EVOLVING NEURAL NETWORKS THAT SUFFER MINIMAL CATASTROPHIC FORGETTING

TEBOGO SEIPONE JOHN A. BULLINARIA
School of Computer Science, The University of Birmingham
Birmingham, B15 2TT, UK
{t.seipone, j.a.bullinaria}@cs.bham.ac.uk

Catastrophic forgetting is a well-known failing of many neural network systems whereby training on new patterns causes them to forget previously learned patterns. Humans have evolved mechanisms to minimize this problem, and in this paper we present our preliminary attempts to use simulated evolution to generate neural networks that suffer significantly less from catastrophic forgetting than traditionally formulated networks.

1. Introduction

Virtually all natural systems gradually forget what they have learned previously, particularly when they learn new information. However, with traditional artificial neural networks, the forgetting is much more catastrophic, and this proves to be a serious limitation for them (McCloskey & Cohen, 1989; Ratcliff, 1990). Some relatively complex systems, involving the interleaving of pseudo-patterns, and/or dual network architectures, have already been shown to deal with this problem quite successfully (French, 1999). However, we wish to explore the possibility of avoiding the problem within much simpler systems. Natural neural networks, such as human brains, have presumably evolved by natural selection to minimize the forgetting, and the aim of this paper is to present a preliminary investigation into using simulated evolution to see how far we can minimize the problem in artificial neural networks. The strategy employed starts by measuring how well traditional networks remember sets of input-output mappings after being trained on new items, and then explores systematically whether these can be evolved into better performing networks.

2. Evolving Neural Networks

We simulate evolution by maintaining a population of individual neural networks, each specified by a number of 'innate' parameters, and using an appropriate fitness measure to determine which to discard and which to use to create the next generation by genetic cross-over and random mutation. Repeating this process should result in increasingly fit populations.

For this study, the architecture of the networks and the learning algorithm were fixed to be standard 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 was to evolve various neural network topology and learning parameters to produce systems that suffer minimal catastrophic forgetting. Each network was initialized with different random weights from a different specific range, and then trained on the same set of initial patterns until it learned all those patterns (i.e. had all output activations within a particular tolerance of their target outputs), or until it had reached a maximum number of epochs of training. It was then trained with a new set of patterns in the same manner, and re-tested to determine how many of the original patterns it still remembered. The fittest individuals were those with the highest number remembered. The least fit half of the population was then removed, and each of the remaining individuals randomly selected a mating partner to produce one child, thus restoring the population size. The children each inherited innate characteristics (i.e. parameter values) from the range spanned by its two parents, and random Gaussian mutations were added to allow values outside that range (Bullinaria, 2003). The initial population had random innate parameters, and then for each new generation, a new global random set of training/remembering data was generated, and each individual had new random initial weights generated from their own 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 issued involved.

3. Simulation Results

For our main study we needed to fix a convenient training set of associated input and memory patterns, that was small enough for the simulations to run reasonably quickly, yet large enough to be representative. After some experimentation, we settled on 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. Throughout, all our networks had 50 hidden units, which is plenty for this task, and we present population averages over 100 individuals.

It is appropriate to start by checking the baseline performances for standard

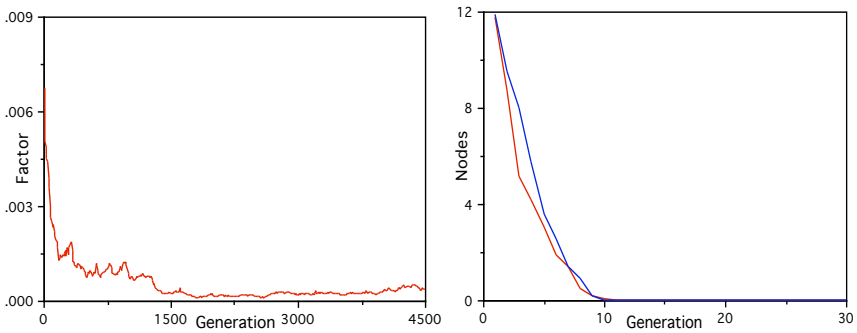

Figure 1: The parameters for hidden unit activation sharpening evolve to zero, indicating that this is not a good strategy for reducing catastrophic forgetting.

network training parameters. For 2500 individuals with different random training data sets, trained using back-propagation learning rates of 0.2 with random initial weights uniformly distributed in the range [–1, +1] the mean remembering rate was 69%. The associated standard deviation of 5% indicates the extent of the variance due to some data sets being 'easier' than others.

The first 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 a 'sharpening factor' of α times the difference. There are two variations to consider. First, when we force $N_H + N_L = N_{Hid}$ so all hidden activations get changed, the sharpening factor invariably evolves to zero, as seen on the left of Figure 1, leaving us with our standard network. If we let N_H and N_L evolved freely, they both evolve very quickly to zero, as seen on the right of Figure 1, again leaving us with our standard network. It seems that this kind of node sharpening does not really help with catastrophic forgetting 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 they chose to turn off node sharpening.

There are a number of traditional network parameters that one can evolve with the hope of improving performance. To get a feel for which are most effective, we shall consider each in turn before evolving them all at once. First we evolve the learning rates. It is now well established that allowing separate gradient descent step sizes η_L for each layer and bias set L is more efficient than a single parameter to control them all (Bullinaria, 2003). Figure 2 shows the evolution of these four parameters for our system, and the associated

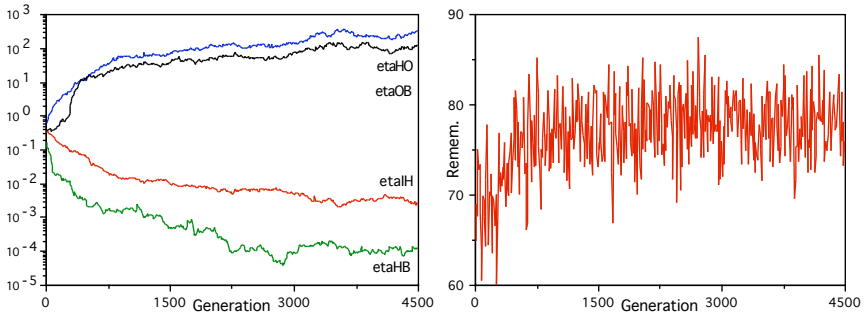

Figure 2: Evolution of the learning rates for the two weight layers and two sets of biases, and the associated improvement in remembering performance.

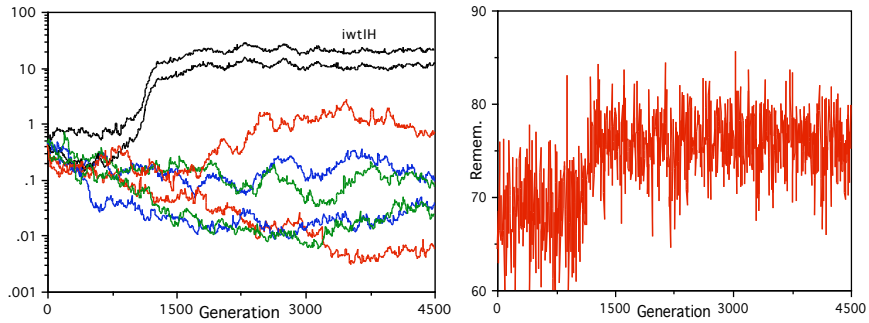

Figure 3: Evolution of the upper and lower limits of the four uniform random initial weight distributions, and the associated improvement in remembering performance.

improvement in remembering performance from the baseline 69% up to around 79%. Note the large differences in size between the four learning rates, and how far removed they are from the values traditionally used.

Associated with each learning rate is a random initial weight distribution. There are several options for specifying and evolving these, such as means and standard deviations of Gaussian distributions (μ_L, σ_L), or as lower and upper limits of uniform distributions ($-l_L$, u_L). Figure 3 shows how a set of uniform distributions evolve, and the associated improvement in remembering from the 69% baseline up to around 76%. It is clear that the sudden improvement in performance corresponds to a ten-fold widening of the input to hidden initial weight distributions.

One major advantage of the evolutionary approach is its ability to evolve simultaneously a number of parameters that interact in a complex manner. Allowing the learning rates and initial weight distributions to evolve together

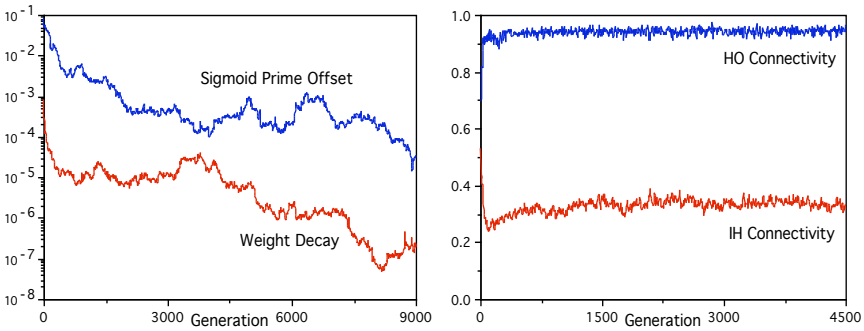

Figure 4: Evolution results in Sigmoid Prime Offset and Weight Decay parameters both taking on effectively zero values, and non-standard connectivity patterns.

leads to rather different patterns of results from when they are evolved separately. By coordinating their values, our system was able to improve its remembering performance up to around 88%. 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.

Two more learning parameters that might conceivably 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). We see on the left of Figure 4 that, 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.

Another factor, that one might expect to reduce 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 do find that proportions significantly less than one emerge as seen on the right of Figure 4. 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, there is virtually no improvement in the remembering performance. As a check, we tried evolving the connectivities with all other parameters held at the baseline values, but still there was little remembering improvement.

So far, our study has been 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 5 shows what happens with different numbers of new patterns. Not surprisingly, the baseline degree of forgetting as a percentage (left bar of each pair) increases with the number of new patterns. The important result is that for every case, evolving

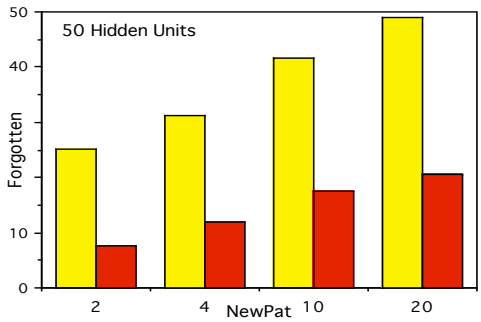

Figure 5: The more new patterns learned, the less of the original set are remembered, but in each case evolution results in improved performance over the baseline.

the network parameters, as described above, leads to a massive reduction in the amount of forgetting (right bar of each pair).

4. Conclusions

Through a systematic series of simulations, we have shown how, compared with traditionally formulated networks, an application of evolutionary techniques can significantly reduce the well known problem of catastrophic forgetting in neural network systems. We were surprised to find that evolving the learning rates and initial weight distributions alone could result in remembering performance increases that evolution of more esoteric factors cannot beat.

References

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

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

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

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