*Connection Science, Vol. 11, No. 2, 1999, 149–186*

# Development of Children's Seriation: A Connectionist Approach

DENIS MARESCHAL & THOMAS R. SHULTZ

*This paper presents a modular connectionist network model of the development of seriation (sorting) in children. The model uses the cascade-correlation generative connectionist algorithm. These cascade-correlation networks do better than existing rule-based models at developing through soft stage transitions, sorting more correctly with larger stimulus size increments and showing variation in seriation performance within stages. However, the full generative power of cascade-correlation was not found to be a necessary component for successfully modelling the development of seriation abilities. Analysis of network weights indicates that improvements in seriation are due to continuous small changes instead of the radical restructuring suggested by Piaget. The model suggests that seriation skills are present early in development and increase in precision during later development. The required learning environment has a bias towards smaller and nearly ordered arrays. The variability characteristic of children's performance arises from sorting subsets of the total array. The model predicts better sorting moves with more array disorder, and a dissociation between which element should be moved and where it should be moved.*

KEYWORDS: Cognitive development, cascade correlation, seriation, sorting.

## 1. Introduction

Seriation (or sorting) was one of the key tasks used by Piaget (1965; Piaget & Inhelder, 1973) to investigate the development of children's thinking. It is generally considered to result from serial, symbolic processing in which the child acquires a set of procedures and then learns how and when to apply them in order to produce an ordered series. A number of serial, symbolic computational models of children's performance on seriation tasks have been constructed to describe the underlying information processing (e.g. Baylor *et al.*, 1973; Frey, 1964; Retschitzki, 1978; Young, 1976). In this paper, we present a connectionist model of the development of seriation abilities in children that breaks radically from the assumptions of previous models.

The relevance of connectionist modelling methods to the study of cognitive development has been established both theoretically (e.g. Bates & Elman, 1993; Elman *et al.*, 1996; Mareschal & Shultz, 1996; McClelland, 1995; Papert, 1963; Plunkett & Sinha, 1992; Shultz, 1991; Shultz *et al.*, 1995) and empirically in the

Denis Mareschal, Department of Psychology, Exeter University, Perry Road, Exeter, EX4 4QG, UK. E-mail: d.mareschal@bbk.ac.uk. Thomas R. Shultz, Department of Psychology, McGill University, 1205 Penfield Ave, Montreal, Quebec, Canada H3A 1B1. E-mail: shultz@psych.mcgill.ca.

ISSN 0954-0091 (print)/ISSN 1360-0494 (online)/99/020149-38 © 1999 Taylor & Francis Ltd

form of explicit models (e.g. Mareschal & French, 1997; Mareschal *et al.*, 1995; McClelland, 1989; Papert, 1963; Quinn & Johnson, 1997; Schyns, 1991; Shultz *et al.*, 1994b). The models described in this paper are a first attempt to apply connectionist techniques to the development of seriation. As such, one of the principal goals of the project is to demonstrate that connectionist modelling methods can be successfully used in this domain.

We propose a parallel approach to seriation. Despite the fact that seriation performance arises from a series of moves, each move can involve parallel consideration of which stick to move and where to move it. We model development of seriation with the cascade-correlation connectionist learning algorithm in which network topology is constructed as part of the learning process (Fahlman & Lebiere, 1990; Shultz *et al.*, 1995). In contrast to static networks with fixed architectures, generative networks can increase their representational and computational power during learning, thus implementing a form of constructivist development (Mareschal & Shultz, 1996; Quartz, 1993).

The model relies on a form of supervised learning analogous to imitation learning rather than reinforcement learning. Imitation is one of the principal ways that children within a society learn (Bandura, 1986; Donaldson, 1986; Rogoff, 1990; Vygotsky, 1978) and has been shown to be an effective means of teaching seriation skills to very young children (Bergan & Jeska, 1980). The networks receive information about which element to move and where it should be moved to, on a small subset of unrelated, randomly selected sorting moves. This information is consistent with what would be obtained by watching someone make an appropriate move on separate occasions. This target information on what move to make induces the networks to develop an internal representation of the sorting problem that allows it to solve complete and sequenced sorting problems. The justification for this training procedure is twofold. First, it assumes fairly minimal relevant experience. It suggests that children can learn about sorting without ever having to witness a complete sorting sequence. They can learn by attending to and encoding unrelated partial sorting events, perhaps when observing an adult or older child making a sorting move. Second, this learning scheme is consistent with Vygotsky's cognitive scaffolding hypothesis (Vygotsky, 1978). According to Vygotsky, children develop within a social context. Cognitive development consists of internalizing problem-solving solutions pervasive in the social environment. Interactions with adults and older siblings provide a schematic framework (cognitive scaffolding) on which the child can build in developing general cognitive strategies.

The modelling of seriation with cascade-correlation is an extension of a larger project that attempts to model cognitive development across a wide range of domains using a single general-purpose learning mechanism (Shultz *et al.*, 1995). Other successful cascade-correlation models include the balance scale task (Shultz *et al.*, 1994b; Shultz & Schmidt, 1991), the developing integration of velocity, time and distance information (Buckingham & Shultz, 1994), conservation (Shultz, 1998) and the identification of shifting pronoun reference (Shultz *et al.*, 1994a).

The specific network configuration required for realistic learning in each domain depends on the precise nature of the task. In our models, the network architecture is constructed by the cascade-correlation algorithm. Static connectionist networks can be effective and powerful tools for modelling development in particular domains. Their ability to develop representations through learning is well established (e.g. Elman *et al.*, 1996; McClelland, 1995; Plunkett & Sinha, 1992). However, static networks must be hand-designed for each task domain, which is

not the case when modelling development with cascade-correlation. Moreover, the set of representations that static networks can possibly learn (i.e. the network's computational power) is fixed from the onset by the network architecture (Baum, 1989; Cybenko, 1989; Mareschal & Shultz, 1996; Quartz, 1993).

The need to pass through periods of more limited representational power (as opposed to having immediate access to full representational power) may be critical when modelling children's cognitive development. For example, in balance scale simulations, cascade-correlation did better than backpropagation at reaching and staying in the final, fourth stage (McClelland, 1989; Shultz *et al.*, 1994b). Static backpropagation networks only stay in stage 4 if they miss stages 1 and 2 (Schmidt & Shultz, 1991). Another developmental domain in which cascade-correlation networks proved superior to backpropagation networks is integrating velocity, time and distance information (Buckingham & Shultz, 1994). A wide variety of static backpropagation networks were incapable of generating the stages observed in children and modelled with cascade-correlation (Buckingham & Shultz, 1996). Backpropagation networks designed with too little power were unable to reach the final multiplicative stages, and those designed with too much power were unable to capture the intermediate additive stages. Thus, it seems that the ability to grow in computational power may be necessary for simulating some developmental phenomena. Similarly, early limitations in network working memory capacity have been found to improve overall learning (Elman, 1993). However, growth in power may not be necessary in all developmental domains. Cascade-correlation provides a way to assess the need for progressive increases in computational power.

To anticipate our results, it was found that the generative power of cascade-correlation was not necessary for the successful modelling of the development of seriation abilities, suggesting that static networks could do just as well in this domain. Although most cascade-correlation networks did recruit new units, some were able to acquire mature seriation abilities without the need for added hidden units. In these cases qualitative changes in behaviour arise from small continuous changes in the underlying mechanisms of information processing (McClelland, 1995; van Geert, 1991; van der Maas & Molenaar, 1992). These results suggest that there may be somewhat different developmental paths underlying the same behavioural development. Some children may acquire new structures whereas others may develop without the addition of new structures.

## 2. Psychology of Seriation

Piagetian theory is perhaps the quintessential stage theory of cognitive development, and Piaget's account of seriation development is no exception. According to Piaget, stages are periods of consistent behaviour reflecting qualitatively different modes of information processing (see Flavell (1963) for an excellent review of Piagetian theory). Transitions between stages are supposed to be abrupt, occur simultaneously across different domains and occur in a fixed and ordered progression. An assessment of the stage concept against children's behaviour (Flavell, 1971) and neural network simulations (Shultz, 1991) suggests that although there is evidence for notions of qualitative change and ordinal progression, there is little evidence for abruptness or domain general transitions.

Piaget initiated the study of the development of seriation by presenting children with a disordered collection of sticks of different lengths and asking them to arrange the sticks in an ordered series (Inhelder & Piaget, 1969; Piaget, 1965). The results

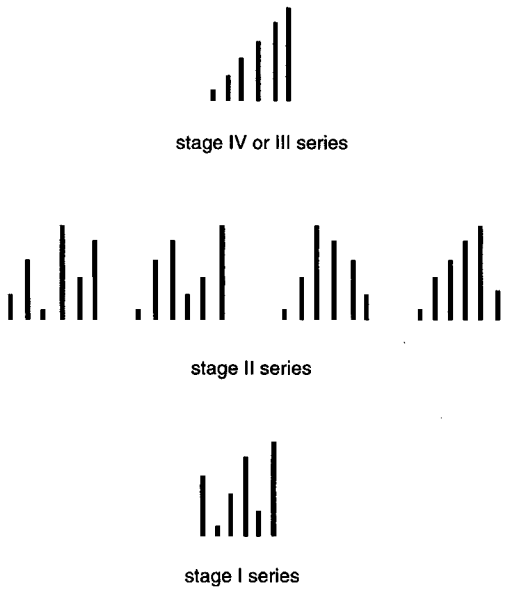



**Figure 1.** Examples of terminal series at various stages.

revealed four different seriation stages, illustrated in Figure 1. Children in the first stage (*circa* 4 years old) make no real effort at ordering the sticks and either line them up in the order they appear or move them about randomly. Children in the second stage (*circa* 5 years old) are unable to construct an overall series. However, these children succeed in combining the sticks in terms of local absolute qualities such as big or small. This leads to series of uncoordinated pairs (pairs of large and small elements), uncoordinated triplets (one large, one medium sized, and one small element), seriation based on the correct alignment of only the tops of the sticks, roof-top seriation (in which the tops rise and then descend or vice versa) and correct seriation of the first few elements followed by an inability to continue. In a third stage (*circa* 6 years old), the child succeeds in constructing a series, but only by trial and error. Finally, there is a fourth stage of 'operational' seriation (*circa* 7 years old) in which the child proceeds systematically in constructing the series by always selecting the shortest (or longest) available stick from among those not yet sorted and placing it in its appropriate place in the series.

Subsequent psychological research has largely supported Piaget's four seriation stages, but has added a number of qualifications and new findings. First, it is clear that even very young children (e.g. 2 to 4-year-olds) are able to sort small (e.g. two, three, or four) numbers of objects (Bullock & Gelman, 1977; Kingma, 1984; Kingma & Roelinga, 1983; Koslowski, 1980; Siegel, 1972). This suggests that these young children have the ability to seriate but do not apply this ability to entire, larger series until later in development. Second, seriation is aided by increasing the size differences between the items (Elkind, 1964; Kingma, 1984; Kingma & Roelinga, 1983). This underscores perceptual characteristics of seriation that may not be easily captured by symbolic operations. Third, there is considerable variation in sorting strategies at all four stages and even within individual children (Gilliéron, 1976, 1977; Kingma, 1982, 1983a; Lautrey *et al.*, 1986; Moore, 1979; Pierre-Puységure *et al.*, 1988; Retschitzki, 1978; Young, 1976). Finally, the

transitions between seriation stages are gradual, rather than sudden and decisive (Kingma, 1983b).

Thus, to be faithful to the psychological literature, seriation models need to capture the following six phenomena: periods of constant stage-like behaviour; correct ordering of the four seriation stages; transition between successive stages; better performance with increasing size differences; variation in emergent strategies; and gradual as opposed to sudden stage transitions.

Most existing symbolic computational models of seriation have only managed to capture the first phenomenon, periods of constant stage-like behaviour (Baylor *et al.*, 1973; Frey, 1964; Nguyen-Xuan, 1976; Retschitzki, 1978; Young, 1976). Typically, these symbolic models employ if–then rules, which have conditions and actions referring to particular objects and events in the world. For example, a rule implementing the systematic, operational sorting of stage 4 might specify that, faced with an unordered array of sticks and the goal of sorting them from small to large, one should move the smallest out-of-order stick to its correct position (the left-most position that contains an out-of-order stick). One rule-based model did learn to sort by modifying its own rules, but was not evaluated for psychological realism (Anzai, 1987). The contribution of the cascade-correlation models described in this article is that they manage to capture all six of the psychological phenomena listed here.

## 3. The Cascade-correlation Algorithm

Our simulations use cascade-correlation, a generative feedforward neural network algorithm developed by Fahlman and Lebiere (1990). Cascade-correlation learns to approximate a response function (defined by a set of input and output patterns) by recruiting new hidden units into the network and by modifying the connection weights in the network. In the interset of brevity we describe only the bare essentials of cascade-correlation. More detailed presentations and derivations of the algorithm can be found elsewhere (e.g. Fahlman, 1988; Fahlman & Lebiere, 1990; Hoehfeld & Fahlman, 1992; Mareschal, 1992; Shultz *et al.*, 1994b).

### *3.1. Network Construction*

Cascade-correlation begins with the minimal network topology of a single input layer fully connected to a single output layer (Figure 2(a)). The input layer includes a compulsory bias unit with an activation clamped at 1.0. A weight-adjusting algorithm then modifies the connection weights between units such that when a pattern is imposed across the input units the corresponding pattern produced on the output units will more closely match that specified by the environment. This is achieved by minimizing an error function $E$:

$$E = \sum_{o} \sum_{p} (A_{op} - T_{op})^2 \tag{1}$$

where $o$ indexes the output units, $p$ indexes the input–output pattern pairs, $A$ is the actual activation of an output unit and $T$ is the target activation for that output unit. The network is effectively learning to approximate a function defined by the set of input/output pairs.

Learning occurs in batch mode; that is, the weights are updated after a single blocked presentation of the complete set of training instances. An epoch consists

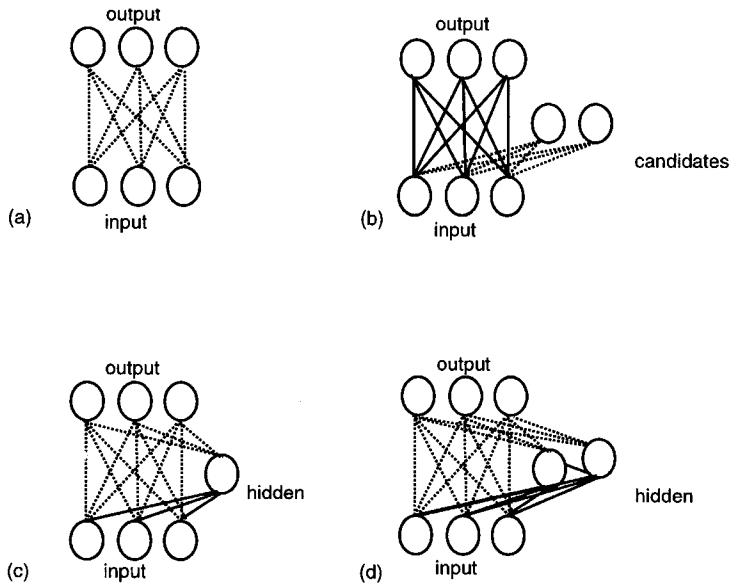



**Figure 2.** Learning in cascade-correlation. Dashed lines represent modifiable connections whereas solid lines represent frozen connections. (a) Initial output training phase; (b) input (candidate) training phase; (c) subsequent output training phase; (d) output training phase with two hidden units installed.

of one such complete presentation. Although batch learning is sometimes regarded as psychologically suspect, there is considerable psychological (Oden, 1987) and physiological (Dudai, 1989; Squire, 1987) evidence for it. Moreover, it is worth noting that even in batch learning outputs are compared to their targets independently of other patterns. Thus, the system never has to process more than one pattern at a time. It must, however, keep a running sum of the error that is eventually used to adjust the weights.

Cascade-correlation changes from adjusting the output weights to recruiting a hidden unit in either of the following two cases: (1) the learning algorithm ceases to improve the fit between the outputs produced and the desired target outputs by more than a critical amount for at least a small specified number of epochs; or (2) the learning exceeds a specified number of epochs without having reached victory. Victory is declared when activations on all output units are within a threshold (0.4 in our simulations) of their target values on all training patterns.

To build the new hidden unit, a number of candidate units are initially connected with random modifiable weights to all units in the network other than the output units (Figure 2(b)). The weights to the candidate units are then modified to maximize the absolute value of a scaled covariance (across the training set) of the activation of the candidate unit and the residual error of the outputs. The covariance function takes the form:

$$C(W,I) = \frac{\sum_o |\sum_p (h_p - \langle h \rangle)\ (e_{op} - \langle e_o \rangle)|}{\sum_{op} (e_{op} - \langle e_o \rangle)^2} \tag{2}$$

where the index $p$ runs across patterns, the index $o$ runs across output units, $e$ is

the error, $h$ is the activation of a candidate unit and angled brackets signify the mean value across all training examples.

Again, when either these covariances are no longer significantly improving or the training process has exceeded a prespecified number of epochs, the recruitment phase is terminated. The candidate unit with the highest absolute covariance has its input weights frozen and is connected to all the output units. The remaining candidate units are discarded (Figure 2(c, d)).

### *3.2. Weight Adjustment*

Cascade-correlation uses the quickprop learning algorithm to modify the weights (Fahlman, 1988; Fahlman & Lebiere, 1990; Hoehfeld & Fahlman, 1992). The weight update rules are:

$$\begin{cases} w_3 - w_2 = \varepsilon f'(w_2) \text{ if } w_2 - w_1 = 0 \\ w_3 - w_2 = \dfrac{f'(w_2)}{f'(w_1) - f'(w_2)}(w_2 - w_1) \text{ if } w_2 - w_1 \neq 0 \text{ and } \left|\dfrac{f'(w_2)}{f'(w_1) - f'(w_2)}\right| < \mu \\ w_3 - w_2 = \mu(w_2 - w_1) \text{ otherwise} \end{cases} \tag{3}$$

where the indices 1, 2 and 3 refer to three consecutive time steps, $\varepsilon$ controls the amount of gradient descent, $\mu$ controls the maximum step size and $f'$ is the derivative of error with respect to weights.

### *3.3. The Psychology of Cascade-correlation*

Although the cascade-correlation learning algorithm reflects some of what is known about neural functioning, it is not intended as a neural-level description of learning and development. Rather, it is an information processing model of performance that leaves open the question of its exact neural implementation. Note, however, that both synaptogenesis and degeneration are an integral part of brain development (Changeux & Dehaene, 1989; Dawson & Fischer, 1994). Synaptogenesis and dendritic growth are far more prevalent in the brain than commonly believed and have been proposed as neural bases for constructivist development (Quartz & Sejnowski, 1997). In this section, we focus on making explicit the nature of the information processing in cascade-correlation and its implications for cognitive development.

As with other connectionist models, learning in these networks is driven by the identification of associations in the environment. Internal representations of environmental features emerge gradually. The representations are explicitly encoded in the form of a pattern of activation across the hidden units and implicitly encoded in the associative weights linking the units together. Knowledge of the world is gradually stored in the connection weights (McClelland, 1995; Plunkett & Sinha, 1993). Beyond the ability to develop representations, cascade-correlation provides an alternative tool for addressing developmental questions. Its ability to recruit hidden units and thus increase its representational power (Baum, 1989; Cybenko, 1989; Mareschal & Shultz, 1996; Quartz, 1993) gives it the power to tackle a wider range of problems than static networks are able to.

There is an ongoing debate in cognitive development concerning the nature of

the transitions that underlie development (Keil, 1990). On the one hand, people sympathetic to Piaget's work have suggested that development coincides with dramatic transitions in the computational power of the child's cognitive apparatus (e.g. Case, 1985). They see the child as developing through successive stages of qualitatively different modes of processing where each successive mode is more powerful than the other. Observed discontinuities in behaviour (stages of performance) are interpreted as arising from qualitatively different modes of processing. On the other hand, others have argued that there is no qualitative change in the mind's computational power during development; the child simply acquires more knowledge. Development also involves restructuring, but from continuous, small, quantitative changes in the child's knowledge (e.g. Chi, 1978). Discontinuities in behaviour are thought to arise naturally from dynamic processes involving only microscopic developmental changes (van Geert, 1991; van der Maas & Molenaar, 1992).

Cascade-correlation provides a way of modelling both types of transitions in development. As with static networks, stage transitions can occur through weight changes (McClelland, 1989; Schmidt & Shultz, 1991). Stage transitions can also occur with the introduction of new hidden units (Shultz, 1991; Shultz & Schmidt, 1991). It is important to distinguish qualitative changes in behaviour from qualitative changes in processing. Qualitative changes in behaviour may or may not reflect a qualitative change in processing. Moreover, a qualitative increase in computational power does not necessarily imply a qualitative change in behaviour. It is possible to increase computational power without changing external behaviour. Indeed, the system may be performing correctly but inefficiently. As a result, processing may undergo restructuring to improve computational efficiency while leaving external behaviour unchanged.

In cascade-correlation, learning that results in only the modification of weights corresponds to a quantitative change in knowledge. The network is accumulating more information about the world and integrating that information with what it already knows. Learning that requires the recruitment of a hidden unit corresponds to a qualitative change in processing. The resulting network is computationally more powerful and can learn more sophisticated relationships than previously. For example, adding the first hidden unit results in a change from a perceptron, a network incapable of computing XOR, to a multilayer network, which can compute XOR. However, this does not imply that behaviour will necessarily change (especially if the network is already performing at a reasonable level).

Cascade-correlation provides a means of implementing a constructivist learning mechanism (Mareschal & Shultz, 1996; Quartz, 1993). Subsequent stages of performance benefit from an increase in the representational power of the network. Moreover, we can make explicit statements about the type of restructuring that occurs and the pressures that precipitate the restructuring. Qualitative changes in processing (hidden unit recruitment) occur in one of two cases (see Section 3.1). The first is when the network is no longer improving its performance. This corresponds to a situation in which the task requirements are too complex for the child's current computational prowess. The second is when output learning has gone on for too long. This corresponds to a situation in which the child's performance is inefficient. In this case, resources are added to provide a means of re-representing the already acquired domain knowledge and thus accelerate learning and increase efficiency (cf. Karmiloff-Smith's (1992) representational redescription hypothesis).

Moreover, installing hidden units in a cascade, where all subsequent units receive input from every previous unit in the network, ensures that there is a hierarchical integration of knowledge throughout development (Boden, 1982). New knowledge structures build on and incorporate old knowledge structures.

Although cascade-correlation provides a means of increasing representational power through learning, it is still possible for learning and development to occur without the need to recruit hidden units. As will be seen, this is the case for networks learning to solve seriation problems.

## 4. Simulation Design

The models presented here focus on the seriation of six-element series in order to keep the number of possible arrangements of the elements within reasonable bounds. Six is fewer than the 10 used by Piaget (Inhelder & Piaget, 1969) but more than the four used in abbreviated tasks (Koslowski, 1980; Siegel, 1972). It is a sufficiently small number to be accessible to young children, but still large enough to remain taxing for older children, and has been used in recent seriation studies (Kingma, 1982, 1983a, 1984; Timmons & Smothergill, 1975).

### *4.1. Architectural Design*

The model is to be conceived of as a specialized system dedicated to processing serial order information. The system receives information about the present state of the series, processes this information and outputs a move. A move is defined as the identification of a stick and of a position to which that stick should be moved. Moves are not executed by the network, but by auxiliary software. As in the production system models of seriation (Baylor *et al.*, 1973; Young, 1976), seriation is composed of the serial juxtaposition of these independent moves.

To implement parallel processing, the seriation system is made up of two distinct modules, each processing the same input array but responding independently of the other (Figure 3). One module computes *which* stick should be moved. A second

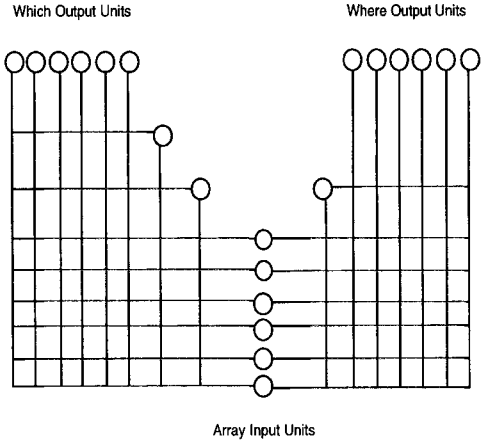



**Figure 3.** Schematic representation of the composite seriation network. The *which* module processes information independently from the *where* module. Here, the *which* network has two hidden units and the *where* network has one hidden unit.

module computes *where* a stick needs to be moved in order to extend the current series. Each component is computed independently of the other and gives partial information about the proper move to make. Because the two modules are trained independently, behavioural development of the whole can be attributed to the developing interaction of the two modules. Pilot simulations with non-modular networks failed both to learn seriation and to capture psychological regularities (Mareschal, 1992). The success of the modular approach is due to better functional decomposition (i.e. the elementary components of a complex function tend to be easier to learn than the complex function itself) and to increased generalizability of the function implemented by the networks (Dodd, 1992).

This modularization of the task was based on information processing considerations. Components computed in parallel by separate modules must be: independently computable; and necessary for completion of a sort. In the same way that form and positional object information are known to be processed separately in the visual cortex (Ungerleider & Mishkin, 1982), it is plausible to assume that *where* order breaks down in a series (based on the external form of the array) is computed separately from *which* element should be moved (based on the position of the element in the array). Admittedly there is not yet any direct evidence for a separation of *which* and *where* information in processing of seriation. Thus, the modularization of the seriation task constitutes a strong prediction of the model. Present support for modularization comes both from the model's success at capturing the six key seriation phenomena and from a new study on children's seriation errors (Mareschal & Shultz, submitted). Although modularization was hardwired in these models, such processing separations can arise as a natural part of learning in a system in which subnetworks compete during learning (Jacobs *et al.*, 1991).

The *which* and *where* modules are separate cascade-correlation networks that can develop at different rates. During the recruitment phase, the model's output does not change while candidate hidden units are being trained. This is not meant to imply that the network has entered some sort of hibernation phase in which no responses are made. Indeed, even the input training mechanism requires that activation be fed through to the outputs in order for the candidate unit activations to be correlated with residual error at the outputs. However, during the input training phase, output activations for a particular input pattern never change because the weights to the output units are frozen. These weights are frozen because the algorithm judges that no further improvement could be achieved through the modification of the output weights given the current architecture. In effect, performance has reached an asymptote. Thus, when a module is in the input training phase, its response is given by the last epoch of the previous output phase. If one modules reaches victory before the other, its weights are frozen until the other module completes training.

### *4.2. Testing the Network*

Sequential seriation is tested by presenting a standard disordered array to the network. The move output by the network is carried out by moving the selected stick and adjusting the others in order to fill the empty slots. The resulting array is then cycled back as input to the network. At each such episode, the resulting array is recorded. The cycling process is terminated when an array has appeared twice because this signifies the presence of a loop (because the networks are determin-

istic). The collection of recorded arrays constitutes a trace of a network's seriation of the initial array. Stage performance is then diagnosed from this trace of arrays. The initial array used is {5 2 4 1 6 3}, where the numbers represent the rank sizes of each element in the array. This is the array suggested for testing by Retschitzki (1978, p. 14) because it is maximally disordered from both decreasing and increasing series.

Stage diagnosis requires information concerning both the final state of the array and the method used to arrive at that final array. Unfortunately, determining which stage a seriator belongs to is not without controversy, even with children (Kingma, 1982). In particular, diagnostic techniques for distinguishing between third and fourth stage seriators, that is, between empirical and operational seriators, are not unanimously agreed upon. The Genevan school has tended to use a variety of different criteria, such as whether the child used the operational procedure, whether the series was anticipated (as tested by a preliminary drawing task), whether the child could insert additional sticks into the series, or whether there were no or very few self-corrections (e.g. Gilliéron, 1977; Lautrey *et al.*, 1986; Pierre-Puységure *et al.*, 1988; Retschitzki, 1978). Most Anglo-Saxon researchers have relied on a 'results only' or terminal series criterion, thereby collapsing the third and fourth stages into a single one (e.g. (Baylor *et al.*, 1973; Elkind, 1964; Moore, 1979; Young, 1976).

Throughout the simulations in this study, stages 1 and 2 are diagnosed as described by Piaget. Examples of terminal series are illustrated in Figure 1. Series of uncoordinated pairs (one small, then one large element), uncoordinated triplets (one small, one medium sized and one large element), roof-top seriation (in which the tops rise for three elements and then descend for the next three, or vice versa) and correct seriation of the first five elements followed by an inability to finish are coded as stage 2 behaviour. Any other behaviour not resulting in an ordered series is diagnosed as stage 1.

To distinguish between operational and empirical seriators (stages 4 and 3, respectively), both the procedure used and the number of self-correction criteria are applied simultaneously. Moreover, because the network is forced to make a move even when the presented array is correctly ordered, two different responses are taken to signify that the network has finished sorting: either the network enters a cycle of selecting a stick and putting it back in the same place, or the network enters a cycle of selecting a stick and moving it over one position, only to reverse that decision on the following move. Under these conditions, a network is classified as performing stage 4 seriation if it correctly constructs a series according to the operational procedure with at most one error from which it continues using the same operational procedure, or if it seriates in the same or fewer number of moves than required by the operational method. If it constructs a completed series in any other way, it is classified as stage 3. This epoch-by-epoch diagnosis of behaviour is carried out automatically by software designed to implement these criteria. The resulting traces of stages produced by the diagnostic procedure are then plotted (see Figures 4 and 5 and the Appendix).

Finally, stages are periods of constant behaviour that extend over time (Flavell, 1971; Shultz, 1991). The procedures above describe how behaviours at every epoch are diagnosed as characteristic of one stage or another. However, for networks to be diagnosed as passing through a sequence of behavioural stages, the behaviours must persist over a number of consecutive epochs. In our simulations, a developmental stage is scored as present if the behaviour characteristic of a stage

lasts for at least four consecutive epochs. Stage diagnosis is applied to the epoch-by-epoch diagnostic trace.

### *4.3. The Models*

*4.3.1. Parameters.* All parameters are set to their default values provided in Fahlman's (5 November 1991) version of cascade-correlation. The value of $\mu$ is set to 2.0 for the training of both output units and candidate hidden units. The value of $\varepsilon$ is set to 0.35 and 1.0 for the training of output units and candidate hidden units, respectively. The input units have linear activation functions, whereas the output units and the hidden and candidate units have sigmoid activation functions with activations ranging from $-0.5$ to $+0.5$

*4.3.2. Encoding.* The input coding represents six elements varying in size by a constant amount. A bank of six input units is used to code the arrays. The position of the units relative to the others represents the position of the elements in the array. The activation values of elements in the array are increased by $n$ times a constant amount, where $n$ is the rank of each element in the array. All arrays use an input of 1.0 to code the smallest element. The incremental value between successive elements in the array is 1.0, 0.5, or 0.25, depending on the model version.

The outputs also employ a band of six units, but with sigmoid activation functions. The *which* network has the task of identifying the stick to be moved. This is done by setting a target value of $+0.5$ for the unit spatially coding the position of the to-be-moved stick. Each of the other output units has $-0.5$ as a target value. In the *where* network, the unit spatially representing the position to which a stick should be moved has a target value of $+0.5$, whereas all other output units have a target value of $-0.5$. The response of each module is computed by selecting the unit with the highest activation. This selection mechanism could be implemented using a winner-take-all constant satisfaction network (Feldman & Ballard, 1982).

Note that the spatial left-to-right representation of stick position is something that the networks have to learn. Naïve networks with random initial weights do not, of course, share our notions of left-to-right spatial representation.

*4.3.3. The training signal.* The networks received *which* and *where* feedback on a small subset of unrelated, randomly elected sorting moves. As noted earlier, this information is consistent with that which would be obtained by watching someone else make an appropriate move. The idea is that the child observes a disordered array, predicts a response and observes a target response made by another person. This has been found to be a particularly effective means of teaching seriation skills to 2- to 5-year-old children (Bergan & Jeska, 1980).

It is not necessary to witness a complete sequence of sorting moves. Again, justification for this training procedure stems from its minimality (children to not have to witness a complete sorting sequence, but can learn from individual sorting moves) and its consistency with Vygotsky's (1978) cognitive scaffolding hypothesis, according to which the child learns from social interaction. By analogy, the networks are provided with a disordered array, predict a sorting move, and then receive a target move consistent with Piaget's operational procedure.

Little is known about the child's exact learning environment. This is partly

because of a lack of naturalistic observations of seriation learning and partly because the uptake environment (what counts as a possible learning experience) may differ significantly from the objective environment. That is, the child's learning may focus on certain aspects of the environment that are not directly obvious to an external observer. Cognitive biases such as a preference to attend to adult behaviours strongly shape the nature of the child's learning experience (Donaldson, 1986; Rogoff, 1990; Vygotsky, 1978). Additional biases in the construction of the network training set (discussed in more detail later) reflect assumptions built into the model about other cognitive biases that may underlie children's learning. One advantage of computational modelling is that it provides a means of testing the role of various possible biases in learning. The target signals provided to our networks are designed to mimic the conditions under which children learn and the type of information that they might have access to.

The network modules are trained to respond as dictated by Piaget's operational procedure, i.e. to move the smallest out-of-order stick to its correct place. The operational procedure reflects adult-level competence from the social environment in which the child is immersed. Note that this is the only target behaviour. Any other network behaviours emerge indirectly through development and are not trained on explicitly. Aside from adopting this operational rule for training, the simulations are not intended to implement Piaget's theory of seriation. The training process is presented in more detail in Section 5.

*4.3.4. The training environment.* In a pilot study, each network was trained on 100 randomly selected input/output pairs from the 720 possible six-element arrays (Mareschal, 1992). An input/output pair consists of a specific array configuration as input and as output the stick to move and the position to move it to as dictated by Piaget's operational procedure described in Section 2. For example, given the array {1 3 5 6 2 4}, the composite model should conclude that the stick in the fifth position needs to be moved to the second position resulting in the array {1 2 3 5 6 4}.

The networks in these pilot simulations failed to develop further than stage 2 performance. A closer look at the behaviour of these pilot networks revealed that they performed very well on a majority of input arrays, but failed consistently on others. They failed to identify the correct move to make significantly more often if presented with a low disorder array (i.e. a series that was almost completely finished). On the other hand, if the array was very disordered there were fewer if any errors.

In order to induce a higher level of performance, the training set was changed so as to increase the proportion of less disordered patterns. Fifty patterns are selected from those with a disorder (as measured by the sum-of-squares distance from the ordered array) less than or equal to 20, and 50 are selected from those with a disorder greater than 20. The natural, unbiased population of possible arrays of six items consists of 569 (79%) high disorder patterns and 151 (21%) low disorder patterns.[1]

This bias corresponds to the psychological assumption that a greater number of the events from which the child learns to seriate involve situations in which series are close to being completed. An example scenario may be one in which the young child watches an older child or adult playing with or manipulating different size dolls. When the dolls are completely disordered, the child is less likely to recognize them as a possible series and, hence, does not capitalize on this event to learn about serial order relations. On the other hand, if the dolls are largely ordered,

the same child is more likely to conceive them as (nearly) a series, and consequently can capitalize on this event to learn about serial order relations. The basic idea is that low-disorder arrays become meaningful before high-disorder arrays because the application of a serial order schema is more readily primed by the appearance of an array similar to the ordered target array. In this context, it is noteworthy that younger children are less likely to apply seriation schemes spontaneously than older, more experienced children are (Liben, 1975).

Connectionist learning algorithms have a tendency to over-fit the training set if trained too deeply (Moody, 1992). Thus, early training is often of more interest, particularly for developmental psychology. Only the first 150 epochs of training are reported for stage diagnosis here because networks have all reached their optimal stage performance by then. Behaviour after 150 epochs may occasionally regress to a lower stage or change to another behaviour characteristic of the same stage. These later changes are not discussed because they reflect an artifact of the training process that would not apply to learning in children. Indeed, over-fitting of a training set can only happen if the test and training instances are kept independent, i.e. if no learning occurs during testing. In children's seriation, every instance is potentially a learning instance because the child can see whether the move increases order.

## 5. Simulation Results: Size Increments and Three-element Subset Sorting

We report three simulations, one on variation in size increments, another with training on smaller, three-element sorting subtasks and a third on developmental changes in the effects of array disorder. The first two of these simulations are reported in this section. Before turning to the details of the simulations, we will first describe different types of network behaviours observed. In the interest of clarity, we begin with a short preamble illustrating the types of network behaviours observed across different conditions before reporting the actual results obtained in each condition.

### *5.1. Characteristic Network Behaviours*

This section focuses on individual network behaviours observed, whereas the subsequent sections report the performance over groups of networks. Readers are encouraged to keep the individual behaviours in mind when evaluating the group performance.

Figure 4 illustrates the epoch-by-epoch developmental profile of a single network. Each diamond corresponds to performance at one epoch. It is worth noting that, although there is a general increase in stage performance with epoch, there is considerable variability in the type of behaviour a network shows over consecutive epochs (i.e. over consecutive tests). As discussed earlier, a network must show four consecutive epochs of the same stage behaviour to be diagnosed as passing through that developmental stage.

Figure 5 shows examples of different patterns of behavioural development that can be found in the different simulation conditions. These figures depict performance at a coarser scale than that shown in Figure 4. It is no longer possible to distinguish epoch-by-epoch behaviour. However, it is important to remember that only one characteristic behaviour can be diagnosed at each epoch. Networks

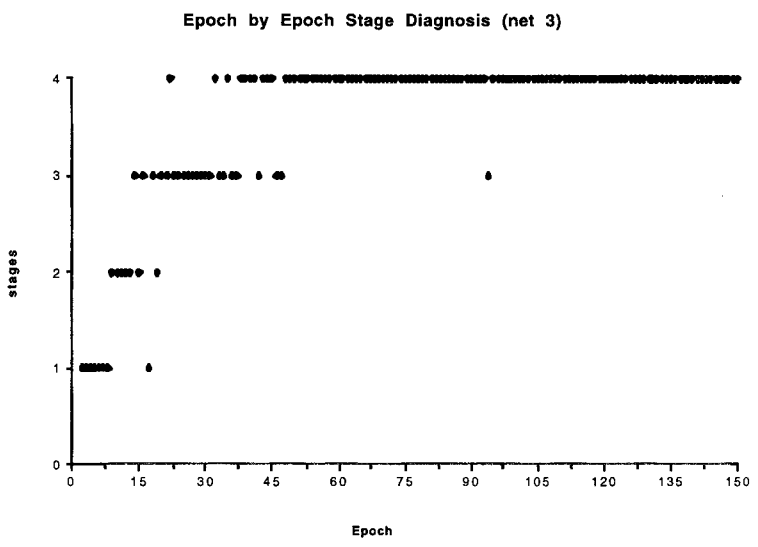



**Figure 4.** Epoch-by-epoch diagnosis of behaviour. Each diamond marks the network's performance at one epoch.

depicted in Figure 5(a–c) are all networks that learn to sort the test array correctly (i.e. they can produce the appropriate fully sorted array when presented with the initial test array)—though not necessarily according to the Piagetian dictum. Networks in Figure 5(d, e) never learn to sort correctly the test array. We now turn to describing network performance under different training conditions.

*5.1.1. The canonical condition.* This set of simulations uses the input values 1, 1.5, 2, 2.5, 3 and 3.5. There are 20 networks. Each network is trained on its own randomly selected set of patterns as described in Section 4.3.3. The mean number of epochs to victory for the *which* networks is 353, with a standard deviation of 112. The median number of hidden units recruited is two. The mean number of epochs to victory for the *where* network is 183, with a standard deviation of 55. The median number of hidden units recruited is one.

The percentage of various stage progressions for each of the several conditions is presented in Table I. Although all networks successfully learn the training set, not all networks go on to show the appropriate sorting behaviours on the novel test case. As discussed in Section 4.2, networks must show at least four consecutive epochs of a same stage behaviour to be counted as having passed through that stage. As shown in the second column of Table I, there is moderate success at producing the appropriate stage development in this condition. Fifty-five per cent of the networks develop the ability to order the array (i.e. to produce the correct final array). A little over half of those (30%) show stage 3 performance, whereas 25% do not show any stage 3 performance.

*5.1.2. The large increment condition.* In this condition, the step size between successive elements is increased in order to enhance successful seriating. This set of simulations uses input values 1, 2, 3, 4, 5 and 6. There are 20 networks. Each network is trained on its own randomly generated patterns. The mean number of epochs to victory for the *which* networks is 311, with a standard deviation of 106.

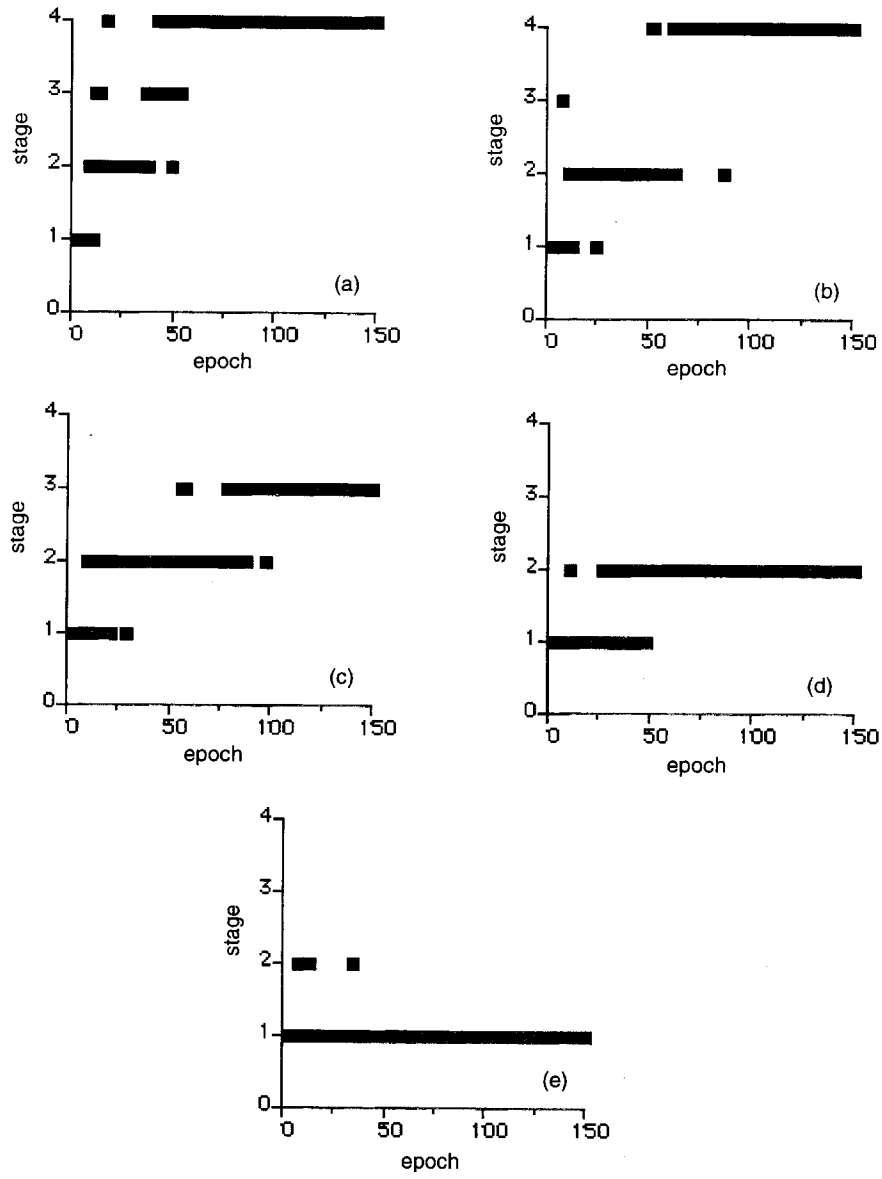



**Figure 5.** Examples of network stage development in different conditions. (a) 1 2 3 4 development; (b) 1 2 4 development; (c) 1 2 3 development; (d) 1 2 development; (e) stage 1 development only.

The median number of hidden units recruited is two. The mean number of epochs to victory for the *where* network is 80, with a standard deviation of 35. The median number of hidden units recruited is zero.

As shown in the first column of Table I, performance on the seriation task definitely improves compared to the canonical condition. Eighty-five per cent of networks learn to sort the array correctly. Moreover, as predicted by Piaget (1965), this improvement is marked by an absence of stage 3 performance. In fact, only two of the 17 networks that learn to seriate ever show a consistent period of

**Table I.** Percentage of networks showing specified developmental progressions ($n = 20$)

| | Simulation | | | |
|---|---|---|---|---|
| Stage progression | Canonical | Large increment | Small increment | Internal subset |
| 1 2 3 4 | 5 | 10 | 0 | 30 |
| 1 3 2 4 | 0 | 5 | 0 | 0 |
| 1 2 3 | 25 | 0 | 10 | 15 |
| 1 2 4 | 20 | 70 | 5 | 15 |
| 1 3 4 | 0 | 0 | 0 | 30 |
| 1 4 | 5 | 0 | 0 | 5 |
| 1 3 | 0 | 0 | 0 | 5 |
| 1 2 | 45 | 15 | 75 | 0 |
| 1 | 0 | 0 | 10 | 0 |
| Percentage learning to order an array | 55 | 85 | 15 | 100 |

empirical seriation. Of the other 15 networks, 11 (73%) show some intermittent stage 3 performance but never reach the criterion of four successive epochs of stage 3 performance. There is some stage 3 behaviour, but it appears to be randomly spread across a large number of epochs and always overlaps with other stages. This suggests that these networks never go through a stage of trial and error, and either fail to generate a global series or move directly from partial sorts to systematic operational seriation.

*5.1.3. The small increment condition.* In this condition, step size is decreased in order to investigate whether the converse relation holds, i.e. a decrease in seriation performance when the step size is too small. This simulation uses input values 1, 1.25, 1.5, 1.75, 2 and 2.25. There are 20 networks, each trained on its own randomly generated set of patterns. The mean number of epochs to victory for the *which* networks is 531, with a standard deviation of 150. The median number of hidden units recruited is three. The mean number of epochs to victory for the *where* network is 286, with a standard deviation of 21. The median number of hidden units recruited is 1.5.[2]

The third column of Table I shows a reduction in performance as compared to the canonical condition. Only three networks (15%) are ever able to construct a completed series. Most networks (15) remain at a stage 2 level of uncoordinated seriation.

### *5.2. Simulation 2: Internal Subset Seriation*[3]

The less than optimal performance in the previous simulations suggests that a key piece of the seriation puzzle is missing. One clue is that even very young children can seriate a small number of elements (e.g. three elements). A second clue is Piaget's (1965) suggestion that one of the causes of stage 2 seriation is that children are unable to extend order on small sets to the entire series, relying on local size comparisons. Together, these psychological results suggest that operational seriation on a subset of the elements in a larger array may be a source of behaviour diagnosed as stage 3 (empirical seriation). Phrasing this in terms of the cognitive

bias discussion in Section 4.3.3, these findings suggest that children have a cognitive bias towards recognizing and learning from the sorting of small sets of elements rather than large sets of elements. It is also possible that small arrays are more commonly available in the child's environment than are large arrays.

To test this hypothesis, the training set in the large increment condition is modified to include three-element series. In addition to the original 50 patterns drawn from the low disorder category and 50 drawn from the high disorder category, there are 20 patterns selected randomly from the 24 possible three-element series (i.e. six orders by four positions). These series are created using the integers 1, 2 and 3 to code the elements and 0 to code the resulting empty spaces in the six-element arrays. There are never any spaces between elements, and any ordering occurs solely within the limits of the series. For example, given the array {0 0 2 1 3 0} the correct operational move is to move element 1 to the position occupied by element 2. There are 20 such networks. The mean number of epochs to victory for the *which* networks is 614, with a standard deviation of 126. The median number of hidden units recruited is five. The mean number of epochs to victory for the *where* network is 252, with a standard deviation of 94. The median number of hidden units recruited is one.

Stage diagnosis is displayed graphically in the Appendix for all 20 networks. In this set of simulations all networks exhibit behaviour in all four of the stages, and stages are spread over a larger time span. All networks are able to produce the corrected final (sorted) array. There is much more of a mixture of different stages. Nevertheless, six (30%) of 20 networks (networks 3, 7, 9, 15, 16 and 19) progress through stages 1, 2, 3 and 4. There is often a large amount of overlap between stages, suggesting that for much of the developmental period no stage is uniquely characteristic of the network's behaviour. Transitions between stages are soft, with stages merging gradually into one another. Note also that network 16 regresses to stage 3 after continued training.

Three (15%) of 20 networks (networks 1, 10 and 17) progress through stages 1, 2 and 4. These networks show stage 3 behaviour, but too inconsistently to meet the stage requirements of four consecutive epochs at the same level. Three (15%) of 20 networks (networks 8, 14 and 18) progress through stages 1, 2 and 3. These networks sometimes show an early but brief stint of stage 4 behaviour followed by regression either to stage 3 or to lower levels. This suggests that over-training may have occurred. Network 14 begins with a brief period of stage 2 behaviour that precedes the 1, 2, 3 progression. This premature stage 2 behaviour is due to the initial random seeding of the weights in the network. Six (30%) of 20 networks (networks 4, 5, 6, 12, 13 and 20) progress through stages 1, 3 and 4. Again, there is evidence of stage 2 behaviour in these networks, but not enough to constitute a stage according to our criterion. Note that network 6 regresses to stage 3 with continued training. The remaining two (10%) of 20 networks (networks 2 and 11) show only two of the four possible stages. That is not to say that the other stages are totally absent, but only that they are very briefly present and always adjacent to another form of behaviour.

### *5.3. Discussion of Simulation Results*

Table I lists the percentages of networks exhibiting the different developmental progressions found in the four simulations (i.e. showing at least four consecutive epochs of same stage behaviours). The essential shortcoming of the large increment

model is an absence of empirical (stage 3) seriation behaviour. The distribution of networks in the canonical conditions confirms that empirical seriation (stage 3 performance) can be induced by decreasing the size difference between successive elements. Moreover, the distribution of networks in the small increment conditions shows that deterioration continues as step size decreases. These results suggest that modular cascade-correlation networks respond to decreasing step sizes in seriation tasks as children in the same way that children do.

Results for the internal subset simulation show that empirical seriation can be induced by the application of a systematic procedure (in this case the operational procedure) to a subset of the elements in the series. Indeed, having the ability to perceive and systematically order internal subsets within the full array of available elements could well be one of the sources of empirical seriation in children. As noted in Section 2, young children can seriate three elements and yet fail to extend this procedure to larger arrays.

Network stage transitions are soft. Transitions consist of the slow and gradual merging of one style of behaviour into another. One type of behaviour is often accompanied by the presence of other types of behaviours. During the development period, there are few regressions to consistent lower stage behaviour, although isolated instances of regression are common. Similar observations have been reported in children by Kingma (1983b), who found that in a longitudinal study of different seriation tasks the majority showed a gradual development of seriating abilities as opposed to all-or-none acquisition.

Network behaviour is flexible. As noted in Section 2, children show considerable variability in seriation strategies within a stage. Even stage 4 seriators use different strategies (Gilliéron, 1977; Lautrey *et al.*, 1986; Moore, 1979; Pierre-Puységure *et al.*, 1988; Retschitzki, 1978). Random selection strategies and partial seriation are present in children of all ages, even those well into the operational stage (Kingma, 1982). The only way that previous models have addressed this issue is to hand-tailor a distinct model for every child (e.g. Young, 1976). However, once a production system is designed to mimic a stage behaviour, it does not produce any other behaviours characteristic of that stage.

The networks in the internal seriation condition show abundant variation in seriating behaviours both between networks and within networks from a longitudinal perspective. For example, recall that there are four ways to exhibit stage 2 behaviour (Figure 1). The median number of different stage 2 diagnoses displayed by the networks is two. Ninety per cent of the 20 networks at some epoch produce uncoordinated pairs, 30% produce uncoordinated triplets, 15% produce a roof-top series and 55% produce a partial ordering of the first four sticks in the series.

Stage 4 performance also shows between and within network variability in behaviours. The median number of different stage 4 diagnoses is 2.5. With a limited number of elements in the test series (six), it is difficult to tease apart the different seriating strategies because they often predict the same moves. However, all of the 20 networks deviate from pure operational seriation at some time during stage 4. The deviations arise from the execution of an apparently random move that increases the total number of moves required to complete the sort. In fact, only 25% of networks never seriate operationally without any deviations. Thus, even when networks seriate according to stage 4, they do not always follow the dictated strategy precisely and can include an apparently random move. The nature of that random move gives rise to different observed behaviours. Random selection

is observed in children of all ages, even those well into the fourth stage of seriation (Kingma, 1982).

It is important to note that networks can show more than one type of within stage behaviour because they are constantly adapting to the environment. Thus, over subsequent epochs the network's experience (and hence its knowledge) differs. Progressions and regressions occur within a stage because of continuous small changes in the connection weights. What is especially interesting is that the networks were only trained on a selection of moves dictated by the operational method. No network was ever trained on any of the various stage 2, stage 3 and alternative stage 4 behaviours that they eventually showed. The individual differences in terminal strategies reflect the individual learning experiences of each network. The addition of hidden units did not necessarily correspond to stage transition. The role of hidden units is discussed further in the next section.

## 6. Network Analysis

To get a better idea of how the arrays are processed, connection weight diagrams were generated at selected epochs. These diagrams provide a visual display of the sign and strength of the weights in the network. Hence, it is possible to follow the weight evolution that leads to the particular solution settled on. The analysis focuses on the internal subset seriation simulation because it produces the largest proportion of networks progressing through all four stages.

The initial architecture of both the *which* and the *where* modules consists of a bias unit and six other input units and six output units. In the diagrams in Figure 6, the bias unit is labelled 0 and the other input units are labelled 1–6 from left to right, respectively. The output units are also numbered 1–6 from left to right. Because the inputs are initially fully connected to the outputs, there are $7 \times 6 = 42$ connection weights at this point. In Figure 6, a black square represents a negative weight, whereas a white square represents a positive weight. The relative magnitude of a weight is represented by the relative size of the side of the corresponding square, with the size of the side increasing in proportion to the magnitude of the weight.

As a prelude to the investigation of network structure it is worthwhile pondering in more detail the nature of the respective subtasks (i.e. identifying a stick and a destination). Understanding the task requirements simplifies interpretation of network weights. An output unit in the *which* module should have a positive activation if the stick in the input array whose position it is coding is the smallest stick not yet in order. Otherwise, an output unit should be negative. Thus, an output unit learns to attend to its corresponding input unit.

In the *where* module, an output unit coding a position should be positive if all the previous positions are correctly filled and the position corresponding to that unit is not correctly filled. Otherwise, an output unit should be negative. Thus, an output unit learns to attend to its corresponding input unit but also to all of the input units of lower rank.

In mature networks, one could thus expect to find large weights along the diagonal of a weight diagram of the *which* module. In a weight diagram of a mature *where* module, there should be large weights not only along the diagonal but also above the diagonal.

The weight diagrams are remarkably similar in the structure they reveal for different networks in the internal subset condition. Presumably, macroscopic

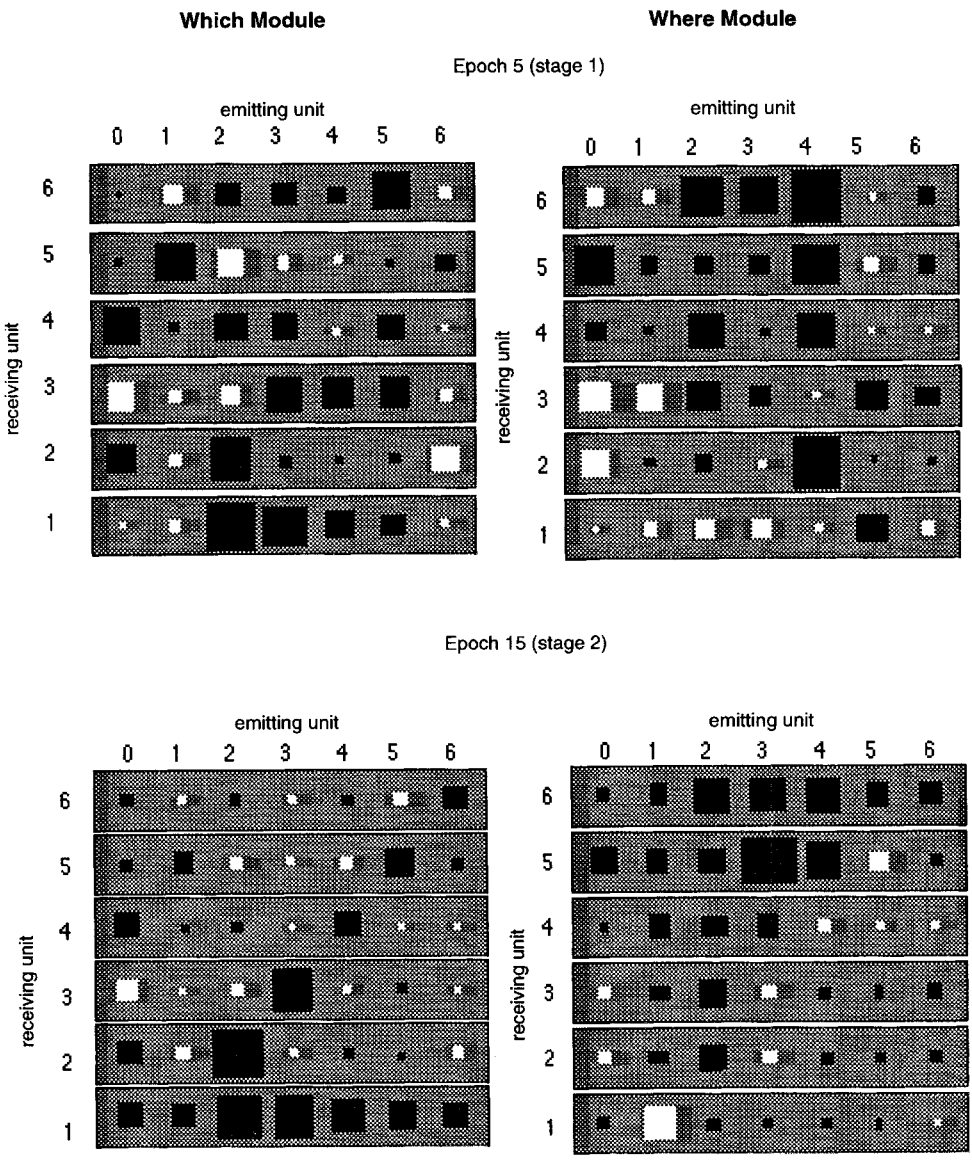



**Figure 6.** Hinton diagrams of the connections in network 3 of the internal subset simulation at selected epochs.

differences in diagnostic behaviour hinge on minor differences in weight magnitudes. Figure 6 shows the weight diagrams derived from network 3 of the internal subset simulation (selected because it shows behaviour at all four stages). These diagrams are representative of those obtained for other networks. The diagrams on the left of the figure are derived from the *which* module, whereas those on the right of the figure are derived from the *where* module. Successive diagrams are drawn at epochs 5, 15, 35, 125 and 145 because these coincided with periods of consistent stage behaviour.

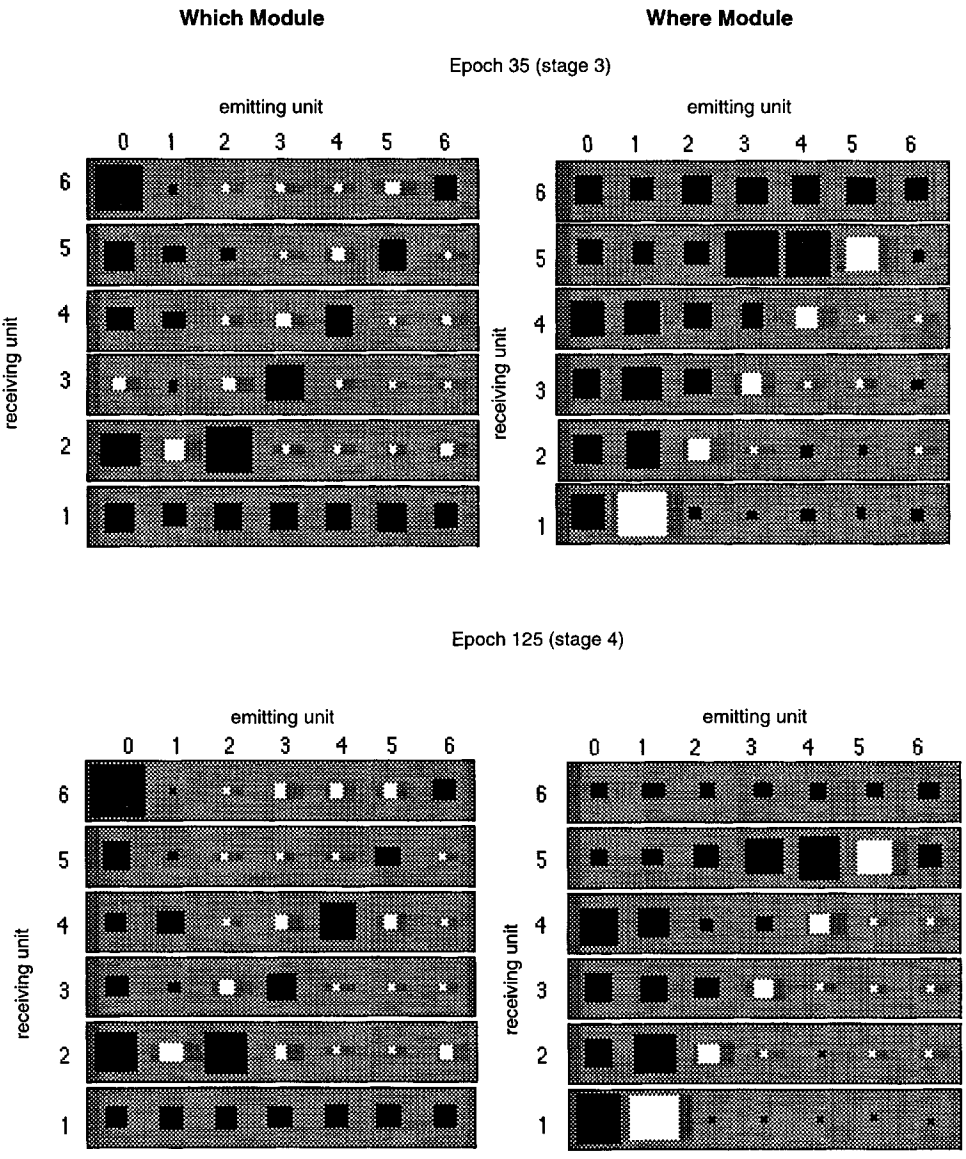



**Figure 6.**—*(Continued)*

### *6.1. The* Which *Network*

Early in the development of the *which* network (epoch 5), the weights still appear to be quasi-randomly distributed. This is reflected by stage 1 performance. The network has not yet organized its behaviour sufficiently to order much of a series.

Considerably more structure becomes evident by epoch 15. As a general rule, development in this set of simulations occurs early in training. Note that all the connections leading to output unit 1 are negative. The other output units have small weights of varying sizes everywhere except from the input unit whose position they are coding, thereby defining a diagonal. Interpreting this is straightforward. The first stick is never selected to be moved, so all weights into the first output

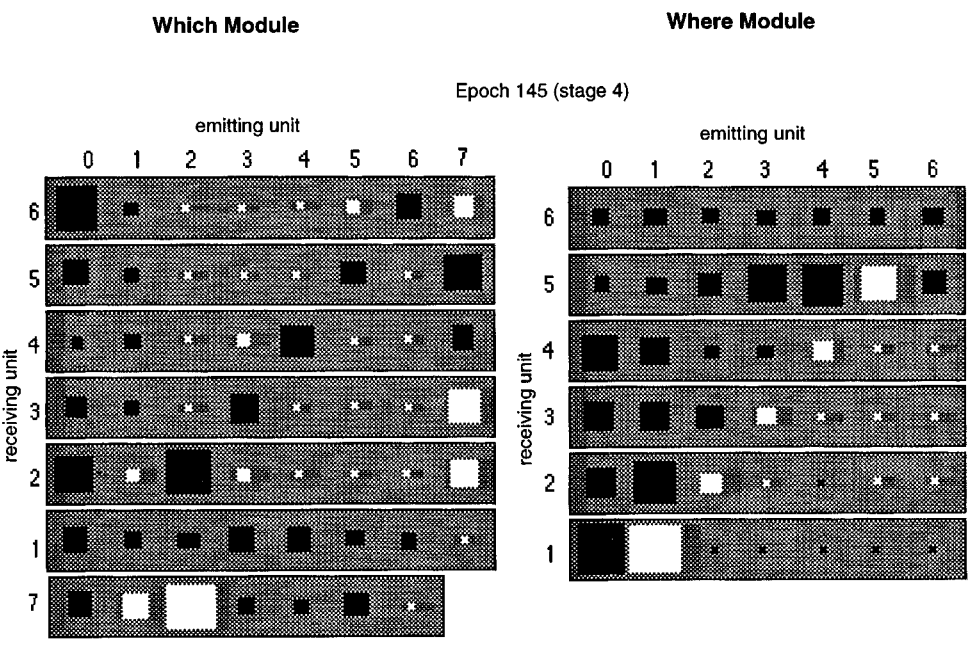



**Figure 6.**—*(Continued)*

unit are negative. For the other output units, the large negative weights along the diagonal ensure that if anything but a small stick is in that position the unit will fire negative. Thus, the strategy is to dampen out any signal other than that arising from the smallest stick.

Also, this network has begun to develop a gradient in the magnitude of the weights along the diagonal. The diagonal weights get smaller progressing from output unit 2, to 3, and to 4. This establishes an order in which out-of-place sticks are identified. For example, the weight to output unit 2 is very large and so only sticks of small length (less than 2) fail to dampen activation of unit 2. The weight into output unit 3 is a little smaller so only sticks with length less than 3 fail to dampen the activation. This ordering breaks down at the longer sticks. This is consistent with stage 2 behaviour in which the child can correctly apply a procedure to the initial elements of the series but fails to extend it to the entire series.

By epoch 35, the ordering of weights along the diagonal extends to all of the outputs units. All the weights from the bias unit except the weight to output unit 3 have been trained to negative values and appear to be tuned to the precarious balance required for the dampening mechanism to function correctly. The remaining weights activate the correct output unit, especially if there are bigger items to the right of the critical position (i.e. no smaller sticks to the right of that position). Thus, positive weights are required on the right of the diagonal. The appearance of stage 3 performance therefore coincides with the extension of an ordering process across the entire array as well as the consolidation of the delicate threshold mechanism through modifications of the bias weights.

Few changes occur between epochs 35 and 125. The essential difference is a change of sign in the weight joining the bias unit to the third output unit. Performance is now diagnosed as stage 4 rather than stage 3. This suggests that the transition from empirical to operational seriation is due to refinement of

procedures already present during stage 3 (and in fact partially present during stage 2) rather than radical structural reorganization. This observation is in agreement with Koslowski's (1980) suggestion that the development of children's seriating abilities is partially driven by improvement in the precision to which operations are carried out.

Epoch 145 still corresponds to stage 4 performance but now shows the presence of a hidden unit (labelled 7). The hidden unit is particularly sensitive to the presence of anything but a small stick in position 2 as well as to a proper ordering of the elements in positions 3, 4 and 5. If one of the sticks in these positions is out of order, the hidden unit dampens the firing of position 6 and enhances activation in positions 4 or 5. The purpose of this hidden unit is to fine-tune procedures that have already been in place since at least epoch 35. The hidden unit conveys the need for added power to extend ordering accurately to the last few elements in the series. It does not seem to be critical for stage 4 performance and may in fact be an artifact of the previously described over-training.

### 6.2. *The* Where *Network*

Turning now to the *where* network, it is again clear that the weights at epoch 5 are still quasi-randomly distributed. As in the *which* network, this results in stage 1 performance. By epoch 15, the essential solution structure has begun to appear. Note that all the weights leading into output unit 6 are negative. This makes sense because location 6 is never a target of the smallest stick out of order. Once again the diagonal plays a distinctive role. The weights below the diagonal are all very small and hence contribute little to computing activation of the output units. The weights above the diagonal are of moderate to large magnitude as the network discovers that this task requires an output unit to attend to its corresponding input unit as well as all the input units of lower ranking positions, but not to the input units of higher ranking positions.

The weights along the diagonal are positive, with one exception. Hence, any output unit tends to be positive if there is a large stick in the position coded by that unit. Keeping in mind that the role of this network is to identify positions to which small sticks are to be moved, it is easy to understand how the strategy of positive weights along the diagonal works. If there is a larger stick in the corresponding position, then it is not a small stick and a small stick should be moved there. Thus, the relevant output unit should fire to signal that a small stick is required. This part of the weight structure is devoted to identifying positions that could potentially receive a small stick.

The other half of this network's task is to verify that all the sticks to the left of the target position are in order. If they are not in order, then the output unit should be inhibited from firing because some lower ranking position is required to be filled prior to this position. This is the role of the negative weights above the diagonal. At this epoch of stage 2 performance, the weights above the diagonal are all negative and of the same size or larger than those along the diagonal. Thus, if any stick larger than the one in the target position should be to the left of that position, the resulting negative activation exceeds the resulting positive activation and the corresponding output unit is inhibited. However, this mechanism is not guaranteed to work if the sticks to the left of the target position are all smaller than the one in the target position (i.e. the network would tend to identify position 4 in both arrays

{1 2 3 5 6 4} and {1 3 2 5 6 4}). Interestingly, such mistakes only become possible when the array exceeds lengths of three or four elements.

By epoch 35, the problem of testing for proper order in the lower ranking positions is solved by a decreasing gradient in weight magnitudes. Note that negative weights leading to output units 3 and 4 decrease in magnitude with a progression from left to right in the input array. A gradient in this direction is sensitive to ordering by increasing size of the sticks in the array, with any exception resulting in an excess of negative activation thereby inhibiting the output unit. Unit 5 appears to be dealt with in a different way. In contrast to other units that can fire positively when presented with different input arrays, this unit should only ever fire positively when presented with the array {1 2 3 4 6 5}. Because the last two values (5 and 6) have the smallest proportional increment, they will be the most difficult to tell apart. In order to resolve the case in which the array is completely ordered from the case in which the last two values are swapped around, the network grows moderately large weights from those two input units. The larger weights will magnify the differences and help distinguish between these two cases. However, it is most important to magnify that difference when the first four elements are already sorted. The left to right increasing gradient ensures that the negative activation entering the unit is a maximum when the four preceding units are correctly ordered. In this case, only the largest input activation (6) multiplied by the large positive weight from input 5 will succeed in tipping the balance towards positive activation. Also, the weight connecting output unit 2 to its corresponding input unit is now positive so that all weights along the diagonal are positive. The resulting behaviour is now diagnosed as stage 3.

Once again, only fine-tuning occurs between epoch 35 (stage 3 performance) and epoch 125 (stage 4 performance). The weights leading from the bias unit have been tuned to adjust the firing thresholds of the output units. A clear example of this is found in the weights leading to output unit 1. Notice that the weights leading to the output from the bias unit and first input unit are of the same magnitude but of opposite signs. Therefore, when stick 1 is in position 1 the resulting activation is counterbalanced by that of the bias unit. The remaining weights (with larger input activations) determine the unit's activation. Because these are all negative, the unit fires negatively. If, on the other hand, some other stick is in the first position, then the resulting activation exceeds that arising from the bias unit and the corresponding output unit fires positive. No differences appear between epoch 125 and epoch 145. In all networks (and in fact in all simulations), the *where* network completed its training before the *which* network.

### *6.3. Hidden Units*

The weight diagram analyses show how a representation of the task is developed. Knowledge of what action to carry out is encoded in the connection weights linking the input and output units. However, no clear functional role is identified for the hidden units recruited into the network. A further 10 networks were trained in the internal seriation condition in order to investigate the environmental pressures that might induce the recruitment of more computational resources into the network.

These networks were only trained for the first 150 epochs used to diagnose stages. Under these conditions, eight of these 10 networks recruited a *which* hidden unit at epochs ranging from 120 to 145 (the other two networks recruited no hidden unit in the *which* module). In the *where* module, six networks did not recruit

hidden units whereas the other four recruited a hidden unit at epochs ranging from 144 to 149. In only one network did the recruiting of a hidden unit (a *which* hidden unit) correspond to an immediate change in seriation strategy: performance changed from one type of stage 4 behaviour to another as result of this hidden unit. No other recruited units produced any immediate diagnosable change in macroscopic behaviour. That is, there is a qualitative change in computational power, but no qualitative change in observable behaviour.

The hidden units are introduced at a point in training where error is already low and is approaching asymptote. However, even at this point residual error remains. The added hidden units do increase the rate at which this residual error is reduced. This becomes clear when examining the relative error drop-off rates for the 10 epochs prior to the introduction of a hidden unit and comparing it to the rate for the 10 epochs immediately after the introduction of the hidden unit. The average post-unit drop-off rate in the *which* module is 4.42 times larger than the pre-unit drop-off rate. The effect is less pronounced but similar in the *where* module, where the average post-unit drop-off rate is 1.37 times larger than the average pre-unit drop-off rate.

### *6.4. Summary of Network Analyses*

This analysis reveals that networks use local size comparisons between sticks. The extent to which these comparisons are extended throughout the whole array depends on the developmental state of the networks. Systematic comparisons are built up progressively from the smaller end, resulting in early stage 2 performance due to an incomplete extension of the comparisons. Note that this is in accord with Trabasso's (1977) suggestion that a linear order mental representation of series is progressively built up from the end. Stage 3 evolves into stage 4 performance because of fine-tuning of an already present structure. These adjustments correspond to a need for more precise perceptual discriminations (Terrace & McGonigle, 1994). This suggests that a perceptual saliency dimension may be involved in the developmental of seriation.

An increase in representational power is not necessary for successful development on this task. However, learning continues even after the networks are performing according to a diagnosable optimal strategy (e.g. using the operational method). This learning is driven by the occasional errors that the network continues to make. Computational power increases to provide a more efficient means of processing that reduces residual errors at a faster rate.

## 7. Simulation 3: Disorder and the Development of Seriation

In discussing the biases in the training set (Section 4.3.4), it was noted that the disorder of the array presented to networks in a pilot study was a significant factor in determining whether a network would choose the correct move (Mareschal, 1992). This is one of a number of perceptual effects evident in our network simulations. In this section we investigate the development of the disorder effect. We focus on the large increment condition where the different input arrays are most distinct from each other.

Generalization is tested on 15 networks. At the end of learning, these networks identify the correct move to make on 691 (96%) of the 720 possible six-element

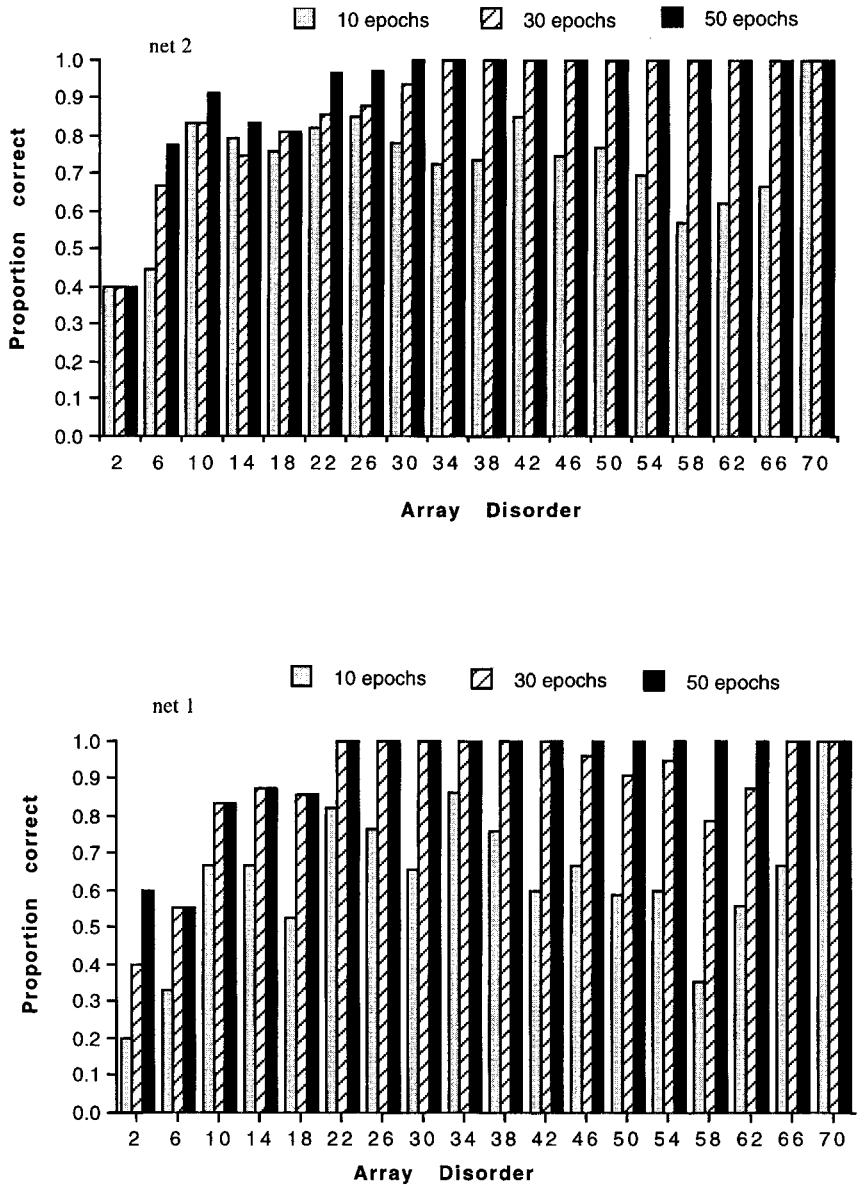



**Figure 7.** Development of three networks' generalization abilities. Early in development, passing and failing are equally distributed along the disorder dimension. Later, failures tend to cluster in the lower disorder region.

arrays they are presented with. It is worth remembering that these networks have only been trained on 100 (14% of the total) randomly selected arrays.

Figure 7 shows the development of generalization abilities during the early part of learning for two networks as a function of the disorder of the array presented. The disorder of an array ranges from minimum of zero to a maximum of 72 in incremental steps of two. In order to reduce the complexity of the figure, only alternate disorder values are plotted in Figure 7 (i.e. at increments of four instead

of two). The proportion of correct responses is presented as a function of these disorder values. Proportional scores are used because there are varying numbers of possible arrays for a given disorder. Initially, errors occur across the whole range of disorder values. However, by 50 epochs these networks are performing with almost 100% accuracy on arrays with disorder values above 20. Errors do persist, but they tend to occur on arrays with disorder values of 20 or less.

These results suggest an alternative interpretation for why children fail seriation tasks. A child begins sorting the array correctly, but as sorting proceeds the child becomes more likely to produce an error simply because the current array has less and less disorder. This could be true even when the child is applying an operational procedure. The child may stop before the array is completed by adult standards or may even stumble on to the finished series, but in so doing would be classified as an empirical seriator. This scenario is consistent with studies reviewed in Section 2, showing that young children possess the requisite skills for seriating small arrays but fail to apply these skills fully to larger arrays.

One conclusion from this section is that unbiased training sets lead to biased results: low-disorder arrays are poorly sorted. Hence, in order for the network to learn, the training set must be biased towards the lower disordered examples, as found in the previous sections. One can see why disorder has such an impact on the network's performance by referring back to the connection weight analyses (Section 6). Networks solve seriation by representing the ordered structure of a target array within their weights. The weights act as negative template filters against which the input arrays are compared. If the input array does not match the template, an appropriate move is computed as activation from the inputs filters through the system. This filter mechanism works well for arrays that are very different from the internalized order template, but becomes increasingly unreliable as arrays come to resemble the target.

## 8. General Discussion

The principal goal of showing that connectionist methods can be successfully applied to modelling seriation has been achieved. The cascade-correlation models presented in this paper capture all six seriation phenomena identified in the psychological literature. They show: periods of constant stage-like behaviour; correct ordering of the four seriation stages; transition between successive stages; better performance with increasing size differences; variation in emergent strategies; and gradual as opposed to sudden stage transitions.

The six seriation phenomena found in cascade-correlation models derive naturally from the information processing characteristics of connectionist networks. Periods of constant stage-like behaviour result from the sensitivity of networks to statistical regularities in the training environment and the fact that learning results in small changes. Correct ordering of the four seriation stages results from the gradual extension of seriation ability (as represented in the network's connection weights) to more and more of the array.[4] Transition between successive stages is due to the dual process of connection weight adjustment and occasional recruitment of new hidden units. The better performance with increasing size differences results from the continuous coupling between input and output in connectionist networks, which ensures that network performance varies continuously in response to changes in the input (Shultz *et al.*, 1994b). Large input differences create clearer activation patterns on the hidden units and more decisive responses by the output units.

networks show variation in strategies because such strategies are not built in as actual computational mechanisms, but rather emerge from underlying topological and weight adjustment mechanisms. Strategy-based stages are merely diagnosed from behavioural performance; they are not the actual means of computation. Often all of the children's seriation behaviours cannot be classified within the strategy inferred by the observer (Greenfield *et al.*, 1972; Moore, 1979). Our model suggests that the apparent diversity of strategies does not necessarily reflect different underlying information processing. Rather than positing a wide range of strategies used in different contexts, it is more parsimonious to posit a single mechanism to account for the different strategies. Finally, networks exhibit gradual, as opposed to sudden, stage transitions for the same reason. Network learning is gradual and so the emergence of diagnosed strategies is also gradual.

Earlier computational models captured only one seriation phenomenon, consistent stage performance. It may be possible to design serial symbolic models that would capture these same six seriation phenomena. Ultimately, however, models must be evaluated on the ease or naturalness with which the required phenomena emerge from the proposed information processing mechanisms.

We approach seriation from a radically different perspective than that of either Piaget or serial symbolic processing. We suggest that, although seriation is executed through a series of actions, the computations underlying choice of each action are carried out in parallel. The required information components are computed in separate modules that process the same input simultaneously and in parallel. In particular, we hypothesize a *which* versus *where* dissociation in the processing of arrays to be sorted. We arrived at this architectural modularization through information processing considerations. The fact that the model captures six seriation phenomena leads us to predict a similar dissociation in children's seriation.

Unlike Piagetian accounts, our model also implies that children can develop through all four stages of performance without requiring an increase in representational power. The same computational apparatus is at work in a child performing at either a stage 1 or a stage 4 level. The basic skills required for seriation are present as early as stage 2. Seriation development is essentially due to an increase in the precision of processing rather than to any fundamental reorganization of knowledge. Hence, these models provide evidence of stage-like development arising through micro-genetic changes. This is radically opposed to Piaget's initial conclusions and provides computational support for some more recent accounts of seriation development as involving a refinement of initial procedures (Koslowski, 1980; Terrace & McGonigle, 1994).

Our model also suggests a new explanation for typical seriation errors observed in children. Rather than lacking seriation operators, the model suggests that children fail to apply these operators as the series gets closer to completion. Stages 2 and 3 should not be characterized by the absence of systematic seriation procedures. Instead, these stages appear because a systematic procedure is being applied, but only to subsets of items, not to the entire array.

The development of seriation expertise is closely constrained by the nature of the learning (or uptake) environment. That environment may differ from the objectively observed environment due to biases introduced by a teacher who selects the kinds of problems the child is faced with (Rogoff, 1990), or by built-in cognitive biases that limit the experiences that can be learned from. Our best model points to a cognitive bias towards learning from smaller arrays and from less disordered arrays. Children may more easily recognize and learn from the sorting of small

arrays that are commonly available in the environment. Also, the child is more likely to perceive the completion of a nearly sorted array as an instance of seriation and to learn from that instance. Moving items in a highly disordered set does not provide feedback for learning about sorting unless the child perceives the array as a potential series and not just as a collection of unrelated items. Finally, the ability to demonstrate expertise is closely tied to perceptual constraints such as array size, increment size and disorder of the current array.

The connectionist models we describe rely on direct environmental information about sorting. Such information could arise when the child watches an adult or older sibling sort objects. Note that the child does not have to witness a complete set of moves to learn to seriate. The child only needs to be exposed to small random sets of independent moves. These could be accumulated over a number of different observations. We focused on learning through imitation rather than reinforcement learning for a variety of reasons. First, it is difficult to assess what might constitute reinforcement for the child's behaviour and when such reinforcement may occur. In contrast, imitation is a more passive form of learning in the sense that the child does not need to act but only observes someone else act, perhaps predicting what the other person will do. Also, imitative learning does not require reinforcement of the child's behaviour by some other agent. Imitation does, however, include a more complete and informative response target than does reinforcement. Reinforcement is essentially a binary signal that a response was correct or incorrect. In contrast, learning by imitation supplies full information on the nature of a correct response. Moreover, imitation is one of the principal means by which children are known to learn (cf. Bandura, 1986; Rogoff, 1990).

Empirical studies in progress with 4- to 7-year-old children seem to confirm the model's predictions about disorder and modularity (Mareschal, 1992; Mareschal & Shultz, submitted). Just like the networks, children are more likely to identify an inappropriate move when presented with a nearly sorted array than when presented with a highly disordered array. Moreover, as with our networks, children's ability to identify which stick should be moved in a partially ordered array emerges later than their ability to identify where a stick should be moved to extend the sort.

In summary then, we propose a parallel processing approach to seriation that breaks radically with previous work. This framework is implemented using the cascade-correlation generative connectionist algorithm. The model captures known empirical phenomena much better than previous models do. Moreover, it provides a parsimonious account of a wide range of seriation strategies by positing a single underlying information processing mechanism. Although this is clearly a first attempt to model seriation using connectionist methods, it is impressive in the number of phenomena that can be captured by appealing to a simple learning mechanism.

The success of our seriation model provides additional support for cascade-correlation as a domain general model of processing and transition in cognitive development. The cascade-correlation algorithm has now been used to model successfully a wide range of cognitive developmental phenomena. In some cases, such as the balance scale, conservation, integration of velocity, time and distance information, and acquisition of personal pronouns, recruitment of new hidden units has been necessary for accurate modelling of stage progressions. Although some hidden units were recruited during seriation acquisition, no novel functional role could be attributed to the hidden units, outside their ability to reduce residual error at a faster rate. The corresponding increase in the network's computational

power did not produce a qualitative shift in diagnosable behaviour. Hence, we suggest that seriation could also be modelled with static networks. Although the generative properties of cascade-correlation do not appear to be necessary for realistic seriation development, this was not clear beforehand. Modelling with cascade-correlation provides a vehicle for testing the necessity of increases in computational power.

## Acknowledgements

This research was supported by a fellowship held by the first author from the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Québec and a research grant to the second author from the Natural Sciences and Engineering Research Council of Canada. We thank Scott Fahlman for providing the source code for cascade-correlation and Chris Schunn for providing source code for the drawing of Hinton diagrams. Finally, we would like to thank the three anonymous reviewers and especially Gary Cottrell for helpful comments. Address correspondence to Denis Mareschal, Centre for Brain and Cognitive Development, Department of Psychology, Birkbeck College, University of London, Malet Street, London WC1E 7HX, UK.

## Notes

1. Earlier work included five no-change patterns in the training set (i.e. ones in which the presented series was complete). Indeed, the results reported in Shultz *et al.* (1995) refer to that set of simulations. However, it was found in the simulations reported here that the five no-change patterns made no appreciable difference to network behaviour. We report simulations without the no-change patterns because we feel that their exclusion reduces the number of *ad hoc* constraints on the training set.
2. In a sample with an even number of measurements the median value falls between two measurements. When those values differ, the median is taken as the mid-point between the two values (Ferguson & Takane, 1989). In this case, the sample size is 20 and the median falls between the values one and two.
3. A preliminary version of this simulation was presented in Mareschal and Shultz (1993).
4. Only the 1324 order is inconsistent with correct ordering, and that occurs in only a few networks in one condition. All other orders that occurred in any condition were consistent with the predicted order; that is, all stages that occurred did so in the proper order. Stage skipping does not imply that stages are incorrectly ordered (Flavell, 1971; Shultz, 1991).

## References

Anzai, Y. (1987) Doing, understanding, and learning in problem solving. In D. Klahr, P. Langley & R. Neches (Eds), *Production System Models of Learning and Development*. Cambridge, MA: MIT Press.

Bandura, A. (1986) *Social Foundations of Thought and Action: A Social Cognitive Theory*. Englewood Cliffs, NJ: Prentice Hall.

Bates, E.A. & Elman, J.L. (1993) Connectionism and the study of change. In M.H. Johnson (Ed.), *Brain Development and Cognition*. Oxford: Blackwell.

Baum, B.E. (1989) A proposal for more powerful learning algorithms. *Neural Computation*. **1**, 201–207.

Baylor, G.W., Gascon, J., Lemoyne, G. & Pothier, N. (1973) An information-processing model of some seriation tasks. *Canadian Psychologist*, **14**, 167–196.

Bergan, J.R. & Jeska, P. (1980) An examination of prerequisite relations, positive transfer among learning tasks, and variations in instruction for a seriation hierarchy. *Contemporary Education Psychology*, **5**, 203–215.

Boden, M.A. (1982) Is equilibration important? A view from artificial intelligence. *British Journal of Psychology*, **73**, 65–173.

Buckingham, D. & Shultz, T.R. (1994) A connectionist model of the development of velocity, time,

and distance concepts. *Proceedings of the 16th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Buckingham, D. & Shultz, T.R. (1996) Computational power and realistic cognitive development. In G.W. Cottrell (Ed.), *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, pp. 507–511. Hillsdale, NJ: Erlbaum.

Bullock, M. & Gelman, R. (1977) Numerical reasoning in young children: the ordering principle. *Child Development*, **48**, 427–434.

Case, R. (1985) *Intellectual Development: Birth to Adulthood.* New York: Academic Press.

Changeux, J. & Dehaene, S. (1989) Neural models of cognitive functions. *Cognition*, **33**, 63–109.

Chi, M.T.H. (1978) Knowledge structures and memory. In R.S. Siegler (Ed.), *Children's Thinking: What Develops?* Hillsdale, NJ: Erlbaum.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2**, 304–314.

Dawson, G. & Fischer, K.W. (Eds) (1994) *Human Behaviour and the Developing Brain.* New York: Guilford.

Dodd, N. (1992) The importance of structure in neural networks. In K. Warwick, G.W. Irwin & K.J. Hunt (Eds), *Neural Networks for Control and Systems.* IEE Control Engineering Series, Vol. 46. London: Peregrinus.

Donaldson, M. (1986) *Children's Minds.* London: Fontana Press.

Dudai, Y. (1989) *The Neurobiology of Memory: Concepts Findings and Trends.* Oxford: Oxford University Press.

Elkind, D. (1964) Discrimination, seriation, and numeration of size and dimensional differences in young children: Piaget replication study VI. *Journal of Genetic Psychology*, **104**, 276–296.

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

Elman, J.L., Bates, E.A., Karmiloff-Smith, A., Johnson, M.H., Parisi, D. & Plunkett, K. (1996) *Rethinking Innateness: Connectionism in a Developmental Framework.* Cambridge, MA: MIT Press.

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

Fahlman, S.E. & Lebiere, C. (1990) The cascade-correlation learning architecture. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2.* Los Altos, CA: Morgan Kaufmann.

Feldman, J.A. & Ballard, D.H. (1982) Connectionist models and their properties. *Cognitive Science*, **6**, 205–254.

Ferguson, G.A. & Takane, Y. (1989) *Statistical Analysis in Psychology and Education.* Toronto: McGraw-Hill.

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

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

Frey, L. (1964) Sériation et transitivité. *Cahiers de Psychologie*, **7**,143–157.

Gilliéron, C. (1976) Décalage et sériation. *Archives de Psychologie*, **44**, Monographie 3.

Gilliéron, C. (1977) Serial order and vicariant order: the limits of isomorphism. *Archives de Psychologie*, **45**, 183–204.

Greenfield, P.M., Nelson, K. & Saltzman, E. (1972) The development of rulebound strategies for manipulating seriated cups: a parallel between action and grammar. *Cognitive Psychology*, **3**, 291–310.

Hoehfeld, M. & Fahlman, S.E. (1992) Learning with limited numerical precision using the cascade-correlation algorithm. *IEEE Transactions on Neural Networks*, **3**, 602–611.

Inhelder, B. & Piaget, J. (1969) *The Early Growth of Logic in the Child.* New York: Norton Library.

Jacobs, R.A., Jordan, M.I. & Barto, A.G. (1991) Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks. *Cognitive Science*, **15**, 219–250.

Karmiloff-Smith, A. (1992) *Beyond Modularity: A Developmental Perspective on Cognitive Science.* Cambridge, MA: MIT Press.

Keil, F.C. (1990) Constraints on constraints: surveying the epigenetic landscape. *Cognitive Science*, **14**, 135–168.

Kingma, J. (1982) A criterion problem: the use of different operationalizations in seriation research. *Perceptual and Motor Skills*, **55**, 1303–1316.

Kingma, J. (1983a) Some behavioural characteristics of the partial seriators reconsidered. *Journal of General Psychology*, **108**, 231–247.

Kingma, J. (1983b) The development of seriation, conservation, and multiple classification: a longitudinal study. *Genetic Psychology Monographs*, **108**, 43–67.

Kingma, J. (1984) The influence of task variations in seriation research: adding irrelevant cues to the stimulus materials. *Journal of Genetic Psychology*, **144,** 241–253.

Kingma, J. & Roelinga, U. (1983) Task sensitivity and the sequence of development in seriation, ordinal correspondence, and cardination. *Genetic Psychology Monographs*, **110,** 181–205.

Koslowski, B. (1980) Quantitative and qualitative changes in the development of seriation. *Merrill-Palmer Quarterly*, **26,** 391–405.

Lautrey, J., Bideaud, J. & Pierre-Puységure, M. (1986) Aspects génétiques et differentiels du fonctionnement cognitif lors de tâches de seriation. *L'Année Psychologique*, **86,** 489–526.

Liben, L.S. (1975) Evidence for developmental differences in spontaneous seriation and its implications for past research on long-term memory improvement. *Developmental Psychology*, **11,** 121–125.

Mareschal, D. (1992) *A Connectionist Model of the Development of Children's Seriation Abilities*. Master's Thesis, Department of Psychology, McGill University, Montreal.

Mareschal, D. & French, R.M. (1997) A connectionist account of interference effects in early infant memory and categorization. In M.G. Shafto & P. Langley (Eds), *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pp. 484–489. Mahwah, NJ: LEA.

Mareschal, D., Plunkett, K. & Harris, P. (1995) Developing object permanence: a connectionist model. *Proceedings of the 17th Annual Conference of the Cognitive Science Society*. New York: Erlbaum.

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*. Hillsdale, NJ: Lawrence Erlbaum.

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

Mareschal, D. & Shultz, T.R. (submitted) *Decomposing the Stick Seriation Task*.

McClelland, J.L. (1989) Parallel distributed processing: implications for cognition and development. In R.G.M. Morris (Ed.), *Parallel Distributed Processing: Implications for Psychology and Neurobiology*. Oxford: Oxford University Press.

McClelland, J.L. (1995) A connectionist perspective on knowledge and development. In T.J. Simon & G.S. Halford (Eds), *Developing Cognitive Competence: New Approaches to Process Modelling*. Hillsdale, NJ: Erlbaum.

Moody, J.E. (1992) The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In J.E. Moody, S.J. Hanson & R.P. Lipman (Eds), *Advances in Neural Information Processing Systems 4*, pp. 847–854. San Mateo, CA: Morgan Kaufmann.

Moore, G.W. (1979) Transitive inferences within seriation assessed by explanations, judgments, and strategies. *Child Development*, **50,** 1164–1172.

Nguyen-Xuan, A. (1976) Suite aux automate de sériation de Frey. *Cahiers de Psychologie*, **19,** 101–108.

Oden, G.C. (1987) Concept, knowledge, and thought. *Annual Review of Psychology*, **38,** 203–227.

Papert, S. (1963) Intelligence chez l'enfant et chez le robot. In L. Apostel, J. Grize, S. Papert & J. Piaget. La filiation des structures. *Etudes D'Epistemologie Genetique*, **15,** 131–194.

Piaget, J. (1965) *The Child's Conception of Number*. New York: Norton.

Piaget, J. & Inhelder, B. (1973) *Memory and Intelligence*. London: Routledge & Kegan Paul.

Pierre-Puységure, M., Bideaud, J. & Legarff, M. (1988) Etude de la sériation des longeurs, des poids et des hauteurs de son chez des enfants de six ans. *Archives de Psychologie*, **56,** 41–57.

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

Quartz, S.R. (1993) Neural networks, nativism, and the plausibility of constructivism. *Cognition*, **48,** 223–242.

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

Quinn, P.C. & Johnson, M.H. (1997) The emergence of perceptual category representations in young infants. *Journal of Experimental Child Psychology*, **66,** 236–263.

Retschitzki, J. (1978) L'évolution des procédures de sériation: étude génétique et simulation. *Archives de Psychologie*, **46,** Monographie 5.

Rogoff, B. (1990) *Apprenticeship in Thinking*. Oxford, Oxford University Press.

Rumelhart, D.E. & McClelland, J.L. (1986) On learning the past tenses of English verbs. In D.E. Rumelhart & J.L. McClelland (Eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 2. Cambridge, MA: MIT Press.

Schmidt, W.C. & Shultz, T.R. (1991) *A Replication and Extension of McClelland's Balance Scale Model*. Technical Report No. 91-10-18, McGill Papers in Cognitive Science, McGill University, Montreal.

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 8th International Workshop*. San Mateo, CA: Morgan Kaufmann.

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

Shultz, T.R., Buckingham, D. & Oshima-Takane, Y. (1994a) A connectionist model of the learning of personal pronouns in English. In S.J. Hanson, M. Kearns, T. Petsche & R.L. Rivest (Eds), *Computational Learning Theory and Natural Learning Systems*, Vol. 2. Cambridge, MA: MIT Press.

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

Shultz, T.R. & Schmidt, W.C. (1991) A cascade-correlation model of balance scale phenomena. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

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

Siegel, L.S. (1972) Development of the concept of seriation. *Developmental Psychology*, **6**, 135–137.

Squire, L. (1987) *Memory and Brain*. Oxford: Oxford University Press.

Terrace, H.S. & McGonigle, B. (1994) Memory and representation of serial order by children, monkeys, and pigeons. *Current Directions in Psychological Science*, **3**, 180–185.

Timmons, S.A. & Smothergill, D.W. (1975) Perceptual training of height and brightness seriation in kindergarten children. *Child Development*, **46**, 1030–1034.

Trabasso, T. (1977) The role of memory as a system in making transitive inferences. In R.U. Kail & J.W. Hagen (Eds), *Perspectives on the Development of Memory and Cognition*. Hillsdale, NJ: Erlbaum.

Ungerleider, L.G. & Mishkin, M. (1982) Two cortical visual systems. In D.J. Ingle, M.A. Goodale & R.J.W. Mansfield (Eds), *Analysis of Visual Behaviour*, pp. 549–586. Cambridge, MA: MIT Press.

van der Maas, H.L.J. & Molenaar, P.C.M. (1992) Stagewise cognitive development: an application of catastrophe theory. *Psychological Review*, **99**, 395–417.

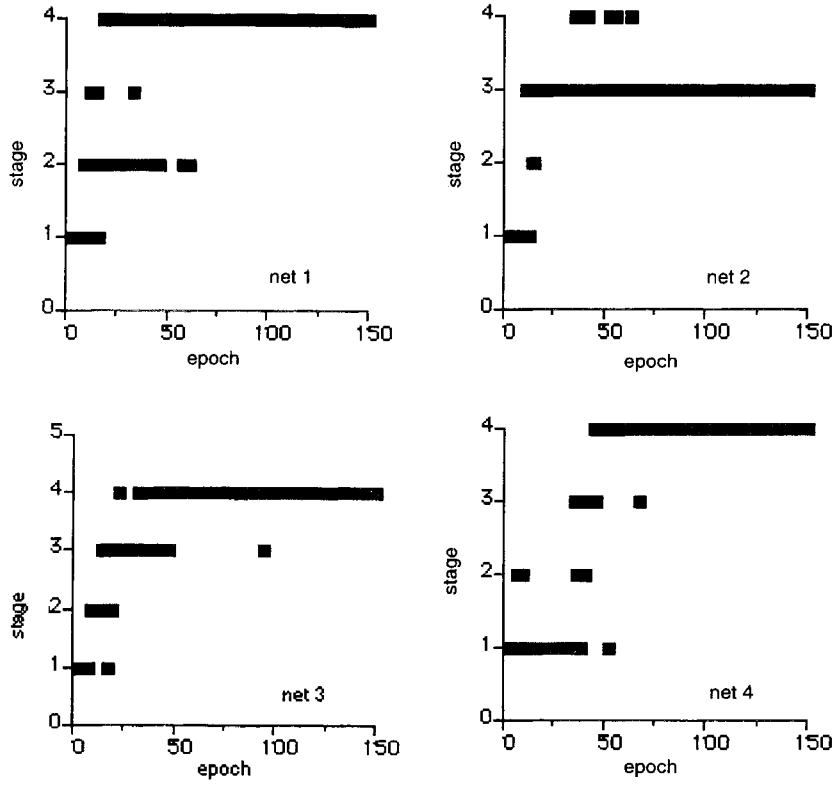
van Geert, P. (1991) A dynamic systems model of cognitive and language growth. *Psychological Review*, **98**, 3–53.

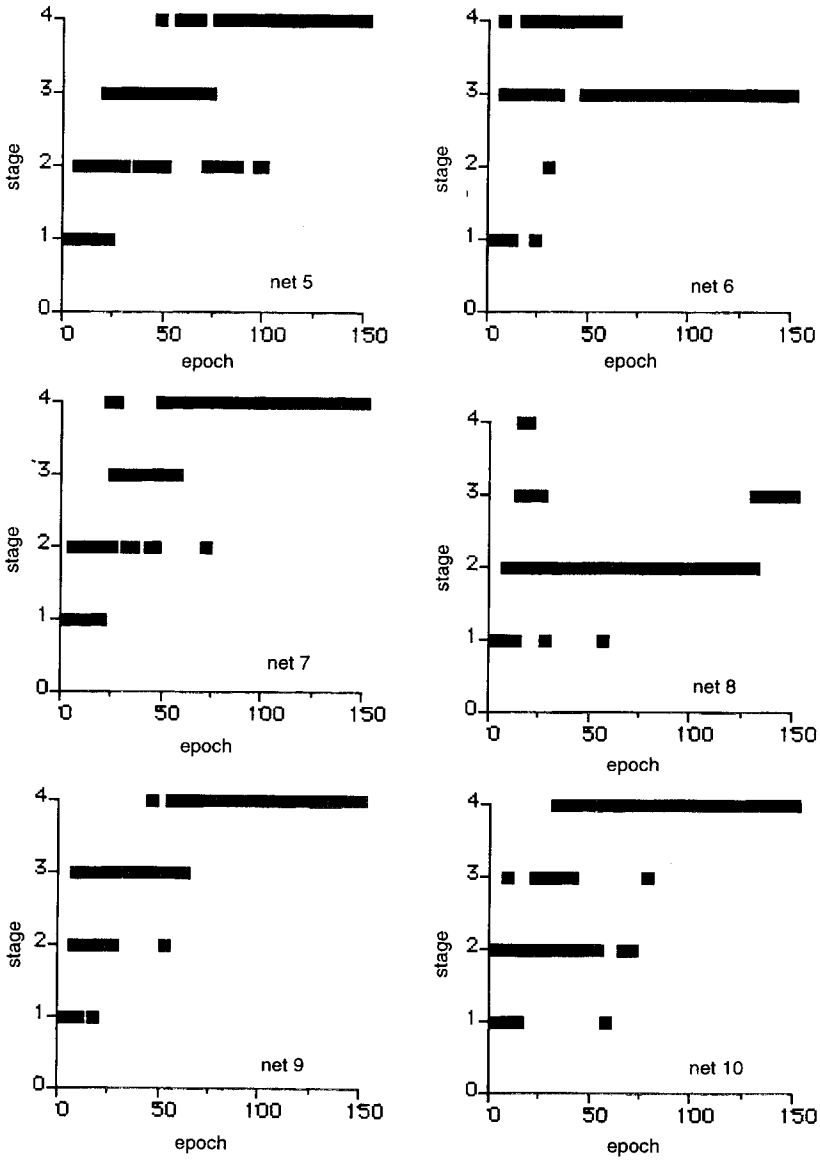
Vygotsky, L.S. (1978) *Mind in Society*. Cambridge, MA: Harvard University Press.

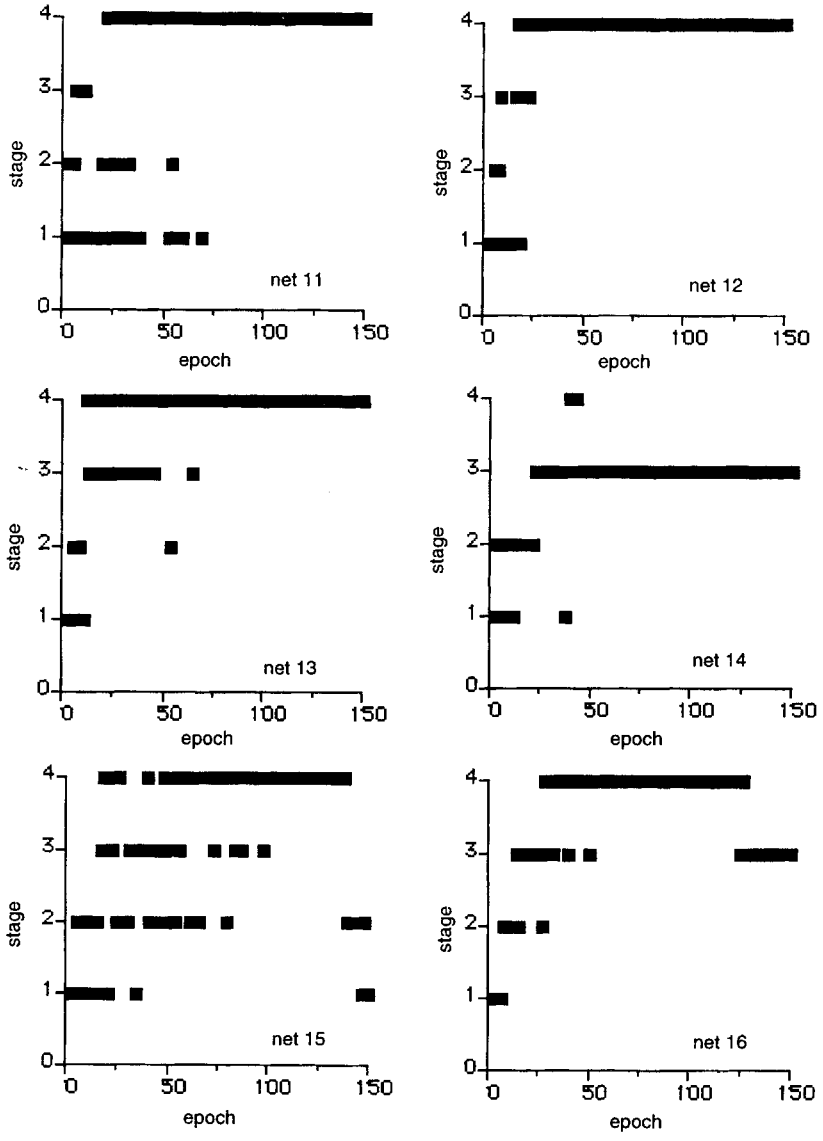
Young, R. (1976) *Seriation by Children: An Artificial Intelligence Analysis of a Piagetian Task*. Basel: Birkhauser.

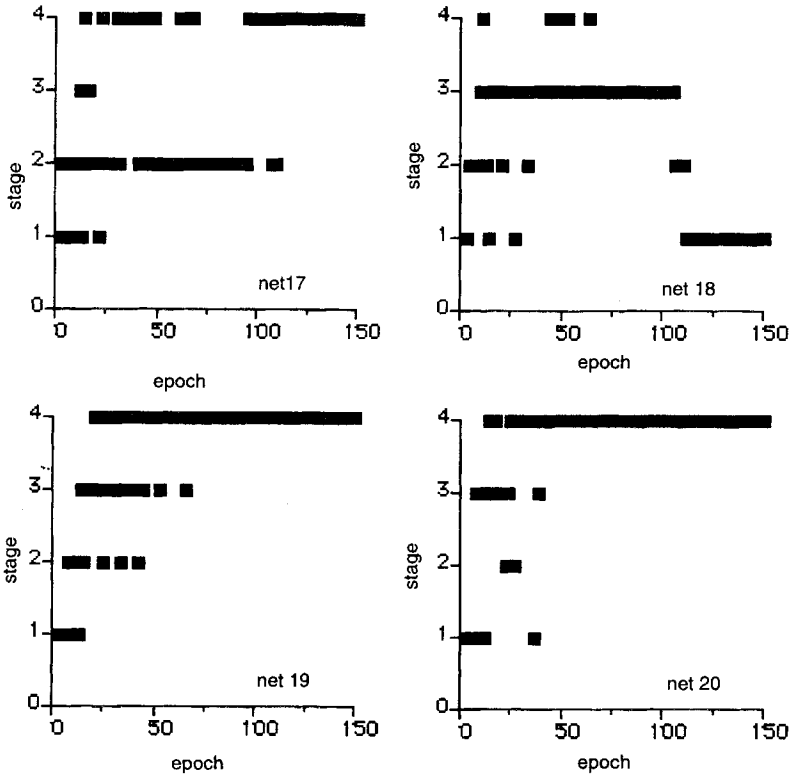
## Appendix

This appendix shows the stage development of all 20 networks in the internal subset condition. All networks were trained on 100 randomly selected six-element arrays and 20 randomly selected three-element arrays. The introduction of three-element arrays into the training set induces periods of consistent stage 3 performance.


stage
epoch
net 1
net 2
net 3
net 4
0
50
100
150
1
2
3
4
5


stage
epoch
net 5
net 6
net 7
net 8
net 9
net 10
0
1
2
3
4
50
100
150


net 11
net 12
net 13
net 14
net 15
net 16
stage
epoch
0
1
2
3
4
50
100
150


stage
epoch
net17
net 18
net 19
net 20
0
1
2
3
4
0
50
100
150