Computational journey from numerical cognition to arithmetic ability

Denis Turcu[1] (dt2626@cumc.columbia.edu), Katharyn Fatehi[2]
[1]Neuroscience Department, [2]Engineering Undergraduate Program - Columbia University, USA

Abstract

Numerical cognition sets the foundation for developing arithmetic abilities, yet performing arithmetic operations is much more abstract, complex and subtle than identifying numbers. This poses the question whether animals that demonstrate numerical cognition are necessarily equipped to develop arithmetic abilities. Recent experimental studies explored this question in multiple species and found that these animals learn to add and subtract. Here, we used recurrent neural networks (RNNs) to investigate possible neural mechanisms underlying arithmetic abilities. We found that our models perform very well on in-distribution test data, but do not generalize well to out-of-distribution test data. The two main reasons for poor generalization are 1) models do not learn the basic underlying arithmetic operation, and 2) bounded activation function prohibits models to compute on arbitrarily large scales. Our work suggests that developing arithmetic abilities requires specific capacity for abstraction on top of learned or innate numerical cognition, consistent with previous cognitive studies. While most times lack of numerical cognition implies lack of arithmetic abilities, we point out that the inverse, i.e. having numerical cognition implies having arithmetic abilities, need not be true, and we demonstrate this result in an RNN model.

Keywords: decision making; numerical cognition; arithmetic ability; recurrent neural network

Introduction

Numerical cognition, i.e. the ability to identify numbers, sets the foundation for developing arithmetic abilities, i.e. adding two numbers of arbitrary magnitudes. Recent experimental studies (Howard, Avarguès-Weber, Garcia, Greentree, & Dyer, 2019; Schluessel, Kreuter, Gosemann, & Schmidt, 2022) found that multiple species demonstrating numerical cognition in previous works, e.g. honeybees (Chittka & Geiger, 1995; Dacke & Srinivasan, 2008; Gross et al., 2009; Skorupski, MaBouDi, Dona, & Chittka, 2018), cichlids (Agrillo, Petrazzini, & Bisazza, 2015; Mehlis, Thünken, Bakker, & Frommen, 2015) and stingrays (Daniel, Alvermann, Böök, & Schluessel, 2021; Kreuter, Christofzik, Niederbremer, Bollé, & Schluessel, 2021), can learn to add and subtract 1 in the range of numbers from 1 to 5. Motivated by the belief that performing arithmetic operations is much more abstract, complex and subtle than identifying numbers (Butterworth, 2005), here, we investigate whether animals that demonstrate numerical cognition are necessarily equipped to develop arithmetic abilities using RNN models.

Consistent with previous cognitive studies (Butterworth, 2005), our work suggests that developing arithmetic abilities requires specific capacity for abstraction, on top of learned or innate numerical cognition. We found that RNNs generalize well when they use the task's underlying arithmetic operation, whether they inferred it during training or they were specifically trained to use it. On the other hand, RNNs that did not infer and were not trained to use the underlying arithmetic operation could only perform well on in-distribution test data, but fail to generalize even to simple cases of out-of-distribution test data. This suggests that specialized ability for abstraction, i.e. broadly learning to add rather than learning training data statistics, is highly important for generalized performance. We also found that bounded activation functions prohibit RNNs to compute on arbitrarily large scales. Possible solutions to this problem include further specializations such as using compositionality, e.g. arithmetic carry method, or using logarithms.

Methods & Results

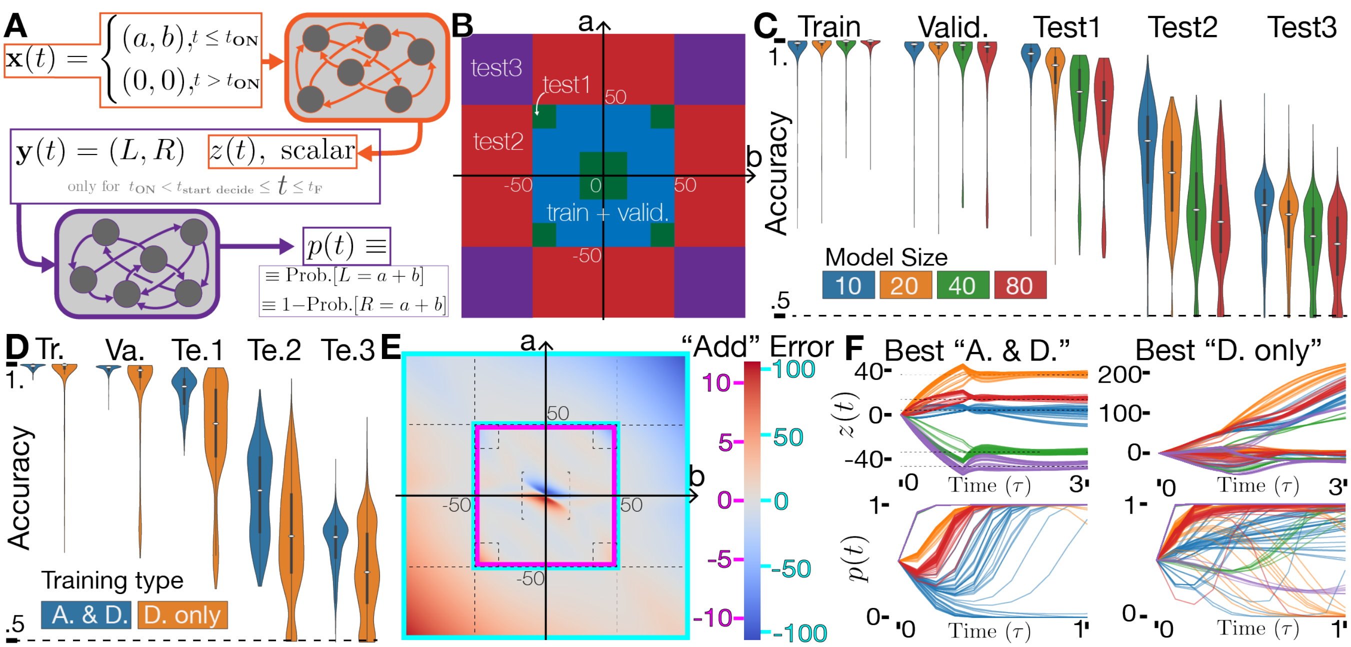
We replicated the task designed by Howard et al. (2019) (and later used Schluessel et al. (2022)) using an RNN model. The task consists of two stages. In the first stage, the subject inspects an input containing information about two numbers, referred to as a and b here. In the second stage, the subject losses direct access to the information presented in the first stage, and it inspects a new input that contains information about two numbers, L and R for the left and right choices respectively. The subject needs to make a choice between the two directions and is required to find which of L or R is equal to $a+b$. We used RNNs to mimic both the long-term memory necessary to acquire the arithmetic and decision rules and the short-term memory needed to solve individual trials.

Our model combines two RNNs to separate the two stages of the task, match the experimental design and allow interpretation of the underlying computation (Fig A). We endowed our model with numerical cognition, meaning we input the numbers a, b, L and R directly, since numerical cognition is our working assumption and previous work showed that various models can convert visual inputs to numbers (Yang, Ganichev, Wang, Shlens, & Sussillo, 2018). We trained RNNs of the form $\dot{\mathbf{r}}(t) = -\mathbf{r}(t) + \sigma\left(W^{\text{rec}}\mathbf{r}(t) + W^{\text{in}}\mathbf{i}(t) + \mathbf{b}\right)$ with readout $W^{\text{out}}\mathbf{r}(t)$ using backpropagation through time (BPTT), where $\mathbf{b}$ is a bias, $\sigma(x) = \tanh(Nx/200) \cdot 200/N$ and N is the number of recurrent units. We also trained models achieving similar performance (data not shown) with $\sigma(x) = \max(x, 0)$, but we focused on the modified $\tanh$ version for biological plausibility – firing rates in real neurons are bounded from both above (saturation) and below (no firing), so arbitrarily large inputs a and b cannot be represented in single neurons. We scaled the amplitude of the activation function inversely proportional to the network size such that we could maintain $W_j^{\text{out}} \sim O(1)$ while ensuring larger networks would not have an advantage over small ones. The constant "200" enables our networks to process numbers well outside of our datasets (Fig B). We split our datasets into multiple regions and tested our models on examples in which both a and b were seen during training, but not together (test1), only one of a or b was seen (test2), or neither a nor b was seen (test3). Our model learns the task and performs well on validation data not seen during training, but performance decreases significantly on test data and with task difficulty (Fig C). We also note that larger networks perform worse on test data, suggesting overfitting and low dimensionality of the task, which we discuss below.

Two different training methods address whether explicit arithmetic instruction is important for the task. We used BPTT to minimize $\lambda \sum_{t>t_{\mathrm{ON}}}^{t_{\mathrm{F}}} (z(t) - (a+b))^2 + \sum_{t \geq t_{\mathrm{s.d.}}}^{t_{\mathrm{F}}} \mathrm{BCE}(p(t), \mathrm{target})$ in the "Add & Decide" version, or just the second term of the sum in the "Decide only" version, where $\mathrm{BCE}(\cdot,\cdot)$ is binary cross entropy. We found that explicit addition training improved performance (Fig D). Nevertheless, even the best "A.&D."–trained model could not accurately carry out addition on the test datasets, not even on the small numbers range (Fig E). While $z(t)$ of the "A.&D."–trained models looks as expected, interpretable and robust on the training dataset, we found that the best "D. only"–trained model employs a different mechanisms to solve the task (Fig F). We further explored the computational mechanisms using dimensionality reduction, both latent circuit inference (Langdon & Engel, 2022) and PCA. We found that networks as small as 3 units for each RNN could solve the task similarly to the other models we trained, however they would not generalize. We found that linear RNNs ($\sigma(x) = x$) learn addition and generalize very well, but rely on arbitrarily large firing rates, an unrealistic feature for biological neurons.

Discussion

Our work demonstrates the importance of inductive bias, internal representation of arithmetic operations in this case, for appropriate task generalization. We argue that numerical cognition on its own, without architecture or training procedure specializations, is not enough to guarantee development of arithmetic abilities via associative learning. Here, we focused on a combination of architecture and training specializations to provide interpretability of our results, but other solutions may employ different types of specializations, for example compositionality or logarithms to solve the problem of computing across arbitrary scales.



A RNNs architecture. Orange RNN processes the input in the first stage and produces a scalar to avoid passing down information about the individual numbers. Purple RNN processes the input in the second stage to produce the probability that the correct answer is in the left position. The RNNs are same size and this architecture ensures that the arithmetic computation takes place in the orange RNN, if at all. **B** Schematic of input data, the two numbers a and b. The two numbers were uniformly sampled for each dataset, train, validation, test1, test2 and test3. **C** RNNs' performance. Chance level (black line), medians (white ellipses), interquartile ranges (thick bars). 10k to 100k examples per dataset. Random seeds $n > 250$ RNNs and $N_{\mathrm{epochs}} = 10$k. **D** RNNs' performance by training type. **E** Addition error heatmap of the best "A.&D." RNN. Dashed lines mark the regions of the different datasets (Fig B). Color map scales differ by one order of magnitude between the magenta and cyan regions. **F** $z(t)$ and $p(t)$ for the best "A.&D." model and best "D. only" model. Each color represents a set of $\{a, b\}$ inputs with $a+b$ constant (thin dashed lines in upper left plot). $p(t)$ plots arranged such that $p(t) = 1$ is always correct.

Acknowledgments

DT is supported by the Gatsby Charitable Foundation GAT3708, The Kavli Foundation, and Boehringer Ingelheim Fonds. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We thank LF Abbott and the Abbott lab for helpful discussions.

References

Agrillo, C., Petrazzini, M. E. M., & Bisazza, A. (2015). At the root of math: Numerical abilities in fish. , *1*, 3–33.

Butterworth, B. (2005). The development of arithmetical abilities. *Journal of Child Psychology and Psychiatry*, *46*, 3–18.

Chittka, L., & Geiger, K. (1995). Can honey bees count landmarks? *Animal Behaviour*, *49*(1), 159–164.

Dacke, M., & Srinivasan, M. V. (2008, 10). Evidence for counting in insects. *Animal Cognition*, *11*, 683–689.

Daniel, M. M., Alvermann, L., Böök, I., & Schluessel, V. (2021). Visual discrimination and resolution in freshwater stingrays (potamotrygon motoro). *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, *207*, 43–58.

Gross, H. J., Pahl, M., Si, A., Zhu, H., Tautz, J., & Zhang, S. (2009). Number-based visual generalisation in the honeybee. *PLOS ONE*, *4*, e4263.

Howard, S. R., Avarguès-Weber, A., Garcia, J. E., Greentree, A. D., & Dyer, A. G. (2019). Numerical cognition in honeybees enables addition and subtraction. *Science Advances*, *5*, 961–967.

Kreuter, N., Christofzik, N., Niederbremer, C., Bollé, J., & Schluessel, V. (2021). Counting on numbers—numerical abilities in grey bamboo sharks and ocellate river stingrays. *Animals 2021, Vol. 11, Page 2634*, *11*, 2634.

Langdon, C., & Engel, T. A. (2022). Latent circuit inference from heterogeneous neural responses during cognitive tasks. *bioRxiv*, 2022.01.23.477431.

Mehlis, M., Thünken, T., Bakker, T. C., & Frommen, J. G. (2015). Quantification acuity in spontaneous shoaling decisions of three-spined sticklebacks. *Animal Cognition*, *18*, 1125–1131.

Schluessel, V., Kreuter, N., Gosemann, I. M., & Schmidt, E. (2022). Cichlids and stingrays can add and subtract 'one' in the number space from one to five. *Scientific Reports 2022 12:1*, *12*, 1–11.

Skorupski, P., MaBouDi, H. D., Dona, H. S. G., & Chittka, L. (2018). Counting insects. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*, 20160513.

Yang, G. R., Ganichev, I., Wang, X. J., Shlens, J., & Sussillo, D. (2018). A dataset and architecture for visual reasoning with a working memory. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11214 LNCS*, 729-745.