Catastrophic forgetting in simple networks: an analysis of the pseudorehearsal solution

Marcus Frean[†] and Anthony Robins[‡]

[†] School of Mathematical and Computing Sciences, Victoria University, Wellington, New Zealand
[‡] Department of Computer Science, University of Otago, Dunedin, New Zealand

E-mail: `Marcus.Frean@vuw.ac.nz` and `anthony@lethe.otago.ac.nz`

Received 27 November 1998

Abstract. Catastrophic forgetting is a major problem for sequential learning in neural networks. One very general solution to this problem, known as 'pseudorehearsal', works well in practice for nonlinear networks but has not been analysed before. This paper formalizes pseudorehearsal in linear networks. We show that the method can fail in low dimensions but is guaranteed to succeed in high dimensions under fairly general conditions. In this case an optimal version of the method is equivalent to a simple modification of the 'delta rule'.

1. Introduction

If a neural network which has learned a training set of items now learns a new item, the result is usually the 'catastrophic forgetting' of the earlier items. Installing the new 'memory' alters the function implemented by the network in a non-local way and is almost certain to alter its outputs for the original training set. In general, unless all of the patterns to be learned are repeated thousands of times in random order, learning any one of them interferes with the storage of the others. While such interference is a very general phenomenon, the term 'catastrophic forgetting' has tended to be associated with networks with a fixed architecture (as opposed to 'dynamic' networks which add or remove nodes) employing supervised learning. Several recent studies have explored this problem in backpropagation type networks [11, 7, 16, 13, 14, 4, 5, 6, 12, 9, 10, 21, 17, 18, 19]. Similar issues have been explored in Hopfield networks [15, 1, 20][§].

The most common practical solution to this problem is simply to form a new training set which includes all the old items as well as the new one, and learn this enlarged training set. This is known as 'rehearsal', and obviously requires the explicit storage of the original training set—something that has its disadvantages, notably that storage of these items is often what the network itself is supposed to do!

'Pseudorehearsal' [17] (see also [18, 6, 20]) is an alternative algorithm which, although very like conventional rehearsal, does not require the explicit storage of prior training items. Instead, *random* inputs are temporarily stored along with their associated outputs. Each time a new item is to be learned, a temporary set of such 'pseudoitems' can be created and learned alongside the genuine item. This simple algorithm works remarkably well, in that it appears

[§] Although not directly connected to the catastrophic learning literature, the 'unlearning' method [8, 3, 23, 2, 22] addresses the same issues of capacity and sequential learning tasks in Hopfield networks.

to substantially reduce interference between sequential training items while still allowing new information to be learned.

Pseudorehearsal can be interpreted in terms of a kind of ‘function fixing’ [17, 19]: pseudoitems have the effect of fixing the function being implemented by the net to its existing values at random points in the input space. Figures 1–3 show this effect for a network with one input, 20 hidden units and one output. All units are standard sigmoidal units, and the weights are trained by gradient descent of the sum of squared output errors (backpropagation). Figure 1 shows five functions arrived at from random initial weights. Figure 2 shows the functions found by the same networks after subsequently learning just one new item—these networks

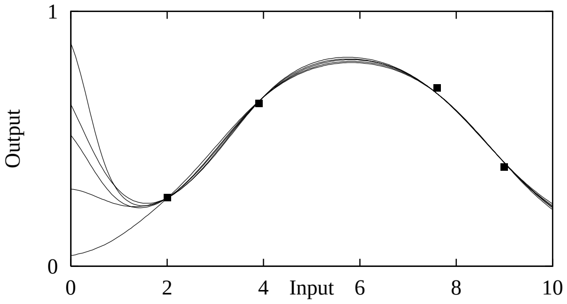

Figure 1. Examples of the function found by backpropagation fitting the initial training set (full squares).

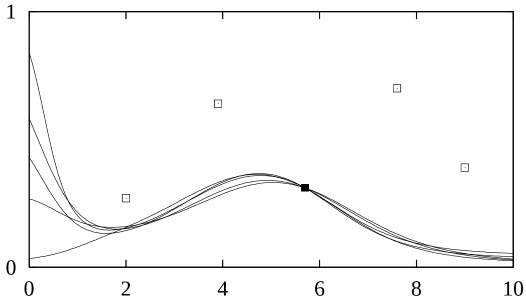

Figure 2. Subsequent naive learning of a single new item (full square). The previously learned items are shown as open squares.

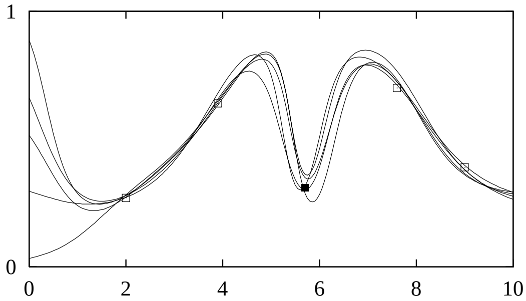

Figure 3. Learning the new item with ‘pseudorehearsal’.

now get the original items completely wrong. Figure 3 shows what happens if we learn the new genuine item along with pseudoitems: in this case a pool of 100 pseudoitems was generated from the function shown in figure 1, and one of these was chosen at random with each training epoch as the new item was learned.

The behaviour of this algorithm has not yet been understood analytically, and a sensible place to begin is to look at its effect on learning in a linear system. Thus section 2 introduces catastrophic forgetting in linear nets and introduces the formalism to be used. Sections 3 and 4 apply this to rehearsal and pseudorehearsal respectively, and section 5 uses the results to show that two intuitions about the method are in fact incorrect. Section 6 extends the analysis to the special case of high input dimension, section 7 shows a simple simulation, and section 8 presents conclusions.

2. Catastrophic forgetting in linear networks

As a first step towards understanding pseudorehearsal in quantitative terms, consider the catastrophic forgetting effect in a linear network. Assume this network has a single output and at least three input lines, meaning it can learn at least two items (input–output pairs) successfully. Because any multi-layer linear network has a single-layer equivalent, it will be sufficient to treat our network as a single unit whose output is a weighted sum of its inputs, that is: $\boldsymbol{w} \cdot \boldsymbol{b}$, where b_i is the ith input and w_i is the corresponding connection weight (vectors are shown in bold throughout this paper). The error for item B is then simply

$$\mathrm{err}_B = t_b - \boldsymbol{w} \cdot \boldsymbol{b}.$$

Let $A = (t_a, \boldsymbol{a})$ denote an input item for which the current weights give the correct output, i.e. $\mathrm{err}_A = 0$. Note that a change of $\Delta\boldsymbol{w}$ to the weights will result in an error of

$$\mathrm{err}'_A = -\Delta\boldsymbol{w} \cdot \boldsymbol{a}. \tag{1}$$

Starting from weights which get A correct then, consider learning a new item B, for which these weights give error $\mathrm{err}_B \neq 0$. The learning rule we will use is of the general form $\Delta w_i = \delta\ \boldsymbol{b}$, with δ yet to be determined†. In vector notation

$$\Delta\boldsymbol{w} = \delta\boldsymbol{b}.$$

We require that the new weights $\boldsymbol{w} + \Delta\boldsymbol{w}$ get item B correct; making the appropriate substitutions it is easy to show that $\delta = \mathrm{err}_B/|\boldsymbol{b}|^2$. Thus the weight change required to get B correct is

$$\Delta\boldsymbol{w} = \eta\, \mathrm{err}_B \boldsymbol{b} \tag{2}$$

where $\eta = 1/|\boldsymbol{b}|^2$.

This is the usual ‘delta rule’, with a learning rate of $1/|\boldsymbol{b}|^2$. In most conventional neural network applications one learns a number of new items together, and as a result the delta rule is used with a much smaller learning rate than this. However, with one item we can consider the net effect of many such iterations as a single step in the same direction (namely $\boldsymbol{b}$) of the correct size. After training then, $\mathrm{err}_B = 0$, but in general err_A (equation (1)) will no longer be zero; hence we will refer to this as ‘naive’ learning. Substituting equation (2) into (1), this is

$$\mathrm{err}_A^{\mathrm{naive}} = -\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\boldsymbol{b} \cdot \boldsymbol{b}}\, \mathrm{err}_B \tag{3}$$

where err_B refers to the *original* error on B. Clearly item A will always be ‘forgotten’ to some extent unless vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal.

† This covers virtually all the commonly used neural net learning rules.

3. Rehearsal

The conventional solution to such forgetting is simply to learn the new item alongside any old items we wish to preserve, so for this linear model consider learning B along with A, keeping the error on A close to zero. Assuming that a simple rule of the same form as before is used, the overall weight change will be

$$\Delta \boldsymbol{w} = \delta_A \boldsymbol{a} + \delta_B \boldsymbol{b} \tag{4}$$

with δ_A and δ_B to be found. We require that (i) $\mathrm{err}'_A = 0$ and (ii) $\mathrm{err}'_B = 0$. The first condition implies $\Delta \boldsymbol{w} \cdot \boldsymbol{a} = 0$, giving

$$\delta_A \boldsymbol{a} \cdot \boldsymbol{a} + \delta_B \boldsymbol{a} \cdot \boldsymbol{b} = 0. \tag{5}$$

The second implies $\boldsymbol{w}' \cdot \boldsymbol{b} = t_b$, so $\Delta \boldsymbol{w} \cdot \boldsymbol{b} = \mathrm{err}_B$, which gives

$$\delta_A \boldsymbol{a} \cdot \boldsymbol{b} + \delta_B \boldsymbol{b} \cdot \boldsymbol{b} = \mathrm{err}_B. \tag{6}$$

One can then solve equations (5) and (6) to obtain δ_A and δ_B, and substitute these into equation (4), giving

$$\Delta \boldsymbol{w} = \eta\, \mathrm{err}_B (\boldsymbol{b} - \gamma \boldsymbol{a}) \tag{7}$$

where

$$\eta = \frac{1}{|\boldsymbol{b}|^2 \sin^2 \theta_{ab}} \tag{8}$$

and

$$\gamma = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\boldsymbol{a} \cdot \boldsymbol{a}}. \tag{9}$$

This is the net weight change which must be made in learning that corrects the output for item B without corrupting item A. Note that $\Delta \boldsymbol{w} \cdot \boldsymbol{a} = 0$, as it must be if the error on item A is to remain zero. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal, the above reverts to the earlier expression, namely the delta rule for B alone.

By definition there is no forgetting of A in this full rehearsal situation.

4. Pseudorehearsal

In pseudorehearsal we generate random inputs, put them through the network and treat the outputs as if they were targets. Suppose we generate a pseudoitem X, and rehearse that instead of A. In this scenario, what is the effect on err_A of learning B? Because the weights $\boldsymbol{w}$ give the correct output for a pseudoitem by definition, the appropriate weight change in learning B is given by equation (7) with $\boldsymbol{x}$ in place of $\boldsymbol{a}$:

$$\Delta \boldsymbol{w} = \eta\, \mathrm{err}_B (\boldsymbol{b} - \gamma \boldsymbol{x}) \tag{10}$$

where

$$\eta = \frac{1}{|\boldsymbol{b}|^2 \sin^2 \theta_{xb}} \tag{11}$$

and

$$\gamma = \frac{\boldsymbol{x} \cdot \boldsymbol{b}}{\boldsymbol{x} \cdot \boldsymbol{x}}. \tag{12}$$

This leaves the output from $\boldsymbol{x}$ unchanged, while correcting the original error on B. Thus $\mathrm{err}'_B = 0$ and $\mathrm{err}'_X = 0$, but now $\mathrm{err}'_A = \Delta\, \mathrm{err}_A = -\Delta \boldsymbol{w} \cdot \boldsymbol{a}$. Substituting for $\Delta \boldsymbol{w}$ (with equation (10)) and simplifying, one finds the error incurred by using a pseudoitem instead of the correct original item to be

$$\mathrm{err}_A^{\mathrm{pseudo}} = -R\,\mathrm{err}_B \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\boldsymbol{b}\cdot\boldsymbol{b}}$$

where

$$R = \frac{1 \ - \ \cos\theta_{xa}\cos\theta_{xb}/\cos\theta_{ab}}{\sin^2\theta_{xb}}. \tag{13}$$

Since there is no forgetting at all with full rehearsal, this corresponds to $R = 0$. Learning with no rehearsal (equation (3)) corresponds to having $R = 1$, so we can say that

$$R = \frac{\mathrm{err}_A^{\mathrm{pseudo}}}{\mathrm{err}_A^{\mathrm{naive}}}.$$

Thus *pseudorehearsal alleviates catastrophic forgetting if and only if* $|R| \leqslant 1$.

5. Properties of R

R depends only on the angles between the three vectors: all dependence on their magnitudes, the targets and even the initial weights has conveniently cancelled out. We would like to get a picture of whereabouts in this 'angle space' $|R| \leqslant 1$. Figure 4 shows lines of constant R on a plot of $\cos_{xa}$ against $\cos_{xb}$, for a particular value of $\cos_{ab}$. In this figure the area outside the ellipse corresponds to combinations of angles that cannot actually occur in a Euclidean space. That is, given any three vectors in a Euclidean space and the three angles between them, the sum of any two angles is necessarily bounded below by the third angle (and above by 360° minus the third angle). When translated into cosines this restricts possible values to the area within the ellipse in figure 4.

There are two claims we expected to be able to make in the light of pseudorehearsal's success on nonlinear problems, but which the figures show cannot be true:

- It is not true that the volume of 'angle space'† for which pseudorehearsal works exceeds that for which it fails. This can be seen by noting that the $R = 1$ isocontours are two straight lines which each bisect the space (for any given $\cos\theta_{ab}$). Thus the volume of space where $R > 1$ is exactly half the space, but since there are finite areas (within the dotted ellipse) where $R < -1$ the $|R| < 1$ volume must be *less than* half.
- It is not true that if $\boldsymbol{x}$ is closer to $\boldsymbol{a}$ than to $\boldsymbol{b}$ then pseudorehearsal will work. This would correspond to all points above the rising diagonal being within the preferred range of R, which is not the case.

5.1. Orthogonal weight changes and R

At this point we should confirm that low values of R correspond to weight updates which are orthogonal to $\boldsymbol{a}$. Given equation (1) we can write R as

$$R = \frac{\boldsymbol{\Delta w}^{\mathrm{pseudo}}\cdot\boldsymbol{a}}{\boldsymbol{\Delta w}^{\mathrm{naive}}\cdot\boldsymbol{a}}.$$

On its own this does not mean that the directions of pseudorehearsal's weight changes are more orthogonal to $\boldsymbol{a}$ than those from naive learning, because the magnitudes of the two

† A more accurate term would perhaps be 'cosine space'.

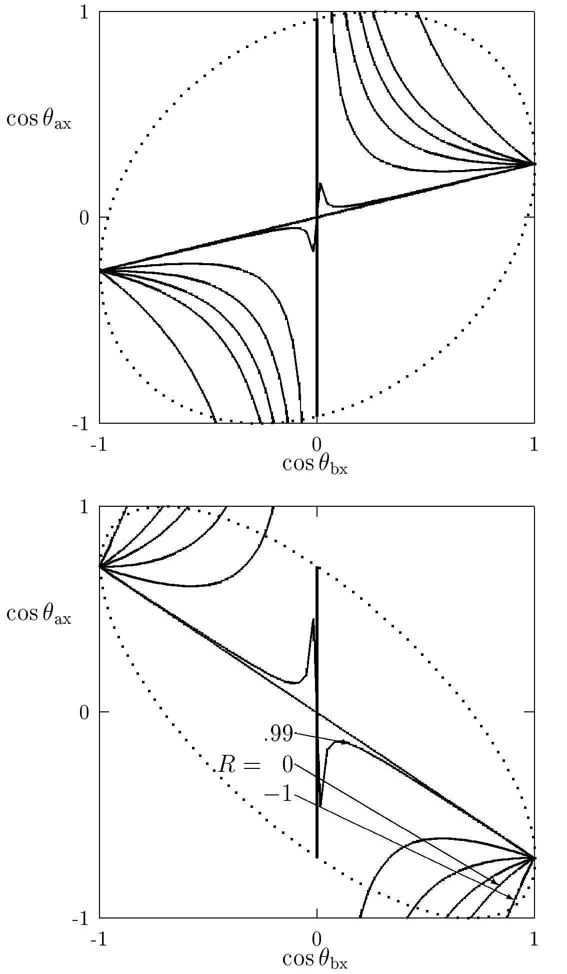

Figure 4. Isocontours of R in the space of angles between vectors. The first plot is for $\cos\theta_{ab} = 75°$ and the second is for $\cos\theta_{ab} = 135°$. R-values of -1, 0 and 0.99 are indicated by the arrows (the other values plotted being 0.25, 0.5 and 0.75). Note that only combinations of angles within the dotted ellipses are possible.

weight changes are different. However $\Delta \boldsymbol{w}^{\rm naive}$ is the *shortest* path from the point in weight space where $\mathrm{err}_A = 0$ to one where $\mathrm{err}_B = 0$, so we know that

$$|\Delta \boldsymbol{w}^{\rm naive}| \leqslant |\Delta \boldsymbol{w}^{\rm pseudo}|.$$

Thus $|R| < 1$ implies that

$$\frac{\Delta \boldsymbol{w}^{\rm naive}}{|\Delta \boldsymbol{w}^{\rm naive}|} \cdot \boldsymbol{a}$$

is larger in magnitude than

$$\frac{\Delta \boldsymbol{w}^{\rm pseudo}}{|\Delta \boldsymbol{w}^{\rm pseudo}|} \cdot \boldsymbol{a}.$$

Hence, in the cases where pseudorehearsal works, the pseudoitems lead to weight changes which are more orthogonal to the original item than with naive learning.

6. The effect of input dimensionality

The distribution of angles between randomly chosen vectors in a Euclidean space narrows as the dimensionality N grows. In this section we look at the consequences of this for pseudorehearsal. It turns out that this means pseudorehearsal will tend to work better as the dimensionality of the input grows. On the other hand we derive an 'optimal' pseudoinput, which is equivalent to altering the delta rule slightly and using no pseudoitems at all.

6.1. Angles in high dimensional spaces

It is straightforward to show that the cosine of the angle between two vectors whose elements are not correlated tends to a product of means divided by root mean squares:

$$\cos\theta_{xa} \xrightarrow{\text{large } N} \lambda_x\lambda_a$$

where

$$\lambda_u = \frac{\sum_i^N u_i/N}{\sqrt{\sum_i^N u_i^2/N}} = \frac{\text{average } u_i}{\text{rms } u_i}.$$

The expected value of the cosine converges on the above value for large N, and its variance tends to zero. This rather unintuitive result tells us that the angles between virtually all vectors are the same in high dimensional spaces. For example, if elements of $\boldsymbol{x}$ are chosen from a zero mean distribution, $\lambda_x = 0$ so the cosine is zero, meaning that $\boldsymbol{x}$ is almost certainly orthogonal to (*any* other) vector $\boldsymbol{a}$. Incidentally this means that zero-mean pseudoinputs will confer no advantage (see figure 4: if both cosines are zero, $R = 1$).

For any distribution which is not zero mean, $\cos\theta = \lambda^2$, so the angle between two vectors tends to some value other than 90°. For example, the uniform distribution between zero and one has mean $1/2$ and root mean square $1/\sqrt{3}$ so $\lambda = \sqrt{3}/2$. If two vectors are chosen in this way the cosine of the angle between them tends to $\lambda^2 = 3/4$, meaning $\theta \approx 41^\circ$. For later reference, note that $\lambda^2 \leqslant 1$ with equality only for vectors of form $(c, c, \ldots, c)$.

From now on we will need to make the simplifying assumption that A and B are drawn from similar distributions so that $\lambda_a \approx \lambda_b$. Note that this is only an assumption about the distribution of inputs, and not of targets.

6.2. A general condition under which pseudorehearsal works in high dimensions

Since the pseudoinput $\boldsymbol{x}$ is chosen at random we can approximate $\cos\theta_{xa}$ and $\cos\theta_{xb}$ by $\lambda_x\lambda_a$ and $\lambda_x\lambda_b$ respectively, in high dimensions. Assuming that $\lambda_a \approx \lambda_b$ it follows that $\cos\theta_{xa} \approx \cos\theta_{xb}$, which we will write as $\cos\hat{\theta}$. We can then write

$$R = \frac{1 - \cos^2\hat{\theta}/\cos\theta_{ab}}{1 - \cos^2\hat{\theta}}. \qquad (14)$$

Some straightforward manipulation shows that this R is between plus and minus one if (and only if)

$$\cos\theta_{ab} > \frac{\cos^2\hat{\theta}}{2 - \cos^2\hat{\theta}}$$

which is itself greater than zero. The right-hand side will tend to be a small number, so intuitively this condition is not difficult to satisfy: $\cos\theta_{ab}$ needs to be larger than a fairly small positive number for pseudorehearsal to work. In other words, if $\boldsymbol{a}$ and $\boldsymbol{b}$ are separated by an angle which is 'not too large', pseudorehearsal with even a single pseudoitem does alleviate forgetting.

6.3. A special case

Suppose that the elements of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ tend to be uncorrelated, and that the pseudoitems are chosen to have a similar value of λ. This means that for large N all three vectors will tend to be separated by $\hat{\theta}$, which easily satisfies the above condition and gives

$$\hat{R} = \frac{1}{1 + \cos\hat{\theta}}. \tag{15}$$

This is between 0.5 and 1 because the cosine (being λ^2) must be positive. Hence pseudorehearsal, although not perfect, always improves matters in this case.

6.4. Optimal pseudorehearsal

Suppose we assume only that the elements of $\boldsymbol{a}$ and $\boldsymbol{b}$ are to be uncorrelated. From equation (14), an optimal pseudoitem (one which gives $R = 0$) has $\cos^2\hat{\theta} = \cos\theta_{ab}$. We can write $\cos\theta_{ab} = \lambda_a\lambda_b$ and $\cos^2\hat{\theta} = \lambda_x\lambda_a \ \lambda_x\lambda_b$, so the optimal pseudoinput has $\lambda_x^2 = 1$. As noted above, this corresponds to an input of the form $\boldsymbol{x} = (c, c, \ldots, c)^{\mathrm{T}}$. In particular, we can now choose $c = \bar{b}$, the average value of the elements in input vector $\boldsymbol{b}$. Substituting for this in equation (10) gives the following simple learning rule, which involves no pseudoitem at all:

$$\Delta \mathrm{w_i} \propto \mathrm{err}_B(b_i - \bar{b}). \tag{16}$$

This is just the delta rule except that the direction of change $\boldsymbol{b}$ is replaced by $\boldsymbol{b} - \boldsymbol{I}\bar{b}$, where $\boldsymbol{I}$ is $(1, 1, \ldots, 1)^{\mathrm{T}}$, i.e. the input vector is effectively shifted so that its average element is zero†. This rule has $R = 0$ (under the assumptions given) and thus gives the optimal orthogonalization of weight changes to input vectors, which pseudoitems only approximate.

7. A simple simulation

In this simulation each element of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{x}$ was chosen from a unit variance Gaussian distribution, so we are dealing with the special case of section 6.3. For the simulation, targets for A and B were chosen randomly between plus and minus 100. Training consisted of first getting item A correct alone, and then training on item B using the delta rule. This latter stage was done for various numbers of pseudoitems (if this is zero we are doing 'naive' learning of B), or using the modified delta rule. This was done for various numbers of input dimensions, N. Tables 1 and 2 show the average, over 1000 trials, of $|\mathrm{err}_A|$ for each case.

Table 1. Case A: inputs with a mean of zero.

	No of pseudoitems						Modified
N	0	1	2	3	5	10	delta rule
5	25	28	34	39	—	—	28
10	14	15	16	18	22	—	15
30	8	8	8	8	9	9	8
100	4	4	4	4	4	4	4
300	2	2	2	2	2	2	2

Case A (table 1) shows just what one would expect: as the dimensionality grows vectors become more likely to be orthogonal (i.e. $\lambda^2 \approx 0$ so $\hat{\theta} = 90°$) and interference between

† Note however that it is *not* shifted in this way in calculating err_B.

items consequently decreases. Moreover in this case $\hat{R} = 1$ (equation (15)), meaning that in high dimensions pseudorehearsal can be no better than naive learning of B alone, as the table confirms. In low dimensions it is clear that pseudoitems can and do make matters substantially worse.

Table 2. Case B: inputs with a mean of one.

	No of pseudoitems						Modified
N	0	1	2	3	5	10	delta rule
5	35	33	34	37	—	—	29
10	30	23	19	19	23	—	16
30	28	19	12	10	9	10	8
100	27	18	9	7	5	5	4
300	27	18	9	6	4	3	3

For case B (table 2), $\lambda^2 \approx 1/2$ so $\hat{\theta} = 60^\circ$: vectors are unlikely to be orthogonal and hence they interfere with one another. In this case $\hat{R} = 2/3$, so in high dimensions a single pseudoitem reduces the error obtained by naive learning by about a third, as the first and second columns show. The modified delta rule deals with non-orthogonal inputs very effectively (as can be seen by comparing the final column of case B with the first column in case A), even in low dimensions. Multiple pseudoitems are substantially better than a single one, this being particularly dramatic in high dimensions. For example, in a 300 dimensional input space, using only 10 pseudoitems was enough to reduce errors to the level of the optimal rule.

8. Conclusions

Pseudorehearsal is an intruiging solution to the problem of catastrophic forgetting in neural networks. In nonlinear networks this method can be seen in terms of 'function fixing': pseudoitems act to resist global changes to the function implemented by the network. In linear networks where the dimensionality of the input is high, we have shown that pseudorehearsal does reduce catastrophic forgetting. This means (see section 5.1) that pseudoitems have the effect of 'orthogonalizing' the weight changes with respect to previously stored items. This orthogonality is the exact corollary of 'function fixing': weight changes that result in learning of a new item do not tend to change the outputs of the net given other inputs.

For small N pseudorehearsal fails in that it is more likely to increase forgetting than to alleviate it. For large N the method provably does work under fairly general conditions on the distribution of inputs. Although our analysis deals with a single pseudoitem, simulations confirm that multiple items increases this effect†. We have shown that in high dimensions a simple modification of the delta rule is equivalent to pseudorehearsal with a single 'optimal' item: simulations indicate that this rule works well even in fairly low dimensions.

References

[1] Burgess N, Shapiro J L and Moore M A 1991 Neural network models of list learning *Network* **2** 399–422

[2] Christos G A 1996 Investigation of the Crick–Mitchison reverse-learning dream sleep hypothesis in a dynamical setting *Neural Networks* **9** 427–34

[3] Crick F and Mitchison G 1983 The function of dream sleep *Nature* **304** 111–4

† If the number of pseudoitems approaches the dimensionality of the inputs, the system becomes overconstrained and would fail to learn the new item, let alone preserve older ones.

[4] French R M 1992 Semi-distributed representations and catastrophic forgetting in connectionist networks *Connection Sci.* **4** 365–77

[5] French R M 1990 Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic forgetting *Proc. 6th Ann. Cognitive Science Soc. Conf.* (Hillsdale, NJ: Earlbaum) pp 335–40

[6] French R M 1997 Interactive connectionist networks: an approach to the 'sensitivity–stability' dilemma *Connection Sci.* **9** 353–80

[7] Hetherington P A and Seidenberg M S 1989 Is there catastrophic interference in neural networks? *Proc. 11th Ann. Cognitive Science Soc. Conf.* (Hillsdale, NJ: Earlbaum) pp 26–33

[8] Hopfield J J, Feinstein D I and Palmer R G 1983 Unlearning has a stabilising effect in collective memories *Nature* **304** 158–9

[9] Lewandowsky S 1991 Gradual unlearning and catastrophic interference: a comparison of distributed architectures *Relating Theory and Data: Essays on Human Memory in Honour of Bennet B. Murdok* ed W E Hockley and S Lewandowsky (Hillsdale, NJ: Earlbaum) pp 445–76

[10] Lewandowsky S and Li S 1995 Catastrophic interference in neural networks: causes, solutions and data *Interference and Inhibition in Cognition* ed F N Dempster and C Brainerd (San Diego, CA: Academic) pp 329–61

[11] McCloskey M and Cohen N J 1989 Catastrophic Interference in connectionist networks: the sequential learning problem *The Psychology of Learning and Motivation* vol 23, ed G H Bower (New York: Academic) pp 109–64.

[12] McRae K and Hetherington P A 1993 Catastrophic interference is eliminated in pretrained networks *Proc. 15th Ann. Mtg of the Cognitive Science Soc.* (Hillsdale, NJ: Earlbaum) pp 723–8

[13] Murre J M J 1992 *Learning and Categorization in Modular Neural Networks* (Hillsdale, NJ: Earlbaum)

[14] Murre J M J 1995 Hypertransfer in neural networks *Connection Sci.* **8** 249–58

[15] Nadal J P, Toulouse G, Changeux J P and Dehaene S 1986 Networks of formal neurons and memory palimpsets *Europhys. Lett.* **1** 535–42

[16] Ratcliff R 1990 Connectionist models of recognition memory: constraints imposed by learning and forgetting functions *Psychol. Rev.* **97** 285–308

[17] Robins A V 1995 Catastrophic interference, rehearsal and pseudorehearsal *Connection Sci.* **7** 123–46

[18] Robins A V 1996 Consolidation in neural networks and the sleeping brain *Connection Sci.* **8** 259–75

[19] Robins A V and Frean M R 1998 Learning and generalization in a stable network *Progress in Connectionist-Based Information Systems: Proc. 1997 Conf. on Neural Information Processing and Intelligent Information Systems* ed N Kasabov *et al* (Singapore: Springer) pp 314–7

[20] Robins A and McCallum S 1998 Catastrophic forgetting and the pseudorehearsal solution in Hopfield type networks *Connection Sci.* **7** 121–35

[21] Sharkey N E and Sharkey A J C 1995 An analysis of catastrophic interference *Connection Sci.* **7** 301–29

[22] van Hemmen J 1997 Hebbian learning, its correlation catastrophe, and unlearning *Network* **8** 1–17

[23] Wimbauer S, Klemmer N and van Hemmen J 1994 Universality of unlearning *Neural Networks* **7** 261–70