JOURNAL OF COGNITION AND DEVELOPMENT, *1*(3), 305–345

The Developmental Course of Distance, Time, and Velocity Concepts: A Generative Connectionist Model

David Buckingham and Thomas R. Shultz
Department of Psychology
McGill University
Montreal, Canada

Connectionist simulations of children's acquisition of distance (d), time (t), and velocity (v) concepts using a generative algorithm—cascade-correlation—are reported. Rules that correlated most highly with network responses during training were consistent with the developmental course of children's concepts. Networks integrated the defining dimensions of the concepts first by identity rules (e.g., $v = d$), then additive rules (e.g., $v = d - t$), and finally multiplicative rules (e.g., $v = d \div t$). The results are discussed in terms of similarity to children's development, the contribution of connectionism to the study of cognitive development, contrasts with alternative models, and directions for future research.

Children's understanding of distance, time, and velocity concepts has been of interest to researchers ever since Piaget (1946/1969, 1946/1970) published two volumes describing different stages in their development. Although different representations and processes have been proposed since Piaget's work, the basic idea that children's knowledge is characterized as a progression through increasingly complex stages has endured (e.g., Acredolo, Adams, & Schmid, 1984; Crépault, 1978; Siegler & Richards, 1979; Wilkening, 1981). Unfortunately, until recently, cognitive development researchers have lacked the tools needed not only to describe knowledge representations of possible stages but also to explain their

Requests for reprints should be sent to David Buckingham, Department of Psychology, McGill University, 1205 Docteur Penfield Avenue, Montreal, QC, Canada H3A 1B1. E-mail: dave@psych.mcgill.ca

emergence. However, this situation has changed during the last decade with the use of connectionist models to describe developmental regularities (Bates & Elman, 1993; Elman et al., 1996; Plunkett & Sinha, 1992).

In this article, connectionist methodology is used to investigate the development of distance, time, and velocity concepts. It is argued that the domain-general constraints of cascade-correlation—the connectionist architecture used (Fahlman & Lebiere, 1990)—and the domain-specific constraints of the learning environment not only suggest knowledge representations different from those previously proposed but also demonstrate how a progression through qualitatively different knowledge structures is possible.

The development of distance, time, and velocity concepts is of interest for two main reasons. First, on any given day, a child experiences many moving objects: animate or inanimate, self- or other-initiated, and self- or other-affected. In all these movements, information concerning the distance, time, and velocity of the movement is perceptually available. The sheer number and quality of experiences a child has with these concepts suggest that their development may play an important role in a more general area of cognition—*compensation development*. During compensation development, children acquire the ability to integrate physical dimensions to predict some potential outcome (Kerkman & Wright, 1988). A number of tasks have been used to investigate this ability. All involve differences, rates, ratios, proportions, or other multidimensional interactions among physical dimensions. Examples include (a) the balance scale task (e.g., Inhelder & Piaget, 1958; McClelland, 1989; Newell, 1990; Schmidt & Ling, 1996; Shultz & Schmidt, 1991; Siegler, 1976, 1981; Wilkening & Anderson, 1982, 1991), (b) area judgment tasks (e.g., N. H. Anderson & Cuneo, 1977; Avons & Thomas, 1990; Lohaus & Trautner, 1989; Wilkening, 1980), and (c) volume judgment tasks (e.g., N. H. Anderson & Cuneo, 1977; Halford, Brown, & Thompson, 1986; Wilkening, 1980).

Second, the relation among the three concepts has a reversible structure in that each concept can be defined by the other two. In classical physics, distance (d) is defined as d = time (t) × velocity (v), time as $t = d \div v$, and velocity as $v = d \div t$. Thus, a child can think about distance in terms of time and velocity, time in terms of distance and velocity, or velocity in terms of distance and time. For example, suppose a child normally walks to school in 10 min. The child might wonder alternatively about how close to school he or she would be after only walking for 5 min, how long it takes to run to school, or how fast he or she would have to ride his or her bike to get to school in 2 min. Moreover, understanding the relations between the variables requires both direct and inverse functional reasoning, that is, less time (*velocity*) implies less distance but more velocity (*time*). For example, if the child walks to a park that is closer than the school, it will take less time. On the other hand, it takes less time to run to the park than to walk to it. Thus, understanding distance, time, and velocity requires knowledge of how each is related to the other two.

The first two sections of this article provide a review of relevant psychological work and a discussion of connectionism and development, respectively. The second section also describes the cascade-correlation algorithm. The remaining sections are concerned with the method, results, and discussion of this modeling effort. To our knowledge, no other computational models exist, connectionist or otherwise, of this important and interesting domain.

LITERATURE REVIEW OF DISTANCE, TIME, AND VELOCITY ACQUISITION

Early Research

Piaget (1946/1969, 1964) began investigating children's concepts of distance, time, and velocity after Albert Einstein inquired about the relation between time and velocity in children's thinking. Einstein wanted to know if one concept was more primitive or if one depended on the other. Given that Newtonian mechanics defines velocity in terms of time and space, whereas relativity theory postulates that time and space are relative to velocity, the source of Einstein's interest is evident. Piaget concluded that children's earliest intuitions were more akin to relativity theory in that the concept of velocity exists independent of notions of duration and distance. In contrast, time is dependent on velocity throughout development (Piaget, 1946/1969).

More specifically, Piaget (1946/1969, 1946/1970, 1964, 1970/1971; see also Flavell, 1963) determined that mature concepts of distance, time, and velocity develop through four stages: (a) intuitive notions, (b) intermediate concepts, (c) concrete operational understanding, and (d) a shift from qualitative to quantitative conceptualization. The intuitive notions of the concepts were said to emerge at 4 to 5 years of age and were thought to be undifferentiated. For example, when children are presented with two mechanical snails that start and stop moving at the same time but travel at different speeds, they choose the snail that travels the least distance as having stopped first. Moreover, the faster moving snail is judged to take more time. In this sense, time is said to be undifferentiated from distance and velocity. Thus, the goal of development is to construct differentiated concepts. Once this occurs, children are able to consider the proportion of distance to time to predict velocity (i.e., $v = d \div t$), for example.

Since Piaget's investigations, research examining the interrelation of the three concepts has been rare. Most has looked at the intradevelopment of time (for a re-

view, see Friedman, 1978, 1990), although some has looked at the interdevelopment of time and velocity (e.g., Weinreb & Brainerd, 1975). Even fewer studies (e.g., Crépault, 1977, 1979, 1981; Montangero, 1977, 1979) have investigated the interdevelopment of all three concepts. One exception, a rigorous experiment by Siegler and Richards (1979), looked at both the intra- and interdevelopment of the three concepts. However, Siegler and Richards were concerned with when the concepts became differentiated and not when children could integrate the dimensions to predict an outcome.

Levin and her collaborators (e.g., Levin & Gilat, 1983; Levin, Gilat, & Zelniker, 1980; Levin, Israeli, & Darom, 1978) conducted a series of experiments that raised questions about Piaget's conclusions and methodology. For example, Levin (1977) argued that in a typical Piagetian task in which children are presented with two moving objects, distance and velocity cues interfere with children's understanding of time. Moreover, Levin (1979) showed that the interference is not due to the fact that they are logically related to time. Cues unrelated to time (e.g., the brightness of a lamp) showed similar interference effects.

Investigating the Interrelation of the Concepts

An exception to the lack of a thorough investigation of the interrelation of the three concepts is work from Wilkening (1981), to which we now turn. Wilkening succeeded in developing three tasks in which mastery involved the quantitative integration of two concepts to predict the magnitude of the third. Moreover, rather than testing the ability of children to ignore the defining dimensions, Wilkening's tasks require an inference based on the defining dimensions. Because such inferences are essential to demonstrating knowledge of the relations specified by the distance, time, and velocity equations, Wilkening's tasks serve as the basis for our simulations and are discussed in some detail.

Wilkening's tasks were conceptualized within the Information Integration Theory framework (N. H. Anderson, 1974, 1991). In brief, the theory posits that children use cognitive algebraic models to integrate graded inputs of physical stimuli. It has been used to explain a diverse range of developmental phenomena, including probability judgments (e.g., Acredolo, O'Conner, Banks, & Horobin, 1989), area judgments (e.g., N. H. Anderson & Cuneo, 1977), and performance on the balance scale task (Wilkening & Anderson, 1982). In general, *development* is described as a progression from identity (centration) to either adding, subtracting, or averaging models to multiplying or dividing models, with a transition stage in which neither type of model entirely explains performance.

Wilkening's tasks involve presenting children with information about the defining dimensions and asking them to predict the magnitude of the concept. For example, in a distance task, children were shown an apparatus that had, at one end of a footbridge, a dog and several animals that were said to be frightened of the dog. The children were told that the animals would run along the bridge as soon as the dog began to bark and would stop when the barking ceased. The task involved determining how far each animal would run. Thus, the children were given the characteristic velocity of the animals and the time they ran (the duration of barking) and asked to infer the distance they would run.

Wilkening studied three age groups: 5-year-olds, 10-year-olds, and adults. To assess performance, Wilkening used *functional measurement* (N. H. Anderson, 1974), a technique that employs graphical and statistical analyses to assess participants' responses. This analysis revealed the following: (a) In the distance task, all age groups used the correct multiplication rule, $d = t \times v$; (b) in a time task, 10-year-olds and adults employed the correct division rule, $t = d \div v$, whereas 5-year-olds used a subtraction rule, $t = d - v$; (c) in a velocity task, the two older age groups used a subtraction rule, $v = d - t$, and the 5-year-olds used an identity rule, $v = d$. Wilkening concluded that young children did have the ability to integrate these dimensions. However, he was unwilling to make comparative claims about the developmental rates of the three concepts because it appeared that there were differing memory demands across the three tasks. For example, in the distance task, but not in the velocity task, participants of all age groups were able to use an eye-movement strategy in which they appeared to "follow" the imaginary animal as it ran across the footbridge.

In a follow-up study, Wilkening (1982) attempted to increase the memory demands of the distance task by presenting time (barking) before velocity information (animal identity) and lessen the memory demands of the velocity task by visually presenting the time information. The modifications partially supported his hypothesis, in that 5-year-olds were observed to use an additive rule ($d = t + v$) in the distance task. However, the results for the velocity task remained unchanged.

As Wilkening (1982) acknowledged, his experiments left open the question of whether adult participants were integrating distance and time information according to the normative multiplicative rule when making velocity judgments. One possibility is that participants were unable to take advantage of the visually presented time information. Another possibility, as noted by Wilkening (1981), is that unlike in the distance and time tasks, the response scale in the velocity task was not objectively linear. To indicate their velocity judgment, participants chose which of seven animals was capable of running the given distance in the given time. Wilkening argued that there was some evidence that the response scale was logarithmic. As such, performance consistent with dividing distance and time information would have yielded parallel functional measurement graphs and, thus, been interpreted as evidence for integration based on subtraction. Therefore, it remains to be seen whether adults' inability to integrate time and distance information multiplicatively is intrinsic or a reflection of task demands.

COGNITIVE DEVELOPMENT WITHIN A CONNECTIONIST FRAMEWORK

A number of researchers (e.g., Elman, 1993; Elman et al., 1996; McClelland, 1989; Rumelhart & McClelland, 1986; Schyns, 1991; Shultz, Schmidt, Buckingham, & Mareschal, 1995) have begun to use connectionist models to investigate cognitive development. Part of the appeal of connectionism is the use of brain-inspired computation. Connectionist models consist of networks of simple processing units that send inhibitory or excitatory signals to each other via weighted connections. In the models relevant to this discussion, learning occurs by adjusting the weights of the connections between an input layer, in which stimulus information is encoded, and an output layer, in which the response is made. Two general classes of networks include static networks that have a constant topology (e.g., Rumelhart, Hinton, & Williams, 1986) and generative networks that grow in size as learning progresses (for a review of generative algorithms, see Alpaydin, 1991). From a developmental point of view, generative models can be viewed as taking the brain-style computation metaphor one step further to include the addition of new connections among units as an important part of development. Indeed, within the field of neuroscience, synaptogenesis and neurogenesis have been suggested as neural bases of development and learning (Gould, Reeves, Graziano, & Gross, 1999; Greenough, Black, & Wallace, 1987; Quartz & Sejnowski, 1997).

In addition to offering new insight into some old problems, connectionism also addresses some developmental issues that either had been overlooked or ignored. As early as 1969, Flavell and Wohlwill argued that an account of cognitive development must concern itself with both the formal and functional aspects of development. In other words: (a) What knowledge structures develop? and (b) How does developmental transition occur? Research concerning what structures develop flourished, but by the mid-1980s relatively little work had been conducted in the area of transition mechanisms (Sternberg, 1984). Bates and Elman (1993) suggested that this was partly due to the widely accepted computer metaphor of cognitive development. In brief, symbolic computational assumptions—such as discrete representations (i.e., symbols), rules to manipulate symbols, a view of learning as programming, and the relative unimportance of possible implementation constraints (i.e., functionalism)—fostered a paradigm that considered mechanisms of change as somewhat unimportant. Conversely, the basic assumptions of connectionism (i.e., distributed representations, graded knowledge in the form of weighted connections, learning characterized as structural change, and consideration of implementation constraints) offered a new approach to study not only what develops but how it develops (for similar points of view, see Churchland, 1990; Plunkett & Sinha, 1992).

Piaget believed that stage transitions resulted from emergent structures mediated by adaptive accommodation and assimilation. However, despite his efforts,

the mechanism that caused transitions remained vague (Bates & Elman, 1993). Recently, a number of authors have suggested that connectionism not only provides a precise account of transition but also new interpretations of assimilation and accommodation (e.g., Bates & Elman, 1993; Plunkett & Sinha, 1992). For static networks, researchers suggest that gradual and continuous weight changes result in stagelike performance (Plunkett & Sinha, 1992). McClelland (1989) argued that *accommodation* occurs when weights are updated during learning, thus modifying the structure of knowledge. *Assimilation* occurs when generalization to a new instance or input does not result in any weight change. Generative architectures provide a second potential transition mechanism—the recruitment of new units that increase the computational complexity of the network (Mareschal & Shultz, 1996; Shultz et al., 1995). Accommodation occurs as new units are added, providing a network with qualitatively different representational power. Weight changes reflect what could be termed assimilative learning, that is, learning without major structural change. More recently, Shultz (1994) suggested that one generative connectionist architecture, cascade-correlation (Fahlman & Lebiere, 1990), can be interpreted as engaging in a type of representational redescription similar to that proposed by Karmiloff-Smith (1992).

The Cascade-Correlation Algorithm

Cascade-correlation is a generative learning algorithm in which the network topology is created dynamically by the addition of required hidden units as training progresses. Figure 1a illustrates the initial topology of a network with three input units and one output unit. The network begins as a simple perceptron with direct connections from the input-to-output layer. There is no intermediary hidden unit layer at this point. There are three input units to code a stimulus and an obligatory bias unit.

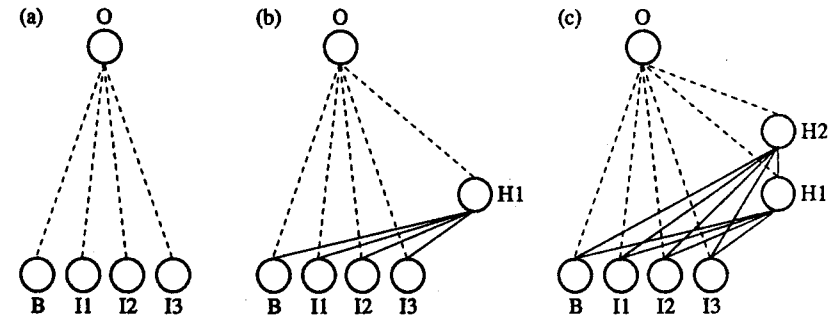

FIGURE 1 Architecture at beginning (a), after one hidden unit is added (b), and after a second hidden unit is added (c). Solid and dashed lines indicate frozen and trainable connections, respectively. B, I, H, and O refer to bias, input, hidden, and output unit, respectively.

The bias unit always has an activation of one and provides learnable thresholds for all units (except input units). Figures 1b and 1c illustrate the network structure following the addition of one and two hidden units, respectively.

Training consists of a series of two-phase training cycles—each cycle involves "epochs" of training in which an epoch entails a single presentation of all training patterns. During an output training phase, weights from input units and existing hidden units to the output layer are adjusted in an attempt to reduce the error between the actual output activations computed by the network and the target (desired) output activations. The output training phase continues until either (a) error has been reduced to some preestablished criterion (in which case "victory" is declared and training is terminated), (b) error can no longer be significantly reduced, or (c) some maximum number of epochs is reached. When either of the latter two conditions is met, an input training phase begins. Input training is used to recruit (add) a hidden unit into the network. At the start of an input phase, a pool of candidate hidden units (default value of eight units) have connections from all input units (including the bias) and any existing hidden units. At this point, candidate hidden units are not connected to output units. The algorithm then adjusts the weights into these candidates to maximize the correlation between activations of candidate hidden units and error at the output units. Weight adjustment continues until either the correlations can no longer be significantly changed or some maximum number of epochs is reached. When this occurs, the highest correlating candidate hidden unit is installed into the network by connecting it to the output units, its input-side weights are frozen (i.e., these weights do not undergo any further training), and a new output training phase begins. (For more detailed treatments of cascade-correlation learning, see Fahlman, 1988; Fahlman & Lebiere, 1990; Hoehfeld & Fahlman, 1991; Shultz et al., 1995; Yang & Honavar, 1998.)

THIS STUDY

In this study, we train cascade-correlation networks in a task environment analogous to the one used in Wilkening's experiments. The networks have to predict, as output, the value of one dimension (e.g., velocity) given information about the other dimensions (e.g., distance and time). We chose this task environment because it addresses the original question of interest: What do children know about the interrelation of distance, time, and velocity?

There are a number of questions we hope to answer with our modeling effort. Within a general computer modeling framework, we can ask if the developmental course of the three concepts is similar when task demands are held constant. Recall that one of the difficulties of interpreting Wilkening's (1981, 1982) results within and across concepts was that task demands may not have been equivalent in terms of memory requirements and response scales. This could explain why children and adults can make time inferences but not velocity inferences using a normative rule. The use of computer simulation allows us to ensure that both the memory demands and the response scales are equivalent across these two inference tasks.

Within a connectionist framework, we can address the following questions. Can networks that represent and process information using weighted connections among simple units account for knowledge representations previously characterized by cognitive algebra? Moreover, if networks are capable of such representations, will they initially perform as if using identity rules (e.g., $v = d$), progress to additive rules (e.g., $d = t + v$), and finally to multiplicative rules (e.g., $t = d \div v$) as they gain experience with the environment? In other words, do networks go through the same developmental course as do children? If networks successfully develop, what aspect of the cascade-correlation algorithm enables transition from one representation to another? Answering these questions could begin to provide a mechanistic account of the development of distance, time, and velocity concepts.

We chose the cascade-correlation algorithm to train the networks for a number of reasons. In general, it seems appropriate to use a feedforward supervised learning algorithm given the abundance of experiences children have with moving objects in which perceptual information can be used not only as input but also as target feedback. For example, imagine a child watching cars traveling down a street. The distance traveled, time taken, and speed of the cars are all perceptually available. When learning how time and speed relate to distance, the child can use the actual distance traveled as a feedback target. In networks, error is computed as the discrepancy between expectations (actual outputs) and observed outcomes (target outputs). Such motion information could be placed in alternative frameworks, such as auto-associator networks, but we favor our feedforward scheme because of its greater capacity to abstract the actual underlying nonlinear functions by means of hidden unit recruitment. Auto-associator networks, for example, have quite limited capacity in terms of both the number and type of patterns they can learn, and they cannot abstract nonlinear relations (Hertz, Krogh, & Palmer, 1991).

Moreover, cascade-correlation already has proven useful in understanding a number of cognitive developmental phenomena, including children's performance on the balance scale task (Shultz, Mareschal, & Schmidt, 1994; Shultz & Schmidt, 1991), the acquisition of personal pronouns (Shultz, Buckingham, & Oshima-Takane, 1993), seriation (Mareschal & Shultz, 1993), conservation (Shultz, 1998), and discrimination shift learning (Sirois & Shultz, 1998). We felt that networks that can increase their nonlinear representational power by recruiting hidden units would provide insight into how children progress from simple centration (identity) rules to more complex additive rules to normative multiplicative rules in the domain of distance, time, and velocity development.

METHOD

Networks

The initial network topology consisted of three input banks, one each for distance, time, and velocity information, connected to a single linear output unit that produces the sum of all weighted input it receives. The number of units per input bank depends on the type of encoding. Five types of input encoding commonly found in the connectionist literature were used to determine the effect of input encoding on performance and the overall robustness of the results. These included three distributed types of encoding: *mercury* (e.g., Harnad, Hanson, & Lubin, 1991), *thermometer* (e.g., J. A. Anderson, 1990), and *gaussian* (e.g., Lacouture & Marley, 1991); one local encoding, *n*th (e.g., McClelland, 1989); and one "partially distributed" encoding, *integer* (e.g., Shultz & Schmidt, 1991). Two variants of integer encoding, *integer-context* and *distributed-integer,* also were investigated.

In distributed representations, more than one unit is used to represent any one input value, and the same unit is involved in the representation of more than one input value. In mercury coding, the first *n* units corresponding to the integer *n* have an activation of 1, and all other units have an activation of zero. The total number of units used is equal to the maximum input value. In thermometer coding, the *n*th, *n*th + 1, and *n*th + 2 units have an activation of 1, and all other units have an activation of zero. Thus, the total number of units is two more than the maximum input value. The gaussian coding used is the same as thermometer coding except that the *n*th + 1 unit has an activation of 3.

In local representations, each unit is used exclusively to represent a given input value. For example, in *n*th coding, for any input value *n,* the *n*th unit has an activation of 1, and all other units have an activation of zero. As with mercury coding, the total number of units is equal to the maximum input value used.

Partially distributed representations combine both local and distributed properties. For example, in integer coding, only one unit is used per input group, but the unit can be used to represent more than one input value by assigning the integer value of the dimension. Distributed-integer coding is an extension of integer coding that uses two units per input group. It is a type of interpolation encoding (Ballard, 1987) in which the activations of the two units are mapped in opposition to each other. In our implementation, the activation on the first unit is simply half the input value. Thus, for integer input values ranging from 1 to 5, the unit's activation would be 0.5, 1.0, 1.5, 2.0, and 2.5, respectively. In contrast, for the same range of integer input values (1–5), the second unit's activation would be 2.5, 2.0, 1.5, 1.0, and 0.5, respectively. Note that for any input value, the sum of total activation across the two units is 3. Finally, integer-context coding is the same as integer with the addition of three context units, one representing each type of inference problem. The appropriate context unit receives an input value of 1, and the other two units receive a value of zero. Thus,

the three input banks are encoded using partially distributed representations, whereas the context is encoded using local representations.

As mentioned, the input banks were connected to a linear output unit. We chose this type of output unit because it is the most direct way of producing quantitative outputs similar to the responses made by participants in Wilkening's experiments.[1] Any hidden units added to the network during training had a sigmoid activation function ranging from –0.5 to +0.5. This S-shaped function enables nonlinear computation because incoming activations that sum to produce a value in the middle range of the function are inflated or suppressed relative to the maximum and minimum values, respectively, whereas extreme values level off.

Training Patterns

Training patterns consist of input values and a target, or output value. Input values are used to encode the event (e.g., the distance traveled and the amount of time it took), and the output value is the outcome (e.g., the velocity). There were three classes of inference patterns: distance, time, and velocity. The distance class, for example, were those patterns in which distance was to be inferred, given time and velocity information as input.

The input values of the two known dimensions were the integers from 1 to 5, whereas the input value of the dimension to be inferred had a value of zero to indicate that the magnitude was unknown.[2] Thus, for any given input pattern, one input group would be all zeros, whereas the other two groups had dimensional values between 1 and 5. An example of each of the three types of input patterns is illustrated in Table 1 using mercury encoding. All combinations of the defining dimension values (1–5) were combined as input patterns, for a total of 25 distance, 25 time, and 25 velocity training patterns, in which distance, time, and velocity were to be

[1]Other encodings, such as those used for the input representation, would require decisions concerning what would be a sufficient implementation to capture output responses and what auxiliary procedures would be used to convert unit activations into magnitude values that could be compared to Wilkening's response scales. Ultimately, these decisions would be arbitrary. For example, in terms of implementation, if we had chosen *n*th encoding, we would have to decide the level of preciseness we required. Would we use 10 units to represent the first decimal place or would 5 units, each corresponding to successive intervals of 0.2, suffice? Would we want to represent more than one decimal place? No matter how we chose to implement the response scale, we would have to decide how to interpret the activation patterns produced by the network. For example, we would have to choose what activation would be sufficient for a unit to be considered "on," how to resolve conflicts when two units were on, what it meant if no units were turned on, and so forth.

[2]Because an input's contribution to the network's response is determined by multiplying its activation by the connection weights to any hidden and output units in the network, an input value of zero removes any influence it might have on the response. Thus, conceptually speaking, an input of zero does not imply that the magnitude of the dimension is zero but rather that there is no input from this dimension.

TABLE 1
Example Input of Distance, Time, and Velocity Inferences for Mercury Networks

	Input Group		
Inference	*Distance*	*Time*	*Velocity*
Distance	00000	11100	11111
Time	11100	00000	11111
Velocity	11100	11111	00000

Note. In these examples, the two known dimensions receive values of 3 and 5. The dimension to be inferred has an input value of 0.

inferred, respectively. Target output values were calculated using the three Newtonian equations ($d = t \times v$, $t = d \div v$, and $v = d \div t$). In addition, distance target values, which would vary between 5 and 25, were scaled by dividing by 5, so that the range would be consistent with that of the time and velocity inference patterns (which vary from 1 to 5).

Procedure

Twenty networks per input encoding type were trained. For each network, training continued until the actual output activation produced in response to each training pattern was within score-threshold (0.1) of the target activation. Each epoch of training involved the presentation of all 75 training patterns (25 distance, 25 time, and 25 velocity patterns) followed by weight adjustments. Testing was conducted once at the beginning and ending of output training phases as well as every five epochs during these phases. Because output activations do not change during input training, testing during the input phase is redundant, as the results are the same as in the last epoch of the output training phase.

At each testing epoch, output activations, connection weights, and the sum of squared error were recorded for the entire training set (75 patterns) and individually per pattern type (distance, time, or velocity). In addition, the total number of hidden units recruited by the network as well as the epoch at which they were installed into the network were recorded.

To investigate the generalization of network performance, five additional networks per encoding condition were trained on 57 randomly selected patterns (19 distance, time, and velocity patterns, respectively). The remaining 18 patterns (6 distance, time, and velocity patterns, respectively) were used to test for generalization.

Across all simulations, Fahlman's default values were used for all cascade-correlation parameters except that score-threshold (the amount that an actual output can deviate from its desired output) was lowered to 0.1, which is more appropriate for linear outputs.

Treatment of Output

The focus of interest was determining what rules best captured the overall performance of a network at a given testing epoch. Because highly accurate computation was not demanded of the human participants in Wilkening's work, an attempt to reflect this was made by obtaining a Pearson product–moment correlation coefficient of the relatedness of output values predicted by possible rules with the actual output of the network.

The set of possible rules included all rules observed by Wilkening (1981, 1982) as well as others derived from Information Integration Theory. For distance inferences, output values were calculated according to the following three classes of rules: (a) two *identity* rules, in which the outcome was determined solely by the time dimension, $d = t$, or the velocity dimension, $d = v$; (b) three *additive* rules, $d = t + v$, $d = t - v$, and $d = v - t$; and (c) three *multiplicative* rules, $d = t \times v$ (the correct Newtonian rule), $d = t \div v$, and $d = v \div t$. Rules analogous to the distance inference rules were used to assess time and velocity inferences.

RESULTS

Observed Stages

To determine the developmental course of a network, we plotted the r^2 values associated with each rule as training progressed. Because this type of plot offers a detailed, quantitative view of the dynamic nature of the networks' development, an example, derived from a single network in the *n*th encoding condition, is shown in Figure 2. However, to illustrate the general development of the networks, it is clearer to simply plot the distance, time, and velocity rules that were both the best predictors of a network's performance and that accounted for a significant amount of the variance, $r^2 \geq .25$, $df = 24$, $p < .01$. Therefore, in Figure 3, we present such plots for one typical network in each input encoding condition. These networks were chosen as illustrative examples of general trends considering the progression of stages, the mean onset and lengths of stages, and when hidden unit recruitment occurred. Note that the network from the *n*th encoding condition depicted in Figures 2 and 3a is the same network to enable a comparison of the two plotting schemes.[3]

In general, most networks exhibited developmental sequences similar to those observed by Wilkening (1981, 1982), that is, for networks in the *n*th, mercury,

[3]A preliminary report of the performance of networks with *n*th unit coding appears in Buckingham and Shultz (1994).

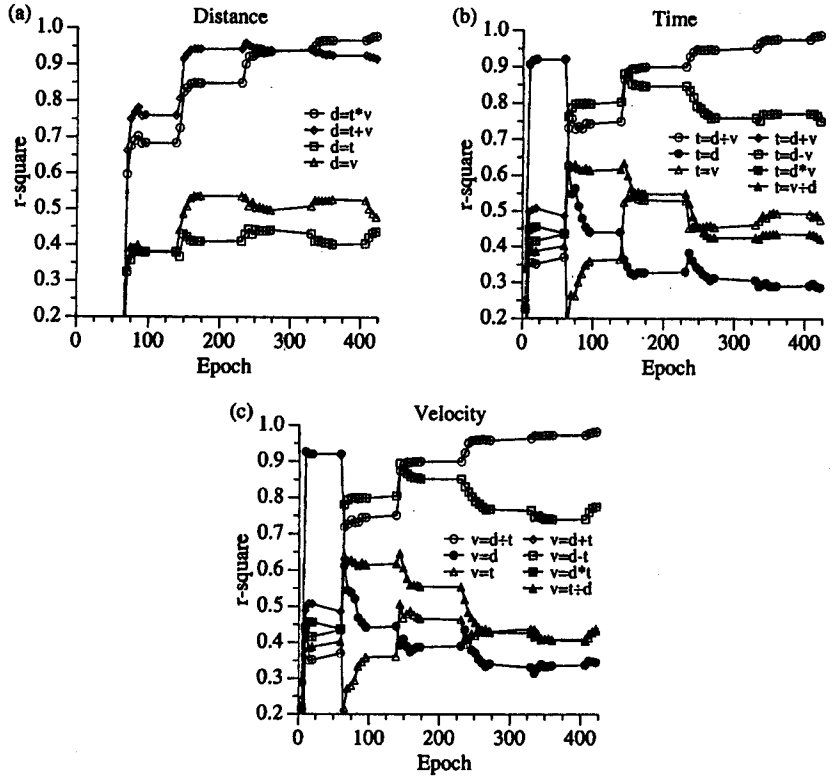

FIGURE 2 Amount of variance accounted for (r^2) in one nth network's responses to (a) distance, (b) time, and (c) velocity inference patterns by the set of possible rules. Only rules accounting for a significant amount of variance, $r^2 \geq .25$, $df = 24$, $p < .01$, at some point in training are shown. Gaps between data points correspond to hidden unit training. During this period, network output does not change and, thus, is not recorded.

thermometer, and gaussian encoding conditions (Figures 3a, 3b, 3c, and 3d), the following sequence was observed: (a) onset of time and velocity identity stages ($t = d$ and $v = d$), followed by (b) additive stages of all three concepts ($d = t + v$, $t = d - v$, and $v = d - t$), then (c) multiplicative stages of time and velocity ($t = d \div v$ and $v = d \div t$), and finally (d) the multiplicative stage of distance ($d = t \times v$). A similar sequence was found in the performance of the integer-context and distributed-integer networks (Figures 3e, 3f), the difference being that the distance additive stage started earlier in training. Only integer networks (Figure 3g) demonstrated a markedly different developmental course. Details of the development of each inference type are provided subsequently.

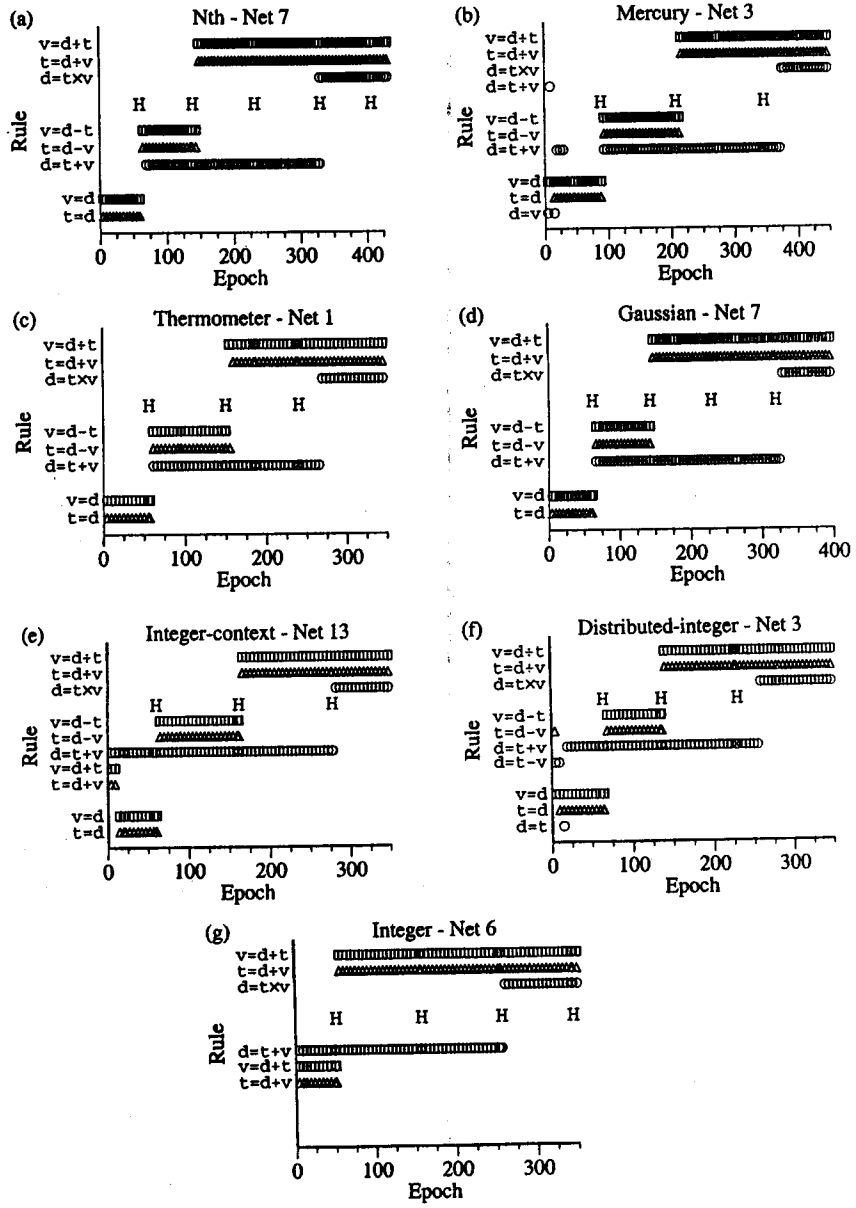

FIGURE 3 Identity, additive, and multiplicative rules by epoch that accounted for both the most variance and a significant amount of variance, $r^2 \geq .25$, $df = 24$, $p < .01$, in network responses during training for (a) nth, (b) mercury, (c) thermometer, (d) gaussian, (e) integer-context, (f) distributed-integer, and (g) integer encoding. Epoch of hidden unit recruitment indicated by H.

Distance inferences. All networks, regardless of input encoding technique, exhibited the same developmental course on distance inferences as the one observed by Wilkening (1982), that is, networks progressed from an additive stage ($d = t + v$) to a multiplicative stage ($d = t \times v$). There were some differences in terms of when the stages began and the amount of variance accounted for across conditions. In the integer-context, distributed-integer, and integer conditions, the additive stage began early in training, typically prior to the recruitment of a hidden unit. At onset, the additive rule accounted for more than 95% of the variance in the network's performance. In the other conditions, the onset of the additive stage occurred only after a hidden unit had been recruited and, at onset, the additive rule accounted for 60% to 71% of the variance. After a second hidden unit was recruited, the additive rule typically accounted for more than 90% of the variance in the networks' responses. Across all conditions, the multiplicative stage began after two to four hidden units had been added to the network. At onset, the multiplicative rule accounted for at least 92% of the variance in the networks' responses. Eventually, nearly all the variance was accounted for. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, the r^2 associated with the distance rules at onset, and the maximum r^2 attained by the rule defining the stage are reported in Table 2.

Occasionally, one of the possible distance rules (usually either $d = t$ or $d = v$) accounted for a significant amount of variance for one or two epochs at the beginning of training. For other networks (see Figures 3b and 3f), this period was marked by a vacillation between one or more rules. Finally, for some networks, one of the identity rules accounted for a significant amount of variance at the epoch just prior to the onset of the additive stage. Most of the networks that demonstrated these patterns were in the mercury, integer-context, and distributed-integer conditions. However, once the additive stage began, they developed as the other networks. Given the very brief appearance of these precocious rules, it is unclear what importance to give them. Moreover, for the majority of networks (58%, including the mercury, integer-context, and distributed-integer conditions; 83% excluding them), the onset of the additive stage was the first epoch at which any of the possible distance rules accounted for a significant ($p < .01$) amount of the variance in the networks' responses to the distance patterns. For all networks, the dominance by the additive rule continued until the onset of the multiplicative stage.

Time inferences. As can be seen in Figure 3, all networks (excluding those in the integer condition) followed a developmental course on time inferences similar to Wilkening's (1981) participants, progressing from an additive stage ($t = d - v$) to a multiplicative stage ($t = d \div v$). Network performance primarily differed in that networks first progressed through an identity stage ($t = d$), during which a maximum of more than 90% of the variance in the networks' responses to time problems was accounted for. The identity stage occurred prior to the recruitment of a hidden unit. The

TABLE 2
Mean Number of Hidden Units, Epochs of Training, and Variance Accounted for in Network Output at Onset of Distance Stages and Maximum Variance Accounted for During Stages

	$d = t + v$				$d = t \times v$			
	Onset				*Onset*			
Encoding	*Hidden*	*Epoch*	r^2	*Max* r^2	*Hidden*	*Epoch*	r^2	*Max* r^2
Nth								
M	1.00	72.50	0.71	0.94	3.10	255.40	0.92	0.9991
SD	0.00	3.76	0.07	0.05	0.91	76.78	0.05	0.0003
Mercury								
M	1.00	82.70	0.60	0.87	2.70	288.85	0.86	0.9990
SD	0.00	6.04	0.09	0.07	0.73	86.08	0.09	0.0004
Thermometer								
M	1.00	67.80	0.60	0.96	3.40	286.60	0.94	0.9991
SD	0.00	4.42	0.10	0.04	0.68	57.45	0.03	0.0002
Gaussian								
M	1.00	66.85	0.70	0.96	3.40	272.10	0.94	0.9992
SD	0.00	3.87	0.08	0.04	0.75	65.17	0.03	0.0004
Integer								
M	0.00	4.75	0.98	1.00	3.55	344.00	0.95	0.9992
SD	0.00	1.97	0.03	0.00	0.94	104.40	0.01	0.0004
Integer-context								
M	0.15	21.75	0.96	1.00	3.60	408.30	0.95	0.9990
SD	0.37	27.25	0.05	0.00	0.94	109.02	0.01	0.0003
Distributed-integer								
M	0.05	16.00	0.95	1.00	2.45	206.75	0.95	0.9989
SD	0.22	11.77	0.05	0.00	0.69	56.04	0.01	0.0004

Note. $n = 20$ for all network types. d = distance; t = time; v = velocity; max = maximum. All r^2 values are based on positive correlations.

additive and multiplicative stages typically began after the recruitment of the first and second hidden unit, respectively. At the onset of the additive stage, more than 74% of the variance in the network's responses to the time problems was accounted for by the $t = d - v$ rule. At the onset of the multiplicative stage, the $t = d \div v$ rule accounted for at least 76% of the variance, on average. By the end of training, nearly all the variance was accounted for. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, the r^2 associated with the time rules at onset, and the maximum r^2 attained by the rule defining the stage are reported in Table 3.

As with distance inferences, brief appearances of other time rules occurred prior to the onset of the first stage. Typically, there were one or two testing epochs prior to the identity stage in which the $t = d + v$ rule accounted for a significant amount of variance (see Figure 3e). This type of performance was mostly confined to the mercury and integer-context networks. Moreover, once the identity stage be-

TABLE 3
lean Number of Hidden Units, Epochs of Training, and Variance Accounted for in Network Output at Onset of Time Stages and Maximum Variance Accounted for During Stages

	$t = d$			$t = d - v$			$t = d \div v$			
	Onset			*Onset*			*Onset*			
icoding	*Epoch*	r^2	*Max* r^2	*Epoch*	r^2	*Max* r^2	*Hidden*	*Epoch*	r^2	*Max* r^2
h										
M	4.75	0.62	0.93	69.50	0.77	0.81	1.95	151.25	0.84	0.9995
SD	1.97	0.17	0.01	6.02	0.05	0.05	0.22	18.15	0.05	0.0002
ercury										
M	9.00	0.67	0.93	82.20	0.75	0.86	2.00	199.25	0.88	0.9992
SD	4.17	0.18	0.02	5.81	0.05	0.04	0.00	17.73	0.02	0.0004
ıermometer										
M	5.25	0.77	0.92	68.30	0.74	0.84	2.00	158.15	0.88	0.9993
SD	1.12	0.13	0.01	3.48	0.04	0.05	0.00	8.89	0.02	0.0003
ıussian										
M	4.75	0.67	0.92	68.10	0.76	0.82	2.00	150.15	0.88	0.9991
SD	1.12	0.14	0.01	4.14	0.05	0.05	0.00	8.81	0.02	0.0004
teger-context										
M	14.00	0.91	0.99	81.40	0.81	0.83	1.70	170.50	0.77	0.9993
SD	6.20	0.05	0.01	8.69	0.03	0.05	0.73	86.78	0.07	0.0003
stributed-integer										
M	6.25	0.97	1.00	65.10	0.85	0.87	2.15	170.20	0.76	0.9992
SD	2.75	0.04	0.00	6.66	0.02	0.05	0.59	55.16	0.10	0.0003

Note. $n = 20$ for all network types. d = distance; t = time; v = velocity; max = maximum. No hidden units were ruited prior to the onset of the identity stage ($t = d$). All networks recruited one hidden unit prior to the onset of the ditive stage ($t = d - v$). All r^2 values are based on positive correlations.

gan, these networks developed as in the other conditions. Across the *n*th, thermometer, gaussian, and distributed-integer conditions, for 90% of the networks, the only rules that were the best significant predictors of time inferences as the networks developed were the $t = d$, $t = d - v$, and $t = d \div v$ rules, in that order. Note that this was also true for 30% of mercury and integer-context networks.

The developmental course of integer networks (Figure 3g) involved only two stages—an additive stage prior to the recruitment of a hidden unit, followed by the multiplicative stage after the recruitment of the first hidden unit. The additive stage was characterized by a different rule ($t = d + v$). On average, the additive stage began at 4.75 epochs ($SD = 1.97$), with the additive rule accounting for 97.95% ($SD = 2.85\%$) of the variance. On average, the multiplicative stage began at 64.35 epochs ($SD = 9.65$), with the multiplicative rule accounting for 62.88% ($SD = 9.65\%$).

Velocity inferences. Ninety-nine of all 100 networks (excluding the integer condition) followed a progression from an identity stage ($v = d$), to an additive stage ($v = d - t$), and then to the multiplicative stage ($v = d \div t$). This is similar to Wilkening's (1981, 1982) results, with the exception that the networks attained the final multiplicative stage. The additive and multiplicative stages typically began after the recruitment of the first and second hidden unit, respectively. At onset, the identity rule accounted for more than 60% of the variance and quickly began to account for more than 90% of the variance. The additive rule accounted for at least 67% of the variance at onset of the additive stage, and it later accounted for more than 80% of the variance. Finally, at the onset of the multiplicative stage, the $v = d \div t$ rule accounted for more than 75% of the variance, on average. By the end of training, nearly all the variance was accounted for. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, the r^2 associated with the velocity rules at onset, and the maximum r^2 attained by the rule defining the stage are reported in Table 4.

TABLE 4
Mean Number of Hidden Units, Epochs of Training, and Variance Accounted for in Network Output at Onset of Velocity Stages and Maximum Variance Accounted for During Stages

	$v = d$			$v = d - t$			$v = d \div t$			
	Onset			*Onset*			*Onset*			
Encoding	*Epoch*	r^2	*Max* r^2	*Epoch*	r^2	*Max* r^2	*Hidden*	*Epoch*	r^2	*Max* r^2
Nth										
M	5.00	0.63	0.93	69.50	0.77	0.82	2.00	155.95	0.85	0.9993
SD	1.62	0.17	0.01	5.52	0.02	0.04	0.00	6.89	0.05	0.0003
Mercury										
M	7.25	0.66	0.93	82.20	0.76	0.87	2.00	198.25	0.88	0.9991
SD	3.43	0.17	0.02	5.90	0.05	0.04	0.00	17.08	0.03	0.0005
Thermometer										
M	4.00	0.70	0.92	68.05	0.74	0.82	2.00	157.90	0.88	0.9993
SD	2.05	0.17	0.01	4.39	0.04	0.05	0.00	9.03	0.03	0.0003
Gaussian										
M	4.75	0.62	0.92	67.60	0.74	0.82	2.10	157.45	0.88	0.9992
SD	1.12	0.13	0.01	4.19	0.05	0.06	0.31	27.21	0.04	0.0003
Integer-context										
M	13.75	0.90	0.99	81.42	0.80	0.82	1.65	168.10	0.76	0.9994
SD	7.23	0.04	0.01	8.93	0.03	0.04	0.67	86.69	0.06	0.0003
Distributed-integer										
M	6.50	0.97	1.00	65.10	0.85	0.87	2.20	174.90	0.76	0.9991
SD	2.86	0.04	0.00	6.66	0.02	0.05	0.52	51.05	0.09	0.0004

Note. $n = 20$ for all network types. One integer-context network skipped the additive stage. d = distance; t = time; v = velocity; max = maximum. No hidden units were recruited prior to the onset of the identity stage ($v = d$). All networks recruited one hidden unit prior to the onset of the additive stage ($v = d - t$). All r^2 values are based on positive correlations.

As was the case with respect to distance and time inferences, the earliest velocity inferences made by mercury and integer-context networks were sometimes best predicted by a rule other than the identity rule. Typically, the $v = d + t$ rule accounted for a significant amount of variance for two testing epochs prior to the onset of the identity stage. Across the *n*th, thermometer, gaussian, and distributed-integer conditions, for 90% of the networks, the only rules that were the best significant predictors of time inferences as the networks developed were the $v = d$, $v = d - t$, and $v = d \div t$ rules, respectively. Note that this was also true for 40% of mercury and integer-context networks.

As with time inferences, integer networks did not first progress through an identity stage but through an additive stage ($v = d + t$) prior to hidden unit recruitment. Then, after the first hidden unit was installed, they attained the normative multiplicative stage. On average, the additive stage began at 4.50 epochs (SD = 2.24) and accounted for 96.31% (SD = 3.92%) of the variance. The multiplicative stage began at 75.13 epochs (SD = 13.87) and accounted for 63.31% (SD = 7.66%) of the variance, on average.

Error Reduction

The mean amount of error at the beginning and ending of the identity, additive, and multiplicative stages is shown in Figures 4, 5, and 6, respectively. Error was measured as the sum of squared differences between target and actual output across patterns. Each stage was associated with successive error reduction. Approximately 20% to 35% of the error that existed at the onset of the time and velocity identity

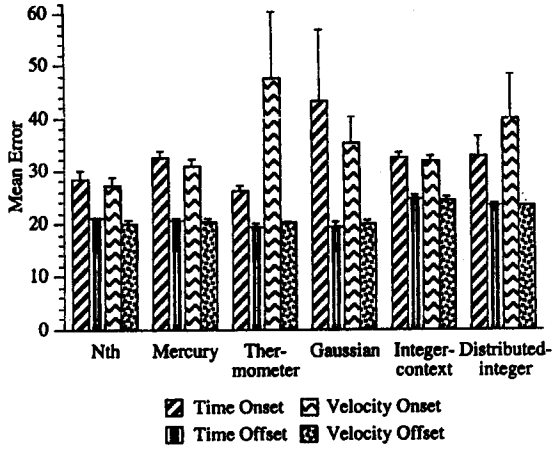

FIGURE 4 Mean error at the beginning and end of time and velocity identity stages. Error bars represent standard error of the mean.

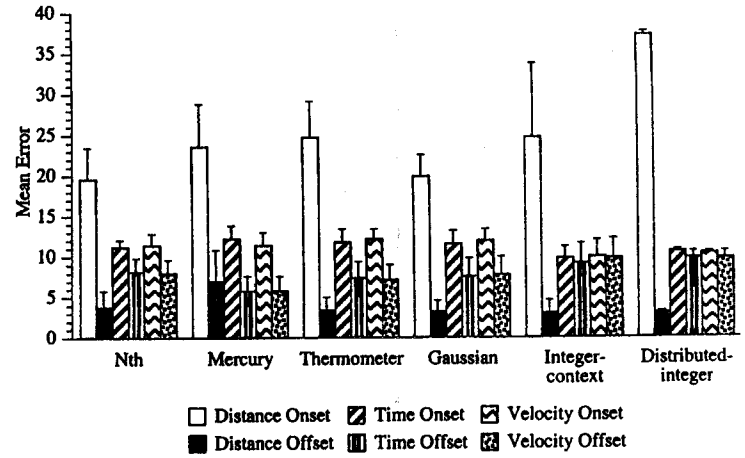

FIGURE 5 Mean error at the beginning and end of distance, time, and velocity additive stages. Error bars represent standard error of the mean.

stages was reduced over the identity stages. Between 25% and 50% of the error was reduced over the time ($t = d - v$) and velocity ($v = d - t$) additive stages, except in the case of integer-context and distributed-integer networks, in which about 5% of the error was reduced. For integer networks, error reduction over the time ($t = d + v$) and velocity ($v = d + t$) additive stages was roughly 15%. Approximately 70% to 90% of the error that existed at the onset of the distance additive stage was reduced by the end of the stage across all networks. Finally, more than 98% of the error that existed at the beginning of the multiplicative stages was reduced by the end of training. Thus, learning was continuous across the stages in all developmental sequences.

Hidden Unit Recruitment and Stage Onset

As can be seen in Figure 3, transition from identity to additive and additive to multiplicative stages typically occurred quickly after the recruitment of a hidden unit. So-called Hinton diagrams were drawn (Figures 7 and 8) to understand the nature of the relation of hidden unit recruitment to stage onset.

In Hinton diagrams, the magnitude and sign of weights from sending units (input and hidden) to receiving units (hidden and output) are indicated by the size and color (white for positive and black for negative) of squares drawn in a row for each receiving unit. The numbers above the squares indicate the sending unit. For integer and integer-context networks, Squares 1, 2, and 3 represent the weights from the distance, time, and velocity input units, respectively. For integer-context networks, Squares 4, 5, and 6 represent the context units. For distributed-integer networks, Squares 1–2, 3–4, and 5–6 represent the weights from the distance, time, and velocity input

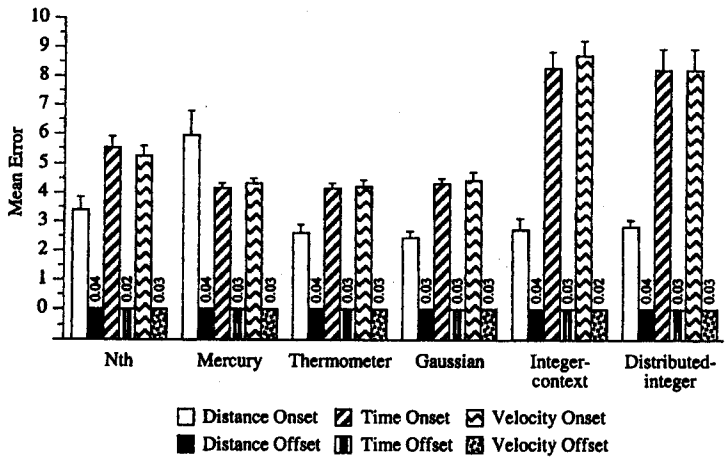

FIGURE 6 Mean error at the beginning of distance, time, and velocity multiplicative stages and at the end of training. Numbers are included for mean error values close to zero. Error bars represent standard error of the mean. (Error axis extends below zero only for illustration purposes.)

groups, respectively. For *n*th and mercury networks, Squares 1–5, 6–10, and 11–15 represent the weights from the distance, time, and velocity input groups, respectively. Finally, for thermometer and gaussian networks, Squares 1–7, 8–14, and 15–21 represent the weights from the distance, time, and velocity input groups. For all network types, Square 0 signifies the bias unit, whereas the last square in the output row is for the weight from the hidden unit. When more than one hidden unit is depicted, the weights from any previous hidden units are depicted after the last of the input group. For example, in Figure 8a, the square numbered *4* of the second hidden unit row represents the weight from the first hidden unit to the second.

For the majority of networks (88%, excluding integer networks), a clear pattern of weights connecting the first hidden unit to the input layer was observed. A Hinton diagram of a typical network that exhibited this pattern is presented in Figure 7 for each of the encoding types for which this pattern was common. As can be seen, the first hidden unit distinguishes distance input units from time and velocity input units, that is, the weights from time and velocity input units generally have the opposite sign from the weights from the distance input units.[4] When the hidden

[4]This same pattern holds for distributed-integer networks but is complicated by the use of two units to represent a dimensional value. As depicted in Figure 7b, the weights for the distance units (Squares 1 and 2) are both positive. In contrast, although one time and velocity weight (Squares 4 and 6, respectively) is positive, the other weight (Squares 3 and 5, respectively) is larger and negative. Thus, the overall effect of either time or velocity input is negative and opposite to that of distance input.

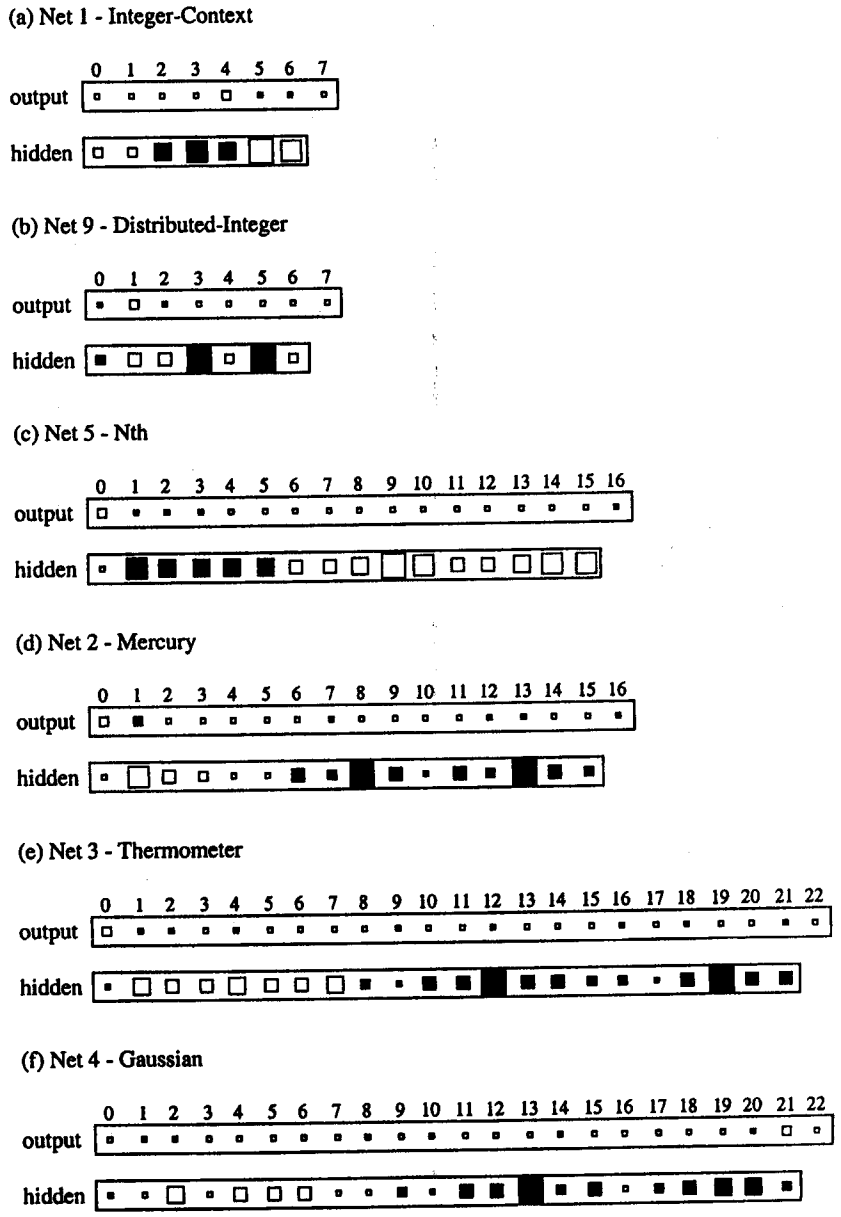

FIGURE 7 Hinton diagrams showing relative size (square size) and direction (white = positive, black = negative) of weights from input layer to first hidden unit and output unit for (a) integer-context, (b) distributed-integer, (c) nth, (d) mercury, (e) thermometer, and (f) gaussian encoding.

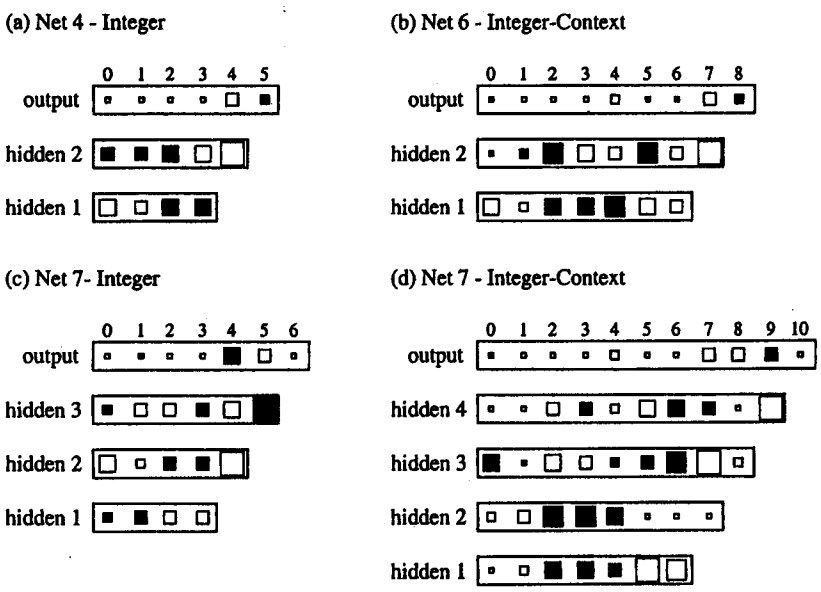

FIGURE 8 Hinton diagrams showing relative size (square size) and direction (white = positive; black = negative) of weights from input layer to hidden units and output unit for (a, c) integer and (b, d) integer-context encoding.

unit is activated by the presentation of a distance inference pattern, information from the time and velocity input groups augments each other because the direction of their weights are the same. Conversely, because the distance weights have opposite signs from the time and velocity weights, when the hidden unit is activated by the presentation of a time or velocity inference pattern, input from one input group counters the effects of input from the other, and vice versa. Thus, the additive rule $d = t + v$ stems from the summing effects of input from time and velocity input groups caused by same-sign weights to the hidden unit. Alternatively, the additive rules $t = d - v$ and $v = d - t$ stem from summing opposite-sign weights of time and velocity input groups and distance input groups.

Unfortunately, a clear pattern did not emerge with respect to transition to the multiplicative stages of time and velocity. However, for integer and integer-context networks, a consistent pattern was observed for the transition to the distance multiplicative stage. For 73% of the networks, the multiplicative stage only emerged after a hidden unit was recruited that received opposite-sign weights from time and velocity input groups. Hinton diagrams of two networks using integer and integer-context encoding are presented in Figure 8. As can be seen in Figures 8a and 8b, the second hidden unit has opposite-sign weights (Squares 2 and 3). The multiplicative stage was subsequently observed. For the networks depicted in Figures 8c and 8d, the multiplicative stage was observed after the installation of the third and fourth hidden units, respectively. Again, it was at this point that the opposite-sign weights for time and velocity input groups were first used by a hidden unit.

Generalization

The sum of squared error for the 57 training (19 distance, 19 time, and 19 velocity) and 18 testing (6 distance, 6 time, and 6 velocity) patterns was recorded across epochs for the five networks per encoding condition that were run to test generalization. The respective error scores were then scaled by dividing by the total number of patterns in the training and testing sets to obtain a mean sum of squared error. Plots of the mean sum of squared error over training and testing patterns by epoch of typical individual networks are illustrated in Figure 9. To choose representative networks, the absolute difference between test and training error at each testing epoch was calculated for a given network. The mean absolute difference across epochs was then obtained for each network. The networks plotted are those with the median score.

In general, the curve of the test errors mimics the curve of the training errors, suggesting that the networks, regardless of input encoding type, are generalizing what has been learned from the training patterns to the testing patterns. One exception, regarding *n*th unit coding, can be seen in Figure 9a. Although the curve of the testing error mirrors that of the training error up to approximately 225 epochs, the two error curves then diverge. This may be the result of overtraining in networks with *n*th encoding.

Another test of generalization would be to determine which rule accounted for the most variance in the networks' responses to the testing patterns. Unfortunately, due to the limited number of total patterns, the size of the testing set was constrained. Given that there were only six distance, time, and velocity testing patterns, accurate assessment of the correlation between various rules and the outputs generated by the testing patterns was not possible for the test patterns. Therefore, as an alternative, rule assessment was conducted using the entire set of patterns (i.e., training and testing patterns).

In general, training on a subset of the entire set of patterns appears to have very little effect on the progression of stages. Specifically, 34 of the 35 networks followed the same distance progression from the additive stage ($d = t + v$) to the multiplicative stage ($d = t \times v$). Four of the networks regressed briefly to the additive stage after attaining the multiplicative stage. One network in the gaussian condition attained only the additive stage.

Twenty of 29 networks (excluding integer) followed the typical time and velocity progression from identity stages ($t = d$ and $v = d$), to additive stages ($t = d - v$ and $v = d - t$), and then to multiplicative stages ($t = d \div v$ and $v = d \div t$). Of the remaining

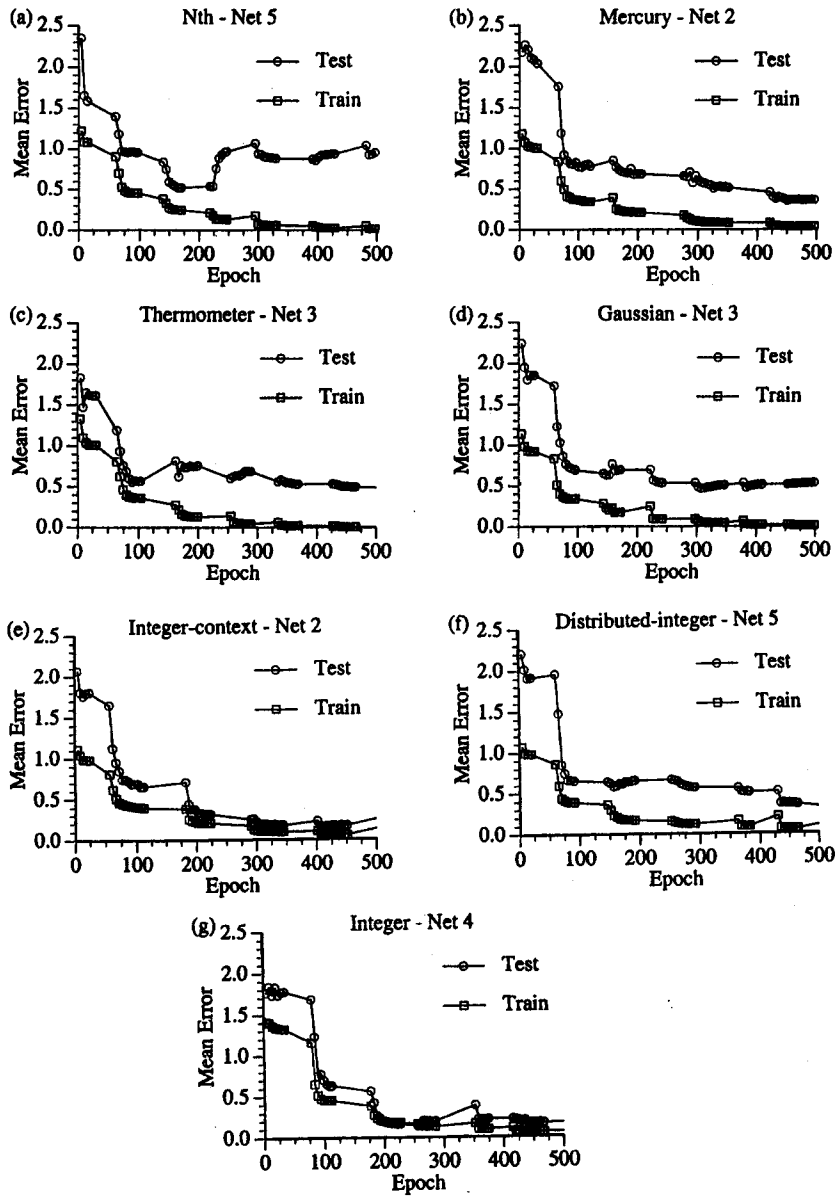

FIGURE 9 Mean sum of squared error on training and testing sets by epoch for (a) nth, (b) mercury, (c) thermometer, (d) gaussian, (e) integer-content, (f) distributed-integer, and (g) integer encoding. Gaps between data points correspond to hidden unit training. During this period, error on patterns does not change and, thus, is not recorded.

networks, five (one gaussian, two mercury, and two thermometer networks) differed only in that they skipped either the time or velocity additive stage, and one distributed-integer skipped the time identity stage. For the majority of the 30 networks, the multiplicative stage was stable, continuing until the end of training. However, seven networks regressed to either one or both additive stages after the multiplicative stage. For three of these networks, the regression was brief. For three networks, the regression was permanent. Finally, the progression of the integer networks differed in the same manner as it did for integer networks that were trained on all patterns. All five networks progressed from the additive stage defined by $t = d + v$ to the correct multiplicative stage. All but one progressed from the additive stage defined by $v = d + t$ to the correct multiplicative stage.

DISCUSSION

Stages in Development

In general, the stages that characterized network performance and the order in which the stages emerged were consistent with those observed in children and adults (Wilkening, 1981, 1982). The first distance stage to emerge was typically defined by the additive rule $d = t + v$. Networks then progressed to performance characterized by the normative multiplicative rule $d = t \times v$. With respect to Wilkening's results, this developmental sequence is identical to the one found in the follow-up study (Wilkening, 1982) in which memory demands of the distance task were increased to prevent young children from using an eye-movement strategy.

The developmental course of time and velocity demonstrated by networks was also comparable with Wilkening's observations. Early performance was characterized by identity rules ($t = d$ and $v = d$) in which networks behaved as if time and velocity inferences were based solely on distance information. Although 5-year-olds in Wilkening's studies were not classified as using the time identity rule, they were found to use the velocity identity rule. Later in training, additive stages ($t = d - v$ and $v = d - t$) emerged in which networks behaved as if time (velocity) inferences were based on subtracting distance information from velocity (time) information. The same time and velocity additive stages were observed by Wilkening (1981). Finally, networks attained multiplicative stages ($t = d \div v$ and $v = d \div t$). Adult participants in both Wilkening's (1981) original study and the follow-up study (Wilkening, 1982), in which he attempted to decrease the memory demands of the velocity task, were found to use the normative rule for time but not for velocity. Thus, in general, networks progressed through the same distance, time, and velocity stages as those observed by Wilkening, with the exception of an early time identity stage and the attainment of a velocity multiplicative stage.

Only integer networks progressed through a qualitatively different development marked by early time and velocity additive stages ($t = d + v$ and $v = d + t$) instead of identity stages, skipping the usual additive stages and, thus, earlier attainment of the time and velocity multiplicative stages compared to the other network types. Why did the performance of networks using integer coding differ? In contrast to other types of networks, both integer and integer-context networks begin training with full-blown knowledge of the ordinal relations of the input, that is, even with initially random weights, an input of 4, for example, has exactly twice the influence (positive or negative) on the outcome than an input of 2.[5] Although both integer and integer-context networks use the same representation for dimensional values, integer-context networks converge more slowly on normative performance and do so after developing in a psychologically realistic manner. The fact that context units give rise to slower convergence may seem counterintuitive. However, when a network begins training, it has to learn not only that the context units provide contextual information but also how to use this information. The relatively slower convergence and realistic development of integer-context networks may be connected. An analogous finding was reported by Schmidt and Shultz (1992) concerning the performance of back-propagation networks on balance scale problems. They argued that any training manipulations that slowed convergence increased the psychological realism of network performance.

Taken together, psychologically realistic performance was demonstrated by networks that either had to construct, to some extent, numerical scales of the input dimensions or, in the case in which the scales were provided, had to contend with additional information that slowed convergence on the normative rules. Although what kind of scaling children initially use in distance, time, and velocity integration remains an open question at this point, the similarity of the results across network types attests to their robustness and strengthens our conclusions.

Domain-General and Domain-Specific Constraints

Because we were able to control the task demands of the learning environment, the simulations show more developmental consistency across concepts than did

[5]All input coding schemes provide number cardinality, either by using different units to represent different input values—as in *n*th, mercury, thermometer, and gaussian encodings—or by assigning different activations on the same unit, as in integer, integer-context, and distributed-integer encoding. When different units are used to represent unique input values, the networks must learn the ordinal relations by adjusting the initial random weights accordingly. When the input values are distributed, the networks will have ordinal knowledge at the beginning of training. For distributed-integer coding, this knowledge most likely will be based on an interval scale. Thus, distributed-integer networks are likely to begin training knowing that an input of 4, for example, has greater influence (positive or negative) than an input of 2 but not that it has twice the influence.

Wilkening's results. Identity, additive, and multiplicative stages emerge as a result of the interaction between domain-general constraints of cascade-correlation and domain-specific constraints of the task at hand. *Domain-general* constraints are those that are common in all cascade-correlation models. Examples include the summing of input values, the adjustment of connection strengths to reduce error between target and actual output, and the recruitment of hidden units. *Domain-specific* constraints are those that are unique to the current simulations. Examples include the type and manner of input and output encoding and the regularities in the training patterns.

Identity stages emerge due to a combination of the limited processing ability of the initial perceptron architecture (a domain-general constraint) and the fact that the network is performing all three inference tasks (a domain-specific constraint). A network without hidden units (i.e., a perceptron) cannot resolve the error associated with the relations that exist between distance, time, and velocity. Given that time and velocity are both directly related to distance but inversely related to each other, one set of input-to-output weights cannot capture all these relations. On the other hand, because distance is directly related to both time and velocity, input-to-output weights are sufficient to capture both direct functional relations.[6] As such, identity rules ($t = d$ and $v = d$) arise from the networks' ability to represent the direct relation between distance and time and distance and velocity but not the inverse relation between time and velocity. This is consonant with evidence that children understand the direct relations before the inverse relations (Acredolo et al., 1984). Note that if networks were trained only on the velocity task, only one relation would exist per input (direct for distance; inverse for time). In that case, there would be no reason to expect identity stages. In all likelihood, the initial perceptron architecture would be capable of additive stages.

Thus, our simulations offer a unique explanation of identity stages based on the reversible structure of the concepts and the functional relations between the dimensions, that is, young children may use identity rules because they understand that each concept is related to the other two but cannot successfully represent all the relations that exist between the concepts. In other words, their reliance on distance when inferring how fast an object traveled does not simply reflect a lack of

[6]If we had used three different output units, one each for distance, time, and velocity integration, the networks would be faced with a task different from that of these networks. Networks with three output units would begin with a priori knowledge that time, for example, can have a relation to the outcome of a distance integration different from that of a velocity integration. Thus, the task would be to find the appropriate weights that capture the different relations. In our networks, the task also includes discovering that time can have a different relation to distance than it does to velocity. Which assumption is correct with respect to children is an open question at this point. Our simulations show that this a priori knowledge is not required.

understanding of how time relates to velocity. Instead, it reflects a realization that changes in time affect the outcome differently depending on whether distance or velocity inferences are being made. However, they lack the ability to represent time successfully in both a direct functional relation (as is needed for distance inferences) and an inverse functional relation (as is needed for velocity inferences). The fundamental point is that even young children may not approach distance, time, and velocity problems as a set of unrelated tasks, but, rather, what they know about one problem influences how they perform on another.[7]

How do networks overcome identity stages? After the first hidden unit is installed, simplistic encodings of both the direct and inverse relations of time and velocity are possible. Analysis of relative size and sign of weights from input units to the first hidden unit revealed that weights from the time and velocity input banks were of the same sign and opposite in sign to weights from the distance input bank. Therefore, when a distance pattern was presented, time and velocity input augmented each other, giving rise to inferences that correlated best with the additive rule $d = t + v$. When a time (*velocity*) pattern was presented, distance input was counteracted by velocity (*time*) input, giving rise to time (*velocity*) inferences that correlated best with the subtraction rule $t = d - v$ (or $v = d - t$). Thus, the recruitment of a hidden unit enables two different relations between the time (*velocity*) input bank and the output depending on whether a distance or velocity (*time*) problem is being presented.

The additive stages of all three concepts eventually were replaced by multiplicative stages. Because of network complexities, Hinton analysis of the second and third hidden units were less revealing. However, given the relatively abrupt transition after the installation of these hidden units to multiplicative stages, the need for increased nonlinearity seems evident. Typically, the time and velocity stage emerged first, followed by the distance multiplicative stage. One reason why the distance additive stage may have lasted longer than either the time or velocity additive stages was that a larger proportion of error was reduced during the distance additive stage than during either of the other two. This, in turn, delayed the onset of the distance multiplicative stage. In other words, the distance additive rule provides a good approximation for distance inferences. It may be that, for people, use of an additive rule persists as a heuristic approach that is generally good enough.

[7]Our implementation assumes that children do not begin thinking about distance, time, and velocity as different integration tasks but rather as different instances involving integration based on the same physical dimensions. Given this conceptual framework, it is incorrect to think of the output unit as some sort of composite concept of distance, time, and velocity. The output is a measure of the magnitude of the integration that was performed. A network's conceptual knowledge of distance, time, and velocity resides in the connections (input-to-output and, in the case in which hidden units have been recruited, input-to-hidden and hidden-to-output connections).

A Connectionist Perspective of Structures and Change

Our simulations suggest that developmental transitions in children result from incremental learning and increases in nonlinear representational abilities. These are implemented in cascade-correlation networks by weight adjustment and hidden unit recruitment, respectively. Although increasingly complex rules can characterize the various stages of distance, time, and velocity development, from our connectionist perspective, the rules are emergent epi-phenomena rather than computational mechanisms, as in the symbolic rule-based approach.

There are several key features that characterize the knowledge structures we are proposing: (a) Representation and processing are based on graded stimulus input from all relevant dimensions throughout development, (b) developmental transition is brought on by qualitative changes in representation, and (c) understanding the reversible nature of the three concepts is an emergent property of the representational system. We deal with each of these considerations in turn.

Representation and processing. Young children's behavior on compensation tasks often appears to be based on a single dimension (Siegler, 1981). Because rule-based representations typically do not rely on graded stimulus input, it often has been assumed that the earliest representations do not include all the perceptually available dimensions. For example, Siegler (1976) proposed a series of binary decision rules to account for children's performance on the balance scale. The first rule does not include distance information but uses only weight information. More recently, Schmidt and Ling (1996) biased their rule learning model to process weight before distance information. However, there is a difficulty with assuming that certain information is ignored. How can the ignored dimension become relevant if it is ignored?

Connectionist researchers have taken a different approach. Rather than temporarily omitting one dimension, they have assumed that the underlying representations include graded connections from all the relevant dimensions but that either the learning environment (e.g., McClelland, 1989; Shultz et al., 1994) or the initial connections of the network (Shultz et al., 1995) limit the influence of one dimension in favor of another dimension.

In this study, the domain-specific constraint of having one unified representation of all three concepts was a sufficient constraint to capture, for example, early velocity identity rule use by children. There was no need to specify a priori that the representations involve only distance or that distance was somehow more salient than time information. Distance and time information was always included in the form of graded stimulus input. Furthermore, in this case, there was no need to bias either the learning environment or the connection weights.

The inclusion of both defining dimensions as input to the earliest representations is not necessarily at odds with the cognitive algebra proposed by Wilkening

(1981) because information integration theory also postulates graded inputs. Therefore, the velocity identity rule, for example, can be viewed as a special form of the general additive rule in which the subjective value of the time stimulus information is at or near zero.

The assessment methodology used by Wilkening and the one used in the present simulations may not be sufficiently fine-tuned to capture the subtle effect of time information on early velocity inferences. There is evidence that young children represent and process time information when making velocity inferences when metric responses are not required. Acredolo et al. (1984) asked children to imagine that two animals had fled a farmer's field in a scenario similar to Wilkening's. However, rather than requiring a metric response, the children were asked to judge the likelihood of possible outcomes. Although Acredolo et al. did not discuss their results in terms of the velocity identity rule, the pattern of responses across problems can be interpreted as providing evidence for its use in all but one case. When children were told the two animals ran the same distance but one ran for more time, the most frequent error was to say that the one that ran for more time ran faster.

The structures that we propose are similar to those suggested by Wilkening in another respect. Both allow for nonnormative integration. Whereas Piaget believed children only integrated the dimensions at the age of mastery and, thus, only allowed for correct integration, Wilkening's research revealed that nonnormative integration (i.e., $d = t + v$, $t = d - v$, and $v = d - t$) occurred earlier in development. As with the use of identity rules, nonnormative integration was an emergent property of networks' representations. It is not clear how symbolic rule-based approaches would account for the development of these nonnormative rules. Simply stipulating that they occur would fail to show how and why they would arise. Capturing the development of nonnormative rules would pose an interesting challenge to rule-based modelers.

However, the two types of structures are clearly different. Although networks were capable of achieving knowledge states that have been assumed to be represented by algebraic rules (i.e., identity, additive, and multiplicative rules), the processing underlying performance is different. During identity stages, simple summation of the graded stimulus input enabled performance consistent with identity rule use. Later, an additional process that passed the summed input through an activation function enabled performance that could be characterized by additive and multiplicative integration. Both processes allowed networks to perform "as if" following algebraic rules. This also may be the case for participants in Wilkening's experiments. For example, it seems unlikely that the 5-year-olds in Wilkening's (1981) experiment knew that distance inferences are based on multiplying time and velocity information. Wilkening himself has discussed the "as if" nature of the integration, that is, children perform as if they were multiplying the dimensions.

Developmental change. What causes developmental change? In terms of the network performance reported here, the answer is clear—weight adjustment and hidden unit recruitment. In this study, weight adjustment seemed generally important for within-stage change, whereas hidden unit recruitment was primarily involved in transitions between qualitatively different stages in performance.

The large reduction of error within stages suggests that, although network performance was stagelike in that long periods of training resulted in the same classification of responses, learning was continuous during each stage. The conception of a stage as a dynamic rather than a static period seems problematic for rule-based approaches that do not rely on graded input. For example, if a child has an explicit rule for making time inferences that involves focusing on distance information alone, it is unclear how improvement beyond the correct application of the rule within a stage might occur, that is, once the child can apply the rule correctly, there would be no more improvement. Thus, the child's accuracy would level off until the onset of a new stage. In contrast, error reduction across, say, the velocity identity stage demonstrated by the networks suggests that improvements in the accuracy of velocity inferences may occur even though the responses remain characterized by the velocity identity rule.

The progression from identity to additive and then multiplicative stages represents qualitative restructuring of knowledge representations. The involvement of hidden unit recruitment in this type of change is clear. Transition from the identity to additive stages of time and velocity followed the recruitment of the first hidden unit. Similarly, transition to the time and velocity multiplicative stages followed the recruitment of the second unit. Finally, transition from the distance additive stage to the multiplicative stage typically followed the recruitment of either the third or fourth hidden unit.

Some researchers have argued that weight adjustment alone is capable of stage transition (McClelland, 1989; Plunkett & Sinha, 1992). However, recent attempts to model distance, time, and velocity development with static back-propagation networks failed to capture the entire developmental course. Using a variety of architectures and learning parameter values, Buckingham and Shultz (1996) were unable to find a suitable static model. Simulations were run 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, three levels of learning rate and momentum were investigated. The input and output representations were the same as for the *n*th networks reported in this article, as was the training corpus. Static back-propagation networks were either too weak, capturing only early stages, or too powerful, skipping the intermediate additive stages. Therefore, growth in computational power is necessary to capture the full range of stages.

Researchers working within the framework of information integration theory have had difficulty formulating a precise mechanism that would account for these

changes. In contrast, the use of cascade-correlation provides a mechanistic account of transition—weight adjustment and hidden unit recruitment. But what does this tell us about changes in the developing child? Our models are not proposed as neural models, although they are based on principles generally consistent with brain functioning. Rather, they are meant to determine what constraints, both domain-general and -specific, are sufficient for explaining the regularities that occur during development. Our view is that the mechanisms of cascade-correlation correspond to analogous mechanisms available to children. Weight adjustment can be viewed as a quantitative process that involves differential weighting of information to increase the similarity between the actual outcome and that expected by the child. In contrast, hidden unit recruitment is analogous to a qualitative restructuring of knowledge based on what is currently known about the problem at hand and consideration of the nature of one's current errors. As discussed earlier, these mechanisms of change can be mapped onto existing psychological constructs, such as Piaget's notions of assimilation and accommodation or Karmiloff-Smith's (1992) representational redescription. Alternatively, the model can be interpreted on its own and used as a tool for guiding thinking about various developmental phenomena. The mapping naturally will get more detailed as research progresses. It is profoundly difficult to identify all the features of detailed knowledge representation and processing in children. That is what makes modeling so appealing—you can at least explore detailed theoretical commitments in one domain (i.e., in the model).

Reversibility as an emergent property. Imagine a child watching a car travel down a street and wondering about how long it will take to get to the end of the block traveling at its current speed. Now imagine that the same child watches a second car traveling down the street to the end of the block taking 5 sec and then wondering how fast it would have to travel to get there in 3 sec. It seems reasonable to assume that although the first instance involves a time inference and the second involves a velocity inference, the distance that both cars traveled (to the end of the block) is represented as the same physical dimension, not only in terms of magnitude but in terms of the concept of distance.

In our simulations, this assumption was implemented as the domain-specific constraint of having one unified representation for drawing inferences about all three concepts. An interesting ramification of this constraint is that it allows for knowledge of the reversible nature of the concepts to become an emergent property of the representational system. Although most theorists, at least implicitly, assume that mastery of distance, time, and velocity integration involves an understanding of the reversible nature of the problem (why else would one test children's understanding on all three tasks?), how this occurs has not been explicitly addressed. The unified representation we propose suggests that what children know about time as it relates to distance and velocity influences what they know about distance as it relates to time and velocity, and so forth.

Empirical Predictions

In addition to covering and explaining existing phenomena, our modeling efforts are geared to providing empirical predictions. Perhaps the most important insight to be gained from our models is that what children know about one concept may influence how they think about the others. In particular, knowledge of how two concepts are related may influence and be influenced by knowledge of how to integrate these concepts to predict a third.

The simulations suggest that when all else is held constant, identity, additive, and multiplicative stages across concepts emerge at similar times, reflecting the processing capacities of the computational system and the constraints of the task environment. In light of this, the simulations make a number of predictions. First, the model predicts an initial identity stage in which time is judged as proportional to distance. Note that children have been found to judge time solely on the basis of distance information in choice task experiments (e.g., Acredolo & Schmid, 1981; Piaget, 1946/1969, 1946/1970). Although findings from choice task experiments do not provide validity for our quantitative inference model, they do show that children sometimes base time judgments solely on distance information.

Second, the model predicts the attainment of a terminal multiplicative stage in which velocity is correctly inferred as the ratio of distance to time. Recall that Wilkening's experiment left open the question of whether adults were using the normative multiplicative or additive rule. We believe the inability of Wilkening's adult participants to integrate time and distance information correctly was due either to extra memory demands or an inappropriate response scale, as Wilkening suspected.

Third, the model suggests parallel development of velocity and time inferences. For example, given that young children were found to make time inferences in an additive manner (Wilkening, 1981, 1982), the simulations predict that same-age children also should integrate distance and time information additively when making velocity inferences. Again this prediction is not at odds with Wilkening's (1981) hypothesis that his velocity task may have been more difficult than his time task. However, it does suggest that his manipulation (Wilkening, 1982) to lessen the memory demands of the velocity task may have been ineffectual.

Fourth, the model predicts relatively late acquisition of the final stage in which distance is the product of velocity and time. If the three tasks were more equivalent, the distance multiplicative stage would emerge after the time and velocity multiplicative stages. Because Wilkening did not study 10-year-olds' performance

when an eye-movement strategy was not possible, it would be necessary to reexamine 10-year-olds under this condition.[8]

The simulations also provide us with two predictions that are unrelated to the issue of the equality of tasks demands. First, recall that neither of the identity rules ($d = t$ or $d = v$) nor the additive rule ($d = t + v$) captured the initial performance of the networks on distance inferences. As implied, this was likely because the relation of time and velocity input to output error was obscure. It seems reasonable to predict that if the relation of time (*velocity*) to distance inferences was made more salient, then both networks and children would perform as if using the identity rule, $d = t$ ($d = v$). Perceptual salience has been proposed as a possible explanation of children's poor performance on time problems (Levin et al., 1980). Thus, it is possible that salience plays a role in distance inferences as well. Second, on a more general note, the simulations suggest that viewing stages as static rather than dynamic periods in development is incorrect. Within stages, we would expect to observe steadily improving performance in terms of increasingly precise inferences that need not be characterized by the normative rules of distance, time, and velocity integration.

CONCLUSIONS

We have argued that the stage progressions observed in network performance result from the domain-general constraints inherent in a generative algorithm and the domain-specific constraint of having one network perform all three related tasks. The main theoretical implications are that children may process distance, time, and velocity information in parallel using a domain-general learning algorithm that allows for increased complexity in knowledge representations as the child's capacity for problem solving increases. The domain-specific constraint of making inferences on the three problem types determines the type and progression of knowledge representations. Early identity rules may result from an inability to conceptualize both the direct and inverse relations of time and velocity rather than from the child ignoring time or velocity information. Increases in the child's capacity may enable simplistic additive representations and then more complex multiplicative representations.

The child is considered as an active participant in his or her environment in that he or she is learning from experience continually. However, learning itself may not always be sufficient for qualitative changes in the knowledge representations. Often, such changes require increases in processing capacity.

[8]These four predictions are currently being investigated by the authors using an experimental design that minimizes the differences in task demands by using similar response scales and stimulus presentation methods across the distance, time, and velocity tasks.

The success of these simulations in capturing the development of distance, time, and velocity integration rules is encouraging because it suggests that other findings of researchers working within the framework of information integration (N. H. Anderson, 1974) also may be captured by connectionist simulations. In general, developmental transitions from simpler additive rules to more complex multiplicative integration rules have been observed in children's performance on a number of other compensation tasks. The advantage of connectionist simulations is that they provide both precise knowledge representation of integration rules and a mechanistic account of how and why development proceeds from simpler to more complex integration rules.

Within the study of cognitive developmental phenomena, this research extends the applicability of cascade-correlation to the acquisition of distance, time, and velocity concepts. A number of insights and predictions have come from this work, supporting cascade-correlation as a promising tool of investigation into cognitive development.

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 Quebec and an operating grant from the Natural Sciences and Engineering Research Council of Canada. Thanks to Sylvain Sirois for comments on an earlier version of the article.

REFERENCES

Acredolo, C., Adams, A., & Schmid, J. (1984). On the understanding of the relationships between speed, duration, and distance. *Child Development, 55,* 2151–2159.

Acredolo, C., O'Conner, J., Banks, L., & Horobin, K. (1989). Children's ability to make probability estimates: Skills revealed through application of Anderson's functional measurement methodology. *Child Development, 60,* 933–945.

Acredolo, C., & Schmid, J. (1981). The understanding of relative speeds, distances, and durations of movements. *Developmental Psychology, 17,* 490–493.

Alpaydin, E. (1991). *GAL: Networks that grow when they learn and shrink when they forget* (Tech. Rep. No. 91–032). Berkeley, CA: International Computer Science Institute.

Anderson, J. A. (1990, June). Data representation in neural networks. *AI Expert,* 30–37.

Anderson, N. H. (1974). Information integration theory: A brief survey. In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.), *Contemporary developments in mathematical psychology: Volume 2. Measurement, psychophysics, and neural information processing* (pp. 236–305). San Francisco: Freeman.

Anderson, N. H. (1991). Functional memory in person cognition. In N. H. Anderson (Ed.), *Contributions to information integration theory: Volume 1. Cognition* (pp. 1–55). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, N. H., & Cuneo, D. O. (1977). *The height + width rule in children's judgments of quantity* (Tech. Rep. No. 69). San Diego: University of California, Center for Human Information Reasoning.

Avons, S. E., & Thomas, S. (1990). Exploring the development of area judgements using a PEST technique. *British Journal of Developmental Psychology, 8,* 51–63.

Ballard, D. H. (1987). Interpolation coding: A representation for numbers in neural models. *Biological Cybernetics, 57,* 389–402.

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, England: Blackwell.

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

Buckingham, D., & Shultz, T. R. (1996). Computational power and realistic cognitive development. In G. W. Cottrell (Ed.), *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 507–511). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Churchland, P. M. (1990). Cognitive activity in artificial neural networks. In D. N. Osherson & E. E. Smith (Eds.), *An invitation to cognitive science: Volume 3. Thinking* (pp. 199–227). Cambridge, MA: MIT Press.

Crépault, J. (1977). Organisation et genèse des relations temps, espace et vitesse [Organization and genesis of time, space, and velocity relations]. In P. Fraisse (Ed.), Du temps biologique au temps Psychologique. *Symposium de l'association de psychologie scientifique de langue français* (pp. 227–253). Paris: Presses Universitaires de France.

Crépault, J. (1978). Le raisonnement cinématique chez le pré-adolescent et l'adolescent I. Esquisse d'un modèle théorique: Concepts de base [Kinematic reasoning in the preadolescent and the adolescent: I. Outline of a theoretical model: Basic concepts]. *Archives de Psychologie, 178,* 133–183.

Crépault, J. (1979). Influence du repérage sur la durée: Etude génétique des inférences cinématique [The effect of spotting on duration: Genetic study of kinematic reasoning]. *L'Année Psychologique, 79,* 43–64.

Crépault, J. (1981). Etude longitudinale des inférences cinématiques chez préadolescent et l'adolescent: Evolution ou régression [Longitudinal study of kinematic reasoning in the preadolescent and the adolescent: Development or regression]. *Canadian Journal of Psychology, 35,* 244–253.

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

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: A connectionist perspective on development.* Cambridge, MA: MIT Press.

Fahlman, S. E. (1988). Faster-learning variations on back-propagation: An empirical study. In D. S. Touretzky, G. E. Hinton, & T. J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 138–151). Los Altos, CA: Morgan Kaufmann.

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.

Flavell, J. H. (1963). *The developmental psychology of Jean Piaget.* Princeton, NJ: Van Nostrand.

Flavell, J. H., & Wohlwill, J. F. (1969). Formal and functional aspects of cognitive development. In D. Elkind & J. H. Flavell (Eds.), *Studies in cognitive development* (pp. 67–120). New York: Oxford University Press.

Friedman, W. J. (1978). Development of time concepts in children. In H. Reese & L. P. Lipsitt (Eds.), *Advances in child development* (Vol. 12, pp. 267–298). New York: Academic.

Friedman, W. J. (1990). *About time: Inventing the fourth dimension.* Cambridge, MA: MIT Press.

Gould, E., Reeves, A. J., Graziano, M. S. A., & Gross, C. G. (1999). Neurogenesis in the neocortex of adult primates. *Science, 286,* 548–552.

Greenough, W. T., Black, T. E., & Wallace, C. S. (1987). Experience and brain development. *Child Development, 58,* 539–559.

Halford, G. S., Brown, C. A., & Thompson, R. M. (1986). Children's concepts of volume and flotation. *Developmental Psychology, 22,* 218–222.

Harnad, S., Hanson, S. J., & Lubin, J. (1991). Categorical perception and the evolution of supervised learning in neural nets. In D. W. Powers & L. Reeker (Eds.), *Machine learning of natural language and ontology (Symposium on symbol grounding: Problem and practice). Working papers of the American Association for Artificial Intelligence, Spring Symposium* (pp. 65–74). Stanford, CA: Stanford University Press.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation.* Reading, MA: Addison-Wesley.

Hoehfeld, M., & Fahlman, S. E. (1991). *Learning with limited numerical precision using the cascade-correlation algorithm* (Tech. Rep. No. CMU–CS–91–130). Pittsburgh, PA: Carnegie-Mellon University, School of Computer Science.

Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence.* New York: Basic Books.

Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental perspective on cognitive science.* Cambridge, MA: MIT Press.

Kerkman, D. D., & Wright, J. C. (1988). An exegesis of two theories of compensation development: Sequential decision theory and information integration theory. *Developmental Review, 8,* 323–360.

Lacouture, A., & Marley, A. A. J. (1991). A connectionist model of choice and reaction time in absolute identification. *Connection Science, 3,* 401–433.

Levin, I. (1977). The development of time concepts in young children: Reasoning about duration. *Child Development, 48,* 435–444.

Levin, I. (1979). Interference of time-related and unrelated cues with duration comparisons of young children: Analysis of Piaget's formulation of the relation of time and speed. *Child Development, 50,* 469–477.

Levin, I., & Gilat, I. (1983). A developmental analysis of early time concepts: The equivalence and additivity of the effect of interfering cues on duration comparisons of young children. *Child Development, 54,* 78–83.

Levin, I., Gilat, I., & Zelniker, T. (1980). The role of cue salience in the development of time concepts: Duration comparisons in young children. *Developmental Psychology, 16,* 661–671.

Levin, I., Israeli, E., & Darom, E. (1978). The development of time concepts in young children: The relations between duration and succession. *Child Development, 49,* 755–764.

Lohaus, A., & Trautner, H. M. (1989). Information integration by children: The identification of rules by an alternative method. *Genetic, Social, and General Psychology Monographs, 115,* 329–347.

Mareschal, D., & Shultz, T. R. (1993). A connectionist model of the development of seriation. In W. Kintsch (Chair), *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (pp. 676–681). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Mareschal, D., & Shultz, T. R. (1996). Generative connectionist networks and constructivist cognitive development. *Cognitive Development, 11,* 571–603.

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). New York: Oxford University Press.

Montangero, J. (1977). *La notion de durée chez l'enfant de 5 à 9 ans* [The notion of duration in 5- to 9-year-old children]. Paris: Presses Universitaires de France.

Montangero, J. (1979). Les relations du temps, de la vitesse et de l'espace parcouru chez le jeune enfant [Relations between time, speed, and distance in young children]. *L'Année Psychologique, 79,* 23–42.

Newell, A. (1990). *Unified theories of cognition.* London: Routledge & Kegen Paul.

Piaget, J. (1964). Relations between the notions of time and speed in children. In R. E. Ripple & V. N. Rockcastle (Eds.), *Piaget rediscovered* (pp. 40–48). A report on the conference of Cognitive Studies and Curriculum Development. Ithaca, NY: Cornell University.

Piaget, J. (1969). *The child's conception of time* (A. J. Pomerans, Trans.). London: Routledge & Kegen Paul. (Original work published 1946)

Piaget, J. (1970). *The child's conception of movement and speed* (G. E. T. Holloway & M. J. Mackenzie, Trans.). London: Routledge & Kegen Paul. (Original work published 1946)

Piaget, J. (1971). *Psychology and epistemology* (A. Rosen, Trans.). New York: Orion Press Grossman. (Original work published 1970)

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

Quartz, S. R., & Sejnowski, T. (1997). The neural basis of cognitive development: A constructivist manifesto. *Behavioural and Brain Sciences, 20,* 537–596.

Rumelhart. D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Exploration in the microstructure of cognition: Vol. 1. Foundations* (pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tense of English verbs. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Exploration in the microstructure of cognition: Vol. 2. Psychological and biological models* (pp. 216–271). Cambridge, MA: MIT Press.

Schmidt, W. C., & Ling, C. X. (1996). A decision-tree model of balance scale development. *Machine Learning, 24,* 203–229.

Schmidt, W. C., & Shultz, T. R. (1992). An investigation of balance scale success. In J. J. Kruschke (Chair), *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (pp. 72–77). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Schyns, P. (1991). A modular neural network model of concept acquisition. *Cognitive Science, 15,* 461–508.

Shultz, T. R. (1994). The challenge of representational redescription [Review of the book *Beyond modularity: A developmental perspective on cognitive science*]. *Behavioral and Brain Sciences, 17,* 728–729.

Shultz, T. R. (1998). A computational analysis of conservation. *Developmental Science, 1,* 103–126.

Shultz, T. R., Buckingham, D., & Oshima-Takane, Y. (1993). A connectionist model of the learning of personal pronouns in English. In T. Petsche (Ed.), *Computational learning theory and natural learning systems* (Vol. 2, pp. 347–362). Cambridge, MA: MIT Press.

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. (1991). A cascade-correlation model of balance scale phenomena. In K. J. Hammond & D. Genter (Chairs), *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 635–640). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Shultz, T. R., Schmidt, W. C., Buckingham, D., & Mareschal, D. (1995). Modeling cognitive development with a generative connectionist algorithm. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 205–261). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8,* 481–520.

Siegler, R. S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development, 46*(2, Serial No. 189).

Siegler, R. S., & Richards, D. D. (1979). Development of time, speed, and distance concepts. *Developmental Psychology, 15,* 288–298.

Sirois, S., & Shultz, T. R. (1998). Neural network modeling of developmental effects in discrimination shifts. *Journal of Experimental Child Psychology, 71,* 235–274.

Sternberg, R. S. (1984). *Mechanism of cognitive development.* New York: Freeman.

Weinreb, N., & Brainerd, C. J. (1975). A developmental study of Piaget's groupement model of the emergence of speed and time concepts. *Child Development, 46,* 176–185.

Wilkening, F. (1980). Development of dimensional integration in children's perceptual judgment: Experiments with area, volume, and velocity. In F. Wilkening, J. Becker, & T. Trabasso (Eds.), *Information integration by children* (pp. 47–69). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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). New York: Academic.

Wilkening, F., & Anderson, N. H. (1982). Comparison of two rule-assessment methodologies for studying cognitive development and knowledge structure. *Psychological Bulletin, 92,* 215–237.

Wilkening, F., & Anderson, N. H. (1991). Representation and diagnosis of knowledge structures in developmental psychology. In N. H. Anderson (Ed.), *Contributions to information integration theory: Volume 3. Developmental* (pp. 45–80). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Yang, J., & Honavar, V. (1998). Experiments with the cascade-correlation algorithm. *Microcomputer Applications, 17,* 40–46.