Computational Power and Realistic Cognitive Development

David Buckingham and **Thomas R. Shultz**
Department of Psychology
McGill University
1205 Penfield Avenue
Montréal, Québec, Canada H3A 1B1
dave@ego.psych.mcgill.ca
shultz@psych.mcgill.ca

Abstract

We explore the ability of a static connectionist algorithm to model children's acquisition of velocity, time, and distance concepts under architectures of different levels of computational power. Diagnosis of rules learned by networks indicated that static networks were either too powerful or too weak to capture the developmental course of children's concepts. Networks with too much power missed intermediate stages; those with too little power failed to reach terminal stages. These results were robust under a variety of learning parameter values. We argue that a generative connectionist algorithm provides a better model of development of these concepts by gradually increasing representational power.

Introduction

The use of connectionist networks to model cognitive development has placed new emphasis on a fundamental question in cognitive development: How is transition from one stage to another possible (Bates & Elman, 1993)? Although many researchers (e.g., Plunkett & Sinha, 1992; McClelland, 1995) conclude that connection weight adjustment can account for transition, the recent success of models employing a generative algorithm questions this conclusion (Shultz, Schmidt, Buckingham, & Mareschal, 1995). Shultz et al. (1995) argue that, in addition to weight adjustment, transition requires increases in non-linear computational power afforded by the recruitment of hidden units into the network as it learns. To assess the importance of hidden unit recruitment, in this article we explore the ability of a static connectionist algorithm to model children's acquisition of velocity (v), time (t), and distance (d) concepts and compare it to research using a generative connectionist algorithm (Buckingham & Shultz, 1994).

Development of Velocity, Time, and Distance

In classical physics, velocity is defined as $v = d \div t$, time as $t = d \div v$, and distance as $d = v * t$. Wilkening (1981, 1982) designed tasks in which children were asked to infer velocity, time, or distance given information about the other two dimensions. Wilkening found the following regularities: (1) In a distance-inference task, 5-year-olds employed an additive rule, $d = t + v$, whereas adults used the correct multiplication rule, $d = v * t$; (2) in a time-inference task, 10-year-olds and adults employed the correct division rule, $t = d \div v$, whereas 5-year-olds used a subtraction rule, $t = d - v$; and (3) in a velocity-inference task, 10-year-olds and adults used a subtraction rule, $v = d - t$, whereas 5-year-olds used an identity rule, $v = d$.

Simulations Using a Generative Algorithm

Buckingham and Shultz (1994) modeled the acquisition of velocity, time, and distance concepts using cascade-correlation (Fahlman & Lebiere, 1990), a generative connectionist algorithm. Cascade-correlation networks begin with a minimal topology determined by the number of input and output units, without any hidden units. During an output training phase, weights from input units and any installed hidden units are adjusted to minimize the sum of squared error between actual and target outputs. When error can no longer be minimized, an input training phase begins in which weights from input units to a pool of candidate hidden units are adjusted to maximize the correlation between hidden unit activation and output error. The hidden unit that attains the highest correlation is then installed into the network and output training recommences.

Simulation results matched those of Wilkening (1981; 1982) for the most part. For distance inferences, there was a progression from the additive ($d = t + v$) to the multiplicative rule ($d = t * v$). For time and velocity inferences, networks began with identity rules ($t = d$ and $v = d$, respectively), progressed to additive ($t = d - v$ and $v = d - t$, respectively), and finally multiplicative rules ($t = d \div v$ and $v = d \div t$, respectively).[1] Wilkening's participants did the same, except that they showed no identity rule for time inferences and failed to use the multiplicative rule for velocity inferences (Wilkening, 1981, attributed this latter failure to task demands).

Buckingham and Shultz (1994) suggested that the transition from identity stages through intermediate additive stages and finally multiplicative stages was made possible by both weight adjustment and hidden unit recruitment. In order to test this hypothesis, we compare the performance of cascade-correlation networks with that of static networks (i.e., networks in which the architecture is fixed throughout training).

[1] Example results are presented in Figure 1a for comparison purposes.

Simulations Using a Static Algorithm

Experiment 1

We used standard back-propagation networks as static networks because these were used by McClelland (1989) in his pioneering work modeling cognitive development on the balance scale task and are the most common connectionist learning networks. To maximize the chances of capturing human performance, we systematically sampled a variety of back-propagation architectures and parameter values. We ran simulations using four differentially powerful architectures: one hidden layer with one, two, or three hidden units; and two hidden layers with two hidden units in each layer. In each architectural condition, 180 networks were run in a crossed experimental design consisting of three levels of learning rate (*eta*) and momentum (*alpha*). The levels of learning rate were 0.025, 0.050 (the default value), and 0.100. The levels of momentum were 0.100, 0.450, and 0.900 (the default value).

The task was the same as in our cascade-correlation simulations. The networks had to predict, as output, the value of one dimension (e.g., velocity) given information about the other dimensions (e.g., distance and time). In order to maximize the ability to compare the performance of static networks with that of generative networks, input and output coding, output unit type, weight updating mode, and training and testing methods were as they had been in the cascade-correlation simulations (Buckingham & Shultz, 1994).

Inference patterns were encoded using *nth* encoding as follows.[2] Two input banks received dimensional values ranging from 1 to 5. The third bank received an input of 0 indicating that it was the dimension to be predicted. Each input bank had five input units for a total of 15 input units. A dimensional value *n* was encoded by assigning an activation of 1 to the *nth* input unit of the bank and 0 to all other units in the bank. Thus, for a given inference pattern, one input bank received activations of 0 on all of its five input units, indicating it was unknown. One unit of each of the other two input banks received an activation of 1. The remaining units in these banks received activations of 0.

As in our work with cascade-correlation, one linear output unit was used. A linear output was used because it is the most natural way of producing a quantitative output similar to the responses made by Wilkening's participants. Target values for the output unit were calculated using the three Newtonian equations ($v = d \div t$, $t = d \div v$, and $d = v * t$), respectively. In addition, distance target values were divided by five so that their range was identical to the ranges of time and velocity target values. Twenty-five instances of each of the three inference problem types were obtained by crossing the five levels of velocity, time, and distance for a total of 75 inference patterns.

At each epoch of training, all 75 inference problems were presented to the network. Weight updates occurred only after all patterns had been presented to the network. This batch training continued for a maximum of 1500 epochs.

To compare network results with human performance, every fifth epoch of training we diagnosed rules that best captured network performance on each problem type. We computed correlations between the network's responses and those predicted by various plausible rules such as identity ($v = d$, or $v = t$), addition ($v = d+t$, or v = d-t), or multiplication (v = d*t, $v = t \div d$, or $v = d \div t$) rules. To be diagnosed as exhibiting stage performance, a rule had to correlate positively with network responses, account for more than 50% of the variance in network responses, and account for more variance than other plausible rules across four consecutive sampled epochs.

Results. A plot of the rules diagnosed as training progressed is shown in Figure 1 (b-e) for one network in each of the architectural conditions. These nets were chosen because they were good exemplars of typical performance across learning rates and momentum values.

For networks with a single hidden unit (Figure 1b), the typical progression involved early onset of time and velocity identity stages, followed by onset of the distance additive stage and, then, oscillation between the additive and multiplicative distance rules. Only 19 of the 180 networks attained a stable multiplicative stage of distance ($d = t*v$). None of the networks attained the multiplicative stages of time and velocity (only four networks progressed beyond the identity stages to the additive stages of time, $t = d$-v, and velocity, $v = d$-t).

In contrast to networks with a single hidden unit, the majority of networks with two hidden units (Figure 1c) progressed beyond the identity stages of time and velocity, attaining the multiplicative stages. However, only 13 of the 180 networks demonstrated the intermediate additive stages of both time and velocity. With respect to distance development, a small majority (94) demonstrated the distance additive stage and, unlike networks with a single hidden unit, a large majority (166) of networks attained a stable distance multiplicative stage.

Performance of networks with three hidden units (Figure 1d) was similar to those with two hidden units although slightly fewer networks demonstrated both time and velocity identity stages (170 vs. 177) and additive stages (4 vs. 13). All 180 networks attained the multiplicative stages of time and velocity. Another difference was that fewer networks (69 vs. 94) demonstrated the distance additive stage. All but one of the 180 networks attained the distance multiplicative stage.

Finally, the majority of networks with two hidden layers (Figure 1e) also failed to demonstrate the time and velocity additive stages. Only six networks attained both intermediate additive stages of time and velocity. The majority of networks (172) attained the multiplicative stages of time and velocity, respectively. Use of a second hidden unit layer increased the number of networks demonstrating the distance additive stage but only slightly (106 vs. 94 networks with one hidden layer of two units). All but six of the 180 networks attained the distance multiplicative stage.

[2] In Buckingham (1993), cascade-correlation networks with nth encoding demonstrated the same qualitative stage progression as those with more distributed input encodings. However, networks with nth encoding had a decided advantage in that their solutions were more transparent.

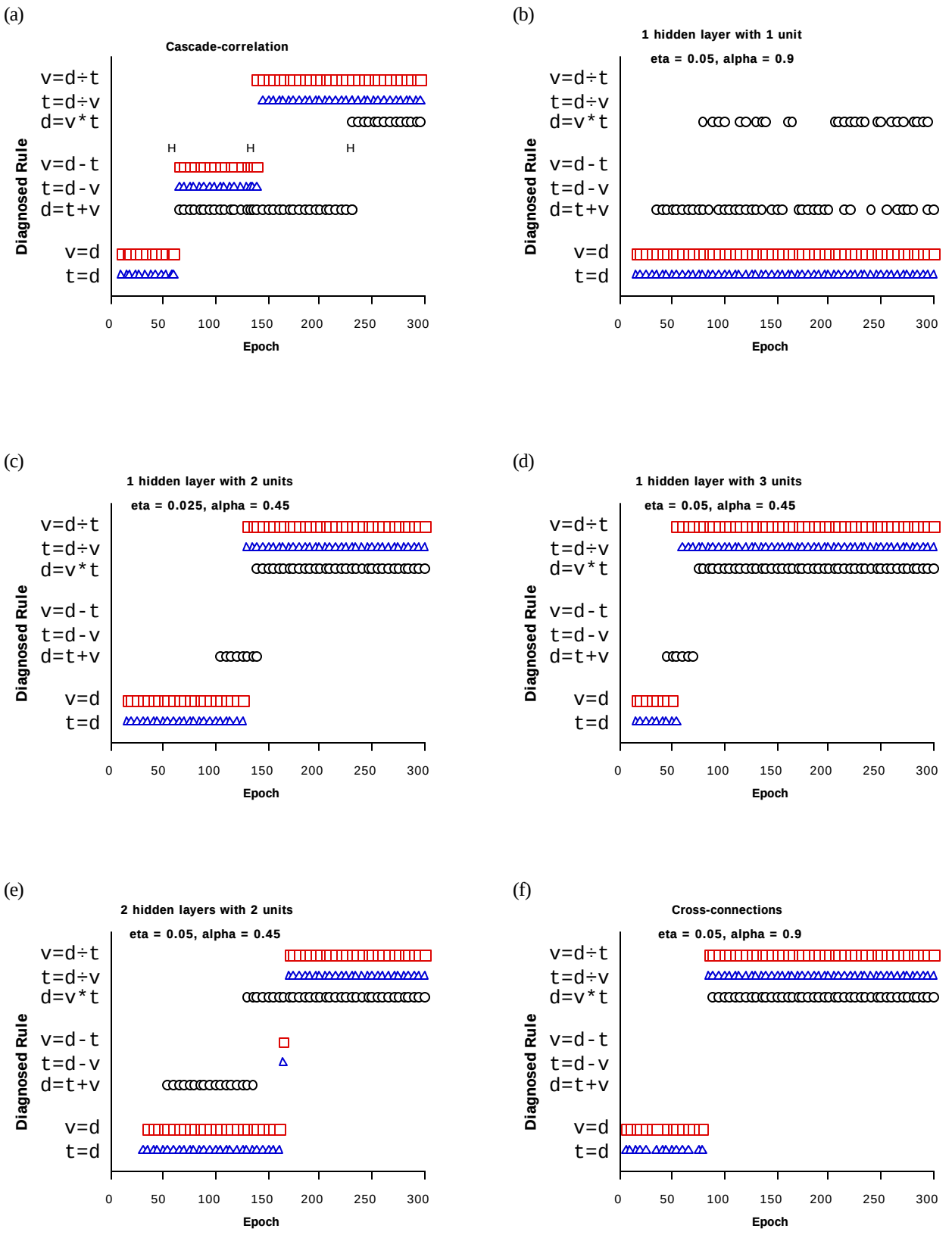

Figure 1: Diagnosed rules of (a) a generative connectionist network from previous research (H indicates hidden unit recruitment); one network in experiment 1 that exemplifies typical performance with one hidden layer of (b) one, (c) two, and (d) three hidden units, and (e) two hidden layers with 2 hidden units in each layer; (f) one network in experiment 2 with one hidden layer containing two hidden units and cross-connections.

In summary, very few static networks demonstrated the entire developmental course: time and velocity identity stages; distance, time, and velocity additive stages; and distance, time, and velocity multiplicative stages. Of the networks with a single hidden unit layer, only five out of 180 networks with two hidden units and three out of 180 networks with three hidden units demonstrated the entire developmental course. None of the networks with a single hidden unit attained the multiplicative stages of time and velocity. Networks with two or three hidden units on one layer typically missed the intermediate additive stages, particularly for time and velocity inferences. Finally, only one of the 180 networks with two hidden layers demonstrated the entire developmental course; these networks also missed the intermediate additive stages for time and velocity inferences.

Experiment 2

Cascade-correlation differs from back-propagation not only in the progressive recruitment of hidden units, but also in the use of cross-connections that bypass hidden unit layers. To assess the possibility that the psychological realism of cascade-correlation simulations might be due to the use of these cross-connections, and not to generative hidden unit creation, we ran 20 static networks with an architecture consisting of cross-connections and one hidden layer with two units. One hidden layer with two units was chosen because it showed the most promise of capturing time and velocity additive stages in Experiment 1. In experiment 2, we used only the default learning rate (0.050) and momentum values (0.900). Everything else was kept constant with Experiment 1.

Results. A plot of the rules diagnosed in one network as training progressed is shown in Figure 1f. This network was chosen because it was a good exemplar of typical performance. Overall the performance of these networks was similar to those in Experiment 1. That is, the majority of networks (14/20) progressed from the identity stages of time and velocity to the multiplicative stages without demonstrating the intermediate additive stages. Of the remaining six networks, three did not exhibit either identity stage and three attained one identity stage but not the other. All 20 networks attained the multiplicative stages of time and velocity. One difference compared to networks in Experiment 1 was that the use of cross-connections resulted in even fewer networks (5/20) first achieving the distance additive stage before the multiplicative stage. All 20 networks attained the distance multiplicative stage.

Discussion

Static networks in both experiments had no difficulty capturing early time and velocity identity stages. The limitation of static networks was their inability to capture both additive and multiplicative stages, regardless of a wide sampling of network architecture and parameter values. Different network architectures could capture one type of stage, but not the other, e.g., additive but not multiplicative, or multiplicative but not additive. Thus, simple connection weight adjustment is insufficient to capture all stage transitions.

The most general failure of static networks with more than one hidden unit was to miss intermediate additive stages. Although there remains some doubt as to the inter-developmental course of additive stages and whether or not the additive stage of velocity is the terminal stage of velocity development, children clearly pass through these additive stages (Wilkening, 1981; 1982). Static networks with only one hidden layer consisting of one hidden unit often captured additive stages, but failed to reach multiplicative stages. Static networks with the limited computational power provided by one hidden layer with one hidden unit seemed too weak to attain multiplicative stages; static networks with more computational power seemed too powerful because they skipped intermediate stages. There seemed to be no static back-propagation architecture capable of simulating the full range of stages in the domain of velocity-time-distance. In contrast, all generative networks captured identity, additive, and multiplicative stages (Buckingham & Shultz, 1994).

The failure of static networks with cross-connections to successfully capture human performance in Experiment 2 suggests that the use of these cross-connections by cascade-correlation is not sufficient for its success. Rather, progressive recruitment of hidden units appears necessary for capturing correct stage progressions. Cross-connections may prove to be necessary as well, particularly in capturing early linearly separable performance, but this would need to be documented in future simulations.

Other, less direct evidence for the superiority of generative over static connectionist algorithms at simulating human development has been reported. For example, generative networks (Shultz, Mareschal, & Schmidt, 1994) captured the terminal stage of balance scale development more successfully than did static networks (McClelland, 1989; 1995). The present results extend these findings to cases in which static networks, with a sufficiently powerful architecture, successfully capture terminal stages (multiplicative stages) but fail to capture intermediate stages. Simulating the full range of psychologically realistic stages appears to rely on the ability of networks to grow in computational power. A similar point in the realm of grammar learning was made by Elman (1993). To learn an English-like grammar, recursive back-propagation networks had to receive either progressively more complex sentences or grow in working memory capacity.

The fact that realistic connectionist models of development need to grow in computational power suggests that human development involves not only incremental learning but also increases in non-linear representational abilities. What factors cause the emergence of these new representational abilities in children remains an open question.

This research compares only a single exemplar of a static algorithm (back-propagation) to a single exemplar of a generative algorithm (cascade-correlation). Using other exemplars of each class of algorithm could indicate the generality of the conclusions. It might also be interesting to explore the capacity of other generative network techniques to capture cognitive developmental phenomena. For example, must the network grow vertically, as in cascade-correlation, or could it grow horizontally on a single layer (e.g., Ash, 1989)? If cognitive development is characterized by the continual redescription of earlier knowledge representations (Karmiloff-Smith, 1992), then vertical, rather than horizontal, growth would seem to be required. Further, how would network pruning techniques (Hanson & Pratt, 1989; Le Cun, Denker, & Solla, 1990) fare in capturing developmental stages? If cognitive development is characterized by the emergence of qualitatively distinct knowledge representations (Carey, 1991), then recruitment ought to work better than pruning.

Acknowledgments

This research was supported in part by a fellowship from the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Québec and an operating grant from the Natural Sciences and Engineering Research Council of Canada.

References

Ash, T. (1989). Dynamic node creation in backpropagation. *Connection Science*, 1, 365-375.

Bates, E. A., & Elman, J. L. (1993). Connectionism and the study of change. In M. H. Johnson (Ed.), *Brain development and cognition* (pp. 623-642). Oxford: Blackwell.

Buckingham, D. (1993). *The developmental course of distance, time, and velocity concepts: A generative connectionist model*. Unpublished Master's thesis, McGill University, Montréal, Québec, Canada.

Buckingham, D., & Shultz, T. R. (1994). A connectionist model of the development of velocity, time, and distance concepts. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 72-77). Hillsdale, NJ: Erlbaum.

Carey, S. (1991). Knowledge acquisition: Enrichment or conceptual change. In S. Carey & R. Gelman (Eds.), *The epigenesis of mind: Essays on biology and cognition* (pp. 257-291). Hillsdale, NJ: Erlbaum.

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

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2* (pp. 524-532). Los Altos, CA: Morgan Kaufmann.

Hanson, S. J., & Pratt, L. Y. (1989). Comparing biases for minimal network construction with back-propagation. In D. S. Touretzky (Ed.), *Advances in neural information processing systems* (pp. 177-185). Los Altos, CA: Morgan Kaufmann.

Karmiloff-Smith, A. (1992). *Beyond modularity*. Cambridge, MA: MIT Press.

Le Cun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal Brain damage. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2*, (pp. 598-605). Los Altos, CA: Morgan Kaufmann.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In R. G. M. Morris (Ed.), *Parallel distributed processing: Implications for psychology and neurobiology* (pp. 8-45). Oxford: Oxford University Press.

McClelland, J. L. (1995). A connectionist perspective on knowledge and development. In T. J. Simon, & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 157-204). Hillsdale, NJ: Erlbaum.

Plunkett, K., & Sinha, C. (1992). Connectionism and developmental theory. *British Journal of Developmental Psychology*, *10*, 209-254.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning*, *16*, 57-86.

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

Wilkening, F. (1981). Integrating velocity, time, and distance information: A developmental study. *Cognitive Psychology*, *13*, 231-247.

Wilkening, F. (1982). Children's knowledge about time, distance, and velocity interrelations. In W. J. Friedman (Ed.), *The developmental psychology of time* (pp. 87-112). NY: Academic Press.