Computational Modeling of Variability in the Conservation Task

Fiona M. Richardson (f.richardson@bbk.ac.uk) **Neil Forrester (n.forrester@bbk.ac.uk)**

Frank D. Baughman (f.baughman@psychology.bbk.ac.uk) **Michael S.C. Thomas (m.thomas@bbk.ac.uk)**

Developmental Neurocognition Laboratory,
School of Psychology, Birkbeck College,
University of London, WC1E 7HX UK

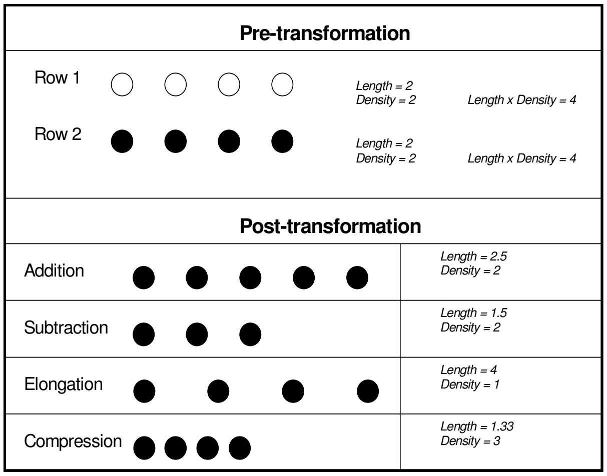
Abstract

The investigation of variability in reasoning tasks can provide valuable insights into key issues in the study of cognitive development. These include mechanisms that underlie developmental transitions, individual differences and developmental disorders. We explored potential sources of variability in the development of knowledge of *conservation* – a classic Piagetian task. Taking the task structure and problem encoding of Shultz (1998) as the normative case, we examined the computational parameters, problem encodings, and training environments that contribute to variability in development, both across groups and within individual cases.

Introduction

Conservation refers to the understanding or belief in the continued equivalence of two physical sets, following a transformation that appears to alter one and not the other. A given transformation may alter a quantity, by *adding* or *subtracting*, or preserve it through *elongation* or *compression.* The acquisition of conservation knowledge involves learning to distinguish between transformations that preserve and those that alter quantity. For example, in a typical number conservation task, as shown in Figure 1, a child is initially presented with two rows of counters (*pre-transformation*). The child is then asked whether these rows have the same number of counters or whether one has more than the other. A transformation is then applied to one row, and the child is asked again whether the two rows are the same, or whether one now has more counters than the other (*post-transformation*).

Piaget (1965) found that young children below 6-7 years are non-conservers, in that when presented with a transformation that preserves number (such as *elongation* or *compression*) they answer that one row has more counters than the other. In contrast children older than 6-7 years are conservers, having learnt that transformations of this type do not alter number. This finding has been corroborated across a range of conservation tasks, such as mass (using modeling clay), liquid quantity (using beakers), and number (using counters) (Brainerd & Brainerd, 1972; Halford & Boyle, 1985; Klah, 1984; Miller & Heldmeyer, 1975; Siegler, 1995; Siegler & Robinson, 1982; Wallach, Wall & Anderson, 1967; Winer, 1974). The rich literature on conservation has also established a series of biases that occur as young children learn to conserve, relating to problem size, length, and mode of presentation. These effects are summarized in Figure 1.



Acquisition	The shift from non-conservation to conservation (which is typically abrupt).
Problem-size	Knowledge of conservation develops on tasks with smaller numbers of items, prior to those with a larger number of items.
Length-bias	For problems with differing row lengths, non-conservers tend to choose the longer row as having more items.
Screening	Younger children conserve better on problems that are screened, where they cannot see the the result of the transformation.

Figure 1: The number conservation task using counters

A range of classic Piagetian tasks such as conservation, seriation and the balance scale, have been subject to computational investigation. Models have sought to specify the mechanisms that generate the behavioral profile of development (Mareschal & Shultz, 1999; McClelland, 1989, 1995; Shultz, 1998; Schultz, Mareschal & Schmidt, 1994). Recent connectionist implementations use an algorithm called cascade-correlation (Falham & Lebiere, 1990). During training, network connections are altered but if learning stagnates, the size of the hidden layer is increased. The success of this generative connectionist approach has been attributed to the change in the network architecture (Mareschal & Shultz, 1999; Shultz, 1998; Schultz, Mareschal & Schmidt, 1994). Thus Shultz (1998) ascribes the ability of his model to capture the abrupt shift from non-conservation (NC) to conservation (C) to the addition of hidden units and an attendant increase in representational power. However, it is possible that other computational parameters have a similar impact upon a model's behavioral profile over the course of development. The influence of diverse learning parameters on development and their relation to cognitive variability is a question under active exploration (Richardson, Baughman, Forrester & Thomas, 2006).

The study of variability is important for three reasons. First, it permits us to explore the conditions under which certain behavioral transitions in development may or may not occur. Second, variability across individuals of the same age gives a window onto general or specific intelligence. Third, variations in development from the normal pathway are found in disorders, sometimes exhibiting delay, failure to reach more complex levels of reasoning, or qualitatively atypical patterns. Implemented models have generally focused on the normative (average) pathway, yet each type of variability must ultimately be explained at a mechanistic level (Thomas & Karmiloff-Smith, 2003).

In the following sections, we report an initial series of simulations that investigated potential sources of variability in the conservation task. First we introduce our normal model of development based on Shultz (1998). Second, we explore how manipulating the model's computational parameters, input encoding, and training environment alter its developmental behavioral profile. Third, we examine within-individual variability by carrying out a case study comparison, contrasting two individual model runs.

The Normal Model

The normal model was defined as a 3-layer feedforward connectionist network consisting of an input layer of 13 units, a hidden layer of 4 units, and an output layer of 2 units. The problem encoding used by this network was based on Shultz (1998) and is shown in Figure 2. Each row of counters was represented over 2 units, encoding row length (ranging from 2 to 6.33) and density (ranging from 2 to 6) respectively, as real numbers. Both rows are shown represented in their pre- and post-transformation states. The row transformed (either row 1 or row 2) was indicated by the activation (-1 or +1) of a single unit. The transformation type was encoded arbitrarily over 4 units, with the activation of a single unit indicating the type as follows: *addition* (1 -1 -1 -1), *subtraction* (-1 1 -1 -1), *elongation* (-1 -1 1 -1), or *compression* (-1 -1 -1 1). The three possible response options were encoded over 2 binary output units as follows: (i) row 1 longer (1 0), (ii) row 2 longer (0 1), (iii) both rows equal (0 0). We differed from Shultz in using a more standard feedforward architecture with a sigmoid rather than hyper-tangent activation function.

Our model was trained using back-propagation for 1500 epochs, with a learning rate of 0.025. Ten network runs were conducted per manipulation, with initial weights randomized between ±0.5. The standard error across runs is depicted in all figures. The composition of the training and test sets was again based on that of Shultz, with patterns having five levels of row length and five levels of density. A total of 400 training patterns were selected from a full set of 600 possible conservation problems (based upon 25 initial rows, 3 possible start states, and 4 possible transformations for each of the 2 rows). Performance was assessed using 100 novel test patterns at 5, 25, 50, 100, and 200 epochs, and then at every subsequent 100-epoch interval until the end of training at 1500 epochs.

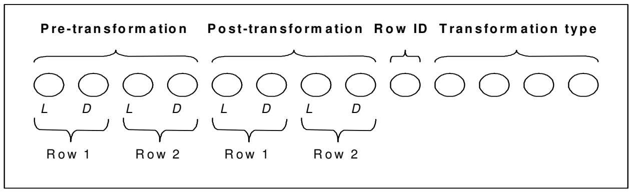


Figure 2: The input encoding

In order to assess the behavior of the model, the test set was used in conjunction with 4 metrics, each reflecting a target behavioral phenomenon described in Figure 1: (i) *Acquisition*, (ii) the *Problem Size Effect,* (iii) *Length Bias Effect,* and the (iv) *Screening Effect.* Metric 1 plotted the development of knowledge of conservation, and calculated the percentage of test patterns correct. Metric 2 calculated the proportion of small vs. large problem types correct. In this case, the test set consisted of 40 patterns, 20 small problem types (<12 items), and 20 large (>24 items). Metric 3 used elongation and compression problems from the test set (a total of 18 patterns, 8 and 10 of each type respectively) to calculate the proportion of patterns where the longer row was selected as having more items than the shorter row. Metric 4 calculated the proportion of unscreened vs. screened problems correct for the complete test set. Test patterns presented to the network were represented as "screened" by replacing post-transformation activation values with zeros.

The normal network learned the training set to an accuracy of 99.5% (SE 0.4%). Training performance exhibited an early shift from NC=>C between 100 and 200 epochs (from 44.58 to 70.35% training patterns correct). This shift was preceded by an initial decline in training performance over the first 50 epochs and followed by small incremental improvements in performance as training progressed. The behavioral profile of the model can be seen in Figure 3, where the shift from NC=>C (*Acquisition*) on novel patterns occurs between 100-200 epochs and performance leaps from 36.2% (SE 1.75%) to 61.7% (SE 4.75%). Normality is defined as the non-linear shift to conserving. The model also exhibited a minor performance advantage for small problem sizes (*problem size effect*) between 100-700 epochs, the time during which the model was doing the bulk of its learning. Normality is defined as an advantage for small problems (+ve values on the chart) during earlier phases of training. The model's bias for selecting longer rows as having more items (*length bias effect*) was also found to reduce after this point in learning. Normality is defined as an early positive spike on the length bias chart, which shows proportional advantage of long problems over short. Unlike Shultz

(1998), our model did not show any preference for "screened" problems early in learning (*screening-effect*), which would appear as an early negative spike on the chart. This shortcoming may relate to our use of sigmoid processing units.

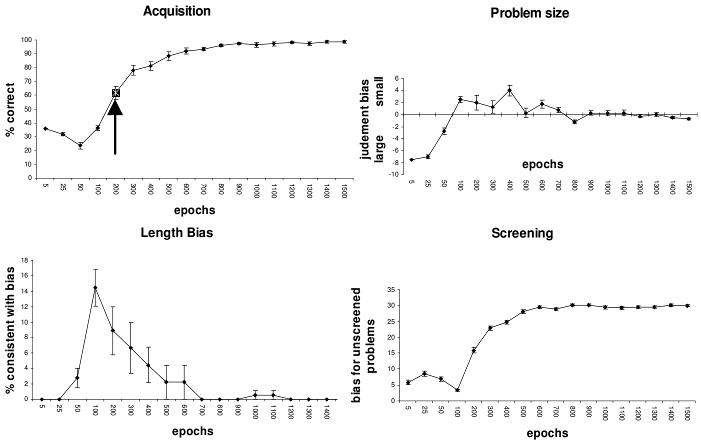


Figure 3: Developmental phases of the normal model. The arrow shows shift from NC=>C for the *acquisition* metric

Exploring Variability

With our base model in hand, we then sought to assess the influence of several factors on development. Variability was explored by systematic changes to (1) the base model's computational parameters, (2) its problem encoding, or (3) the training environment.

Variability and Computational Parameters

The computational parameters that were varied included: (i) the number of hidden layers, (ii) the number of hidden units in a single layer, (iii) the learning rate, and (iv) the slope of the sigmoid transfer function for hidden layer units.

Increasing the number of hidden layers

The performance of the model was tested over learning with 2 and 3 hidden layers (HL), with 4 units per layer. Additional hidden layers tend to increase the computational complexity of the mappings that can be learned by a network while slowing down learning, since the error signal must filter back through more levels. Learning rate (*lr*) was held constant (at 0.025) in this condition (this was the case for all subsequent architectures unless stated otherwise). These networks achieved mean accuracy levels on the training set of 99.5, 99.7, and 92.9% (SE 0.4, 0.02, and 6.64%), respectively. The developmental trajectories of the networks are shown in Figure 4. The profiles of networks with 1HL and 2HL were very similar. Both 1HL and 2HL networks showed a shift from NC => C between 100-300 epochs, which was slightly larger for networks with 2HL than those with 1HL (25.5 and 39.2% respectively). Networks with 3HL showed an incremental improvement in performance with no obvious shift, attaining knowledge of conservation at 700 epochs. There was a sustained negative bias for *problem size* in networks with 3HL, as well as an increase in variability. The variability for the *length bias effect* was very high, particularly for 2HL and 3HL networks. As for *screening*, there was no bias in early learning for screened problems. However, the developing bias for "unscreened" problems increased over learning.

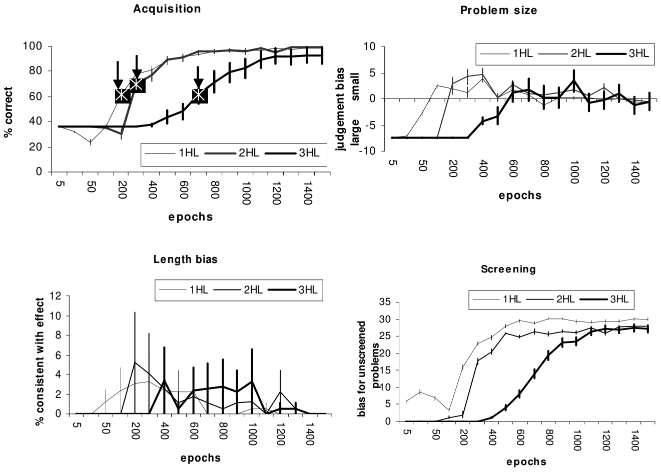


Figure 4: Profile for models with 1 (normal), 2 and 3 HL. Arrows show shifts from NC=>C

Increasing the number of hidden units in a single layer

Adding extra units to a given hidden layer allows a network to learn more patterns of a given complexity, and to solve a given problem with smaller weight values, thereby requiring less training. We assessed networks with 4, 10, and 20 units in the hidden layer (HU) for the normal 1HL model. At the end of training networks with 4HU had a mean accuracy of 99.48%; 10HU and 20HU networks had reached 100%. 10HU and 20HU networks showed earlier *acquisition* of conservation knowledge (between 50 and 100 epochs). This shift was also larger than networks with 4HU (30-30.3% in comparison to 25.5%). The behavioral profile across metrics can be seen in Figure 5. All networks showed a similar profile across testing metrics, with variability being uniformly low. Interestingly, networks with 4HU did show a slightly larger *length bias effect* of an extended duration, in comparison to 10HU and 20HU networks. It is likely that this is related to the initial learning of the 4HU network being lower than that of 10HU and 20 HU networks. Therefore, increasing the number of hidden units improved training performance, resulting in an earlier shift for those networks with more hidden units, but showed a similar trajectory in comparison to the normal case. Extending this manipulation to 2HL and 3HL networks yielded the same results. Thus, expanding the capacity of the system in terms of parallel processing resources alters the onset of learning, but not the overall developmental profile. In conjunction with the findings from the hidden layer condition, this result suggests that the structure of any additional processing resources can have a marked impact upon developmental process.

Reducing the learning rate

The term *delay* is sometimes used to describe individual differences, as well as the trajectories of developmental disorders. An obvious means of slowing learning would be to decrease the learning rate. Though this method would not provide an explanation of why different cognitive abilities are often differentially delayed in disorders it does nevertheless allow us to explore how learning rate affects the transitions the system exhibits during learning. Learning rate was reduced in the normal network in four decrements from 0.025 to 0.02, 0.015, 0.01, and 0.005. After 1500 epochs, these networks achieved mean accuracies 99.8, 98.5, 96.6, and 86.3% respectively. Figure 6 depicts their developmental phases, with the four steps labeled from LR4 to LR1 as the learning rate decreases.

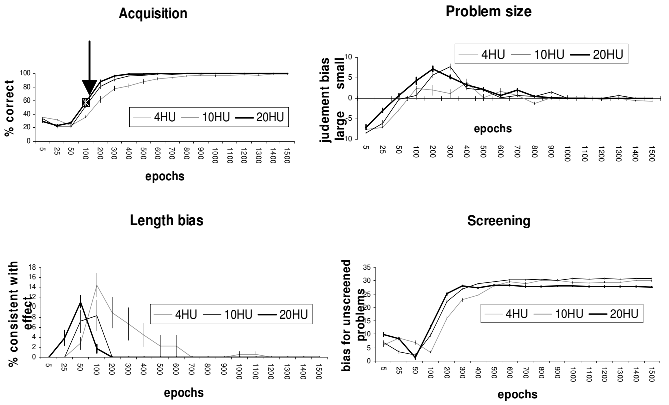


Figure 5: Profile for models with 4 (normal), 10 and 20 HU in a single layer. Arrow shows shift from NC=>C

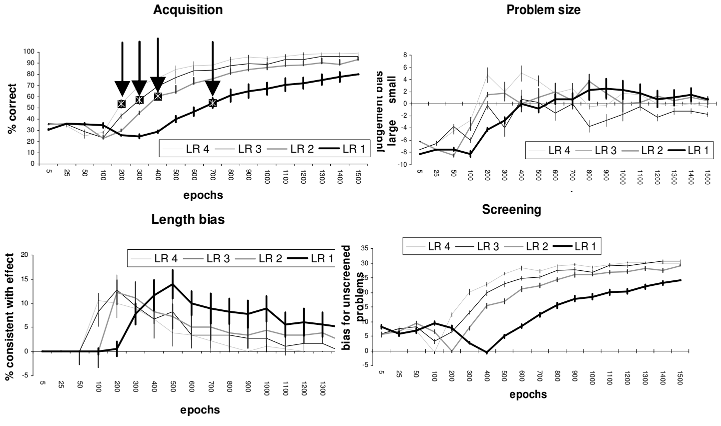


Figure 6: The 1HL model with reducing learning rates. Arrows show shift from NC=>C

Predictably learning rates slowed development down. As a result, improvements in performance behavior were more incremental. Extending this manipulation to networks with 2HL and 3HL, displayed a similar pattern of results. Though networks with a lower learning rate had a lower level of performance at end of training (at 1500 epochs), the overall performance was high, but could have improved further through extended training time. In contrast, for developmental disorders, performance typically asymptotes at a less complex level in comparison to the normal case. In terms of individual differences, it is also doubtful whether everyone eventually 'catches up' to a fixed final cognitive level. From this perspective, a reduced learning rate does not seem a good (sole) candidate to explain the type of developmental delay found in disorders.

Decreasing the sigmoid slope

Changing the slope of a transfer function has the effect of altering the type of category distinctions a model can make. For example, a steep sigmoid slope results in sharp category boundaries and is good for tasks where the model is required to make rule-like distinctions. Whereas a shallow slope is better suited to fine-grained distinctions and tasks with broad category boundaries. Altering the level of processing unit discriminability has been shown to produce patterns of deficits consistent with those seen in developmental disorders (Thomas & Karmiloff-Smith, 2003). This condition explores the impact of changing the general properties of processing resources of hidden units, through decreasing the slope of the sigmoid transfer function for the entire hidden layer. The slope of the sigmoid was reduced (from a value of 1) in the normal model, by four levels of decreasing discriminability as follows: 0.8, 0.6, and 0.25, to 0.125.

The profiles across metrics are shown in Figure 7, where sigmoid slope is labeled as four steps from S4 to S1 as the sigmoid slope decreases. Changing the slope of the sigmoid produced a profile that at the surface level appears similar to that found for the learning rate condition, with development slowing down as the slope decreases. However, in contrast, with the exception of the *problem size* metric, there appears to be more convergence across the different slope levels in the later stages of learning. This difference on this metric appears to be for the shallowest two slope decrements, and may be related to the later and more incremental trajectory shown during task *acquisition.* Overall, this result shows how two different parameters may produce a similar developmental trajectory, but also subtle differences, as in the persisting problem size bias for the shallowest slope.

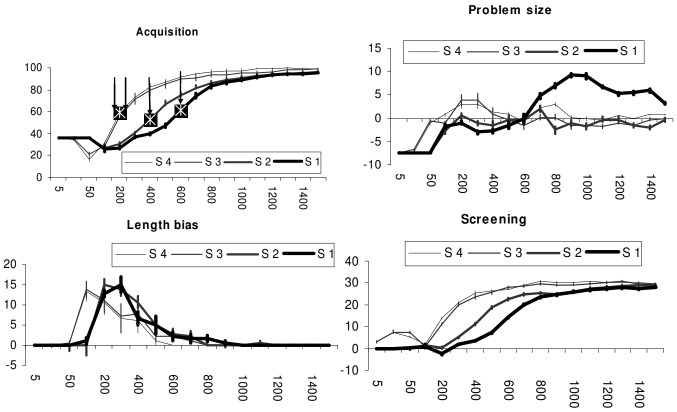


Figure 7: The 1HL model with declining sigmoid slope

Variability and the Problem Encoding

We explored a variation in problem encoding where the salience of transition type was increased. The number of units encoding transition information (as shown in Figure 2) was doubled from 4 to 8, resulting in an input layer consisting of 17 units, with 8 units encoding pre- and post-transformation information, and 8 units encoding transformation type. This manipulation was carried out for networks with 1HL, 2HL and 3HL. The final performance of the models was found to be similar to that shown for equivalent models trained without increased transition information. The overall profile of development and *Acquisition* of conservation knowledge was also the same as the equivalent models. Therefore, for these simulations, changing the salience of a dimension of information did not have any notable impact upon the developmental trajectory of the model.

Variability and the Engaged Environment

Since development in the conservation task corresponds to the child's active exploration of the domain, we refer to the training set as the *engaged environment*. We created a training set with a limited coverage of the problem space. It consisted of 400 problems with a small quantity of items only (<12 items). The normal architecture and problem encoding was used. Networks with 1HL, 2HL and 3HL were trained on this environment to explore any interaction between representational power and the engaged environment. Interestingly, this environment did not appear to have notable impact upon the overall performance, irrespective of the number of hidden layers in the model. At the end of training 1HL, 2HL and 3HL networks reached the mean accuracies of 99.75, 99.78, and 91.53%, respectively. The profile of 1HL and 2HL networks over metrics was similar to that shown for equivalent models trained on a normal engaged environment. Limiting the engaged environment to problems with a small number of items did not impact upon the developmental trajectory of the model.

Individual Variability: A Case Comparison

Variability also occurs during the development of individual children. The risk of averaging across individuals is that the resulting trajectory may not actually be found in any one, and this possibility also exists for simulation data. In this section we conduct an in-depth comparison of two individual cases: (i) a single normal model with 1HL (henceforth *normal case*), and (ii) a 1HL model with a reduced learning rate (*lr*=0.005, henceforth *lr case*). Both models were trained using the normal input encoding and engaged environment using the same randomly initialized starting weights. The behavioral profile of each model was assessed using our 4 metrics. In addition, a detailed analysis of the development of conservation according to (i) transformation type, and (ii) problem size was conducted for test items. The training performance of both models can be seen in Figure 8, where the *lr case* shows a slower developing, more incremental trajectory, in comparison to the *normal case*. The shift from NC=>C is also clearly later (by approximately 500 epochs) than that of the *normal case*, and subsequent improvements in training performance are also smaller. This pattern in training performance can also be seen in the behavioral profile for metric *acquisition* (calculated on novel test items) shown in Figure 9. For *problem size* and *length bias* metrics, the *lr case* shows an extended *problem size* and *length bias* effect. These effects are in parallel with the protracted learning window of this model. For the *screening* metric, the trajectory of the *lr case* deviates from that of the *normal case*, showing a minor preference for "screened" problems at the onset of acquisition of conservation knowledge.

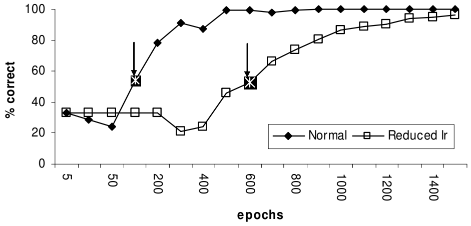


Figure 8: Training performance for the normal and reduced learning rate models. Arrows show the shift from NC=>C

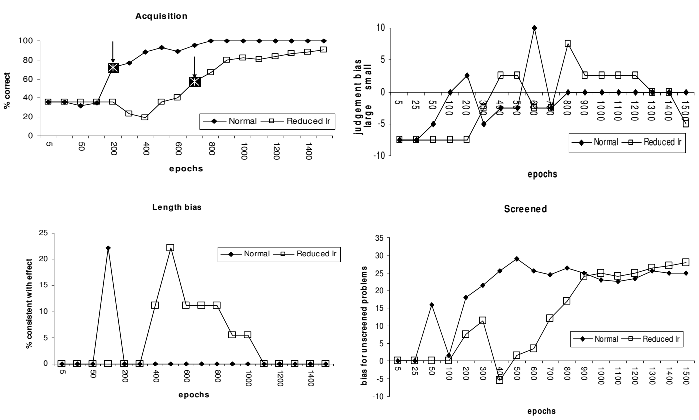


Figure 9: Profile for the normal and reduced learning rate models. Arrows show the shift from NC=>C

Exploring the development of conservation knowledge in the *normal case* across problem types (as shown in Figure 10) revealed a difference in the initial profile for problems that alter number (*addition* and *subtraction*), in comparison to those that preserve number (*elongation* and *compression*). *Addition* and *subtraction* problems showed a static level of performance early in learning, whereas *elongation* and *compression* problems showed an initial dip in performance. As a consequence, performance over learning on transformations that preserve number was poorer than those that alter it. This dip was seen on all problem types in the *lr case*.

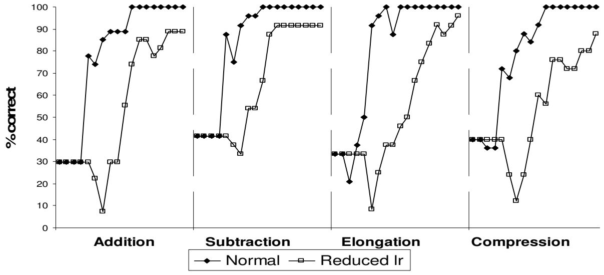


Figure 10: Profile of performance across problem types during learning for reduced *lr* and *normal* cases

An initial dip in performance can also be seen for problems of differing sizes (as shown in Figure 11). In the *normal case*, this dip was exaggerated for large problem sizes, resulting in poorer performance on large problems during learning. For the *lr case*, the converse pattern is seen, where the performance for larger problem types is better.

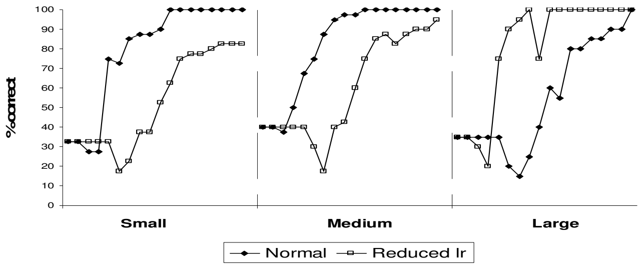


Figure 11: Profile of performance across problem types during learning for reduced *lr* and *normal* cases

Discussion

These simulations fall within a wider program of considering the effects of computational parameters on cognitive and language development. The exploration of mechanisms underlying variability in cognitive development may enhance our understanding of the origins of individual differences and developmental disorders, as well as transitions in the normal development of individual children. In this case simulations of the conservation task indicated that changes to the internal computational parameters of the model had a marked impact upon the acquisition of conservation knowledge. Notably, changes to the internal discriminability of processing units through reducing the slope of the sigmoid transfer function, as well as decreasing the learning rate delayed acquisition of conservation knowledge. The profile of performance from these two manipulations illustrates how different parameters can have a similar impact upon the trajectory of development. By contrast, changes to the problem encoding at input or the engaged environment had little impact on the model's developmental trajectory. These results contrast with a similar series of computational simulations of variability on the balance scale task, where changes to the model's engaged environment produced marked alterations in the developmental profile (Richardson et al, 2006). In tandem, these results paint a picture where the effect of alterations to the constraints that shape development depends on the nature of the cognitive task. The same parameter may not exert a uniform influence across cognitive domains.

References

Brainerd, C.J. & Brainerd, S.H. (1972). Order of acquisition of number and quantity in conservation. *Child Development, 43(4),* 1401-1406.

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

Halford, G.S., & Boyle, F.M. (1985). Do young children understand conservation of number? *Child Development, 56 (1),* 165-176.

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

Klahr, D. (1984). Transition processes in quantitative development. In R.J. Sternberg (Ed.), *Mechanisms of cognitive development* (p. 101-139). New York: Freeman.

Mareschal, D., & Shultz, T.R. (1999). Development of children's seriation: A connectionist approach. *Connection Science, 11(2),* 149-186.

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

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

Miller, P.H. & Heldmeyer, K.H. (1975). Perceptual information in conservation: Effects of screening. *Child Development, 46,* 588-592.

Piaget, J. (1965). *The child's conception of number.* New York: Norton.

Richardson, F.M., Baughman, F.D. Forrester, N., & Thomas, M.S.C. (2006). Computational modeling of variability in the balance scale task. *Proceedings of the 7th International Conference on Cognitive Modeling* (pp. 256-261). Trieste, Italy: Edizioni Goliardiche.

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

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

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

Siegler, R.S. (1995). How does change occur: A microgenetic study of number conservation. *Cognitive Psychology, 28*, 225-273.

Siegler, R. S., & Robinson, M. (1982). The development of numerical understandings. *Advances in Child Development and Behavior, 16*, 241-312.

Thomas, M. S. C., & Karmiloff-Smith, A. (2003). Connectionist models of development, developmental disorders and individual differences. In R. J. Sternberg, J. Lautrey, & T. Lubart (Eds.), *Models of intelligence: International perspectives*, (p. 133-150). APA.

Wallach, L., & Wall, J., & Anderson, L. (1967). Number conservation: The roles of reversibility, addition-subtraction, and misleading perceptual cues. *Child Development, 38(2),* 425-442.

Winer, G.A. (1974). Conservation of different quantities among preschool children. *Child Development, 45(3),* 839-842.

Acknowledgements: This research was supported by UK MRC CE Grant G0300188 to Michael Thomas