Memory for Multiplication Facts

Richard Dallaway*
School of Cognitive & Computing Sciences
University of Sussex
Brighton BN1 9QH, UK
richardd@cogs.susx.ac.uk

Abstract

It takes approximately one second for an adult to respond to the problem "7 × 8" The results of that second are well documented, and there are a number of competing theories attempting to explain the phenomena [Campbell & Graham 1985; Ashcroft 1987; Siegler 1988]. However, there are few fully articulated models available to test specific assumption [McCloskey, Harley, & Sokol 1991]. This paper presents a connectionist account of mental multiplication which models adult reaction time and error patterns. The phenomenon is viewed as spreading activation between stimulus digits and target products, and is implemented by a multilayered network augmented with a version of the "cascade" equations [McClelland 1979]. Simulations are performed to mimic Campbell & Graham's [1985] experiments measuring adults' memory for single-digit multiplication. A surprisingly small number assumptions are needed to replicate the results found in the psychological literature—fewer than some (less explicit) theories presuppose.

Phenomena

When asked to recall answers to two digit multiplication problems "as quickly and accurately as possible" [Campbell & Graham 1985], both children and adults exhibit well documented patterns of behaviour. In general, response times (RTs) increase across the multiplication tables: problems in the nine times table tend to take longer to answer than problems in the two times table. However, this "problem size effect" has plenty of exceptions (e.g., the five times table is much faster than its position would suggest—see figure 1). In addition, "tie" problems (2×2, 3×3 etc.) are recalled relatively quickly. Campbell & Graham [1985] found that adults under mild time pressure make errors at the rate of 7.65 per cent, and 92.6 per cent of those errors fall into the following five categories (after McCloskey et al. [1991]):

- Operand errors, for which the erroneous product is correct for a problem that shares a digit (operand) with the presented problem (e.g., $6 \times 4 = 36$, because the problem shares 4 with $9 \times 4 = 36$).
- Close operand errors, a subclass of operand errors, where the erroneous product is also close in magnitude to the correct product. That is, for the problem $a \times b$, the error will often be correct for the problem $(a \pm 2) \times b$ or $a \times (b \pm 2)$ (e.g., $6 \times 4 = 28$). This phenomenon is referred to as the "operand distance effect"
- Frequent product errors, where the error is one of the five products 12, 16, 18, 24 or 36.
- Table errors, where the erroneous product is the correct answer to some problem in the range 2×2 to 9×9, but the problem does not share any digits with the presented problem (e.g., 6×4=15).
- Operation errors, where the error to $a \times b$ is correct for $a + b$.

Despite being drilled on the multiplication tables at school, children and adults make these systematic slips in recall. The problem is to produce a model which has correctly learnt the multiplication tables, yet can make slips when recalling answers. Given the observations on the types of erroneous responses, and the RT for correct responses, what assumptions must be made to account for these phenomena? The model presented here suggests that the initial skew in the frequency and order of presentation of multiplication facts [Campbell 1987, p. 118] is one of the important factors.

*Thanks to Harry Barrow & David Young. Funded by the SERC in conjunction with Integral Solutions Ltd. Simulations were performed using a modified version of the McClelland & Rumelhart [1988] *bp* program, and POPLOG POP-11.

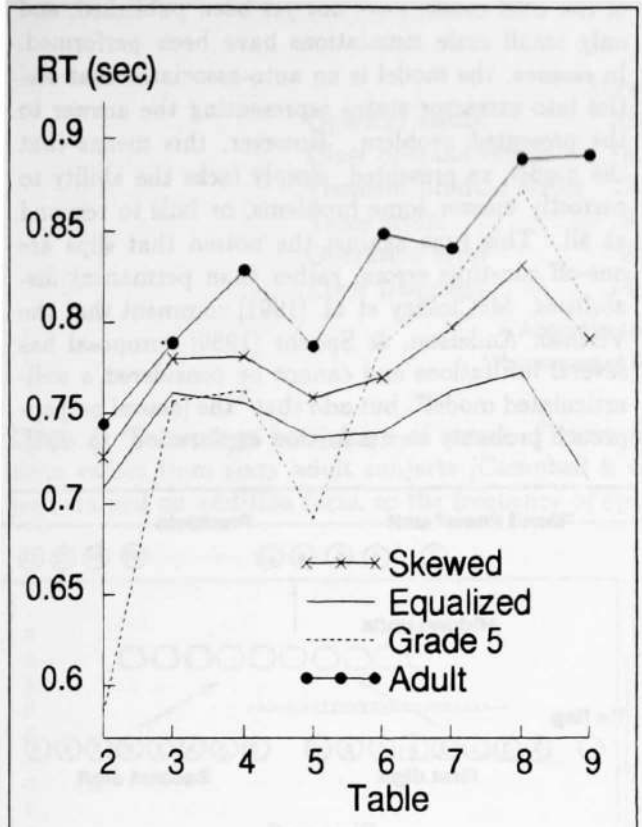

Figure 1: Plot of mean correct RT per multiplication table collapsed over operand order for mean RT of: 60 adults [Campbell & Graham 1985, app. A]; 26 children in grade 5, RT scaled down from a range of 1.19–2.97 seconds to fit graph [ibid. app. B]; 20 networks trained on skewed frequencies; and, the same 20 networks after continued training on uniform frequencies. The RT for both networks has been scaled by the same amount.

Architecture of the model

The structure of the network is shown in figure 2. This architecture has evolved in a number of stages since it was first used as a subnetwork in a sequential network for long addition (and later long multiplication). Initially the output layer was divided into "tens" and "units", and by adding a simple RT measure it was found that the network produced a prominent dip in the RT curve for the five times table. This effect was increased by training sequentially through the tables, but the network did not produce the kinds of mistakes reported by Campbell & Graham [1985]. Changing the output layer to a representation of products, and using a coarse encoding of the input digits produced more realistic errors.

The current network is trained on all the problems 2×2 through 9×9 in a random order using back-propagation. The two digits that comprise a problem are coarse encoded on the two sets of eight input units, with the activation decaying exponentially from the presented digit (e.g., when encoding "5", the input vector would contain 1.0 for the five unit, and 0.5 for the four and six units, and so on). For tie problems, an additional tie bit is set to 1.0. Without this, the tie problems were consistently among the slowest problems. The tie bit can be thought of as reflecting the perceptual distinctiveness of tie problems. Activation flows through a hidden layer of ten units to the output layer. There is one output unit per product type plus a "don't know" unit. The network is trained to activate one output unit per problem (a one-of-N encoding).

During training the presentation frequency of each pattern is linearly skewed in favour of the smaller problems (relative frequency of 1.0 for 2×2 to 0.1 for 9×9, based on correct product). Although small problems do occur more frequently in textbooks, there is no reason to believe this skew continues into adulthood [McCloskey et al. 1991, p. 328]. Hence, after training to an error criterion (total sum squared, TSS) of 0.05 on the skewed training set (taking approximately 8 000 epochs), the network is trained for a further 20 000 epochs with equal frequencies (reaching a mean TSS of 0.005). At the end of training both the "skewed" networks and "equalized" networks correctly solve all problems. An initial worry was that the skew would lead the networks into a local minima from which the task could not be completed. To avoid this possibility, a low learning rate of 0.01 was used during training (momentum was 0.9).

The skew was produced by storing the relative frequency (between zero and one) of a problem alongside the problem in the training set. When a problem was presented to the network, the weight error derivative was multiplied by the relative frequency value for that pattern. (This can be thought of as providing each input pattern with a different learning rate.) This method allowed accurate control over the presentation frequencies, without duplicating entries in the training set.

The "cascade" activation equation [McClelland & Rumelhart 1988, p. 153] is used to simulate the spread of activation in the network. Each unit's activation is allowed to build up over time:

$$\mathrm{net}_i(t) = k \sum_j w_{ij} a_j(t) + (1 - k)\ \mathrm{net}_i(t-1),$$

where k is the cascade rate which determines the rate with which activation builds up, w is the weight ma-

trix, and $a_j(t)$ is the activation of unit j at time t. For the simulations described here, $k = 0.05$. The net_i is passed through a logistic squashing function to produce the activation value, a_i. The response values are taken to be the normalized activation values (the sum of the output layer activity is 1.0).

McClelland & Rumelhart [1988] point out that the asymptotic activation of units under the cascade equation is the same as that reached after a standard feed-forward pass. Hence, the network is trained without the cascade equation (or with $k = 1$, if you prefer), and then the equation is switched on to monitor the network's behaviour during recall.

At the start of cascade processing the initial state of the network is the state that results from processing an all-zeros input pattern. This gives a common starting point for all problems. The network is trained to activate the "don't know" unit for an all-zeros input. Figure 3 is a time plot of output activation using the cascade equations.

Simulations

Method

On each trial (presentation of a problem) the network randomly selects a threshold between 0.4 and 0.9. Processing then starts from the all-zeros ("don't know") state, and proceeds until a product unit exceeds the threshold. The RT (number of cascade steps) is recorded for a correct response, and erroneous responses are classified into the five categories itemized above. The network is presented with each of the 64 problems 50 times, and the mean correct RT is recorded. This is repeated with 20 different networks (different initial random weights).

Given enough time (usually no more than 50 cascade steps), the network will produce the correct response for all 64 problems. For example, figure 3 shows the response of a network to the problem 3×8. After the "don't know" unit has decayed, the unit representing 27 becomes active until the network settles into the correct state, 24. This is a demonstration of the operand distance effect, but there is slight activation of other products: $3 \times 7 = 21$, $2 \times 8 = 16$, $4 \times 8 = 32$, $3 \times 3 = 9$, and $2 \times 7 = 14$.

With a high threshold the networks will reliably produce the correct response to a problem. However, early in processing erroneous products are active (e.g., 27 in figure 3), and with a low threshold these errors are reported. Note that this is rather different to previous connectionist (Brain-state-in-a-box, BSB) model of mental arithmetic [Viscuso 1989; Anderson, Spoehr, & Bennett 1991]. The full details of the BSB model have not yet been published, and only small scale simulations have been performed. In essence, the model is an auto-associator that settles into attractor states representing the answer to the presented problem. However, this means that the model, as presented, simply lacks the ability to correctly answer some problems, or fails to respond at all. This runs against the notion that slips are one-off run-time errors, rather than permanent disabilities. McCloskey et al. [1991] comment that the Viscuso, Anderson, & Spoehr [1989] "proposal has several limitations and cannot be considered a well-articulated model", but add that "the [neural net] approach probably merits further exploration" [p. 395].

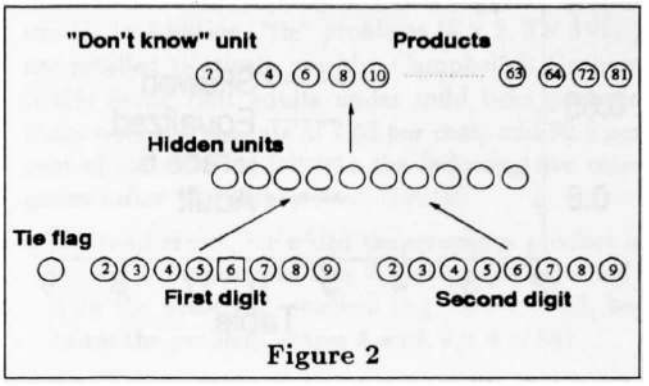

Figure 2

Results

The mean RTs plotted in figure 1 show some of the basic features of the problem size effect. For the skewed networks the RT correlates $r = 0.36$ ($p = 0.0018$) with adult RT [Campbell & Graham 1985]. This falls to $r = 0.19$ ($p = 0.063$) after substantial training on the equalized patterns. Note that the RTs have reduced and flattened out for the equalized network, which is just what is expected after continued practice [Campbell & Graham 1985, p. 349]. The obvious feature of the RT plot is the drop in RT for the nine times table. Children in grades 3 to 5 respond faster to $9\times$ than $8\times$ problems [Campbell & Graham 1985], but this levels out for adults.

The inclusion of a ties unit is necessary to ensure that ties are among the fastest problems. Implicit in this is an assumption that there is something perceptual about ties which results in a flagged encoding—perhaps the effect of being taught the notion of "same" and "different". The RTs of 6 (out of 8) of the tie problems were below the mean RT for their table, increasing to 7 ties for the equalized networks (6×6 remaining above the mean for the six times table).

Table 1 shows the error percentages of the net-

	Networks		Adults
	Skewed	Equalized	
Operand errors	90.04	86.51	79.1
Close operand errors	78.98	73.75	76.8*
Frequent product errors	27.76	23.68	30.6
Table errors	9.74	13.49	13.5
Operation error	3.98	3.22	1.7*
Error frequency	14.10	18.64	7.65

* Approximate percentage.

†Percentage of operand errors.

Table 1: Percentage breakdown of errors. Figures are mean values from twenty different networks, and mean values from sixty adult subjects [Campbell & Graham 1985, app. A]. Note that the model has not been trained on addition facts, so the frequency of operation errors is coincidental.

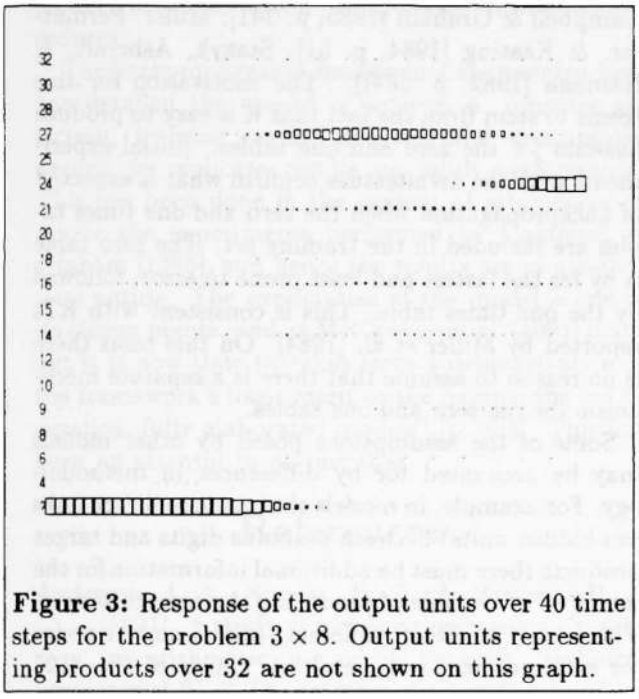

Figure 3: Response of the output units over 40 time steps for the problem 3×8. Output units representing products over 32 are not shown on this graph.

works compared to those of adults. Both sets of networks have error distributions that are similar to that of adults, and there is little difference between the skewed and equalized networks.

It should be noted that human subjects sometimes respond with a number that is not a correct product for any of the problems 2×2 to 9×9 (e.g., $2 \times 3 = 5$). The current network cannot produce non-table errors. However, Campbell & Graham [1985] report that only 7.4 per cent of errors are of this kind. (An account of non-table errors might begin by augmenting the network with a tens and units read-out layer.)

A further point of interest is the correlation be-

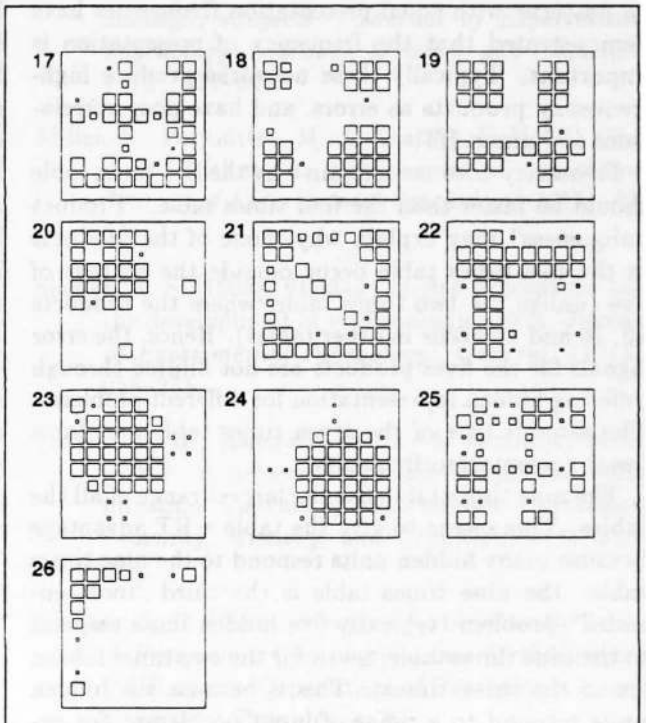

Figure 4: Hidden unit activation for one network. Each large rectangle represents one hidden unit. Within each rectangle, the size of the smaller rectangles represents the activation of the hidden unit to a particular problem. Each large square mimics the multiplication table (top-left for 2×2, and bottom-right for 9×9). For example, unit 22 responds to problems in the three and four times tables.

tween problem error rate and correct RT. Campbell [1987, p. 110] reports a correlation of 0.93 for adults. For the skewed and equalized networks $r = 0.74$ and $r = 0.76$ respectively. It is not obvious that any model would necessarily predict that slower problems produce more errors.

Analysis

RT depends on the net input to a unit, and this can be increased by having some large (or many small) weights. Although there is no easy way to determine why certain weights develop, some of the factors involved can be described.

The presentation frequency of a problem and product should have a strong effect on the weights: those problems seen more often should develop larger weights. Simulations with networks trained only on patterns with equal presentation frequencies have demonstrated that the frequency of presentation is important. Typically these networks produce high-frequency products as errors, and have poor correlations to human RT.

Frequency does not explain why the five times table should be faster than the four times table. "Product uniqueness" may explain why: none of the products in the five times table occur outside the context of five (unlike the two times table, where the products 12, 16 and 18 occur in other tables). Hence, the error signals for the fives products are not diluted through differing hidden representation for different problems. The same is true of the seven times table, but for a lower presentation frequency.

The nine times table has the largest range of all the tables. This seems to give the table a RT advantage because many hidden units respond to the nine times table: the nine times table is the third "most encoded" problem (typically five hidden units respond to the nine times table: seven for the two times tables; six to the three times). This is because the hidden units respond to a range of input problems. For example, figure 4 shows that unit 26 responds to small products; unit 23 to medium products; and unit 24 responds to larger products.

The hidden units' preference for responding to bands of inputs explains the mechanism behind the operand distance effect. Hidden units' activities change smoothly during the course of processing, but at differing rates. This change affects groups of related products, and due to the overlap in encoding (e.g., between unit 23 and 24 in figure 3), some hidden units may force incorrect products to exceed threshold.

Discussion

Apart from the training frequency skew, the other main assumption of the model is the coarse coding of the input pattern. The importance of this assumption has been demonstrated by simulations using a one-of-N input encoding (the same encoding that was used for the outputs). The results of those simulations produced comparable RT correlations, but poor error distributions. The assumption is that the coarse encoding is due to general knowledge of number (perhaps from counting).

This study has focused on mean adult performance on the problems 2×2 to 9×9 because these problems have detailed published results. There are persistent statements in the literature that zero and one times tables are governed by procedural rules (e.g., Campbell & Graham [1985, p. 341]; Miller, Permutter, & Keating [1984, p. 51]; Stazyk, Ashcraft, & Hamann [1982, p. 334]). The motivation for this seems to stem from the fact that it is easy to produce answers for the zero and one tables. Initial experiments with the architecture confirm what is expected of backpropagation when the zero and one times tables are included in the training set. The zero table is by far the fastest and least prone to error, followed by the one times table. This is consistent with RTs reported by Miller et al. [1984]. On this basis there is no reason to assume that there is a separate mechanism for the zero and one tables.

Some of the assumptions posed by other models may be accounted for by differences in methodology. For example, in models that assume direct links (no hidden units) between stimulus digits and target products there must be additional information for the model to be capable of producing the correct answer. There must be either: different (token) answer nodes for each problem (e.g., multiple copies of the "12" node for 2×6 and 3×4 as used by Ashcroft [1987]); or input nodes representing whole problems (e.g. a "3×4" input node as in Campbell & Graham [1985]); or both [Siegler 1988].

However, other assumptions were not found to be needed in this model. For example, there was no need for explicitly learning incorrect associations, as suggested by both Siegler [1988] and Campbell & Graham [1985]. Nor was there need for connections between product units (Campbell & Graham [1985] and Ashcroft [1987]), nor connections from general "magnitude" units as used by Campbell & Graham [1985]. These models have been criticised by McCloskey et al. [1991] for not specifying the rationale for these additional connections.

Of course, there are a number of shortcomings to

the model presented here. There is no empirical evidence to suggest that adults are exposed to a skew in the frequency of multiplication problems, and this was modelled by further training the skewed networks on equal frequency problems. Although the RTs for the equalized networks diverge from the adult RTs, they retain the basic features of the problem size effect and error distributions. One conclusion that can be drawn from this is that it is quite possible for the effect of training on skewed problems to continue to be felt even after a significant period of training on non-skewed problems.

As it stands the model makes no attempt to account for a number of important aspects of arithmetic. Future directions for this work could focus on: modelling single digit addition; the role of back-up (counting) procedures: error priming; and the model's position in long (multi-digit) arithmetic procedures.

The backpropagation cascade model presented here has detailed the spread of activation, response selection, training regime and minimal assumptions needed to replicate results on adult performance. This has been done in the context of attempting to mimic the experiments performed by Campbell & Graham [1985], and hence the results are of a statistical nature. The explicitness of the model is one of its strong points, and as McCloskey et al. [1991] point out it is now time to "shift from a demonstration of the framework's basic merit to the hammering out of detailed, fully elaborated models" [p. 394]. This has been an attempt to do just that.

References

Anderson, J. A., Spoehr, K. T., & Bennett, D. J. [1991]. A study of numerical perversity: Teaching arithmetic to a neural network. Technical report 91-3, Department of Cognitive and Linguistic Sciences, Brown University. To appear in Levine, D. S. and Aparicio, M. (eds) *Neural Networks for Knowledge Representation and Inference*, Lawrence Erlbaum Associates, Hillsdate, NJ.

Ashcroft, M. H. [1987]. Children's knowledge of simple arithmetic: A developmental model and simulation. In Bisanz, J., Brainerd, C. J., & Kail, R., eds., *Formal Methods in Developmental Psychology*, chapter 9, pp. 302–338. Springer-Verlag, New York.

Campbell, J. I. D. [1987]. The role of associative interference in learning and retrieving arithmetic facts. In Sloboda, J. A., & Rogers, D., eds., *Cognitive Processes in Mathematics*, pp. 107–122. Clarendon Press, Oxford.

Campbell, J. I. D., & Graham, D. J. [1985]. Mental multiplication skill: Structure, process, and acquisition. *Canadian Journal of Psychology*, *39*(2), 338–366.

McClelland, J. L. [1979]. On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review*, *86*(4), 287–330.

McClelland, J. L., & Rumelhart, D. E. [1988]. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. MIT Press, Cambridge, MA.

McCloskey, M., Harley, W., & Sokol, S. M. [1991]. Models of arithmetic fact retrieval: An evaluation in light of findings from normal and brain-damaged subjects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(3), 377–397.

Miller, K., Permutter, M., & Keating, D. [1984]. Cognitive arithmetic: Comparison of operations. *Journal of Experimental Psychology*, *10*(1), 46–60.

Siegler, R. S. [1988]. Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General*, *117*(3), 258–275.

Stazyk, E. H., Ashcraft, M. H., & Hamann, M. S. [1982]. A network approach to mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*(4), 320–335.

Viscuso, S. R. [1989]. *Memory for Arithmetic Facts: A Perspective Gained from Two Methodologies*. Ph.D. thesis, Department of Psychology, Brown University, Providence, RI.

Viscuso, S. R., Anderson, J. A., & Spoehr, K. T. [1989]. Representing simple arithmetic in neural networks. In Tiberghien, G., ed., *Advances in Cognitive Science*, Vol. 2. Ellis Horwood, Chichester, UK.