Psychologica Belgica, Vol.36, No#, pages ***–***. ©

MULTIPLICATION NUMBER FACTS: MODELING HUMAN PERFORMANCE WITH CONNECTIONIST NETWORKS

Betty Edelman, Hervé Abdi†, Dominique Valentin

School of Human Development, The University of Texas at Dallas, Richardson, TX 75083–0688, U.S.A., †also Université de Bourgogne, 21004, Dijon, France.

ABSTRACT

Three connectionist models of human performance on simple multiplication number facts, commonly called "times tables," are reviewed. Also, human data from normal subjects and brain-damaged patients, which constrain these models, are presented. These human data include the problem size effect, error effects, priming effects, use of strategies and rules, and number representation. The connectionist models presented are: a simple auto-associator (J.A. Anderson's Brain-State-in-a-Box), a standard back-propagation model, and McCloskey and Lindemann's MATHNET. The review of human data and connectionist models of memory retrieval provides some insight into the strengths of, differences between, and challenges for, this approach to computational modeling. Particular attention is paid to the representation of number used by these models, and a related ability to generalize learning.

1. Introduction

For many years experimental psychologists employed simple arithmetic problems solely as distractor tasks. However, with the reawakening of interest in cognition, researchers have begun to use basic number facts to probe the nature of memory representation and retrieval processes. Number facts, especially sums and products of single digit numbers, have several advantages as experimental stimuli (e.g., practically everyone has learned addition and multiplication tables). The response time

Correspondence should be addressed to Betty Edelman, School of Human Development, The University of Texas at Dallas, MS: GR4.1, Richardson, TX 75083–0688, email bedelman@utdallas.edu.

The authors thank Richard Golden, Guy Lories, and an anonymous reviewer for helpful comments on previous drafts of this paper.

to answer various problems can be examined, and time differences provide important clues to possible mental organizations and processes. Also, even adults, make occasional mistakes. Anderson (1995), commenting on how the human brain does arithmetic, uses the catchy phrase, "Not only is it slow, it is also inaccurate" (p. 586).

We present a review of three connectionist models designed to predict human performance on multiplication number facts: J. A. Anderson's Brain-State-in-a-Box (BSB), a standard back-propagation model, and McCloskey and Lindermann's MATHNET. These models, by no means the only ones addressing this area of cognition, were selected to highlight some strengths of, and challenges for, the connectionist approach, and the important contribution of stimulus representation. Other well developed models, such as Campbell's network-interference (see Campbell & Oliphant, 1992), are not included because they have not been explicitly implemented in the connectionist framework at this time, and excellent reviews exist (e.g., Ashcraft, 1992; Graham & Campbell, 1992; McCloskey, Harley, & Sokol, 1991).

Prior to describing the connectionist models, it is necessary to identify the human subject phenomena that constrain their performance. The first section presents data on multiplication number fact performance across the life span, including single case studies of brain-damaged patients (see also Ashcraft, 1992; Dehaene, 1992; McCloskey, Harley, et al., 1991). In this brief review, we restrict our attention to data relevant to the models presented. The second section describes the models, including architecture, data representation, operation, and application to multiplication number facts.

2. Data from Human Subjects

Many studies have examined the number fact behavior of normal children and adults, and also the disruption of number fact performance in brain damaged patients. The subjects being tested have typically been asked to perform two types of tasks, *verification* and *production*. In a *verification* task, subjects are shown a problem (e.g., $3 \times 5 = 16$) and are asked to respond "true" if they believe the answer to be correct, or "false" otherwise. In a *production* task, subjects are shown only the operands and asked to supply the answer (e.g., $5 \times 4 = ?$). For both types of tasks, response time and errors are recorded and analyzed as dependent variables. Several researchers have posited that verification consists of production followed by an additional stage of comparison to the given answer (e.g., Ashcraft, 1987; Parkman, 1972). However, others maintain an alternative view that the stated answer in a verification task alters the process of retrieval and therefore, possibly, the outcome (Zbrodoff & Logan, 1990).

2.1. *Problem size effect*

The problem size effect is the most robust of all the phenomena and is the most widely reported (Ashcraft, 1992). It occurs when subjects produce longer response

times and higher error rates for large operand problems. For example, problems such as 9×7 are more difficult than 3×4. The problem size effect has been found across the entire age span (Allen, Ashcraft, & Weber, 1992; Ashcraft & Fierman, 1982; Geary & Wiley, 1991; Hamann & Ashcraft, 1986; Koshmider & Ashcraft, 1991) and across the number facts of the four operations: addition (e.g., Ashcraft & Battaglia, 1978; Groen & Parkman, 1972; Parkman & Groen, 1971), subtraction (e.g., Siegler, 1987; Woods, Resnick, & Groen, 1975), multiplication (e.g., Campbell & Graham, 1985; Parkman, 1972; Stazyk, Ashcraft, & Hamann, 1982) and division (e.g., Campbell, 1985). Both production (e.g., Miller, Perlmutter, & Keating, 1984) and verification (e.g., Geary, Widaman, & Little, 1986) tasks show the effect.

There are some exceptions to the problem size effect, for example, "ties" like 4×4 (Miller et al., 1984; Parkman, 1972). For these problems, response time for normal subjects is either constant, or increases only somewhat with larger operands. Although, in brain-damaged patients, impairment for multiplication facts with large operands is generally greater than that for problems with small operands (McCloskey, Harley et al., 1991), impairment has been shown to be non-uniform (McCloskey, Caramazza, & Basili, 1985; Warrington, 1982).

There are two theories commonly cited as the cause of the problem size effect. The *frequency theory* (Ashcraft, 1992) proposes that, because small problems occur more often, frequency (i.e., practice) effects yield stronger memory traces. The *order of presentation theory*, based on the fact that small problems are typically learned first, posits proactive inhibition impedes learning when larger facts are presented, giving rise to order effects (Campbell & Graham, 1985). A recent study of adults, showing that practice attenuates, but does not eliminate, the problem size effect, is more consistent with the frequency theory than with the order of presentation theory (Fendrich, Healy, & Bourne, 1993).

2.2. ***Error Effects***

The errors made by human subjects on multiplication number facts can be classified into several types, which are described below using the terminology of McCloskey, Harley, et al. (1991).

2.2.1. *Operand Errors*

The most common error is termed an operand error or an operand related error (Ashcraft, 1992). In this type of error the incorrect answer given is correct for another problem that shares an operand (e.g., $8 \times 7 = 40$ is given, and 40 is the correct answer to 8×5). Campbell and Graham (1985) found that this type of error accounted for over 79% of the errors made by subjects. Also, 85% of the errors made by brain-damaged patient PS were of this type (Sokol, McCloskey, Cohen, & Aliminosa, 1991).

When an operand error is made, the incorrect answer not only shares one operand, but usually, the other operand is only off by a magnitude of 1 or 2 (e.g., $9 \times 7 = 72$). This phenomenon, termed the *operand distance effect*, was present in 95% of the

operand errors made by patient PS and 60% of those made by patient GE (Sokol et al., 1991).

2.2.2. *Table Errors*

In a table error, the answer does not share an operand with a correct answer, but the answer given does reside in the multiplication table (e.g., $6 \times 9 = 56$, when 56 is the correct response to 7×8). Campbell and Graham (1985) found that this type of error accounted for 13% of the multiplication errors made by adults.

2.2.3. *Operation Errors*

Operation errors, also termed *cross-operation confusions* (Ashcraft, 1992), occur when the answer given is correct for a problem having the same two operands, but a different operation (e.g., $9 \times 8 = 17$). These cross-operation confusions increase response time in production (e.g., Campbell, 1987a; Campbell & Clark, 1989; Campbell & Graham, 1985; Miller et al., 1984) and verification tasks (e.g., Ashcraft & Battaglia, 1978; Winkelman & Schmidt, 1974; Zbrodoff & Logan, 1986). In one study, this mistake accounted for 24% of the errors made by normal adults (Miller et al., 1984). This effect caused 69% of the errors of brain-damaged patient GE (Sokol et al., 1991).

2.2.4. *Non-table Errors*

A non-table error occurs when the answer given is not an answer to any problem in the multiplication tables (e.g., $4 \times 9 = 38$) and is not frequently committed by normal adults. Campbell and Graham (1985) found that this type of mistake accounted for only 7% of all errors made by subjects. This error is also rarely committed by brain-damaged patients (Sokol et al., 1991).

2.3. ***Priming Effects***

Multiplication problem response time and accuracy can be affected by priming. Correct responses to recently practiced arithmetic facts are given quicker than other correct responses (Campbell, 1987b; Stazyk et al., 1982). Erroneous responses have also been found to relate to both positive and negative priming (Campbell & Clark, 1989). *Positive error priming* is shown when previous answers (usually given two or three trials before) occur as errors with a probability higher than chance. *Negative error priming* is shown when a particular incorrect response is less likely to occur if it is the answer given for an immediately preceeding trial. Errors of *operand intrusion*, when an operand is erroneously repeated in an answer (e.g., $4 \times 8 = 28$), may also arise from priming (Campbell & Clark, 1992).

2.4. ***Strategies and Rules***

Multiplication, unlike addition, is not easily accomplished by a counting approach, but other rules and strategies can facilitate production or verification. The generalized rule of commutativity greatly reduces the number of distinct problems to be memorized: If the answer to 8×6 is forgotten, the answer to 6×8 can be used.

There are several ways to verify the plausibility of a stated answer, for example when one of the operands is 5, the product always ends in 0 or 5.

Generalizations for multiplication by 0 and 1 are perhaps the most frequently applied (i.e., $0 \times N = 0$ and $1 \times N = N$). Problems including the operands of 0 or 1 have been shown to exhibit patterns of response time and error effects different from other problems (Aiken & Williams, 1973; Ashcraft, 1982, 1992; Parkman, 1972; Stazyk et al., 1982). Neuropsychological data lend support to the theory that multiplication by zero is stored as a rule. PS, a brain-damaged patient, performed quite differently on zero problems than on non-zero problems (McCloskey, Aliminosa, & Sokol, 1991; Sokol et al., 1991).

2.5. *Representation of Number*

The mental representation of numerical quantities may provide bases for the effects described above. Moyer and Landauer (1967) first suggested that an analog magnitude representation is included in the concept of number. The *symbolic distance effect*, often called the *split effect*, demonstrated by a decrease in time to choose the larger of a pair of digits as the difference between them increases, was interpreted to imply magnitude representation. Moyer and Landauer observed the similarity between the results from numerical comparison tasks and the time for discriminations along perceptual continua (e.g., brightness or weight) as characterized by Weber-Fechner laws. This observation led them to suggest that numbers are internally represented as magnitude analogs that are, approximately, a logarithmic function of digit size. Further studies of normal and brain-damaged subjects support a duality of number representation: analog and digital. Magnitude is commonly considered to be analog in nature, and compressive as numbers increase in size (Banks & Hill, 1974; Dehaene & Cohen, 1991; Michie, 1985; Todd, Barber, & Jones, 1987; Warrington, 1982). This number scale compression could make larger numbers less discriminable, and thus contribute to the problem size effect.

Other investigations of the number concept have uncovered further complexity. The application of multidimensional scaling to data from number similarity judgments has yielded dimensions of "magnitude," "odd versus even parity," and "prime versus composite numbers" (Shepard, Kilpatrick, & Cunningham, 1975), and also shown native language influences (Miller, 1992). Campbell (1994) found interaction effects between two number formats (Arabic digit and English number-word) and problem size effects, operation errrors, and operand-intrusion errors. Inter-trial error priming effects were also differentially affected by number format. These results were interpreted as suggesting "notation-dependent" activation of number facts, and "interpenetration of number reading and number-fact retrieval processes."

3. Connectionist Models of Multiplication Number Fact Retrieval

All three models reviewed in this section adhere to the connectionist tenet of distributed representation and employ a nonlinear retrieval mechanism. However, they vary in motivation, scope, and emphasis. Anderson's BSB applies the simplest of

structures and algorithms to learning number facts, thus showing the power of neural networks to learn, and generalize, through massive parallelism. The back-propagation model of McCloskey and Cohen (1989) serves mainly to highlight a learning problem of some models. McCloskey and Lindemann's MATHNET uses a probabilistic approach to simulate number fact retrieval within the framework of a broader modular model of numerical processing.

We will pay particular attention to the number representation of the models. If the data representation is not appropriate, a network does not learn well, even with the most powerful learning rules. Determining a scheme to transform a problem such as $4 \times 7 = 28$ into a model representation is not trivial. Numbers perceived as being similar should be represented so as to be similar in structure, and numbers perceived as different should be dissimilar in structure. If this is not the case, the mapping to be learned by the network becomes quite complicated. Moreover, the structure of the pattern (typically expressed as a vector) should reflect some psychological validity.

3.1. *The Anderson Brain-State-in-a-Box Model*

J. A. Anderson and his associates (Anderson, 1992; Anderson, Spoehr, & Bennett, 1994; Viscuso, 1989; Viscuso, Anderson, & Spoehr, 1989) have employed a simple neural network model (i.e., the BSB) to generate many of the basic number fact effects of human arithmetic learning. The model represents data with large high-dimensionality vectors, which allow for manipulating the amount of correlation between stimuli, therefore making it possible to use simple learning and retrieval algorithms.

The BSB couples a classic associative memory with nonlinear retrieval dynamics[1]. The associative memory links a set of "neuron-like" units to itself, and is therefore termed an auto-associator. Such systems are often used in pattern recognition applications, because associating a pattern to itself allows the regeneration of a complete pattern response as output when only a partial or a degraded copy of a stored pattern is input. This property of the auto-associator can be applied to multiplication number facts by first training the network to associate the pattern representing a problem (e.g., $6 \times 7 = 42$) to itself in a learning phase. In a later test phase, a partial stimulus (e.g., $6 \times 7 =$) can be presented as input and the network will complete the pattern, thus supplying the answer.

3.1.1. *Architecture of the Model*

The architecture of the BSB consists of one layer of units that connect to themselves as illustrated in Figure 1. The connection weights between units are bidirectional and symmetric. The units may be fully connected, as illustrated in the figure, or only partially connected by randomly setting some of the weights to 0. Anderson and his colleagues have frequently used 50%, or less, connectivity. Partial connectivity

[1] For neural network aficionados, it is a nonlinear energy-minimizing feedback auto-associator.

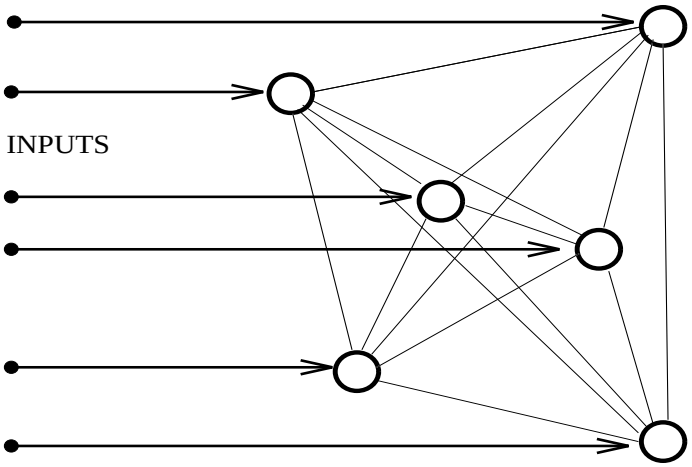

FIGURE 1. The architecture of an auto-associator comprised of six units. Each unit is connected to all other units providing feedback when input is presented

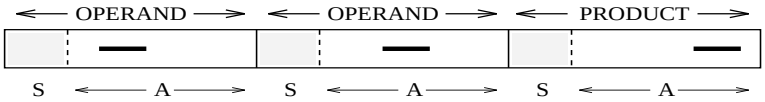

FIGURE 2. Organization of a state vector for a BSB multiplication problem. The representation of each number consists of two kinds of information, symbolic (S) and analog magnitude (A). The bar moves from left to right as the magnitude of a number increases. Bars for numbers close in magnitude will overlap in their position.

does not qualitatively affect the network performance, but reduces computational time, and provides some increase in biological realism (Anderson, 1995).

3.1.2. *Multiplication Problem Representation*

The problem representation, called "state vector coding," is a hybrid scheme first described by Viscuso (1989) and Viscuso et al. (1989) that includes different facets of number representation. The general layout of a state vector, which represents a multiplication number fact, is illustrated in Figure 2. Each operand and the product are constructed of two parts, one part being an arbitrary abstract symbolic representation and the other part being a sensory representation that is roughly analog. This analog representation is, according to Anderson (1992, 1995), responsible for much of the BSB performance.

The analog portion of the BSB state vector, provided for each operand and the product, reflects the magnitude of the number with a bar consisting of a series of

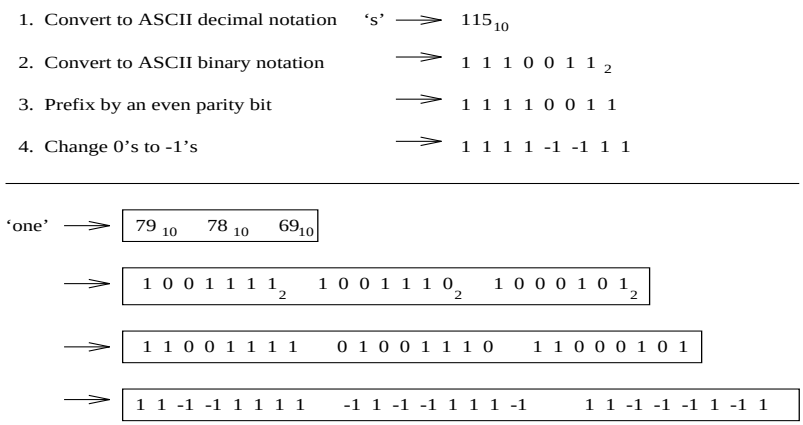

FIGURE 3. Example of BSB symbolic representation.

+1's, placed within a field of −1's. The position of the bar shifts from left to right as the number to be represented increases in magnitude. The bars are positioned within the magnitude field in a staggered fashion so that those representing numbers of similar magnitude will partially overlap. The width of the bars can be adjusted to represent the correlation between nearby numbers (Anderson, 1995).

The symbolic portion of the representation maps the spelling of a number word (e.g., four, twenty) into −1's and +1's. This is accomplished by first converting each letter into a numeric value as it would appear in ASCII code. For example, as illustrated in the top of Figure 3, the letter 's' becomes 115 in decimal notation. The ASCII value is then expressed binarily as a string of seven 1's and 0's, a parity bit[2] is added to the front of the string, and finally all 0's are changed to −1's. These eight character patterns for each letter are concatenated following the spelling order of the number word. This process is detailed at the bottom of Figure 3 for the symbolic representation of the word "one." This particular representation is, in fact, arbitrary: Other symbolic coding schemes could be used to achieve the same effect.

The Anderson symbolic representation has the advantage of being easily decoded on the output side, in a reverse fashion, affording human interpretation of the network response to a problem. Activation values exceeding a threshold in absolute value are set to the closest limit, either +1 or −1. This is done only in the final decoding of the network response and serves to eliminate inconsistent and noisy answers. The −1's are then changed back to 0's, and the rightmost 7 binary digits

[2] Strictly speaking, only seven bits are required to represent the ASCII code with binary values. Using eight bits is essentially a matter of convenience. According to Anderson (1995, p. 200), one consequence of adding a parity bit is to make possible the construction "of orthogonal bytes with pairs of characters."

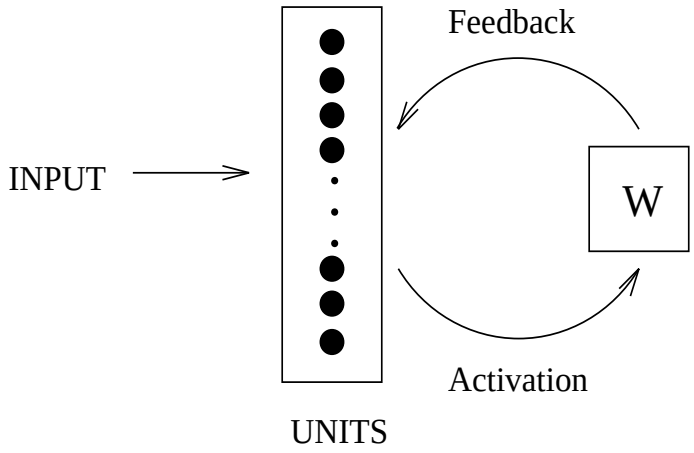

FIGURE 4. Operation of the BSB. The activity of the units is multiplied by the weights in the connection matrix **W** and the resulting feedback is added to the activity level of each connected unit.

for each letter are converted to the equivalent ASCII letter (or other symbol). Thus, the BSB output is converted into a "spelled-out" response.

3.1.3. *Operation of the Model*

The model contains two phases: a learning phase and a testing (i.e., retrieval) phase.

Learning phase. First, the learning phase establishes the weights for each connection between units of the auto-associative memory. Using either standard Hebbian or Widrow-Hoff techniques, each representation of a multiplication number fact is associated with itself (see Anderson, Silverstein, Ritz, & Jones, 1977; Abdi, 1994a for details).

Retrieval phase. After the learning phase has completed the creation of the connection matrix, the testing phase is initiated. In this phase, an incomplete or degraded version of a learned pattern, or a novel (i.e., never learned) pattern, is fed to the network by setting each unit to one element of the pattern. Each unit calculates a response by combining its initial activation with the feedback arising from "filtering" the activation of all connected units through the weighted connections (see Figure 4). The units compute the sum of the products of all inputs times their respective weighted connection values, and adjust their current activation level by incorporating this sum. Both the unit activation, and the feedback, may be scaled by a decay constant. Optionally, a multiple of the initial stimulus is also incorporated (termed *clamping* of the input). The entire summation process (shown by the arrows in Figure 4) constitutes one cycle (i.e., time step) of retrieval. This feedback cycle is done repeatedly, and the final state represented by the activation values for all units is taken as the response of the model.

A limit function prevents unit activation from growing boundlessly. If unit activation exceeds an upper limit (e.g., +1), or falls below a lower limit (e.g., −1), it is set to that limit. This clipping forces the network state to stay within a hypercube, hence the "in-a-Box" part of the name. When all units reach their limit, a stable state has been attained.

Under certain conditions, with repeated feedback cycles, the activation of each unit is drawn toward the limits of the clipping function, so that the state approaches a vertex of the hypercube (see Golden, 1986). Therefore, the network response can be assessed in terms of both speed (i.e., response time) and accuracy. Response time can be measured by the number of feedback cycles needed to reach stability, and accuracy by the similarity of the output to the correct answer (e.g., by taking the correlation or the cosine between the correct or "taught" vector pattern and the reconstructed vector). The BSB function is described in more mathematical detail in Appendix A.

3.1.4. *Application to Multiplication Number Fact Recall*

Viscuso (1989) and Viscuso et al. (1989) applied the BSB model to the production and verification of qualitative[3] multiplication, for which an approximation, or estimation, of the answer is given (e.g., $2 \times 6 \approx 10$ and $8 \times 6 \approx 50$).[4] Each problem of the qualitative multiplication table was constructed using a 640 element hybrid representation, part magnitude and part symbolic, as previously described. Results were correct for only over 50% of the problems, but comparable to human performance for associative interference, practice effects, and the symbolic distance effect. The model did well on zeros problems, appearing to extract this "rule." A confusion error matrix showed that, in general, wrong answers clustered around the correct magnitude. Moreover, errors were of an associative nature, comparable to those of human performance as observed by Campbell and Graham (1985) and Norem and Knight (1930). Both human subjects and the model provided large products for problems containing a nine as an operand, and frequently confused problems with sixes and sevens as operands.

Frequency effects. To simulate the influence of practice effects (i.e., frequency of exposure to certain problems) on the model, Viscuso (1989) and Viscuso et al. (1989) biased the connection matrix to give more importance to particular problems. This was accomplished by modifying the Widrow-Hoff learning rule for particular problems by multiplying their representations by a scaling constant set to 1.5 (i.e., using vectors of values +1.5 and −1.5 instead of +1 and −1 in the learning phase). Problems having the largest products, specifically those in the 6 to 9 times tables, were treated this way. In the retrieval phase, wrong answers to other problems reflected this bias and were larger in magnitude than they were without this treatment (e.g., $2 \times 3 = 20$). These results are similar to a practice

[3] Qualitative multiplication has fewer product responses than quantitative multiplication. The BSB is best suited to mapping a large amount of input to a small number of categories.

[4] The application was as described by Equation 1 , with $\gamma = 0.9$ and $\delta = 0$ (no clamping of the input).

effect shown with human subjects: Answers to extensively practiced problems are more likely to be given as incorrect answers to other problems (Campbell, 1987b; Norem & Knight, 1930).

Order effects. In all the simulations described above, stimuli were presented repeatedly in the learning phase, but always in randomly mixed order. This is not the way most children are presented with number facts. Addition is usually presented prior to multiplication, and multiplication tables are typically presented and practiced in order from small to large. An attempt to simulate order effects with the BSB illustrates an interesting problem. Viscuso (1989) notes that when the 30 learning presentations for one problem were presented sequentially to the Widrow-Hoff learning algorithm, a very strong "recency interference" effect arose (an effect now often termed *catastrophic interference*). For example, if the last problem learned was $9 \times 9 = 80$, the network tended to give the answer 80 to many other problems. The tendency for the sequentially trained BSB to corrupt previously learned material with new material is much stronger than the sequential and positive error priming effects observed in human subjects.

Other Effects. In more recent simulations (Anderson et al., 1994) the BSB performance improves to 70% average correctness, due to changes in the model representation, and a slight modification of operation. The number of elements in the problem representation is increased to 1266, with 422 elements allotted to each operand and the product. The symbolic part for each number is 72 elements, leaving 350 elements for the analog portion, which contains a 78 element bar. The bar consists of mostly $+1$'s (about 13 -1's are randomly placed to provide some noise) and is positioned in a field of all zeros. Bars representing numbers of similar magnitude overlap in a compressed manner, roughly logarithmic, as magnitude increases. During the retrieval phase, the BSB algorithm is applied as before, except that the operands of the problem are always clamped, and that an activation threshold is used to keep the zero parts of the vector at zero.

Anderson et al. (1994) succesfully simulated the effects of symbolic distance and priming, and with somewhat less success the generalization of learning. Priming, implemented by increasing connection weights for certain problems previously answered correctly, subsequently decreased the number of iterations (i.e., response time) to answer these problems. These priming effects were generalized, to a lesser degree, to related problems sharing operands or answers. To test its ability to generalize to the "rule" of commutativity the network was trained with a problem in one order (e.g., 7×9) and then tested with the opposite order (e.g., 9×7). The network response to the new order was correct 50% of the time, and always twice as slow as the response to the learned problem. Thus, although generalization was accomplished to some degree, it was far from perfect. However, when a problem was omitted from the training set and this "novel" problem was tested, the response was typically correct, and required only a few more cycles than the responses to the learned problems.

3.1.5. *Discussion*

The strength of the BSB model is to show that a simple and understandable neural network can simulate number fact effects for multiplication. Specifically, the model simulates the human phenomena of associative interference and problem size in production, the split effect for verification, priming effects, and some ability to generalize learning. The representation used for the problems, with its emphasis on magnitude relationships, is the key to many of these effects, and suggests an explanation for the problem size effect other than order of learning and frequency. However, while not discounting the achievements of the model, it may be instructive to consider its weaknesses.

The problem size effect, which is extremely robust in human subjects, is not very robust in the BSB model (Anderson et al., 1994). The effect was best simulated when the bar coding was compressed logarithmically. Although frequency or practice effects were generally simulated successfully, a simulation of order effects produced an overwhelming recency effect. The question is: Is this problem unique to the BSB or is it characteristic of other neural network models? McCloskey and Cohen (1989) further investigate the problem of what they term "catastrophic interference" when serial learning is simulated by a three layer back-propagation neural network.

3.2. ***The McCloskey and Cohen Back-propagation Model of Arithmetic Learning***

To investigate the effects of sequential learning on a back-propagation neural network, McCloskey and Cohen (1989) modeled the learning of number facts. What is most interesting about this model is not its successful predictions, but its failure to simulate the human ability to sequentially acquire and retain information. Although newly learned associations may degrade performance on previously learned material (i.e., *retroactive interference*), the decline is not "catastrophic." In this model, sequential presentation of stimuli during learning results in network "retrograde amnesia."

3.2.1. *Architecture of the Model*

The standard back-propagation model used by McCloskey and Cohen (1989) contains three layers consisting of 28 input units, 50 hidden units, and 24 output units (see Abdi, 1994b for more details about this type of architecture).

3.2.2. *Multiplication Problem Representation*

McCloskey and Cohen (1989) used a "coarse-coded" representation for numbers. As illustrated in Figure 5, each number between 0 and 9 inclusive is represented by 12 units. In a manner similar to that used by the analog portion of the BSB representation, three consecutive units are set to +1 with the remaining nine having a value of 0. As the size of a number increases, the bar of +1's moves to the right. The input stimulus vector is constructed by concatenating these number representations, one for each operand, with a four unit operation representation.

The correct response for an output vector is formed by the concatenation of two number representations, one for the tens digit of the answer and the other for the ones digit.

3.2.3. *Operation of the Model*

This standard three layer back-propagation model has a learning phase and a retrieval phase. After the learning phase has established weights for the connections, retrieval is tested. The pattern of activity levels of the output units is interpreted as the answer to the problem.

3.2.4. *Application to Multiplication Number Fact Recall*

When the network was trained concurrently on the 200 single digit addition and multiplication number facts, recall was virtually perfect. Such high performance is not unusual for back-propagation models. Next, McCloskey and Cohen (1989) investigated the effects of sequential training and compared their model predictions to the results obtained in a classic learning study that showed a clear effect of *retroactive interference*. In this study by Barnes and Underwood (1959), subjects learned lists of eight paired nonsense syllables and adjectives using an A–B/A–C paradigm. After perfectly learning a first list pairing nonsense syllable A with adjective B, they were presented with a second learning list, which paired the same nonsense syllable A with a different adjective C. Subjects were tested after various numbers of practice trials (1, 5, 10 or 20) on the second list and asked to recall both the B and the C response when shown A. Figure 6 illustrates how recall of the two lists changed as a function of practice trials on the second list. As performance on the second list improved, performance on the associations of the first list steadily declined to about 50 percent recall. However, note that although the subjects demonstrated some forgetting of the first list, they did retain some information.

McCloskey and Cohen (1989) first trained the network to respond correctly to 17 ones addition facts (i.e., $1+1$ through $9+1$, and $1+2$ through $1+9$). Training on the twos addition facts (i.e., $2+1$ through $2+9$, and $1+2$ through $9+2$) was then initiated. Recall of both sets of number facts was tested after each learning iteration for the twos facts. A result of this testing is illustrated in Figure 7, showing rapid and almost complete "forgetting" of the ones facts, with performance declining from 100% to 57% after only one learning trial.

McCloskey and Cohen (1989) report on a series of experimental manipulations of the network parameters in an attempt to isolate the cause of this problem: changing the number of hidden units, changing the learning rate parameter, overtraining on the first list, freezing of some weights, changing of target activation values, and changing representation of the stimuli. None of these manipulations caused the results to approach human performance. McCloskey and Cohen, expressing a somewhat pessimistic view of the capability of neural networks to model human cognition, conclude that networks of this type can not handle a sequential training regimen, because new learning will always modify the weight configuration and thus change the solution space. This modified space may no longer be compatible with

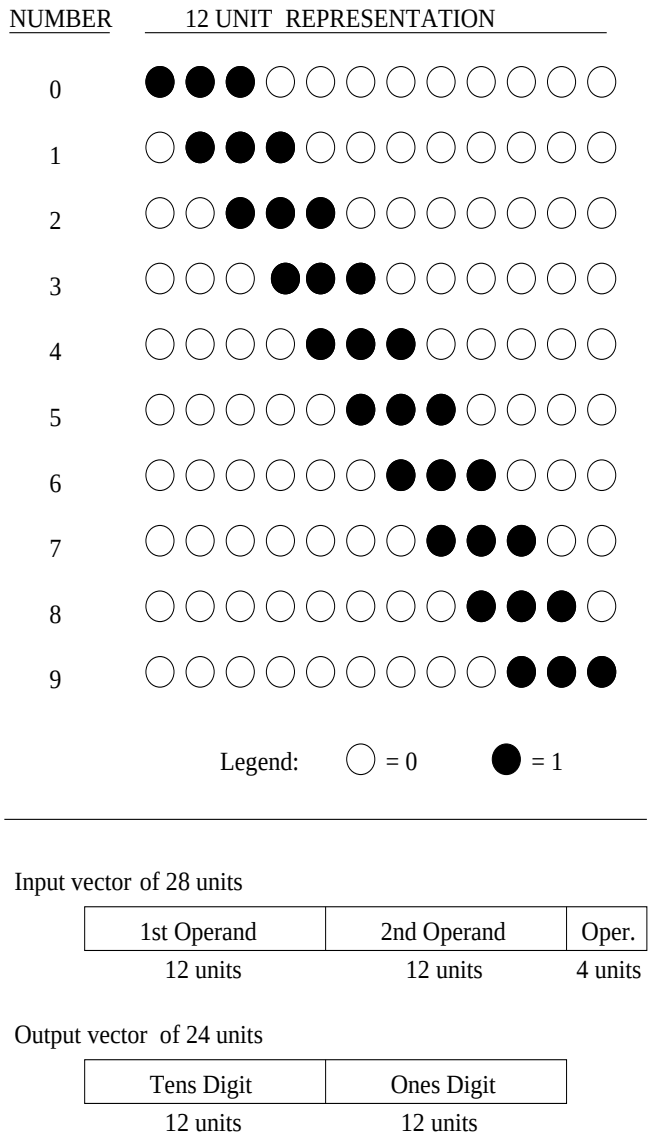

FIGURE 5. Coarse-coded representation used by McCloskey and Cohen (1989). The top panel illustrates the 12 element representation used for each number from zero to nine inclusive. The bottom panel shows the construction of the input stimuli and the desired output, both of which use these numeric representations.

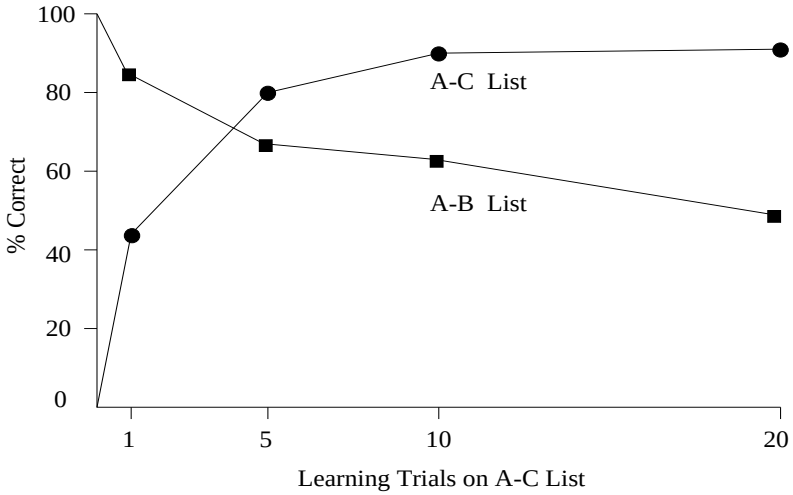

FIGURE 6. Results of the Barnes and Underwood (1959) study. After 20 learning trials on the A–C list the percentage correct for the A–B list has decreased to about 50%.

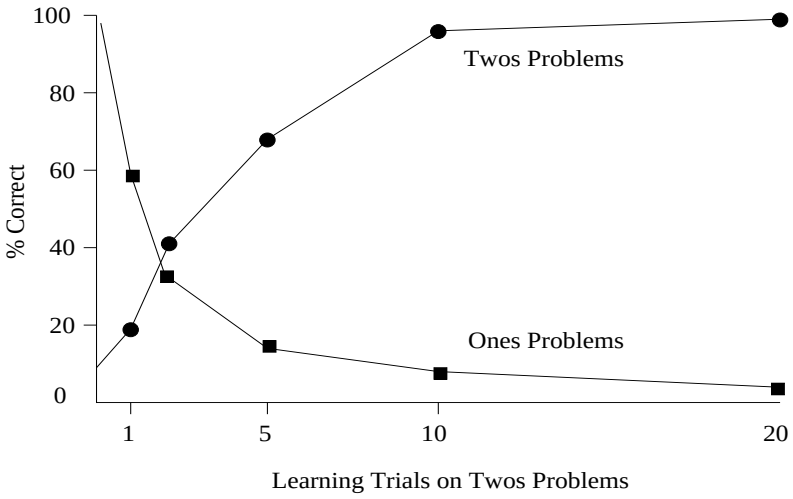

FIGURE 7. Results of the McCloskey and Cohen (1989) model. Unlearning of previously learned ones facts is rapid and almost complete after training the network on the twos facts.

previously learned material. These results, raising some concern about the wisdom of basing models of human learning performance on this type of architecture, have led to further investigation of this phenomenon.

3.2.5. *Further Investigation of the Model*

One of the drawbacks of a layered connectionist model, such as that of McCloskey and Cohen (1989), is that it is not easy to attribute specific results to particular characteristics of the network. However, Lewandowsky (1991, 1994) has undertaken such a task to identify the cause(s) of, and possible remedies for, catastrophic interference in this type of model. His results provide a somewhat more optimistic outlook for the cognitive modeling capabilities of neural networks.

Lewandowsky (1991), concentrating primarily on the abrupt steepness of unlearning in sequentially trained distributed models, shows the main cause to be the nature of the representation used. By creating stimuli as random valued zero-centered vectors (i.e., the expected value of correlation between vectors is 0) and applying a continuous retrieval measure (e.g., cosine), he produced a more gradual unlearning using the same network as McCloskey and Cohen (1989). However, another problem still remains. Even though unlearning is no longer as rapid, it continues to be practically complete when the mastery of the second list is achieved. Also, using random vectors fails to capture an important feature of arithmetic fact stimuli: The facts are naturally correlated, giving rise to confusion effects (e.g., consider $4 \times 8 = 32$ and $4 \times 6 = 24$ which have one operand in common). Therefore, another approach is needed to eliminate the sequential learning problem.

In a more recent article, Lewandowsky (1994) has reported on possible alternative solutions. First he notes that the ability to generalize, as well as the tendency for catastrophic interference, arise from a distributed representation. Thus, solutions for catastrophic interference that create less than fully distributed representations result in a decline of generalization ability. In contrast, solutions that modify the learning rule can reduce the overlap between *internal* representations (i.e., those created by the hidden units), and also maintain generalizability. One such solution, the *novelty rule*, uses only the differences of a new stimulus from previously learned material to update connection weights. A drawback to this approach, however, is that it is only applicable when the input and the output are the same, as in an auto-associator. Another solution, termed *activation sharpening*, reduces the overlap of (i.e., de-correlates) internal representations by raising the activity level of the hidden units that are already the most active, and decreasing the activity level of all the other hidden units (French, 1992). This solution can be applied even when the input and the desired output differ.

3.2.6. *Discussion*

To conclude, in addition to the severe interference difficulties encountered in this standard back-propagation model when sequentially trained, there are other drawbacks. First, because of the back-propagation algorithm, the model must be given

the answer to be able to learn. Second, learning itself is quite computationally intensive and can require thousands of iterations.

A further problem with this model is its failure to replicate another aspect of human behavior. Like the BSB, this model is deterministic (for a given stimulus and set of connections, the model always produces the same output). In contrast to human subjects, the model will never make an occasional mistake. Educated adults do make occasional mistakes when tested on basic number facts (e.g., under speeded testing, adults make errors in their responses to single-digit multiplication problems about 7.7% of the time, Campbell & Graham, 1985). The final model to be reviewed, MATHNET by McCloskey and Lindemann (1992) simulates this human behavior.

3.3. *The McCloskey and Lindemann* MATHNET *Model*

MATHNET is embedded within a general model of numerical processing (see Figure 8). This general model, first proposed by McCloskey et al. (1985), arises from dissociations observed in the numerical processing of brain-damaged patients (e.g., McCloskey, Aliminosa, et al., 1991; McCloskey et al., 1985; Warrington, 1982). MATHNET implements the arithmetic fact retrieval component of this general model.

The implementation of MATHNET, although in some ways similar to that of the McCloskey and Cohen (1989) model, contains some interesting differences, particularly in the areas of the retrieval process and the training regimen that is used. The architecture of the MATHNET model is also somewhat more complex.

3.3.1. *Architecture of the Model*

Like the McCloskey and Cohen (1989) model, MATHNET has a three layer structure consisting of 26 input units, each connected to all of the 40 hidden units, which are in turn connected to all of the 24 answer units. However, for MATHNET, unlike back-propagation models, the answer units are also interconnected. All the connections are bidirectional and symmetric. Also, each hidden and answer unit is provided with a bias (i.e., a tendency toward a positive or negative activation level).

3.3.2. *Multiplication Problem Representation*

The MATHNET representation for multiplication number facts is like that used by McCloskey and Cohen (1989) as illustrated in Figure 5 with two exceptions: 1) a -1 and $+1$ activation level is used instead of 0 and $+1$ and 2) the operation is coded by only two units (set to -1 and $+1$ for multiplication) instead of four units.

3.3.3. *Operation of the Model*

As described so far, this model does not appear to be extremely different from that of McCloskey and Cohen (1989). It is in the retrieval and learning processes that a major difference appears. Specifically, in contrast to the back-propagation model, which feeds forward simultaneously to calculate concurrently the responses (activation levels) of all units in a layer on each iteration, MATHNET employs an asynchronous technique, calculating the response of each unit in turn, in a random

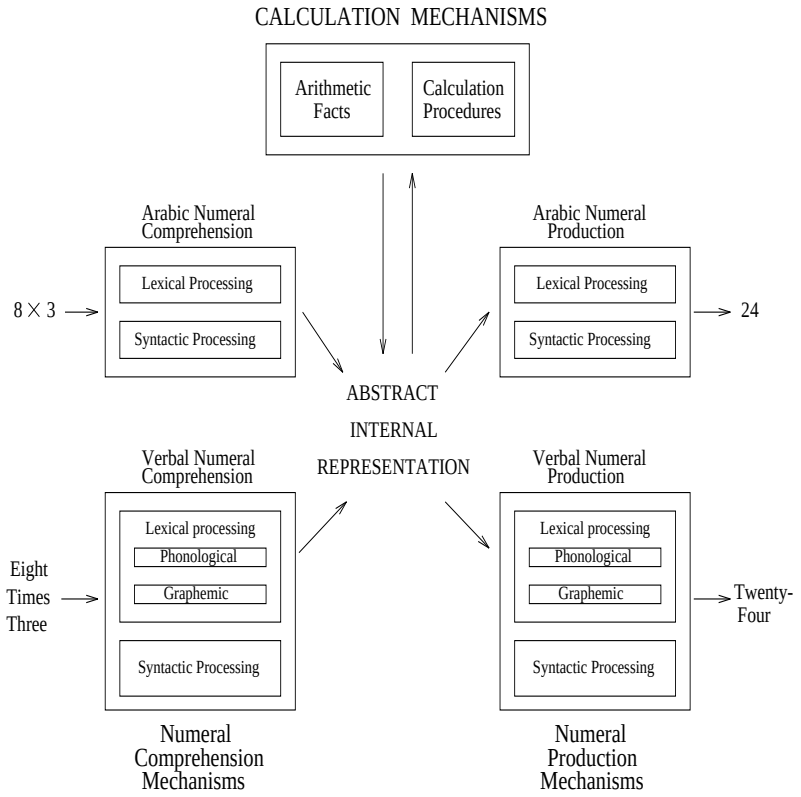

FIGURE 8. General model of numerical processing first proposed by McCloskey, Caramazza, and Basili (1985).

order fashion. During this activation update the bidirectional connections enable the answer units to affect hidden units, as well as themselves.

Retrieval process. To retrieve an answer to a multiplication problem, the network is first initialized by setting the activation level of the problem units to the problem representation and that of the hidden and answer units to 0. During the retrieval process the activation of the hidden and answer units is modified, but the problem units are *clamped* (i.e., remain unchanged and always supply input). *Simulated annealing* is then used. This term is adapted from a process of heating and gradually cooling materials, such as glass or metals, to free them from internal stress. For MATHNET, this process iteratively calculates activation values for each unit that is free to vary. On each iteration, the activation values of all the hidden and answer units are updated one at a time in a random order. This asynchronous updating of unit activation provides the ability of MATHNET to arrive at different answers for the same problem, because the activation value attained by a particular unit depends upon which other units have been updated previously. When the annealing

schedule is complete, the problem answer is determined by scoring the tens digit and ones digit responses of the answer units separately, by taking the best match to the representations used for the digits 0 to 9.

Figure 9 (see also Appendix B) presents an alternate view of the specific MATHNET architecture. This figure illustrates the three sets of layer connections as matrices: **W** is a 26×40 matrix connecting each problem unit to each hidden unit, **Z** is a 24×40 matrix connecting each hidden unit to each answer unit, and **A** is a 24×24 matrix connecting the answer units to themselves. Recall that in the MATHNET architecture the connections are bidirectional.

To provide a specific example, Figure 9 illustrates the activation update of one particular hidden unit, hidden unit 4. Input to the activation update comes from three sources: 1) the current activation levels of each problem unit multiplied by its connection weight to hidden unit 4, 2) the current activation level of each answer unit multiplied by its connection weight to hidden unit 4, and 3) a bias weight.

Learning phase. As in the models previously described, a learning algorithm is used to create the weight values assigned to each connection. Mathematically, the retrieval process just described, and the learning process, are both based on the *mean field theory* technique. More information on this technique, and its relationship to other network approaches, can be found in Peterson and Anderson (1987), Peterson & Hartman (1989), and Haykin (1994). Appendix B gives more detail on learning and retrieval.

3.3.4. *Application to Multiplication Number Fact Retrieval*

To ensure robustness of the MATHNET model, McCloskey and Lindemann ran three separate simulations using the same architecture. Different random connections were assigned initially, as well as a different order of presenting problems on each learning cycle, and a different order of unit selection for updating on each iteration of the annealing schedule. The authors report a unique training regimen, which simulates human experience, and also appears to eliminate the problem of catastrophic interference.

Training regimen. During the learning phase, MATHNET was presented with 64 different multiplication number facts (i.e., 2×2 through 9×9). Two factors often posited to be the cause of the problem size effect in human subjects, order of learning and frequency of presentation, were simulated. The training of the network began with an ordered presentation of stimuli. Eight sets of stimuli were presented for five learning cycles each. The order of the problem presentation for a set was varied randomly in each cycle. The first training set contained the problems with an operand of 2, which are called the 2's problems. The second training set included the 2's problems and also 3's problems, excepting those 3's facts that were previously contained in the 2's set (e.g., 2×3). The third set included 2's, 3's, and 4's problem, and so on. The new problems in each set were included twice each, and the old problems just once.

Following the ordered training, all 64 problems were presented to the network in one set to simulate frequency effects. In this final training set, the inclusion

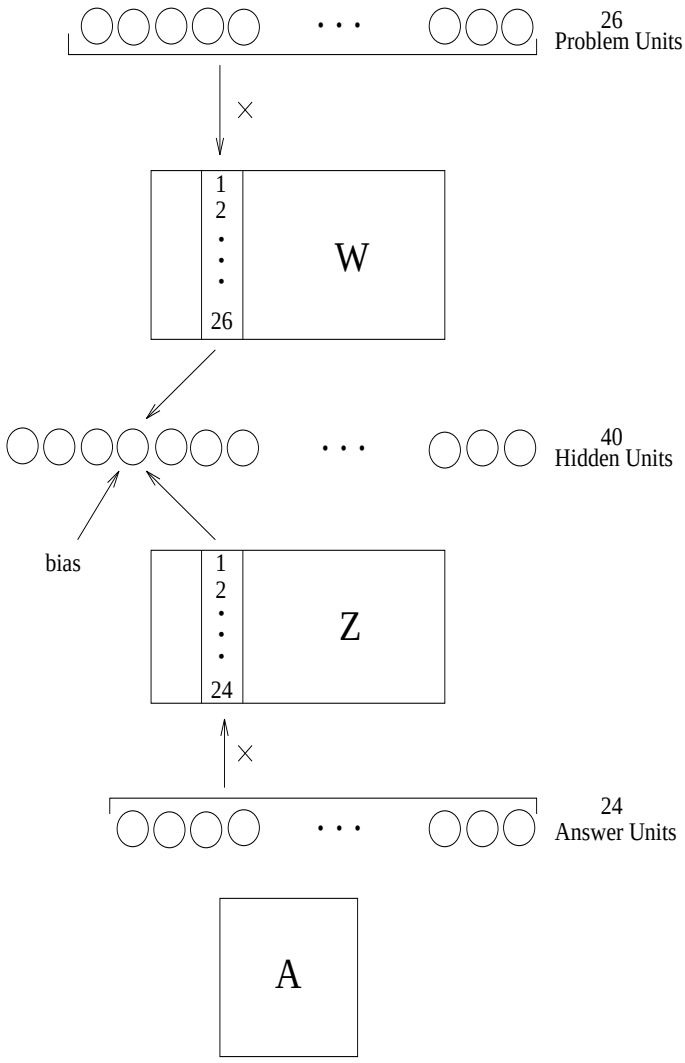

FIGURE 9. A view of the MATHNET architecture illustrating the activation update of hidden unit 4. See Appendix B for more detail.

frequency of a particular problem was based on a size class, with problems classified as "small" included more often than those classified as "large."

In the learning phase, performance of the networks was tested every five learning cycles until the answers produced in the free phase were totally correct, requiring 123 learning cycles on the average across all networks. Then, testing was done by presenting the 64 problems to the 16-iteration annealing schedule until equilibrium was reached, and the answer scored. This test was done ten times, and the results

for all networks were almost perfect (only one problem out of 640 was missed). Thus, MATHNET demonstrated the ability to learn the times tables in less training time than would be required by back-propagation.

Speeded testing effects. To simulate the pressure of speeded testing conditions in human subjects, McCloskey and Lindemann (1992) shortened the annealing schedule used in testing, by omitting the first five iterations and discontinuing iterations as soon as a stable state was achieved. Each of the three networks was tested 30 times on the set of 64 problems. This speeded testing resulted in occasional errors, reducing the mean accuracy to 97.3%. Examination of the types of errors made shows that 79% were operand errors, matching exactly that found in human subjects by Campbell and Graham (1985), 5% were table errors, and 16% were errors classified as non-table. Moreover, in 91% of the operand errors, the non-shared operand was within one unit of the correct operand, thus simulating the operand distance effect.

Problem size effects. To determine if the network showed a problem-size effect, response time was correlated to problem size and to error rate. Correlation of the network response time (number of iterations to reach stability) and the sum of the problem operands as a measure of problem size was .69 ($p < .001$). This compares well to studies of human subjects, which have found correlations ranging from .6 to .8 (Campbell & Graham, 1985; Miller et al., 1984; Stazyk et al., 1982). The correlation of the network error rate and the sum of problem operands was .52 ($p < .001$) as compared to Campbell and Graham's (1985) correlation of .63 for human subjects.

To investigate the cause of the problem size effect found in MATHNET, McCloskey and Lindemann (1992) varied the training regimen for additional networks. No problem size effect arose when networks were trained without the order and frequency manipulations. Likewise, various ordered training regimens without a subsequent frequency manipulation did not yield the problem size effect. In fact, the problem size effect was only shown when frequency alone was manipulated in the same manner as in the original regimen. Therefore, the authors conclude that frequency is the main determinant in the MATHNET problem size effect, as was suggested by Fendrich et al. (1993) for human subjects.

Lesioning effects. McCloskey and Lindemann (1992) simulated brain-damaged human subjects, "lesioning" MATHNET by reducing each connection weight by a random percentage of its magnitude. Their results showed definite impairment, with accuracy of the networks declining to 69%, 86%, and 81%. Also, similar to brain-damaged human performance, the impairment was non-uniform, with correct answers given to some problems on every test, mistakes made on other problems in only some of the tests, and other problems consistently missed. The network also showed a problem size effect on errors, but this was considerably weaker than for human subjects. As found in human studies the percentages of types of error varied across networks. One of the networks yielded a large percentage (34%) of non-table errors as is occasionally observed in patients (e.g., McCloskey, Harley, et al., 1991 report results of this type for two patients).

Several differences between the performance of the lesioned MATHNET and impaired humans, in addition to the weak problem size effect, are noted by McCloskey and Lindemann (1992). One of these discrepancies is the performance on complementary problems (e.g., 8×6 and 6×8). Unlike most patients, the network does not show similar error rates between the two related problems, but only a weak tendency in this direction. In fact, each of the networks showed a very high error rate on one particular problem, and a very low error rate on its complement. Lastly, the network made only errors of commission, always arriving at some answer. Brain-damaged subjects also make errors of omission, by failing to give any response at all (McCloskey, Harley, et al., 1991).

Additional MATHNET lesioning simulations indicate that the location of lesioning, in addition to the amount of damage, must be considered when evaluating the simulation results (Lories, Aubrun, & Seron, 1994). In addition to confirming the McCloskey and Lindemann (1992) results, Lories et al. obtained interactions between location and error type, and between amount of damage and error type. In brief, using the terms of Figure 9 : Increasing damage to **W** results in a shift from operand errors to in-table errors, increasing damage to **Z** induces out-of-table errors, and increasing damage to **A** induces operand errors.

3.3.5. *Discussion*

The MATHNET model, embedded in a general framework that is based on dissociations demonstrated by brain-damaged patients, appears to be the strongest model of those reviewed in this paper for producing quantitative results comparable to human data. It performs strongly in its ability to learn multiplication number facts, as evidenced by the almost perfect results attained when the full annealing schedule was used for retrieval. Also, the speeded testing technique yields occasional errors with a pattern resembling that of human subjects. Lesioning tests generate network results somewhat similar to that of brain-damaged human subjects. However, there are certain shortcomings in this network as a model of human performance in multiplication number facts.

The weaknesses of MATHNET include a relatively small task scope, and a failure to predict some of the phenomena observed in human subjects. The current focus of MATHNET is quite narrow, including only production of answers, and excluding those problems containing 1's and 0's as operands. McCloskey and Lindemann (1992) state that they plan to enlarge this scope. The MATHNET model shows an error percentage identical to that of human subjects on operand errors. However, it fails to match human performance for other types of errors, namely exhibiting proportions of table and non-table errors that are the reverse of human subjects. The network also fails to demonstrate the typical human subject advantages with the five times table or problems that are ties. Perhaps most interesting is the failure of the network to extract and employ the principle of commutativity. It appears that brain-damaged human subjects may be able to exploit this complementary relationship to compensate for failure of retrieval on a specific problem. The network does not show this capability.

4. Conclusion

In summary, the connectionist approach can predict the problem size effect, and the typical human error pattern for multiplication number fact retrieval, as demonstrated most robustly by MATHNET. Investigations of possible causes of the problem size effect have shown frequency and number representation to be major factors in modeling. The probabilistic approach of MATHNET even produces occasional errors, and simulates the performance of brain-damaged patients. Also, the connectionist approach does not preclude the prediction of human behavior that appears to be rule driven. The BSB and back-propagation simulations, which included zero operand problems, were able to extract the "rule" of zero multiplication. However, although some ability to generalize learning is demonstrated, none of the models reviewed extracts the principle of commutativity as well as even brain-damaged human subjects. Simulations of order effects have not only failed to produce the problem size effect, they have, except in MATHNET, produced "catastrophic interference." Therefore, despite the successes of these models, some challenges still remain for the future. The ability to form a large set of specific related associations and also generalize that learning does not as yet equal that of human subjects. Alternative solutions to the sequential learning problem of catastrophic interference call for further investigation. Also, the biological plausibility and psychological relevance of these architectures and algorithms remain somewhat in question.

Additional key issues, in regard to human cognition, are the role and the characteristics of number representation within the model architecture. These models illustrate a relationship between the components of a connectionist model: architecture, stimuli representation, and algorithm. The architecture can be quite simple, as in the one-layer BSB, or multi-layered, as in the back-propagation and MATHNET models. In contrast to the BSB, with the number of representation units sometimes exceeding 1000, the three-layer networks employ less than 30 units. The richness of the BSB representation provides a wide ranging predictive capacity. In the more complex architectures, power is provided by internal representations derived in the hidden units from more complex learning algorithms. Although a meaningful interpretation of this representation remains to be done, de-correlation of these internal values (e.g., through activation sharpening) may offer relief from the problem of catastrophic interference.

A major issue for these, and other, models is the nature of numerical representation at the time of arithmetic fact retrieval. One of the features of MATHNET, and a cornerstone of the general modular model, is the "abstract internal representation" of number. This representation has generated considerable controversy (see Campbell, 1992; Campbell & Clark, 1988; Campbell & Clark, 1992; but see McCloskey, Macaruso, & Whetstone, 1992). In contrast to an abstract representation, Campbell and his colleagues posit an "encoding-complex" memory representation, based on their empirical data. This view proposes that numbers are represented internally, and available for use in calculation, in closely associated multiple modalities. This concept of multiplicity of representation is also found in the Anderson BSB model,

although in a simpler fashion. Debate on this issue drives to the heart of a core problem in modeling human behavior, namely the fundamental question of human mental representation of information.

Finally, what is the contribution of these models to the domain of cognitive studies? Although they simulate some of the known human data, they are not able to produce comparable results in all areas, nor have they predicted new phenomena. Therefore, in a strict sense, they cannot be considered as having the predictive power of a theory. However, they have served to help us examine the feasibility, or lack of feasibility, of specific mechanisms to explain existing data. These attempts to simplify and formalize our existing knowledge have helped to pinpoint issues and gaps existing in current theories, especially those relating to mental representation, thus fueling new empirical research.

References

[1] Abdi, H. (1994a). A neural network primer. *Journal of Biological Systems*, **2**, 247–281.

[2] Abdi, H. (1994b). *Les réseaux de neurones*. Grenoble: Presses Universitaires de Grenoble.

[3] Aiken, L. R., & Williams, E. N. (1973). Response times in adding and multiplying single-digit numbers. *Perceptual and Motor Skills*, **37**, 3–13.

[4] Allen, P. A., Ashcraft, M. H., & Weber, T. A. (1992). On mental multiplication and age. *Psychology and Aging*, **7**, 536–545.

[5] Anderson, J. A. (1992, September). Neural-network learning and Mark Twain's cat, *IEEE Communications Magazine*, **30**(9), 16–23.

[6] Anderson, J. A. (1995). *An introduction to neural networks*. Cambridge, MA: MIT Press.

[7] Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception and probability learning: Some applications of a neural model. *Psychological Review*, **84**, 413–451.

[8] Anderson, J. A., Spoehr, K. T., & Bennett, D. J. (1994). A study in numerical perversity: Teaching arithmetic to a neural network. In D.S. Levine & M. Aparicio IV (Eds.), *Neural networks for knowledge representation and inference* (pp. 311–335). Hillsdale, NJ: Erlbaum.

[9] Ashcraft, M. H. (1982). The development of mental arithmetic: A chronometric approach. *Developmental Review*, **2**, 213–236.

[10] Ashcraft, M. H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research* (pp. 302–338). New York: Springer-Verlag.

[11] Ashcraft, M. H. (1992). Cognitive arithmetic: A review of data and theory. *Cognition: International Journal of Cognitive Science*, **44**, 75–106.

[12] Ashcraft, M. H., & Battaglia, J. (1978). Cognitive arithmetic: Evidence for retrieval and decision processes in mental addition. *Journal of Experimental Psychology: Human Learning and Memory*, **4**, 527–538.

[13] Ashcraft, M. H., & Fierman, B. A. (1982). Mental addition in third, fourth, and sixth graders. *Journal of Experimental Child Psychology*, **33**, 216–234.

[14] Banks, W. P., & Hill, D. K. (1974). The apparent magnitude of number scaled by random production. *Journal of Experimental Psychology*, **102**, 353–376.

[15] Barnes, J. M., & Underwood, B. J. (1959). "Fate" of first-list associations in transfer theory. *Journal of Experimental Psychology*, **58**, 97–105.

[16] Campbell, J. I. D. (1985). Associative interference in mental computation. Unpublished doctoral dissertation. University of Waterloo.

[17] Campbell, J. I. D. (1987a). Network interference and mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **13**, 109–123.
[18] Campbell, J. I. D. (1987b). The role of associative interference in learning and retrieving arithmetic facts. In J. A. Sloboda & D. Rogers (Eds.), *Cognitive processes in mathematics* (pp. 107–122). Oxford: Clarendon Press.
[19] Campbell, J. I. D. (1992). In defense of the encoding-complex approach: Reply to McCloskey, Macaruso, & Whetstone. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 539–556). Elsevier Science B.V.
[20] Campbell, J. I. D. (1994). Architectures for numerical cognition. *Cognition*, **53**, 1–44.
[21] Campbell, J. I. D., & Clark, J. M. (1988). An encoding-complex view of cognitive number processing: Comment on McCloskey, Sokol, and Goodman (1986). *Journal of Experimental Psychology: General*, **117**, 204–214.
[22] Campbell, J. I. D., & Clark, J. M. (1989). Time course of error-priming in number fact retrieval: Evidence for excitatory and inhibitory mechanisms. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **15**, 920–929.
[23] Campbell, J. I. D., & Clark, J. M. (1992). Cognitive number processing: An encoding-complex perspective. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 457–491). Elsevier Science B.V.
[24] Campbell, J. I. D., & Graham, D. J. (1985). Mental multiplication skill: Structure, process, and acquisition. *Canadian Journal of Psychology*, **39**, 338–366.
[25] Campbell, J. I. D., & Oliphant, M. (1992). Representation and retrieval of arithmetic facts: A network-interference model and simulation. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 331–364). Elsevier Science B.V.
[26] Dehaene, S. (1992). Varieties of numerical abilities. *Cognition: International Journal of Cognitive Science*, **44**, 1–42.
[27] Dehaene, S., & Cohen, L. (1991). Two mental calculation systems: A case study of severe acalculia with preserved approximation. *Neuropsychologia*, **29**, 1045–1074.
[28] Fendrich, D. W., Healy, A. F., & Bourne, L. E. Jr. (1993). Mental Arithmetic: Training and retention of multiplication skill. In C. Izawa (Ed.), *Cognitive psychology applied* (pp. 111-133). Hillsdale, NJ: Erlbaum.
[29] French, R. M. (1992). Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, **4**, 365–377.
[30] Geary, D. C., Widaman, K. F., & Little, T. D. (1986). Cognitive addition and multiplication: Evidence for a single memory network. *Memory and Cognition*, **14**, 478–487.
[31] Geary, D. C., & Wiley, J. G. (1991). Cognitive addition: Strategy choice and speed-of-processing differences in young and elderly adults. *Psychology and Aging*, **6**, 474–483.
[32] Golden, R. M. (1986). The "brain-state-in-a-box" neural model is a gradient descent algorithm. *Journal of Mathematical Psychology*, **30**, 73–80.
[33] Graham, D. J., & Campbell, J. I. D. (1992). Network interference and number-fact retrieval: Evidence from children's alphaplication. *Canadian Journal of Psychology*, **46**, 65–91.
[34] Groen, G. J., & Parkman, J. M. (1972). A chronometric analysis of simple addition. *Psychological Review*, **79**, 329–343.
[35] Hamann, M. S. & Ashcraft, M. H. (1986). Textbook presentations of the basic addition facts. *Cognition and Instruction*, **3**, 173–192.
[36] Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York, NY: Macmillan.
[37] Koshmider, J. W., & Ashcraft, M. H. (1991). The development of children's mental multiplication skills. *Journal of Experimental Child Psychology*, **51**, 53–89.
[38] Lewandowsky, S. (1991). Gradual unlearning and catastrophic interference: A comparison of distributed architectures. In W. E. Hockley & S. Lewandowsky (Eds.), *Relating theory and data: Essays on human memory in honor of Bennet B. Murdock*, (pp. 445–476). Hillsdale, NJ: Erlbaum.

[39] Lewandowsky, S. (1994). On the relation between catastrophic interference and generalization in connectionist networks. *Journal of Biological Systems*, **2**, 307–333.

[40] Lories, G., Aubrun, A. & Seron, X. (1994). Lesioning McCloskey & Lindemann's Mathnet: The effect of damage. *Journal of Biological Systems*, **2**, 335–356.

[41] McCloskey, M., Aliminosa, D., & Sokol, S. M. (1991). Facts, rules, and procedures in normal calculation: Evidence from multiple single-patient studies of impaired arithmetic fact retrieval. *Brain & Cognition*, **17**, 154–203.

[42] McCloskey, M., Caramazza, A., & Basili, A. G. (1985). Cognitive mechanisms in number processing and calculation: Evidence from dyscalculia. *Brain and Cognition*, **4**, 171–196.

[43] McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In G. H. Bower (Ed.), *The psychology of learning and motivation: Vol. 24* (pp. 109–165). San Diego: Academic press.

[44] 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**, 377–397.

[45] McCloskey, M., & Lindemann, A. M. (1992). Mathnet: Preliminary results from a distributed model of arithmetic fact retrieval. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 365–410). Elsevier Science B.V.

[46] McCloskey, M., Macaruso, P., & Whetstone, T. (1992). The functional architecture of numerical processing mechanisms: Defending the modular model. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 493–537). Elsevier Science B.V.

[47] Michie, S. (1985). Development of absolute and relative concepts of number in preschool children. *Developmental Psychology*, **21**, 247–252.

[48] Miller, K. F. (1992). What a number is: Mathematical foundations and developing number concepts. In J. I. D. Campbell (Ed.), *The nature and origins of mathematical skills* (pp. 3–38). Elsevier Science B.V.

[49] Miller, K., Perlmutter, M., & Keating, D. (1984). Cognitive arithmetic: Comparison of operations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **10**, 46–60.

[50] Moyer, R. S., & Landauer, T. K. (1967). The time required for judgments of numerical inequality. *Nature*, **215**, 1519–1520.

[51] Norem, G. M. & Knight, F. B. (1930). The learning of the one hundred multiplication combinations. In *National Society for the Study of Education: Report on the Society's Committee on Arithmetic, Vol. 15, NSSE Yearbook 29*, 551–567.

[52] Parkman, J. M. (1972). Temporal aspects of simple multiplication and comparison. *Journal of Experimental Psychology*, **95**, 437–444.

[53] Parkman, J. M., & Groen, G. J. (1971). Temporal aspects of simple addition and comparison. *Journal of Experimental Psychology*, **89**, 335–342.

[54] Peterson, C. & Anderson, J. R. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, **1**, 995–1019.

[55] Peterson, C. & Hartman, E. (1989). Explorations of the mean field theory learning algorithm. *Neural Networks*, **2**, 475–494.

[56] Shepard, R. N., Kilpatric, D. W., & Cunningham, J. P. (1975). The internal representation of numbers. *Cognitive Psychology*, **7**, 82–138.

[57] Siegler, R. S. (1987). Strategy choices in subtraction. In J. Sloboda & D. Rogers (Eds.), *Cognitive processes in mathematics*. Oxford: Oxford University Press.

[58] Sokol, S. M., McCloskey, M., Cohen, N. J., & Aliminosa, D. (1991). Cognitive representations and processes in arithmetic: Evidence from the performance of brain-damaged patients. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **17**, 355–376.

[59] 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**, 320–335.

[60] Todd, R. R., Barber, P. J., & Jones, D. (1987). The internal representation of number: Analogue or digital? In J. A. Sloboda, & D. Rogers (Eds.), *Cognitive processes in mathematics* (pp. 142–156). Oxford: Clarendon Press.
[61] Viscuso, S. R. (1989). Memory for arithmetic facts: A perspective gained from two methodologies. Unpublished Doctoral Thesis, Providence, RI: Brown University.
[62] Viscuso, S. R., Anderson, J. A., & Spoehr, K. T. (1989). Representing simple arithmetic in neural networks. In G. Tiberghien (Ed.), *Advances in cognitive science Vol. 2: Theory and applications* (pp. 141–164). Chichester: Ellis Horwood Limited.
[63] Warrington, E. K. (1982). The fractionation of arithmetic skills: A single case study. *Quarterly Journal of Experimental Psychology*, **34A**, 31–51.
[64] Winkelman, J. H., & Schmidt, J. (1974). Associative confusion in mental arithmetic. *Journal of Experimental Psychology*, **102**, 734–736.
[65] Woods, S. S., Resnick, L. B., & Groen, G. J. (1975). An experimental test of five process models for subtraction. *Journal of Educational Psychology*, **67**, 17–21.
[66] Zbrodoff, N. J., & Logan, G. D. (1986). On the autonomy of mental processes: A case study of arithmetic. *Journal of Experimental Psychology: General*, **115**, 118–130.
[67] Zbrodoff, N. J., & Logan, G. D. (1990). On the relation between production and verification tasks in the psychology of simple arithmetic. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **16**, 83–97.

Appendix A. BSB Retrieval Algorithm

The retrieval function of the simplest BSB variant is:

$$\mathbf{x}_{[n+1]} = f(\gamma \mathbf{x}_{[n]} + \eta \mathbf{W} \mathbf{x}_{[n]} + \delta \mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is a vector representing brain state, n an iteration step, $\mathbf{W}$ the connection matrix, γ a decay constant, η a feedback constant, and δ a clamping constant. The limit function constraining the unit activation to a lower limit of -1 and an upper limit of $+1$, is:

$$f(x) = \begin{cases} -1, & \text{for } x < -1 \\ x, & \text{for } -1 \leq x \leq 1 \\ +1, & \text{for } x > +1. \end{cases}$$

As Equation 1 shows, the limit function is applied to an expression consisting of three components. The first component of the expression, $\gamma \mathbf{x}_{[n]}$, represents the activation state achieved by all the units at iteration step n scaled by the decay constant γ. The second component, $\eta \mathbf{W} \mathbf{x}_{[n]}$, represents the feedback arising from "filtering" this activation through the connection matrix, scaled by the feedback constant η. The third component, $\delta \mathbf{x}$, represents a multiple of the initial problem input and implements a clamping option. This function is performed iteratively until no change in state occurs or a maximum number of iterations is reached. When $\mathbf{x}_{[n+1]} = \mathbf{x}$ the association of a pattern to itself has been perfectly learned.

Appendix B. MATHNET Retrieval

The activation formula for any unit i free to vary, denoted y_i, may be expressed as:

$$y_i = \tanh(\sum_j y_j x_{ij}/T) \tag{2}$$

where y_j is the activation of a connected unit j, x_{ij} the connection weight between unit i and unit j, T is the temperature parameter declining in value on each iteration, and tanh the hyperbolic tangent. McCloskey and Lindemann (1992) used a 16 iteration annealing schedule with a beginning temperature value of 30 and a final value of .5. When the temperature parameter is large (i.e., hot) the activation function does not yield extreme values. As the temperature decreases, the resulting unit activations approach the limits of either -1 or $+1$.

If the problem units are represented by column vector $\mathbf{f}$, the hidden units by column vector $\mathbf{h}$, and the answer units by column vector $\mathbf{g}$, the resulting activation of hidden unit 4 (h_4), as illustrated in Figure 9, can be expressed as:

$$h_4 = \tanh(\mathbf{f}^T \mathbf{w}_{.,4} + \mathbf{g}^T \mathbf{z}_{.,4} + b_{h_4}) \tag{3}$$

where T is the transpose, $\mathbf{w}_{.,4}$ is the fourth column vector of $\mathbf{W}$ containing connections from all problem units to hidden unit 4, $\mathbf{z}_{.,4}$ is the fourth column vector of $\mathbf{Z}$ connecting answer units to hidden unit 4, and b_{h_4} is the bias value for hidden unit 4. In like manner the activation of answer unit 4 can be calculated as:

$$g_4 = \tanh(\mathbf{h}^T \mathbf{z}^T_{4,.} + \mathbf{g}^T \mathbf{a}^T_{4,.} + b_{g_4}) \tag{4}$$

where $\mathbf{z}^T_{4,.}$ is the fourth row of $\mathbf{Z}$ transposed into a column vector, and $\mathbf{a}^T_{4,.}$ is the fourth row of $\mathbf{A}$.

MATHNET Learning

Connection weights are initialized to random values uniformly distributed from -0.5 to $+0.5$ and the retrieval process described above is done for a specific problem. This is termed the *free phase* because the answer units, as well as the hidden units, are free to vary. When the annealing schedule is complete, the product of the activations of each pair of connected units is calculated and retained. The annealing schedule is now repeated for the problem in a *clamped phase* by setting the answer units to the correct response, and not allowing either problem or answer units to vary. When the schedule is complete, the product of the activation values for each pair of connected units is again calculated. The correction for each connection weight between any two units i and j is now computed as a function of the difference between the product of their activations, denoted $y_i y_j$, obtained in the clamped phase and that obtained from the free phase:

$$\Delta x_{ij} = \eta(y_i y_{j_{[\text{clamped}]}} - y_i y_{j_{[\text{free}]}}) \tag{5}$$

where Δx_{ij} represents an element in any one of the three connection matrices, and η is a small positive learning constant that was set to .003. This correction may be

expressed for each of the three connection matrices as:

$$\Delta\mathbf{W} = \eta(\mathbf{f}\mathbf{h}^T_{[\text{clamped}]} - \mathbf{f}\mathbf{h}^T_{[\text{free}]})$$
$$\Delta\mathbf{Z} = \eta(\mathbf{g}\mathbf{h}^T_{[\text{clamped}]} - \mathbf{g}\mathbf{h}^T_{[\text{free}]})$$
$$\Delta\mathbf{A} = \eta(\mathbf{g}\mathbf{g}^T_{[\text{clamped}]} - \mathbf{g}\mathbf{g}^T_{[\text{free}]})$$

where $\mathbf{f}$ is a problem vector, $\mathbf{h}$ is the corresponding hidden unit vector, and $\mathbf{g}$ the corresponding answer vector. The corrections thus determined may be applied to the weights after each problem is processed, or accumulated over several problems before being applied. McCloskey and Lindemann (1992) adjusted weights every 10 problems for small learning sets, and every 64 problems for larger sets.