21

Models of Cognitive Development

THOMAS R. SHULTZ*

One of the important unsolved problems in cognitive development is the precise specification of developmental transition mechanisms. Transition is one of the two major issues in cognitive development, the other being the issue of structure. The question of structure focuses on what develops at particular ages and stages. The question of transition focuses on how these structures develop. How does the child progress from one stage or level to the next? It has been estimated that 95 % of literature on cognitive development deals with issues of structure and diagnosis of structure (Sternberg, 1984). It is likely that transition issues are ignored largely because they are so difficult.

Past theories of developmental transitions, such as Piaget's (1972) assimilation-accommodation model of adaptation, although interesting, have been too vague to be of much use (Boden, 1982). I believe that computational modeling can provide insights into the problem of transition mechanisms. Connectionist modeling techniques, based loosely on principles of neuronal computation, appear to be especially promising in this respect.

In this paper, I argue that generative algorithms provide the right sort of connectionist model. I focus on the cascade-correlation algorithm which my colleagues and I have applied to several diverse areas of cognitive development, including the balance scale, seriation, and the integration of velocity, time, and distance cues.

* Laboratoire de psychologie cognitive, McGill university, Montréal, Québec.

I. CONNECTIONISM

Connectionist networks are composed of many inter-connected units. Each unit's activity level is modulated by the weighted sum of inputs from other units, passed through a non-linear squashing function. Each unit runs a simple program in which the weighted sum of inputs from other units is first computed. Then the unit outputs a number reflecting its activity level, which is a non-linear function of this weighted sum. Passing the weighted sum through a non-linear squashing function ensures that units tend to be either off or on, input has to reach a learnable threshold in order for the receiving unit to become active, and amount of activity in a unit is limited. Once the unit's activity level has been computed, the activity is then sent on to other units running this same simple program.

Connection weights among the units are adjusted by learning so that the network can learn to map inputs to appropriate outputs without error. Consistent with the analogy to brain-style computation, units correspond roughly to neurons, unit activity levels to firing rates, and connection weights to synapses.

There has been considerable skepticism about the appropriateness of connectionist models of higher level cognition and cognitive development. But there is a growing list of connectionist models in the areas of cognitive (McClelland, 1989), perceptual (Harnad, Hanson, et Lubin, 1994), and language development (Chauvin, 1989 ; Elman, 1993 ; MacWhinney, Leinbach, Taraban, et McDonald, 1989 ; Plunkett et Marchman, 1991 ; Schyns, 1991 ; Seidenberg et McClelland, 1989). Each of these models provides interesting explanatory insights. It is also noteworthy that each of these models uses static, pre-designed networks.

I believe that generative connectionist algorithms are capable of providing better models of a wide variety of cognitive developmental problems. Generative algorithms start with a minimal network structure and construct the rest of the network as learning progresses (Fahlman et Lebiere, 1990 ; Frean, 1990 ; Marchand, Golea, et Ruján, 1990 ; Mézard et Nadal, 1989). This affords a more principled approach to network construction than is typical in connectionist research and allows for growth in computational power as well as learning.

II. CASCADE-CORRELATION

With a number of colleagues, I have applied a specific generative connectionist algorithm, cascade-correlation (Fahlman et Lebiere, 1990), to modeling transitions in cognitive development. Cascade-correlation utilizes a particularly sensible way of building up network structure. Like other generative algorithms, cascade-correlation builds its own network topology by recruiting new hidden units as it needs them to solve a problem. It starts with a minimal network of input units and output units. During training, the algorithm may add hidden units one-by-one, installing each new hidden unit on a new layer. From a developmental point of view, the importance of generative connectionist algorithms like cascade-correlation is that they are able to simulate underlying developmental changes that are either qualitative or quantitative.

There are two alternating, recurrent phases in the cascade-correlation algorithm: an output phase in which connection weights entering output units are adjusted in order to reduce the network's error, and an input phase in which new hidden units are selected and installed in the network. During the output phase, connection weights are adjusted according to a gradient descent procedure known as Quickprop (Fahlman, 1988). The focus is on output side weights, *i.e.*, those connection weights going into output units. Quickprop modifies each connection weight to minimize the error at the network's output units. Error is computed as the sum of squared discrepancies between the output activations the net should be producing and those it is actually producing. Both first and second derivative information from the error function are used to compute connection weight changes. Weight changes are proportional to the slope and inversely proportional to the estimated curvature of the error function. This allows connection weight changes to be decisive and effective.

When error is no longer decreasing or the algorithm looses patience at not having solved the problem in some specified number of passes through the training examples, there is a shift to the input phase. In the input phase, a pool of candidate hidden units receives trainable input from the input units and any existing hidden units. Outputs from candidate hiddens are not yet connected to the output units. The purpose of the input phase is to recruit a hidden unit whose activations correlate highly with errors at the output units. Connection weights into the candidate units are adjusted using Quickprop to maximize the correlations between activations on the candidate units and the network's current error. When the correlations are no longer increasing or a set number of passes through the training examples has occurred, the candidate hidden unit whose activations come to correlate best with the network's current error is selected for installation. Selected hidden units are installed into the network in a cascade, such that each new hidden unit receives input from the input units and from any previous hidden units. After installation of a new hidden unit, the algorithm reverts back to the output phase. These recurring phases of network growth are portrayed in figure 21.1 for a generic cascade-correlation network with two input units and a single output unit.

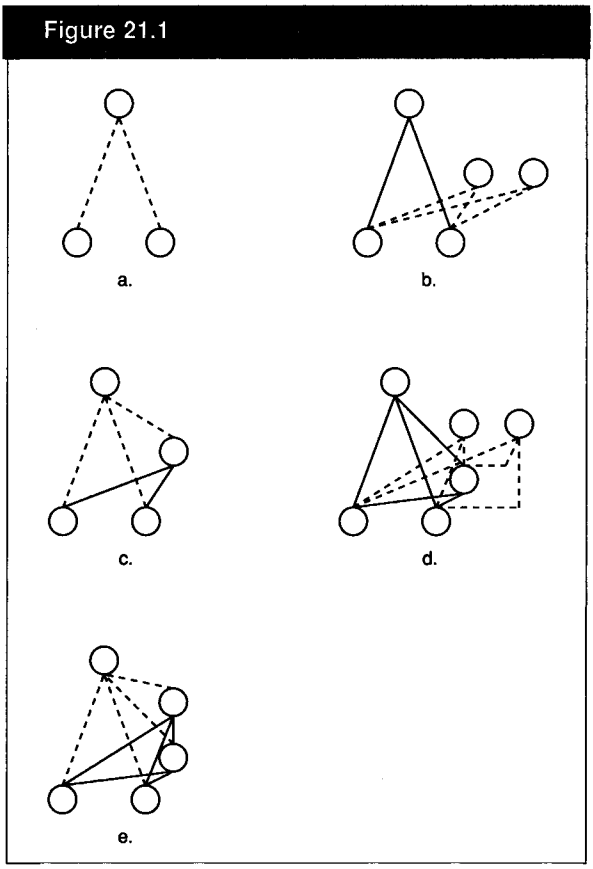
Effectively, cascade-correlation is systematically searching not only weight space, but also the space of network topologies. The algorithm efficiently finds a network topology and a set of connection weights to solve the problem it is being trained on.

In this paper, I summarize cascade-correlation simulations of three basic phenomena in cognitive development: the balance scale, seriation, and the integration of velocity, time, and distance cues. Each is a classic developmental task from Piaget's work that has received considerable attention in the contemporary literature. Moreover, each of the three domains involves sequential stages of acquisition.

III. THE BALANCE SCALE

In balance scale tasks, the child is typically presented with a rigid beam on which a number of pegs have been placed at different distances to the left and right of a fulcrum. The experimenter places some number of equally valued weights on a peg on

Figure 21.1



Generic cascade-correlation nets in various phases of training. Each net has two input units drawn at the bottom and one output unit drawn at the top. Frozen connection weights are drawn with solid lines, trainable connection weights with dashed lines. a. Initial output phase before any hidden units have been installed. b. Input phase, in which each of a pool of candidate units is trained to predict the error at the output unit. c. Next output phase, after installation of best candidate hidden unit. d. Next input phase, to recruit a second hidden unit. e. Next output

the left side and on a peg on the right side[1]. The child's task is to predict what will happen when supporting blocks are removed. Will the scale tip to the left, to the right, or will it balance?

By giving the child the six different kinds of problems shown in figure 21.2, Siegler (1976, 1981) was able to infer the rules children use to solve balance scale problems. Balance problems have an equal number of weights on each side of the fulcrum at equal distances. In weight problems, distance is held constant, but one side of the scale has more weight than the other. In distance problems, the number of weights is held constant, but distance from the fulcrum varies.

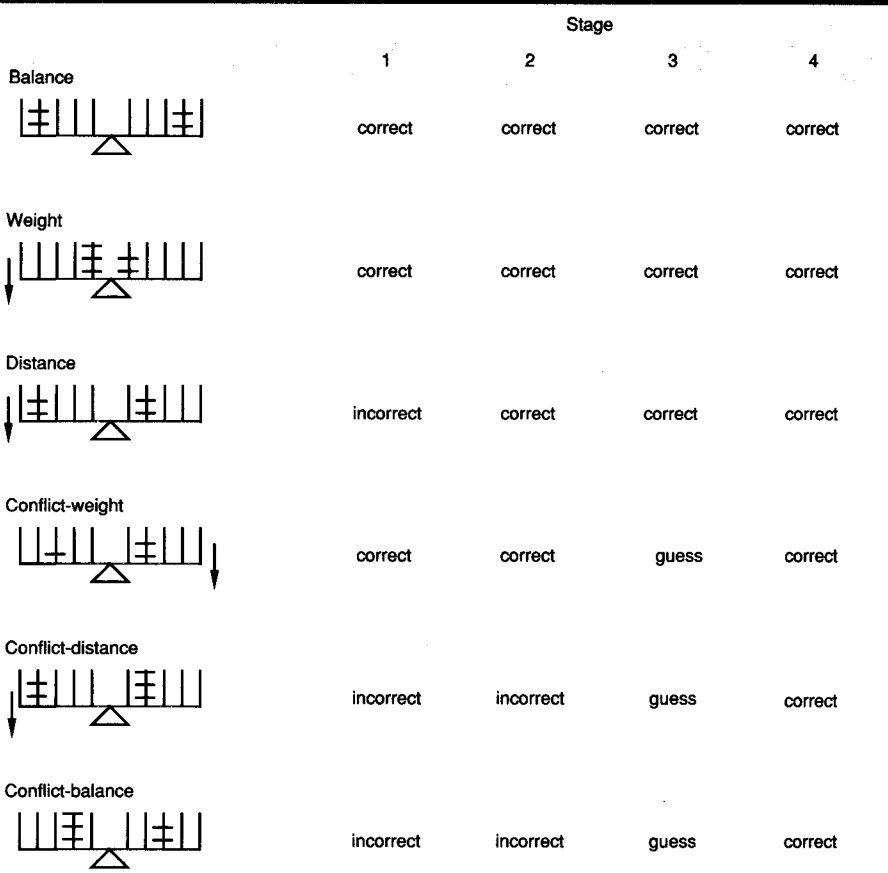
In conflict problems, one side has greater weight and other side has greater distance. In conflict-weight problems, the side with greater weight goes down. In

1. Inhelder and Piaget (1955) initially used baskets of weights suspended from a beam.

conflict-distance problems, the side with greater distance goes down. And in conflict-balance problems, the scale balances.

Siegler (1976, 1981) found that children progress through four distinct rule-based stages on this task between the ages of 5 and 17 years of age. Children's expected performance on the six types of balance scale problems at each of these stages is shown on the right side of figure 21.2. Children in Stage 1 predict outcomes on the basis of how many weights have been placed on each side. In Stage 2, children continue to use weight information, and begin to use distance information when the two sides have equal weights. By Stage 3, they are using weight and distance about equally, but become confused when one side has greater weight and the other side has greater distance. In the final Stage 4, children perform correctly on a wide range of balance scale problems, suggesting to some that they may be comparing the torques on each side of the fulcrum. Torque is the product of weight and distance on one side of the fulcrum.

Figure 21.2

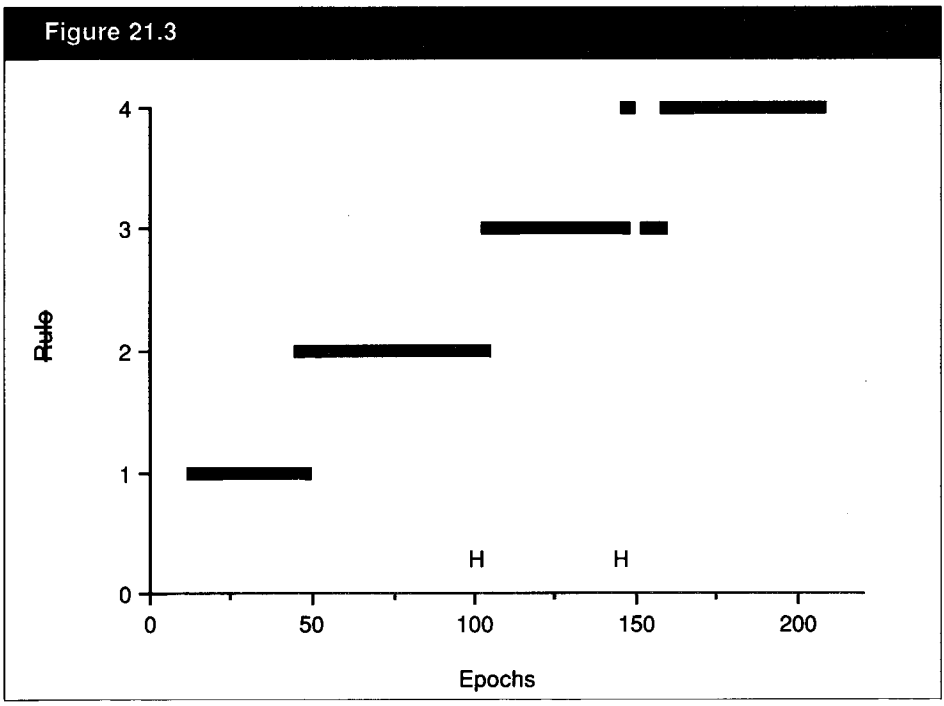


Six types of balance scale problems and predicted performance at four different rule-based stages. The arrows indicate the side of the balance scale that goes down when supporting blocks are removed.

Such stages are characteristic of a large number of problems in which information on two dimensions must be integrated. This generality, clarity, and replicability of these stages have made the balance scale a benchmark for detailed computational modeling in cognitive development. There are now both connectionist and rule-based models of balance scale development.

The four stage sequence of performance on the balance scale has been simulated in cascade-correlation nets (Shultz, Mareschal, et Schmidt, 1994). Rule diagnosis over training epochs for a typical network is shown in figure 21.3. An epoch is a sweep through all of the training patterns. Figure 21.3 shows an orderly progression through Siegler's four rules. The overlap of rule diagnoses at transition points suggests that the stage transitions are quite tentative. The Hs at the bottom of the Figure show the epochs at which a hidden unit was recruited. About one-half of the hidden units installed in these nets were followed by quick stage progressions. Across networks, there was some skipping and some temporary regression back to earlier stages, characteristics that are also common to stage progressions in children.

Figure 21.3



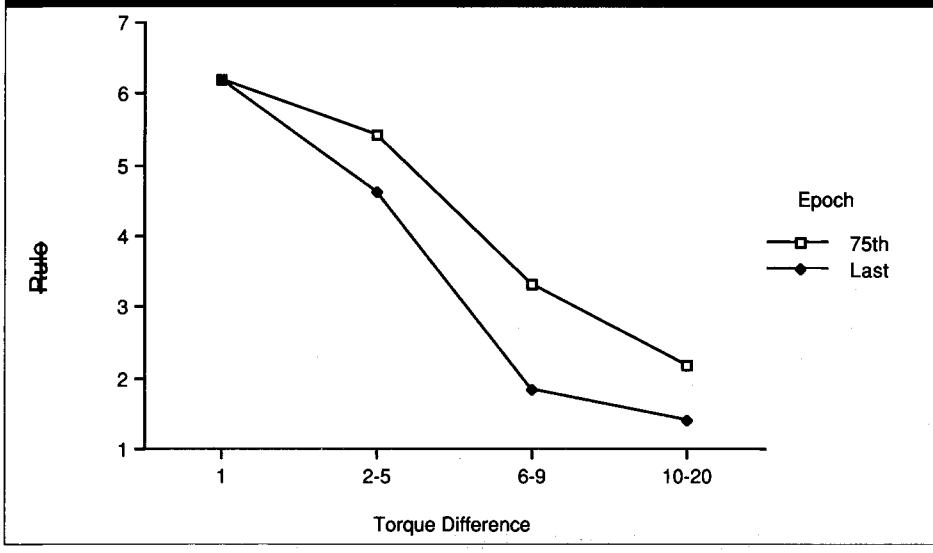
Rule diagnosis over epochs in a representative balance scale network. H marks the epochs at which hidden units are recruited.

The second major psychological regularity in the balance scale literature is the torque difference effect (Ferretti et Butterfield, 1986), wherein balance scale problems with large torque differences are easier for children to solve than problems with small torque differences. Torque difference is the difference between the torque

on one side of fulcrum and the torque on the other side. Results indicate that the larger the absolute torque difference, the easier the problem is for children to solve. This is a perceptual effect, not explainable by Siegler's rules since any such rule should apply regardless of the torque difference.

We performed a second simulation in the same fashion except that test problems were chosen at each of four levels of torque difference. Figure 21.4 plots the mean error midway through training and at the end of training for 16 nets. The nets showed faster and deeper error reduction with increasing torque difference, just as with children.

Figure 21.4



Mean error in 16 balance scale nets as a function of torque difference at two epochs.

Nets in both simulations were trained to predict balance scale outcomes, given various configurations of weight and distance information as input. Two environmental constraints were necessary to produce these results. First, there had to be a strong bias in favor of equal distance problems. This follows McClelland's (1989) assumption that, although children have lots of experience lifting differing numbers of objects, they have relatively little experience placing those objects at differing distances from a fulcrum. The second environmental constraint was a gradual expansion of the training problems, conforming to our assumption that the child's environment changes gradually, providing exposure to an increasing variety of problems.

Our results with cascade-correlation nets were better than all previous computational models of balance scale development. McClelland's (1989) static back-propagation network model could not stay in Stage 4, and it required segregated hidden units for weight and distance information. Newell's (1990) Soar program failed to reach Stage 4 and its ability to acquire the first three stages in order may well have depended on receiving balance scale problems in a particular order. Soar is a production system that learns rules through search-based problem solving. Langley's

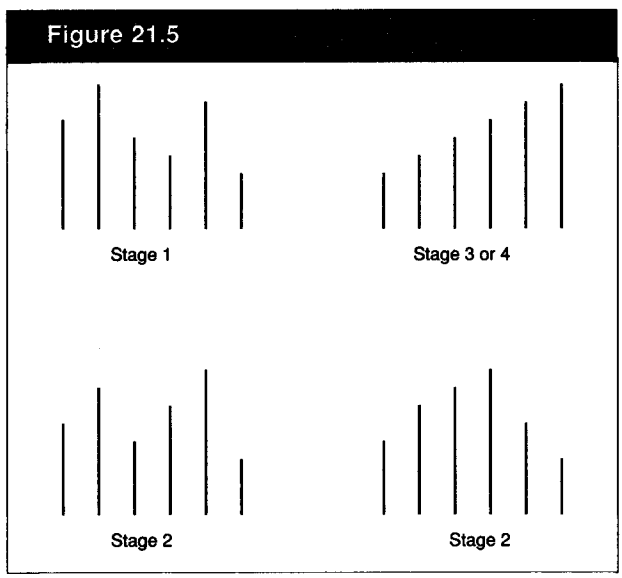
(1987) production rule model modifies existing rules through discrimination learning, and only captured Stage 3 on the balance scale.

None of these previous models tried for the torque difference effect, but we know from our own research (Schmidt et Shultz, 1991) that McClelland's (1989) back-propagation network model can capture it. The rule-based models would seem to be incapable of capturing the torque-difference effect since they do not represent amounts of weight and distance differences. In this sense, perceptual effects like torque-difference are a kind of lever to separate rule-based from connectionist models.

IV. SERIATION

The four stages of development on Piaget's (1941) seriation task have also been simulated with cascade-correlation nets (Mareschal et Shultz, 1993). In the seriation task, the child is asked to sort by length a set of sticks of different lengths that are arranged in a random fashion. Example results from the four stages are presented in figure 21.5. In Stage 1, children move the sticks randomly or seem unable to make any move. In Stage 2, children sort a few sticks, creating sorted subsets of two, three, or four items, but seem unable to complete the entire array. By Stage 3, they achieve a complete sort by a trial and error process, in which moves are often corrected. Finally, in Stage 4, children complete a full sort without errors, by using a systematic procedure such as moving the smallest out of order stick to its correct position. Piaget's evidence indicated that children progress through these stages between four and seven years of age.

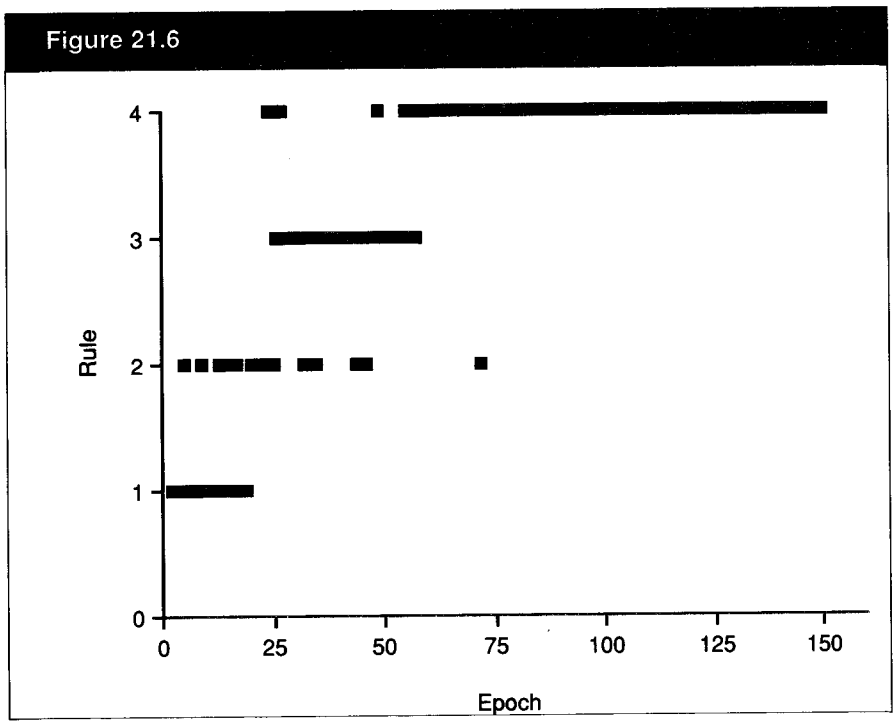
Figure 21.5



Examples of stages of seriation performance.

Figure 21.6 shows stage diagnosis in a representative network over epochs of training. As in the balance scale simulations, there was mostly correct ordering of stages, soft transitions between stages, and some regression back to earlier stages.

Figure 21.6

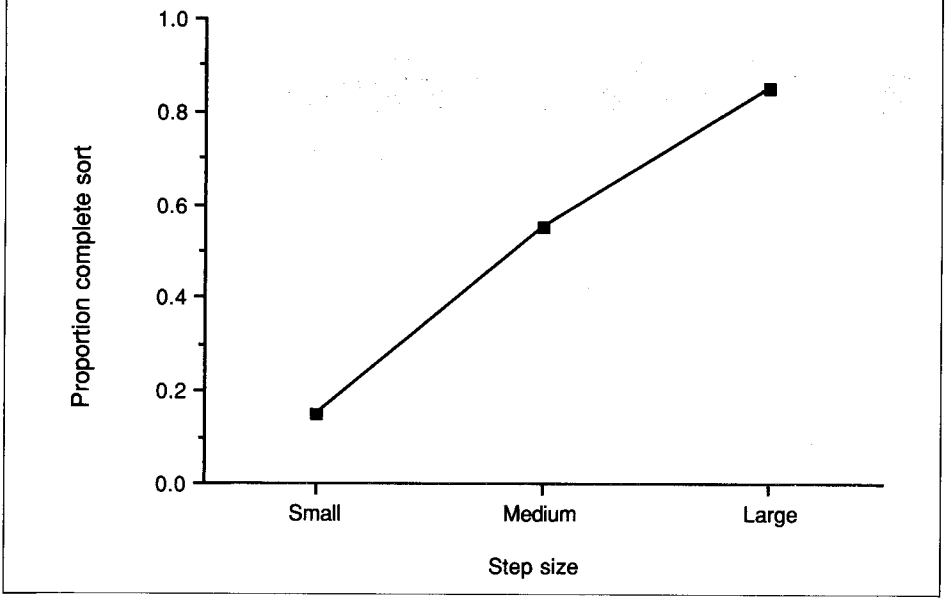


Rule diagnosis over epochs in a representative seriation network.

The seriation simulations also captured well known perceptual effects on seriation tasks such as the tendency for seriation difficulty to increase with decreases in size differences among the sticks (Elkind, 1964; Klingma, 1984). Figure 21.7 indicates that the proportion of nets achieving a complete sort (*i.e.*, reaching Stages 3 or 4) increases with the step size of the inputs. Step size corresponds to the difference in length between adjacent sized sticks.

Certain architectural and environmental constraints were required to capture these seriation effects. First, there had to be two modular nets, one to identify which stick to move and another to identify where to move it. Both modules were presented with the same input, namely the current status of the array of sticks. Second, there had to be a small bias in favor of nearly ordered arrays, conforming to our assumption that such arrays would be more common in the environments of young children. A nearly sorted array could, for example, serve as a cue for a child to finish the sort. Third, there had to some smaller arrays in the training set. We worked principally with arrays of six items, but it was important to include a few arrays of three items in training. It is reasonable that such small arrays would be common in the child's environment, and evidence suggests that children are able to sort these small arrays before they can sort large ones (Koslowski, 1980).

Figure 21.7



Mean proportion of nets performing a complete sort as a function of step size difference.

Again, performance of our cascade-correlation nets were superior to that of previous computational models of seriation. A number of rule-based models (Baylor, Gascon, Lemoyne, et Pothier, 1973; Retschitzki, 1978; Young, 1976) captured the static behavior characteristic of particular seriation stages, but these models showed no transitions, no perceptual effects, and no spontaneous variation between or within individual children.

V. VELOCITY, TIME, AND DISTANCE

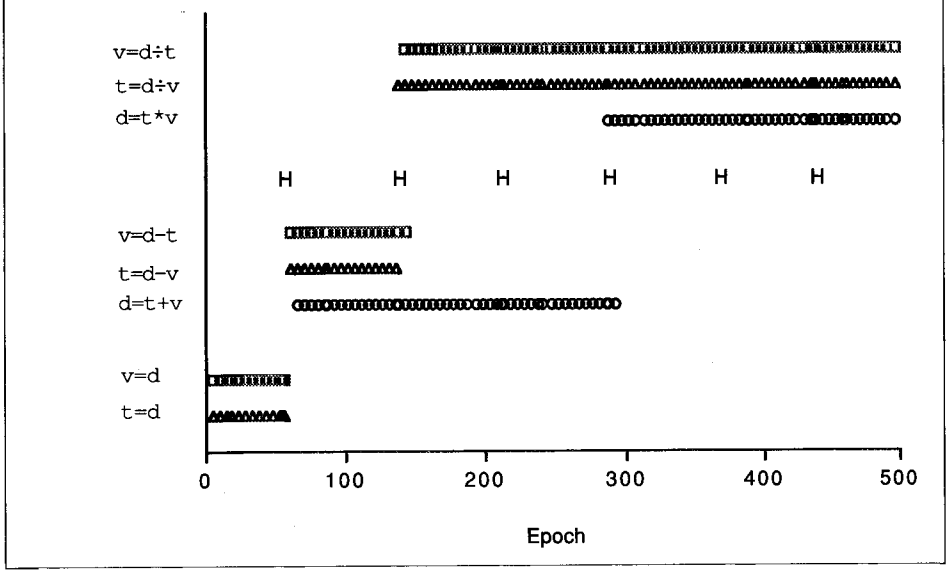
Cascade-correlation nets have also simulated rule-based stages in the integration of velocity, time, and distance information (Buckingham et Shultz, 1994). In classical physics, velocity = distance / time. Thus, distance = velocity * time, and time = distance / velocity. Piaget (1946a et b) wrote two books on how children come to integrate these concepts, and many other researchers followed up his initial investigations. For example, Piaget would show the child two trains running along parallel tracks and ask «Which train travels for the longer time?» Four-year-olds chose the train that traveled for the longer distance, suggesting that their notion of time was based on spatial distance. Later researchers criticized this technique on the basis that it tested the child's ability to ignore information, not to integrate it (Levin, 1977; Wilkening, 1981). For example, on the just mentioned task, travel time information could be directly read from the trains if the child could ignore data on distance, velocity, and all other variables.

A pure inference task was designed by Wilkening (1981). Children had to predict one dimension (*e.g.*, velocity) from knowledge of the other two (*e.g.*, time and distance). For example, three levels of velocity information were represented by a turtle, a guinea pig, and a cat. These three animals were said to be fleeing from a barking dog, and the child was asked to imagine these animals running while the dog barked. The child's task was to infer how far an animal would run given the length of time the dog barked. This would be an example of inferring distance from velocity and time.

Cascade-correlation nets learning similar tasks typically progressed through an identity stage (*e.g.*, velocity = distance), followed by an additive stage (*e.g.*, velocity = distance – time), and finally the correct multiplicative stage (*e.g.*, velocity = distance / time). Many of these stages have been found with children (Wilkening, 1981), and others remain as predictions for future psychological research.

Figure 21.8 shows rule diagnosis in a representative net learning all three inference tasks. Rule diagnosis is based on correlations between network outputs and various algebraic rules like those observed in children, computed every fifth epoch during training. To characterize network performance, an algebraic 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 any other rules did. For velocity and time inferences, this net exhibited an identity rule, followed by a difference rule, followed in turn by the correct ratio rule. Results were similar for distance inferences, except that there was no identity rule. There is no reason the net should favor either velocity or time information in making distance inferences because both velocity and time vary proportionally with distance.

Figure 21.8



Rule diagnosis over epochs in a representative velocity, time, and distance network. H marks the epochs at which hidden units are recruited.

Such rule progressions are natural for cascade-correlation nets. The shift from linear to non-linear solutions occurs because of the progressive recruitment of hidden units. Linear rules include identity (*e.g.*, velocity = distance), sum (*e.g.*, distance = velocity + time), and difference (*e.g.*, time = distance – velocity) rules, whereas non-linear rules include product (*e.g.*, distance = velocity * time) and ratio (*e.g.*, velocity = distance / time) rules. In contrast, static back-propagation nets are unable to capture these stage sequences (Buckingham, personal communication). If a back-propagation net has too few hidden units, it fails to reach the correct multiplicative rules; if it has too many hidden units, it fails to capture the intermediate difference stages on velocity and time inferences. There seems to be no pre-designed back-propagation net topology that can capture all three stages on these tasks.

No alternative computational models have been applied to the velocity, time, and distance phenomena.

CONCLUSION

Cascade-correlation nets implement the right sort of model for simulating cognitive development. They show the ability to capture rule-based stages (all three simulations), perceptual effects (balance scale and seriation), and developmental transitions (all three simulations).

The basis for rule-like behavior in cascade-correlation nets, as found in these and other simulations (Shultz, Schmidt, Buckingham, et Mareschal, in press), is the ability of the nets to extract statistical regularities from the learning environment. These include simple linear regularities as well as more complex non-linear regularities, signaled in cascade-correlation by the recruitment of new hidden units into the network. Simple linear regularities include the use of weight information on the balance scale and identity rules in the integration of velocity, time, and distance cues. More complex non-linear regularities include the torque rule on the balance scale and correct ratio rules in integrating velocity, time, and distance cues.

Learning of rule-like behaviors in psychologically realistic stage sequences is a matter of both domain-specific factors like environmental bias and task modularization and domain-general factors like a summative activation rule and the recruitment of hidden units. Environmental bias favoring equal distance problems forced balance scale nets to focus on weight information to the temporary exclusion of distance information. Modular nets were required for generating seriation phenomena. Use of an activation rule that sums the inputs to units was important in producing early additive rules on balance scale and velocity, time, and distance judgments. Recruitment of hidden units was important in the eventual acquisition of non-linear rules, such as the torque rule on the balance scale (compare the torques on each side of the fulcrum) and the correct ratio rules in integrating velocity, time, and distance cues.

Perceptual effects reflect the continuous nature of network computations in tasks where quantitatively described items are mapped to a qualitative comparison. In such cases, different sources of quantitative information must be compressed to reach a qualitative decision. Whenever the relevant quantitative inputs are large and

clear, the qualitative decision is easier. This characterizes the torque difference effect on the balance scale and the effect of stick size differences in seriation. No perceptual effects would be expected on tasks like the integrating velocity, time, and distance where quantitative inputs are mapped onto a quantitative output. No matter what size differences are represented on the inputs, the net must learn predict the output as exactly as possible.

The performance of cascade-correlation nets can be contrasted with that of other modeling techniques, both rule-based and connectionist. Symbolic rule-based models often have difficulty with stage transitions (balance scale and seriation simulations), perceptual effects (balance scale and seriation simulations), and variation within and between subjects (seriation simulations). Static connectionist networks, such as back-propagation, can capture perceptual effects and variations, but often have difficulty with stage sequences, as in the balance scale simulations.

Cascade-correlation nets simulate developmental transitions with the dual technique of hidden unit recruitment and connection weight adjustment. This allows modeling of both underlying qualitative and quantitative changes. As such, this sort of model allows a novel and precise re-formulation of Piaget's notions of assimilation and accommodation (Shultz *et al.*, in press).

It is possible to imagine three types of cognitive encounters in cascade-correlation nets. First, there is pure assimilation without learning. This occurs via correct generalization to previously unseen patterns, with neither connection weight changes nor hidden unit recruitment. Second, there is assimilative learning via connection weight adjustment, but without hidden unit recruitment. Here, the net learns new patterns that do not require non-linear increases in representational power. Piaget had no way of describing learning without accommodation, *i.e.*, without underlying qualitative change. Finally, there is accommodation via hidden unit recruitment, where the net needs to increase its computational power.

These types of cognitive encounter are not merely qualitative distinctions, but rather regions on a quantitative continuum of learning difficulty. Pure assimilation requires minimal learning and pure accommodation requires extensive learning. Furthermore, all three processes are driven by same underlying mechanism of error reduction, *i.e.*, reducing the discrepancies between expectations and outcomes. Further development of these ideas could lead to a novel and productive theory of cognitive development.