Title
A Neural Model of Number Comparison with Robust Generalization

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 45(45)

Authors
Shultz, Thomas
Nobandegani, Ardavan S.
Wang, Zilong

Publication Date
2023

Peer reviewed

A Neural Model of Number Comparison with Robust Generalization

Thomas R. Shultz[12], Ardavan S. Nobandegani[14], & Zilong Wang[3]
thomas.shultz@mcgill.ca, ardavan.salehinobandegani@mail.mcgill.ca
[1]Department of Psychology, McGill University
[2]School of Computer Science, McGill University
[3]Integrated Program in Neuroscience, McGill University
[4]Mila – Quebec AI Institute

Abstract

We propose and implement a relatively simple computational neural-network model of number comparison. Training on paired comparisons of the integers 1-9 enables the model to efficiently and accurately simulate some fundamental empirical phenomena (distance and ratio effects on accuracy and response time). It also generalizes robustly to more advanced tasks involving multidigit integers, negative numbers, and decimal numbers. The work demonstrates that small neural networks can sometimes efficiently learn a powerful system that exhibits extremely robust generalization to untrained items. Some important alternate models of number comparison are considered to establish a broader context. Several predictions and suggestions are made for future empirical and computational research in this area.

Keywords: number comparison; neural networks; generalization.

Introduction

We all occasionally engage in comparing two numbers to each other to determine which is smaller or larger. Such comparisons are common, seemingly automatic, and roughly accurate. Examples include an investor estimating returns on two different investments, a shopper looking for a good deal, or a young child selecting a lesser or greater collection of similar items such as candies or toys.

Early psychological research quickly established that such number comparisons are not done by consulting a memory lookup table. That solution would be inefficient because of the number of possible numerical comparisons being potentially infinite, and the inability of table lookup to account for subtle characteristics of the numbers being compared. For example, a sample of Stanford University students asked to quickly and accurately select the larger of two single-digit integers in the range 1-9 took longer to decide and made slightly more errors when the difference between the two integers was small than when it was large (Moyer & Landauer, 1967). The authors noted that such phenomena also occurred when people were asked to compare physical quantities such as length, weight, or loudness, suggesting that magnitudes had to be estimated and compared before a decision could be made.

Subsequent experiments found that the ratios between the two integers created similar phenomena for both reaction time and errors (e.g., Adriano, Girelli, & Rinaldi, 2021). When the smaller/larger ratio of the two integers was large (e.g., 8/9 = .889), reaction time and errors increased as opposed to comparisons involving smaller ratios (e.g., 2/3 = .667). Summarizing these empirical findings, integer comparisons are quicker and more accurate when the relevant magnitudes are easier to distinguish from each other.

Eventually, various computational models were formulated to help explain these and other related psychological phenomena. Most of these models utilized artificial neural networks. There are currently at least 15 associated empirical phenomena to explain, and no model that simulates and explains all of these phenomena, although one model comes close (Huber, Nuerk, Willmes, & Moeller, 2016).

In this relatively short paper, there is insufficient space to summarize the many empirical and computational modeling studies on number comparison. We focus instead on devising and implementing a relatively simple computational model that does two things. It first simulates some of the fundamental empirical phenomena in the number comparison literature, namely the effects of distance and ratios between the numbers being compared. Second, our model explores the issue of how well learning the relative magnitudes of the 1 to 9 digits generalize to more complex number-comparison skills, involving multi-digit, negative, and decimal numbers.

Method

To simulate learning how to compare numbers, we use an artificial neural network algorithm called cascade-correlation (Fahlman & Lebiere, 1990) that has been used effectively in many simulations of cognitive development (Dandurand & Shultz, 2014; Shultz, 2017). During learning, connection weights are adjusted so that overall network error is reduced:

$$E = \sum_o \sum_p \left(A_{op} - T_{op}\right)^2 \quad \text{(Equation 1)}$$

where E is sum-of-squared error, A is the actual output activation for unit o and pattern p, and T is the target output activation for this unit and pattern.

Cascade-correlation training starts with a two-layer network (i.e., only the input and the output layer), and then recruits hidden units one at a time, if needed, to solve the problem being learned. Number comparison turns out to be sufficiently simple that there is no need to recruit any hidden units when the 1-9 integers are being compared. Our cascade-correlation networks use an asymmetric sigmoid activation function for the output units:

$$y_i = 1/(1 + e^{-x_i}) \quad \text{(Equation 2)}$$

where y is the receiving unit i's output, x_i is the net input to unit i, and e is the exponential function. Thus, output unit activations range from a floor of 0 to a ceiling of 1.

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society.*

A drawing of our number comparison network (dubbed NCN) is presented in Figure 1. As is common in neural networks, input values are specified numerically. For example, to describe a comparison of the numbers 2 and 3, activity of the left input unit is 2 and that of the right input unit is 3. The activation signal sent to the output units is the sum of products of each sending unit activation and its connection weight, passed through the activation function specified in Equation 2. Output target values are 1 for larger and 0 for smaller. A bias unit is typical in neural networks, functioning like the intercept added into a linear equation. The bias unit has a constant input of 1 and learnable output connection weights. This allows shifting the activation function in binary comparisons to the left or right, which may be important for successful learning.

A problem in feed-forward neural networks is an inability to simulate variation in reaction time because activation passes through the network in constant time. To simulate reaction times, we add a module that elaborates the network's final decision by cycling the network output to decision units in a gradual fashion, while recording the number of cycles required for a decision to satisfy a specified constraint. In our case, the two decision weights, fixed at 0.1, convey the magnitude information to corresponding decision units that are initialized to 0. When the absolute difference between the decision units reaches a value of .9, cycling stops and the number of cycles reached is reported as an index of reaction time. The more decisively different the magnitude estimates are, the less time it takes to satisfy this criterion. Reaction time requires two or more outputs.

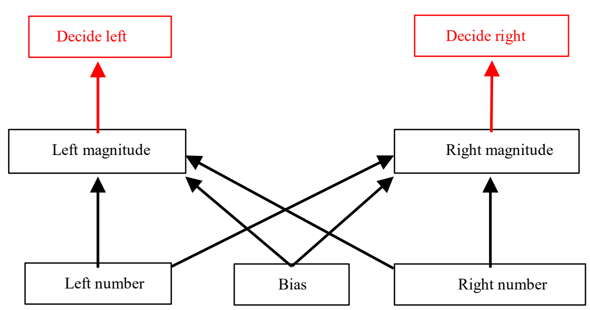

Figure 1: Diagram of our NCN model. Magnitude outputs of a cascade-correlation learning network (colored black) are fed to a decision module (colored red) which eventually generates an explicit decision and reports the number of cycles required to reach that decision. The seven units in the cascade-correlation network (black rectangles) are variously active and send their activity signals over connection weights (black arrows), exciting or inhibiting activity in receiving units. The six connection weights in each CC network are learned in the service of reducing sum-of-squared-error in the magnitude estimations, while the two connection weights in the decision module (red arrows) are each fixed at 0.1, allowing a gradual buildup of decision strength.

Results

Simulation results focus on two important empirical effects (difference and ratio) and eventual generalization to later developing skills (with multidigit numbers, negative integers, and decimal numbers). We run 20 number comparison networks in each simulation, allowing for statistical analysis of the results. Network performance varies as each network is randomly initialized with small connection weights in the learning module.

Difference Effect

Difference and ratio effects are evident in measures of both error and reaction time. Figure 2 shows that error decreases as a function of the absolute difference between each pair of integers being compared, reflected in a large main effect in a repeated-measures ANOVA, $F(7, 133) = 191$, $p < .00001$, $\eta_p^2 = .91$. Tukey comparisons between adjacent means are statistically significant at differences from 1-6, but not from 6-8, where error is virtually 0.

In this simulation, networks learn in a mean of 9.65 epochs, with an SD of 4.20. Each epoch presents each of the 72 paired-integer comparisons once, along with the correct feedback about which is larger. At the end of each epoch, the 6 connection weights in the learning network are adjusted to lower the sum-of-squared error (Equation 1).

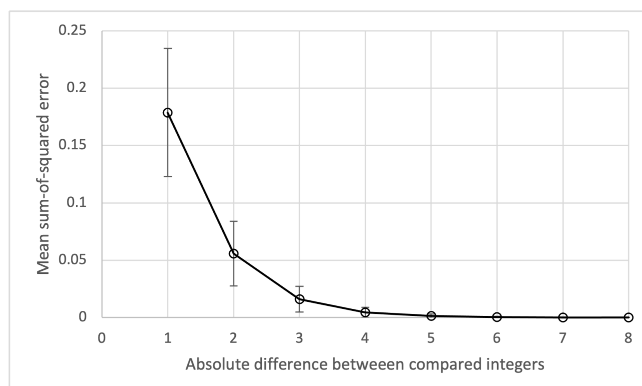

Figure 2: Mean error (with SD) as a function of difference between the two integers being compared.

Two examples of learning progress by individual networks are shown in Figure 3. The first network starts with no pairs correct and reaches complete correctness in 7 epochs. The second network starts with half of the pairs correct (by chance) and reaches fully correct performance in 9 epochs. Such up and down progress is typical in this learning. Because initial connection weights are initialized to random values, networks vary in their learning performance. Importantly, these two examples demonstrate that, without learning, a network has no knowledge of the relative magnitude of numerical digits.

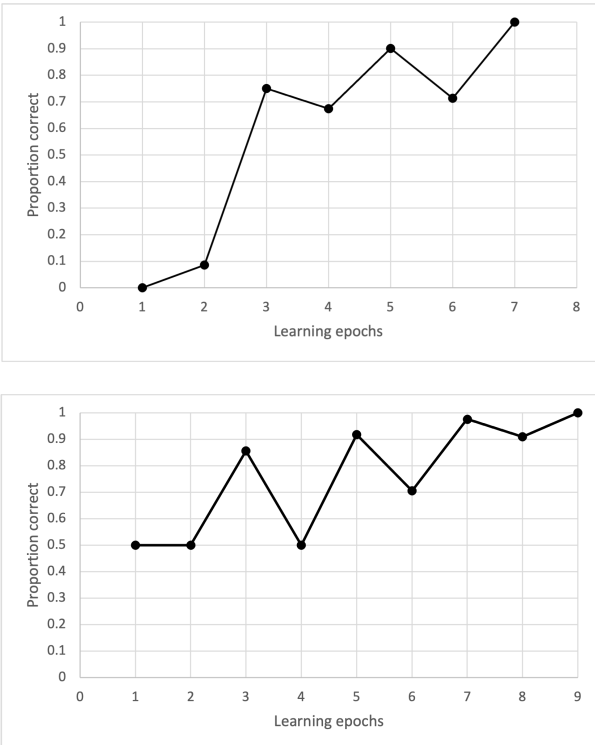

Figure 3: Learning progress in each of two representative networks across epochs.

In an analogous ANOVA of cycles to reach a decision, Figure 4 shows that mean cycles to reach a decision decrease as a function of the absolute difference between the integers being compared, with a large main effect of difference, $F(7, 133) = 158$, $p < .00001$, $\eta_p^2 = .89$. Tukey comparisons between adjacent means are statistically significant at differences from 1-7, but not from 7-8, where cycles approach a floor asymptote.

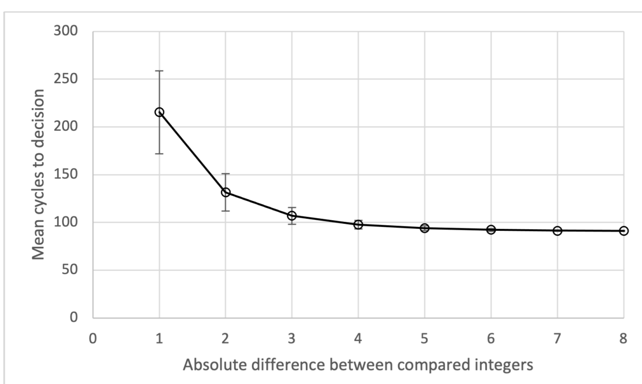

Figure 4: Mean cycles to reach a decision as a function of difference between the two integers being compared, in 20 neural networks.

In other work, we developed a mathematical model of NCN system to gain insights into how and why this neural network accomplishes accurate number comparison. An interesting prediction of the math model is that the positive and negative weights entering an output node are additive inverses of each other, ensuring that they sum to about 0. Other aspects of the learned connection weights vary randomly, as initial weight values are set randomly.

We test this additive inverse prediction by analyzing the sizes and signs of the learned connection weights. A typical example of this analysis is shown in Figure 5, which presents the final connection weights learned by a single representative network in the number-difference simulation. There, the weight pairs that become approximate additive inverses are:

- LN-LM (1.4) and RN-LM (-1.22). These are the two weights multiplying the left and right integers leading to the **left** magnitude output, which for this network are 1.4 and -1.22, respectively.
- RN-RM (.97) and LN-RM (-1). These are the two weights multiplying the left and right integers leading to the **right** magnitude output, which for this network are .97 and -1, respectively.

Direct weights (that favor left side or right side) become positive, while crossover weights (from one side to the other side) become negative, which helps to inhibit the opposite alternative. This connection-weight pattern is approximated by all the networks in all the present simulations, although each network solution is effectively unique because of different initial weight values.

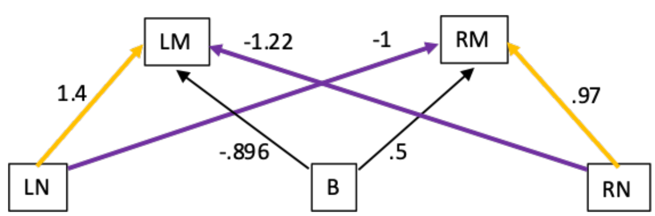

Figure 5: Connection weights in a representative single network in the difference-effect simulation. LN is left input number, B is bias unit, RN is right input number, LM is left magnitude output, and RM is right magnitude output. Positive input weights are colored gold, negative input weights are colored purple, and bias weights are colored black.

The additive-inverse prediction of the mathematical model is confirmed overall in Figure 6 which plots the value of the crossover weight as a function of the direct weight entering a magnitude output unit. Linear regressions explain nearly all the R^2 variance, as noted in Figure 6.

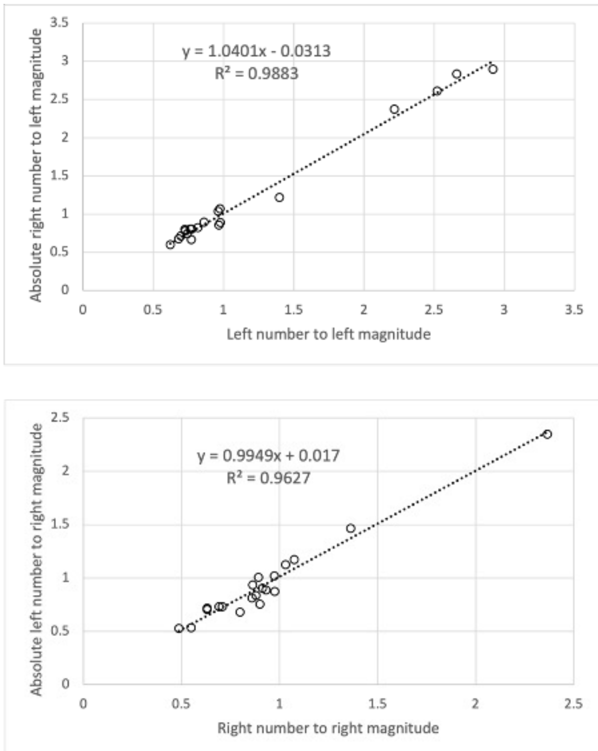

Figure 6: Crossover weights as a function of direct weights for each of 20 networks in the number-difference simulation. Each circle represents a pair of weights entering the left or right magnitude output unit. The fact that these weights cling to the estimated line shows that the two weights are nearly additively inverse, summing to approximately 0 and confirming the math model prediction.

Ratio Effect

A second simulation addresses the ratio effect, the idea that increasing ratios of the smaller to larger integer would increase difficulty in the form of both increases in error and slower reaction time. Figure 7 presents the impact of these ratios on error. A repeated-measures ANOVA of error yields a large main effect of ratio, $F(26, 494) = 92$, $p < .00001$, $\eta_p^2 = .83$. Sum of squared error increases with the ratio of the smaller to the larger integer being compared.

Similarly, an analogous ANOVA produces a large main effect of ratio on mean cycles to make a decision, $F(26, 494) = 83$, $p < .00001$, $\eta_p^2 = .81$. As shown in Figure 8, it takes longer to reach a decision as the ratio of smaller to larger integer increases.

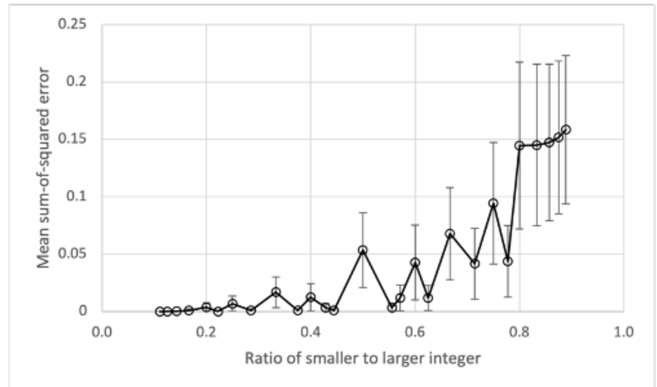

Figure 7: Mean error as a function of the ratio of the smaller to larger integer being compared.

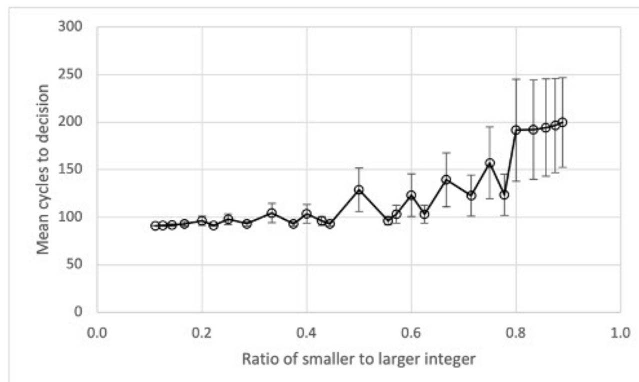

Figure 8: Mean cycles to reach a decision as a function of integer ratio.

Although it was noted at the dawn of number comparison research that distance and ratio measures could be related (Moyer & Landauer, 1967), we precisely quantify this relationship by computing a Pearson correlation between absolute differences and smaller/larger ratios for the integer pairs 1-9, $r(34) = -.843$, $p < .00001$. There are 72 of these unequal pairs, but only 36 if the direction of the differences is correctly ignored. This strong relationship implies that difference and ratio effects could be confounded in some studies of the 1-9 integer pairs, although the distance to a perfect correlation (.157) leaves room for the two effects to be somewhat independent in this integer range.

Generalization to Advanced Number Types

Three additional simulations examine whether training only on pairs of the 9 single digits would generalize to novel combinations of two, three, and four digit numbers. The answer is an emphatic *yes*. Each of these simulations uses 20 randomly drawn pairs of 2-, 3-, or 4-digit numbers. As shown in Table 1, training on the 1-9 single digits generalizes strongly to untrained novel comparisons of 2-, 3-, and 4-digit integers.

In the last row of Table 1 are the potential numbers of digit pairs to which this level of generalization applies. This potential number is computed with the standard equation for permutations of n numbers taken r at time, where order matters: $nPr = n!/(n-r)!$

Although the number comparison model could generalize to untrained integers of more than 4 digits, we do not have access to a computer that could generate the potential number of patterns at 5 digits. Suffice it to say that NCN generalizes robustly from learning the 72 pairs of the first 9 integers.

Table 1: Mean proportion of correct generalizations out of 20 randomly selected example pairs, with SD and number of possible permutations of those digits.

	2 digits	3 digits	4 digits
Integer range	10-99	100-999	1000-9999
Mean correct	.97	.96	.93
SD correct	.04	.02	.08
Potential number of pairs	8,010	809,100	80,911,000

Two further simulations show similarly strong generalization from learning the 1-9 pairs to untrained negative integers and 2-digit decimal numbers (Table 2).

Table 2: Mean proportion of correct generalizations to 72 test pairs of negative numbers (-1 to -9) and 72 test pairs of decimal numbers (1.5 to 9.5), each increasing in steps of 1.

Indicator	Negatives	Decimals
Mean correct	.90	.9993
SD correct	.07	.003

Discussion

Our NCN system simulates a range of number comparison phenomena: distance and ratio effects and strong generalization to multidigit integers and other advanced number types (negative numbers and decimal numbers). It is essential to cover distance and ratio effects, showing that the underlying computation is not a memory-based lookup table and supporting a more abstract principle that accuracy and reaction time both improve with numbers that are easier to distinguish, whether by differences or ratios.

As far as we know, this is the first computational model to explore generalization across number types. Previous computational models of number comparison invariably trained and tested on a single type of number, whether 2-digit integers, negative numbers, or decimal numbers. The NCN model uniquely generalizes robustly from training on only the 1-9 digits to multidigit numbers, negative numbers, and decimal numbers, with success rates in the mid .90s.

We do not claim children automatically generalize that well to these advanced types because they obviously do not. But we do believe that training on the single digit numbers establishes a strong foundation for eventually moving on to these advanced number types. To generalize well, individuals would also have to learn the technical vocabularies for auditory presentation of number pairs, e.g., millions, billions, trillions, quadrillions, etc., as well as appropriate scientific notation for written input number presentations.

It is interesting that the strong generalization ability of the NCN model provides a notable exception to the claim that artificial neural networks require immense amounts of training and still do not generalize nearly as well as humans do (Lake, Ullman, Tenenbaum, & Gershman, 2017).

An important reason for such strong NCN generalization here is that this small CC artificial neural network learns the basics of a powerful number system, not merely a large collection of unrelated numerical facts. This powerful system, variously named Arabic, Hindu-Arabic, or Western-Arabic, first emerged in the tenth century BC (1000 BC – 901 BC) (Kunitzsch, 2003; Plofker, 2009). It is a base 10 system, encompassing not only the digits 1-9 but also 0. Importantly, it enables progression to more advanced numeric forms such as multidigit numbers, negative numbers, and decimal numbers.

Despite harnessing all this potential numeric power, the NCN system is simpler than other computational models of number comparison. For example, one of the leading number comparison models employs at least six networks and many parameters and connection weights, and requires up to 100,000 epochs of training (Huber et al., 2016). This higher degree of complexity is designed to deal with fifteen other identified empirical phenomena in number comparison.

An alternative input option used in number comparison experiments is that of dot patterns or physical objects, presenting a more perceptual task. A popular and effective neural-network technique for coding such perceptual stimuli is *thermometer* coding, in which a number is represented by a set of activated units corresponding to its numerosity (Zorzi & Butterworth, 1999; Zorzi & Testolin, 2018). For example, 6 would be represented by activating units 1-6 from left to right, and 16 by activating units 1-16, again from left to right. Although thermometer coding may appear to be a natural way to represent a collection of dots or objects, it seems implausible that a biological number comparison system would be equipped with precisely the correct number of input units in anticipation of processing a virtually infinite range of number pairs to be compared. We are currently working on a different approach to dot pattern or object inputs using convolutional deep-learning networks that learn to map perceived object collections onto numbers, avoiding such implausible evolutionary engineering.

Many of the coding techniques used to model number comparison in neural networks share the assumption that larger numbers create higher activation levels than do smaller numbers (e.g., Huber et al., 2016; Zorzi & Testolin, 2018). This makes sense and works well, particularly by relying on the use of sigmoidal activation functions that effectively convert all numbers to a standard range, most often 0 to 1. This allows a network to benefit from important characteristics of the Arabic number system, including proper subsets, a number line, and compression effects. Smaller numbers are proper subsets of larger numbers. The number line orders the numbers from left to right and seems to compress into smaller steps on the right where numbers get large. As our simulations demonstrate, these two characteristics are either preserved (proper subset) or constructed (number line, compression) in the NCN model. Any parts of a number line can be constructed by ordered NCN results in a particular number region. Larger small/large ratios can make it seem that far right regions are more compressed than are far left regions. Although a number line and its perceived compressions are often viewed as primary causes of psychological phenomena, NCN shows how those constructions could naturally arise during learning and generalization; they do not need to be assumed.

The NCN model also provides a relatively simple example of how rules could be implemented in neural systems. Here, a simple feed-forward neural network learns and generalizes with such strong regularity that the outcomes are only distinguishable at the behavioral level by virtue of a few small errors. Further study of such simple problems might eventually provide insights into how more complex symbol systems could be implemented in neural systems. Here, a small NCN network generalizes so well from a bit of learning that it only very rarely makes mistakes on more complex, related problems. Symbolic systems could explain error-free performance but would likely have trouble accounting for very rare errors and difference and ratio effects, all of which are empirically well established and thus stand as worthy simulation targets for computational models.

Our findings with NCN networks demonstrate an important role for learning in the development of mathematical skills. The NCN model starts without any knowledge of integer comparisons and thus must learn them. This necessity of learning is consistent with a recent theoretical review supporting the view that mathematical knowledge is not innate but rather is learned (O'Shaughnessy, Gibson, & Piantadosi, 2022). Those authors examined six predictions of strong numerical nativism, finding that each is contradicted by evidence from anthropological research. Their extensive and detailed review notes a lack of number systems in some human groups and considerable variability in the form of numerical systems that do emerge, highlighting the importance of social and economic factors in constructing cognitive systems that satisfy culturally specific goals.

We hope that our NCN model would inspire further empirical research on number comparison in children, including young children. Our modeling was, in fact, partly inspired by a spontaneous game of number comparison with a grandson at 3 years, 3 months of age. He answered several questions comparing two-digit numbers that he had never encountered before, and he was invariably correct. He also detected our intentional mistakes on 2-digit comparisons that he spontaneously posed to us. More systematic number comparison studies should be done on children as young as three years and older as they move up to larger-digit comparisons and comparing advanced number forms such as negative and decimal numbers.

We expect that results of such studies would roughly conform to the first-appearance stages of several documented number-knowledge acquisitions (Siegler, 2022): single-digit integers at 3-5 years, 2-digit integers at 5-7 years, 3-digit integers at 7-12 years, and decimal and negative number forms at 11+ years. Interestingly, there is a representational shift from logarithmic to linear within each of these three age periods. Such a shift is consistent with our view that ratio effects and number-line compressions are emergent mental constructions in humans and neural networks rather than innately hard-coded features.

It is important to note that our NCN model is a neural network which is initially ignorant about the relative magnitudes of its numerical input. The network learns these magnitudes by training on pairs of the digits 1-9. Such training enables very strong, but not perfect, generalization to more advanced tasks involving multi-digit, negative, and decimal numbers. From its performance alone, a trained network looks *as if* it is operating at a symbolic level, apart from producing a very few errors. Such tiny error rates are in the neighborhood of those made by adult participants on single-digit pairs (Moyer & Landauer, 1967).

It has also been found that children's discovery of new arithmetic strategies proceeds with few or no flawed strategies (Siegler & Jenkins, 1989). When engaged with a rigorous number system, children's advances are strongly constrained by the system being learned, which is also true in the NCN model.

An advantage of the NCN model is that it naturally integrates with our model of the learning and use of probability distributions, Neural Probability Learner and Sampler (NPLS) (Shultz & Nobandegani, 2022). NPLS has simulated a series of empirical studies documenting that infants from about 6 months of age learn and use probability distributions to select actions that are more likely to provide desirable outcomes (Denison, Reed, & Xu, 2013; Denison & Xu, 2010, 2014; Xu & Garcia, 2008). Because magnitude is often confounded with probability, it is important to disentangle them in such experiments, by including conditions in which the two are in conflict, where higher probability of getting the desired object has fewer of those objects. Both processes are encompassed by the CC algorithm, with probability learning often recruiting hidden units. This integration of NCN and NPLS facilitates coherent simulations of empirical phenomena that involve the use of probabilistic and magnitude strategies across ages and conditions.

Acknowledgements

This research was supported in part by an operating grant to TRS from the Natural Sciences and Engineering Research Council of Canada.

References

Adriano, A., Girelli, L., & Rinaldi, L. (2021). The ratio effect in visual numerosity comparisons is preserved despite spatial frequency equalisation. *Vision Research*, *183*(January), 41–52.

Dandurand, F., & Shultz, T. R. (2014). A comprehensive model of development on the balance-scale task. *Cognitive Systems Research*, *31–32*, 1–25.

Denison, S., Reed, C., & Xu, F. (2013). The emergence of probabilistic reasoning in very young infants: evidence from 4.5- and 6-month-olds. *Developmental Psychology*, *49*(2), 243–249.

Denison, S., & Xu, F. (2010). Twelve- to 14-month-old infants can predict single-event probability with large set sizes. *Developmental Science*, *13*(5), 798–803.

Denison, S., & Xu, F. (2014). The origins of probabilistic inference in human infants. *Cognition*, *130*(3), 335–347.

Fahlman, S., & Lebiere, C. (1990). The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems*, 524–532.

Huber, S., Nuerk, H., Willmes, K., & Moeller, K. (2016). A general model framework for multisymbol number comparison. *Psychological Review*, *123*(6), 667–695.

Kunitzsch, P. (2003). The Transmission of Hindu-Arab Numerals Reconsidered. In J. Hogendijk & A. Sabra (Eds.), *The Enterprise of Science in Islam: New Perspectives* (pp. 3–22). Cambridge, MA: MIT Press.

Lake, B., Ullman, T., Tenenbaum, J., & Gershman, S. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*(e253), 1–72.

Moyer, R., & Landauer, T. (1967). Time required for judgments of numerical inequality. *Nature*, *215*, 1519–1520.

O'Shaughnessy, D., Gibson, E., & Piantadosi, S. (2022). The cultural origins of symbolic number. *Psychological Review*, *129*(6), 1442–1456.

Plofker, K. (2009). *Mathematics in India*. Princeton, NJ: Princeton University Press.

Shultz, T. (2017). Constructive artificial neural-network models for cognitive development. In N. Budwig, E. Turiel, & P. D. Zelazo (Eds.), *New Perspectives on Human Development* (pp. 13–26). Cambridge: Cambridge University Press.

Shultz, T., & Nobandegani, A. (2022). A Computational Model of Infant Learning and Reasoning With Probabilities. *Psychological Review*, *129*(6), 1281–1295.

Siegler, R. (2022). Development of numerical knowledge. In O. Houdé & G. Borst (Eds.), *The Cambridge Handbook of Cognitive Development* (pp. 361–382). Cambridge, UK: Cambridge University Press.

Siegler, R., & Jenkins, E. (1989). *How children discover new strategies*. Hillsdale, NJ: Erlbaum.

Xu, F., & Garcia, V. (2008). Intuitive statistics by 8-month-old infants. *Proceedings of the National Academy of Sciences*, *105*(13), 5012–5015.

Zorzi, M., & Butterworth, B. (1999). A Computational Model of Number Comparison. In M. Hahn & S. Stoness (Eds.), *Proceedings of the Twenty First Annual Conference of the Cognitive Science Society* (pp. 772–777). Vancouver, British Columbia: Cognitive Science Society.

Zorzi, M., & Testolin, A. (2018). An emergentist perspective on the origin of number sense. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*(1740).