The Acquisition of Intellectual Expertise: A Computational Model

Lisa C. Kaczmarczyk (lisak@cs.utexas.edu)
Department of Computer Sciences, The University of Texas at Austin
1 University Station C0500, Austin, Texas 78712 USA

Risto Miikkulainen (risto@cs.utexas.edu)
Department of Computer Sciences, The University of Texas at Austin
1 University Station C0500, Austin, Texas 78712 USA

Abstract

Intellectual expertise means knowledge and ability that a person has that allows them to solve complex problems. It is important to understand how people become experts so that we can improve educational strategies, and help learners achieve their full academic potential. Unfortunately, the process of acquiring intellectual expertise is not well understood. Artificial neural networks (ANNs) have already been successful in modeling other types of human learning. This paper shows that they can also be trained as a model of expert human learning, and address many of the difficulties found in trying to study expertise in humans. The results confirm three hypotheses: (1) An artificial neural network can be used as a model to investigate how people learn under different training scenarios; (2) Different methods for delivering the training material result in different final performance; (3) The best performance is achieved by incrementally increasing the complexity of the material. These results provide educators with computational evidence that structured, integrated delivery methods are better for learners than oversimplification and isolation of learning tasks.

Introduction

An intellectual expert has achieved a level of cognitive development in which she or he can rapidly grasp subtleties of complex problems, and produce very high quality solutions. A goal of formal education is to help students achieve an expert level of understanding in their chosen field. It is important to understand the nature of expertise so that we can improve educational strategies. As a result of many research studies about expertise, we know a lot about the characteristics of experts. However, there is a lot we do not understand about how to become an expert. It is not easy to create experts, whether human or computational. The learning process is complex and human studies are difficult. Understanding how to acquire intellectual expertise has proven elusive for educators, psychologists and students alike.

A primary goal of the study reported here is to increase understanding of the process by which humans become intellectual experts. In particular, how can people develop the ability to look at a problem statement and immediately select the best solution strategy? The second main goal is to understand this process in the context of formal instruction; specifically, how does the strategy by which material is delivered to the learner affect learning and conceptual development?

This paper presents results from a series of computational experiments examining how different delivery methods influence learning and conceptual development. These experiments use a real-world adult educational problem: the ability to identify correct solution strategies for calculus integration problems. The goal is to show that an artificial neural network can be used as a model to investigate how people learn under different training scenarios, and that different delivery methods result in different overall performance. The main results include: (1) errors are higher on final exams when different problem types are learned in isolation; (2) cramming just prior to taking final exams does not significantly improve performance. Different delivery strategies affect learning in different ways: (1) traditional sequential delivery methods inhibit learning and retention; (2) integrated delivery methods increase learning and retention; (3) the best performance comes from delivery methods that incrementally increase the complexity of material. These results can be applied to developing better training methods for people.

Prior Research on Intellectual Expertise

Studies of human expertise and understanding have revealed key information about experts. We know that experts and novices categorize problems differently, and that this categorization takes place before the subject attempts to solve the problem (Chi, Feltovich, and Glaser 1981). We also know that experts can categorize problems without solving them (Robinson and Hayes 1978). Finally, there is strong evidence that routine problems are solved not by intense calculating but rather by recognizing a type of problem (categorizing) and then using the stored knowledge about how to solve problems of that type (Reiman & Chi '89 referenced in (Ross and Spalding 1991).

Most studies of expertise have focused on what an expert knows, rather than the process by which she or he attained expertise. As a result, we know a lot less about this learning process than we do about expertise itself. Expert behavior does not simply follow a script: the greatest expertise is the result of long-term practice (Hayes 1989) that is consciously goal directed, self-monitoring, and self-adjusting within the setting of each particular task (Garner 1990). In addition, many studies have shown that meta-cognition (self-appraisal and self-management of cognition) is critical for successful academic learning (literature surveyed by Paris and Winograd (1990)). Since we know that experts categorize extremely well, it is possible that categorization ability and goal-directed meta-cognition enhance one another. When these abilities merge, intuition may be the result: there is strong evidence that experts rely upon their accurate intuition and a holistic recognition of appropriate actions (Dreyfus and Dreyfus 1986).

Published in *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, 2004.

Cognitive scientists have often studied mathematics learning, due to the abstract nature of its concepts. Bruner has even suggested that learning mathematics may be viewed as a microcosm of all intellectual development (Bruner and Kenney 1965). A particularly interesting early connectionist model of mathematics learning was presented by Viscuso, Anderson, and Spoehr (1989). Their artificial neural network (ANN) simulated qualitative reasoning while doing multiplication. In summarizing their model, Viscuso et al correctly pointed out that the most important contribution of their model was that it mimicked the manner in which experts rely not so much on formal logic and rules but on their "sense" of what is correct. Another interesting ANN system learned to perform arbitrarily long addition problems (Cottrell and Tsung 1993). Their model learned the implicit underlying rule of addition. This system showed that ANNs can account for conceptual development: the network learned an important concept on which it had not been explicitly trained. In the decade since these studies were published, there has been quite a bit of work in related areas, such as the development of basic numerical abilities in infants and children (literature surveyed in (Ahmad, Casey, and Bale 2002)), and in childhood strategy development (Bray, Reilly, Villa, and Grupe 1997). However, we still do not understand how adult human experts learn to "sense" important concepts. It is important to understand this ability, so that we can better educate students.

The Calculus Domain

Calculus, at its most fundamental level, is based upon abstract cognitive concepts. As a result, understanding how people best learn calculus requires understanding the mind. The current educational debates over mathematics and science education partly result because we do not understand enough about how the brain produces cognition and conceptual understanding. In order to become calculus experts, students need to understand complex concepts and intuitively select the most efficient methods to solve problems. Educators need to understand what methods of delivering material will most help students achieve these abilities.

In the last decade math and science have been at the center of an increasingly wide-spread national concern with properly educating citizens for the new technological age. In college, students who want to major in science or engineering usually have to first perform well in calculus, which turns out to be a major obstacle for many of them.

In order to clearly identify what kinds of problems calculus students were having at at the University of Texas at Austin (UT), we conducted structured interviews with mathematics faculty and teaching assistants (TAs). The results fit well with the psychological literature on expert/novice behavior. Faculty and TAs reported that novice learners (in this case UT students) are often unable to select the correct solution strategy. This problem arises before they even have a chance to exhibit computational difficulties and prevents many from reaching timely, correct solutions. Conversely, the experts claimed an ability to "just see" the correct strategy, yet were unable to articulate how they knew. Probing revealed that although there are "rules of thumb" to assist in this domain, they are not comprehensive and do not cover many common scenarios. Experts instead pointed to general patterns and categorization that they have learned to recognize via extensive practice.

Successful problem solvers categorize math problems based upon underlying structural similarities and fundamental principles (Schoenfeld and Herrmann 1982). These categories are often grouped based upon solution strategies, that the experts then use to calculate an answer (Owen and Sweller 1989). How such strategies are formed is poorly understood. What regularities are most likely to be noticed, and how does the form in which the initial procedure is learned affect what is noticed? From the point of view of education, are there ways of managing how learners practice, to enhance the likelihood that they will notice these regularities, and incorporate this information into their problem-solving strategies?

One of the first instructional decisions is what order to present the material in, and how to move from one concept to the next. There are many possible orderings of material, and a computational model can be used to explore them. The model presented in this paper, described in the next section, contributes to achieving this research goal.

The ANN Model

The particular calculus problem chosen for the study is to decide whether a given integration problem should be solved with Simple Integration (Simple), Integration by U-Substitution (Usub), or Integration by Parts (Parts). This section describes the architecture of the artificial neural network as well as the training and test data, its encoding, and the experimental methods used in all the experiments described in this paper.

Architecture and Data

The model is an artificial neural network utilizing the backpropagation algorithm (for details of the algorithm see Bishop 1995) created using the LENS network simulator (Rohde 1999). The network is fully connected, and has 55 input nodes and 20 hidden nodes. The 55 input nodes make up a vector large enough to represent the features of one calculus integration problem containing up to four terms. The 20 hidden units were determined to be appropriate by experimentation; the results were not effected by small changes in size.

The input data consists of 957 calculus integration problems based upon examples found in college level calculus textbooks. Feature coding is a logical choice for representing them, given that both novices and experts use the features of a problem to determine which approach to use (Chi et al. 1981). The 55 unit input vector contains a series of 0s and 1s that map operators/operands to their location in the calculus integration problem. Short problems are padded with blanks. The vector consists of

- Four 2-unit terms representing constants and variables.
- Four 8-unit Unary Operators, representing $\sin, \cos, \tan, \cot, \sec, \csc, \ln$, exponentiation $e(x)$.
- Three 5-unit Binary Operators, representing multiplication, division, exponentiation $\hat{}$, addition, subtraction.

For example, the problem

$$3 + cos(x) - sin(y) + ln(x)$$

is coded in postfix form as: 01 00000000 10 01000000 00 10000000 10 00000010 00010 00001 00010, where the components are

01 : No Variable; Constant (i.e. 3)
00000000 : NONE (i.e. no unary operator for the constant)
10 : Variable (i.e. x); No Constant
01000000 : cos (of the variable x)
10 : Variable (i.e. y); No Constant
10000000 : sin (of the variable y)
10 : Variable (i.e. x); No Constant
00000010 : ln (of the variable x)
00010 : +
00001 : -
00010 : +

The network has three output nodes, each of which represents one of the possible integration strategies, Simple, Usub, Parts. Because the network is trained with one active target at a time, it learns to represent how confident it is in each choice (Bourlard and Wellekens 1990). For example, if the network reports activation values at 12%, 85%, 3%, then it is quite confident in the second category, considers the first category possible but unlikely, and the third category extremely unlikely (but not absolutely impossible). This percentage represents the *confidence level* that the network has in each answer.

Experimental Design

The calculus integration problems were divided into 10-fold cross-validation training and test sets (splits, or learning experiments). In each experiment the training set was input to the network, one problem at a time, in random order, and the test set was used to measure performance. Validation sets were not used because each learning experiment represented training one subject and the training time had to be constant, to compare how well the subjects learned. Three different types of learning experiments were run. Each experiment was run ten times, randomly resetting the initial network weights each time. Thus the whole study consisted of 300 learning experiments. This way it was possible to model the behavior of many different subjects and watch for both emergent patterns and individual variation.

During the test phase, there was always only one correct answer to a problem. This answer, called the "Best", was the answer suggested in a textbook, or by a calculus expert (faculty, TA). For each test problem the network reported how confident it was that the solution strategy was either Simple, Usub or Parts. If the confidence level for all solutions was below 80%, the problem was considered having "stumped" the network.

Results

Two sets of experiments (Drill and Test, Fully Integrated Learning) validated the ANN as a model of human learning. These experiments showed that the model accurately matches results from past educational research. In addition, these experiments provide insight into how the learning process occurs. The third set of experiments provided a computational prediction that a different type of learning (Incremental) produces the best performance.

Validating the Model: Drill and Test Learning

The first set of experiments, called "Drill and Test", mimicked a classic form of delivery that results in poor long-term retention in humans (Resnick and Ford 1981). In this method, concepts are introduced to the learner one at a time, with no overlap between topics. At the end of each topic, the learner is given a midterm exam (of previously unseen examples) on that concept.

After it has been trained with all concepts, the learner is given an opportunity to "cram", i.e. train on all concepts for a short period of time. At the end of all material, there is a comprehensive exam consisting of the entire test set.

In order to monitor the progress of learning quantitatively, and to compare to other approaches, each network was also tested during each epoch in two ways: (1) with the current midterm exam, illustrating the performance that the teacher would see in the classroom (Figure 1a), and (2) with the comprehensive exam, monitoring progress in learning the entire task, but broken into separate numbers for the different concepts (Figure 1b).

The main result was that the model, like humans, only remembers the most recently introduced concept well. More specifically, in 100 experiments run using Drill and Test, most networks (83%) rapidly learned to identify each of the concepts in turn (Simple, Usub, Parts). On midterm exams, the network often recognized 100% of the problems belonging to the concept that had just been studied. However, in spite of the opportunity to cram first, when the comprehensive final exam was given, these learners performed poorly, averaging 41.65% (standard deviation 6.35). The highest score was 54.55%. The remaining 17% of network learners were unable to make the switch from Simple to Usub problems, and then to Parts problems: their Usub and Parts midterms usually scored 0%. When these learners crammed and then took their comprehensive exams, they scored on average 17.29% (standard deviation 4.95), with a high score of 26.92% All learners in these experiments were extremely confident in their answers, even when they were wrong.

Validating the Model: Fully Integrated Learning

A second set of experiments mimicked human learning using an approach called "Fully Integrated Learning". This approach is inspired by the immersion experiences popular in foreign language learning (Spolsky 1989): the learner is placed in an environment where she or he is completely surrounded by the stimuli to be learned. The cognitive mechanisms that enable a foreign language student to sort out important grammatical features might not be that different from those cognitive mechanisms that sort out features of mathematical structures. In the Fully Integrated Learning experiments, there was only one training period, during which the networks were trained on all of the problem types simultaneously. During each epoch, the Simple, Usub and Parts training problems were input to the network in random order. Exams using the entire test set were given after every training epoch.

Fully Integrated Learning produced significantly better results than the Drill and Test delivery experiments (Figure

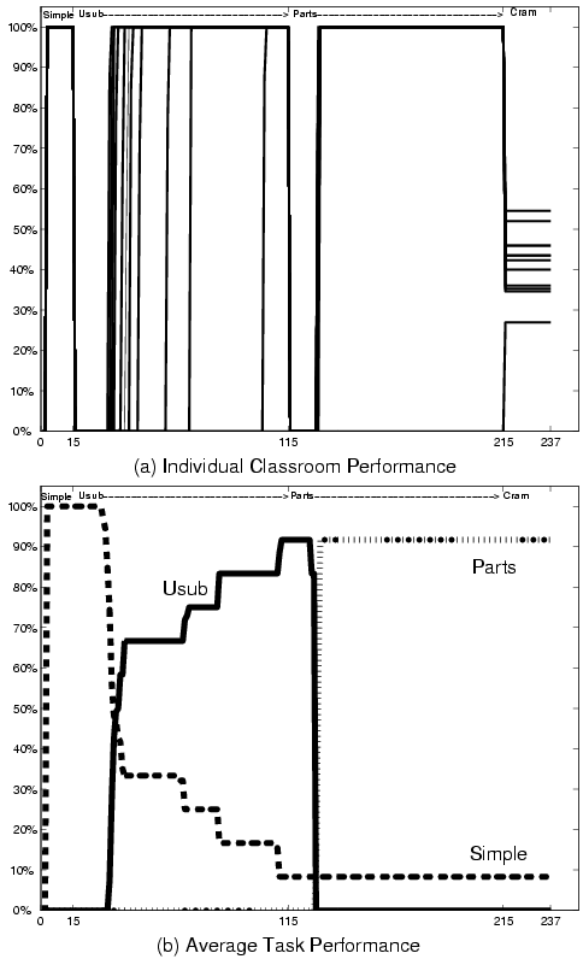

Figure 1: **Drill and Test Learning.** (a) The classroom performance of 12 representative learners, i.e. their accuracy on the current midterm (Simple, Usub, Parts, Cram periods) and comprehensive exam. Exam scores are on the y-axis, and the training epoch is shown along the x-axis. Scores on the comprehensive exam were poor - even with the aid of a cram session the highest score was 54.55%. (b) The average performance of all learners on the comprehensive exams, broken down by concept. Each problem type is forgotten when a new topic is learned.

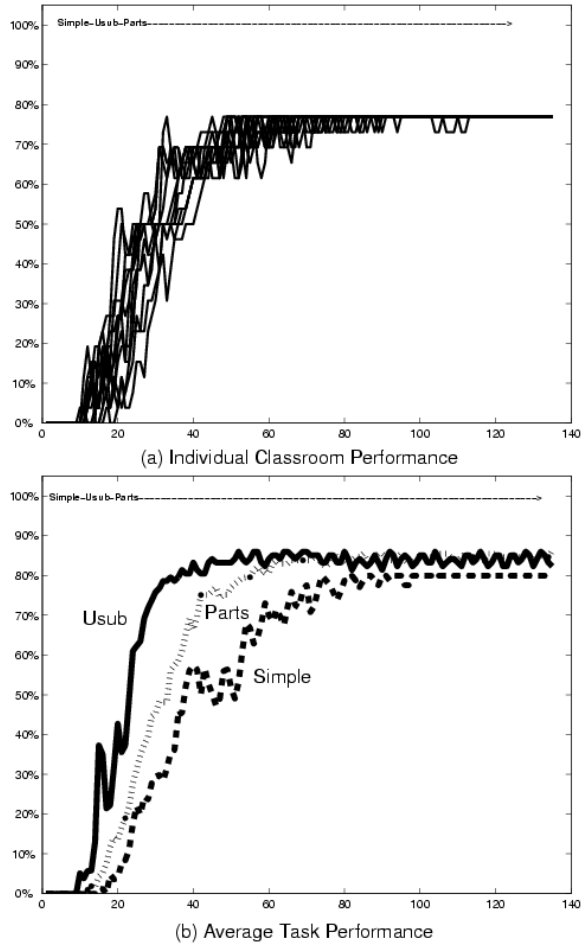

Figure 2: **Fully Integrated Learning.** (a) The classroom performance of 12 representative learners on the comprehensive test set over the course of learning. The learners initially failed the exams, but their scores rapidly increased, and finally plateaued. Improvement was not smooth, reflecting the trial and error process of learning. The best exam score was 80.76%. (b) Average performance of all learners broken down by concept. Usub problems were learned fastest, Simple problems slowest. Final results for Simple, Usub and Parts were similar.

2). The average score on the final comprehensive exam was 76.99% (standard deviation 7.94). The highest score was 80.76%. In contrast to Drill and Test, confidence in Fully Integrated learning closely reflected exam scores. The errors that were made on the exams followed a pattern of slow, gradual learning, spread across all problem types. The Fully Integrated Learning results as a whole replicated human data showing that immersion results in better longer-term retention than does Drill and Test.

Extending the Model: Incremental Learning

The third set of experiments was designed to test the hypothesis that the best learning of material is obtained by a delivery approach called "Incremental Learning". This approach is inspired by the result in the machine learning community that it is often most effective to tackle large computational tasks by starting with small problems and gradually increasing their complexity (Elman 1991; Gomez and Miikkulainen 1997). When there are a large number of co-dependent variables, it is hard to discover the role that each one plays in the problem and its solution. Therefore, an Incremental Learning delivery introduces new, increasingly complex concepts along with reinforcement of old concepts.

As with the Drill and Test experiments, there were three training periods. The network was first trained to identify Simple problems. During the second training period, Usub problems were added to the Simple problems, and for the third training period, Parts problems were added. The classroom performance was measured with Simple tests during the the first period, Simple and Usub test problems during the second, and the entire test set during the third (Figure 3a). The progress in learning the entire task was monitored with the entire test set, broken down by concept (Figure 3b). As in the Drill and Test experiments, Simple-only midterms very rapidly reached scores of 100%. When Usub prob-

lems were introduced, test scores began to fluctuate severely. Scores would drop to, or near, zero, rebound, and then drop again, as the network struggled to distinguish the new concept (Usub) from the old concept (Simple). Over time, although fluctuation continued, overall test scores increased. In a few cases, SU midterm scores reached 100%, however the majority of cases peaked at 70-75%. When Parts problems were introduced, the pattern of fluctuating scores was accentuated. Midterm scores immediately plummeted, although it is interesting to note that even the downward drop was often not smooth, but marked by brief plateaus and recoveries. Performance continued to deteriorate for longer than in the SU training segment, with scores fluctuating lower and lower. In contrast to the SU midterm scores, SUP midterm scores appeared to tighten in closer and closer to complete failure (for a while nearly all midterms fluctuated well under 20%). This behavior is predictable, because it is harder to distinguish three concepts from one another than two concepts. Eventually, performance began to improve, with prominent individual differences, as each network learner identified subtle patterns to accurately identify each concept. Eventually, virtually all midterm scores surpassed 70%. The average score on the final comprehensive exam was 81.9% (standard deviation 8.23). It is important to note that the final test results for Incremental Learning were better than either Drill and Test or Fully Integrated Learning, in spite of interim results that sometimes appeared poorer than either other type of experiment. The maximum exam SUP score was 95.6%, higher than any score reached in a Fully Integrated learning experiment. As evaluated with a t-test, the Incremental Learning final exam scores were higher than those of the Fully Integrated learning ($t = 1.9574, df = 11.869, p = 0.07423$).

The types of errors that the network made followed a pattern. As each new training period began, the network appeared to "flail", choosing first one answer then another on successive exam questions. However, this "flailing" gradually lessened and the network learned to correctly select each problem type simultaneously. As with the Fully Integrated Learning experiments, the learners' confidence levels closely reflected exam scores. The Incremental learning experiments showed that the best performance is achieved by introducing increasingly complex concepts gradually, allowing learners to build on their existing knowledge, and gradually pay more attention to finer distinctions.

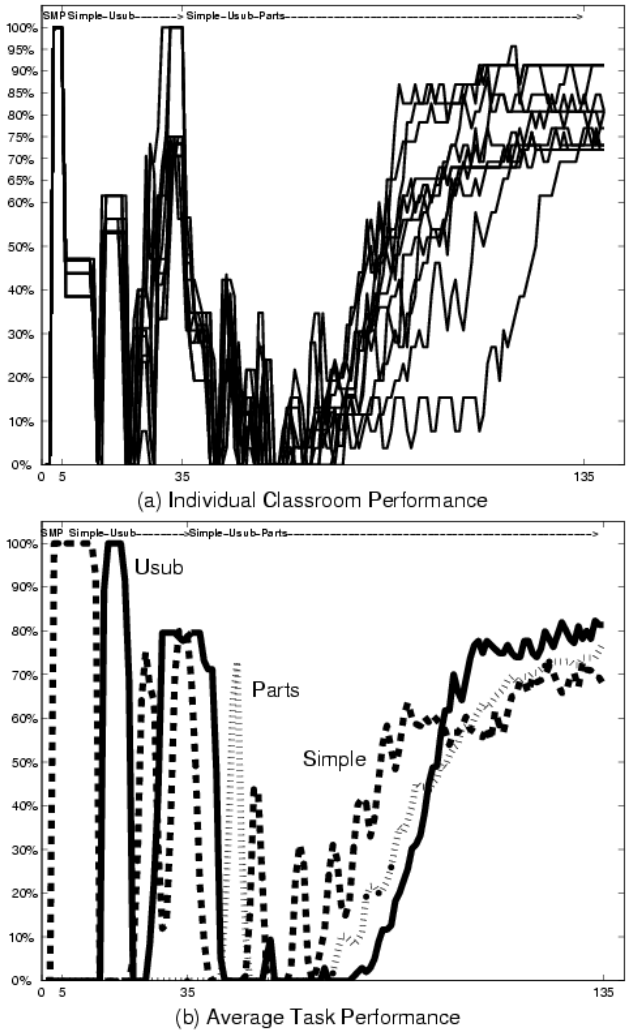

Figure 3: **Incremental Learning.** (a) The classroom performance of 12 representative learners on the current midterm (Simple, Simple-Usub, Simple-Usub-Parts). The maximum comprehensive exam score was 95.6%, higher than any score reached in a Fully Integrated learning experiment. (b) Average performance of all learners broken down by concept. Each problem type followed the same pattern of fluctuation between learning and apparent forgetting. Over time fluctuation lessened and performance improved for all problem types. Simple problems fluctuated the most and the longest.

Discussion and Future Work

Calculus integration problems that are often given to novice learners were used to study the process of learning to accurately categorize them by solution strategy. These strategies - Simple Integration, Integration by U-substitution, Integration by Parts - represent complex concepts that students need to intuitively master in order to become calculus experts. Drill and Test experiments and Fully Integrated experiments validated the model by showing that it can mimic known data about human learning. Drill and Test experiments supported the hypothesis that delivery methods that rigidly separate concepts during learning result in poor long-term retention of material. Also supported was the hypothesis that when concepts are reinforced inconsistently, only the most recently introduced concept is remembered, and that cramming does not improve learning. The nearly perfect midterm exam scores seen in Drill and Test experiments were misleading. They implied a level of interim learning and understanding which was not supported when the final exam require the learner to distinguish complex concepts.

Fully Integrated learning experiments supported the hypothesis that if problems that belong to one concept are introduced along with problems that belong to other concepts, error rates are smaller than when the same concepts are introduced separately. Over time, Fully Integrated learners performed quite well on their exams and although they are not perfect, can be claimed to have learned the task.

The results for Incremental Learning were very different from either Drill and Test or Fully Integrated learning. By introducing new problem types in a structured manner, the

network learner is allowed to focus on a smaller set of characteristics at the beginning of learning. Just as the first concept (Simple integration problems) is acquired, additional problems (Usub) are mixed in. The resulting confusion is apparent in the fluctuating midterm scores. Over time, as the learner grapples with the two contrasting problem types, confusion diminishes and midterm scores rise. When Parts problems are introduced, it becomes again more difficult to discriminate between the concepts. However, it is far more difficult to compare three related problem types than two. The confusion lasts longer and is more difficult to resolve, and individual learner differences become more apparent. Fortunately, the "priming" effect of the previous training segments allows most Incremental Learning learners to eventually do well, and in most cases better than the Fully Integrated learners.

An interesting direction of future research is to analyze the conceptual development that took place in the model during the different types of delivery methods. Using techniques such as Independent Component Analysis (ICA) of hidden layer representations it may be possible to discover how the network learners represent the problems as the learning progresses. In addition, the predictions on Incremental Learning can be tested in a study with human subjects. If confirmed, these results strongly suggest that a structured incremental approach should be used in teaching for expertise.

Conclusion

The experiments reported in this paper support the following three hypotheses: 1) An artificial neural network can be used as a model to investigate how people learn under different training scenarios 2) Different delivery methods result in different overall performance 3) Incremental Learning results in better performance than either Drill and Test or Fully Incremental learning. These results provide new insight into how humans learn complex cognitive tasks. As a result, educators have computational evidence that structured, integrated delivery methods lead to better performance for learners than oversimplification and isolation of learning tasks. They also have evidence that introducing many complex concepts at the same time does not produce the best learning either. The work encourages educators to focus on finding the optimal balance between introducing complexity and providing structured guidance. Finally, educators are reminded that interim results that reflect struggle with complex concepts will result in longer term performance gains than near perfect results in the short term.

References

Ahmad, K., Casey, M., and Bale, T. (2002). Connectionist simulation of quantification skills. *Connection Science*, 14(3):165–201.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press.

Bourlard, H., and Wellekens, C. J. (1990). Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12:1167–1178.

Bray, N., Reilly, K., Villa, M., and Grupe, L. (1997). Neural network models and mechanisms of strategy development. *Developmental Review*, 17(2):525–566.

Bruner, J. S., and Kenney, H. J. (1965). Representation and mathematics learning. In Morrisett, and Vinsonhaler, editors, *Monographs of the Society for Research in Child Development Ser. 99*, vol. 30-1. Univ. Chicago Press.

Chi, M., Feltovich, P., and Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5:121–152.

Cottrell, G., and Tsung, F. S. (1993). Learning simple arithmetic procedures. *Connection Science*, 5(1):37–58.

Dreyfus, H. L., and Dreyfus, S. E. (1986). *Mind Over Machine: The Power of Human Intuition and Expertise*. New York: Macmillan.

Elman, J. L. (1991). Incremental learning, or the importance of starting small. Technical Report 9101, CRL, La Jolla, CA.

Garner, R. (1990). When children and adults do not use learning strategies. *Review of Educational Research*, 60(4):517–529.

Gomez, F., and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342.

Hayes, J. R. (1989). *The Complete Problem Solver*. Hillsdale, NJ: LEA.

Owen, E., and Sweller, J. (1989). Should problem solving be used as a learning device in mathematics? *JRME*, 20(3):322–328.

Paris, S. G., and Winograd, P. (1990). How metacognition can promote academic learning and instruction. In *Dimensions of Thinking and Cognitive Instruction*. Hillsdale, NJ: Erlbaum.

Resnick, L. B., and Ford, W. W. (1981). *The Psychology of Mathematics for Instruction*. Hillsdale, NJ: Erlbaum.

Robinson, C. S., and Hayes, J. R. (1978). Making inferences about relevance in understanding problems. In Revlin, R., and Mayer, R. E., editors, *Human Reasoning*. Washington, DC: V.H. Winston and Sons.

Rohde, D. L. (1999). A connectionist model of sentence comprehension and production. Dissertation Proposal, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Ross, B. H., and Spalding, T. L. (1991). Some influences of instance comparisons on concept formation. In Fisher, D. H. J., Pazzani, M. J., and Langley, P., editors, *Concept Formation*. San Francisco: Kaufmann.

Schoenfeld, A. H., and Herrmann, D. (1982). Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, Cognition*, 8:484–494.

Spolsky, B. (1989). *Conditions for Second Language Learning*. Oxford, UK: Oxford University Press.

Viscuso, S. R., Anderson, J. A., and Spoehr, K. T. (1989). Representing simple arithmetic in neural networks. In Tiberghien, G., editor, *Advances in Cognitive Science*, vol. 2. Hoboken, NJ and Chichester: Horwood and Wiley.