ANSR-FR-ED9702

Development of Neural Network Tools and Techniques for Arithmetic Training of Children with Learning Disabilities:

Phase I -- Theory and Simulations

by

William C. Mead

wcm@ansr.com

Adaptive Network Solutions Research, Inc.

http://www.ansr.com

February 22, 1997

Prepared for presentation at the AERA '97, March 24-28, Chicago, IL

Work performed under the auspices of the

U. S. Department of Education

Solicitation SBIR/RFP No. 95-025, Topic 5
Contract no. RW95169013, effective date 9/1/95
Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views or policies of the Department of Education.

Abstract

This research applies the emerging technology of Neural Networks (NN) to the task of training students with Learning Disabilities in fluent, automatized recall of basic arithmetic facts. Three new NN-based tools have been developed: (1) an adaptive tutor, (2) a guidance/ assessment module, and (3) a student simulator. These three tools inter-operate as a teaching and learning simulation laboratory, **MAN**. The student simulator (**STU**) provides a controllable model of human learning to test the adaptive tutor and guidance modules. The adaptive tutor (**TUT**) automatically monitors student knowledge and uses this information to orchestrate the presentation of new and review facts to help the student achieve near-optimum learning. The guidance/assessment module (**GUY**) shares components of the adaptive tutor and presents information on the student's learning progress and knowledge status to a supervising teacher. All three components are complete in a form that performs the required functions with very simple user interface. A number of simulations demonstrate the feasibility and attractiveness of the ideas, and provide some insights into the processes of teaching and learning arithmetic facts, and the possible effects of Learning Disabilities on the learning process. Very preliminary testing by educators and two LD students is encouraging.

1. INTRODUCTION and BACKGROUND

Special Education is a large subfield of education [1] that has evolved as a vehicle to provide children that have various special physical, mental, or emotional needs with equal access to educational opportunities. The special needs, which include "Learning Disabilities (LD)", are manifested, in part, as slower-than-average learning rate or less-flexible-than-usual learning modes. One result of this is that special education tends to be labor-intensive and costly. Even with extra training and smaller classes, the special education teacher may be hard-pressed to provide all the support, guidance, and interaction that the students need.

The field of Artificial Neural Networks has burgeoned in the past several years, with many new, successful applications [2] in areas where computers had previously proven quite limited, or too difficult or costly to apply. The general idea of an artificial neural network (ANN or NN) is that it roughly mimics the mechanisms that nature has developed to do "intelligent" tasks: a network of many independent, simple processing units (nodes) that work together by exchanging information, broadly similar to the mode of operation of the human brain. The network "learns" by adjusting its internal structure (connections and weights) to conform to data that is presented to it. Tasks that have been successfully done with NN's include modeling and prediction of very complicated physical systems [3], control of complex, poorly understood machines [4], and classification and recognition of objects based on images, sounds, or other measurements [2].

The goal of this project is *to apply neural network technology to improve the teaching of basic arithmetic skills to students with Learning Disabilities (LD).* Phase I is completed and has created three specific applications of neural network techniques to the training of LD students in basic arithmetic, as indicated schematically in Fig. 1. Each of the modules has been developed as an "engine," i.e., a basic, functional unit, without a refined/glossy user interface.

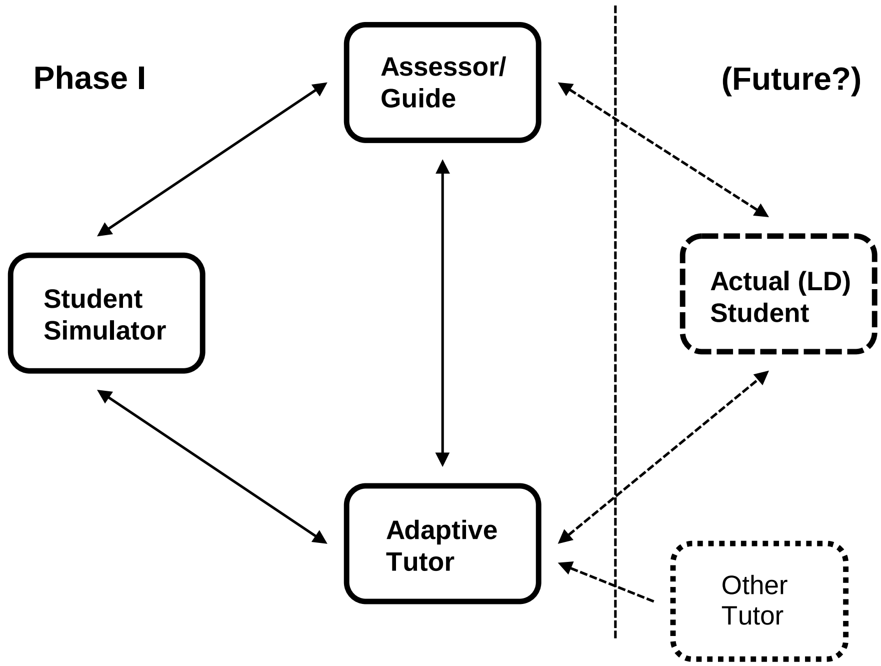

Figure 1. Adaptive Teaching and Learning Laboratory concept. All three Phase I components interoperate to provide a flexible, rapid research and prototyping environment. In the future, an actual student will replace the Student Simulator; or another CBI program could replace the Adaptive Tutor for comparison testing.

2. A SCHEMATIC MODEL FOR LEARNING

To understand Learning Disabilities, it is important to understand the learning process itself in human beings. Cognitive models of the learning process are rudimentary, though progressing towards greater detail and with stronger research foundations. There is mutual interaction between the fields of neuropsychology/learning and artificial neural networks (NN) [5-8]. Figure 2 shows a schematic model of some of the processes involved in learning that is either implicit or explicit [7] in much of the current educational and psychological literature. The usual goal of learning is to suitably encode and store knowledge in long-term Memory so that it can be later recalled and used, as needed.

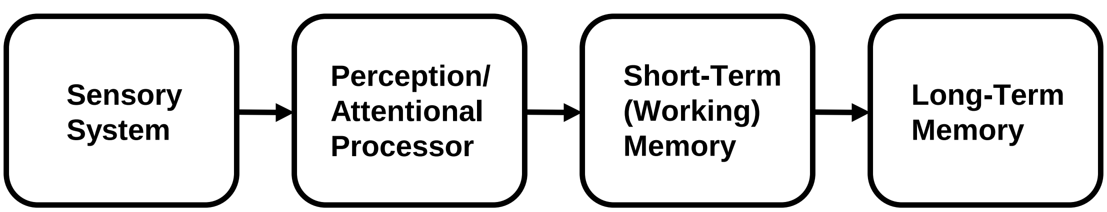

Figure 2. Simplified schematic model of human learning processes.

3. LEARNING DISABILITIES and ARITHMETIC

Learning Disabilities can be operationally defined in terms of impairment or deficiency in one or more of the basic functions of learning [9,10]. Definitions based on discrepancies between achievement and IQ (potential) typically lead to identification of about 5% of the student population as Learning Disabled.

The most frequent capability (or functional) deficits found in LD students can be described in the following categories:

- Auditory perception and (short-term) memory
- Visual perception and (short-term) memory
- Visual symbol to phoneme mapping
- Attentional mechanisms
- Inference or deduction using internal symbols
- Problem solving or strategic (metacognitive) decisions

The degree and kind of functional deficit(s) vary in individual cases.

The effects of Learning Disabilities on learning arithmetic are of particular interest here. An extensive search identified previous research relevant to this work [11-17]. There is considerable evidence that weakness in basic skills impedes later progress on the part of LD students in both reading and math [15-17]. This contributes to a gap between normal and LD students' knowledge and performance that grows with time as the students mature. Information on successful interventions in the arithmetic training of LD students is very sparse.

A clear evaluative consensus exists in the literature reviewed on Computer Assisted Instruction [18,19]: the use of computers in K-12 education has been less revolutionary than early expectations, but has made solid gains in both quantity and quality of practice, particularly in the past ten years. Computer Based Instruction (CBI) has been used extensively in Special Education for "Drill and Practice," and somewhat less extensively, but with increasing emphasis, for educational simulations and multimedia presentation of material. Educators generally appreciate the greatest strength of CBI: affordable, prompt computer-pupil interaction and feedback. However, along with this appreciation is the concern that Special Ed students may be subjected to more drill and practice than is optimum. To date, adaptation of instruction to an individual student's needs has been used, within limits, and found useful. Improvements are desired.

The use of computers for instructional management and student assessment has also improved and increased over time. Though not as widespread as CBI, Computer Managed Instruction (CMI) and Computer Based Assessment and/or Testing (CBA or CBT) are being used in school settings, and some research indicates that significant educational benefits accrue [20,21].

The literature search was augmented by the formation of a "Virtual Advisory Panel" (VAP) that shared their ideas and knowledge from the literature and their experience. The "Virtual Advisory Panel" (VAP) reached a peak membership of eight, briefly, but consisted of six participants for most of the Phase I project. Panel members who comprised the VAP during most of Phase I are listed below, along with their affiliations and a capsule description of their expertise.

Table 1. Members of the Virtual Advisory Panel (VAP).

Consultant/organization	**Professional expertise/background**
Dr. Patricia B. Cerrito Univ. of Louisville, KY	mathematics and statistics; teaching mathematics to students with ADHD via CAI; NCTM standards; neural networks
Dr. Carol Ann Christensen Univ. of Queensland, AU	K-12 Classroom teaching; College and Graduate level teaching and research; learning disabilities; cognitive processes; early math education
Mr. Timothy Pelton Duncan, BC	mathematics and computer science; classroom math and computer science teaching; studying for PhD in Instructional Science
Ms. Mittie T. Quinn, MA Practice in McLean, VA	licensed school psychologist; public school and private practice; diagnosis and counseling for students w/ learning and emotional disabilities
Ms. Carolyn K. Gardner Linn-Benton Comm. College Corvallis, OR	teaching and research in adult basic education; works with handicapped and learning-disabled adults; technologies for math/science instruction
Mr. Ronald A. Sveen Calgary, Alberta, CA	taught math at two schools for students with severe learning difficulties; associate of Learning Centre; now Learning Strategy Specialist and teaches math at junior-high level

The panel held wide-ranging discussions of LD's and arithmetic training via email in two intensive one-month sessions during this project.

Here are some of the most promising arithmetic-teaching strategies gleaned from the literature and the VAP discussions:

- group the arithmetic facts in conceptually helpful ways [16; VAP]
- use animations to illustrate arithmetic concepts [VAP; existing software and teaching techniques]
- help the student discover strategies [22; VAP]
- vary the mix of new vs. review facts to optimize the learning rate [11]
- vary the visual symbolism [VAP; existing software and teaching techniques]
- vary the visual/audio mix and functions [VAP]

These represent some of the independent control variables that can be adapted to individual student's needs.

Two VAP Straws Polls were held to ensure accurate communication of the advisors' opinions. The second (Table 2) was definitive, and provided a foundation

Table 2. VAP opinions are summarized, in part by the results of "Straw Poll II."

Item	**Points**
II.2 Adapt instructional content to student's current factual knowledge and speed of learning	6.9
II.3 Implicitly teach student basic arithmetic concepts	3.9
II.5 Provide animations to illustrate basic math concepts & operations	3
II.7 Provide positive coaching & feedback to student during instruction	2.9
II.22 Diagnose student's conceptual math errors and present remedial information	2.4
II.11 Provide student with verbal/audio instruction & advice	2
II.21 Perform NN-based, on-line assessments that match current assessment standards & categories	2

for the decisions made about the project's scope and content in both teaching and assessment. Many of these ideas have been incorporated in the work of Phase I and the plans for subsequent research and development. In particular, the VAP input was used to choose the approach to performing student assessments: highest priority was given to those assessment tasks that contribute directly to the teaching tasks at hand.

4. THE "MAN" ARITHMETIC TEACHING AND LEARNING SIMULATOR

The three components of the adaptive computer code being developed in this project (see Fig. 1) operate together within a single "framework" provided by the **MAN** program. **MAN** provides the functions of start-up and termination, display, and shared storage used by all modules for exchanging code status and messages. The presentation format was kept simple to facilitate rapid development. The next three sections will describe the functional modules: the adaptive tutor **TUT**, the student simulator **STU**, and the guidance/assessment module **GUY**.

5. THE ADAPTIVE TUTOR

This section will report highlights of the "engine" devised for an adaptive-network-based arithmetic tutor module capable of adapting to the learning needs of LD students. The tutor module, "**TUT**", is based upon a simple, previous program that was successfully used to teach subtraction facts to one such student. Considerable additional thought and work have led to several potential improvements, in the process of implementing the concept in the C++ language on the PC/Windows platform. The improvements can be summarized as follows:

- FactBases for number recognition/typing, addition, subtraction, multiplication, and division facts
- FactGroups within the FactBases encourage use of learning strategies and recognition of patterns
- Phasing scheme sequences the overall learning progression
- Select-Net interprets the student's knowledge and performance and improves choice of facts for presentation
- Multi-speed adaptation of presentation
- Multi-timescale tracking of student's knowledge

As frequently occurs when computer codes improve, the code "**TUT**" (as in tutor) is considerably larger than its predecessor: the current version has expanded to about 4000 lines, or about 8 times the original version's size!

A. Student's Performance Measurements

The tutor module makes a few physical measurements as the student responds to the presented arithmetic facts. First, the code measures the response time for each presentation. Second, the code logs the presentation sequence and any erroneous responses. The logging features offer human-auditable post-analysis capabilities that are useful for debugging and assessing the instructional module and also provide a growing database that may be useful for developing higher-level instructional decision-making approaches.

B. Select Net

At the core of the tutor is a custom-designed Adaptive Network, dubbed “Select Net.” This network performs the fact selection for presentation, based upon processed measurements of the students responses. The next paragraph will give a simplified description of Select Net’s operation. Readers who would like background material on NN’s will find a fairly readable account in the book by Wasserman [23].

The Select Net maintains a fact-by-fact record of the student’s recent, averaged response times. Each response time is converted into a probability of fact presentation, using the network’s activation function (Fig. 3). For example, if the student responds in less than 2 seconds, the fact is considered as “known” and the relative probability for presenting that fact is set low. If the average response time is greater than 2 seconds, the fact is considered to be in “learning” status. For response times between 2 and 4 seconds, the relative probability of presentation increases linearly. If the response time is longer than 4 seconds, the relative probability is fixed at a maximum value. The relative probabilities of all facts are added in sequence, and a random number is chosen that selects a fact for presentation (Fig. 4).

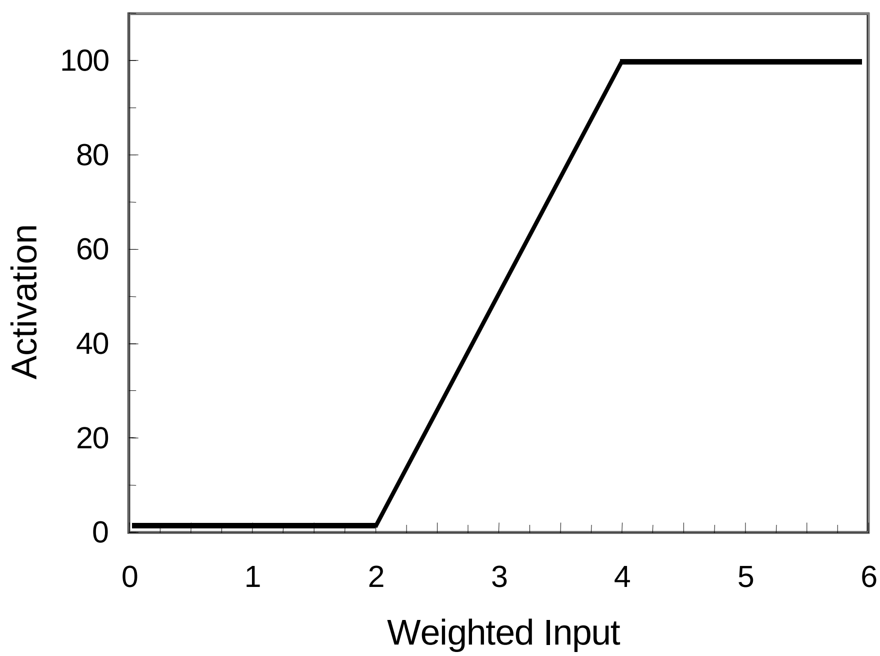

Figure 3. Select Net’s activation function.

0+0	**0+1**	**0+2**	**0+3**	**Etc.**

Figure 4. Select Net uses roulette-style selection to choose facts for presentation. In this schematic, the circumference of the roulette wheel is straightened into a line. The width of each “slot” corresponds with the student’s recent response time for each arithmetic fact. As drawn, the student’s response to the “0+3” problem is slower than that for “0+0”, for example. Thus, the “0+3” problem will be presented with greater frequency.

6. THE STUDENT SIMULATOR

The student simulator module, **STU**, is an implementation of the learning model discussed above, including a simple response module, as shown schematically in Fig. 5. Two very different Neural Networks serve as the Working Memory and the Long-Term Memory. The Working Memory Network has been developed specifically for this application as part of the Phase I project. The NN used for the Long-Term Memory is the CNLS network, which has been used by the author and colleagues in a number of other projects at Los Alamos National Laboratory.

It is worth noting at the outset that significant aspects of human capabilities will not be adequately represented in this model: some more complex characteristics of human learning are difficult to simulate. A very important aspect of human learning is metacognition, a process that is completely absent from the student simulator developed here. Nonetheless, the student model in its present form provides a very useful benchmark for testing the adaptive tutor, and has already lead to some insights about the learning processes in human students.

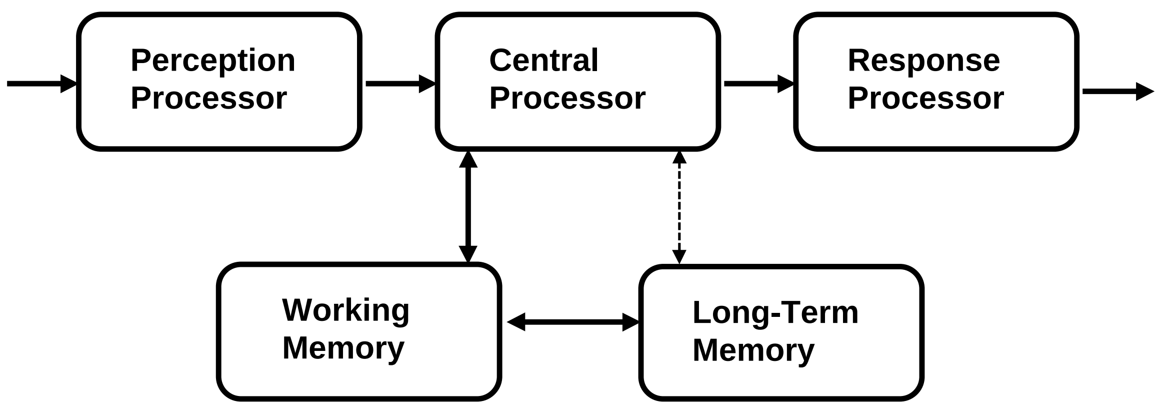

Figure 5. Architecture of the student simulator module, **STU**.

A. Central Processor

The “Central Processor” module currently contains mainly the functional “glue” that binds the other parts of the student simulator together. The central processor serves as a supervisor, calling the other four modules, as needed.

B. Perception and Response Processors

The perception model uses a probability matrix that specifies the “odds” of how a given input digit will be represented (or misrepresented) in the student’s mind. To realize the model within Phase I resources, ad hoc probabilities were assigned. A simple conceptual motivation was used, based on visual similarities of the digits and the author’s experience with reading numbers. For example, the worst case digit was taken to be 9, with a significant probability of being misrepresented as a 6 or possibly a 0. A scaling technique has been chosen that allows a single parameter (the “perception” coefficient) to smoothly vary all the probability matrix entries.

The “Response Processor” uses the same architecture as the perception model. In the current implementation, the student is not required to type any entries except digits, so the response matrix contains the probabilities for converting a given internal digit into a particular computer entry. For conceptual purposes, the response matrix has been initially chosen to be identical with the perception matrix.

C. Working Memory: WM Net

The “Working Memory Network” (WM-Net) has been newly invented as part of this work. WM-Net is a symbol-oriented network that features short-term lifetimes for contents and network connections. Each node is capable of storing one symbol. The network “learns” by forming conceptual, symbolic “fact strings”. For example, suppose the WM has to learn “2+2=4.” WM-Net connects a series of five nodes and puts one of the symbols into each of the linked members of the string, in sequence. The network includes modeling for various characteristics and flaws that are exhibited by human short-term memory. For example, the memorized string “2+2=4” can be “forgotten” through breaking one or more of the connections or by distortion of one or more of the symbols.

D. Long-Term Memory: CNLS Net

The choice of the Neural Network to serve in the role of “Long Term Memory” had to satisfy several constraints. Desirable properties sought include

- incremental learning over a controllable number of iterations,
- good numerical stability and robustness,
- low computational overhead, and
- high accuracy.

The Connectionist Normalized Local Spline Network (CNLS Net, Fig. 6) chosen for

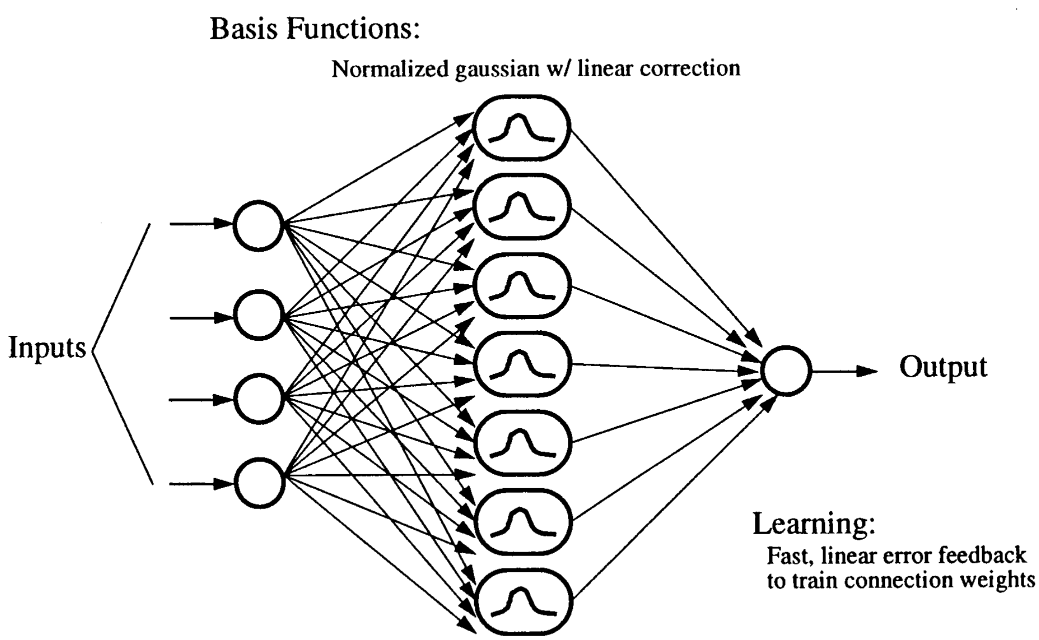

Figure 6. Schematic of CNLS-Net, indicating a few attractive network features.

this function satisfied all these conditions. CNLS Net has been described in detail in several publications [3,4]. A brief overview is included here.

CNLS-Net can be thought of as a “mapping device” that takes several input quantities and maps them into an output quantity. The input quantities are distributed to each of the NN’s nodes, and each node performs a simple calculation to determine what output should be produced, using the node’s basis function and the current “weight” settings. The NN “learns” by using an iterative method to adjust the nodes’ weights until a given dataset is matched with minimal error.

Choosing the NN is only the first step of developing a successful NN application. The next step is to choose inputs and outputs for the network, scaling factors, and the NN architecture (number of inputs, nodes, and values for the control parameters). In this work, the digits and operation sign of the arithmetic problem are converted into a numerical code; the NN calculates a numerically coded output that is translated back into the digits of the answer to the problem. The storage mechanism is “analog” (like the setting of a light dimmer): the information stored in each neuron is a continuous quantity, adjusted to a value that is to be retained in memory. The choice of an analog mechanism for long-term memory function is largely speculative. The network’s behavior, discussed below, has been quite suitable for the purpose here.

7. THE GUIDANCE/ASSESSMENT MODULE

The Guidance/Assessment module currently serves two purposes. First, many of the guidance elements are used also by the adaptive tutor to make decisions during the teaching/presentation process. Second, the guidance module presents summary information on the student’s learning progress and knowledge status to a supervising teacher.

The Guidance/Assessment Module, “**GUY**”, is the youngest of the **MAN** components, so it should be considered to be a work in progress. In its current implementation, **GUY** generates a summary report at the end of each session that provides a supervising teacher with a snapshot of the progress and status of the teaching/learning process. Currently, GUY provides four kinds of output at the end of each student’s session:

- Most recent session summary
- FactBase status summary
- FactBase knowledge map
- Learning history

Much of this information is presented graphically for quick assimilation by the supervising teacher. The learning history curve is available to the student as a clear indication of concrete progress.

8. SIMULATION RESULTS

This section will present several examples of early simulations using MAN, the composite Teaching/Learning Laboratory developed in Phase I. The reader should be aware that the simulator is nearly entirely a theoretical (and in some areas hypothetical and speculative) construct, and no claims to realism or fidelity to human teaching and learning are made beyond the general features described below.

A. Example "MAN" Simulation Dialog

An excerpt from a MAN trial/game transcript with WM-Net in operation is shown below:

```
tut: Q:     0 + 17 =
stu: P:     0 + 17 =
stu: W:      17       cert=  0.95
stu: L:     16       cert=  0.91
stu: LW:     16       cert=  0.86
stu: N:     17
stu: R:     17
stu: A:     17
tut:                        dtime= 1.746000    0.161241
tut: Right!
wmn: Disconnected snode 8.  39 in use.
cnn01: Trn stop (trndevstop) at epoch 756, sdev= 2.4659e-003
cnn06: Trn stop (maxtrndepochs) at epoch 780, sdev= 5.3491e-003
cnn11: Trn stop (maxtrndepochs) at epoch 804, sdev= 3.7557e-003
cnn16: Trn stop (trndevstop) at epoch 820, sdev= 1.4606e-002
```

In sequence, **TUT** poses the question (Q), then the student perceives the question (P). Next, the student Central Processor module consults the Working Memory and obtains its response (W), along with a "certainty" rating between 0 and 1. The student module, **STU**, also initiates a query to Long Term Memory (LTM), which responds with its answer and certainty rating. In the mode of operation selected for this run, the LTM response (L) is first transferred to WM, and then returned to the Central Processor (LW) with an adjusted certainty. In this example, the LTM response is both incorrect and less certain than the WM response, so the "Net" response (N) is taken to be the WM answer. The Response Module processes the output (R) and the student's answer (A) is sent back to **TUT**. The tutor confirms to the student that the answer is correct, and **STU** uses the confirmation to update the WM and LTM modules. During this transaction, one of the temporary connections in WM-Net (wmn) expires during the WM-Net refresh operation, leaving 39 data chains in use.

B. The "Super-Student" Reference Case

The first case studied in detail was the "Super Student". This case was constructed to provide a "blue sky" example to ensure that all modules and basic simulation functions were working according to intent. Figure 7 shows a plot of the number of facts "known" as a function of game or trial number for this example. At TUT's maximum allowed presentation rate, the 926 facts for Typing, Addition, Subtraction, Multiplication and Division are learned and reviewed with an average of 3 presentations per fact, over the course of 119 games. It might be possible to reduce the minimum number of presentations per fact to 2, but this could be done only at the risk of reducing the assurance that the student actually knows every fact.

In actual usage, the learning of the four different arithmetical operations would probably occur in separate blocks, and in at least two different grades. This would break up the very large memorization task into more tolerable blocks. The importance of splitting the different operations' learning is partly conceptual, but also completely reasonable in terms of work load. Students that are slower than the Super Student dealt with here, would spend correspondingly more time learning each fact, as indicated by the additional examples, below.

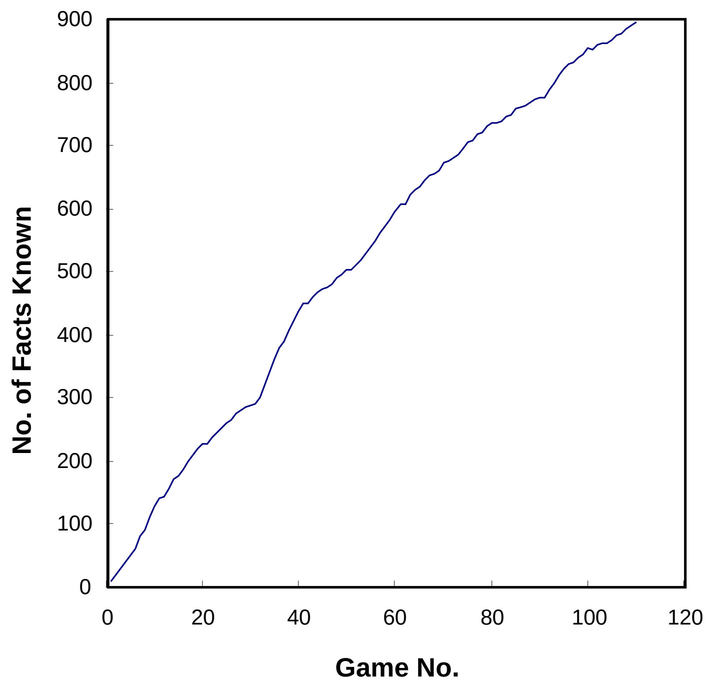

Figure 7. Learning history for "Super Student", a simulation created to learn at **TUT**'s maximum rate. Small variations in learning rate occur due to the phasing structure and the requirement that a student complete a given logical set of facts (Fact Group) before proceeding to the next set.

C. Varying LTM Learning Rate: Successful Test of Adaptive Tracking

The series of simulations produced by varying the learning rate of the Long-Term Memory (CNLS-Net) has proven to be fundamental for verifying that **TUT** can track over a fairly wide range of student learning rates while maintaining reasonably near-optimum performance. The Student Simulator presents **TUT** with a set of students for which the internal learning rate is precisely adjustable. **TUT**'s task is to use only the information from each student's responses to determine the optimum rate for presenting new and review facts. Earlier versions of **TUT** could perform this "student tracking" only poorly. The most recent **TUT** code passes this stringent test nicely after improvements to the adaptive training algorithms. This section discusses details of the testing methods and results.

The Super Student was set up so that CNLS-Net performed 8 internal learning "rehearsals" for each fact presentation. The simulations of this series reduced this internal rehearsal number by successive factors of 1/2: to 4, 2, and 1. Using this approach, an exactly calibrated set of students that differ by an overall factor of 8 in LTM learning rate was available to test **TUT**'s adaptive capabilities.

This series led to important qualitative tests and improvements of the adaptive tutor, as well. After the addition of the WM-Net and the implementation of the full student simulator architecture, this series pointed out the importance (known to every real teacher!) of fact review in ensuring that the learning is occurring at the Long-Term Memory level. **TUT**'s initial runs of this series using the full student simulator leaped forward: all four "students" in the series apparently learned the facts at the maximum **TUT** speed. Not much reflection was required to realize that if **TUT** presents the facts in only one game, only the short-term or Working Memory is tested. The students with lower LTM learning rate would not be able to recall the facts once the Working Memory recall of the facts had faded. This led to the creation of the multi-timescale knowledge tracking that allows **TUT** to perform properly when the Working Memory is "faster" than the Long-Term Memory.

Learning curves for the four students with successively decreasing LTM learning rate are shown in Fig. 8, using the current version of **MAN**. If the adaptive tutor behaved perfectly, each of these curves would be separated by a factor of 1/2. The current version of **TUT** deviates from the perfect spacing by a moderate factor, indicating some error in the adaptive tracking.

The accuracy of the adaptive tracking can be seen more readily by viewing the endpoints of a fixed number of games as a function of LTM learning rate, as shown in Fig. 9. The plot shows the endpoints of the 60-game runs shown in Fig. 8, plus one additional run at a learning rate of 0.75 times the maximum (6 internal rehearsals). The linearity of the actual **TUT** adaptive tracking results compares quite favorably with the ideal line shown, except for the "Super Student". In this case, **TUT** has reached the minimum permitted fact repetition, and the student cannot advance faster than the limit imposed. Even for the Super Student, **TUT** is only 25% below optimum. The adaptive tutor's performance may be close enough to ideal for most real-student training purposes.

D. Perception and Response Accuracy: Preliminary Answers

In this test series, I varied the perception or response accuracies (for digits only--no inaccuracy for reading the operation signs) using the same pattern of errors in either case. This was done to help gain an understanding of whether perceptual errors and response errors have different effects when they occur before or after the "brain's" internal learning processes. Also, this study gives the simulations' estimate of how a given level of perceptual or response errors alters the "student's" effective learning rate.

Figure 10 shows the effects of perception or response inaccuracies, for numerals only. The ordinate is the number of facts learned over the course of 60 games, normalized to the best performance obtainable. The abscissa is the probability of a mistake in perceiving or generating a response for the digit 9, which is the worst-case among all the digits. Simulations suggest that a 10% error rate slows down learning by a factor of 1/10.

One could reasonably ask two kinds of questions about the simulation results. First, can we understand the learning-rate decrease produced by perception or response inaccuracies in any simple way? Second, is there any systematic and important difference between inaccuracies in perception compared with inaccuracies in response? These are areas where considerable thought and research are needed to arrive at more definitive conclusions, but which are beyond the scope of this work.

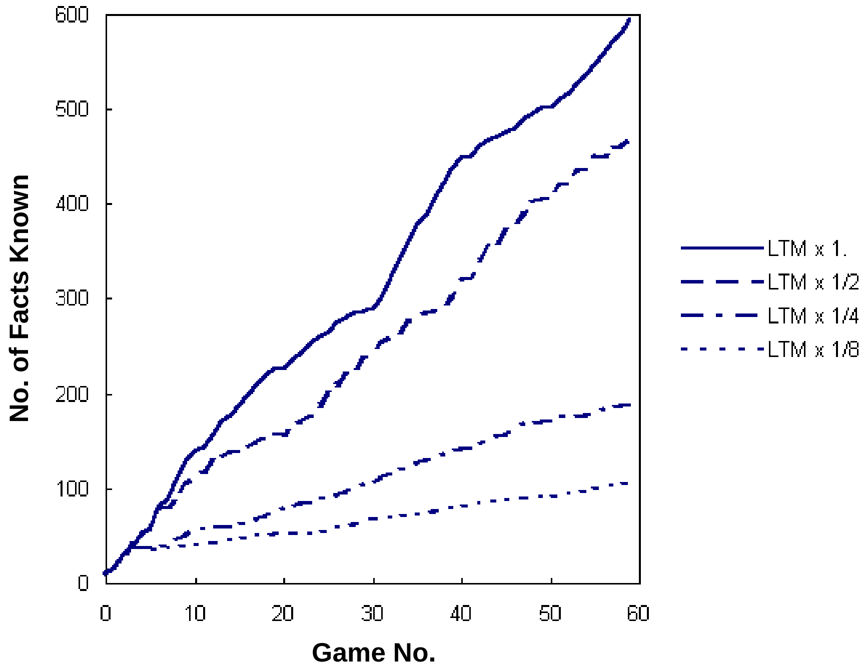

Figure 8. Learning curves for series of simulations with varying Long-Term Memory learning rates.

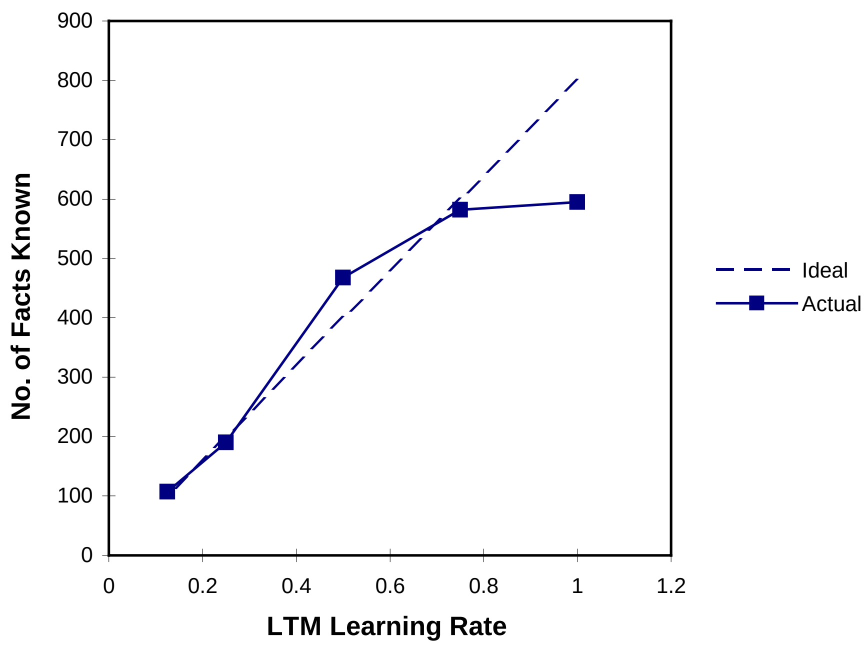

Figure 9. Adaptive tracking accuracy can be seen in this plot of total no. of facts learned in 60 games as a function of LTM learning rate. Tracking error is 5-17% for all simulated students except the "Super Student" simulation (25% error).

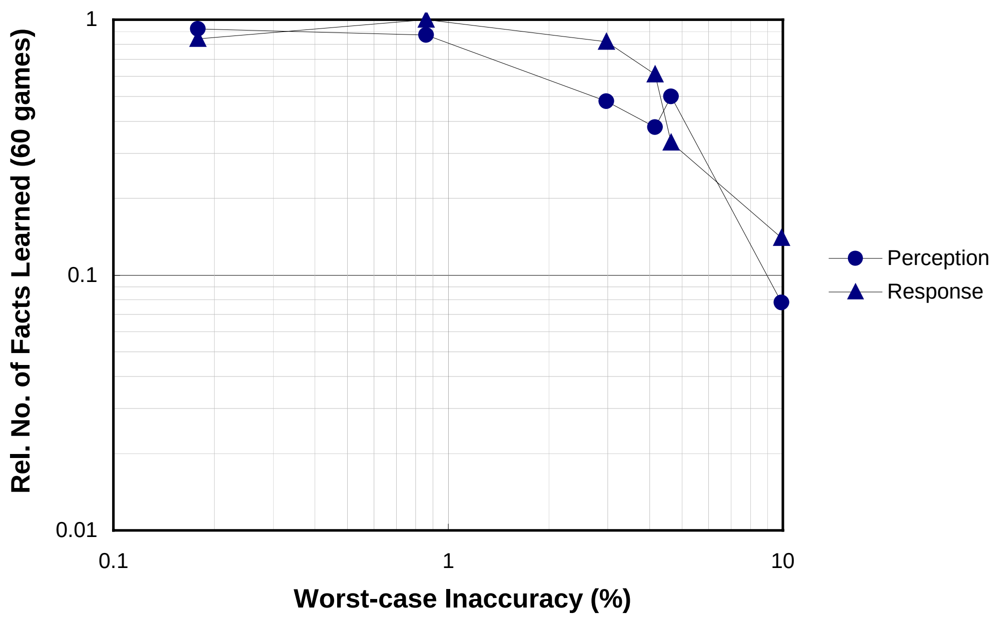

Figure 10. Effects of perception and response inaccuracies. Plot shows the relative number of facts learned by the simulated student (over 60 games, and relative to the best achievable) as a function of the inaccuracy rate for perceiving or responding to the digit 9, which is the worst case for the models used.

E. Varying Response Speed

Next, let us consider the capability of **TUT** to adapt to variations in a student's mean response speed or in the deviation of response times. In terms of response speed, the findings are both easy to understand and very favorable. The effect of varying mean response speed over a factor of 5 to 1/5 (times the author's measured response speeds) is nearly completely compensated for by **TUT**'s adaptive algorithms: the no. of facts learned in 60 games changes by only about 11%. Of course, the slower student will spend longer times completing the TUT tasks. But he will not pay any penalty in number of repetitions required to complete the course of learning desired.

9. PRELIMINARY TUTOR TESTING BY REAL STUDENTS

Two kinds of very preliminary pilot tests have been performed: (1) tests by adult educators who have experience with LD students, and (2) tests with two LD students who are learning basic arithmetic.

Preliminary tests of the **TUT/GUY** codes have been performed by some of the VAP panelists. Three panelists reported successful installations and initial use (one panelist also tried **TUT** briefly with her son). This testing has provided a number of comments and insights that bear on usability by others and with students, and this information will be valuable to future development.

The second kind of pilot testing was performed by Mr. Ronald Sveen, who is himself a tutor for LD students who are having difficulties learning arithmetic. Mr. Sveen informally tested **TUT** with two LD students for one session each, and he has observed and reported on the students' comments and experiences. This testing was too brief to obtain any overall indications of the usefulness and value of using **TUT** to help such students, but did provide a number of useful observations and comments, and also some real-student data. However, both students became acclimated quickly and seemed to find the practice a positive experience.

10. CONCLUSIONS and STATUS

This research has successfully developed tools and techniques to establish the feasibility of using Adaptive and Neural Networks to efficiently train LD students in fluent recall of arithmetic facts. The teaching and learning simulation code **MAN** serves as a framework for an adaptive tutor (**TUT**), a student simulator (**STU**), and a guidance/assessment module (**GUY**). Each of the three components uses adaptive or neural networks to perform a central part of its function.

In this project, three Neural Networks have been applied. A well-known existing NN, called CNLS Net, serves as the student simulator’s Long-Term Memory. Two new NN’s have been invented. Select Net provides information to the adaptive tutor to control the strategic presentation of new and review facts; in addition, it provides information to the guidance/assessment module to communicate to the supervising teacher the status of the student’s progress and knowledge. The new Working-Memory Net (WM Net) provides a symbol- and fact-oriented Short-Term Memory for the student simulator. Perhaps more properly called a neuro-model, WM Net simulates many of the salient characteristics of human short-term memory.

Simulations for “students” having a variety of learning deficits have shown excellent ability to adapt the presentation of new arithmetic facts in order to optimize the student’s learning. The adaptive tutor has successfully tracked over a factor of 8 in the student’s Long-Term Memory learning rate, over a factor of 25 in student’s mean response time, and over perception and response (worst-case) error rates of 0-12%. The first two adaptations can be theoretically understood to be near-optimum.

The **TUT** and **GUY** modules have been fielded as stand-alone codes to permit pilot testing by educators and with real LD students. Opportunities to pilot-test the codes with both kinds of users have arisen, and have been pursued. Three adults, one “ordinary” child, and two LD students have used the adaptive tutor for one or more sessions. Useful and encouraging feedback has been obtained.

There remain a number of open issues, and also much work is needed to convert the three “engines” into usable and viable products. Here is a brief list of research and development issues that remain to be addressed:

- How well does the adaptive tutor work with various LD students?
- Is the tutor handling perception and response errors at near-optimum?
- Can the tutor detect and adapt adequately to attentional deficits?
- What interventions work well with what LD’s?

- What conceptual ingredients are appropriate?
- How do game/story-lines affect student progress and motivation?

In addition, a number of features need to be added that improve the robustness and attractiveness of the user interface, and allow **TUT** to be used with a variety of student starting points and needs.

ACKNOWLEDGMENTS

I extend my sincerest thanks and regards to the members of the Virtual Advisory Panel. I thank each for contributing to the forthright discussions and maintaining a professional approach. Working with the panelists has been a real pleasure.

In addition to the VAP panelists who've participated in the testing, I thank Mr. Brooks Masterton of the Peel Board of Education, Brampton, Ontario, Canada for his interest and work towards real-student pilot testing; and especially Mr. Ronald Sveen, who performed the very first pilot tests with LD students.

This work was supported by the U. S. Department of Education. I appreciate the efficiency with which the Department of Education and its agents Carol B. O'Leary and Donna M. Hoblit administered the project.

REFERENCES

[1] References to a large literature in Special Education can be found in the *ERIC Database* (Educational Resources Information Center, U.S. Department of Education, Office of Educational Research and Improvement, Washington, DC, 1995). Access information can be obtained by calling (202) 219-1846, or by sending an email request to "AskERIC@ERICIR.SYR.EDU".

[2] A compendium of recent NN applications can be found in the **Proc. of the World Congress on Neural Networks-- San Diego** (Lawrence Erlbaum Associates, Hillsdale, NJ, 1994).

[3] R. D. Jones, Y. C. Lee, S. Qian, C. W. Barnes, et al., "Nonlinear Adaptive Networks: A Little Theory, A Few Applications," in **Proc. of the First Los Alamos Conf. on Cognitive Modeling in System Control**, June 10-14, 1990, Santa Fe, NM; and ref.'s therein.

[4] "Optimization and Control of a Small-Angle Negative Ion Source Using an On-line Controller Based on the Connectionist Normalized Local Spline Neural Network," W. C. Mead, P. S. Bowling, S. K. Brown, R. D. Jones, et al., *Nucl. Instr. & Meth. in Phys. Res.* **B72**, 271-289 (1992).

[5] J. A. Anderson, "Neural-Network Learning and Mark Twain's Cat," *IEEE Communications Magazine*, p. 16 (Sept., 1992).

[6] W. Bechtel and A. Abrahamsen, **Connectionism and the Mind** (Basil Blackwell, Oxford, 1991).

[7] H. L. Swanson, "Short-Term Memory and Working Memory: Do Both Contribute to Our Understanding of Academic Achievement in Children and Adults with Learning Disabilities?" *J. Learning Disabilities* **27**, 34 (1994).

[8] C. A. Christensen, "Thinking and Learning," Manuscript in preparation, *Priv. comm.*, Nov., 1995.

[9] P. T. Ackerman, R. A. Dykman, and M. Y. Gardner, "Counting Rate, Naming Rate, Phonological Sensitivity, and Memory Span: Factors in Dyslexia," *J. Learning Disabilities* **23**, 325 (1990).

[10] C. Watson and D. M. Willows, "Information-Processing Patterns in Specific Reading Disability," *J. Learning Disabilities* **28**, 216 (1995).

[11] N. L. Cooke, "Effects of Using a Ratio of New Items to Review Items during Drill and Practice: Three Experiments," *Education and Treatment of Children* **16**, 213 (1993).

[12] S. S. Zentall, Yvonne N. Smith, Y. B. Lee, and C. Wieczorek, "Mathematical Outcomes of Attention-Deficit Hyperactivity Disorder," *J. Learning Disabilities* **27**, 510 (1994).

[13] N. C. Jordan, S. C. Levine, and J. Huttenlocher, "Calculation Abilities in Young Children with Different Patterns of Cognitive Functioning," *J. Learning Disabilities* **28**, 53 (1995).

[14] C. Weedon, "Specific Learning Difficulties in Mathematics," Stirling Univ., Dept. of Education (1992).

[15] C. M. Okolo, "The Effect of Computer-Assisted Instruction Format and Initial Attitude on the Arithmetic Facts Proficiency and Continuing Motivation of Students with Learning Disabilities," *Exceptionality- A Research Journal* **3**, 195 (1992).

[16] K. Garnett, "Developing Fluency with Basic Number Facts: Intervention for Students with Learning Disabilities," *Learning Disabilities-- Research and Practice* **7**, 210 (1992).

[17] P. T. Ackerman, J. M. Anhalt, and R. A. Dykman, "Arithmetic Automatization Failure in Children with Attention and Reading Disorders: Associations and Sequela," *J. Learning Disabilities* **19**, 223 (1986).

[18] T. V. Hanley, "The Future has been a Disappointment: A Response to Woodward and Noell's Article on Software Development in Special Education," *J. Special Ed. Technology* **XII**, 164 (1993).

[19] C. M. Okolo, C. M. Bahr, and H. J. Rieth, "A Retrospective of Computer-Based Instruction," *J. Special Ed. Technology* **XII**, 1 (1993).

[20] C. R. Greenwood and H. J. Rieth, "Current Dimensions of Technology-Based Assessment in Special Education," *Exceptional Children*, p. 105 (Oct./Nov., 1994); see also the articles he introduces in this special issue on assessment technology in Special Education.

[21] L. S. Fuchs, D. Fuchs, C. L. Hamlett, and C. Ferguson, "Effects of Expert System Consultation within Curriculum-Based Measurement, Using a Reading Maze Task," *Exceptional Children*, p. 436 (March/April, 1992).

[22] H. L. Swanson and J. B. Cooney, "Strategy Transformation in Learning Disabled and Nondisabled Students," *Learning Disability Quarterly* **8**, 221 (1985).

[23] P. D. Wasserman, **Neural Computing** (Van Nostrand Reinhold, New York, 1989); and ref's therein.