Appearing in the Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, Lawrence Erlbaum Associates, Mahwah, N.J. Pittsburgh, PA July 22-25, 1995.

Training Regimens and Function Compatibility: Implications for Understanding the Effects of Knowledge on Concept Learning

Sheldon J. Tetewsky Thomas R. Shultz Yoshio Takane
Department of Psychology
McGill University
Montreal, Quebec H3A 1B1
tetewsky@ego.psych.mcgill.ca
shultz@psych.mcgill.ca
takane@takane2.psych.mcgill.ca

Abstract

Previous research has indicated that breaking a task into subtasks can both facilitate and interfere with learning in neural networks. Although these results appear to be contradictory, they actually reflect some underlying principles governing learning in neural networks. Using the cascade-correlation learning algorithm, we devised a concept learning task that would let us specify the conditions under which subtasking would facilitate or interfere with learning. The results indicated that subtasking facilitated learning when the initial subtask involved learning a function compatible with that characterizing the rest of the task, and inhibited learning when the initial subtask involved a function incompatible with the rest of the task. These results were then discussed with regard to their implications for understanding the effect of knowledge on concept learning.

Introduction

One of the most effective ways to improve learning in neural networks is to structure the way the training patterns are presented. Rather than including all of the patterns in each training epoch, learning is often faster and more efficient when training patterns are divided into subsets representing different parts of the overall task. In recent years, the use of structured training regimens has assumed two complementary forms. Performance on a complex task can be improved by first training separate networks to do different parts of the task, and then combining the various subnetworks to produce a structure that can do the entire task (Pratt, Mostow, & Kamm, 1991; Waibel, Sawai, & Shikano,1989). Alternatively, complex problems can be learned more quickly if a network's training set is divided into a series of increasingly difficult subtasks that are learned sequentially (Cottrell & Tsung, 1993; Elman, 1989, 1991a; Fahlman, 1991). Moreover, the effect of subtasking can even be accomplished with a constant training set, provided that the processing capacity of a network is increased during the course of learning (Elman, 1991b).

In contrast to these findings, however, there is also evidence that not all problems are learned better when they are broken down into smaller parts. Incrementally increasing the size of the training set in an eight-bit parity problem does not improve learning, and it may even make it more difficult (Harris, 1991). There is also evidence that networks can learn a given task better when they learn simultaneously several related tasks (Caruana, 1992).

Although these two sets of findings appear to be contradictory because they show that training with subtasking both facilitates and interferes with learning, Elman (1993) has argued that such effects actually illustrate some fundamental properties about how learning occurs in connectionist models. Neural networks are commonly viewed as function approximators that are trying to discover the function that underlies a given set of training patterns. Because learning algorithms are statistically driven, they are highly sensitive to statistics of the training set. Given these assumptions, Elman argued that one of the main reasons that learning is difficult is that a particular set of training patterns often has a number of different regularities, and it is not always clear which regularity a network will extract. By reducing the size of the training set, a training regimen that uses subtasking can make it easier to identify some of the regularites in the data, and so learning might be faster. However, a reduced training set can also cause problems because when statistics are computed on subsets of a total set of patterns, there is a danger that they may not provide a good estimate of the population statistics. In this case, when the size of the pattern set is increased, regularities that appeared in the smaller pattern sets may no longer apply, resulting in interference. (For related ideas, see Harris, 1991, p. 15-16; Rosenberg and Sejnowski, 1986, p. 84-85).

Although Elman (1993) was able to explain how subtasking can both facilitate and interfere with learning, he did not provide any criteria for predicting which effect will occur on a particular task. For example, it is not clear why subtasking improved performance on Elman's (1991b) task, but failed to improve learning on Harris' (1991) task. In addition, it is useful to note that the simulations (Elman, 1991b) used to develop these ideas are somewhat artificial because they cannot be related to a specific psychological task. In spite of the fact that they were intended to model the kind of learning that occurs in language acquisition, each training set contained 10,000 sentences, which far exceeds the processing capacity of an adult or a child. Moreover, the network's task was to predict the next word in a sentence, something that would not ordinarily absorb human listeners.

Given these considerations, we decided to use a simpler and more psychologically plausible task to identify some principles that can predict how different tasks are learned under different training regimens.

Analyzing Training Regimens in a Concept Learning Task

Because neural nets learn to approximate functions, we hypothesized that the opposing effects associated with subtasking can be understood in terms of function compatibility. If the function learned on an early task is compatible with the function to be learned later, the new learning will be facilitated. In contrast, if the function learned on an early task is incompatible with the function to be learned later, the new learning will be inhibited.

To test this hypothesis, we used a concept learning task originally developed by Whitman and Garner (1963). Although the experiments they carried out were motivated by a very different set of research questions, the specific stimuli that they used provided a straight-forward way of analyzing how function compatibility is manifested in differernt training regimens.

The training patterns that we used in our simulations are diagrammed in Table 1, which is adapted from a set of visual stimuli shown in Garner (1974, p. 83). Each pattern was comprised of four binary dimensions, and the total set of sixteen patterns was used to define two different classification tasks. In each task, patterns 1-8 were the positive instances of the category and patterns 9-16 were the negative instances of the category.

Although both classification tasks used each of the sixteen patterns once, they differed with regard to the statistical relations between the component dimensions. The task on the left has a simple correlational structure because the two categories of patterns can be distinguished from each other with regard to the correlation between dimensions one and four. In category S the values on these dimensions always disagree, whereas in category ~S the values on these dimensions always agree. In contrast, the task on the right has a complex correlational structure because there is no overall relationship between any of the four dimensions in either category C or category ~C.

Table 1 also illustrates that each task can be divided into two distinct subtasks, as shown by the dotted line. In the first subtask, patterns 1-4 were positive instances of the category and patterns 9-12 were negative instances of the category. In the second subtask, patterns 5-8 were positive instances of the category and patterns 13-16 were negative instances of the category. An examination of the statistical relations in each subtask suggests that the relative difficulty of learning each concept may change when the patterns are presented in terms of subtasks.

For the task with a simple correlational structure, dimensions one and four are correlated in the same way in each of the subtasks, but dimensions two and three are also correlated in each subtask, albeit in different ways. In the first subtask, dimensions two and three agree in category S and disagree in category ~S, whereas in the second subtask, dimensions two and three disagree in category S and agree in category ~S. As noted by Elman (1993), because there is more than one regularity in each subtask, it is not clear which one will be learned at any given time. Some networks might focus on dimensions two and three in the first subtask, thereby learning a function that is incompatible with the function they need to learn for the entire task. We therefore predicted that when networks are trained on the simple correlational structure in terms of subtasks, it would be harder to learn the classification relative to an unstructured training regimen.

For the task with a complex correlational structure, in spite of the fact that there are no simple correlations between any of the dimensions, within each subtask there are two different correlations that interact with each other. In category C, dimensions one and four have a correlation that is opposite in sign to that between dimensions two and three. If the values on dimensions one and four disagree, then the values on dimenions two and three agree, as in the first subtask. But, if the values on dimensions one and four agree, then the values on dimension two and three disagree, as in the second subtask. In category ~C, these relations are reversed, so that dimensions one and four have the same correlation as dimensions two and three. If the values on dimensions one and four disagree, then so do the values on dimensions two and three, as in the first subtask. If the values on dimensions one and four agree, then so do the values on dimensions two and three, as in the second subtask. Given these relationships, the function necessary for learning the first subtask in the complex concept is quite compatible with that necessary for the entire task. Namely, dimensions one and four have correlations opposite to dimensions two and three in one category and identical correlations in the other category. Because the function

Table 1: Binary coding scheme for concept learning tasks containing a simple correlational structure (left) and a complex correlational structure (right).

	S				~S				
p1	1	0	0	0	1	0	1	1	p9
p2	1	1	1	0	1	1	0	1	p10
p3	0	0	0	1	0	0	1	0	p11
p4	0	1	1	1	0	1	0	0	p12
p5	1	0	1	0	1	0	0	1	p13
p6	1	1	0	0	1	1	1	1	p14
p7	0	0	1	1	0	0	0	0	p15
p8	0	1	0	1	0	1	1	0	p16

	C				~C				
p1	1	0	0	0	1	0	1	0	p9
p2	1	1	1	0	1	1	0	0	p10
p3	0	0	0	1	0	0	1	1	p11
p4	0	1	1	1	0	1	0	1	p12
p5	1	0	1	1	1	0	0	1	p13
p6	1	1	0	1	1	1	1	1	p14
p7	0	0	1	0	0	0	0	0	p15
p8	0	1	0	0	0	1	1	0	p16

learned in the first subtask is likely to be compatible with that required for the rest of the task, we predicted that subtasking would be superior to an unstructured training regimen on the complex task.

In summary, these considerations led us to the following prediction: When subtasking exploits function compatibility it will be superior to unstructured training, but, when subtasking promotes function incompatibility it will be inferior to unstructured training.

Simulations

To test these predictions, we used Fahlman and Lebiere's (1990) cascade-correlation learning algorithm (CC) because it has certain design features that are particularly relevant for understanding facilitation and interference effects that occur during learning. CC is a generative algorithm that starts out with a minimal typology, consisting of an input layer that is fully connected to an output layer. To solve a problem, it first tries to reduce the error between the observed and desired activation across the output units by modifying the weights between the input and output units. If it fails to reduce this error within an acceptable criterion, it then recruits a hidden unit from a pool of candidates that are connected only to the input units. The weights from the input units to the candidate hidden units are then trained so as to maximize the correlation between each candidate hidden unit's activation and the residual error at the output units. When these correlations reach asymptote, the input weights leading to best candidate hidden unit are frozen, this hidden unit is connected to the output units, and the network reverts back to error minimization by modifying the weights connected to the output units. The process of recruiting additional hidden units is then repeated as needed. (For more detailed discussions about the CC architecture, see Fahlman & Lebiere, 1990; Shultz, Schmidt, Buckingham, & Mareschal, 1995).

Because the weights from the input units to each hidden unit are fixed when each hidden unit is added to the network, knowledge acquired during the course of learning is more likely to be preserved during subsequent learning, and so its effects will be more salient than with other algorithms. This design feature makes CC more resistant to retroactive interference than backpropogation (Tetewsky, Shultz, & Buckingham, 1995).

The tasks were presented in the same way that they appear in Table 1, with two exceptions. (1) The dichotomous coding in the input patterns was represented in terms of 1 and -1, rather than 1 and 0, to speed up learning.[1] (2) When subtasking was used, after a network had been trained sequentially on subtask 1 and subtask 2, it was then trained on the total set of patterns associated with the classification (i.e., subtask 1 + subtask 2). This third phase of training provided a way to determine the extent to which the function that had been approximated from the subtasks was compatible with the function in the overall task. After a network learned the second subtask it would have learned all 16 training patterns. However, if the function the network learned to approximate was different from the function contained in the overall task, then the network would require additional training in this third phase of learning. Furthermore, if the number of epochs needed in the third phase of training was less than the number of epochs needed to learn the task when the patterns were presented all-at-once, there would be evidence for function compatibility; if the number of epochs needed in the third phase of training was greater than the number of epochs needed during all-at-once presentation, there would be evidence for function incompatibility.

The simulations were carried out as a 2 x 2 factorial design, in which there were two types of conceptual structures (simple and complex) crossed with two types of training regimens (all-at-once presentation and subtasking). The primary dependent variable was the number of training epochs needed to learn a particular concept. Training was stopped when all of the output values for the patterns in a given training set fell within 40% of their desired values (i.e., the value of the score-threshold parameter was 0.4). Fifty networks were run in each of the four conditions of the design and the results were averaged across networks.

Results

The total number of training epochs required for each of the four conditions specified in the 2 x 2 design are shown in Figure 1. Note that in assessing overall performance, numbers of training epochs required in each phase of subtasking were summed to get the total epochs for learning the entire task.

Although there was no overall difference between the two training regimens (128 vs. 133 epochs), F (1, 196) = .955, p = .33, there was a main effect of task structure (191 vs 70), F (1, 196) = 668, $p < .001$, such that the complex structure was harder to learn than the simple structure. Of more importance, however, is the fact that the interaction between task structure and training regimen was highly significant, F (1,196) = 44.9, $p < .001$. Paired contrasts on the means confirmed that it was easier to learn the complex structure under subtasking (177 vs 204 epochs), F (1,196) = 16.4, $p < .001$, and it was harder to learn the simple structure under subtasking (88 vs. 52 epochs), F (1,196) = 29.5, $p < .001$.

An examination of the frequency distributions for the total number of epochs needed in each of the four conditions provided some interesting qualifications. When the patterns were presented all-at-once during training, the distributions for the simple and complex structures tended to be normal. However, when the patterns were presented in subtasks, the simple and complex structures produced distinctly bimodal distributions. For the simple structure, 25 of the networks were in the range of 42-55 epochs, and 25 were in the range of 113-142. For the complex structure, 39 were in the range of 132-167, and 11 were in the range of 216-333. Because these subgroups fell on either side of the overall mean for the respective concept structures, for purposes of convenience they will be referred to as the easy and hard versions of subtasking for each structure.

[1] This point was suggested to us by Yasser Hashmi.

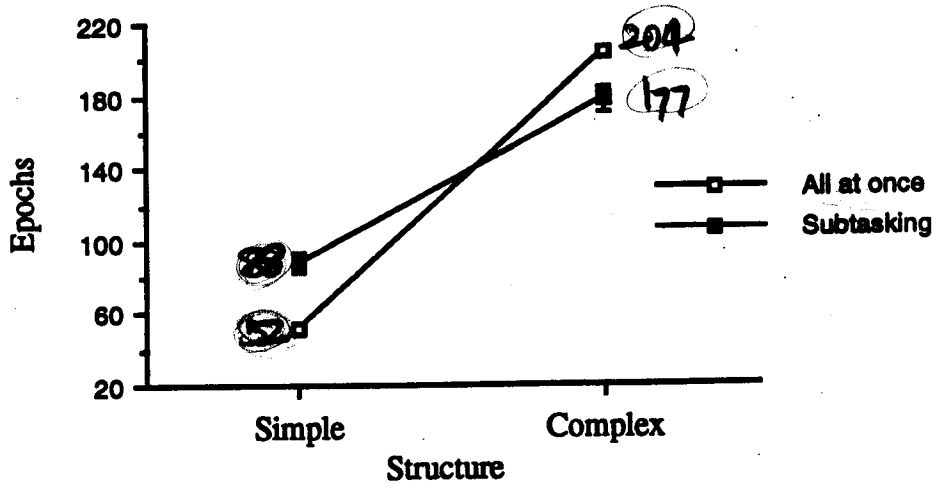

Figure 1. Mean number of epochs needed to learn two concepts under two training regimens. Error bars represent standard errors of the mean.

The two alternative forms of subtasking for a task with a simple structure are shown in Table 2, broken down according to the number of epochs needed in each of the three phases of learning. In the hard version of subtasking, subtask 1 required an average of 51 epochs to be learned, whereas subtask 2 only required 12 epochs. This difference was significant, F (1, 48) = 1727, $p < .001$, and it suggested that the knowledge the networks had acquired from subtask 1 facilitated their ability to learn subtask 2. But in spite of this facilitation, even though the networks had learned to classify all 16 patterns correctly after completing the first two phases of learning, they apparently did not possess the knowledge embodied in the overall task, because the number of training epochs required in phase 3 was clearly different from zero. In fact, phase 3 training required more epochs than were needed to learn the simple task during all-at-once presention (63 vs. 52, t (73) = 11.3, $p < .001$) providing evidence for function incompatibility. Thus, as a result of learning sequentially subtasks 1 and 2, networks experienced proactive interference in learning the entire set of patterns. In contrast to these findings, however, for the easy version of subtasking, it appears that these networks learned to approximate the correct function in subtask 1, so that there was perfect transfer across the next two phases of learning (i.e., no training epochs were required to learn either subtask 2 or the entire set of patterns.)

Table 2. Mean number of epochs needed in the easy and hard versions of subtasking on a simple structure. Numbers in parentheses are standard errors of the mean.

	Easy (n=25)	Hard (n=25)
Subtask 1	50 (0.7)	51 (0.6)
Subtask 2	0 (0.0)	12 (0.6)
Subtasks 1&2	0 (0.0)	63 (0.9)
	50	126

The two alternative forms of subtasking for a task with a complex structure are shown in Table 3. In the easy version of subtasking, subtask 1 required 50 epochs to be learned whereas subtask 2 only required 11. Once again, this difference was significant, F (1, 76) = 1634, $p < .001$, and it suggested that the knowledge the networks had acquired from subtask 1 facilitated their ability to learn subtask 2. But, in contrast to the hard version of the simple structure, the number of epochs required in phase 3 training was less than the number of epochs required to learn the complex structure when it was presented all-at-once (92 vs. 204, t (87) = 32.5, $p < .001$), providing evidence for function compatibility. Thus, the knowledge that developed as a result of learning sequentially subtasks 1 and 2, facilitated learning the entire set of patterns. However, in the hard version of subtasking, in spite of the fact that subtasks 1 and 2 followed the same trend as in the easy version, the number of epochs needed in phase 3 training did not differ from the number of epochs required to learn the complex structure when it was presented all-at-once (202 vs 204, t (59) = .268, $p > .05$.). This result therefore implied that these networks had essentially learned the same function that occurs when the training set is unstructured. This particular finding is noteworthy because given the nature of the training patterns, there is no a priori reason to expect that subtasking on the complex structure would produce two different kinds of solutions. One possible way to interpret this result is by examining the number of hidden units that were recruited by the two kinds of networks. In the easy version of subtasking, networks recruited one hidden unit in both phase 1 and phase 3, whereas in the hard version, networks recruited one hidden unit in phase 1 and at least two hidden units in phase 3. Because most of the networks that learned the complex task under an unstructured training regimen required two hidden units, there is reason to believe that in the hard version of subtasking, networks were somehow ignoring the information from the hidden unit recruited in phase 1, and learning the task as if the training set was unstructured. However, this conclusion, as well as the other inferences that we made about the relative difficulty of learning the different correlational structures under the different training regimens, can only be confirmed by more detailed analyses of network knowledge representations and the corresponding functions the networks learned to approximate.

Table 3. Number of epochs needed in the easy and hard versions of subtasking on a complex structure. Numbers in parentheses are standard errors of the mean.

	Easy (n=39)	Hard (n=11)
Subtask 1	50 (0.6)	51 (1.3)
Subtask 2	11 (0.4)	9 (0.5)
Subtasks 1&2	92 (1.1)	202 (8.9)

153 262 177

Discussion

These simulations were carried out to determine the conditions under which subtasking will either improve or impair learning in neural networks, relative to unstructured, all-at-once training regimens. The results indicated that when a network extracts a function that is compatible with later learning, subtasking will facilitate learning, whereas when a network extracts a function that is incompatible with later learning, subtasking will interfere with learning.

In evaluating these results it is important to note that in general, there is no a priori reason to expect that certain kinds of tasks will always be learned better under subtasking, whereas other kinds of tasks will always be learned better under unstructured training regimens. Depending on the way the subtasks are formed, it is possible that subtasking could improve performance on the simple correlational structure and impair performance on the complex structure. As a practical matter, it is therefore necessary to devise a training regimen that is appropriate for a given task. In terms of the present findings, it would therefore be useful to form other types of subtasks for each structure and examine the extent to which the general notion of function compatibility can account for the resulting outcomes.

Finally, these simulations have some important implications with regard to understanding the effects of knowledge on learning. Previous psychological research in this area has primarily been concerned with showing how existing knowledge structures can reverse the difficulty of learning various formal category structures (Pazzani, 1991; Wattenmaker, Dewey, Murphy, & Medin, 1986; Waldmann & Holyoak, 1989.) Although these findings are impressive, they are also somewhat limited because they do not show how the critical knowledge that produced the reversal was itself acquired. The present experiment is therefore noteworthy because it shows how knowledge, which is acquired during the course of learning, can affect the acquisition of new knowledge. Hence, these results can be viewed as a way of bridging the gap between formal models of categorization, which try to describe specific learning algorithms, and knowledge-based approaches, which are concerned with how concept learning is influenced by existing knowledge. Most connectionist models of human learning have learned from initially random weights. Although such results may provide important existence proofs, they are generally unrealistic as models for human learning, which ordinarily occurs from a base of initial knowledge. Research on subtasking illustrates the potential importance of prior knowledge in learning.

Acknowledgements

Thia research was supported by an NSERC Grant awarded to Thomas R. Shultz. The authors would like to thank Yuriko Oshima-Takane, David Buckingham and Yasser Hashmi for their comments and criticisms.

References

Caruana, R. (1992). Multitask learning: A knowledge-based source of inductive bias. *Proceedings of the Tenth International Machine Learning Conference*, 41-48. San Mateo, CA: Morgan Kaufman.

Cottrell, G.W. & Tsung, F. (1993). Learning simple arithmetic procedures. *Connection Science, 5*, 37-58.

Elman, J. (1989). Representation and structure in connectionist models. *Center for Research in Language Technical Report 8903*, UCSD, LaJolla, CA.

Elman, J. (1991a). Incremental learning, or the importance of starting small. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, (pp. 443-448). Hillsdale, NJ: Lawrence Erlbaum Associates.

Elman, J. (1991b). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7, 195-225.

Elman, J. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, 48, 71-99.

Fahlman, S. E. (1991). The recurrent cascade-correlation architecture. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), Advances in Neural Information Processing Systems 3, 190-196. Los Altos, CA: Morgan Kaufmann Publishers.

Fahlman, S. E. & Lebiere, C. (1990) The cascade-correlation learning architecture. *Technical Report, CMU-CS-90-100*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Garner, W. R. (1974). *The processing of information and structure*. Potomac, MD: Lawrence Erlbaum Associates.

Harris, C. L. (1991). Parallel distributed processing models and metaphors for language and development. Unpublished doctoral dissertation, UCSD, La Jolla, CA.

Pazzani, M. J. (1991). The influence of prior knowledge on concept acquisition: Experimental and computational results. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17, 416-432.

Pratt, L. Y., Mostow, J., & Kamm, C. A. (1991). Direct transfer of learned information among neural networks. *Proceedings of the Ninth National Conference on Artificial Intelligence*, (pp. 584-589). Anaheim, CA.

Rosenberg, C. R. & Sejnowski, T. J. (1986). The spacing effect on NETtalk, a massively parallel network. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, (pp. 72-89). Hillsdale, NJ: Lawrence Erlbaum Associates.

Shultz, T. R., Schmidt, W.C., Buckingham, D., & Mareschal, D. (1995). Modeling cognitive development with a generative connectionist algorithm. In G. Halford & T. Simon (Eds.), Developing cognitive competence: New approaches to process modeling. Hillsdale, NJ: Erlbaum.

Tetewsky, S., Shultz, T. & Buckingham, D. (1995). Interference and savings in connectionist models of a sequential recognition memory task. Manuscript submitted for publication.

Waldmann, M. R. & Holyoak, K. J. (1990). Can causal induction be reduced to associative learning? *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, (pp. 190-197). Hillsdale, NJ: Lawrence Erlbaum Associates.

Waibel, A., Sawai, H, & Shikano, K. (1989). Modularity and scaling in large phonemic neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12), 1888-1898.

Wattenmaker, W. D., Dewey, G. I., Murphy, T. D., & Medin, D. L. (1986). Linear separability and concept learning: context, relational properties, and concept naturalness. *Cognitive Psychology*, 18, 158-194.

Whitman, J. R., & Garner, W. R. (1963). Concept learning as a function of form of internal structure. *Journal of Verbal Learning and Verbal Behavior*, 2, 195-202.