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

David Buckingham
Department of Psychology
McGill University, Montréal

November, 1993

A Thesis submitted to the Faculty of Graduate
Studies and Research in partial fulfillment of the
requirements of the degree of Masters of Arts

The Development of Distance, Time, and Velocity: A Connectionist Model

Abstract

Two sets of connectionist simulations of children's acquisition of distance (*d*), time (*t*), and velocity (*v*) concepts using a generative algorithm, cascade-correlation (Fahlman & Lebiere, 1990), are reported. *Pure condition* simulations represent a situation in which memory demands across the concepts are equal. The *limited memory condition* explores the effects of differing memory demands.

It was found that the rules that correlated most highly with network responses during training were consistent with the developmental course of children's concepts (Wilkening, 1981; 1982). 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 effects of memory demands, the contribution of connectionism to cognitive development, and directions for future research. It is argued that cascade correlation provides an explicit mechanism of developmental change -- weight adjustment and hidden unit recruitment.

Résumé

Cette étude présente deux séries de simulations connexionistes du développement des concepts de distance (*d*), de temps (*t*), et de la vitesse (*v*) en utilisant l'algorithme "cascade-correlation" (Fahlman & Lebiere, 1990) qui construit de façon dynamique la topologie du réseau. Les simulations "pure condition" représentent une situation dans laquelle les demandes mnémoniques sont équivalentes pour chaque concepts. Les simulations "limited memory condition" explorent l'effet de différences dans les demandes mnémoniques entre les concepts.

Les résultats démontrent que les règles les plus hautement corrélées avec les réponses du réseau pendant la phase d'entraînement étaient en concordance avec le parcours développemental des concepts observé chez les enfants (Wilkening, 1981; 1982). Dans un premier temps, les réseaux intégraient les dimensions définissants les concepts en adoptant une règle d'identité (e.g., $v = d$), par la suite, par une règle additive (e.g., $v = d - t$), et finalement, en adoptant une règle multiplicative (e.g., $v = d \div t$).

Les résultats sont considérés en termes de leur similarité au développement de l'enfant, des effets des demandes mnémoniqes, des contributions de l'approche connexioniste au domaine du développement cognitif et d'avenues de recherches futurs. L'argument est fait que "cascade-correlation" nous fournit avec un mécanisme explicite du changement développemental; soit l'ajustement de connexions et le recrutement d'unité au besion.

Acknowledgments

This work was supported in part by Natural Science and Engineering Research Council of Canada grant 144926 and Fonds pour la Formation de Chercheurs et l'Aide à la Recherch grant 920978.

I gratefully thank Dr. Thomas R. Shultz for providing me with support, motivation, and the opportunity to conduct this research in a stimulating environment. I would also like to thank Denis Mareschal for all the informal discussions we had concerning connectionism, cognitive development, and the cascade-correlation algorithm. Furthermore, thanks are also due to Serge Arsenault for translating the abstract and Scott Fahlman for having provided our lab with the cascade-correlation computer code.

Finally, I would like to thank my long time companion Marguerite (Gydee) Roy for commenting on very early draft versions of this thesis.

Table of Contents

List of Tables

List of Figures

CHAPTER ONE - INTRODUCTION

Children's behaviour is often characterized as progressing through a series of stages of increasingly complex knowledge. Although this notion has had enduring appeal, it is only recently that researchers within the field of cognitive development have possessed a tool that enables them not only to describe the knowledge representations of possible stages but also how these states might emerge. Within the last decade, connectionism has provided new insights into developmental regularities of children's behaviour.

One widely researched area in cognitive development that is frequently characterized as stage-like is children's performance in compensation tasks. Compensation tasks involve differences, rates, ratios, proportions, or other multidimensional interactions among physical dimensions (Kerkman & Wright, 1988). Examples include: (1) the balance scale task (e.g., Inhelder & Piaget, 1958; Siegler, 1976, 1981; Wilkening & Anderson, 1982, 1991; Ferretti, Butterfield, Cahn, & Kerkman, 1985; McClelland, 1988; Newell, 1990; Shultz & Schmidt, 1991) in which information about weight and distance from the fulcrum must be multiplied (i.e., torque) in order to predict which side of the balance scale will go down; (2) area judgement tasks (e.g., Anderson & Cuneo, 1977; Wilkening, 1980; Lohaus & Trautner, 1989; Avons & Thomas, 1990) in which information about the height and width of an object must be integrated to predict area; (3) volume judgement tasks (e.g., Anderson & Cuneo, 1977; Wilkening, 1980; Halford, Brown & Thompson, 1986) in which information about height and diameter are necessary to predict liquid volume in a beaker for example; and (4) tasks involving the integration of distance, time, and velocity information (e.g., Piaget, 1946/1969; Levin, 1977, 1979; Siegler & Richards, 1979; Wilkening, 1981).

This thesis is concerned with the last set of compensation tasks. Specifically, it is an attempt to further our understanding of children's acquisition of the concepts of distance, time, and velocity by means of computer simulation using a generative connectionist architecture, cascade-correlation (Fahlman & Lebiere, 1990).

In classical physics, distance (d) is defined as the relationship between time (t) and velocity (v), $d = t \times v$, time as the relationship between distance and velocity, $t = d \div v$, and velocity as the relationship between distance and time, $v = d \div t$. Very few studies have looked at the simultaneous acquisition of these three concepts. Moreover, the findings of two approaches that have had a significant impact on this topic (Siegler & Richards, 1979; Wilkening, 1981) differ with respect to the general developmental course and the specific knowledge and representations they attribute to children over this period. In order to understand these differences, the theories and measurement techniques behind these approaches are discussed in some detail in the following review.

1. Literature Review of Distance, Time, and Velocity Acquisition

1.1. Piaget's Approach

The beginning of modern research into children's concepts of distance, time, and velocity is rather distinguished. Albert Einstein asked Jean Piaget (1946/1969; 1964) about the relationship between time and velocity in children's thinking. Is one more primitive than the other? Does one depend on the other? Given that Newtonian mechanics defines velocity in terms of the fundamental notions of time and space whereas Relativity theory postulates that time and space are relative to an absolute velocity (i.e., speed of light), Einstein's interest is evident.

In order to answer Einstein, Piaget undertook a full scale investigation (Flavell, 1963; Piaget, 1946/1969, 1946/1970, 1964, 1970/1971) and determined that the acquisition of distance, time, and velocity concepts occurred primarily in three stages. Intuitive notions emerge at approximately four to five years of age (Stage I: intuitive or pre-operational), and are reconstructed via an intermediary stage (Stage II: articulated intuitions or operational), until the adult-like concepts are obtained at roughly eight to nine years of age (Stage III: concrete operational). An additional fourth stage (Stage IV: formal operational) marks the progression from qualitative to quantitative understanding. Piaget's claims are based on a number of tasks developed to measure conceptual knowledge he believed to be essential in understanding the three concepts.

Distance

The child's notion of distance evolves from intuitions based on spatial order, and in particular the stopping point of a movement, to an understanding that distance is comprised of the interval between starting and stopping points which can be subdivided into smaller units that can be used for measurement.

Figure 1 shows the hypothetical path trajectories of two streetcars (beads) used in one experiment (Piaget, 1946/1970). The examiner moves his/her car from points 0 to 4 on path B for example, and asks the child to move his/her car the same distance on path A. A child at Stage I would move his/her car to position 2 thus

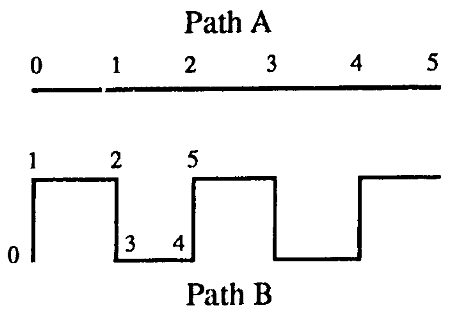


Figure 1. Adaptcd from Piagct (1946/1970, p. 57). Pathway followed by strectcars A and B startıng at zero. (Intervals marked only for ease of explanation)

only focusing on the stopping points of the two cars. During Stage II, the child judges the length of the path traversed intuitively moving his/her car to a position around point 4. At Stage III, the child is able to make direct comparisons of the paths traversed by measuring them.

At Stage IV, children can reason abstractly about distance. For example, when a snail moves along a plank which is moving at the same speed in the opposite direction, children understand immediately that the snail's position relative to its starting point does not change.

Time

Piaget's (1946/1969) research into children's notion of time examined their understanding of the temporal concepts of succession (order of events) and duration (interval of time). Piaget maintained that time for the young child is not homogeneous in that its rate varies from one motion to the next, or is relative to velocity, and that it is confused with the spatial order. The child progresses from this "localized" time to being able to represent time on uniform scale that can be used to compare objects traveling at different velocities. Thus, time according to Piaget, is the coordination of motions at different velocities.

Piaget's examination of simultaneity (i.e., of events that start and stop at the same time) is used to illustrate the developmental course of the child's notion of time. Children are asked which of two mechanical snails traveling at different speeds stops first or travels for the greater time. At Stage I they choose the snail that travels the least distance as having stopped first. Thus, the simultaneity of the stopping points is not understood. Moreover, the faster moving snail is judged to take more time. At stage two, children may grasp either the simultaneity of the starting and stopping points or the equality of the durations, but not both. Alternatively, they understand neither, and simply judge duration as inversely proportional to distance (i.e., more distance = less time). Finally, at Stage III, the concurrent development of notions of succession and duration lead to the acquisition of a concrete operational understanding of temporal simultaneity.

Comparing successive durations represents the child's crowning achievement as he/she progresses from a qualitative to quantitative understanding of time. Piaget argues that successive durations can only be compared if the units of duration can be removed from the actual events, thus enabling quantification. Although it is not clear from Piaget's (1946/1969) writings whether or not this constitutes a new stage (i.e., Stage IV), it is reasonable to conceive it as such given his conceptualizations of stage IV with respect to distance and velocity.

Velocity

The child's intuitive notion of velocity is based on relating the order of events in time with the order of events in space (Piaget, 1964, 1970). It is not dependent on either temporal or spatial intervals (i.e., durations or distances) but rather on overtaking and overreaching. That is, an object which at one point in time ("before") is behind another and then later ("after") is in front of it, is judged as having greater velocity. However, if the overtaking of a faster object is not visible either by making it hidden (e.g., traveling through tunnels), having the objects travel at angles to one another, or on concentric circular paths, the child at Stage I will say that the objects traveled at the same velocity.

The importance of overreaching is clearest in the absence of overtaking. When two cars start and stop simultaneously with one commencing considerably further behind the other but just failing to overtake it, the child thinks the car that traveled the lesser distance traveled faster because it's stopping point is further along the path. Similarly, if the two cars start in succession from the same point of origin but stop simultaneously having traveled equal distances, the child will think that they went the same speed since their respective stopping points are the same.

During Stage II, the child generalizes the notion of visible overtaking to situations in which it is hidden by imagining the continuation of a movement to establish if a "potential" overtaking is likely. Similarly, the child may reconstruct the movements form their points of origin. In any event, attention is decentered from the stopping points to include the starting points and this paves the way for the concrete operational solutions of Stage III.

As with distance and time, Stage IV represents a shift from qualitative to quantitative understanding. At this point the child masters the conservation of uniform speeds, grasps the idea of uniformly accelerated movement, and is able to construct the proportion of distance to time to predict velocity (i.e., $v = d \div t$)

when unequal distances and unequal times of movements in succession are involved.

Conclusion

In answer to Einstein's question concerning which concept is more primitive, Piaget concluded that whereas none of the concepts are innate by virtue of the fact that they are constructed through a series of stages, an early intuition of velocity based on temporal ("before" and "after") and spatial order ("behind" and "in front") exists independent of notions of duration and distance. On the other hand, time is dependent on velocity at all stages. Thus, Piaget concluded that children's early notions are more akin to relativistic concepts and it is only through development that they acquire their Newtonian qualities.

Supporting evidence. Most of the research aimed at replicating and extending Piaget's claims has been done with respect to children's understanding of the time concept. Early support for Piaget came from a comparative study of children with special needs and "normals" by Lovell and Slater (1960). Looking at the concepts of simultaneity, equality of synchronous events, and children's ability to order events, they found considerable support for Piaget's claims, although they did note that children's notion of time is situation dependent, performing better in some tasks than others. Cross-cultural studies also tended to replicate Piaget's findings with the caveat that cultural time lags did exist (e.g., Dempsey, 1971; Bentley, 1986). Additional evidence has come from a variety of areas including research in music education looking at the effects of tempo on judged duration of the music (e.g., Bickel, 1984). For a review of studies of children's time concept, the interested reader should consult Friedman (1978).

A few studies have looked at the relationship between time and velocity. For example, Weinreb and Brainerd (1975) found support for between concept developmental course of time and velocity (i.e., that velocity precedes time) although they question some of the inter-concept development claims.

A few researchers (e.g., Montangero, 1977, 1979; Crépault, 1980) have looked at the relationship amongst all three concepts. One interesting series of studies has been conducted by Crépault (1977, 1979, 1981) looking at the "relativity" of the concepts. In these studies, children are shown a mechanical device that pulls a paper ribbon past a mechanical arm that has a pen at the end of it. As the arm moves up and down, marks are drawn on the paper ribbon. The frequency of the pen tapping and the ribbon pulling can be manipulated independently. Subjects are then presented with two ribbons and asked, for example, if it took more time for the pen to make the first two marks on one tape than the other. Similarly, they are asked questions about the time it took the ribbon to go between the first two points on either tape. Crépault has found that common errors include differentiating the time required by the pen and that of the tape.

Based on his research, Crépault (1978) has formulated an elaborate three-stage theory of the developmental course of the concepts in which pairwise combinations of the relations emerge in the following order: At first, more speed implies less distance and more time implies more distance; then the child learns that more speed implies more distance; and finally that more time implies less distance.

Finally, general support for Piaget's position has come from one of the first comprehensive studies to look at not only the relationship amongst the three concepts but also their inter-developmental course (Siegler and Richards, 1979).

Their methodology, theoretical assumptions, and results are discussed at length in the following section.

1.2. Siegler's Approach

Sequential Decision Theory

Siegler's theory of cognitive development is greatly influenced by Piaget's stage theory of development. Siegler's *Sequential Decision Theory* postulates that cognitive development can be characterized as a series of increasingly powerful rules for solving problems (Siegler, 1981).

Siegler (1981) argues that within Piaget's work on compensation tasks, there is a modal form to the developmental course that children are thought to follow. First, children focus on the *dominant* dimension. Then, there is a transition stage in which the child focuses on both dimensions but does not know how to integrate them. Finally, the child correctly integrates the two dimensions. Siegler proposed four modal rules to explain this developmental sequence. Each is explained in Table 1.

At each stage in development, a child's performance is based on binary decision rules (Siegler, 1976). For example, a child using Rule I determines

Table 1
Modal Rules Used to Explain Development

Modal Rule	Description
Rule I	Base judgment on *dominant* dimension.
Rule II	When *dominant* dimension values are unequal base judgments on this. If equal, consider *subordinate* dimension.
Rule III	Consider both dimensions but when one alternative has greater value on the *dominant* dimension and the other has greater value on the *subordinate* dimension, "muddle through" (i.e., guess!).
Rule IV	Apply the appropriate quantitative or qualitative formula for combining the dimensions.

whether the values of the *dominant* dimension are equal among the alternatives. If they are, the child maintains that the two alternatives are equal. If the values are not equal, the child chooses the alternative with the greater *dominant* dimension value.

Sequential Decision Theory has been used successfully to explain performance in tasks such as the balance scale (e.g., Siegler 1981, 1976) and the inclined-plane (Ferretti *et al.*, 1985).

Rule Assessment

In order to ascertain whether or not children's performance can be characterized by the rules postulated under Sequential Decision Theory, Siegler looks at children's correct and incorrect responses to different problem types.

Problem types in choice tasks. Although Siegler (1981) argues that his Rule Assessment procedure does not require a methodology in which children choose among alternatives, most if not all of the research pertaining to Sequential Decision Theory has been done within the choice paradigm. With this in mind, six modal problem types based on dominant and subordinate values have been used to assess children's knowledge. These are presented in Table 2.

Table 2
Problem Types in Rule Assessment

Problem Type	Values of Dominant and Subordinate Dimensions
Equal	Equal *dominant* and *subordinate* values.
Dominant	Unequal *dominant* but equal *subordinate* values.
Subordinate	Equal *dominant* but unequal *subordinate* values.
Conflict-Dominant	One choice has greater *dominant* values, and the other has greater *subordinate* values. Former choice is correct.
Conflict-Subordinate	Same as above except latter choice is correct.
Conflict-Equal	Same as above except both choices produce the same outcome.

The set of problem types is chosen such that it yields a distinctive pattern of errors and correct answers for children following the modal rules described above. For example, a child using Rule I (basing judgments solely on the basis of the *dominant* dimension) is expected to perform correctly on the Equal, Dominant, and Conflict-Dominant problem types but to make errors on the other three problems.

Assumptions underlying the rule assessment technique. According to Siegler (1981) there are only two prerequisites for using the Rule Assessment technique. The first being that the researcher must have clear hypotheses about people's knowledge of two or more distinct strategies such that the different strategies will yield different response patterns. The second prerequisite is that the problem types allow for simple application of the strategies. In other words, the experimenter must control for such things as memory constraints so that the subjects are able to use the strategies.

Distance, Time, and Velocity

Siegler and Richards (1979; see also Richards, 1982) argued that there were several reasons to question Piaget's results including inconsistent findings, methodological problems in that Piaget did not study the same children nor did he use comparable tasks across the three concepts, and the vagueness of Stage II.

Siegler and Richards attempted to clarify Piaget's account of children's understanding of the logical concepts of distance, time, and velocity by presenting two toy trains which traveled along parallel tracks and asking the children to judge which of the two traveled either the greater distance, for the longer time, or at the faster velocity. The trains could differ along seven physical dimensions: (1) time, (2) velocity, (3) distance, (4) starting position, (5) stopping position, (6) starting time, and (7) stopping time. Using *Sequential Decision Theory* as a guide, they hypothesized three rules based on Piaget's work (1946/1969, 1946/1970). Children using Rule I would judge solely on the basis of stopping points. Those using Rule II would consider starting positions when the trains stopping positions were the same. Finally, children using Rule III would solve the problems correctly. Based on Siegler's *Rule Assessment Approach* (1976, 1981, 1982) they created six problem types to differentiate children's performance.

With respect to the distance concept, 5-year-olds generally used either Rule I or Rule II. However, Rule II was modified after analyzing the children's verbal protocols as it was found that the children based their judgements on stopping time rather than starting points. Eight and 11-year-olds were found to use Rule I and III while all adults used Rule III.

Analysis of children's time concept was less clear. Although 5-year-olds consistently used either Rule I or Rule II, the majority of 8-year-olds were not classifiable using the rule assessment procedure. Eleven-year-olds appeared to use a *distance rule* in which amount of time traveled was based on distance traveled. That is, they chose the train that had traveled further as having traveled for the greater time. Only the adults were found to use Rule III.

The acquisition of the velocity concept was shown to be similar to that of distance. Five-year-olds used either Rule I or II, 8-year-olds were either unclassifiable or used Rule 3, while 11-year-olds and adults consistently used Rule III.

In summary, 5-year-olds were found to use Rule I across all three tasks. That is, these children chose the train which had stopped the furthest down the tracks to have traveled for the greatest distance, time, or at the greatest speed. At the other end of the age spectrum, adults were found to use Rule III. Children in between these two age groups regularly confused speed and distance, distance and time, and time and speed. Moreover, Siegler and Richards found that children understood velocity and distance concepts before the time concept. Additional evidence for this inter-developmental sequence was found by Acredolo and Schmid (1981) even though they extended the stimulus set to include situations where the trains ran for equal times, equal distances, equal velocities, or all three.

1.3. Criticisms of Piaget and Siegler

Recent investigators have criticized previous research since it appears to have tested the child's ability to ignore rather than integrate the distance, time, and velocity dimensions.

Brendt and Wood (1974) found that pre-operational children have a primitive capacity for judging duration when visual cues are not present. In a task similar to Piaget's, Brendt and Wood presented young children with two toy trains running along parallel tracks. The tracks could be covered by tunnels or in full view. In addition, each locomotive had a distinctive whistle which sounded continuously as the train moved. The researchers found that when the tracks were covered, hence the only cue being the whistles, young children were able to tell which train had traveled for the greater duration. However, when the tunnels were removed revealing a conflict situation where the train that ran for a longer time actually traveled a shorter distance, the children would revert to a distance-only model, choosing the train which had traveled the greater distance.

Levin and her collaborators have conducted a series of studies aimed at showing that Piaget underestimated young children's early notions about time. They have shown that: *1)* children's early concepts of time are basically temporal in that they understand the interdependency of succession and duration when interfering cues are held to a minimum (Levin, 1977); *2)* this understanding is extremely unstable in that even cues that are not logically related may interfere with it (Levin, 1979); *3)* some cues interfere more than others (Levin & Gilat, 1983); *4)* succession and duration do not develop concurrently (Levin, Israeli, & Darom, 1978) as suggested by Piaget and finally; *5)* young children's poor performance reflects the perceptual salience of end-points (Levin, Gilat, & Zelnicker, 1980) rather than a conceptual deficit as assumed by Piaget, and Siegler and Richards. While most of Levin's work has been done in relation to children's qualitative understanding of time, her more recent work has focused on time quantification (Levin, Wilkening, & Dembo, 1984; Wilkening, Levin, & Druyan, 1987).

Levin *et al.'s* research interest has been guided by the assumption that typical Piagetian tasks and the methodology used to assess performance obscure young children's knowledge of the relationship between succession and duration. For example, Levin (Levin & Globerson, 1984) has raised concerns about the ability of the rule assessment technique to capture children's knowledge given that children do not consistently apply rules across problems.

On a more conceptual level, Levin argues that distance and velocity serve as interfering cues in these tasks (for a review see Levin, 1982). Levin (1977) demonstrated this point by comparing children's duration judgements of two events in three tasks which differed according to the amount of information available. In the *still time* task, only succession information was given. The children were asked to judge "sleeping" times of two dolls which either went to sleep and/or woke simultaneously or at different times. In the *rotational time* task, succession and velocity information were presented to children by using two figures that rotated on individual turntables (obscuring distance information) either at the same or different speeds. Finally, in the *linear time* task, succession, velocity, and distance information were given by using the same type of apparatus as Piaget and Siegler (i.e., two trains).

In line with Levin's expectations, most preschoolers were able to solve the *still time* problems, first graders were also able to solve *rotational time* problems, and even third graders found *linear time* problems difficult. Thus, Levin suggested that not only did the distance and velocity cues interfere with children's understanding of the relationship between succession and duration, they did so in an additive manner. That is, the problems with two interfering cues (linear) were harder than problems with one (rotational) which were in turn harder than problems in which there were no such cues (still). Although later research suggested that the type of interfering cue might be a more important factor than the number of interfering cues (Levin & Gilat, 1983), the major premise that Piagetian tasks were not adequately reflecting children's knowledge was upheld. Moreover, Friedman (1990) notes that Levin's research revealed that when interfering cues were absent, children were able to construct a common time for two events, the sleeping dolls.

Levin was able to demonstrate that the interference with children's understanding of duration and succession is not due to the fact that distance and velocity are related to time. In fact, Levin (1979) found that cues that were logically unrelated to time (the brightness of a lamp) showed similar types of interference effects. Levin concluded that children were using a *more is more* rule whereby any event that was perceived to be greater on some dimension (intensity or velocity for example) was judged to have lasted for a greater length of time.

Although Levin's work suggested that results obtained using Piagetian tasks misrepresented children's understanding of the concept of time, she did find that when the two events differed in temporal endings as opposed to beginnings, children found it easier to judge the relative durations. However, rather than assuming a conceptual deficit (focusing on end-points) as Piaget and Siegler and Richards did, she postulated that this phenomenon was due to the perceptual salience of the end-points. This assumption was supported in a study by Levin, Gilat, and Zelnicker (1980).

Levin's work has led her to suggest that rather than the notions of succession and duration developing concurrently as maintained by Piaget, children first understand succession. This knowledge then may mediate an understanding of duration. Support for this two-stage development comes from the fact that children as young as five years of age often refer to succession in their rationalizations about duration (Levin, 1977; Levin, Israeli, & Darom, 1978), while they offer tautological explanations of succession. Moreover, Levin has argued that only after they grasp the dependency of duration and succession do they begin to understand the relationship between distance and velocity with respect to time, an assumption she claims is supported by the fact that linear time problems are solved much later in development than still time problems.

Richie and Bickhard (1988) have questioned Levin's two stage theory by demonstrating that children can solve time problems when non-temporal cues such as succession are absent. Using a methodology similar to Levin (1979; Levin et al. 1983) Richie and Bickhard asked children to compare the durations of two lamps. However, they used problem types in which only sensed temporal information could be used to solve the problems correctly. That is, in these problems there is no onset/offset succession information that would be helpful in that the bulb that came on first would also go off first. Moreover, larger disparities between the durations of the two lights were investigated in addition to the standard durations used by Levin.

Given that children with a mean age of 4.5 years of age were able to solve the problems that lacked succession information and given that when large enough duration disparities were provided, the children could solve the same problem-types used by Levin, the authors concluded that any theory of children's logical concept of time that assumes that time *necessarily* develops from the coordination of non-temporal perceptually based cues (e.g., succession, distance, and velocity) is not viable. In other words, children have a perceptual experience of time separate from one derived inferentially from non-temporal knowledge.

1.4. Wilkening's Approach

Wilkening (1981) has investigated children's acquisition of distance, time, and velocity concepts within the framework of Anderson's *Information Integration Theory* (1974, 1980).

Information Integration Theory

Anderson's (1974, 1980, 1991) Information Integration Theory maintains that knowledge is *not* represented as a series of rule-governed binary decisions, but rather, assumes that children's representation of knowledge is characterized by *cognitive algebra*. Anderson has argued that most algebraic models fall into two classes. The first class includes *adding*, *subtracting*, and *averaging* models. The second includes *multiplying* and *dividing* models. The basic idea is that children integrate physical stimuli using one of these models such as *dimension 1* + *dimension 2*. The results of the application of an algebraic model are then expressed in subjective or psychological values.

Three laws. Anderson has posited three laws to explain the sequence from sensation of some stimuli to the response based on those stimuli. The first step in the sequence, governed by a Psychophysical Law, involves the processing of physical stimuli by a *valuation function*. This function determines the *scale value* and *weight* of the stimuli. The former refers to the location of the stimulus on the individual's subjective dimension of judgment whereas the latter indicates the amount of information in the stimulus or the emphasis placed on the stimulus. The key point is that both depend on the dimension of judgment. In other words, each task the child performs sets up a valuation function.

Next in the sequence, the individual combines or integrates the stimuli according to the Psychological Law. In this step, an *integration function* is applied to the stimuli. As noted above, this involves the application of some algebraic model. The Psychological Law determines an implicit response which is then transformed by a Psychomotor Law into the explicit response of the individual. Again this last step is intimately tied to the task at hand. The Psychomotor Law mediates between the implicit response and the constraints of the measuring apparatus.

Developmental course. Information Integration Theory has been used to explain a diverse range of developmental phenomena including children's moral judgments (e.g., Leon, 1980, 1982); probability judgments (e.g., Acredolo, O'Conner, Banks, & Horobin, 1989; and Homers, 1980); area judgments (e.g., Anderson & Cuneo, 1977; Wilkening, 1980); and performance on the balance scale task (e.g., Wilkenıng & Anderson, 1982) to name but a few.

Generalizing across these tasks, the following developmental course is revealed. At first a child's performance can be characterized by algebraic models from the first class mentioned (e.g., adding or subtracting). Then the child goes through a transition stage in which the child's performance cannot entirely be explained by models from either class. Finally, the child is classified as using a model from the second class (e.g., multiplying or dividing).

Although performance during the transition phase is most often explained *post hoc*, some interesting models have been put forward. For example, Wilkening and Anderson (1982) explained children's transition phase on the balance scale task from adding to multiplying weight and distance information as a combination of the two. During the transition period, children were found to multiply when lower levels of weight were involved and to add when higher levels of weight were involved.

Functional Measurement

Ordinal vs. interval response scales. A major force behind the creation of Functional Measurement is Anderson's (1974) belief that people's responses are expressed as subjective, psychological values. Anderson argues that for psychological tasks in general, it cannot be expected that such responses fall on an interval scale, but rather, fall on an ordinal scale. Moreover, since psychological scales are typically interval, any ordinal response cannot fit the model that describes the process underlying the behaviour. Functional Measurement is meant to provide a means by which subjects' responses can be expressed and accurately measured on an interval scale.

There are two essential aspects to the Functional Measurement method. The first is the use of factorial designs of stimulus combinations. For example, if one were testing for children's judgments of distance traveled by three objects of intrinsically different speeds (for example, very fast, moderately fast, and slow) given three different times (for example, long, medium, and brief), the experimenter would use all nine possible stimulus combinations.

It is unnecessary for the experimenter to know *a priori* what the subjective values of velocity and time values are in terms of numerical indices. However, it is essential that the subject's judgments are numerical responses. This is the second constraint of functional measurement. If the overt, measured response is on a linear scale, the data table of the factorial design provides a functional scale of the subject's responses. That is, the marginal means are a linear scale of the subjective values of the stimuli.

Parallelism, diverging-fans, and ANOVA statistics. Instead of creating different problem types to assess subject's knowledge, Functional Measurement relies on graphical patterns of responses and Analysis of Variance statistics. Only a brief sketch of these two methods is provided here.

To ascertain graphical patterns, a factorial plot of the data is undertaken. Within the context of the experiment mentioned above, the plot would have measured *distance* on the y-axis and either categorical values of *time* (long, medium, and brief) or *velocity* (very fast, fast, or slow) on the x-axis. If *time* was used for the x-axis, then three separate lines would be plotted, one for each of the three levels of *velocity*. If the factorial plot yields three parallel lines then the underlying cognitive algebraic model is either an addition or averaging type model. The tesι for multiplicative models is similar except that marginal means (i.e., their subjective numerical value) of one dimension are plotted on the x-axis. If the lines taken together form a diverging fan pattern, the underlying process is said to be a multiplicative model.

While the graphical patterns yields a heuristic type assessment, Analysis of Variance provides a statistical test of significance of any observed patterns. Additive type models are indicated by significant main effects but non-significant interactions whereas multiplicative models yield a significant interaction effect.

Distance, Time, and Velocity

Wilkening has done extensive research in this area (Wilkening, 1981, 1982; Wilkening & Anderson 1982; Levin, Wilkening, & Dembo, 1984; Wilkening, Levin, & Druyan, 1987; Anderson & Wilkening, 1991). Wilkening's (1981) quintessential work assessed children's ability to integrate information to infer distance, time, or velocity. He argued that previous research within the Piagetian approach (e.g., Siegler and Richards, 1979) had encouraged and assessed children's ability to ignore rather than integrate the dimensions.

Wilkening designed three tasks, one to reveal children's understanding of each concept. In each task the subject was given information about two dimensions and asked to infer the third. In the *distance task*, in which velocity and time are to be integrated, children are shown an apparatus which at one end of a footbridge has a shed that is said to be occupied by a fearsome dog. At this same end, there are three animals with inherently different velocities (turtle, guinea pig, and cat). These animals are said to be "afraid" of the dog and will run along the footbridge when the dog starts barking and stop when the barking ceases. The children are asked to predict how far the animals will run given the amount of time the dog barks. Thus, the children are given information about the velocity dimension (i.e., the inherent speed of each of the three animals) and the time dimension (i.e., the length of barking) and asked to infer the distance traveled (i.e., the point along the footbridge that the animal runs to).

In the *time task*, children are asked to infer time from distance and velocity information. In this task, one of the three animals is placed at one of three distances along the footbridge. The child is then asked to indicate how long it would take the animal to reach that point by "playing" a continuous recording of the dog barking for the appropriate time.

In the velocity task, children are asked to infer velocity from distance and time information. In this task, the dog barks for one of three time intervals and then a white piece of cardboard is place at one of three distance along the footbridge. The child uses a response scale of seven animals of differing velocity to indicate which animal would attain the distance of the piece of cardboard given the length that the dog barks.

Wilkening studied the performance of three age groups; 5-year-olds, 10-year-olds, and adults and discovered the following: (1) in the *distance task*, all age groups use the correct multiplication rule ($d = t \times v$), (2) in the *time task*, 10-year-olds and adults employ the correct division rule ($t = d \div v$), whereas 5-year-olds use a subtraction rule ($t = d - v$), and (3) in the velocity task, the two older age groups use a subtraction rule ($v = d - t$), and 5-year-olds use a proportional distance rule ($v = d$). Wilkening concluded that young children did indeed have the ability to integrate dimensions. Moreover, he found that children understand distance before time and that even adults do not fully master the velocity concept.

Wilkening discussed the possibility that differing memory demands might be responsible for the varying levels of integration across the three tasks and subsequent inter-developmental sequence. In particular, Wilkening noticed that both children and adults appeared to use an *eye movement strategy* in the *distance task.* That is, subjects would follow the imaginary movement of the animal when the barking started and would point to the place where they were looking when the barking stopped. This strategy was also used by 10-year-olds and adults in the *time task.* However, when 5-year-olds attempted to use the strategy, they had difficulty presumably because of the simultaneous demands of focusing attention on holding the "play" button of the tape recorder and imagining the animals movement. Finally, with respect to the *velocity task,* this strategy would involve being able to imagine all seven animals running at the same time. Even for adults this was not a viable strategy.

Wilkening argued that the use of the eye movement strategy did not invalidate the conclusion that children could in fact integrate dimensions. Instead he claimed that in order for children and adults to "know" that the strategy was useful, they implicitly had to understand the relationships between the dimensions. Nevertheless, Wilkening admitted that strong conclusions about the acquisition order of the three dimensions relative to one another were unwarranted given the different memory demands of the three tasks.

In an attempt to further understand this inter-developmental course, Wilkening (1982) attempted to equalize memory demands across the three tasks. Wilkening endeavored to meet this goal by trying to increase the memory demands of the *distance task* and to decrease them in the *velocity task*. In the *distance task*, the order of time and velocity information was reversed. That is, the subjects heard the dog bark before knowing which animal's distance was to be inferred. Thus, an eye-movement strategy is unlikely since the subjects do not know *a priori* which velocity is relevant.

In the *velocity task*, the memory demands were simplified by visually presenting the time information. This was done by using a "cartoon bubble" with the words "bow wow" that moved along the footbridge with a constant speed and stopped when the barking ceased. The length of the bubble itself was different depending on the time that the dog barked. This correspondence between bubble length and time of barking was explicitly revealed to subjects.

The manipulations were partially successful. In the *distance task*, 5-year-olds were found to use an additive rule ($d = t + v$), whereas adults used the multiplicative rule as was discovered in the first experiment. In contrast, the results of the velocity task were the same as in the first experiment. That is, 5-year-olds used the proportional distance rule and adults used the subtraction rule.

Thus, when the use of the eye-movement strategy is not viable, 5-year-old children appear to use an additive (or subtraction) rule when integrating velocity and time, and velocity and distance information. Adults, on the other hand, can successfully use a multiplicative (or division) rule. Unfortunately Wilkening did not investigate 10-year-olds' performance as was done in the first study. Therefore, with respect to distance and time concepts no conclusions can be drawn about one being mastered before the other since it is just as likely that these children would use either the additive rule or the multiplicative rule.

The discrepancies between children's and adult's performance in the *time task* and *velocity task*, remains unclear. That is, one explanation is that Wilkening's results are valid in that velocity development lags behind time development. Alternatively, Wilkening's memory demand manipulation may have been ineffective.

Metric versus Non-metric Integration

Acredolo, Adams, and Schmid (1984) have argued that although Wilkening's (1981) results show that children can integrate the appropriate distance, time, and velocity dimensions metrically (i.e., to predict metric distance, for example), the results say nothing about whether or not children can integrate dimensions non-metrically. To examine this, they used an analogous situation to Wilkening's in which two animals were imagined to have been chased from a farmer's field by his dog. Given statements such as "they ran the same speed, and *x* ran for a longer time", the children had to choose which outcomes were possible: (1) *x* ran farther, (2) *y* ran farther, or (3) they ran the same distance. Moreover, they were presented with "open ended" situations. For example, if "*x* ran faster, and *y* ran for a longer time" all three outcomes above are possible.

Acredelo et al. found that children understood the direct relationship between speed and distance, and between duration and distance before they understood the indirect relationship between speed and duration. By "direct relationship", the authors meant that a perceptually greater value on one dimension correlates with a perceptually greater value on the other dimension. However, they concluded that the majority of 11-year-olds did not integrate the dimensions non-metrically in that they could not recognize logical conflicts between pairs of non-metric relations.

Crépault (1980) has used a similar approach. Children were asked to judge the compatibility of statements concerning two dimensions and to make inferences about the third dimension. For example, the child was presented with the following statements and questions: "The red car goes faster than the black car. The two cars go the same distance. Is this possible? Can we tell from these two statements if the two cars went for the same length of time or did one travel for a longer time?"* The results indicated that some statements were easier to judge than others. For example, the situation above was judged compatible more readily by the subjects than one in which one car traveled faster but both traveled for the same time.

Wilkening's concerns with non-metric designs. Wilkening (1982) has discussed four potential problems with studying non-metric relations. In summary, they are as follows. First, the processing capacity required in such tasks is often large given that the child needs to remember starting times, ending times, starting points, and ending points of both events when asked to compare velocities for

* Translated and adpated from Crépault (1980).

example. Second, the child is assumed to understand linguistic terms such as "more" or "less". However, Wilkening argues that past research has shown that terms such as these may have different meanings for children than adults. Third, findings concerning non-metric relations cannot be generalized to knowledge of continuous, quantitative relations. Finally, unless one dimension is held constant, it is logically impossible to infer a third (*cf.* the "open ended" questions used by Acredolo *et al.*). Since one needs to be held constant, then it is uncertain whether this dimension was considered or ignored.

1.5. Summary

Piaget (1946/1969, 1946/1970) concluded that the concepts of distance, time, and velocity undergo gradual construction, characterized by four stages, until a logical understanding of their inter-relationship emerge. While very few researchers looked at the inter-developmental course of these concepts, Siegler and Richards (1979) conducted a comprehensive study within Siegler's (1976) Sequential Decision Framework, and concluded that the concept of distance was understood before time which was in turn understood before velocity.

Investigators began to question Piagetian methodology. One the one hand, Levin and her co-workers (Levin, 1982) argued that within the typical Piagetian task, distance and velocity information interfered with children's judgements about time. Moreover, she successfully demonstrated that this interference was not due to the fact that distance and velocity are logically related to time. This led Levin to look at children's conception of time independent of distance and velocity, focusing on children's notions of succession and durations. Richie and Bickhard (1988) went even further, demonstrating that it was possible to study children's conception of time independent of their notion of succession.

On the other hand, Wilkening (1981) argued that the Piagetian tasks were solved more easily by ignoring the interrelation of the dimensions. This led Wilkening on a different path than Levin and others by refocusing the issue of the nature of children's understanding on the interrelation of distance, time, and velocity. Using functional measurement, derived from Anderson's (1974) Information Integration Theory, he was able to study children's understanding of the three dimensions when they were required to infer one dimension's value based on information about the other two. Wilkening concluded that children could integrate distance, time, and velocity, particularly if the memory demands of the tasks were at a minimum.

2. Connectionism and Development

Connectionist, or parallel distributed processing (PDP) models employ simple processing units which send inhibitory or excitatory signals via weighted connections to other simple units, forming what is called a "network". In the models relevant to this discussion, learning occurs by small adjustments of the connections, or weights, between an input layer where stimulus information is encoded and an output layer where the "response" is made. Two broad categories of networks within this domain exist: static and generative networks. Static networks (typically back-propagation networks) employ network structures that do not change from the beginning of learning (training) to the end. Conversely, generative networks change the structure of the network as they learn by recruiting or pruning units in the network. A more detailed discussion is presented in Chapter 2.

Recently a number of researchers (e.g., Rumelhart & McClelland, 1986; McClelland, 1988; Elman, 1991; Schyns, 1991; Shultz, Schmidt, Buckingham, & Mareschal, in press) have begun to use these models as a means of understanding and characterizing cognitive development. In addition to offering fresh insight into some old ideas, connectionism is believed to provide new insights into cognitive development by addressing some issues that have been either overlooked or ignored.

2.1. What Develops and How Does it Develop?

As Flavell and Wohlwill (1969) have pointed out, any account of cognitive development must concern itself with both the formal and functional aspects of development. In other words, the study of cognitive development must seek to answer two fundamental questions: (1) What knowledge structures develop?, and; (2) How does developmental transition occur?

Research concerning what structures develop has flourished. However, relatively little work has been conducted in the area of transition mechanisms (Sternberg, 1984). Bates and Elman (1992) offer an intriguing supposition concerning why this is so based on the computer metaphor of cognitive development derived from the symbolic computational approach widely accepted in the 1970s. In brief, symbolic computational assumptions such as discrete representations (i.e., symbols), absolute rules to manipulate the symbols, characterizing learning as programming, and the relative unimportance of possible implementation constraints (i.e., *functionalism*), fostered a paradigm which considered mechanisms of change as somewhat unimportant. Conversely, Bates and Elman maintain that the assumptions of connectionism such as distributed representations, graded rules in the form of weighted connections, learning characterized as structural change, and consideration of implementation constraints offers a new manner in which to study not only what develops but how it develops (for a similar points of view, see Churchland, 1990; Plunkett & Sinha, 1992) .

Stages in cognitive development. One issue that has been revitalized by the connectionist approach to cognitive development is the *stage* versus *continuous learning* issue. As in Piaget's theory, children's cognitive development is often characterized as progressing through a series of stages. The child's knowledge is considered to be constant across a given stage with learning having little or no effect on performance. An alternative to this would be that learning is continuous across the stage and that the only reason stages are seen at all is because the methods researchers use to evaluate performance are not subtle enough to capture continual change.

Flavell (1971) summized that Piaget's stage theory had four major implications and conducted an extensive review of the literature to see if these were borne out . Essentially, stage theories implied that cognitive development could be characterized as an *abrupt, qualitative* restructuring of highly *organized knowledge* that occurred *concurrently* across many domains. Flavell found support for the notions of progression through qualitatively different knowledge structures that were organized. However, he concluded that abrupt transition from one stage to another was not likely and, moreover, that changes across domains were often independent of each other.

Shultz (1991) has found good agreement between connectionist findings and its assumptions about cognitive development and those offered by Flavell. Shultz maintains that learning in connectionist networks results in qualitative changes in network performance brought on by both small quantitative changes in weights amongst units and qualitative network restructuring through changes in topology (in the case of generative architectures). Thus, rather than characterizing development as *either* continuous or discontinuous, connectionism provides an integrated account of both these aspects of development. Moreover, lack of concurrent changes across domains is a natural result of the learning environment of these networks given that they rely heavily on experience. Change in network performance is most often gradual but when abrupt changes do occur, they are limited to the domain in which the network is trained. Finally, the inter-connection of units provides an organized knowledge structure.

Transition mechanisms: Assimilation and accommodation revisited. Piaget believed that through emergent structures brought on by the *equilibration* of the processes of *accommodation* and *assimilation*, transition from one stage to another resulted. However, despite his efforts, the mechanism that caused the transition remained vague (Bates & Elman, 1992). Recently a number of authors have offered the view that connectionism provides a precise account of the transition mechanism in addition to new interpretations of assimilation and accommodation (e.g., Bates & Elman, 1992; Plunkett & Sinha, 1992).

Within the static network domain, researchers have argued that gradual and continuous weight changes result in stage-like performance (Plunkett & Sinha, 1992). Moreover, McClelland (1988) has argued that accommodation occurs when the network's weights are updated during learning thus modifying the structure of knowledge in the netwcrk. Conversely, assimilation occurs when generalization to a new instance or input does not result in any weight change.

Shultz, Schmidt, Buckingham, and Mareschal, (in press) have suggested that generative architectures go one step further by suggesting that recruitment of new units into the network provides a second potential transition mechanism by increasing the computational complexity of the network. Furthermore, they argue that accommodation occurs as new units are added *instead* of simple weight changes reflecting what the authors termed *assimilative learning*.

2.2. Connectionist Simulations of Cognitive Development

In this section, a review of two cognitive developmental tasks are presented -- children's acquisition of the past-tense of verbs and their performance on the balance scale.

Past-tense

Children's acquisition of past-tense morphology is often cited as an argument that children possess explicit, albeit tacit rules. Rumelhart and McClelland (1986) challenged this assumption by demonstrating that a connectionist network could also account for children's performance. In particular, the networks exhibited a U–shaped developmental pattern similar to that characteristically observed in children. That is, it appears as though children first correctly use regular and irregular past-tense forms, then seemingly form an explicit rule (adding "ed") which they over-generalize to irregular verbs, before finally learning only to apply it to regular verbs.

Rumelhart and McClelland's model came under severe attack led most vigorously by Pinker and Prince (1988). The main criticism was the use of varying degrees of exposure to regular and irregular forms that were not representative of children's linguistic environment. This issue was addressed in a set of simulations by Plunkett and Marchman (1989) in which the type and token frequencies of irregular and regular verbs were manipulated in a manner which reflected biases that were present in children's linguistic environment. Despite this more realistic environment, U-shaped development was again observed in the networks.

The important point is that the networks never possessed anything close to an explicit "ed" rule, but learned to discriminate irregular and regular verb endings based on experience in a manner which was similar to children.

The Balance Scale

The balance scale task has become a benchmark of sorts for modeling cognitive development. Siegler (1981, 1976) hypothesized four rules based on the modal rules discussed above (Table 1) to account for children's performance on the balance scale. For example, a child using Rule I, focuses solely on the weight dimension whereas a child using Rule II considers distance but only when the weights are equal, etc. Siegler's general findings were that children between the ages of 3 years to 12 years progress from not using any rule, to rule I, followed by Rule II, and then Rule III. Some, but not all, older children and adults reach Rule IV type performance. These results have been replicated by Ferretti *et al.* (1985).

McClelland (1988) used a back-propagation network to simulate the developmental stages found by Siegler. On the input side, the network was presented with weight and distance information for both sides of the balance scale respectively. The network was segregated in that the weight information was processed by one set of hidden units (an intermediary level of units between the input and output layers) while the distance information was processed by another set of hidden units. The networks exhibited stage-like performance progressing from Rule I to III. Occasionally Rule IV behaviour was achieved, however, this performance was unstable and the networks would regress to an earlier stage. Thus, using simple weight updates, the network's performance was seen to change qualitatively.

McClelland's results were extended by Shultz and Schmidt (1991) using a generative connectionist architecture by capturing stable Rule IV behaviour and demonstrating the product-difference effect (Ferretti & Butterfield, 1986). The product-difference effect describes the finding that subjects generally perform better on problems in which the difference between weight and distance on one side of the fulcrum is much greater than it is on the other side.

2.3. Summary

Connectionism has provided not only a new impetus for looking at transition mechanisms in cognitive development but also a precise formulation of what this mechanism might be. In particular, gradual weight adjustment and, in some cases, changes in network topology have led to performance that conforms to behaviours that have typically been explained by the acquisition of explicit rules.

3. Rationale

Wilkening (1981, 1982) succeeded in establishing a methodology that allowed for the examination of Piaget's original interest in the interrelation of distance, time and velocity, and addressed issues that were problematic in Piagetian tasks. By providing children with information about the two defining dimensions of a given concept and asking them to infer its value, he was able to study how children integrated these dimensions. Wilkening (1981) found that children as young as 5 years correctly integrated the dimensions of time and velocity using a multiplicative rule to infer distance. However, when the use of an eye-movement strategy was made more difficult, 5-year-olds were found to fall back on an additive rule. With respect to the time concept, 5-year-olds used a subtraction rule whereas 10-year-olds and adults used the correct rule. Finally, in the velocity task, 5-year-olds used a centration rule focusing only on distance whereas the other groups used a subtraction rule.

Connectionism has shown that it allows for a characterization of not only what develops but how it develops by providing precise transition mechanisms while at the same time exhibiting stage-like performance similar to that reported in the developmental literature.

In particular, one generative connectionist architecture, cascade-correlation (Fahlman & Lebiere, 1990; see chapter two of this thesis), has been found to provide a good account of compensation tasks such as the balance scale (Shultz & Schmidt, 1991) and the effects of potency and resistance on the magnitude of a physical effect (Shultz *et al.* in press). Furthermore, cascade-correlation has the potential for capturing both qualitative and quantitative shifts in knowledge representations.

The present study attempts to further our understanding of children's ability to integrate the dimensions of distance, time, and velocity by modeling their performance with cascade-correlation. In addition to the reasons mentioned above, it is expected that the increasing non-linear representational power of a system that recruits hidden units as needed will provide insight into how children progress from simple centration rules to more complex multiplicative rule-like performance.

Two sets of simulations are reported testing the adequacy of cascade-correlation to capture the developmental regularities found by Wilkening. In both, the network is provided with information about two dimension and asked to predict the third for a given event. By using computer simulations to understand children's acquisition of distance, time, and velocity, it is possible to control the effects of differing memory demands across tasks eliminating possible confounds such as an eye-movement strategy. The first set of simulations represents an ideal situation in which the memory demands of all three tasks are identical. The second set examines Wilkening's assumption that the developmental sequence he observed reflects differing memory demands. If the effects Wilkening observed were in fact dependent on memory demands, then the second set of simulations should conform to Wilkening's results more closely. Moreover, while Wilkening failed to look at the effects of 10-year-olds the simulations will attempt to investigate the entire range of development.

CHAPTER TWO - CASCADE-CORRELATION

The following chapter discusses cascade-correlation -- a learning algorithm that constructs a generative feedforward connectionist network. For more details, see Fahlman (1988), Fahlman and Lebiere (1990), Hoehfeld and Fahlman (1991), Sjøgaard (1991), and Yang and Honavar (1991).

Feedforward connectionist architectures employ simple processing units that send and/or receive either excitatory or inhibitory signals to other simple units in the network via weighted connections. At minimum, there is an input layer in which stimulus information is encoded as a pattern of activation across a given number of input units and an output layer where the response associated with a particular input pattern is produced. Similarly, this response is determined by the pattern of activation on the output units making up the output layer.

Minsky and Papert (1969) argued that networks consisting of only an input and output layer, so-called perceptrons, cannot learn some types of problems. For example, the classic *XOR* problem in which the task of the network is to respond positively when either one of two inputs is "on" but not when both are either "on" or "off", is theoretically impossible for a perceptron to solve. Thus, multi-layered networks consisting of one or more hidden unit layers between the input and output layers are often used. Hidden units receive no external input and are used to build internal representations of the stimuli.

Cascade-correlation belongs to a large class of gradient-descent learning connectionist environments (for a review see Hertz, Krogh, & Palmer, 1991).

Learning involves repeated presentations of input patterns and gradual adjustment of weights or connections between units in the network to reduce the error between the pattern of activation across output units, or simply the output, and the desired or target output. As Hoehfeld and Fahlman (1991) point out, the error is typically measured as the sum-squared differences between the current activations of the output units and their target values,

$$E = \frac{1}{2}\sum_{i,p}(a_{ip} - t_{ip})^2 \qquad (2.1)$$

where a_{ip} is the current activation of output unit *i* for input pattern *p*, and t_{ip} is the target output value.

A unit's activation is determined by either a linear or non-linear activation function. The activation of a linear unit is the sum of the weighted inputs to the unit such that

$$a_i = X_i = \sum_j w_{i,j} x_j \qquad (2.2)$$

where a_i is the activation of receiving unit *i*, X_i is the sum of weighted input to unit *i*, $w_{i,j}$ is the weight of the connection from sending unit *j* to unit *i*, and x_j is the activation of unit *j*. In all simulations reported in this thesis the output unit had a linear activation function.

As mentioned, Minsky and Papert (1969) showed that purely linear networks, *perceptrons*, were limited with respect to what they could learn. Generalizing to multi-layered networks, if the hidden unit layer has units with linear activation functions then the network becomes very much like a simple peiceptron. Therefore, in modern day networks hidden units typically have a non-linear activation function.

In all simulations reported in this thesis hidden units with a sigmoid activation function having a range from -0.5 to +0.5 were used. Thus, the activation of a hidden unit is

$$a_i = \frac{1}{1+e^{-x_i}} - 0.5 \qquad (2.3)$$

where e is the exponential function and x_i is the sum of the weighted input to hidden unit i as in Equation 2.2.

1. Cascade Architecture: Dynamic vs. Static

Cascade-correlation is a generative learning algorithm (for a review of generative learning algorithms see Alpaydin, 1991). That is, unlike typical back-propagation networks (Rumelhart, Hinton & Williams, 1986), the network architecture created using cascade-correlation is dynamic in that hidden units are added to the network as training progresses. Figure 2 illustrates the initial or starting architecture of the network with three input units and one output unit, and the structure following the addition of one and two hidden units respectively.

As can be seen the network begins as a simple perceptron. The networks initial configuration is based solely on the manner in which the input is encoded.

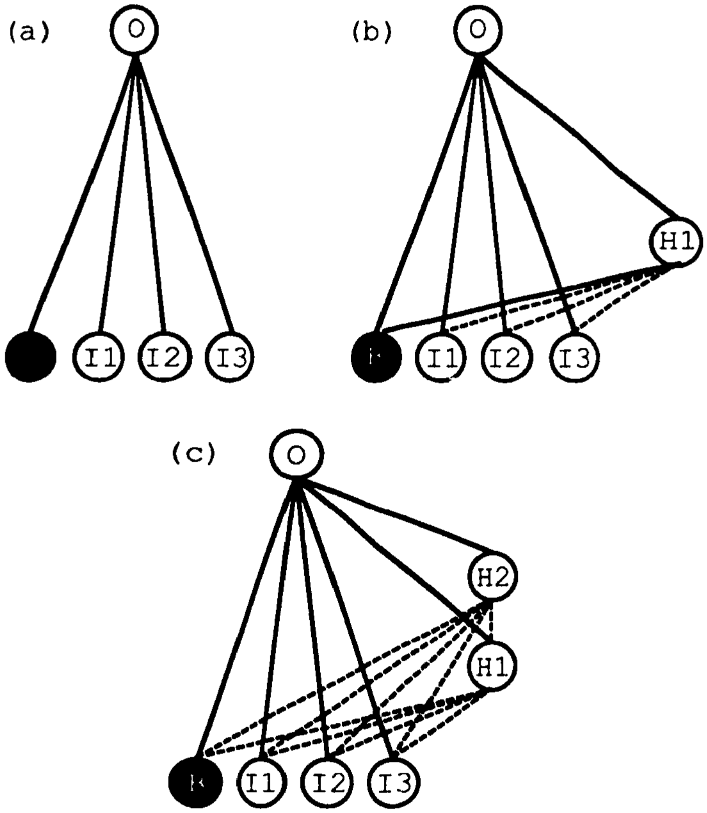


Figure 2. Architecture at beginning (a), after one hidden unit is added (b), and after a second hidden unit is added (c). Solid and dashed lines indicate trainable and frozen connections respectively. I, H and O refer to input, hidden, and output unit respectively. Bias unit shown in black.

If, as in Figure 2, the input is encoded on three units, the resulting network will have four input units -- the three input units and an obligatory bias that always has an activation of one. The bias unit acts as a threshold unit. All units from this input layer have direct connections, or in other words, straight-through connections to the output layer, in this case a single unit.

1.1. Output training phase

Training using the cascade-correlation algorithm consists of a series of two-phased training cycles until *victory* or some maximum number of epochs is reached. An epoch consists of a single presentation of all training patterns. Victory is declared when the number of incorrect error bits across all patterns is zero. Each output unit is considered as an error bit. If the output unit is not within a certain distance (score-threshold) of the target value it is considered incorrect. For example, if a sigmoid output unit is used with a range from -0.5 to +0.5, the target output for a given input pattern is +0.5, and the score-threshold is set to 0.4 (the default for sigmoid outputs) then any response above +0.1 would count as a correct error bit. Thus, if there are 20 patterns in the training set and the network has two output units, then the total possible number of error bits is 40.

During the first phase of the training cycle (i.e., the output training phase) the algorithm attempts to reduce error (as measured by Equation 1) between the target outputs and the actual outputs computed by the network. Three parameters directly control the duration of the phase. The first and second parameters, *output patience* and *output change threshold*, are related to error reduction. The latter refers to the minimum error reduction that is considered to be significant. Output patience is the number of consecutive epochs allowed that do not meet the minimum criterion of the output change threshold. For example, if the output patience is eight epochs (the default value), the algorithm will discontinue the output training phase if eight consecutive epochs occur in which the error is reduced by less than the output change threshold at each epoch.

The third parameter related to the length of the output training phase is the *outlimit*. This is the maximum number of epochs that the algorithm will continue in the output training phase even if the output patience criterion is not met. For example, if the outlimit is set to 50 epochs, output training will progress until 50 epochs have occurred without the output patience having been reached or victory declared. Then the algorithm will begin the input training phase.

1.2. Input training phase

The input training phase is used to recruit (add) hidden units to the network that are maximally correlated with output error. At the start of this phase, a pool of candidate hidden units (default value of eight units) are each connected to all input units (including the bias) and any existing hidden units. All candidate hidden units are then connected to all output units of the output layer. Input training continues until one of two input training parameters is reached -- *input patience* and *inlimit.* Input patience is synonymous with it's output training counterpart except that the minimum criterion of the *input change threshold* reflects the amount of change in the correlation rather than the error. Similarly, the inlimit is the maximum number of epochs in which the correlation between candidate hidden units and the output error is maximized.

When either the inlimit or input patience is met, the candidate hidden unit is installed into the network. It's input-side weights are then frozen. In other words, these weights do not undergo any further training, remaining the same until learning is completed. In contrast, the weights leading from the hidden unit to the output layer are not frozen and hence, are adjusted as needed. In Figure 2, this feature of frozen versus trainable weights is illustrated by dashed and solid lines respectively. The other hidden units in the pool are then "removed" and the two cycled training phase begins again with output training.

In sum, Figure 2 illustrates a hypothetical architecture of a network beginning the first output training phase (a), after completing one input training phase (b), and after another output-input training cycle (c).

2. Quickprop Algorithm

In cascade-correlation, the weights are updated or adjusted using the Quickprop algorithm (Fahlman 1988). Quickprop is a variant of the back-propagation weight update rule in which the change in the connection strength of a given weight is given by the equation

$$\Delta w_{(t)} = -\varepsilon \frac{\partial E}{\partial w_{(t)}} + \Delta w_{(t-1)} \qquad (2.4)$$

where ε is the learning rate, $\partial E / \partial w_{(t)}$ is the error derivative, and $\Delta w_{(t-1)}$ is the weight change of the previous epoch. The updated connection strength of the weight is then obtained by

$$w_{(t+1)} = w_{(t)} + \Delta w_{(t)} \qquad (2.5)$$

where $w_{(t)}$ is the connection strength of the weight at the previous epoch.

Fahlman and Lebiere (1990) have argued that gradient descent using Equation 2.4 is relatively slow since only the partial first-order derivative of the error is computed. They argue that if infinitesimal steps down the error slope are taken this is adequate. However, using second-order derivatives in which not only the slope of the error space is considered but also it's curvature, much larger steps can be used in gradient descent.

Fahlman's Quickprop algorithm makes use of a heuristic type approach to using these second-order derivatives. Quickprop uses information about the previous first-order error derivative, $S_{(t-1)}$, the current error derivative, $S_{(t)}$, and the previous change in connection strength of the weight to define a parabola in weight-space. The object is then to make a weight change that reduces error to the bottom of this parabola. Specifically, the change in connection strength for weight *i* is given by

$$\Delta w_{(t)} = \frac{S_{(t)}}{S_{(t-1)} - S_{(t)}} \Delta w_{(t-1)} \tag{2.6}$$

where S is simply the slope or first-order derivative ($\partial E / \partial w$) used in back-propagation.

Two special cases arise when Equation 2.6 is not used for computing the weight change. One is when the previous weight change is zero. Jn this case, the change in weight strength is computed by

$$\Delta w_{(t)} = \varepsilon S_{(t)} \tag{2.7}$$

where ε is some constant controlling the amount of gradient descent. The second case occurs when the current slope is the same size or greater than the previous slope. Here the weight change is limited such that

$$\Delta w_{(t)} = \mu \Delta w_{(t-1)} \tag{2.8}$$

where μ is the maximum step size allowed.

Fahlman (1988) has shown that using this second-order method, Quickprop out performs standard Back-propagation on several benchmarks.

3. Correlation

The goal of input training, as mentioned above, is to install the hidden unit whose activation most closely correlates with the output error into the network. Again the Quickprop algorithm is used according to the criterion stated above (Equations 2.6, 2.7, and 2.8). However, in this case the object is not to perform gradient descent to minimize sum-square error but rather to perform gradient ascent to maximize

$$C = \sum_{o} \left| \sum_{p} (V_p - \overline{V})(E_{p,o} - \overline{E}_o) \right| \tag{2.9}$$

where C is the sum of the magnitude of the correlation between V_p, the value or activation of the hidden unit given input pattern p, and E_o, the residual output error of output unit o, over all output units.

Finally, the partial error derivative of C is calculated as follows;

$$\partial C / \partial w_i = \sum_{p,o} \sigma_o (E_{p,o} - \overline{E}_o) f'_p I_{i,p} \tag{2.10}$$

where σ_o is the sign of the correlation for output o, f'_p is the derivative is the hidden unit's activation function with respect to the sum of it's inputs for pattern p, and $I_{i,p}$ is the input the hidden unit receives from unit i for pattern p. Defining S in Equations 2.6 and 2.7 as $\partial C / \partial w_i$ instead of $\partial E / \partial w$, the Quickprop algorithm for updating weights is used for gradient ascent in the input or hidden unit training phase as it is used for gradient descent in the output training phase.

As mentioned above, the candidate hidden unit with the greatest correlation is then installed into the network, it's input-side weights frozen, and output training resumes.

CHAPTER THREE - METHOD

Two sets of simulations were conducted -- one representing the possible effects of memory constraints, the *limiteu memory condition,* the other without them, the *pure condition.* The network architecture employed in both sets was the same and is discussed first, followed by a description of the training and testing patterns used, the parameter settings of the algorithm, the procedure for training and testing the network, and how the output of the network was treated.

The network's task in both sets of simulations is to determine the value of the missing dimension (*distance*, *time*, or *velocity*) given information about the other two (*time* and *velocity*, *distance* and *velocity*, or *distance* and *time*) for a single event. For example, an event might include information about the velocity of an object and the total time that it traveled. The network's task is to compute the total distance traveled.

1. Network Architecture

The initial architecture consists of a fully connected network of input and output units only. Three groups of input units are used to represent the dimensions of distance, time, and velocity respectively. The number of units per group is dependent on the type of encoding used and is discussed below. The input units, including the obligatory bias unit, are connected to a single output unit with a linear activation function (Equation 2.2). A linear output unit was used because it was the most natural way of producing a quantitative output similar to the responses made by subjects in Wilkening's (1981, 1982) experiments. As training progresses, any hidden unit incorporated into the network receives connections from all input units as well as from any existing hidden units and is connected directly to the output unit. The hidden units have a sigmoid activation function (Equation 2.3).

1.1. Input encoding

Five types of input encoding commonly found in the literature were used to determine the effect of encoding on performance. These included four distributed types of encoding; *mercury* (e.g., Harnad, Hanson, & Lubin, 1991), *thermometer* (e.g., Anderson, 1990), *gaussian* (e.g., Lacouture & Marley, 1991), and *integer* (e.g., Shultz & Schmidt, 1991); and one local encoding, *nth* (e.g., McClelland, 1988).

In distributed representations, more than one unit is used to represent any one input value. Moreover, 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 one 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 one and all other units have an activation of zero. Thus, the total number of units is *n* + 2. The gaussian coding used is the same as thermometer coding except that the *n*th + 1 unit has an activation of three.

In integer coding, only one unit is used per input group. However, the unit is involved in the representation of more that one input value. Thus, in this sense it is also a distributed representation. The unit is assigned the integer value of the dimension.

Table 3
Example Input Encoding For a Input Value of Three

Type	Code
Integer	3
Nth	0 0 1 0 0
Mercury	1 1 1 0 0
Thermometer	0 0 1 1 1 0 0
Gaussian	0 0 1 3 1 0 0

Finally, in nth coding, for any input value *n*, the *n*th unit has an activation of one and all other units have an activation of zero. Thus, it is a local representation in that each unit is used exclusively to represent a given input value. As with mercury coding, the total number of units is equal to the maximum input value used. An example of each of these encoding types for an input value of three is 'llustrated in Table 3.

Thus, a given network architecture will consist of a total of:

$$IU = (G \times E) + 1 \qquad (3.1)$$

input unit (IU), where G is the number of input groups, and E is the number of units ıequired for the encoding type used, plus 1 for the obligatory bias unit. For example, using thermometer encoding with the maximum dimensional value of five, the number of input units would be $(3 \times 7) + 1$ for a total of 22 units. Each of these would be connected to the linear output unit.

2. Training and Testing Patterns

Training and testing patterns consist of both input values and a target, or output value. Input values are used to encode the event, for example the distance traveled and the amount of time it took, while the output value is the outcome, velocity in this case. 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.

2.1. Pure Condition

Input values

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 of this dimension was unknown. Thus, for any given input pattern, one input group would be all zeros (nth, mercury, thermometer, and gaussian coding) or zero (integer coding) while the other two groups had dimensional values between one and five. For example, for a given distance problem, the distance input group had an input value of zero, and the time and velocity input groups had input values ranging from one to five.

All combinations of the defining dimension values (1 to 5) were used as input patterns for a total of 25 distance, 25 time, and 25 velocity training patterns where distance, time, and velocity was to be inferred respectively. An example of each of the three types of input patterns is illustrated in Table 4 using mercury encoding.

Target values

Training patterns. Target values for the output unit were calculated using the three Newtonian equations ($d = t \times v$, $t = d \div v$, and $v = d \div t$) respectively. In addition, distance target values were scaled by dividing by five (the maximum input

Table 4
An Example of Distance, Time, and Velocity Input Patterns Using Mercury Encoding

	Input Group		
Problem Type	Distance	Time	Velocity
Distance	0 0 0 0 0	1 1 1 0 0	1 1 1 1 1
Time	1 1 1 0 0	0 0 0 0 0	1 1 1 1 1
Velocity	1 1 1 0 0	1 1 1 1 1	0 0 0 0 0

Note. In these examples the two defining dimensions receive values of 3 and 5 respectively and the dimension to be inferred has a value of 0.

value) so that the range would be identical to the target values of time and velocity inference patterns.

Thus, a velocity training pattern representing an object traveling a distance of five units for one unit of time would have input values of 5, 1, and 0 for the distance, time, and velocity input groups respectively and an output, or target value of 5, as computed by $v = d \div t = 5 \div 1$.

Testing patterns. Several sets of testing patterns were created corresponding to potential rules that might best capture network performance. All rules observed by Wilkening (1980, 1981) were included in addition to other rules derived from Information Integration Theory.

The same input values as those in the training set were used. However, the output value associated with a given input pattern was calculated as follows. For distance patterns, output values were calculated according to the following three classes of rules: (1) two *identity* rules in which the output value was determined either solely by the time dimension, $d = t$, or the velocity dimension, $d = v$, (2) three *additive* type rules, $d = t + v$, $d = t - v$, and $d = v - t$, and (3) three *multiplicative* type rules, $d = t \times v$ (the correct Newtonian rule), $d = t \div v$, and

Table 5
Rules Used to Calculate Output Values of the Training Set and Testing Sets

	Rule Types		
Inference Task	Identity	Additive	Multiplicative
Distance	d=t & d=v	d=t+v	d=t*v
Time	t=d & t=v	t=d+v & t=d-v	t=d÷v
Velocity	v=d & v=t	v=d+t & v=d-t	v=d÷t

Note. d = distance; t = time; v = velocity.
Multiplicative rules were used to calculate target output values of the training set.

$d = v \div t$. Thus, there were a total of eight distance testing sets of 25 patterns each (i.e., all input patterns) used to diagnose performance on distance inference patterns. For time inference patterns, the following rules were used: (1) the *identity* rules $t = d$ and $t = v$, (2) the *additive* type rules, $t = d + v$, $t = d - v$, and $t = v - d$, and (3) three *multiplicative* type rules, $t -\ d \div v$ (the correct Newtonian rule), $t = v \div d$, and $t = d \times v$. Analogous rules were used to calculate the output values for the velocity testing patterns.

Pilot simulations revealed that for distance problems both identity rules, one additive rule ($d = t + v$), and the correct multiplicative rule were useful for capturing performance and therefore the remaining distance testing sets were dropped from the testing phase of the simulations to reduce computational complexity. Similarly, for time and velocity problems respectively, the additive type rules, $t = v - d$ and $v = t - d$ as well as four multiplicative rules, $t = d \times v$ and $t = v \div d$, $v = d \times t$ and $v = t \div d$ were dropped. Table 5 illustrates the final 14 testing sets that were used.

2.2. Limited Memory Condition

This condition was an attempt to address some of the issues put forth by Wilkening (1981, 1982) with respect to the cross-developmental course of the three concepts. Wilkening maintained that the reason why subjects performed differently across tasks was due to memory constraints inherent in his study. For example, Wilkening argued that one possible reason why children and adults performed worse on the velocity than the time task was that while both defining dimensions, distance and velocity, were available at the moment of inference in the time task, the same was not true in the velocity task. In particular, time information had to be retrieved from memory and this may have been reflected in the relatively poor results.

Input values

In order to capture the possible memory constraints inherent in Wilkening's (1981) velocity task, the input values of the training patterns were modified as follows. It was assumed that the likelihood of correct recall followed a normal distribution in that values closer to the actual time value would be more likely to be recalled than distant values. As such, at each epoch the time dimension of a given velocity problem was altered such that in general there was a 34%, 13%, 2%, and less than 1% probability that the time input value "recalled" by the network would be 1, 2, 3, or 4 integers away respectively from the actual value. In the remaining instances, the actual value was used as input.

An additional constraint was that the modified time input fall within the same range (1 to 5) used in the training set. This was necessary to avoid any confound due to differences in network topography when comparing the two conditions (pure vs. limited) . Without such a restriction, networks in the limited memory condition would require more input units than those in the pure condition. For example, using thermometer input encoding, an additional five time input units would be needed to reflect the improbable case in which the actual value was five and the integer "recalled" was nine. Moreover, the lack of input values less than one seems self-evident.

The likelihood and degree of modification of the actual input value followed the probabilities outlined above where permitted by the range restriction. If the selection of an integer either above or below the actual value was possible, a random choice was made amongst the two equally distant values. For example, if the actual input value was 3 there was a 34% chance that the network would receive 2 or 4 as input and a 13% chance that it would receive 1 or 5. Otherwise, 3 was used. If the input value was 5, then the chance of a 4, 3, 2, or 1 being "recalled" was 34%, 13%, 2%, and 1% respectively.

Target values

The target values for training and testing patterns remained the same regardless of any modification of the input values. For example, if the velocity problem presented to the network was an event in which the object traveled 5 distance units in 1 time unit, the target value was 5 regardless of whether or not the network received an input value of 1 for the time dimension.

It should be clear that only the velocity patterns were modified. Furthermore, only the time dimension of these patterns were selected according to the criteria above. In all other respects the pure and limited memory conditions were identical.

3. Parameter Settings

Table 6 contains a list of parameter settings used in the simulations and a brief description of their function. Default settings were used for all parameters except output-unit type and score-threshold. The default output-unit type is sigmoid. However, as mentioned, a linear output unit was used. The default score-threshold of 0.4 did not adequately differentiate among output targets. Therefore, a more strict score-threshold of 0.01 was used. All parameters were held constant across all simulations.

4. Procedure

Networks were trained until victory was declared or 1500 epochs were reached. The networks were presented with all 75 training patterns (25 distance, 25 time, and 25 velocity patterns) at each epoch of training. Testing was conducted once at the beginning and ending of the output training phase as well as every five epochs during this interval. Since the output activations don't change during input training, testing during this phase is redundant as the results are the same as the last epoch of the output training phase.

At each testing epoch, the total number of correct patterns and the sum-squared error were recorded for both the entire training set (75 patterns) and individually per pattern type (distance, time, or velocity). In addition, the output activations corresponding to each input pattern were saved for later analysis. Weights were stored at given epochs so that *Hinton* diagrams could be drawn to analyze hidden unit functioning. Finally, 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.

Table 6
Variables and Their Values Used in the Simulations

variable name	value	description
sigmoid-prime-offset	0.1	Value added to *sigmoid-prime function* to avoid situations where it goes to zero.
weight-range	1.0	Initial input and direct input to output weights are randomly set to "+" or "-" values within this range.
weight-multiplier	1.0	Used to calculate initial output weights of candidate units.
output-mu	2.0	QP parameter that sets maximum possible step size in gradient descent during OT.
input-mu	2.0	Same as above except used during IT.
output-shrink-factor	0.67	QP parameter used to calculate if proposed step in gradient descent during OT is too large.
input-shrink-factor	0.67	Same as above except used during IT.
output-epsilon	0.35	QP parameter controlling amount of gradient descent or learning rate in OT.
input-epsilon	1.0	Same as above except used during IT.
output-decay	0.0001	QP parameter added to computed slopes of each weight in OT to avoid *floating-point* overload.
input-decay	0.0	Same as above except used during IT.
output-patience	8	Number of epochs allowed for nonsignificant error reduction in OT after which IT begins.
input-patience	8	Same as above with respect to nonsignificant changes in maximizing correlation in IT after which OT begins.
output-change-threshold	0.01	Error reduction must be greater than output-change-threshold to be consider significant.
input-change-threshold	0.03	Same as above with respect to changes in correlation.
out-limit	100	Maximum number of epoch during OT allowed.
in-limit	100	Maximum number of epochs during IT allowed.
unit-type	sigmoid	Activation function of hidden unit(s) is sigmoidal.
output-type	linear	Activation function of output unit(s) is linear
ncandidates	8	Number of candidate hidden units trained in IT.
score-threshold	0.01	Output units must be within score-threshold to be considered correct.

Note. QP = Quickprop; OT = output training; IT input training.

A total of 30 networks per input encoding were run. Computational complexity made it impractical to store weights for all networks. Thus, only 10 runs were analyzed by Hinton diagrams.

5. Treatment of Output

5.1. Diagnosing Rules

The focus of interest was the rules that best captured the overall performance of the network at a given testing epoch. Since accurate computation on the stimulus input was not demanded of the human subjects in Wilkening's work, an attempt to reflect this was made by assessing the correlation of the testing rules and the actual output of the network. Moreover, by using correlation as the method of assessment, it was unnecessary to scale output values predicted by the testing rules so that they fell into the range in which the network was being trained.

For each of the testing sets listed in Table 5, a Pearson product-moment correlation coefficient was obtained to measure the relatedness of the 25 output target values and the 25 actual outputs produced by the network. These correlations were then converted to r^2 values so that the results could be interpreted in terms of the variance of the actual outputs accounted for by a given rule. However, the sign of the correlation was maintained so that it was possible to determine if the r^2 associated with a given rule was for a rule which positively or negatively predicted performance.

5.2. Diagnosing Stages

Since the simulations were designed to capture developmental trends, it was necessary to diagnose consistent network performance (i.e., stages). Several criteria were used to asses stages. However, before any determination of stage-like behaviour could be done, the rules that best predicted output values of the distance, time, and velocity pattern sets respectively were ascertained.

The best rule per pattern type (distance, time, or velocity) was determined using the following criteria. First, at a given epoch, a rule was required to account for at least 50% of the variance ($r^2 = 0.50$, $r \approx 0.71$) of the pattern set to which it belonged in order to be considered at all. Second, the rule had to be a positive predictor of the output values. Third, the rule had to be a better predictor than its competitors by at least 1% regardless of the direction of the correlation. Although 1% may seem insignificant, two facts warrant the use of this value. First, the degree of relatedness amongst rules is extremely high. For example, the $d = t + v$ rule accounts for approximately 90% of the variance associated with the correct, or defining rule of the distance pattern set ($d = t \times v$). Second, visual inspection of graphs plotting the r^2 of the rules by epoch indicated that this small difference between rules was systematically associated with the beginnings of consistent better performance by the rule with the higher value (i.e., the beginning of a stage). Finally, if two rules met the first and second criterion above but not the third, it was considered to be a tie between the two rules and was noted as such. Otherwise, the epoch being considered was deemed undiagnosible.

In order to determine consistent stage-like performance, the best-rule classification was required to be the same for at least four consecutive testing epochs. Thus, if the r^2 associated with the $d = t + v$ rule was at least 1% better than all the other possible distance rules, positively correlated with the actual output of the network, accounted for at least 50% of the variance, and observed on four consecutive testing epochs, then the network was classified as being in an additive stage with respect to performance on distance patterns.

CHAPTER FOUR - RESULTS

Results of the pure condition are discussed first followed by the limited memory condition. The results of networks using an additional encoding type, ***integer-context***, are reported. As will be shown, performance across four of the original five encoding types was qualitatively similar. However, networks that used integer input encoding differed. It was hypothesized that this difference may be due to the purely linear nature of integer encoding. In order to verify this, networks using a sixth type of encoding, integer-context, were also investigated. This is similar to integer except that three context units, one representing each type of inference problem , are used in addition to the normal integer input groups. The appropriate context unit receives an input value of one while the other two units receive a value of zero. Thus, the input taken as a whole is less linear than straight integer encoding.

PURE CONDITION

1. General Learnability

All networks learned successfully to the point where the normative rules ($d = t \times v$, $t = d \div v$, and $v = d \div t$) accounted for the greatest amount of variance in the output activations of the networks. In fact only one network, in the integer encoding condition, failed to reach the maximum r^2 value (1.00). The mean epoch, sum-squared error, and percent of error reduction from epoch 5 to the point during training where each normative rule accounted for all the variance in output activations are presented in Table 7. Epoch 5 was chosen for comparing percent of

Table 7
Mean Epoch, Error, and Percent Reduction When R^2 of Normative Rules Reached 1.00: Pure Condition

	Inference Problem Type								
	Distance			Time			Velocity		
Encoding	Epoch	*E*	%	Epoch	*E*	%	Epoch	*E*	%
Integer									
M	840.50	0.46	97.88	996.90	0.34	99.17	*971.35	0.31	99.26
SD	160.14	0.14	0.90	159.43	0.07	0.18	137.36	0.04	0.13
Int-cont									
M	920.17	0.40	98.08	814.30	0.33	99.17	808.53	0.31	99.18
SD	106.00	0.09	0.82	160.20	0.13	0.34	150.16	0.05	0.24
Nth									
M	510.23	0.34	99.19	430.20	0.30	98.89	445.97	0.31	98.85
SD	48.27	0.05	0.15	42.76	0.07	0.28	53.03	0.07	0.38
Mercury									
M	721.27	0.38	98.81	666.70	0.32	99.14	657.93	0.36	99.03
SD	66.26	0.09	0.37	93.87	0.07	0.25	79.80	0.09	0.29
Therm									
M	533.90	0.38	98.98	493.87	0.26	99.07	479.90	0.27	99.14
SD	59.67	0.08	0.17	75.51	0.08	0.42	65.65	0.06	0.31
Gaussian									
M	521.73	0.38	99.22	464.13	0.30	98.82	485.43	0.30	99.00
SD	49.42	0.05	0.17	57.56	0.05	0.39	60.24	0.04	0.34

n = 30
* 1 network reached a maximum r^2 = .98

error reduced because it was the first observation after Epoch 0. Epoch 0 was not used because error at this point simply reflects the initial random weights of the network and is thus inconsistent across networks.

A one-way repeated measures ANOVA was performed for each encoding type to determine if the epoch at which the r^2 attained a value of 1.00 was significantly different for distance, time and velocity inference patterns. The difference was significant ($p < .05$) for all encoding types. The following F values were observed: integer $F_{(2,56)} = 13.76$; integer-context $F_{(2,58)} = 7.39$; nth $F_{(2,58)} = 30.28$; mercury

$F_{(2,58)}$ = 7.68; thermometer $F_{(2,58)}$ = 8.29 and; gaussian $F_{(2,58)}$ = 9.73. Thus, for all encoding types except integer, the normative rules of time and velocity reached their maximum r^2 before the distance rule. The opposite occurred when integer encoding was used.

It was theoretically possible to achieve a perfect r^2 even if the actual outputs of the network were far from their targets. That is, r^2 is an indicator of the amount of variance accounted for between the actual outputs and target values by a given rule. It is not dependent on how close in magnitude the outputs are to their targets. However, as can be seen in Table 7, the sum-squared error at this point was very small. For example, a sum-squared error of 0.30 indicates that, on average, the 25 distance inference patterns were 0.15 away from their target values.

The attainment of perfect r^2 and the low sum-squared error suggests that as far as learning is concerned all that remained was simple weight adjustment. However, training continued until 1500 epochs for all networks without victory being declared indicating that the algorithm was incapable of reaching a score-threshold of 0.01.

2. Stage Diagnoses

Overall, stage diagnosis produced a coherent classification of network performance. The majority of the networks exhibited developmental sequences similar to those observed by Wilkening (1981, 1982). All networks, collapsed across input encoding types, exhibited the same distance developmental sequence ($d = t + v$ followed by $d = t \times v$). Ninety-two percent of the networks (excluding those in the integer encoding condition) followed a similar time developmental course progressing from a stage defined by $t = d - v$ to a stage defined by

$t = d + v$. The difference between the sequence observed by Wilkening and the one characteristic of network performance was that the networks first progressed through an identity stage defined by the rule $t = d$. Similarly, 93% of the networks (excluding integer encoding) followed a progression from an identity ($v = d$) to an additive ($v = d - t$) and then a multiplicative ($v = d + t$) stage. Again, this is similar to Wilkening's results with the exception that the networks attained the final multiplicative stage.

2.1. Distance

As stated, performance on the distance inference problems was captured by the same developmental sequences as the one observed by Wilkening (1982). That is, networks first demonstrated the additive rule, $d = t + v$, followed by the correct multiplicative rule $d = t \times v$. For networks with nth, mercury, thermometer, and gaussian encoding, the additive stage began after the recruitment of one hidden unit and lasted 200 epochs on average. The multiplicative stage began at approximately 300 epochs following the recruitment of a second and third hidden unit on average.

For networks using integer and integer-context encoding, the additive stage began very early on in training and thus, was not preceded by the recruitment of a hidden unit. It was longer than in other encodings, lasting for approximately 300 epochs. As a result, the multiplicative stage began at approximately the same time for integer and integer-context encodings as the other encodings. At this point, 3 or 4 hidden units had been recruited on average. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, and the length of the stage are reported in Table 8 for each encoding type.

Table 8
Distance Developmental Sequence by Encoding Type: Pure Condition

	d=t+v			d=t*v	
Encoding	hid	onset	length	hid	onset
Integer					
M	0.00	4.17	301.97	3.53	341.43
SD	0.00	1.90	103.05	1.07	117.68
Int-cont					
M	0.07	15.60	327.17	4.00	446.93
SD	0.25	19.64	104.68	1.02	108.31
Nth					
M	1.00	70.93	164.77	3.07	248.97
SD	0.00	3.81	78.40	1.02	83.48
Mercury					
M	1.00	82.23	217.03	3.27	331.37
SD	0.00	9.22	87.20	0.94	101.66
Therm					
M	1.00	66.80	163.63	3.20	262.67
SD	0.00	3.91	55.67	0.76	70.59
Gaussian					
M	1.00	65.87	165.20	3.30	258.07
SD	0.00	5.40	64.51	0.92	71.78

n = 30 for all encoding types
Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

Performance Attributable to Stage

During the additive stage, the amount of variance in the output values of the distance inference patterns accounted for by the additive rule ($d = t + v$) greatly exceeded the minimum set by the *best rule* criterion ($r^2 \geq 0.50$). The mean maximum r^2 of the additive rule attained during the additive stage is shown in Figure 3 for each encoding type. As can be seen, even for mercury encoding, in which the mean maximum r^2 attained was the lowest of the six encodings, 89% of the variance was accounted for. Moreover, for all integer and integer-context

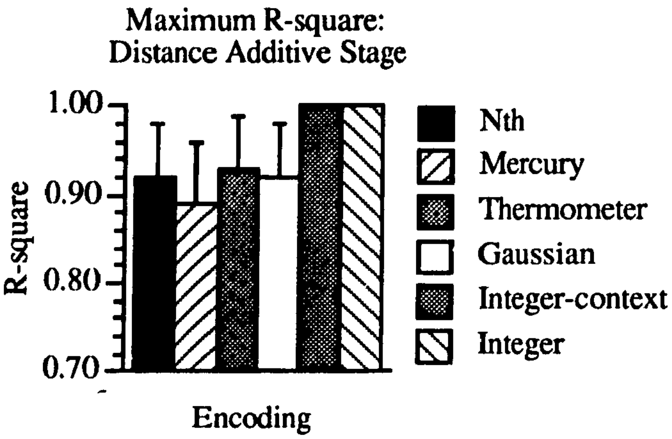


Figure 3. Maximum r^2 of d=t+v rule attained during additive stage of distance developmental sequence.

networks the r^2 of the additive rule reached 1.00. As reported in Table 7, during the multiplicative stage the defining rule $d = t \times v$ reached an r^2 of 1.00.

Thus, within the stages of the distance developmental sequence, a large percentage of the variance in the output values was accounted for.

Error Reduction

Each stage was associated with successive error reduction. The mean percent of error reduction from epoch 5 to the epoch prior to the onset of the additive and multiplicative stages is shown in Figure 4a for all encodings except integer and integer-context. For these two encodings the additive stages typically began at Epoch 5 and are therefore not included in the figure.

Prior to the onset of the additive stage, the error associated with the distance inference patterns was reduced by between roughly 40-60%. Thus, although this period was not associated with a particular stage there was a sizable amount of error reduction. Learning continued over the additive stage. Approximately 85-90% of

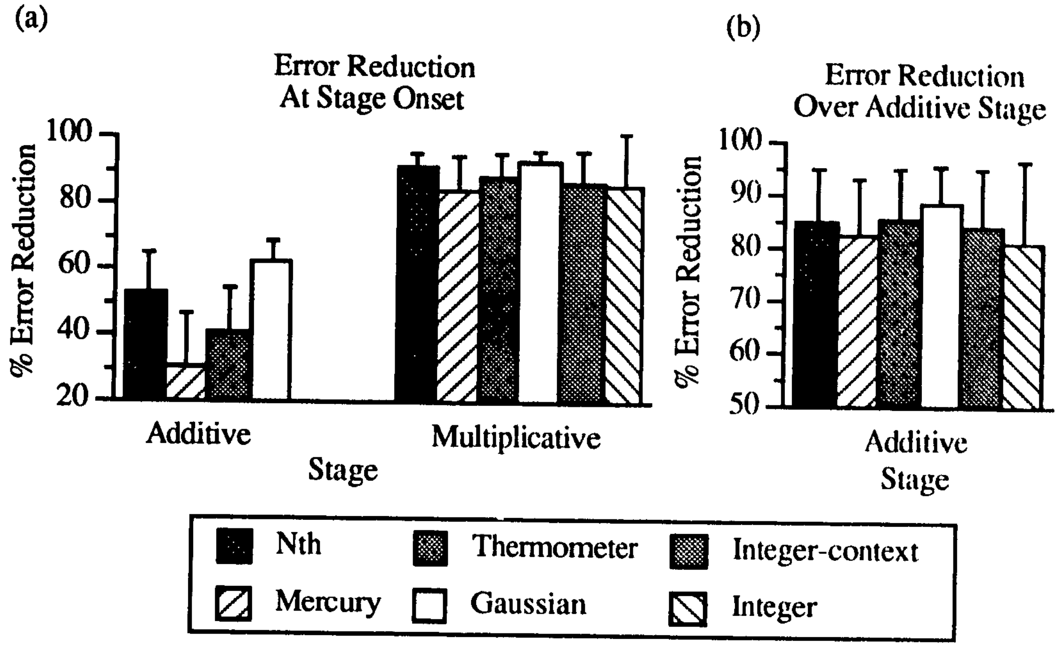


Figure 4. Mean percent of error reduction in distance inference patterns (a) from epoch 5 to additive and multiplicative stage onset and (b) over additive stage. Error bars are standard deviations from the mean.

the error that existed just prior to the onset of the stage was reduced by the end of the stage. The mean percent of error reduction over the additive stage relative to the epoch prior to stage onset is shown in Figure 4b for all encoding types.

Finally, the percentage of total error reduced from epoch 5 to the epoch prior to the onset of the multiplicative stage was approximately 90%. Taken together, the amount of error reduced over the additive stage and the total error reduction prior to the onset of the multiplicative stage suggests that while the additive rule may not be the normative rule, it is a good approximation.

Stability and Transitions

Overall the distance developmental sequence was extremely stable. Regressions were observed in only two networks, one with integer and one with integer-context encoding, and were not permanent. Both of these networks regressed to a tie between the additive and multiplicative rule. A tie consisted of a period of four or more testing epochs where the difference between the r^2 values of two rules, in this case the additive and multiplicative rules, was less than or equal to 0.99.

For the majority of networks, transition from the additive stage occurred in few epochs (M = 11.62, SD = 17.56). Typically this transition involved fluctuations between the additive and multiplicative rules. However, for 30% of the networks there was a period characterized by a tie between the additive and multiplicative rules before the onset of the multiplicative stage. On average this lasted for approximately 74.67 epochs (SD = 40.90). Note that ties often occurred just prior to the beginning of the input training phase. Thus the length of a tie is influenced by the fact that network responses do not change during the input training phase.

Thus, for the majority of networks there was a rapid transition between stages. Although some networks went through a tie before the onset of the multiplicative stage, the methodology used to diagnose stages was able to characterize this period as well.

Summary of Distance Developmental Sequence

For the majority of encoding types, the distance developmental sequence began with the additive stage following a period in which the error was reduced but not captured by identity ($d = t$ or $d = v$), additive, or multiplicative rules. For integer and integer-context networks the additive stage began almost immediately after training. During the additive stage, a large proportion of the variance in the outputs of the distance inference patterns was accounted for across encoding types. Finally, learning continued over the additive stage until the eventual onset of the multiplicative stage after either a brief transition period or a tie between the additive and multiplicative stages.

2.2. Time

With the exception of integer encoding, 92% of the networks exhibited a developmental sequence characterized by a progression from an *identity* stage ($t = d$) that started after about 5 to 10 epochs of training and lasted for approximately 55 epochs, to an *additive* stage ($t = d - v$) that lasted for an average of 80 epochs before attaining the correct *multiplicative* stage $t = d \div v$ after approximately 160 epochs of training. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, and the length of the stage are reported in Table 9 for each encoding type.

The remaining 8% of the networks (10 using integer-context and 2 using gaussian encoding) proceeded directly from the identity stage to the multiplicative stage. For the 10 integer-context networks, the identity stage started at 12.00 (*SD* = 6.75) epochs and lasted for 65.00 epochs (*SD* = 6.95) on average. None of the networks had recruited any hidden units at this time. The mean epoch of onset of the multiplicative stage was 96.00 (*SD* = 10.39). This occurred after all networks had recruited 1 hidden unit. For the two networks in the gaussian encoding condition, the identity stage began at 0 and 10 epochs and lasted until epoch 57 and 61 respectively. The multiplicative stage began at 82 and 66 epochs respectively . The networks' had not recruited any hidden units prior to the onset

Table 9
Time Developmental Sequence by Encoding Type: Pure Condition

		t=d			t=d-v			t=d÷v	
Encoding	*n*	hid	onset	length	hid	onset	length	hid	onset
Int-cont	20								
M		0.00	15.00	64.35	1.00	84.85	98.35	2.25	199.55
SD		0.00	6.49	8.52	0.00	10.02	55.03	0.55	51.31
Nth	30								
M		0.00	6.67	55.60	1.00	67.77	77.90	2.00	150.60
SD		0.00	2.40	5.40	0.00	4.95	7.87	0.00	8.26
Mercury	30								
M		0.00	10.17	67.40	1.00	85.07	95.23	2.03	189.83
SD		0.00	3.82	10.58	0.00	9.57	10.23	0.18	19.46
Therm	30								
M		0.00	5.50	56.30	1.00	66.80	80.87	2.00	152.50
SD		0.00	2.40	4.71	0.00	3.82	8.53	0 00	10.99
Gaussian	28								
M		0.00	6.79	54.75	1.00	66.89	81.25	2.07	153.04
SD		0.00	4.13	6.33	0.00	6.30	22.42	0.26	22.78

Note. d = distance; t = time; v = velocity, hid = number of hidden units, onset = epoch at which stage begins, length = length of stage in epochs

of the additive stage. One hidden unit was recruited prior to the onset of the multiplicative stage.

For all networks using integer encoding the first diagnosable stage was characterized by the additive rule $t = d + v$. The mean onset of the stage was at 4.50 epochs (*SD* = 2.01) and it lasted for 61.43 epochs (*SD* = 13.10) on average. None of the networks had recruited any hidden units before the onset of the stage. For 28 networks the next diagnosable stage was defined by the normative rule $t = d \div v$. The mean onset of the stage was at 75.21 epochs (*SD* = 13.42). All networks had recruited 1 hidden unit by the beginning of the stage. For the other two networks, the multiplicative stage was preceded by an additive stage defined by $t = d - v$. On average, this additive stage began at 71.50 epochs (*SD* = 21.92),

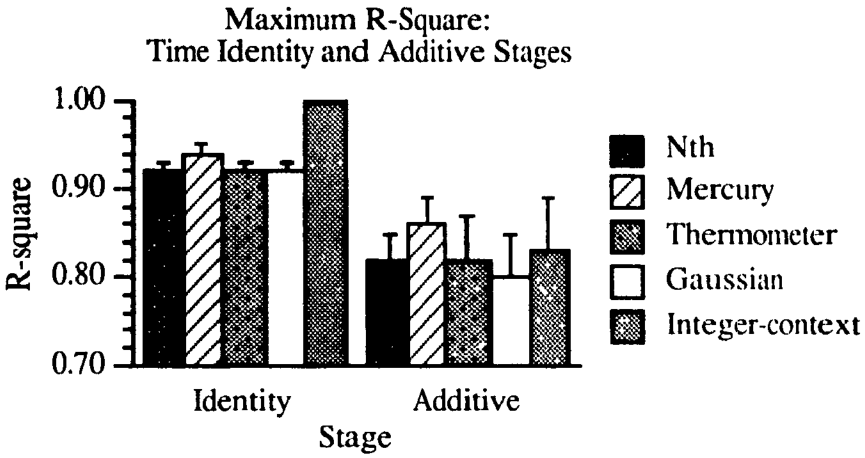


Figure 5. Maximum r^2 of t=d and t=d-v rules attained during identity and additive stages of time developmental sequence respectively.

was preceded by the recruitment of one hidden unit, and lasted for 81.5 epochs (SD = 3.54). The stage characterized by the normative rule began on average at 160.50 epochs (SD = 21.92) following the recruitment of a second hidden

Performance Attributable to Stage

For networks progressing through an identity stage (all encoding types except integer), a maximum of over 90% of variance in the output values of the time inference patterns was accounted for by the identity rule ($t = d$). In fact, all integer-context networks reached the maximum r^2 (1.00). The mean maximum r^2 of the additive rule ($t = d - v$) during the additive stage was slightly less but still accounted for more than 80% of the variance on average. Thus, as with the distance developmental sequence, a large percentage of the variance in the output values was accounted for. The mean maximum r^2 values attained during both identity and additive stages are shown in Figure 5.

For integer networks there was no identity stage. During the additive stage ($t = d + v$) the mean maximum r^2 value of the additive rule attained was 1.00 (SD = .01).

Finally, during the multiplicative stage the defining rule $t = d \div v$ reached an r^2 of 1.00 for all encoding types (see Table 7.).

Error Reduction

Each stage was associated with successive error reduction. The mean percent of error reduction over the identity stage and additive stage relative to the epoch prior to stage onset is shown in Figure 6a. As can be seen, approximately 35-55% of the error that existed prior to the onset of the identity stage was reduced over the stage. Slightly more error was reduced over the additive stage (50-65%). Thus, learning was continuous across the stages as in the distance developmental sequence.

The mean percent of error reduction from epoch 5 to the epoch prior to the onset of the multiplicative stage is shown in Figure 6b. On average, the amount of error reduced prior to the onset of the time multiplicative stage was slightly less than prior to the distance multiplicative stage. This suggests that the additive time rule was not as good an approximation of the time inference patterns as the analogous case was for distance patterns.

Stability and Transitions

Overall the time developmental sequence was again very stable. Regressions were observed in only 4 integer networks and 9 integer-context networks, and were never permanent. The regressions were to either the additive stage or to a tie

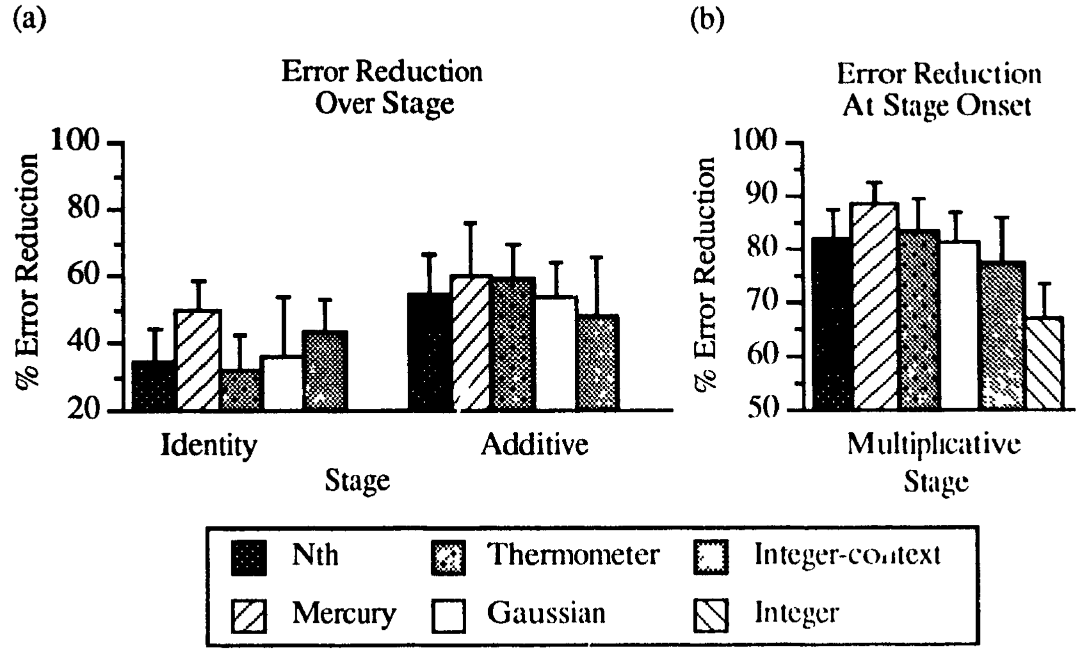


Figure 6. Mean percent of error reduction in time inference patterns (a) over identity and additive stages and (b) from epoch 5 to multiplicative stage onset. Error bars are standard deviations from the mean.

between the additive stage and the multiplicative stage.

Transition from the identity stage to the additive stage took 5.80 epochs on average (*SD* = 2.72). For almost all of the networks, the transition from the additive rule to the multiplicative rule was rapid taking 5.89 epochs on average (*SD* = 3.63). However, for 4 networks, there was a period between the additive and multiplicative stages characterized by a tie between the additive and multiplicative rules. On average this lasted for approximately 53.00 epochs (SD = 15.60).

Summary of Time Developmental Sequence

For all encoding types except integer, the time developmental sequence began with an identity stage followed by an additive stage and then the multiplicative stage. During the identity and additive stages, a large proportion of the variance in the outputs of the time inference patterns was accounted for by the defining rules. The transition between all stages was typically rapid. Finally, learning was continuous across all stages as indicated by analysis of error reduction.

For the majority of integer networks the first diagnosable stage was an additive stage defined by the rule $t = d + v$ that began soon after training and was followed by the normative multiplicative stage.

2.3. Velocity

Ninety-three percent of the networks (excluding integer) exhibited a developmental sequence characterized by a progression from an *identity* stage that started after approximately 5 to 10 epochs of training and lasted 55 to 60 epochs on average, to an *additive* stage that lasted for about 70 to 80 epochs, to the correct *multiplicative* stage after approximately 160 epochs of training. The identity, additive, and multiplicative stages were characterized by the rules $v = d$, $v = d - t$, and $v = d \div t$ respectively. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, and the length of the stage are reported in Table 10 for each encoding type.

The remaining 7% of the networks (10 using integer-context and 1 using gaussian encoding) proceeded directly from the identity stage to the multiplicative stage. For the integer-context networks, the identity stage started at 8.33 (*SD* = 5.59) epochs and lasted for 64.78 (*SD* = 7.45) epochs on average. None of the networks had recruited any hidden units at this time. The mean epoch of onset of the multiplicative stage was 96.44 (*SD* = 9.29). All networks had recruited 1 hidden unit at this point. For the network in the gaussian encoding condition, the identity stage began at 10 epochs and lasted 51 epochs. The

Table 10
Velocity Developmental Sequence by Encoding Type: Pure Condition

		v=d			v=d-t			v=d÷t	
Encoding	*n*	hid	onset	length	hid	onset	length	hid	onset
Int-cont	20								
M		0.00	14.00	67.25	1.00	86.75	130.85	2.53	234.90
SD		0.00	7.36	5.75	0.00	9.65	59.45	0.49	59.84
Nth	30								
M		0.00	6.67	55.43	1.00	67.77	77.93	2.00	150.27
SD		0.00	2.40	4.53	0.00	4.48	7.04	0.00	8.52
Mercury	30								
M		0.00	8.67	67.90	1.00	82.73	97.67	2.03	190.50
SD		0.00	4.14	9.45	0.00	8.21	8.02	0.18	20.78
Therm	30								
M		0.00	5.83	56.63	1.00	68.13	80.37	2.00	156.30
SD		0.00	1.90	4.82	0.00	4.18	9.55	0.00	15 44
Gaussian	29								
M		0.00	6.38	54.83	1.00	66.72	75.97	2.10	155.79
SD		0.00	2.64	4.77	0.00	5.99	8.82	0.31	20.47

Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

multiplicative stage began at 66 epochs after 1 hidden unit had been recruited.

For all 30 integer networks the first diagnosable stage was characterized by an additive stage defined by the rule $v = d + t$ followed by a multiplicative stage defined by the normative rule $v = d \div t$. The mean onset of the additive stage was at 4.50 epochs (*SD* = 2.01) and it lasted for 61.47 epochs (*SD* = 13.23) on average. None of the networks had recruited any hidden units before the onset of the additive stage. The mean onset of the normative stage was at 75.39 epochs (*SD* = 13.71). All networks had recruited 1 hidden unit by the beginning of this stage.

Performance Attributable to Stage

The mean maximum r^2 values attained during both identity and additive stages

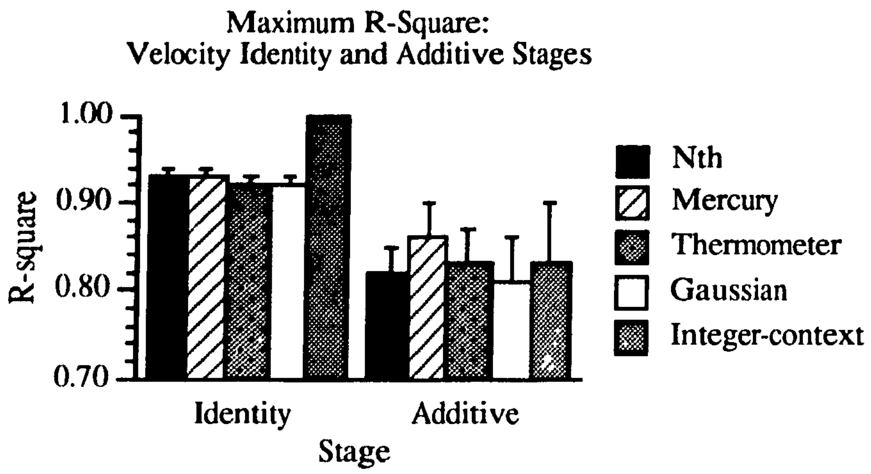


Figure 7. Maximum r^2 of v=d and v=d-t rules attained during identity and additive stages of velocity developmental sequence respectively.

are shown in Figure 7 for all encodings except integer. As in the time developmental sequence, a maximum of over 90% of variance in the output values was accounted for by the identity rule. Again, all integer-context networks reached the maximum r^2 (1.00). The mean maximum r^2 of the additive rule accounted for more than 80% of the variance on average.

For integer networks there was no identity stage. During the additive stage ($v = d + t$) the mean maximum r^2 value of the additive rule attained was 1.00 (SD = .01).

As reported in Table 7, for all networks except one using mercury encoding, the defining rule $v = d \div t$ of the multiplicative stage reached an r^2 of 1.00.

Error Reduction

The mean percent of error reduction over the identity stage and additive stage relative to the epoch prior to either stage onset are shown in Figure 8a for each encoding type. As can be seen, error reduction over the two stages was very

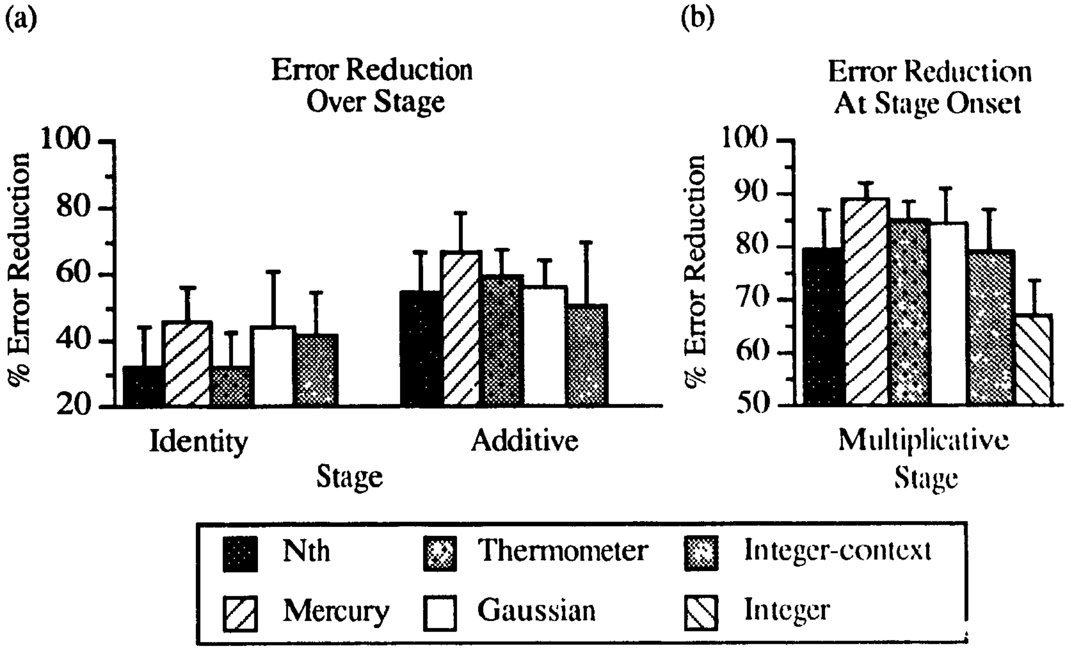


Figure 8. Mean percent of error reduction in velocity inference patterns (a) over identity and additive stages and (b) from epoch 5 to multiplicative stage onset. Error bars are standard deviations from the mean.

similar to that observed in the time developmental sequence. Approximately 30-50% of the error that existed prior to the onset of the identity stage was reduced over the stage. Slightly more error was reduced over the additive stage (55-70%).

The mean percent of error reduction from epoch five to the epoch prior to the onset of the multiplicative stage is shown in Figure 8b. Again this is very similar to the time developmental sequence. Thus, like the additive time rule, the additive velocity rule was not as good an approximation of the velocity inference patterns as was the case for distance patterns.

Stability and Transitions

Overall the velocity developmental sequence was very stable. Regressions were observed in only 4 integer and 7 integer-context networks, and were never permanent. The regressions were to either the additive stage or a tie between the additive stage and the multiplicative stage.

As with the time developmental sequence, transitions between stages of the velocity developmental sequence were rapid. The transition from the identity stage to the additive stage took 5.72 (*SD* = 2.45) epochs on average. For the vast majority of networks, the transition from the additive stage to the multiplicative stage took 6.02 epochs on average (*SD* = 3.10). However, for 6 networks, there was a period characterized by a tie between the additive and multiplicative rules before the attainment of the multiplicative stage. On average this lasted for approximately 60.67 epochs (*SD* = 12.61).

Summary of Velocity Developmental Sequence

The velocity developmental sequence was very similar to the one observed for time inference patterns. That is, for all encoding types except integer, a progression through identity, additive, and then multiplicative stages was observed. A large proportion of the variance in the outputs of the velocity inference patterns was accounted for during the identity and additive stages. The transition between all stages was rapid. Finally, learning was continuous across all stages as indicated by analysis of error reduction.

For the majority of integer networks the first diagnosable stage was an additive stage defined by the rule $v = d + t$ that began soon after training. This was followed by the onset of the normative multiplicative stage.

3. Inter-developmental Course

A stage by epoch plot of the three developmental sequences of a typical network is illustrated in Figure 9 for each encoding type. These networks were chosen as illustrative examples considering the mean onset and length of the stages, when hidden unit recruitment occurred, and the progression from one stage to another within each inference problem type. Statistical analyses were conducted to test the significance of any observed differences in mean stage onsets between the three concepts for each encoding type. Although qualitative similarities of the developmental course across the different types of input encoding were of interest, quantitative similarities (e.g., the number of epochs prior to the onset of the distance additive stage using mercury versus gaussian encoding) were not. Therefore, all tests conducted in this section were one-way repeated measures ANOVAs using an alpha level of .05.

3.1. Nth, Mercury, Thermometer, and Gaussian Ei coding

As can be seen in Figure 9 (c,d,e,f), the most common inter-developmental course for networks using nth, mercury, thermometer, and gaussian encoding involved a progression from the identity stages of time and velocity ($t = d$ and $v = d$), followed by the onset of the additive stage for all three concepts ($d = t + v$, $t = d - v$, $v = d - t$), followed by the onset of the multiplicative stages of time and velocity ($t = d \div v$ and $v = d \div t$) and finally, the multiplicative stage for distance ($d = t \times v$).

The means for all comparisons except for networks using gaussian encoding are those presented in Table 8, 10, and 11. Two gaussian networks "skipped" the additive time stage. Therefore comparisons are based on 28 networks rather than 30 as with the other encoding types.

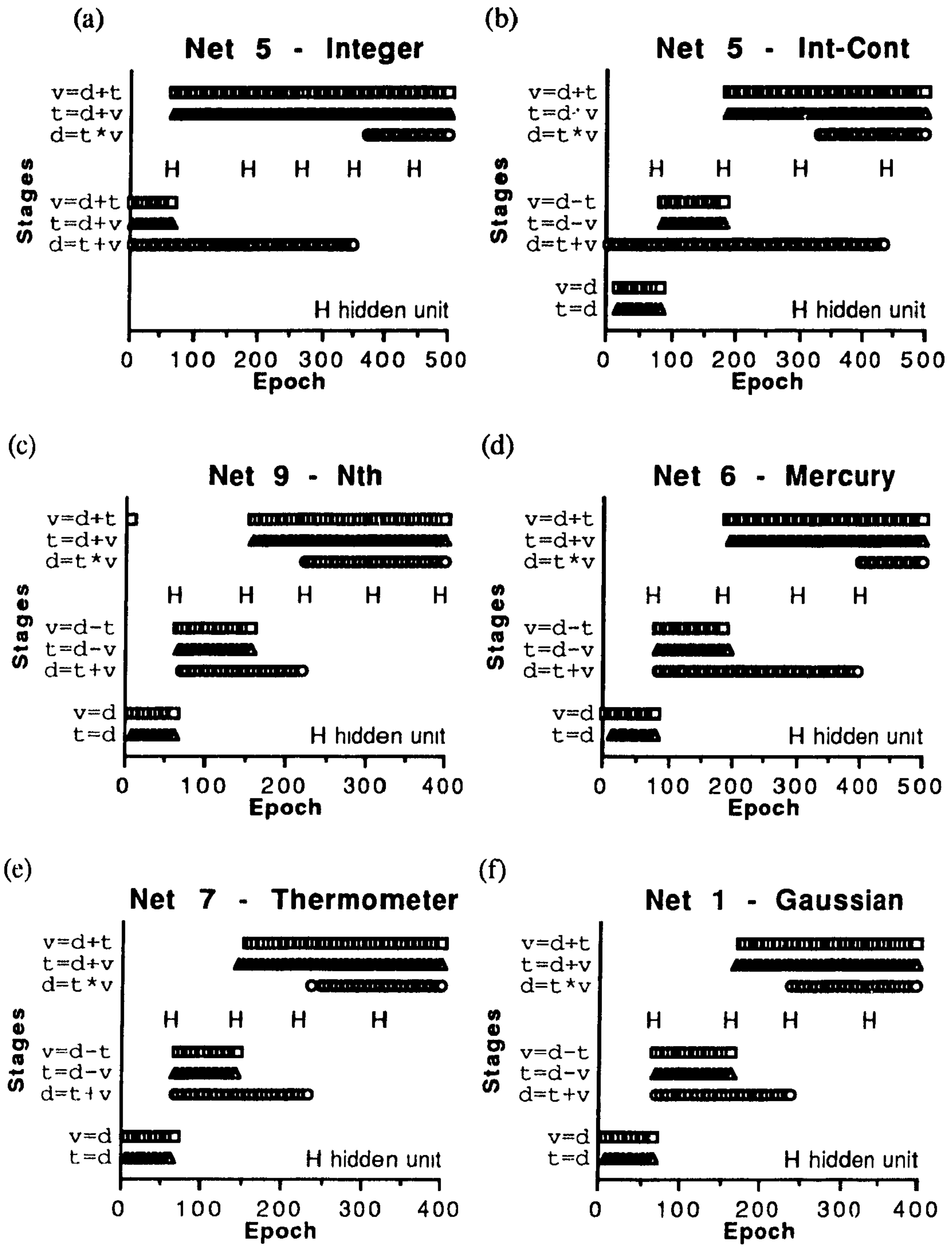


Figure 9 Pure Condition. Identity, additive, and multiplicative stages by epoch observed during training for (a) integer, (b) integer-context, (c) nth, (d) mercury, (e) thermometer and, (f) gaussian encoding. Epoch of hidden unit recruitment indicated by "H".

There were no significant differences between the onset of time and velocity identity stages for nth ($F_{(1,29)} = 0.00$); mercury ($F_{(1,29)} = 2.58$); thermometer ($F_{(1,29)} = 0.39$) and; gaussian encoding ($F_{(1,27)} = 0.33$, $F_{(1,27)} = 0.06$). For gaussian, the mean stage onset was at 6.79 ($SD = 4.13$) and 6.43 ($SD = 2.67$) epochs for time and velocity respectively.

Significant differences in onset of distance, time and velocity additive stages were observed for nth ($F_{(2,58)} = 15.19$); mercury ($F_{(2,58)} = 5.22$) and; thermometer encoding ($F_{(2,58)} = 5.69$). However, this is likely due to restricted variance. For gaussian encoding the mean onset of the additive stages were 66.00 ($SD = 5.54$), 66.89 ($SD = 6.30$), and 66.89 ($SD = 6.03$) for distance, time and velocity respectively. The differences were nonsignificant ($F_{(2,54)} = 2.17$).

The differences in onset of distance, time and velocity multiplicative stages were significant for all encodings: nth ($F_{(2,58)} = 37.95$); mercury ($F_{(2,58)} = 50.24$), thermometer ($F_{(2,58)} = 68.40$) and; gaussian encoding ($F_{(2,54)} = 62.68$). For gaussian encoding the mean onset of the multiplicative stages were 260.46 ($SD = 71.10$), 153.04 ($SD = 22.68$), and 153.93 ($SD = 18.16$) for distance, time and velocity respectively.

In order to ascertain how representative the mean data were of individual network performance the frequency of networks showing this inter-developmental pattern was calculated. For this purpose, a stage was considered to have started significantly later than another if the difference was a minimum of 20 epochs. The majority of networks in each encoding type did in fact follow the mean progression (nth, $n = 19$; mercury, $n = 20$; thermometer, $n = 24$ and; gaussian, $n = 21$). The majority of the remaining networks differed only in that the multiplicative stages of all three concepts began at the same time: nth n = 11; mercury, n = 5; thermometer, n = 4 and; gaussian, n = 5.

3.2. Integer-Context Encoding

Tv major inter-developmental courses were observed in the performance of networks using integer-context encoding. The first is depicted in Figure 9b. This accounted for the performance of 57% of the networks and was similar to the one just described with the exception that the distance additive stage began at approximately the same time as the identity stages of time and velocity.

On average, the identity stages of time and velocity began at 15.59 (SD = 6.59) and 14.41 (SD = 7.48) epochs respectively. The differences in onsets were nonsignificant ($F_{(1,16)}$ = 0.41). The distance additive stage began at 12.35 (SD = 19.29) epochs whereas the time and velocity additive stages began later, following the identity stages, at virtually the same point, 86.41 (SD = 10.08) epochs. The differences between the three additive stage onsets were significant ($F_{(2,32)}$ = 195.19). The mean onset of the distance, time, and velocity multiplicative stages were 417.41 (SD = 100.02), 208.53 (SD = 48.95), and 229.47 (SD = 58.14) respectively. The differences were significant ($F_{(2,32)}$ = 47.11).

Individual and statistical analysis agreed in terms of identity and additive stage onsets. For all but one network, the distance multiplicative stage began after the multiplicative stages of time and velocity.

The second typical inter-developmental course, characteristic of the performance of 23% of the networks, was similar except that the networks proceeded directly from the identity stage to the multiplicative stage of time and velocity. On average, the identity stages of time and velocity began at 11.43 (*SD* = 6.90) and 10.71 (*SD* = 6.08) epochs respectively. The difference was nonsignificant ($F_{(1,6)} = 0.04$). The mean epoch of onset were 516.00 (*SD* = 121.95), 92.71 (*SD* = 7.95), and 99.86 (*SD* = 11.77) for the distance, time, and multiplicative stages respectively. The differences were significant ($F_{(2.58)} = 136.16$).

Again, individual analysis and statistical analysis were very consistent. That is, for six of the seven networks, the identity stage onsets were the same. For one network, the velocity stage started slightly earlier. For all networks the distance multiplicative stage began later than the time and velocity multiplicative stages For five of the networks the time and velocity multiplicative stages began at the same time. For two networks the time stage began slightly earlier.

In networks not following either inter-developmental courses, the time or velocity multiplicative stage began before the other because of a progression through one additive stage but not the other.

3.3. Integer Encoding

For 93% of the integer networks the inter-developmental progression did not begin with an identity stage but instead began with additive stages before eventually attaining the time and velocity multiplicative stages and finally the distance multiplicative stage (Figure 3a.). Moreover, the additive stage of time and velocity were defined by the rules $t = d + v$ and $v = d + t$ respectively rather than $t = d - v$ and $v = d - t$. The mean epoch of onset of the additive stages were 4.11 (*SD* = 1.95), 4.46 (*SD* = 2.08), and 4.46 (*SD* = 2.08) for distance, time and velocity respectively. The differences were nonsignificant ($F_{(2,54)} = 0.79$). The mean onset of the distance, time, and velocity multiplicative stages were 354.00 (*SD* = 111.42), 75.21 (*SD* = 13.42), and 75.39 (*SD* = 13.71) epochs respectively. The differences were significant ($F_{(2,54)}$ = 175.54).

Individual network analysis revealed that this progression was seen in all 28 networks. The two exceptions to this inter-developmental course attained the multiplicative rule of velocity, then distance and time together.

3.4. Summary: Inter-developmental Course

Statistical and individual analysis of network performance revealed that for the majority of networks using nth, mercury, thermometer, and gaussian encoding, the inter-developmental course was as follows. First, time and velocity identity stages began at the same time early in training. Next, all three additive stages began. Although the differences in onsets were statistically significant, individual analysis revealed that for all intents and purposes the three stages began at the same time. Finally, the onset of the multiplicative stages of time and velocity began at approximately the same time. Moreover both began before the distance multiplicative stage.

This same sequence was observed in the majority of integer-context networks whereas no integer networks performed in this manner. It is likely that the differences in inter-developmental courses across encodings reflects the linear nature of the encodings themselves. Unlike other encoding types, with integer and integer-context encoding the magnitude of input values are in some sense given to the networks without prior training. For example, an input value of 5 will cause more activation than an input of 1. Thus, a time inference pattern with distance and velocity input values of 2 and 3 respectively will produce a smaller output value (or larger depending on the sign of the weights) than one with distance and velocity values of 5 and 5 respectively. Moreover, the difference between time or velocity inference patterns will reflect the linear nature of the inputs. This, in turn makes additive rules such as $t = d + v$ and $v = d + t$ possible. Conversely, with the other encoding types, for example nth, there is no a priori information concerning the magnitude of the input values since a value of 1 is assigned to the appropriate input unit regardless of the magnitude of the input value being encoded.

For the most linear encoding, integer, the early onset of additive stages defined by $t = d + v$ and $v = d + t$ seems to inhibit the onset of additive stages defined by $t = d - v$ and $v = d - t$. This was partly overcome by the use of context units (integer-context encoding).

4. Hidden Unit Recruitment and Stage Onset

As can be seen in Figure 9, transition from identity to additive and additive to multiplicative stages typically occurred quickly after the recruitment of a hidden unit. The mean epochs from hidden unit recruitment to additive and multiplicative stage onset are reported in Table 11. In general, stage transition occurred within 5 to 10 epochs of hidden unit installation. This would seem to indicate that hidden unit recruitment was very instrumental in stage transition.

4.1. Hinton Analysis

Hinton diagrams were drawn to understand the nature of the relationship of hidden unit recruitment to stage transition. A description of how Hinton diagrams are read is presented here. Examples of Hinton diagrams are presented in Figures 10 and 11.

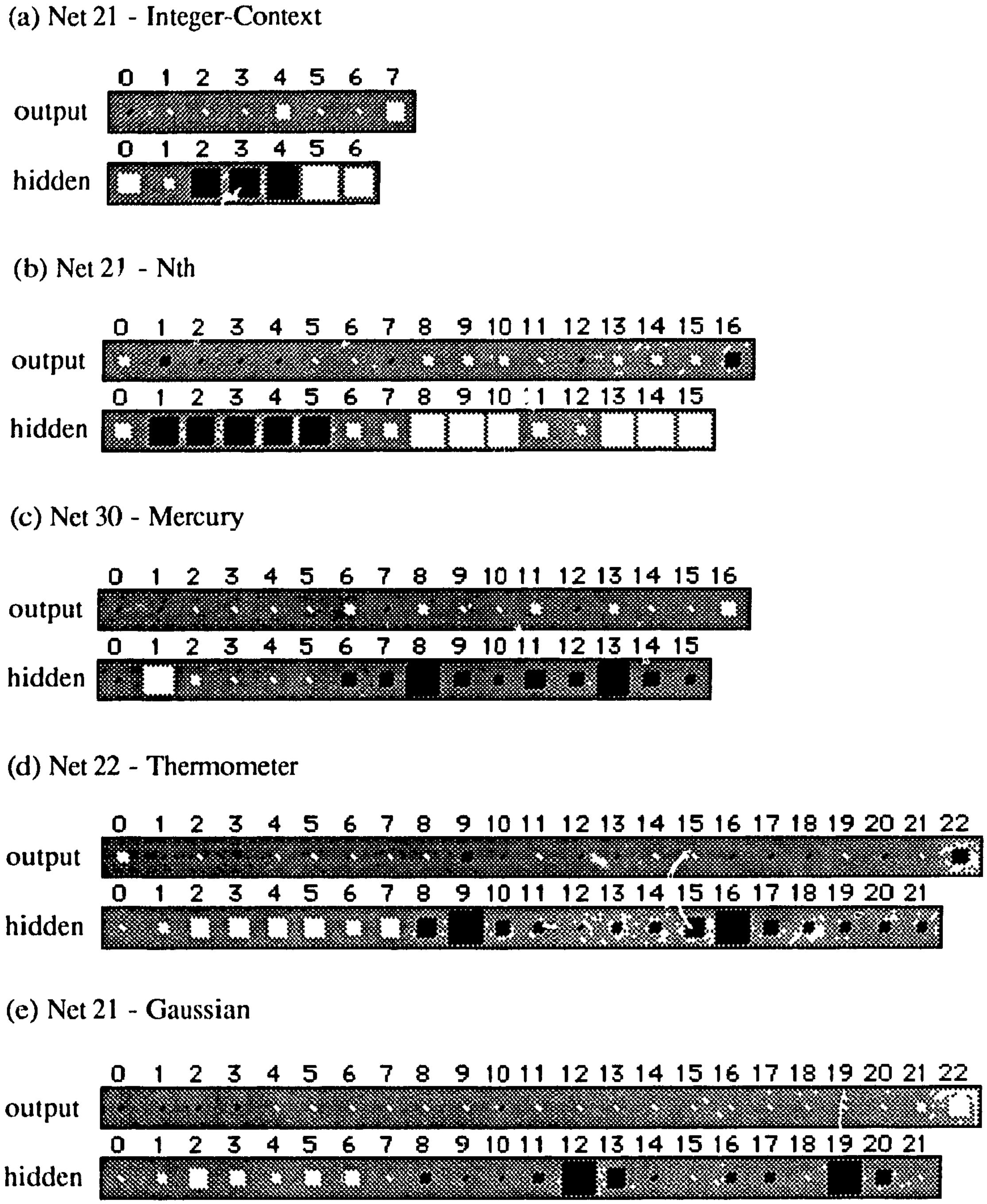
Table 11
Epochs From Hidden Unit Recruitment to Stage Onset: Pure Condition

	Additive Stage			Multiplicative Stage		
Encoding	d=t+v	t=d-v	v=d-t	d=t*v	t=d÷v	v=d÷t
Integer						
n		2		30	30	30
M		10.00		15.30	8.33	9.17
SD		7.07		13.22	2.84	3.50
Int-cont						
n	2	20	19	30	30	30
M	5.00	7.50	7.63	16.80	15.00	12.67
SD	0.00	2.57	2.57	26.07	12.93	11.80
Nth						
n	30	30	30	30	30	30
M	9.50	6.33	6.33	10.00	5.17	4.83
SD	1.53	3.20	2.92	7.88	3.08	3.59
Mercury						
n	30	30	30	30	30	30
M	5.67	8.50	6.17	9.50	8.83	9.17
SD	1.73	4.58	3.40	12.06	5.36	2.65
Therm						
n	30	30	30	30	30	30
M	5.17	5.17	6.50	6.87	5.00	6.33
SD	0.91	0.91	2.68	9.58	3.47	4.72
Gaussian						
n	30	28	29	30	30	30
M	5.20	6.11	6.07	9.47	5.67	5.83
SD	0.93	2.56	2.45	11.23	6.26	3.96

Note. d = distance; t = time; v = velocity.

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 colour (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 encoding, the squares numbered 1, 2, and 3, represent the weights



(a) Net 21 - Integer-Context

(b) Net 21 - Nth

(c) Net 30 - Mercury

(d) Net 22 - Thermometer

(e) Net 21 - Gaussian

Figure 10. 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) nth, (c) mercury (d) thermometer, and (e) gaussian encoding.

from the distance, time, and velocity input units respectively. With integer-context encoding, the additional squares numoered 4, 5, and 6 represent the distance, time, and velocity context units. For nth and mercury encoding, 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 encoding, squares 1-7, 8-14, and 15-21 represent the weights from the distance, time, and velocity input groups. For all encoding types, the square numbered 0 is for the bias unit whereas the last square in the output row is for 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 11a, the square numbered 4 of the second hidden unit row represents the weight from the first hidden unit to the second.

Transition to Additive Stages

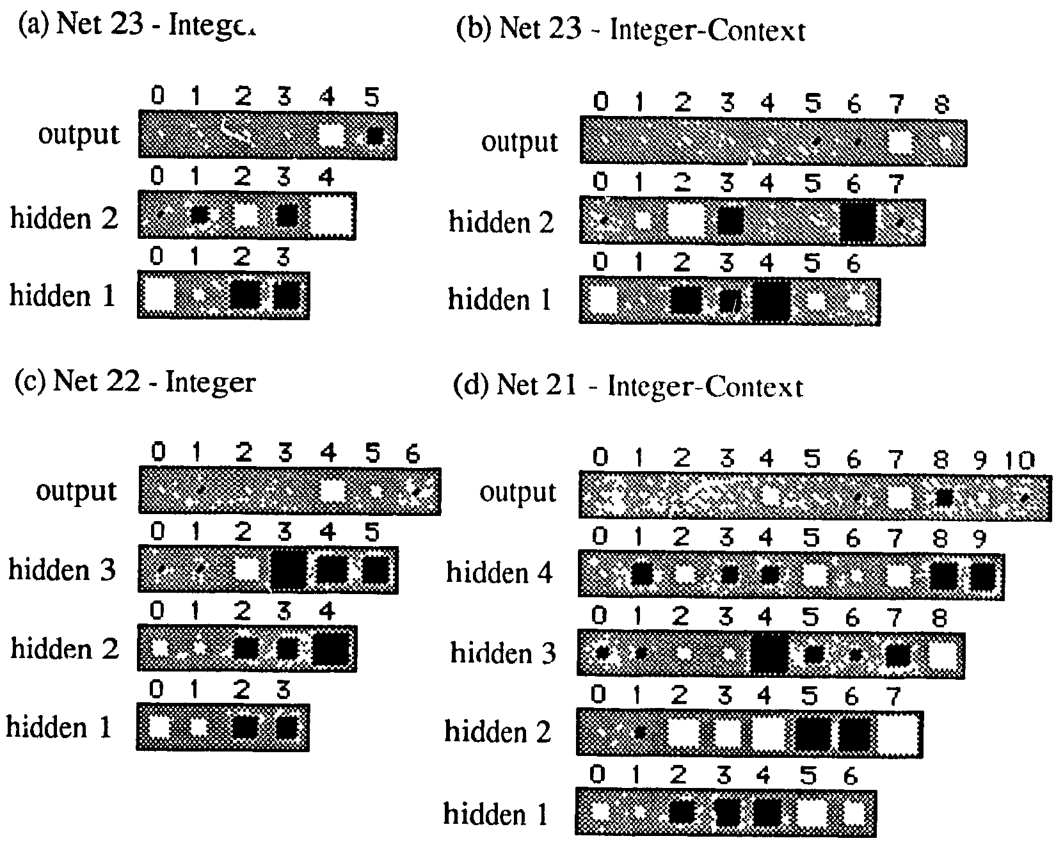
Hinton diagrams proved useful in understanding how the distance, time, and velocity additive stages of networks using nth, mercury, thermometer, and gaussian encoding emerged either at the time that the first hidden unit was recruited or shortly thereafter. For networks using integer-context encoding, the first hidden unit was associated with the additive stages of time and velocity. The additive stage of distance began prior to hidden unit recruitment. For the majority of networks (78% across the 5 encoding types) 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 10 for each encoding type.

As can be seen, the first hidden unit distinguishes distance input units from time and velocity units. That is, the weights from time and velocity input units have the opposite sign from the weights from the distance input units. When the hidden unit is activated by the presentation of a distance inference pattern, information from the time and velocity input groups will augment each other since the direction of their weights are the same. Conversely, since 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 will counter 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.

Transition to Multiplicative Stages

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 encoding type, a consistent pattern was observed for the transition to the distance multiplicative stage. In 19 of the 20 networks across the two encoding types, performance characterized by the normative distance rule only emerged after a hidden unit was recruited which received opposite-sign weights for time and velocity input groups.

Hinton diagrams of two networks using integer and integer-context encoding are presented in Figure 11. As can be seen in Figures 11a and 11b, the second hidden unit has opposite-sign weights (squares 2 and 3). The multiplicative stage was subsequently observed. For the networks depicted in Figures 11c and 11d, the multiplicative stage was observed after the installation of the third and fourth hidden



(a) Net 23 - Integer

(b) Net 23 - Integer-Context

(c) Net 22 - Integer

(d) Net 21 - Integer-Context

Figure 11. 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.

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. Although not as intuitive as the explanation provided above with respect to the transition to additive stages, it appears that as long as the weights from the time and velocity input groups continue to have the same-sign weights, only additive rules for distance inference patterns are possible. Only when hidden units that treat the time and velocity input as different, with respect to weight sign, is the multiplicative rule observed.

LIMITED MEMORY CONDITION

The limited memory condition was designed to investigate Wilkening's (1981) hypothesis that children's retarded performance on the velocity task relative to the time task reflected the increased memory demands of the velocity task. In the following simulations the probability of presenting the network the correct time information of a given velocity inference pattern was manipulated to simulate the possible difficulties children may have had in remembering the time information.

1. General Learnability

As in the pure condition, all networks trained for the maximum number of epochs (1500) without declaring victory. The mean sum-squared error and the percent of error reduced from epoch 5 to the end of training for each inference problem type are reported in Table 12.

The error at the end of training was greater for velocity patterns than either time or distance inference patterns. A one-way repeated measures ANOVA for each encoding type revealed that the differences were significant ($p < .05$): integer $F_{(2,58)} = 93.96$; integer-context $F_{(2,58)} = 117.11$; nth $F_{(2,58)} = 265.09$; mercury $F_{(2,58)} = 127.07$; thermometer $F_{(2,58)} = 104.47$ and; gaussian $F_{(2,58)} = 214.56$.

Across all encodings approximately 97%, 98%, and 87% of the error that existed at epoch 5 was reduced by the end of training for distance, time, and velocity inference patterns respectively. The differences were significant ($p < .05$): integer $F_{(2,58)} = 45.76$; integer-context $F_{(2,58)} = 62.74$; nth $F_{(2,58)} = 136.77$; mercury $F_{(2,58)} = 112.01$; thermometer $F_{(2,58)} = 42.47$ and; gaussian $F_{(2,58)} = 171.00$.

Table 12
Mean Error and Percent Reduction in Error at End of Training: Limited Memory Condition

	Inference Problem Type					
	Distance		Time		Velocity	
Encoding	*E*	%	*E*	%	*E*	%
Integer						
M	1.67	92.65	1.71	95.68	5.13	88.05
SD	0.92	3.84	0 82	1.99	1.68	4.17
Int-cont						
M	1.33	93.59	1.44	96.55	6.03	85.16
SD	0.47	3.44	1.05	2.84	2.57	6.88
Nth						
M	0.41	98.86	0.34	98.81	5.01	85.20
SD	0.18	0.59	0.14	0.56	1.58	6.42
Mercury						
M	0.89	96.99	0.62	98.20	4.56	88.08
SD	0.38	1.43	0.25	0.77	1.77	4.59
Therm						
M	0.50	98.56	0.48	98.41	4.93	86.43
SD	0.11	0.45	0.14	0.72	1.40	5.84
Gaussian						
M	0.81	97.89	0.64	97.92	4.84	86.89
SD	0.32	0.93	0.32	1.06	1.62	4.98

n = 30

Thus, at this macro level of analysis, the limited memory manipulation affected network performance in the desired direction causing the velocity inference patterns to be more difficult to learn.

2. Stage Diagnoses

The majority of networks progressed through qualitatively similar or identical developmental sequences as those observed in the pure condition. Ninety-seven percent of networks, collapsed across all encoding types, exhibited the same distance developmental sequence ($d = t + v$ followed by $d = t \times v$). Ninety-three percent of the networks (excluding integer encoding) followed the same time developmental course progressing from the identity stage ($t = d$) to the additive stage ($t = d - v$) and the multiplicative stage ($t = d \div v$). Similarly, 83% of the networks (excluding integer encoding) followed the same progression from an identity ($v = d$) to an additive stage ($v = d - t$). However, unlike in the pure condition, several of these networks (23%) did not progress beyond the additive stage. The remaining networks went on to the multiplicative stage ($v = d \div t$).

For the sake of brevity, the amount of variance in actual outputs accounted for by a given rule during a stage is not discussed in the following sections since the results of this analysis proved similar to that in the pure condition. That is, the maximum amount of variance in network responses accounted for by the defining rules of the stages greatly exceeded the minimum set by the *best rule* criterion ($r^2 \geq$ 0.50). Likewise, the error reduction prior to and across stages was similar to that observed in the pure condition in that a large proportion of error was reduced. The interested reader may consult Appendices A and B respectively for a summary of the limited memory findings.

2.1. Distance

For networks using all encodings except integer, the additive stage began after the recruitment of one hidden unit and it lasted 300-400 epochs on average. The Multiplicative stage began at approximately 500 to 600 epochs following the recruitment of a fourth hidden unit on average. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, and the length of the stage for the distance developmental sequence are reported in Table 13. For the

Table 13
Distance Developmental Sequence by Encoding Type. Limited Memory Condition

		d=t+v			d=t*v	
Encoding	*n*	hid	onset	length	hid	onset
Integer	30					
M		0.00	4.83	483.97	3.83	611.27
SD		0.00	0.91	221.08	1.86	275.38
Int-cont	30					
M		1.00	146.67	436.40	4.20	668.50
SD		0.00	5.79	225.07	1.67	243.18
Nth	28					
M		1.00	155.86	317.46	3.50	545.39
SD		0.00	5.10	159.70	1.37	197.80
Mercury	29					
M		1.00	146.83	445.24	4.00	631.17
SD		0.00	9.06	163.08	1.31	186.80
Therm	27					
M		1.00	143.89	297.63	2.97	457.87
SD		0.00	15.46	158.52	1.35	200.20
Gaussian	30					
M		1.00	144.23	424.20	4.27	645.17
SD		0.00	9.90	227.36	1.64	241.79

Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

five networks that do not appear in Table 13, the multiplicative stage was the first diagnosable stage.

One difference between the pure and limited memory conditions is that networks using integer-context encoding required the recruitment of a hidden unit before the onset of the additive stage in the limited memory condition whereas in the pure condition no hidden unit was necessary. Moreover, an identity stage was observed for all 30 integer-context networks defined by $d = t$ before the additive stage.

For networks using integer encoding, the additi'/e stage began almost as soon as training began as was the case in the pure condition. After approximately 600 epochs, the multiplicative stage emerged.

Stability and Transitions

The distance developmental sequence was stable although not as stable as in the pure condition. For 80% of the networks across all encoding types the additive best-rule classification was constart. That is, at each testing epoch within the stage, the rule that accounted for the most variance in network responses was the additive rule. For the remaining networks, 11.00% (*SD* = 13.60%) of the epochs within the additive stage were classified as either multiplicative or were not diagnosable. Regressions after the attainment of the multiplicative stage were rare and never permanent. Only 9% percent of the networks regressed and it was always to a tie between the additive and multiplicative rules.

Unlike in the pure condition, the majority of the networks (60%) across encoding types went through a period characterized by a tie between the additive and multiplicative rules before the attainment of the multiplicative stage. On average this lasted for approximately 86.97 epochs (*SD* = 76.61). For the remaining 40% of the networks, the transition from the additive to multiplicative stage took 17.85 epochs (*SD* = 16.92).

Summary of Distance Developmental Sequence

As in the pure condition, for networks using nth, mercury, thermometer, or gaussian encoding, the distance developmental sequence entered an additive stage following an initial period not captured by either identity, additive, or multiplicative rules. However, unlike the pure condition, this initial period was associated with an identity stage defined by the rule $d = t$ for integer-context networks. For integer networks the additive stage began almost immediately after training. For all networks across encoding types, the multiplicative stage emerged following either a fairly short transition or a period characterized by a tie between additive and multiplicative rules.

In general, the major difference between the pure and limited memory conditions is in terms of later starting and longer lasting stages. This difference likely results from two related factors. First, the overall sum-squared error is less stable than in the pure condition due to the random nature of recall in the *velocity* inference patterns. As such, the output-patience (the maximum number of successive epochs that result in nonsignificant changes in the sum-squared error) is less likely to be achieved and thus output training is more likely to go until the outlimit is reached (100 epochs). Second, the weights from the time input group are less precise since they are used to propagate information for time input in both the distance inference patterns and in the velocity inference patterns. This imprecision of the time input weights may have obscured the relationship of time and velocity information in terms of distance inferences making it more difficult for the network to find an appropriate set of weights.

For some networks the transition from the additive to multiplicative stage was quick as in the pure condition. However, for the majority of networks there was a tie before the attainment of the multiplicative stage. Again this is likely attributable to the imprecision of weights due to the limited memory manipulation.

In general, the limited memory manipulation increased the variability across networks of a given encoding as indicated by the large standard deviations found in Table 13. However, this variability was limited to the length of the additive stage and subsequent onset of the multiplicative stage and did not affect the developmental sequence qualitatively.

2.2. Time

For the majority of networks the time developmental sequence was qualitatively similar to that observed in the pure condition. Ninety-three percent of the networks (excluding integer networks) progressed from the identity stage ($t = d$) that started after five to ten epochs of training and lasted for approximately 130 epochs, to the additive stage ($t = d - v$) that lasted 150 epochs on average before attaining the correct multiplicative stage $t = d \div v$ after approximately 340 epochs of training. As in the pure condition, the additive stage emerged after the first hidden unit was installed. Moreover, the onset of the multiplicative stage typically followed the installation of a second hidden unit. Occasionally three hidden units had been recruited before onset. The mean number of hidden units recruited prior to stage onset, the epoch at which the stage began, and the length of the stage are reported in Table 14.

The remaining 10 networks (9 using integer-context and 1 using thermometer encoding) proceeded directly from the identity stage to the multiplicative stage. For the 9 integer-context networks, the identity stage started at 15.00 (*SD* = 5.00) and lasted for 125.67 epochs (*SD* = 5.55) on average. None of the networks had recruited any hidden units at this time. The mean epoch of onset of the multiplicative stage was 180.11 (*SD* = 22.23). This occurred after all networks had

Table 14
Time Developmental Sequence by Encoding Type: Limited Memory Condition

		t=d			t=d-v			t=d÷v	
Encoding	*n*	hid	onset	length	hid	onset	length	hid	onset
Int-cont	21								
M		0.00	15.95	125.24	1.00	146.31	168.33	2.19	349.95
SD		0.00	4.64	11.03	0.00	9.59	104.17	0.68	102.05
Nth	30								
M		0.00	7.50	147.23	1.00	163.57	156.13	2.13	330.93
SD		0.00	2.54	15.01	0.00	15.26	46.19	0.35	50.94
Mercury	30								
M		0.00	8.67	134.83	1.00	152.17	151.77	2.10	333.10
SD		0.00	3.70	7.09	0.00	9.06	19.09	1.63	46.16
Therm	29								
M		0.00	5.86	138.83	1.00	151.07	166.31	2.35	347.76
SD		0.00	2.70	16.05	0.00	16.37	49.51	0.55	71.59
Gaussian	30								
M		0.00	5.83	135.53	1.00	146.87	173.90	2.27	339.30
SD		0.00	1.90	9.73	0.00	9.74	60.74	0.45	72.20

Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

recruited 1 hidden unit. For the network in the thermometer encoding condition, the identity stage began at 5 epochs and lasted for 138 epochs. The multiplicative stage began at 158 epochs after 1 hidden unit had been recruited.

As in the pure condition, the first diagnosable stage for all networks using integer encoding was characterized by the additive rule $t = d + v$. This occurred before a hidden unit had been recruited. The mean onset of the stage was at 4.88 epochs (*SD* = 1.60) and it lasted 135.50 epochs (*SD* = 6.48) on average. Twenty-seven of the networks then progressed directly to the multiplicative stage ($t = d \div v$). The mean onset of the stage was at 172.00 epochs (*SD* = 49.05). The networks had recruited 1.15 hidden units on average (*SD* = 0.36) by the beginning of the stage. The remaining three networks progressed from the additive stage defined by $t = d + v$ to one defined by $t = d - v$ before the multiplicative stage onset. On average, the additive stage ($t = d - v$) began at 251.00 epochs (*SD* = 101.30), was preceded by the recruitment of one or two hidden units, and lasted for 138.33 epochs (*SD* = 26.54). For one network the additive stage was preceded by the identity stage ($t = d$). It lasted for 140 epochs. For all three networks, the multiplicative stage began on average at 394.66 epochs (*SD* = 98.72) following the recruitment of a second hidden or third hidden unit.

Stability and Transitions

Overall the time developmental sequence was again very stable. The identity best-rule classification was constant across encoding types with the exception of one network in which 2% of the epochs were undiagnosible. With respect to the additive stage, for 90% of all networks the additive best-rule classification was constant. For the remaining networks, 18.93% (SD = 13.61%) of the epochs within the additive stage were not classified as additive. These epochs were typically classified as multiplicative. Occasionally, they were classified as regressions to identity rules or undiagnosible. Only 7% percent of the networks regressed after the attainment of the multiplicative stage. The regressions were to the additive stage and/or to a period characterized by a tie between the additive and multiplicative rules. The regressions were never permanent.

The transition from the identity stage to the additive stage took 7.07 (*SD* = 5.22) epochs on average across encoding types. For the majority of the networks, the transition from the additive stage to the multiplicative stage took 10.52 epochs (*SD* = 10.38) on average. For the remaining networks (14%), there was a tie preceding the multiplicative stages between the additive and multiplicative rules. This lasted for approximately 62.15 epochs (SD = 49.39).

Summary of Time Developmental Sequence

The time developmental sequence observed in the limited memory condition for a given encoding type was the same as the one seen in the pure condition. For all encoding types except integer, it began with the identity stage followed by the additive stage and then the multiplicative stage. For the majority of integer networks the first diagnosable stage was the additive stage ($t = d + v$) that began soon after training and was followed by the multiplicative stage.

In general, the major difference between the pure and limited memory conditions is in terms of later starting stages that last longer on average. Again it seems reasonable to assume that this is due to the termination of the output training phase because of the outlimit being reached rather than the output patience. Also, there was a slight increase in the variability between networks of the same encoding.

2.3. Velocity

The most common velocity developmental sequence was the same as the one demonstrated in the pure condition. That is, 63% of the networks (excluding integer) progressed from the identity stage ($v = d$) that started after approximately 10 to 15 epochs of training and lasted 100 to 200 epochs on average, to an additive stage ($v = d - t$) that lasted for about 300 to 400 epochs, to the correct multiplicative stage ($v = d \div t$) after approximately 600 epochs of training. The mean number of hidden units recruited prior to stage onset, the epoch at which the

Table 15
Velocity Developmental Sequence 1 by Encoding Type: Limited Memory Condition

		v=d			v=d-t			v=d÷t	
Encoding	*n*	hid	onset	length	hid	onset	length	hid	onset
Int-cont	21								
M		0.00	23.81	120.24	1.00	149.10	367.00	3.43	561.10
SD		0.00	17.17	15.10	0.00	5.18	260.10	1.72	279.69
Nth	18								
M		0.00	11.67	191.17	1.28	211.28	434.17	4.78	733.94
SD		0.00	4.85	60.41	0.46	65.43	330.06	2.67	402.54
Mercury	12								
M		0.00	12.08	138.17	1.00	167.75	276.50	3.25	535.08
SD		0.00	6.20	16.20	0.00	28.76	201.96	1.49	246.71
Therm	23								
M		0.00	9.35	179.26	1.32	202.78	425.74	4.39	676.57
SD		0.00	5.90	86.61	0.57	83.93	278.29	1.92	294.30
Gaussian	21								
M		0.00	10.48	133.62	1.14	185.81	446.86	4.33	679.48
SD		0.00	7.89	46.11	0.36	57.62	323.34	2.03	295.37

Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

stage began, and the length of the stage are reported in Table 15. As can be seen, the additive stage emerged typically after one hidden unit had been recruited. However, some networks required two hidden units before the onset of the additive stage. The most striking difference between the pure and limited memory conditions is the number of hidden units installed before the onset of the multiplicative stage. On average, 4 hidden units had been installed whereas only 2 had been installed in the pure condition.

Twenty percent of the networks (excluding integer) behaved as Wilkening's subjects (1981, 1982) progressing from the identity stage to the additive stage but not to the multiplicative stage. The mean number of hidden units recruited prior to

Table 16
Velocity Developmental Sequence 2 by Encoding Type: Limited Memory Condition

		v=d			v=d-t	
Encoding	*n*	hid	onset	length	hid	onset
Int-cont	6					
M		0.00	16.67	131.50	1.00	155.67
SD		0.00	7.53	12.50	0.00	11.64
Nth	11					
M		0.00	8.18	218.00	1.55	246.46
SD		0.00	3.37	96.72	0.69	98.69
Mercury	8					
M		0.00	13.13	135.63	1.00	155.00
SD		0.00	7.04	14.42	0.00	12.08
Therm	2					
M		0.00	12.50	195.00	2.00	238.00
SD		0.00	10.61	169.71	0.00	117.38
Gaussian	2					
M		0.00	7.50	181.50	1.50	233.50
SD		0.00	3.54	136.47	0.71	120.92

Note. d = distance; t = time; v = velocity; hid = number of hidden units; onset = epoch at which stage begins; length = length of stage in epochs.

stage onset, the epoch at which the stage began, and the length of the stage are reported in Table 16.

Taken together, more than 80% of the networks across encoding types (excluding integer) followed a progression from the identity stage to the additive stage. The majority of these networks then progressed to the multiplicative stage. However, several networks did not progress beyond the additive stage. Finally, with respect to the networks that did not demonstrate either of these two sequences (17% of all networks excluding integer), the majority progressed from the identity stage to the multiplicative stage.

As in the pure condition, the first diagnosable stage for all networks using integer encoding was characterized by the additive rule $v = d + t$. On average it began at 4.50 epochs (*SD* = 2.01) before any hidden units had been recruited and lasted for 135.83 epochs (*SD* = 6.31). Twenty-one of the networks then progressed directly to the multiplicative stage ($v = d \div t$) after having recruited 1.05 hidden units on average (SD = 0.22). The mean onset of the stage was at 167.95 epochs (*SD* = 30.66). The nine networks progressed from the additive stage defined by $v = d + t$ to one defined by $v = d - t$ followed by the correct multiplicative rule. On average, the additive stage ($v = d - t$) began at 153.67 epochs (*SD* = 13.86), was preceded by the recruitment of one hidden unit, and lasted for 187.33 epochs (*SD* = 64.38). The multiplicative stage for all nine networks began at 366.22 epochs (*SD* = 72.23) following the recruitment of two or three hidden units.

Stability and Transitions

For the majority of networks (84%) the identity best-rule classification was constant across all encoding types. For the remaining networks 13.27% (*SD* = 8.80%) of the epochs were either undiagnosible or classified as additive. For 45% of the networks the additive best-rule classification was constant across all encoding types. For the remaining networks, 13.68% (SD = 12.69%) of the epochs within the additive stage were undiagnosible.

Regressions after the attainment of the multiplicative stage were more frequent than in either the distance or time developmental sequences. For all encodings excluding integer, 51% of the networks progressing from the identity and additive stage to the multiplicative stage did not regress. Twenty-four percent regressed to a tie between additive and multiplicative rules or to the additive stage but then recovered. Fourteen percent of the networks regressed permanently -- the majority to the additive stage, the others to a tie between the additive and multiplicative rules. The remaining networks regressed to oscillations between the additive and multiplicative stages.

For 76% of networks progressing from the identity stage to the additive stage but no further, stage diagnosis was stable. Of the remaining networks, a tie between the additive and multiplicative rules followed the additive stage. However, for the majority of the networks this period was temporary and followed by a return to the additive stage.

Thirty percent of networks using integer encoding regressed permanently to the additive stage $v = d - t$. Thirteen percent regressed permanently to a tie between the additive and multiplicative stage. Seventeen percent regressed to a tie between additive and multiplicative rules and/or the additive stage but then recovered. Finally, 40% of the networks did not regress.

In general, transitions between stages took longer between velocity stages than time stages. Across all encoding types, the transition from the identity stage to the additive stage took 16.73 epochs on average (*SD* = 25.79). For the majority of the networks the transition from the additive stage to the multiplicative stage took 10.52 epochs on average (*SD* = 10.38). However, for 44% of the networks there was a period characterized by a tie between the additive and multiplicative rules before the attainment of the multiplicative stage. On average this lasted for 43.78 epochs (SD = 25.07).

Summary of Velocity Developmental Sequence

Although the limited memory manipulation increased the variability both in terms of when stages emerged across networks using the same encoding and in terms of less stable stages within individual networks, a similar developmental sequence to the one observed in the pure condition was demonstrated by the majority of networks. However, this similarity held only up to a point. That is, across all encoding types except integer encoding, the majority of networks progressed from the identity stage that began early on in training to the additive stage. However, the limited memory manipulation affected the attainment and stability of the multiplicative stage. Not only did a large number of networks (23%) not progress beyond the additive stage, many networks (24%) that did so regressed back to the additive stage or to oscillations between the additive and multiplicative stages for some time before recovering and 17% regressed permanently. Moreover, a greater number of hidden units were installed prior to the onset of the multiplicative stage.

The majority of integer networks followed the same developmental sequence as in the pure condition following a progression from an additive stage ($v = d + t$) to the multiplicative stage. However, for 30% of the networks, there was an intermediate stage between these two defined by the rule $v = d - t$. No networks in the pure condition progressed through this intermediate stage.

3. Inter-developmental Course

A stage by epoch plot of one network from each encoding type is shown in Figure 12. The increased variability in terms of the individual developmental

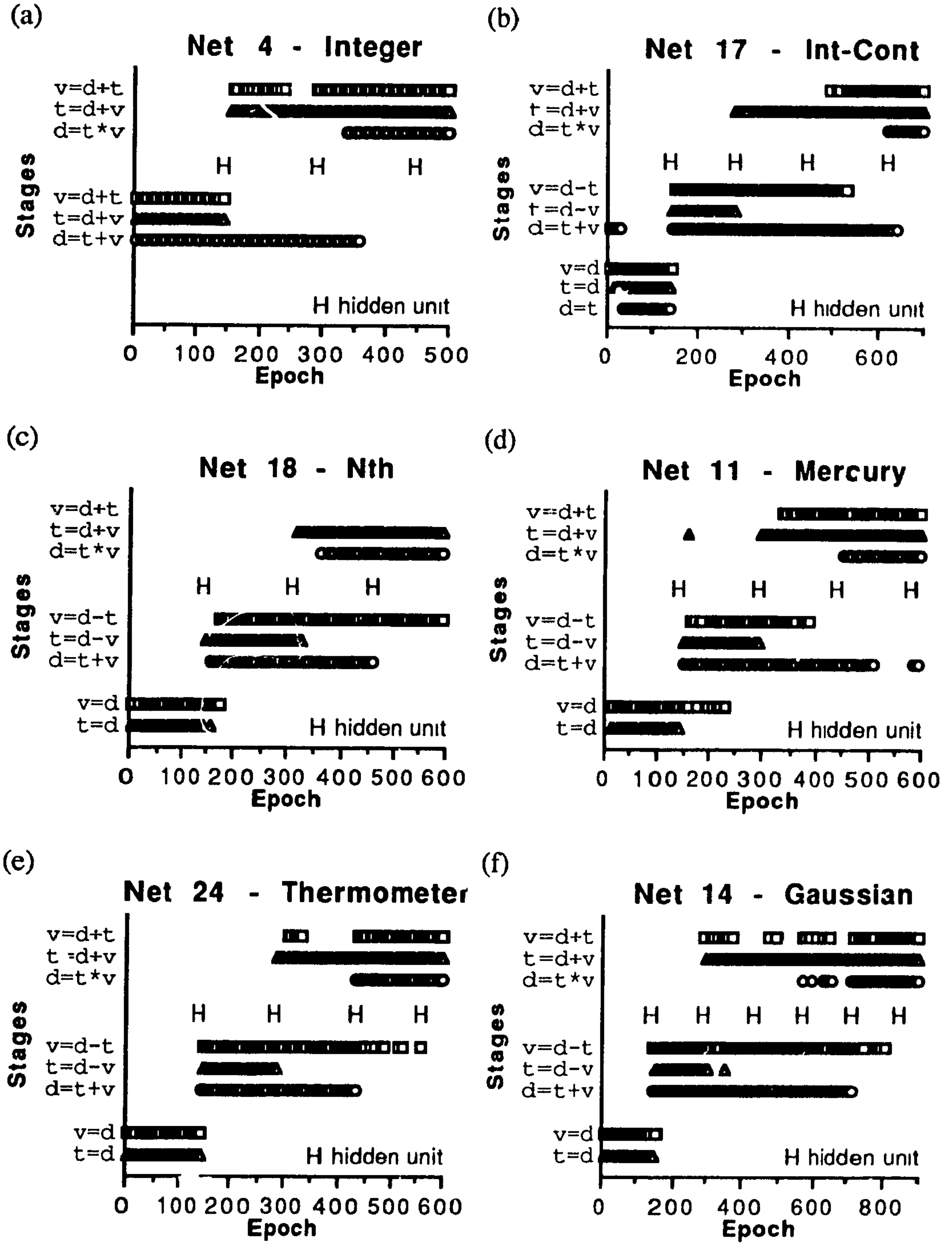


Figure 12. Limited memory condition. Identity, additive, and multiplicative stages by epoch observed during training for (a) integer, (b) integer-context, (c) nth, (d) mercury, (e) thermometer and, (f) gaussian encoding. Epoch of hidden unit recruitment indicated by "H".

sequences, in particular the distance and velocity sequences, made it difficult to find typical networks in terms of all variables including identity, additive, and multiplicative stage onsets and lengths as well as types of transitions (slow versus quick). Thus, the displayed networks were chosen as representative examples in terms of qualitative similarities to other networks within the same encoding type. As in the pure condition, all tests for significant differences in stage onsets conducted in this section are one-way repeated measures ANOVAs using an alpha level of .05.

3.1. Nth, Mercury, Thermometer, Gaussian, and Integer-Context Encoding

As can be seen in Figure 12 (b, c, d, e, f), the most common inter-developmental course for networks across encodings (excluding integer) is similar to that found in the pure condition in that the identity stages of time and velocity emerged first followed by the onset of additive stages of distance, time, and velocity. The limited memory condition differs in that the time multiplicative stage emerged first, followed next by either the distance or velocity multiplicative stage, depending on the encoding. In contrast, in the pure condition, the time and velocity multiplicative stages typically occurred together before the onset of the distance multiplicative stage.

Another difference is that the majority of integer-context networks followed this sequence with the exception that they demonstrated an early distance identity stage. In the pure condition, the majority of the integer-context networks diverged from this sequence only in that the distance additive stage began early in training before the recruitment of a hidden unit. In the limited memory condition, the

Table 17
Identity Stage Onset: Limited Memory Condition

		t=d		v=d			
Encoding	*n*	*M*	*SD*	*M*	*SD*	*F*	df
Int-cont	20	15.50	5.36	20.50	11.91	2.44	1,19
Nth	27	7.41	2.55	10.00	4.34	8.30*	1,26
Mercury	20	8.75	4.25	12.50	6.39	6.73*	1,19
Therm	22	5.91	1.97	9.55	5.96	7.11*	1,22
Gaussian	25	5.60	1.66	9.80	7.43	7.38*	1,24

* = significant at .05 level
Note. d = distance; t = time; v = velocity.

distance additive stage began after the recruitment of a hidden unit as in the other encodings.

The mean epoch of onset of identity, additive and multiplicative stages for networks following this developmental sequence and the results of a one-way repeated measures ANOVA for each encoding type are reported in Tables 17, 18, and 19 respectively. Note that the smaller *n* in Table 19 reflects the fact that some networks did not progress beyond the velocity additive stage. In addition, networks not included in this analysis typically progressed from the time and/or velocity identity stages directly to the respective multiplicative stages. A lack of a consistent inter-developmental course made it difficult to draw any conclusions from these networks.

Unlike in the pure condition, where differences between the onset of time and velocity identity stages were not significant, the differences in the limited memory condition were significant for all encodings except integer-context. However, the mean difference between onsets was extremely small (less than 5 epochs). Since testing epochs were conducted only after every five epochs, the importance of

Table 18
Additive Stage Onset: Limited Memory Condition

		d=t+v		t=d-v		v=d-t			
Encoding	*n*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*F*	df
Int-cont	20	145.90	6.03	147.40	9.19	151.20	7.81	7.68	2,38
Nth	27	156.00	5.14	160.63	8.30	218.26	79.06	16.41	2,52
Mercury	20	146.90	8.66	149.65	8.40	166.40	29.15	9.24	2,38
Therm	22	143.27	17.02	148.05	17.28	193.05	82.16	8.29	2,42
Gaussian	25	145.04	10.54	147.80	10.12	189.84	59.62	14.49	2,48

all *F* tests were significant at .05 level
Note. d = distance, t = time; v = velocity.

statistically significant differences is questionable. Moreover, individual network analysis according to the criterion outlined previously indicated that the time identity stage began before the velocity identity stage in very few networks (2 integer-context, 1 thermometer, and 1 gaussian network). Thus, for all intents and purposes, the identity stages of both concepts can be considered to have started at the same time.

Across the five encoding types, the distance and time additive stages began, on average, within 5 epochs of each other and the velocity additive stage began between 20 and 60 epochs later. The only exception was for integer-context networks where all three additive stages began within 5 epochs of each other. The differences in onset of distance, time and velocity additive stages were significant for all encoding types. Individual analysis revealed that for the majority of networks the onset of the distance and time stages occurred at the same time. Moreover, for most networks, the velocity additive stage began within 20 epochs of the other two. Thus, the significant differences observed were due to a few networks in which the velocity additive stage began somewhat later than either the distance or time additive stages. In these networks, the onset of the velocity

Table 19
Multiplicative Stage Onset: Limited Memory Condition

		d=t*v		t=d÷v		v=d−t			
Encoding	*n*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*F*	df
Int-cont	16	677.25	254.61	323.94	95.42	512.88	165.20	22.03	2,30
Nth	17	550.94	185.04	320.41	31.91	693.06	374.43	9.31	2,32
Mercury	12	615.33	123.62	340.42	49.18	539.42	243.42	12.13	2,22
Therm	20	479.40	176.30	351.90	72.74	683.50	303.95	12.39	2,38
Gaussian	23	683.10	234.23	339.74	68.82	658.74	289.91	16.74	2,44

all *F* tests were significant at .05 level
Note. d = distance; t = time, v = velocity.

additive stage was delayed by either a longer lasting identity stage or a slow transition between the two stages as demonstrated by the network shown in Figure 12d.

Across encoding types, the time multiplicative stage emerged first. Individual network analysis supported this finding. Including networks in which the multiplicative stage of velocity did not emerge (as in Figure 12c.), more than 80% of networks using integer-context, nth, mercury, and gaussian showed this. For 55% of the thermometer networks, the time multiplicative stage also emerged before either the distance or velocity stage. For the remaining thermometer networks the distance multiplicative stage typically began within 20 epochs of the time multiplicative stage. The velocity stage emerged later.

With respect to the onset of the multiplicative stages of distance and velocity there is some variability across encodings in terms of the mean data. Individual network analysis revealed that this variability was also reflected within encoding types. That is, even in networks where the mean data suggested that the distance multiplicative stage emerged after the velocity stage, a number of networks demonstrated the opposite pattern.

3.2. Integer Encoding

The majority (70%) of integer networks demonstrated the same developmental sequence as those in the pure condition. That is, the inter-developmental course began with the emergence of all three additive stages ($d = t + v$, $t = d + v$, and $v = d + t$) before eventually attaining the time and velocity multiplicative stages at approximately the same time, and finally the distance multiplicative stage (Figure 12a.). The mean epoch of onset of the additive stages were 4.76 (*SD* = 1.09), 5.00 (*SD* = 1.58), and 5.00 (*SD* = 1.58) for distance, time and velocity respectively. The differences were nonsignificant ($F_{(2,40)} = 0.32$). Individual network analysis revealed that for all networks the onsets of the three stages were the same. The mean onset of the distance, time, and velocity multiplicative stages were 560.19 (*SD* = 234.74), 152.24 (*SD* = 8.24), and 167.95 (*SD* = 30.66) epochs respectively. The differences were significant ($F_{(2,40)} = 62.00$). Individual analysis revealed that for all networks the distance multiplicative stage began after the time and velocity stages. Moreover, for 76% of these networks, the time and velocity multiplicative stages began at the same time. For the remaining networks, the multiplicative stage of velocity began following that of time.

The majority of the remaining networks (30% of all networks) differed in that rather than progressing from the additive stage defined by $v = d + t$ to the multiplicative stage, they first progressed to a second additive stage defined by $v = d - t$.

3.3. Summary: Inter-developmental Course

For all encodings except integer, the identity stages of time and velocity began early in training at approximately the same time. During this period, the potential distance rules typically did not attain sufficient r^2 values to meet the best-rule criteria consistently. Consequently no distance stages were diagnosed. One exception was with networks using integer-context encoding where a distance identity stage defined by the rule $d = t$ was observed. After the identity stages, the three additive stages emerged. For some networks the velocity stage emerged later either because of a longer identity stage or a slower transition period. However, it was more typical for the three stages to emerge at roughly the same time. Thus, up to this point the inter-developmental sequence observed in the pure and limited memory conditions are qualitatively similar with the exceptions noted above (later stage onsets and longer stages in general, and the occurrence of a distance identity stage and subsequent delay in the onset of the distance additive stage of integer-context networks in particular).

Inter-developmental sequences differ between conditions with respect to the onsets of the multiplicative stages. In the pure condition, the time and velocity stages emerged at approximately the same time. Later on, the distance multiplicative stage began. In the limited memory condition, the time multiplicative stage also emerged first. However, for the majority of networks the velocity multiplicative stage began later, either before or after the distance multiplicative stage. Moreover, for some networks, the velocity multiplicative stage was not attained at all.

Thus, the limited memory manipulation did not affect the time and velocity identity stages except in terms of the length of the stages which were typically the same for both concepts. Moreover, in general, it did not affect the onset of the additive stages. However, it did delay or prevent the onset of the velocity multiplicative stage. The delay in onset resulted from either a longer additive stage (Figure 12b), a slower transition, or a combination of both (Figure 12e).

With respect to the inter-developmental course of networks using integer encoding, the limited memory manipulation does not appear to have substantially influenced the progression of stages. That is, the majority of networks progressed from early additive stages defined by the rules $d = t + v$, $t = d + v$, and $v = d + t$ respectively to the multiplicative stages of time and velocity first and then distance. However, for a sizable minority of networks, after the velocity additive stage, a different additive stage emerged defined by the rules $v = d - t$ before the onset of the multiplicative stage. Moreover, as was discussed previously, the velocity multiplicative stage was often unstable resulting in regressions to the additive stage $(v = d - t)$.

Thus, the limited memory manipulation offset the differences between the more linear encodings and the other encodings to the extent that it delayed the onset of the distance additive stage in networks using integer-context encoding and increased the likelihood of the emergence of the velocity additive stage defined by $v = d - t$.

4. Hidden Unit Recruitment and Stage Onset

In general, additive and multiplicative stage onsets did not occur as quickly after the recruitment of a hidden unit as in the pure condition. The mean epochs from hidden unit recruitment to additive and multiplicative stage onset are reported in Table 20. The onsets of multiplicative stages of distance and velocity were affected the most. Typically they occurred 20 to 50 epochs after the installation of

Table 20
Epochs From Hidden Unit Recruitment to Stage Onset: Limited Memory Condition

	Additive Stage			Multiplicative Stage		
Encoding	d=t+v	t=d-v	v=d-t	d=t*v	t=d÷v	v=d÷t
Integer						
n		3	9	30	30	30
M		15.00	13.89	40.47	10.17	18.50
SD		13.23	12.44	34.09	6.09	17.08
Int-cont						
n	30	21	27	30	30	24
M	6.33	7.14	10.04	38.00	29.83	41.88
SD	3.46	2.54	6.51	36.33	29.67	38.39
Nth						
n	28	30	29	30	30	18
M	9.64	17.17	20.69	24.67	12.33	35.00
SD	3.31	14.30	10.33	23.63	9.98	33.69
Mercury						
n	29	30	22	30	30	22
M	5.86	11.33	25.00	29.50	23.83	48.73
SD	4.02	8.90	25.54	33.64	25.85	34.45
Therm						
n	27	29	25	30	30	28
M	5.37	11.90	12.20	21.43	15.50	26.79
SD	2.37	5.58	7.92	24.77	30.12	30.10
Gaussian						
n	30	30	25	30	30	27
M	6.03	8.67	27.80	28.77	14.17	47.96
SD	2.48	2.92	23.50	32.78	22.09	39.67

Note. d = distance; t = time; v = velocity.

the hidden unit whereas in the pure condition they occurred between 5 and 15 epochs later. Thus, more weight adjustment during the output training phase was required before the onset of the distance and velocity multiplicative stages than in the pure condition. Again it seems likely that this is due to the imprecision of weights from the time input bank due to the random nature of time inputs in velocity inference patterns.

4.1. Hinton Analysis

Transition to Additive Stages

The same general pattern of connectivity between the first hidden unit and the input layer that was seen in the pure condition was again found in the 82% of networks in the limited memory condition. That is, the weights from the distance input units had the opposite sign to those from the time and velocity input units. Thus, the first hidden unit clearly distinguished distance from time or velocity inference patterns.

It should be noted that the pattern was not always perfect. For example, occasionally one or more of the distance input units would have the same sign weight as weights from the time and velocity input units or vice versa. However, this variability was limited such that if the majority of weights from distance input units were of one sign, then the majority of weights from time and velocity units respectively were of the opposite sign.

For all of the networks demonstrating this pattern of connectivity, the distance and time additive stages emerged after some weight adjustment during the output training phase. This was also true with respect to the onset of the velocity additive stage for the majority of networks. However, for 24% of the networks the velocity additive stage did not emerge following the installation of the unit. Instead, these networks remained at the identity stage until the second hidden unit had been recruited. Thus, the random nature of the velocity inference patterns affected the ability of the algorithm to adjust the weights in such a manner that would enable the emergence of the velocity additive stage.

The results of the limited memory condition serve to qualify the importance of weight adjustment during the output training phase in the attainment of the additive stages. That is, in the pure condition, it appeared to have a minor role since the distance, time, and velocity additive stages emerged soon after the installation of the first hidden unit. However, the fact that some networks in the limited memory condition were unable to attain the velocity additive stage given a qualitatively similar pattern of connectivity demonstrates the importance of weight adjustment during the output training phase.

Transition to Multiplicative Stages

As in the pure condition, a clear pattern did not emerge with respect to transition to the multiplicative stages of time and velocity. With respect to the transition from the additive to the multiplicative distance stage in integer and integer-context networks, the same pattern of connectivity between the hidden unit and the time and velocity input units was observed. That is, the multiplicative stage emerged only after a hidden unit was recruited which received opposite-sign weights for time and velocity input groups. All 20 networks across the two encodings recruited such a hidden unit. For 13 of the 20 networks tne transition followed soon after the installation of the hidden unit as in the pure condition. However, for the remaining networks, one or more additional hidden units were installed before the multiplicative stage emerged.

Thus, the majority of networks demonstrated the same direct relationship between the installation of the *opposite-sign* hidden unit found in the pure condition. For the remaining networks the relationship was less direct. That is, even though the emergence of the multiplicative stage followed the recruitment of one or more additional hidden units, it nevertheless emerged only after such a unit had been installed at some point. It would seem that the limited memory manipulation prevented the algorithm from making the necessary weight adjustments during the output training phase that resulted in the onset of the distance multiplicative stage.

CHAPTER FIVE - DISCUSSION

1. Stages in Development

In general, network performance across training could be characterized by a coherent progression through a series of increasingly complex stages during which a large amount of the variance in the networks' responses to distance, time, and velocity inference patterns was accounted for. Moreover, both the rules that defined stages and ιhe order in which the stages emerged were, for the most part, consistent with those observed in children and adults (Wilkening, 1981; 1982).

1.1. Distance

The first stage in the distance developmental sequence was typically defined by the additive rule $d=t+v$. The 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 in order to prevent young children from using an eye-movement strategy. Thus, the simulation results agree with Wilkening's findings to the extent that when such eye-movement strategies are not available, early performance of both children and networks is characterized by the integration of time and velocity information in an additive manner.

1.2. Time and Velocity

The time and velocity developmental sequences demonstrated by the networks were also comparable with those observed by Wilkening. Early performance was characterized by identity rules ($t = d$ and $v = d$) in which the networks behaved as if responses to time and velocity inference patterns 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. However, Wilkening seems to have been more concerned with proving that this was the exception rather than the norm. That is, Wilkening (1981) concluded that the velocity identity stage resulted from the heavy memory demands inherent in the velocity task. Given his theoretical assumption concerning information integration, his reluctance to legitimize identity stages seems self evident. However, identity stages of both time and velocity have also been observed by other researchers (e.g., Piaget, 1946/1969, 1946/1970; Siegler and Richards, 1979).

The additive stages of time and velocity followed the identity stages. Again the networks exhibited performance that was characterized by the same rules as observed by Wilkening. That is, for time inference patterns, the networks' responses across the entire problem set were best captured by the rule $t = d - v$. Responses to velocity inference patterns were captured by the additive rule $v = d - t$.

Finally, adult subjects in both Wilkening's (1981) original study and the follow-up study (Wilkening, 1982) where he attempted to decrease the memory demands of the velocity task were found to use the normative multiplication rule for time but not for velocity. However, the simulations revealed multiplicative rules for both time and velocity inferences. This would seem to suggest that Wilkening's assumption of differing memory demands may be correct but his manipulation to control these differences was not. That is, in the simulations capacity demands were the same for either concept. As a result, the networks achieved multiplicative stages in both time and velocity inferences.

In sum, with respect to the distance, time, and velocity developmental sequences considered individually, the networks progressed through the same stages observed by Wilkening with the exception of an early time identity stage and the attainment of a velocity multiplicative stage.

1.3. Development of Concepts and Architectural Constraints

The simulations suggest that when all else is held constant the developmental course is more consistent across concepts than Wilkening's results indicated. That is, identity stages emerge early for both time and velocity concepts followed by the additive stages of all three concepts and then the onset of time and velocity multiplicative stages prior to the eventual attainment of the distance multiplicative stage. It is proposed that the architectural constraints of cascade-correlation are responsible for this consistency across stages.

Identity Stages

The simulation results suggest that rather than being considered as exceptions, identity stages may be viewed as naturally arising from a generative architecture. Specifically, identity stages emerge due to a combination of the limited processing ability of the initial perceptron-like architecture and the fact that the network is performing all three inference tasks.

It is proposed that, given the limited computational power of the initial network topology, the network is not able to encode both the inverse relationship between time (velocity) and distance when making velocity (time) inferences *and* the direct relationship between time and velocity when making distance inferences. That is, when making velocity inferences, time information diminishes the influence of distance information by virtue of being the denominator in the normative rule. Alternatively, when making distance inferences, time information augments velocity information by virtue of the multiplicative relationship. The analogous situation exists for velocity information when making time and distance inferences.

Given the limitations of the initial topology, the algorithm is unable to find a set of weights to accommodate both roles of the time and velocity input. That is, since only one set of weights exists for the time input and one for the velocity input, the network cannot encode the dual nature of the inputs. Therefore, the relationship of time (velocity) information to the output error of distance and velocity (time) inferences is obscured. In contrast, the relationship of distance information to the output error is more salient since the role of distance is the same in either time or velocity inference problems. As a result, when the algorithm attempts to reduce the error across all three problem types, weight adjustment may be primarily influenced by the relationship of distance input to the error. Therefore, when presented with a time (velocity) inference pattern, distance information will be more of an influence than velocity (time) information. In contrast, when presented with a distance inference pattern, neither time nor velocity information will have a greater influence nor will the weights have encoded the direct relationship between time and velocity information. Thus, identity stages emerge for time and velocity inference problems. However, neither identity, additive, nor multiplicative rules are able to capture the role of time and velocity with respect to distance inference problems during this period.

Note that additive rules of distance, time, and velocity would be possible, and indeed probable, if the network was performing only one of the tasks. That is, since all a perceptron can do is sum its inputs, performance characterized by additive rules would be expected. This is why it is proposed that it is a combination of the initial network topology and the performance of three tasks within one network that gave rise to the identity stages.

A similar argument could be put forward with respect to children. Early on, the child is confused about the differing effects of time and velocity and focuses his/her attention on distance information when making velocity or time inferences respectively. With respect to distance inferences, the child is at a loss as to how to solve the problem and may possibly choose time or velocity depending on their salience. Perceptual salience has been proposed as a possible explanation of children's poor performance on time problems (Levin, Gilat, & Zelnicker, 1980). Thus, it is possible that salience plays a role in distance inferences as well.

Additive Stages

Additive stages of each concept typically emerged after the installation of the first hidden unit. This hidden unit clearly differentiated distance from time and velocity information by assigning one sign to the weights from the distance inputs and the opposite sign to weights from time and velocity inputs. As a result, when the network was presented with a distance inference problem, the time and velocity information augmented each other. In contrast, when either a time or velocity inference pattern was presented, the distance information countered the information of the other dimension. Thus, the first hidden unit is able to encode the dual nature of time and velocity, at least in a simplistic manner, and as a result the additive stages emerge.

Multiplicative Stages

The additive stages of all three concepts eventually were replaced by multiplicative stages. 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 in either of the other two. This in turn delayed the onset of the distance multiplicative stage. Thus, the distance additive rule would seem to be a very good approximation of distance inference patterns. It may be that for people, use of an additive rule persists as a heuristic type approach that is generally good enough.

Summary

The developmental course demonstrated by the networks was consistent across concepts in terms of what types of rules emerged. This was in large part due to the architectural constraints of cascade-correlation. That is, time and velocity identity stages emerged early in training prior to the recruitment of a hidden unit. Once the hidden unit had been installed, the network was able to differentiate among the dimensional information in such a manner that additive performance was possible by augmenting or countering the information of the defining dimensions. Finally, time and velocity inference patterns were mastered, in the sense that the respective normative rules accounted for the greatest amount of variance in the networks' responses, before the distance inference patterns. This was likely due to the fact that, in terms of error reduction, the distance additive rule was sufficient to reduce a large proportion of the error.

1.4. Memory Demands

Wilkening (1981) had three findings that he explained by differing memory demands: (1) the use a velocity identity rule by 5-year-olds, (2) the inability of adults to use the normative velocity rule, and (3) 5-year-olds apparently precocious use of the normative distance rule. According to Wilkening, the velocity task was more difficult than the time task since time information had to be recalled after the distance information was presented. In other words, the time information wasn't immediately available in the velocity task and had to be retrieved from memory. Thus, the 5-year-olds regressed to using an identity rule and adults to a subtraction rule. With respect to the young children's use of the distance multiplicative rule, he argued that the task was sufficiently easy so that children could employ an eye-movement strategy which revealed an implicit understanding of the correct integration of time and velocity information to infer distance.

The pure condition simulation represents the ideal situation that Wilkening (1982) was striving for in the follow-up study. That is, the memory demands of the tasks were the same in that for any given inference pattern the defining dimension information was available at the moment the inference was to be made. Moreover, the use of an eye-movement strategy was equally impossible across the three tasks.

A direct test of Wilkening's hypothesis of differing memory demands was undertaken in the limited memory condition. There, the time information for velocity inference patterns was degraded by varying the probability that the correct time information would be presented to the network.

Although network performance was much more variable both across networks and within networks, the most general finding was that the normative velocity stage defined by the rule $v = d \div t$ was delayed and in some cases prevented. This would seem to support Wilkening's original contention that the reason the velocity multiplicative stage was not attained by his subjects was due to the extra memory demands of the task. Moreover, even for networks that eventually attained the multiplicative stage, it could be argued that they did so only after an inordinate amount of training not representative of the amount of learning that people would normally have.

In the ideal situation, the velocity identity stage would have lasted longer than the time identity stage. Thus, at the same time that the networks would have been classified as using an additive rule for time inference patterns they would be still solving velocity problems by primarily focusing on distance information as subjects in Wilkening's experiments did. Although some networks did in fact demonstrate this developmental pattern, it was more typical for the additive stages of both time and velocity to emerge at the same time. This would seem to argue that early identity stages result from a lack of computational power regardless of memory demands.

In the limited memory condition, the time multiplicative stage emerged before either the velocity or distance multiplicative stages. No strong conclusions can be drawn about the emergence of distance versus velocity since this varied across networks. Nor can any strong conclusions be drawn from the fact that the distance stage emerged later than the time stage. That is, although Wilkening found that adults employed the correct normative rules in both the distance and time task, he did not include 10-year-olds in the follow-up study. Thus, although he found that 10-year-olds used the normative time rule in the first experiment, it is possible that the same aged children may use the distance additive rule when the eye-movement strategy is not available. The simulation results of both the pure and limited memory conditions suggest that the use of the correct distance rule emerges later than use of the correct time rule.

1.5. The Issue of Encoding

A secondary issue in this study was the effects of different encoding methods on network performance. It was discovered that the psychological realism of the developmental course of the three concepts was dependent on the choice of input encoding.

Initially five encodings were compared -- integer, nth, mercury, thermometer, and gaussian. For the most part, performance using the different encodings was qualitatively similar. However, when the inference patterns were encoded as integers, substantial differences were observed. That is, for integer networks the distance additive stage began very early in training prior to the installation of a hidden unit. At the same time, performance on the time and velocity inference patterns was characterized by the additive rules $t = d + v$ and $v = d + t$, respectively. Moreover, the normative multiplicative stages of all three concepts emerged following these additive rules. Although the distance developmental sequence is the same as the one observed by Wilkening (1982), the time and velocity developmental sequences diverged greatly. That is, neither time nor velocity identity stages were observed nor were the additive stages defined by the rules $t = d - v$ and $v = d - t$. It appears that the error reduction that occurred during the period characterized by the aberrant additive rules ($t = d + v$ and $v = d + t$) was sufficient to allow for the emergence of the normative multiplicative rules.

Construction of the Dimension

Why did the performance of networks using integer encoding differ? It appears that for the encodings other than integer the dimension itself has to be constructed by the network. That is, there is no intrinsic information in these encodings that would indicate that a time value of 5, for example, is greater than a time value of 2. This must be learned by the network. Alternatively, with integer encoding, a dimensional value of 5 has a greater influence (positive or negative depending on the weights) on the outcome than a value 4, 3, etc.., regardless of the initial weights in the network. Therefore, integer encoding can be considered more inherently linear than the other encodings.

To test this hypothesis, the integer encoding was modified by adding three context units. This was assumed to decrease the linearity of the input pattern across the dimensions to the extent that the respective context unit of a given inference pattern always had a value of one. Thus, the pattern as a whole was less linear.

The results of this additional encoding, integer-context, supported the linearity hypothesis in so far as time and velocity identity stages were the first diagnosable stages and were followed by the typical additive rules ($t = d - v$ and $v = d - t$) observed with other encodings. The only difference was that the additive stage of distance emerged before the recruitment of the first hidden unit as was the case with integer encoding. However, in the limited memory condition, the distance additive stage was delayed, emerging only after the installation of the first hidden unit. Since the weights from the time input were imprecise due to the limited memory manipulation, any advantage of the linear input was obscured. In sum, when the linearity of the integer encoding was diminished by using a context unit, the performance of the networks was more in line with that of the other encoding types.

2. Connectionism and Development

As Flavell and Wohlwill (1969) surmised, the two fundamental issues in development are: (1) What knowledge structures develop?, and (2) How does developmental transition occur? With respect to distance, time, and velocity development, the structures that develop are increasingly complex representations that enable performance characterized by identity, additive, and multiplicative integration of defining dimensions. The connectionist simulations presented here suggest that the underlying representations arise from inter-connected, simple processing units. Moreover, the simulations suggest that developmental transition results from both weight adjustment and the recruitment of hidden units. In terms of human cognitive development, transition is likely due to incremental learning and increases in non-linear representational abilities.

2.1. Structure

Organized Knowledge in the Form of Weighted Connections

Siegler (1981) has conceptualized children's cognitive development as a progression through increasingly complex binary decisions. Through the use of his rule assessment methodology he has found support for his theory in a number of cognitive domains including the development of distance, time, and velocity concepts (Siegler & Richards, 1979). While the simulations in this study support their findings of an early emphasis on distance information to solve time and velocity problems, the underlying representation responsible for this type of performance and subsequent stages in development is radically different.

Networks naturally integrate dimensions in that unit activations are based on summing all the inputs to the unit. Nevertheless, these networks initially performed as if focusing only on one dimension. It has been suggested that identity stages emerge as a result of an inability to encode the dual nature of time and velocity information when making inferences about distance and time or velocity respectively. Thus, it is proposed that children do not ignore the other dimension (time or velocity) but rather that they lack the ability to resolve their dual roles because of limited processing ability. That is, children consider both defining dimensions but place more emphasis on distance when making time or velocity inferences because distance information is more consistently related to the target performance.

Additive integration of dimensions emerged as a result of recruiting a hidden unit that differentiated distance from time and velocity information based on a pattern of connectivity in which the weights from time and velocity inputs had the opposite sign to weights from the distance inputs. Moreover, there was some evidence that the distance multiplicative stage emerged following a hidden unit that further differentiated time and velocity information.

Thus, networks were capable of achieving knowledge states that have typically been assumed to be represented by explicit symbolic rules (i.e., identity, additive, and multiplicative algebraic rules) by means of weighted connections among simple processing units whose only function is to send and/or receive excitatory and inhibitory signals.

It should be stressed that although network performance is characterized by explicit rules, this in no way implies that the networks have represented the rules explicitly. That is, the distributed representation among the units in the network enable the networks to perform "as if" following explicit rules. This may also be the case for subjects 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. The present connectionist models show in detail how such "as if" performance is possible.

Continuous Learning Across the Stage

Although network performance was stage-like in that long periods of training resulted in the same classification of responses, learning did occur within stages. The large reduction of error within stages suggests that learning continued even though the overall responses did not undergo qualitative change. Moreover, the amount of variance accounted for by a given rule defining a stage was not constant across the stage. Together, these results suggest that learning is continuous during a stage.

The conception of a stage as a dynamic rather than a static period seems problematic for rule-based approaches. For example, if a child has an explicit rule for solving time problems that involves focusing on distance information alone (i.e., Siegler and Richard's rule I), it is unclear how improvement beyond the correct application of the rule within a stage might occur. In contrast, the error reduction across the time identity stage demonstrated by the networks suggests that weight adjustment can improve the accuracy of time inferences even though the responses remain characterized by the time identity rule.

2.2. Transition

The progression from identity to additive and then multiplicative stages represents qualitative restructuring of a knowledge representation. Researchers working within the framework of information integration theory (Anderson, 1974) have had difficulty formulating a precise mechanism that would account for these qualitative changes. In contrast, the use of cascade-correlation provides an explicit mechanism of transition. That is, it was demonstrated that qualitative changes in knowledge were brought on by both quantitative changes in the form of weight adjustments and underlying qualitative changes due to hidden unit recruitment.

Hidden Unit Recruitment and Weight Adjustment as Mechanism

Some researchers have argued that weight adjustment alone is capable of stage transition (Plunkett & Sinha, 1992; McClelland, 1988). However, the simulations presented here suggest that weight adjustment *and* hidden unit recruitment are responsible for developmental changes.

The evidence for hidden unit recruitment as a mechanism of stage transition is strong. In general, transition from the identity stages to additive stages of time and velocity followed the recruitment of the first hidden unit. The first hidden unit also brought about a transition from the initial period not captured by any distance rules to the additive stage of distance. 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. Moreover, new stages emerged soon after the installation of hidden units, often requiring just five to ten epochs of weight adjustment.

In these simulations there is strong evidence that simple weight adjustment alone was not sufficient to cause transition from one stage to another. For example, it has been argued that the transition to additive time and velocity stages *required* the recruitment of a hidden unit. That is, given the initial perceptron-like architecture of the networks, weight adjustment alone would not have brought about transition to additive stages.

The argument is not that weight adjustment was unimportant but rather that transition is brought about by a combination of both weight adjustment and hidden unit recruitment. An interesting example of the interaction of weight adjustment and hidden unit recruitment was found in those networks where there was a period characterized by a tie following the additive stages, prior to the onset of the multiplicative stages. It appears that weight adjustment was sufficient to bring about a change in network responses from performance characterized by additive rules to performance characterized by both additive and multiplicative rules but that a new hidden unit was necessary to complete the transition to performance characterized solely by the multiplicative rule.

Given that the emergence of stages appears to be intimately related to the constraints of a generative architecture, it seems unlikely that simulations using a static architecture such as back-propagation would demonstrate the same psychologically realistic developmental course. In particular, the early emergence of identity stages may be difficult to capture in back-propagation networks. That is, if the initial network topology includes a hidden unit layer, then the ability to encode the dual nature of time and velocity would exist. Naturally, this is speculative since, at present, no attempt to simulate the acquisition of distance, time, and velocity concepts within a back-propagation architecture has been undertaken.

2.3. Summary

The developmental course demonstrated by the networks is consistent with the conclusions of Flavell (1971) concerning stages of cognitive development. That is, the networks progressed through *qualitatively* different, highly *organized* knowledge structures, by means of relatively gradual transitions. Knowledge representations in the form of inter-connected simple processing units emerged as learning progressed by means of gradual weight adjustment and hidden unit recruitment. These representations were characterized by increasingly complex algebraic rules.

3. Possible Criticisms and Limitations

3.1. Single Network for Three Tasks

It was suggested that identity stages resulted from the use of a single network to perform all three tasks. Some might argue that the use of a single network is unwarranted. That is, why not use one network for each task? The simple answer is that by using one network the algorithm is forced to reduce error that is attributable to all three inference types. Moreover, any hidden units installed into the network may represent knowledge of all three inferences. Since the question is really about the inter-relationships among the three concepts, using one network is the simplest means of representing knowledge about these inter-relationships.

3.2. Correlation

Strict connectionists might object to the use r^2 values of different rules as a means of testing performance rather than the more typical measure of sum-squared error. However, unlike other connectionist simulations such as children's performance on the balance scale (e.g., McClelland, 1988; Shultz & Schmidt, 1991) it was not possible to create problem sets such as those used in rule assessment (Siegler, 1981). Problem sets that would reflect one rule versus another are precluded when a binary decision is not demanded of the networks.

Moreover, the measurement of the correlation between responses generated by a given algebraic rule and those actually made by the network in response to the different inference patterns is qualitatively similar to how subjects' performance was assessed in Wilkening's experiments. That is, subjects were not required to provide accurate responses. What was assessed was the overall relationship of subjects' responses relative to one another. Therefore, determining the correlation of different potential rules seems appropriate.

3.3. Diagnosing Performance

Overall, the vast majority of epochs across training were diagnosable. However, one limitation was the inability to classify the period prior to the onset of the distance additive stage even though it was associated with significant error reduction. However, as discussed above, it may be that since neither time nor velocity information is any better at predicting outcomes, any preference that children show for one or the other might reflect extraneous task factors such as the salience of the dimension on a given task.

3.4. Limited Memory Manipulation

One limitation of the limited memory manipulation was that it affected distance inference patterns as well as the intended velocity inferences. That is, since the weights from the time input bank are used to propagate information of both velocity and distance inference patterns, they also unavoidably affect performance on distance inference patterns. The qualitative progression of stages within the distance developmental sequence remained unchanged. However, the limited memory manipulation makes conclusions about the development difficult since the length of the stages was affected.

Also with respect to the limited memory manipulation, the amount of error in recall was arbitrary. Any strong conclusions would have to wait until other levels of error were used.

3.5. Generalization to Choice Tasks

Another possible criticism is that the results of these simulations may not generalize to binary choice tasks. Even more seriously, it might be argued that the use of a linear output predisposes the algorithm to find additive and multiplicative rules. However, preliminary work suggests that these criticisms are not warranted. That is, work is underway that extends the findings of the present study to situations in which a decision about which of two objects has traveled the greater distance, for the longer time, or at a faster velocity given information about the two defining dimensions of both objects. This is analogous to the present study but requires a binary decision. The responses are encoded on two outputs with sigmoid activation functions. The same identity, additive, and multiplicative stages have been observed. Therefore, the emergence of additive and multiplicative stages observed in the present simulations is not due simply to the use of a linear output.

3.6. Network Analyses

Although Hinton analyses proved useful in understanding the emergence of distance, time, and velocity additive stages, they had limited use in revealing how transition to multiplicative stages occurred. This is because as more hidden units are installed into the network, the role of any one unit becomes less obvious since the representations become increasingly distributed and non-linear. Each new hidden unit is installed on a separate layer and receives input from all previous hidden units and the input layer. As such, Hinton diagrams are probably more useful in non-generative learning architectures such as back-propagation with a limited number of units and layers.

A promising new approach is being undertaken by Shultz and Elman (1993). They have extended a technique developed by Sanger (1989) called "contribution analysis" to analyze knowledge representations in generative algorithms. Essentially, the technique involves determining the contribution or influence of a given unit in determining the network's response. Contribution analysis differs significantly from Hinton analysis in that activations of sending units are considered in addition to size and sign of weights. That is, the contribution of any one unit is a function of both the unit's activation given a particular training pattern and the sign and size of the weight connecting it to a given output unit. Principle components analysis is then performed on a contributions by input pattern matrix. The advantage is that many contributions may be reduced to a few interpretable components. Future work will apply contribution analysis to the present network models.

3.7. Realism

A general limitation of the simulations is a lack of realism and a high degree of simplification compared to both the child and his/her environment. That is, the resources that a child brings to the task are far more complex than those represented in the simulations. Moreover, these resources are applied to both a far ranging type and number of tasks. Nevertheless, the model shows that an architecture employing brain-style computation can capture the known qualitative aspects of the development of distance, time, and velocity concepts based on input from the environment.

4. Predictions and Future Research

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 child (network). Thus, the primary direction of any future research on this issue is to determine under what conditions this is true.

The simulation results, together with Wilkening's studies, make several predictions. First, the results suggest that if 5-year-olds can integrate distance *and* velocity and time *and* velocity information in an additive manner to infer time and distance respectively, they should also be able to integrate distance and time information additively to make velocity inferences. That is, Wilkening's hypothesis about the velocity task being more difficult than the time task in terms of memory demands is likely, but his manipulation to equalize the demands of the two tasks was ineffective. Thus, if memory demands were reduced in all three tasks, the simulations predict that all three types of inferences would be solved additively by 5-year-olds.

Second, but related, the simulations suggest that children *younger* than five years of age solve time and velocity inference problems by focusing on distance information and solve distance inference problems based on either time or velocity information depending on which is more salient. Thus, the simulations suggest the velocity developmental sequence observed by Wilkening is qualitatively accurate but that differences in onsets of stages relative to time and distance were due to different task demands. That is, Wilkening's observation that 5-year-olds use an identity rule to make velocity inferences is likely related to the extraneous task demands, not to the children's understanding of the relationship of time and distance. However, the simulations predict that children younger than five years ot age make velocity inferences by focusing on distance information because of processing limitations. Therefore, future research should include younger children and manipulate the salience of velocity and time information in the distance task.

Third, the simulations suggest that the distance multiplicative stage emerges after both time and velocity multiplicative stages. Since Wilkening did not study 10-year-olds' performance when an eye-movement strategy was not possible, it would be necessary to re-examine 10-year-olds under this condition. The simulation results predict that if the memory demands of the three tasks were similar, 10-year-olds would use the additive rule for distance inferences and the normative multiplicative rules for both time and velocity inferences.

Finally, as suggested, if the task demands of the velocity task were reduced it is predicted that 10-year-olds would make velocity inferences by integrating the dimensions with the correct multiplicative rule. Therefore, the inability of Wilkening's subjects to correctly integrate time and distance information is likely due to extra memory demands.

Several issues in terms of future connectionist simulations are also relevant. First, differing levels of probability of correct recall could be examined. It may be that by decreasing the likelihood of correct recall of time information in the velocity task would lengthen the identity stage of velocity and thus delay the onset of the additive stage. Moreover, to make the limited memory manipulation more realistic, the probability of correct recall might increase as training progresses.

Second, with respect to analyzing network performance, an interesting approach would be to use functional measurement. However, this is somewhat problematic since functional measurement relies on analysis of variance of group data. In terms of network data, it is difficult to know how individual epochs of different networks relate to one another. Moreover, although epochs certainly can be related qualitatively to age in that more epochs imply greater age, it is not clear how epochs and age relate precisely.

Third, as mentioned, the use of contribution analysis may provide more insight into the underlying representations embodied in the network's connections. Although Hinton analyses were useful in understanding networks with smaller topologies (i.e., after one or two hidden units) the representations after additional hidden units were installed in larger networks were obscured by the sheer number of units and consequent distributed representations.

Fourth, it was argued that stage transition by means of weight adjustment *and* hidden unit recruitment, or incremental learning *and* increases in non-linear representational power in humans, are responsible for early identity stages and subsequent transitions to more advanced stages. Given that static networks do not allow for structural increases in representational power, such networks would not be expected to capture the stage progressions observed in both the current simulations and humans. Identity rules have been observed in simulations using back-propagation networks. For example, McClelland (1988) captured the "weight only" rule in his simulations of the balance scale. However, it is believed that in the current simulations identity stages emerge from the inability of the initial perceptron-like architecture to find a set of weights that would accommodate the three inference tasks. Naturally this hypothesis needs to be empirically tested by attempting to simulate the acquisition of distance, time, and velocity concepts using a back-propagation network.

Finally, as mentioned, work is underway that attempts to simulate children's responses in a binary choice task. Although there are parallels with both Piaget's (1946/1969, 1946/1970) and Siegler and Richards' (1979) work, among others, these simulations employ Wilkening's (1981) methodology of presenting information about two defining dimensions and requiring inferences about the third dimension. This should offer insight into the importance of assessing knowledge with either choice or non-choice tasks in addition to the generalizability of the present simulations.

5. Conclusion

It has been argued that the stage progressions observed in network performance have resulted from the constraints inherent in a generative algorithm. That is, identity stages arise from a lack of computational power and subsequent inability to encode both the inverse and direct relationship of time (velocity) to distance and velocity (time), respectively. Installation of the first hidden unit enables the networks to form a simplistic representation of this relationship in that additive rules are used to integrate the dimensions. Finally, after further changes to the network topology by hidden unit recruitment, performance that is characterized by the normative rules of distance, time, and velocity emerge.

The main theoretical implications are that children use a domain general algorithm or learning rule that allows for increased complexity in knowledge representations as the child's capacity for problem solving increases. Moreover, it is the domain specific constraint of making inferences on the three problem types that determines the type and progression of knowledge representations. Early identity rules result from an inability to conceptualize the dual role of time and velocity rather than from the child ignoring distance information. Increases in the child's capacity enable simplistic additive representations and then more complex multiplicative representations.

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

The success of the present 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 (Anderson, 1974) may also 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 compensation tasks in addition to distance, time, and velocity integration. The advantage of connectionist simulations is that they provide both precise knowledge representations of integration rules and a precise mechanism of how development proceeds from simpler to more complex integration rules.

Cascade-correlation has already proven useful in understanding a number of cognitive developmental phenomenon including children's performance on the balance scale task (Shultz & Schmidt, 1991; Shultz, Mareschal, & Schmidt, in press), the acquisition of personal pronouns (Shultz, Buckingham & Oshima-Takane (in press), and children's seriation ability (Mareschal & Shultz, 1993). The present 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, validating cascade-correlation as a promising tool of investigation into cognitive development by means of connectionist simulations.

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, distance, and durations of movements. *Developmental Psychology*, **17**, 490-493.

Alpaydin, E. (1991). GAL: Networks that grow when they learn and shrink when they forget. Technical Report, 91-032, International Computer Science Institute, Berkeley, CA.

Anderson, J. A. (1990, June). Data representation in neural networks. *AI Expert*, pp. 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 II: Measurement, Psychophysics, and Neural Information Processing* (pp. 236-305). San Francisco, CA: W. H. Freeman and Company.

Anderson, N. H. (1980). Information integration theory in developmental psychology. In F. Wilkening, J. Becker, & T. Trabasso (Eds.), *Information Integration by Children* (pp. 1-45). Hillsdale, NJ: Erlbaum.

Anderson, N. H. (1991). Functional memory in person cognition. In N. H. Anderson (Ed.), *Contributions to Information Integration Theory: Volume I: Cognition* (pp. 1-55). Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, N. H. & Cuneo, D. O. (1977). The height + width rule in children's judgments of quantity. Technical Report, no. 69, Center for Human Information Reasoning, University of California, San Diego, CA.

Anderson, N. H. & Wilkening, F. (1991). Adaptive thinking in intuitive physics. In N. H. Anderson (Ed.), *Contributions to Information Integration Theory: Volume III: Developmental* (pp. 1-42). Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Bates, E. A. & Elman, J. L. (1992). Connectionism and the study of change. Technical Report 9202, Center for Research in Language, University of California, San Diego, CA.

Bickel, F. (1984). A time-velocity ratio investigation. *Journal of Research in Music Education*, **32**, 105-111.

Bentley, A. M. (1986). The development of Swazi children's understanding of Duration: The influence of speed and distance. *International Journal of Psychology*, **21**, 355-361.

Brendt, T. J., & Wood, D. J. (1974). The development of time concepts through conflict based on primitive duration capacity. *Child Development*, **45**, 825-828.

Churchland, P. M. (1990). Cognitive activity in artificial neural networks. In D. N. Osherson & E. E. Smith (Eds.), *An Invitation to Cognitive Science: Volume III: Thinking* (pp. 199-227). Cambridge, MA: MIT Press.

Crépault, J. (1977). Organisation des relations temps, espace et vitesse. *Du temps biologique au temps Psychologique. Symposium de l'association de psychologie scientifique de langue français*, 227-253.

Crépault, J. (1978). Le raisonnement cinématique chez le pré-adolescent et l'adolescent I. Esquisse d'un modèle théorique: Concepts de base. *Archives de Psychologie*, **178**, 133-183.

Crépault, J. (1979). Influence du repérage sur la durée: Etude génétique des inférences cinématique. *L'Année Psychologique*, **79**, 43-64.

Crépault, J. (1980). Compatibilité et symétrie: Etude génétique des inférences cinématiques chez des sujets de 11 et 13 ans. *L'Année Psychologique*, **80**, 81-97.

Crépault, J. (1981). Etude longitudinale des inférences cinématiques chez préadolescent et l'adolescent: Evolution ou régression. *Canadian Journal of Psychology*, **35**, 244-253.

Dempsey, A. (1971). Time conservation across cultures. *International Journal of Psychology*, **6**, 115-120.

Elman, J. (1991). Incremental learning, or the importance of starting small. Technical Report 9101, Center for Research in Language, University of California, San Diego, CA.

Fahlman, S. E. (1988). An empirical study of learning speed in back-propagation networks. Technical Report, CMU-CS-88-162, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

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

Ferretti, R. P. & Butterfield, E. C. (1986). Are children's rule-assessment classifications invariant across instances of problem types? *Child Development*, **57**, 1419-1428.

Ferretti, R. P., Butterfield, E. C., Cahn, A. & Kerkman, D. (1985). The classification of children's knowledge: Development on the balance-scale and inclined-plane tasks. *Journal of Experimental Child Psychology*, **39**, 131-160.

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.* NY: Oxford University Press.

Flavell, J. H. (1971). Stage-related properties of cognitive development. *Cognitive Psychology,* **2**, 421-453.

Friedman, W. J. (1978). Development of time concepts in children. In H. Reese & L. P. Lipsitt (Eds.), *Advances in Child* Development: *Volume 12* (pp. 267-298). NY: Academic Press.

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

Halford, G. S., Brown, C. A., & Thompson, R. McL. (1986). Children's concepts of volume and flotation. *Developmental Psychology*, **22**(2), 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.), *Working papers of the AAAI Spring symposium on machine learning of natural language and ontology,* 65-74.

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

Hoehfeld, M. & Fahlman, S. E. (1991). Learning with limited numerical precision using the cascade-correlation algorithm. Technical Report, CMU-CS-91-130, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Homers, W. (1980). Information processing in children's choices among bets. In F. Wilkening, J. Becker, & T. Trabasso (Eds.), *Information Integration by Children* (pp. 99-112). Hillsdale, NJ: Erlbaum.

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

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**(4), 401-433.

Leon, M. (1980). Integration of intent and consequence information in children's moral judgments. In F. Wilkening, J. Becker, & T. Trabasso (Eds.), *Information Integration by Children* (pp. 71-97). Hillsdale, NJ: Erlbaum.

Leon, M. (1982). Rules in children's moral judgements: Integration of intent, damage, and rationale information. *Developmental Psychology*, **18**(6), 835-842.

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. (1982). The nature and development of time concepts in children: The effects of interfering cues. In W. J. Friedman (Ed.), *The developmental psychology of time* (pp. 47-85). NY: Academic Press.

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. & Globerson, T. (1984). The development of time concepts among advantaged and disadvantaged Israeli children. *Journal of Genetic Psychology*, **145**, 117-125.

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

Levin, I., Wilkening, F., & Dembo, Y. (1984). Development of time quantification: Integration and nonintegration of beginnings and endings in comparing durations. *Child Development*, **55**, 2160-2172.

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**(3), 329-347.

Lovell, K. & Slater, A. (1960). The growth of time: A comparative study. *Journal of Child Psychology and Psychiatry*, **1**, 179-190.

Mareschal, D. (1991). Cascade-correlation and the genetron: Possible implementations of equilibration. Technical Report 91-10-17. McGill Cognitive Science Centre, McGill University, Montréal.

Mareschal, D. & Shultz, T. R. (1993). A connectionist model of the development of seriation. *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, 676-681.

McClelland, J. L. (1988). Parallel Distributed Processing: Implications for cognition and development. Technical Report AIP-47, Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA.

Minsky, M. & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT 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. Riffle & V. N. Rockcastle (Eds.), *Piaget rediscovered*. A report on the conference of Cognitive Studies and Curriculum Development, 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)

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, **28**, 73-193.

Plunkett, K. & Marchman, V. (1989). Pattern association in a back propagation network: Implications for language acquisition. Technical Report 8902, Center for Research in Language, University of California, San Diego.

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

Richards, D. D. (1982). Children's time concepts: Going the distance. In W. J. Friedman (Ed.), *The developmental psychology of time* (pp. 13-45). NY: Academic Press.

Richie, D. M. & Bickhard, M. H. (1988). The Ability to perceive duration: Its relation to the development of the logical Concept of time. *Developmental Psychology*, **24**(3), 318-323.

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: Volume 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: Volume II: Psychological and Biological Models* (pp. 216-271). Cambridge, MA: MIT Press.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. *Connection Science*, **1**, 115-138.

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

Shultz, T. R. (1991). Simulating stages of human cognitive development with connectionist models. In L. Birnbaum & G. Collins (Eds.), *Machine learning: Proceedings of the Eighth International Workshop,* pp. 105-109. San Mateo, CA: Morgan Kauf.nan.

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

Shultz, T. R., & Elman, J. L. (1993, November). *Analyzing cross connected networks.* Paper presented at the Neural Information Processing Systems, Seventh Annual Meeting, Denver, CO.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (in press). Modeling cognitive development on balance scale phenomena. *Machine Learning*.

Shultz, T. R., & Schmidt, W. C. (1991). A Cascade-Correlation model of balance scale phenomena. In: *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pp. 635-640. Hillsdale, NJ: Erlbaum.

Shultz, T. R., Schmidt, W. C., Buckingham, D., & Mareschal, D. (in press). Modeling cognitive development with a generative connectionist algorithm. In T. Simon & G. Halford (Eds.), *Developing Cognitive Competence: New Approaches to Process Modelling*. Hillsdale, NJ: Lawrence Erlbaum.

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

Siegler, R. S. & Richards, D. D. (1979). Development of time, speed, and distance concepts. *Developmental Psychology*, **15**(3), 288-298.

Siegler, R. S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development*, **46**(2).

Siegler, R. S. (1982). The rule-assessment approach and education. *Contemporary Educational Psychology*, 7, 272-288.

Sjøgaard, S. (1991). *A conceptual approach to generalization in dynamic neural networks.* Dissertation, Computer Science Department, Aarhus University, Aarhus Denmark.

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

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: Erlbaum.

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

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

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 III Developmental* (pp. 45-80). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wilkening, F., Levin, I., & Druyan, S. (1987). Children's counting strategies for time quantification and integration. *Developmental Psychology*, **23**(6), 823-831.

Yang, J. & Honavar, V. (1991). Experiments with the cascade-correlation algorithm. Technical Report, # 91-16, Department of Computer Science, Iowa State University, Ames, IA.

Appendix A - Performance Attributable to Stage: Limited Memory Condition

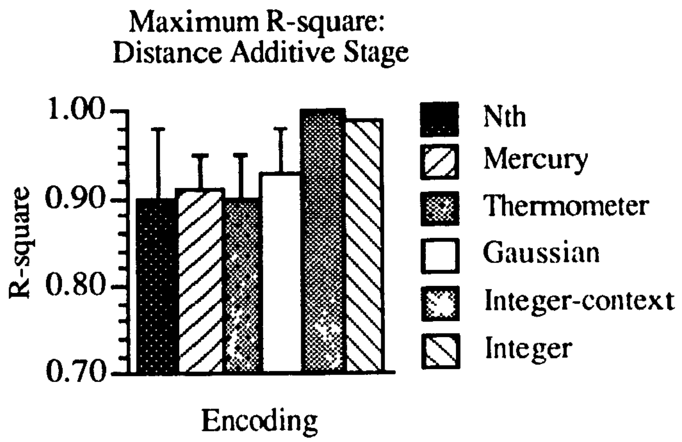


Figure A 1. Maximum r^2 of d=t+v rule attained during additive stage of distance developmental sequence.

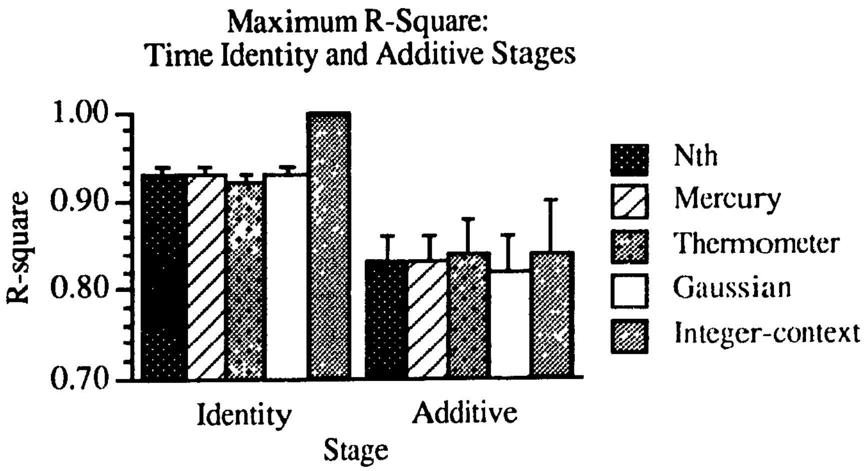


Figure A 2. Maximum r^2 of t=d and t=d-v rules attained during identity and additive stages of time developmental sequence respectively.

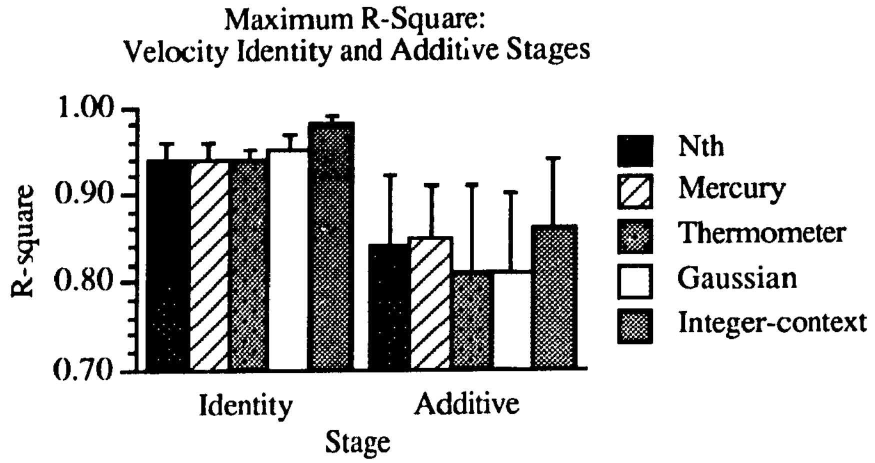


Figure A 3. Maximum r^2 of v=d and v=d-t rules attained during identity and additive stages of velocity developmental sequence respectively.

Appendix B - Error Reduction: Limited Memory Condition

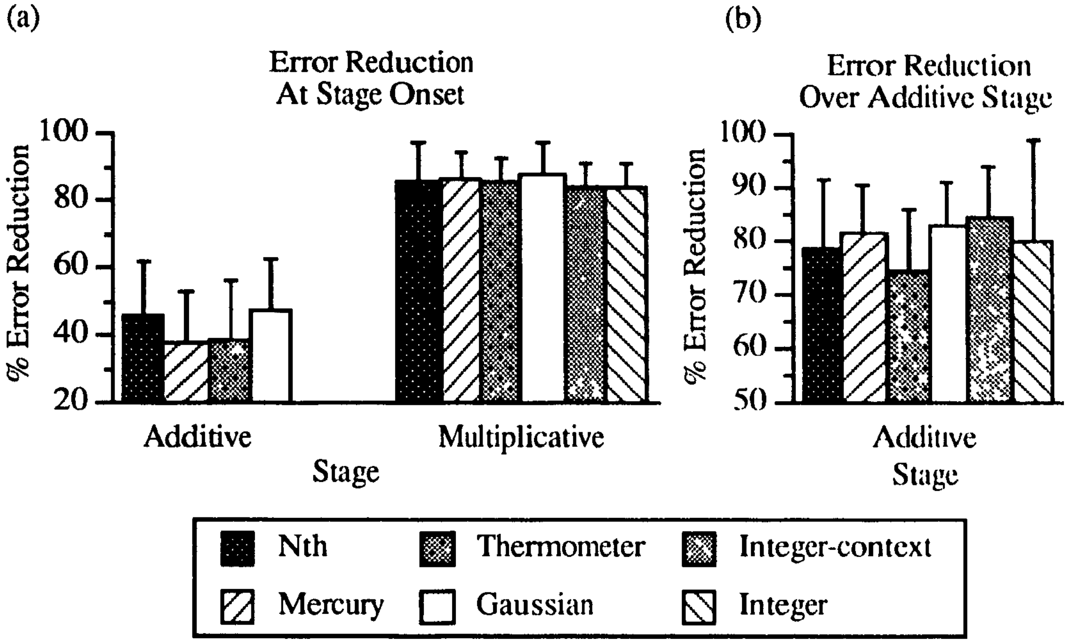


Figure B 1. Mean percent of error reduction in distance inference patterns (a) from epoch 5 to additive and multiplicative stage onset and (b) over additive stage. Error bars are standard deviations from the mean.

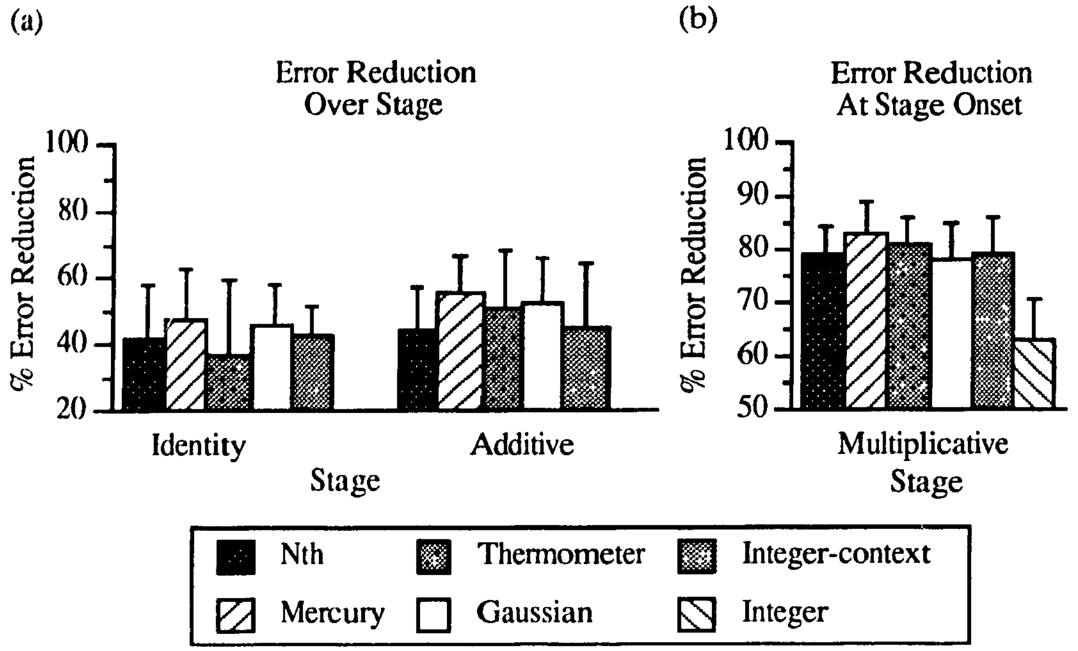


Figure B 2. Mean percent of error reduction in time inference patterns (a) over identity and additive stages and (b) from epoch 5 to multiplicative stage onset. Error bars are standard deviations from the mean.

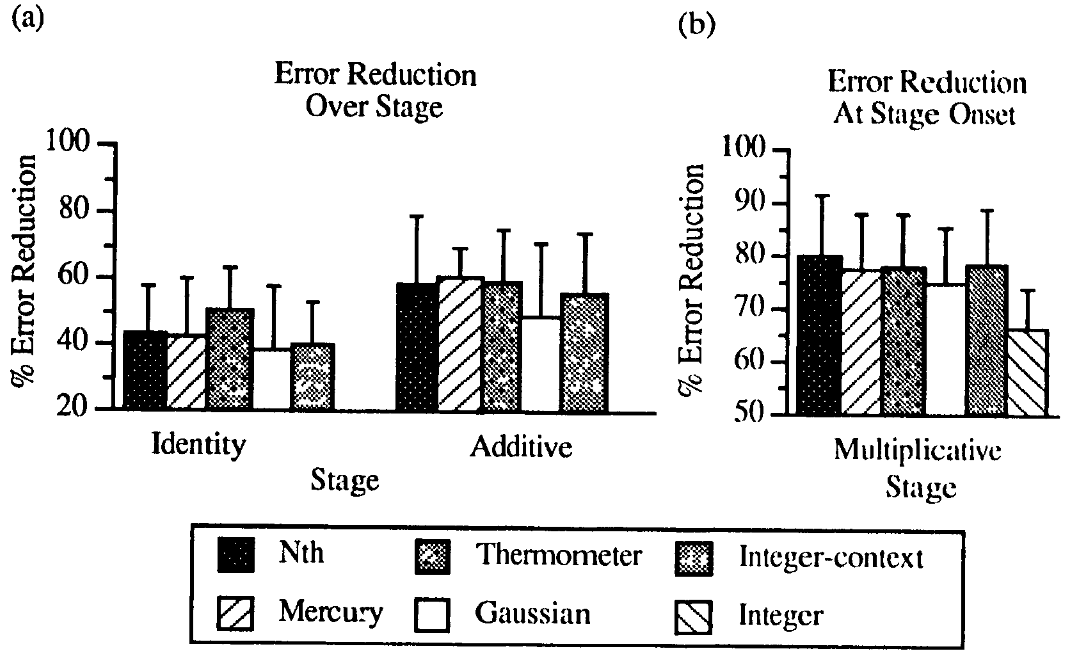


Figure B 3. Mean percent of error reduction in velocity inference patterns (a) over identity and additive stages and (b) from epoch 5 to multiplicative stage onset. Error bars are standard deviations from the mean.