained on a different set of 4 such patterns, and the number remembered correctly from the original 20 was measured using a tolerance of 0.2. All our populations consisted of 100 networks.

It is appropriate to start by establishing the baseline performance levels obtained for standard neural network training parameters. Figure 1 shows the mean percentages remembered over 2500 populations with different random training data sets, trained using traditional back-propagation learning rates of 0.2 and random initial weights uniformly distributed in the range [–1, +1]. We see the extent of the variance due to some data sets being 'easier' than others, and that the performance increases with the number of hidden units. The first test of our simulated evolution was therefore to allow the number of hidden units N_{Hid} to evolve, and on the right of Figure 1 we see that there is a steady rise towards the maximum allowed (10,000 in this case). We shall return to this issue again later, but for the bulk of our simulations we keep the number of hidden units fixed at the more computationally feasible number of 50, for which the baseline remembering performance is 68.7%.

The next thing we wanted to explore was the suggestion of French (1991) that hidden unit activation sharpening could reduce the forgetting by developing semi-distributed representations in the hidden layer. The idea is that, at each epoch of training, the input to hidden weights are modified to bring the N_H highest activation hidden units closer to one, and the N_L lowest activations closer to zero, by some 'sharpening factor' of α times the difference. There are two variations to consider. First, when we force $N_H + N_L = N_{Hid}$ so that all hidden activations get changed, the sharpening factor invariably evolves to zero, leaving us with our standard network. If we let N_H and N_L evolve freely, they both evolve very quickly to zero, again leaving us with our standard network. It seems that node sharpening does not really help with catastrophic forgetting, at least for our class of training data. As a check, these parameters were left free to evolve in all our subsequent simulations, but in each case node sharpening was quickly 'turned off'.

There are several traditional network parameters that one can evolve with the hope of improving performance. To get a feel for which were most effective, we considered each one in turn before evolving them all at once. First we tried evolving the learning rates. It is now well established that allowing separate gradient descent step sizes η_L for each

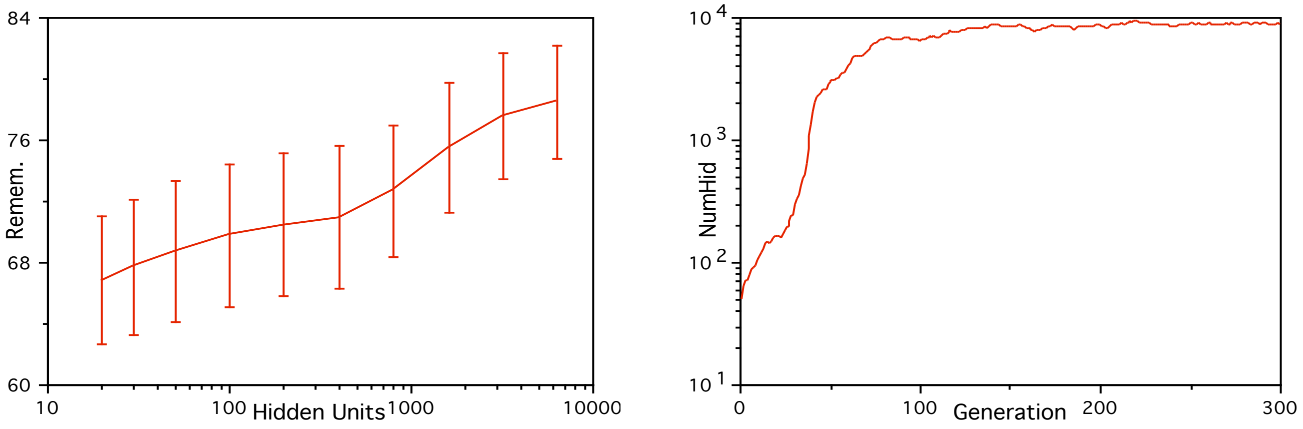

Figure 1: Remembering performance improves with the number of hidden units (left), and simulated evolution consequently causes the number of hidden units to increase to the maximum allowed (right).

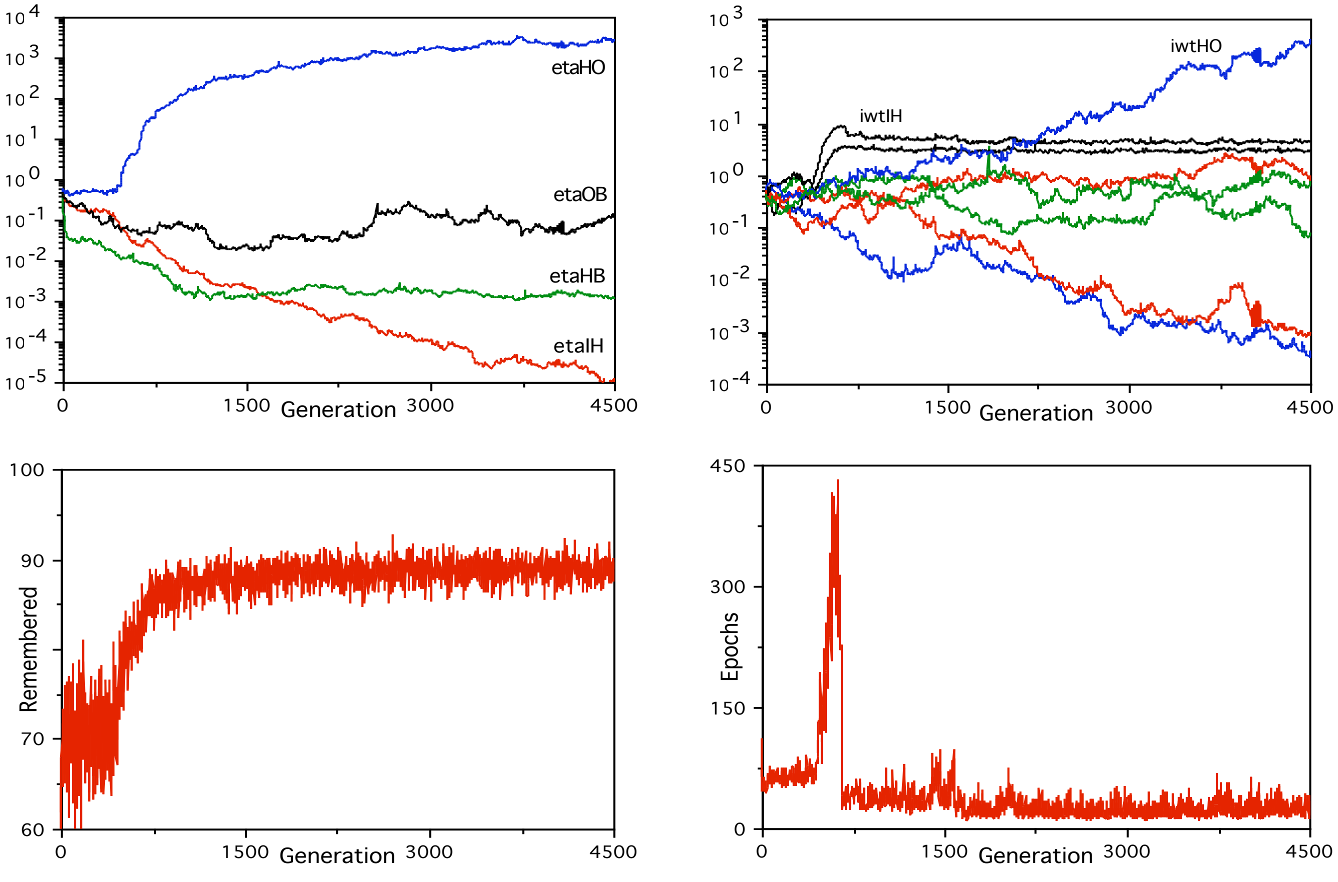

Figure 2: Simultaneous evolution of the learning rates and initial weight distributions (top), and the associated improvements in remembering performance and learning speed (bottom).

layer and bias set L is more efficient than a single parameter to control them all (Bullinaria, 2003). Evolving the four η_L resulted in a significant improvement in remembering performance from the baseline 68.7% up to around 79%.

Associated with each learning rate is a random initial weight distribution. There are several options for specifying these, such as means and standard deviations of Gaussian distributions (μ_L, σ_L), or as the lower and upper limits of uniform distributions ($-l_L$, u_L). Evolving the upper and lower limits resulted in an improvement in remembering from the 68.7% baseline up to around 76%.

A major advantage of evolutionary approaches is their

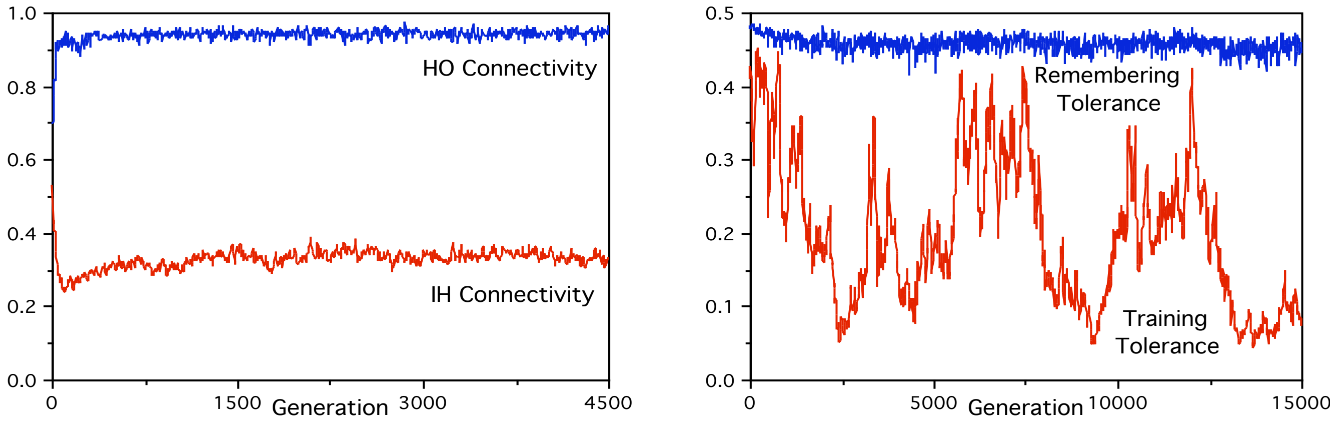

Figure 3: Evolution of the connectivity proportions between layers results in significantly reduced connectivity between the input and hidden layer (left), and evolution of better training and remembering tolerances (right).

ability to evolve simultaneously a number of parameters that have highly complex interactions. Allowing the learning rates and initial weight distributions to evolve together leads to a rather different patterns of results from when they are evolved separately. Figure 2 shows what happens. By coordinating their values, our system has now improved its remembering performance up to around 88%. Note the large differences in size between the four learning rates, and how far removed they are from the values traditionally used. It is clear that the sudden improvement in performance corresponds to a ten-fold widening of the input to hidden initial weight distributions, and the thousand fold increase in learning rate for the hidden to output connections.

It is interesting to note that the improved remembering automatically brings with it faster learning, so there is no need to build that explicitly into the evolutionary fitness function. However, as can be seen in the final graph of Figure 2, during the evolutionary history there can be periods of very slow learning. These can be avoided, if necessary, by including the number of epochs of learning in the fitness sorting. For example, instead of taking for breeding the 50% of the population that remembers best, we can take the best 60% and then reduce that to 50% by removing the slowest learners.

Two more learning parameters that one might expect to affect our results are the *Sigmoid Prime Offset* which prevents saturation and poor learning at the hidden layer, and weight decay regularization which prevents over-fitting of the training data (Bullinaria, 2003). However, if we allow these to evolve, their parameters both take on values that are so low that they have no significant effect on the learning or remembering performance.

Another factor that could be expected to influence the interference that causes forgetting is the connectivity between layers. We can evolve parameters that specify the proportion of possible connections that are used by the network, and find that proportions significantly less than one do emerge as seen on the left of Figure 3. There is almost full connectivity between the hidden and output layer, but only about one third of the input to hidden layer connections are used. However, this only resulted in about one percent improvement in the remembering performance over the evolved fully connected networks. As a check, we tried evolving just the connectivities, with all the other parameters held at the baseline values used for Figure 1, but there was a similarly small remembering improvement.

Two final parameters that could affect our results are the output error tolerances that determine when a particular output activation is deemed ‘correct’. Previously, these allowed an error on the binary targets of up to 0.1 for training, and 0.2 for testing/remembering, but other values could conceivably give better performance. We see on the right of Figure 3 what happens if we let these parameters evolve, in addition to those already considered above. The remembering tolerance comes out at just under 0.5, meaning that almost any output on the right side of 0.5 is deemed correct. This makes sense, as it renders the remembering as easy as possible without affecting the learning. The training tolerance is very erratic and fails to settle down, with no noticeable effect on the remembering performance. With the original and new patterns trained to the same tolerance, it seems to make little difference what that tolerance actually is. It turns out that evolving these parameters only produces a marginal improvement in performance (about 0.5%), and closer investigation reveals some unwanted side-effects, such as learning times which are as erratic as the training tolerances, and this can cause the learning rates to drift with an eventual deterioration in performance.

We now move on to the evolution of more sophisticated networks, allowing them to have two sets of additive weights along the lines of Hinton & Plaut (1987), with one standard set as above, and one set of ‘fast weights’ that has learning rates larger by some scale factor and a weight decay rate that prevents them having long term memory. Evolving the scale factor and decay rate, along with all the other details described above, results in a further level of

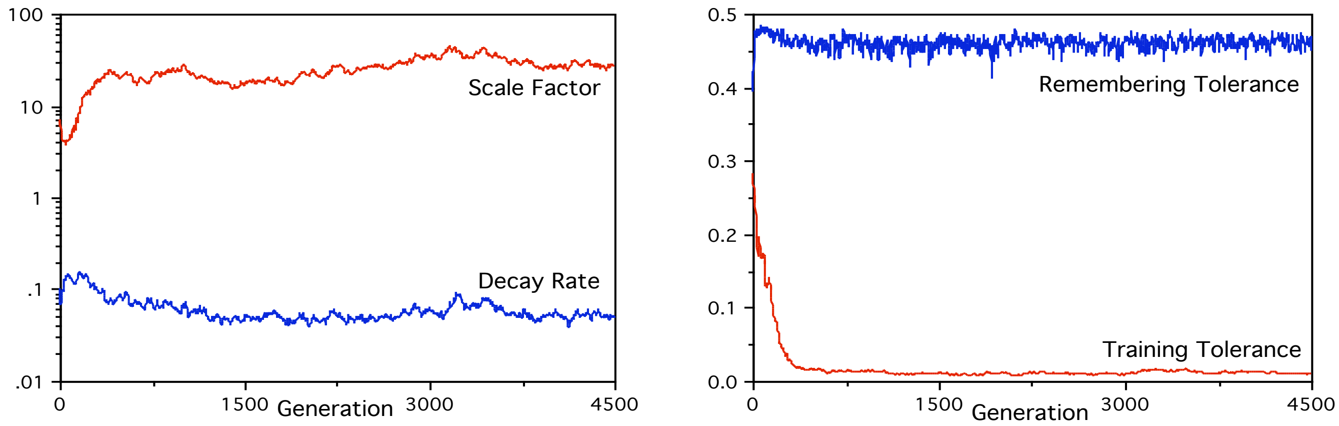

Figure 4: Evolution of the dual weight system: the learning rate scale factor and fast-weight decay rate (left), and the evolved training and testing tolerances (right) that now take on a somewhat different form to the single weight systems.

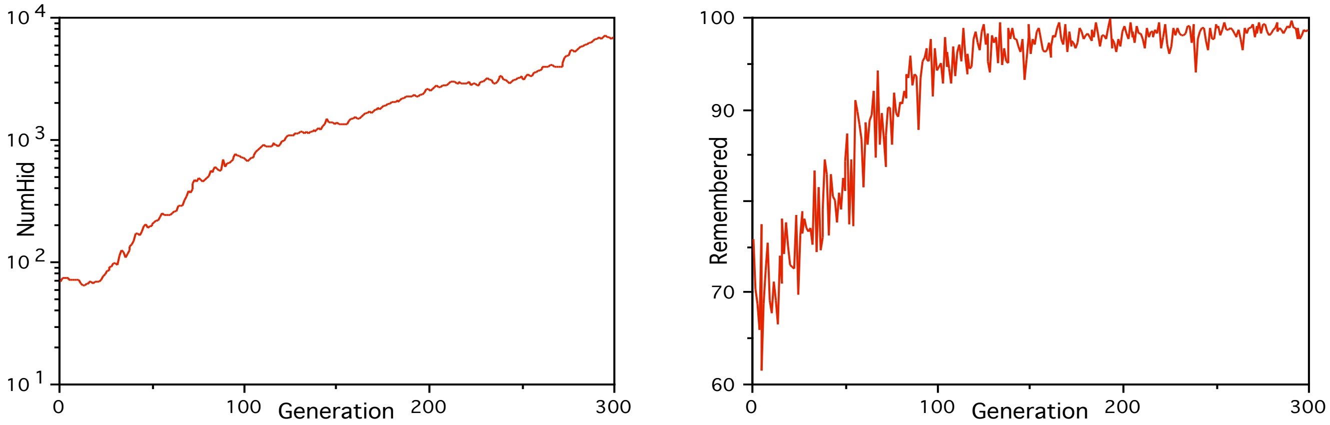

Figure 5: The final run evolving everything including the number of hidden units: the number of hidden units rising towards the maximum allowed (left), and the associated proportion remembered (right).

performance improvement up to 94.8%. Figure 4 shows the evolving parameter values that achieve this. Interestingly, the evolved training tolerance takes on a much lower value than before (around 0.01), with an associated large increase in the number of epochs training required (around 500 rather than 70). Of course, the importance of slow consolidation is not a new idea to memory modelling.

Finally, we return to the matter of the number of hidden units. Evolving all the other network parameters, with the number of hidden units fixed at 50, has led to improvements far superior to the improvement achievable by simply increasing the number of hidden units by a factor of 100. We now need to check whether allowing more hidden units can improve our evolved performance even further, or if we have reached a performance ceiling. Figure 5 shows that, even with all the other parameters evolved at the same time, the population still takes on increasing numbers of hidden units, though noticeably slower than before. Now the remembering performance rises to 98.5%, compared to the 10,000 hidden unit baseline of 78.8%, and the 94.8% maximum achieved with only 50 hidden units. Fortunately, large parameter interactions are relatively rare, though we do frequently find complex interactions with significant effects that are automatically resolved by evolutionary approaches, but are very difficult to get right 'by hand'.

So far our study was based on how many of 20 original patterns were remembered after training on 4 new patterns. Now we need to explore the extent to which the number of new patterns affects the results. Figure 6 shows what happens for our fixed 50 hidden unit case (left graph), and when the number of hidden units is free to evolve up to 10,000 (right graph). Not surprisingly, the baseline degree of forgetting as a percentage (left bar of each triple) increases with the number of new patterns. Also, consistent with our earlier results, for each number of new patterns, the baseline decreases with the number of hidden units. The important result is that for every case, evolving the network parameters, as described above, leads to a massive reduction

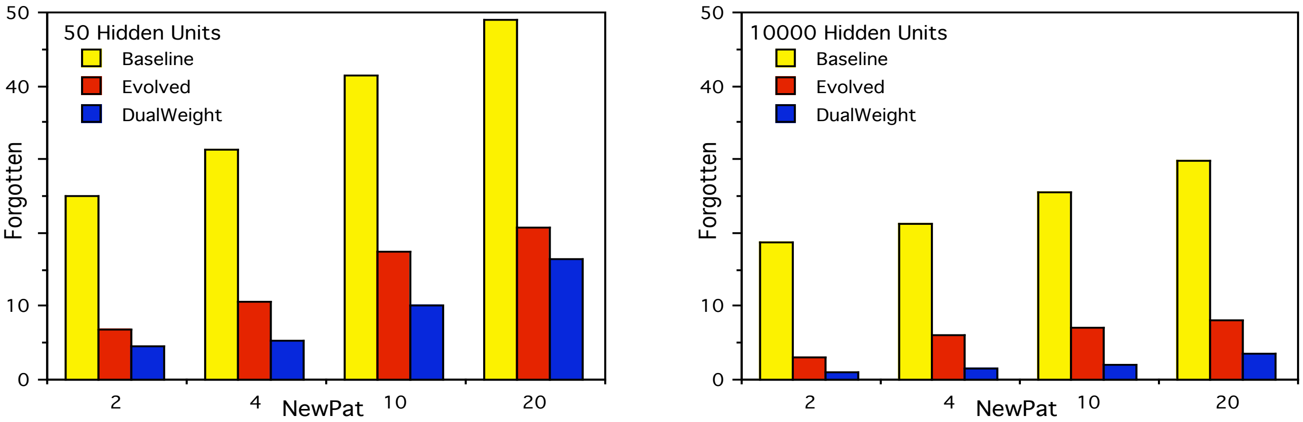

Figure 6: The more new patterns learned, the more of the original set are forgotten, but in each case the evolved networks and the evolved dual weight networks have improved remembering performance over the baseline.

in the amount of forgetting (middle bars), with even further reductions for evolved dual weight systems (right bars).

Discussion and Conclusions

We have taken ideas from natural evolution and shown, through a series of simulations, how they can significantly reduce the well known problem of catastrophic forgetting in artificial neural network systems trained on simple pattern association memory tasks. Simply evolving the traditional neural network parameters (such as numbers of hidden units, degrees of connectivity, initial weight distributions, learning rates and tolerances) leads to big improvements in remembering performance, and allowing dual weight architectures provides even further increases.

It is always difficult to relate simplified cognitive models to real brain processes. However, our results are robust across numbers of hidden units and problem complexity (i.e. number of new patterns learned), so we can expect them to scale up well. Reliably extrapolating up to human type memory tasks and brain like numbers of hidden units is very difficult, of course, but it does seems that the problem of catastrophic forgetting in neural networks may not be quite as problematic as previously thought.

The next stage will be to extend our models to include more esoteric factors, such as the dual model architectures of French (1997) and Ans & Rousset (1997), and let evolution decide on the solution it prefers. Our models already contain the necessary factors to prevent over-fitting, and preliminary results indicate that we are able to evolve appropriate parameters for networks to work as well on generalization tasks as they have here on memory tasks.

References

Ans, B. & Rousset, S. (1997). Avoiding Catastrophic Forgetting by Coupling Two Reverberating Neural Networks. *CR Academie Science Paris, Life Sciences*, 320, 989-997.

Bullinaria, J.A. (2003). Evolving Efficient Learning Algorithms for Binary Mappings. *Neural Networks*, 16, 793-800.

French, R.M. (1991). Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectionist Networks. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 173-178). Hillsdale, NJ: LEA.

French, R.M. (1997). Pseudo-Recurrent Connectionist Networks: An Approach to the “Sensitivity-Stability” Dilemma. *Connection Science*, 9, 353-379.

French, R.M. (1999). Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, 4, 128-135.

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

McClelland, J.L., McNaughton, B.L. & O’Reilly, R.C. (1995). Why There Are Complementary Learning Systems in the Hippocampus and Neocortex: Insights From the Successes and Failures of Connectionist Models of Learning and Memory. *Psychological Review, 102,* 419-457.

McCloskey, M. & Cohen, N.J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 24, 109-165.

Ratcliff, R. (1990). Connectionist Models of Recognition and Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review*, 97, 205-308.

Robins, A. (1995). Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7, 123-146.

Sharkey, N.E. & Sharkey, A.J.C. (1995). An Analysis of Catastrophic Interference. *Connection Science*, 7, 301-329.

Yao, X. (1999). Evolving Artificial Neural Networks. *Proceedings of the IEEE,* 87, 1423-1447.