Symbolic Variables in Distributed Networks that Count

Satchel Grant (grantsrb@stanford.edu)
Stanford University
Deptartment of Psychology

Zhengxuan Wu (wuzhengx@stanford.edu)
Stanford University
Deptartment of Computer Science

James L. McClelland (jlmcc@stanford.edu)
Stanford University
Deptartment of Psychology

Noah D. Goodman (ngoodman@stanford.edu)
Stanford University
Dept. of Psychology and Computer Science

Abstract

The discrete entities of symbolic systems and their explicit relations can make symbolic systems transparent and easy to communicate. This is in contrast to neural systems, which are distributed, often with opaque inner mechanisms. It is understandable that psychologists tend to pursue interpretations of human cognition using symbolic characterizations, and it is clear that the ability to find symbolic variables within neural systems would be beneficial for interpreting Artificial Neural Networks (ANNs). Symbolic interpretations can, however, oversimplify non-symbolic systems. This has been demonstrated in children's performance on tasks thought to depend on a concept of exact number, where recent findings suggest a gradience of counting ability in children's learning trajectories. In this work, we take inspiration from these findings to explore the emergence of symbolic representations in ANNs. We align recurrent neural representations with symbolic representations of number by causally intervening on the neural system. We find that symbolic representations of numbers do emerge in ANNs. We use this to inform the discussion on how neural systems represent quantity. We also show that the symbol-like representations evolve with learning, and can continue to vary after the neural network consistently solves the task, demonstrating the graded nature of symbolic variables in distributed systems.

Keywords: Mathematical cognition; symbolic number; connectionist models; Distributed Alignment Search; neural networks

Introduction

Both biological and artificial Neural Networks (NNs) have powerful modeling abilities. Aside from the impressive capabilities of human cognition within biological NNs, artificial NNs have had recent successes crowning them as the "gold standard" in many machine learning communities (Alzubaidi et al., 2021). Despite their success, the inner workings of NNs remain largely opaque to humans, partly because representations in NNs are often highly distributed. Individual neurons in NNs often play multiplex roles (McClelland, Rumelhart, & PDP Research Group, 1986; Smolensky, 1988). Meanwhile, current tools often lack the ability to uncover precise mechanisms of artificial NNs (ANNs) from their distributed representations.

Symbolic systems, in contrast, defined by clear, discrete entities and explicit rules and relations, have the benefit of greater interpretability. These systems can often be designed so as to maintain mechanistic simplicity and interpretablity. Often this comes with the benefit of causal power over the systems, enabling us to change intermediate components for a desired output. Symbolic systems can lack, however, the expressivity and performance capabilities of NNs. This is apparent in the field of natural language processing where NNs such as Transformers (Vaswani et al., 2017) have swept the field. The field has witnessed a transformation from the power of scalable learning objectives and model scale using ANNs (Brown et al., 2020; Kaplan et al., 2020), easily surpassing the existing purely symbolic approaches. Is it possible to gain the interpretability of a symbolic system in these NNs? How can we explore the causal relationships of ANN representations?

Regardless of the performance differences of symbolic vs neural models, a main goal of scientific discovery is centered on the generation of simplified, symbolic interpretations of complex systems. These simplifications are necessary for understanding the essential parts of a system and how they causally interact. This form of understanding grants goal directed agents causal power over the system. Indeed, it could be argued that symbolic simplifications are necessary for goal directed agents to successfully interact with the world, or at least to share information with each other about how to do so.

One symbolic system is the system of natural or counting numbers – a system that many, but not all, humans know. In addition to young children, adults in some aboriginal tribes have been found to lack both number words and the ability to perform numeric equivalence tasks (Gordon, 2004; Frank, Everett, Fedorenko, & Gibson, 2008; Pica, Lemer, & Izard, V., & Dehaene, 2004; Pitt, Gibson, & Piantadosi, 2022). This has sparked intrigue about the necessary conditions for humans to learn representations of exact number. Is formal education required for humans to learn about numbers? Can symbolic representations of numbers exist without possessing words for them? How did humans develop symbolic number systems in the first place?

Theories of the development of numerical abilities have often been cast within a symbolic framework, but recent evidence suggests that many of these theories may fail to capture signs of gradience in children's acquisition of number. Gelman and Gallistel (1986) proposed that children's early performance on such tasks depends on 3 symbolic counting principles: *one-to-one correspondence*, *stable ordering*, and *cardinality*. This theory was challenged, however, by Wynn (1992), who showed that children who demonstrated the ability to perform counting tasks with very small numbers often

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

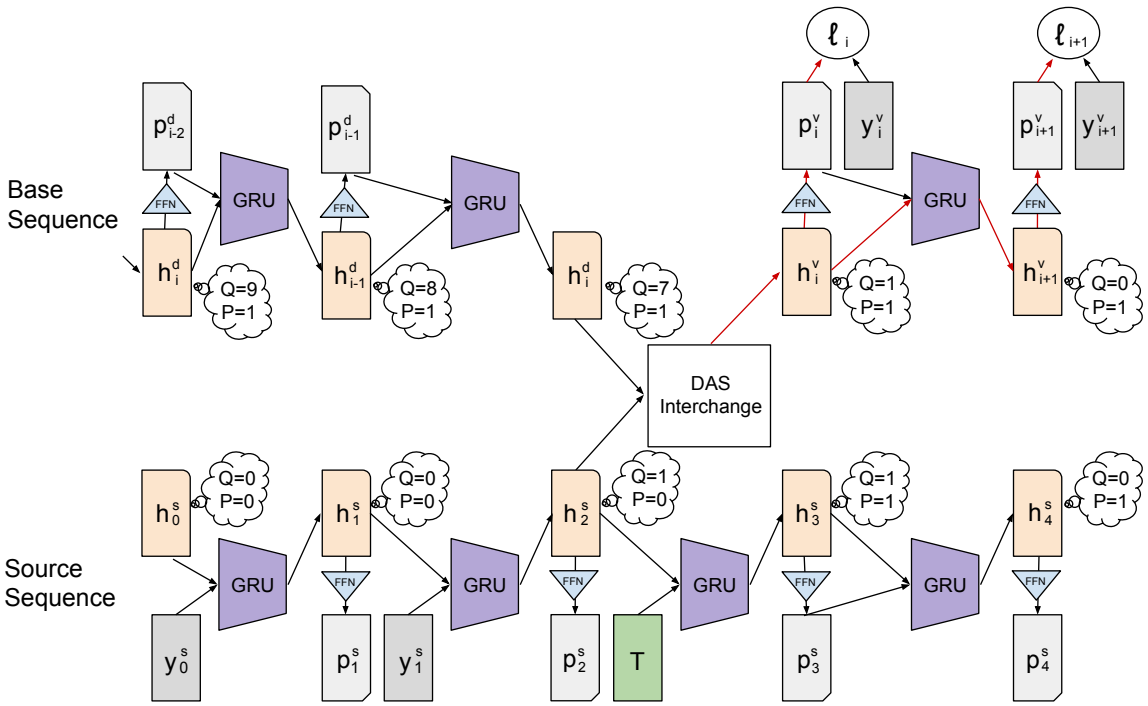

Figure 1: Diagram of a causal interchange performed on the hidden state of the GRU in the base sequence. The triangles are Feed Forward Networks (FFNs); the trapezoids are GRU cells; the thought clouds show the values of the hypothesized variables where Q is the Quantity and P is the Phase; in the upper right corner, ℓ_i and ℓ_{i+1} denote the loss over predicted logits. All other shapes other than the DAS rectangle are either h, for hidden state vector; p for predicted logits; or y which is a one hot encoding of ground truth. During the demonstration phase of the task, ground truth tokens are fed into the GRU. During the response phase, we use the models' predictions as input to each subsequent step in the GRU. The trigger token T signals the phase transition from demonstration to response. For simplicity, we draw an arrow directly from the logits or one hot encodings into the GRU cell, but in actuality, we use an embedding selected from the predicted unit with the maximum value. We harvest variables from the source sequence and inject them into the destination sequence for the causal interchange. In this example the source hidden state is h_2^s and the destination hidden state is h_i^d where the subscript denotes the index in the sequence, and the superscript denotes what sequence it is used in—s for source, d for destination (referred to as "base" in previous work), and v for intervened. We use a trained model with frozen weights to produce all predictions. We back-propagate into the DAS rotation matrix using prediction error on the counterfactual sequence under the hypothesized symbolic variables and causal interchange. Red arrows indicate the gradient flow in back-propagation.

failed to perform correctly with larger values in their count lists. The idea then arose that the induction of such principles coincided with the ability perform such tasks with sets containing more than 3 or 4 items. Sarnecka and Carey (2008) offered additional tasks thought to assess reliance on these principles, but the idea that these principles applied generally to all numbers in the child's count list was not supported by the subsequent work of Davidson, Eng, and Barner (2012); instead these authors found evidence of gradual acquisition of the ability to perform such tasks, progressing from smaller to larger numbers within the child's count list as several number-relevant abilities improved, consistent with the view that children's numerical abilities emerge gradually, and raising the possibility that their behavior may come to better align with symbolic principles as they gain more and more experience.

In our work, we take inspiration from the number cognition literature to ask if we can find symbolic representations of number in ANNs trained on numeric tasks. We then probe deeper into the relationship between model performance and symbolic alignment. We first show that we can find symbolic representations of number in Gated Recurrent Unit (GRU) (Cho et al., 2014) Recurrent Neural Networks (RNNs). We source this evidence by training a GRU to 100% accuracy on held-out test quantities in a counting task, and then we find a causal alignment between the latent representations of the trained GRU and a symbolic program. We find this alignment using a technique called Distributed Alignment Search (DAS) (Geiger, Lu, Icard, & Potts, 2021; Geiger, Wu, Potts, Icard, & Goodman, 2023) which enables us to test for the existence of symbolic programs in distributed representations using causal interventions. We find that the GRUs use a count up, count down strategy which increments and decrements a

single count variable based on the phase of the task. We contrast this with an alternative symbolic program that solves the counting task with two separate count variables—one to track the target quantity and another to track the response quantity. We further demonstrate the significance of our results by showing that the symbolic number alignment does not emerge in models trained on a similar task that can be solved using an exact token copy operation.

We use our results to demonstrate the utility of symbolic alignments within cognitive models. The alignments in this paper demonstrate the emergence of symbolic numbers within connectionist models. These findings have implications for the origins of exact symbolic number systems and for the necessary conditions for neural systems to learn how to count. These results serve as an initial step towards the unification of the symbolic and connectionist camps of thought in cognitive modeling.

We then explore deeper into the relationship between the symbolic alignment and the model performance to show that the symbolic alignment has a strong correlation with model accuracy, often being a leading indicator of a training phase transition. The relationship between the two, however, is not a perfect one-to-one correspondence, and frequently the alignment accuracy continues to change despite the model's ceiling task performance. We relate this to the symbolic gradience observed in children's number cognition, noting the similarity of these findings to that of children learning to count. Although this work is in its early stages, it has the potential to enhance our understanding of the emergence of symbols in the human mind.

We summarize the contributions of this paper as follows. 1. We demonstrate how and why to use DAS to interpret recurrent cognitive models. 2. We find the emergence of symbol-like counting variables in models trained to solve a numeric matching task, serving as a proof-of-principle for symbolic representations of number in human cognition. 3. We find that the symbolic representations of number strongly co-vary with model performance, although there is not a perfect one-to-one correspondence. 4. We use our findings to enhance our understanding of emergent symbolic variables in neural systems, making an initial step towards unifying symbolic and connectionist frameworks.

Methods

Overview

At a high level, we first train a model to completion on a numeric equivalence task, and then we use Distributed Alignment Search (DAS) (Geiger et al., 2021, 2023) to test for alignments between hypothesized symbolic abstractions/programs and the model's neural representations. Specifically, DAS learns an orthonormal rotation matrix (a change-of-basis matrix) at a given model layer to align a subspace of the neural representations at that layer with a causal variable from a hypothesized symbolic program. Concretely, we use the rotation matrix on the hidden representations at some layer and then swap a subspace (some fixed number of dimensions of the rotated representations) from the rotated representations under one model input into the rotated representation under a different input. We then invert the rotation and allow the model to make predictions using the intervened representation. The training objective for the rotation matrix is to find the change-of-basis such that the model will predict the hypothesized counterfactual outputs under the intervention given the hypothesized symbolic program. To evaluate how well the NN aligns with the symbolic abstraction, we perform these causal interventions on held out data and record the NN's performance on the counterfactual labels. The causal interventions allow us to verify whether or not the model is performing the task in a way that is consistent with the existence of the hypothesized symbolic abstraction.

Tasks

The tasks we focus on in this work consist of variable length sequences of tokens. We compare two relatively simple numeric equivalence tasks in this framework. The first task is called the *Copy Task*, and the second is called the *Counting Task*. Both tasks begin by uniformly sampling a target quantity from 1 to 20. After determining the target quantity, the sequence is constructed from two phases. The first is called the **demonstration phase** which begins with a Beginning of Sequence (BOS) token and continues with a number of demonstration tokens provided by the environment. Once the number of demonstration tokens following the BOS token is equal to the initially sampled target quantity, the environment provides a Trigger token (T) indicating the beginning of the **response phase**. The model is then tasked with outputting the same number of tokens as was observed during the demonstration phase followed by an End of Sequence (EOS) token to indicate that it has finished its response. The Copy and the Counting task variants differ as follows:

Copy Task: other than the BOS, T, and EOS token types, there is a single token type, C, that is used to both indicate the target quantity during the demonstration phase and the response quantity during the response phase. It is possible to solve this task without using an explicit notion of quantity, by copying the literal sequence of tokens in the demonstration sequence.

Counting Task: in addition to the BOS, T, and EOS token types, there are 3 demonstration token types (D1,D2, and D3) and a single response token type (RESP) that is different from the possible demonstration token types. During the demonstration phase, the demonstration tokens are uniformly sampled from the possible demonstration token types. The response phase deterministically uses the unique response token type. This task variant prevents a solution that uses a direct copy operation, and this variant has N^3 possible input sequences for each target quantity, N, making it more difficult to solve using a memorization strategy.

During DAS training we sampled 1000 sequences for training, and 500 samples of held out target quantities for validation. The held out target quantities were 4, 9, 14, and 17, selected to be relatively evenly distributed amongst the possible quantities.

Counting Model

We trained Gated Recurrent Units (GRUs) (Cho et al., 2014) to solve the aforementioned tasks. GRUs are a specific type of RNN that are similar to a Long Short-Term Memory (LSTM) networks except that GRUs have the advantage of using a single recurrent state vector. The hidden state, h_t, of the GRU at a particular step, t, is updated according to the following equations:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_t + b_{hr}) \quad (1)$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_t + b_{hz}) \quad (2)$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t * (W_{hn}h_t + b_{hn})) \quad (3)$$

$$h_{t+1} = (1 - z_t) * n_t + z_t * h_t \quad (4)$$

After each update step in the sequence, we make a prediction p_{t+1} of the next token, y_{t+1}, using softmax classification on the vector h_{t+1}. Concretely, we feed h_{t+1}—with a size of 20—into a Feed Forward Network (FFN) with a single hidden layer of 80 units, a Gaussian Error Linear Unit (GELU) nonlinearity (Hendrycks & Gimpel, 2023), dropout applied on the hidden activations with 0.5 probability to drop (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), and a softmax applied to the outputs of the second layer to create a probability distribution over possible output tokens. We use the cross entropy of the predictions with the ground truth tokens as the loss, $\mathcal{L}$, and minimize the loss using batch gradient decent with a batch size of 128. The loss for a batch of training data is calculated as follows:

$$\mathcal{L}_i = -\sum_{t=1}^{S-1} y_t^\top \log(p_t) \quad (5)$$

$$\mathcal{L} = \frac{1}{NS}\sum_{i=1}^{N} \mathcal{L}_i \quad (6)$$

Where i refers to the index of a single sequence in the batch, t is the index of the step in the sequence of length S where the 0^{th} step is not predicted, y_t is a ground truth, one-hot encoded column vector, and p_t is the output of the softmax function at the end of the FFN.

We used the ground truth tokens as input at each step in the sequence during the demonstration phase (i.e., teacher-forcing) up to and including the trigger token. After the trigger token, we use the predicted token of the current time step as the input token for the next time step (i.e., autoregressive) prediction vector to select the input token for the next step.

We used PyTorch with AutoGrad (Paszke et al., 2019), an Adam optimizer (Kingma & Ba, 2017) with default settings, an initial learning rate of 0.001, a learning rate decay schedule following that described in the original transformer paper (Vaswani et al., 2017). We trained 5 model seeds, each for 300 epochs. Models were trained using Nvidia Titan X GPUs. All models achieved $> 99.9\%$ accuracy on both training and validation sequences, where accuracy refers to the proportion of responses with the correct number of RESP tokens following the T token and ending with an EOS token.

Recurrent DAS

Causal abstraction is a hypothesis testing framework that manually tests if the causal mechanism of a variable in a symbolic program abstracts the causal mechanism of neural representations relative to a particular alignment (Geiger et al., 2021). Distributed Alignment Search (DAS) (Geiger et al., 2023) is a recent variant of causal abstraction where it turns the brute-force hypothesis search process into an optimization problem. DAS actively learns an aligned linear subspace of the full vector space by rotating the original neural representations using an orthonormal matrix. The question remains, how does the rotation matrix learn to find this subspace?

In this work, in each DAS training, we fix the number of dimensions of the intervention subspace. We allow these dimensions to be contiguous within the rotated representation under the assumption that a learned rotation matrix will be able to equivalently adapt to the particular dimensions that we isolate for the intervention. This narrows the number of possible subspace dimension choices to the range 0-D, where D is the dimensionality of the hidden state vector, h_t. We then perform a hyper-parameter search, trying different sizes of the subspace, and pick one based on alignment performance. For the rest of this work, we set the subspace to 10 dimensions given that the performance was relatively unchanged between approximately 5 and 15 dimensions.

With a fixed number of dimensions established, we can now write the the DAS intervention as follows:

$$h_i^v = R^{-1}(m * Rh_j^s + (1 - m) * Rh_i^d) \quad (7)$$

Where h^s and h^d refer to the GRU's hidden states under some randomly selected source and destination sequences respectively. h^v is the resulting, intervened hidden state. The subscripts i and j refer to the indices of the hidden states within their respective sequences. m is a mask of 1s and 0s of the same size as h where $*$ denotes the Hadamard product. We use m to select the neural subspace from the source vector, h_s, and inject it into the destination vector, h_d. Conceptually, we use m to inject some number of units from the rotated source vector to the rotated destination vector. R is the learned rotation matrix that is constrained to be orthonormal using PyTorch's orthogonal parametrization module.

The last piece of the DAS procedure is creating the training signal for the rotation matrix. The general idea is to predict the counterfactual tokens under the hypothesized symbolic program given the causal intervention. See Figure 1 for a diagram of the full process. Specifically, we create the training signal by using the symbolic causal program to generate the tokens we would expect under the causal intervention. We

then use these tokens for the same autoregressive, next token prediction task that we used to train the neural model on the original task. We freeze the model weights and backpropagate the learning signal into the orthogonally parametrized rotation matrix. We use the model's accuracy on the counterfactual labels to evaluate the validity of the alignment. In the case where there exists a meaningful alignment between the neural model and the symbolic program, we expect the model's accuracy to be high under the fully trained causal interventions. In the case where there does not exist a valid alignment, the resulting accuracy should be low.

In sequence based modeling, the ordering of the tokens conveys important information. We have the option of which indices to sample for the destination and source vectors when performing DAS. In this work, we exclude indices that correspond to BOS, trigger, and EOS tokens. We show results that uniformly sample the source and destination indices j and i from both the demonstration and response phases (excluding sampling of indices that correspond to the BOS and EOS tokens).

For the training, we used 10000 destination-source sequence pairs with 1000 pairs for validation. Similar to the standard model training, we used the held out target quantities 4, 9, 14, and 17 for the validation sequences. Using held out target quantities gives more insight into the generalization capabilities of the models' solutions. It also gives insight into DAS's capabilities, as it is a relatively young technique for interpreting neural models. We used a learning rate of 0.001 and a batch size of 512 sequence pairs. We trained the rotation matrix until convergence on the autoregressive loss.

Table 1: Alignment Results

Task	Program	Variable	Acc
Counting	Stack	Quantity	0.94±0.02
Counting	Stack	Phase	0.91±0.02
Counting	Match	Demo Quant	0.40±0.03
Counting	Match	Resp Quant	0.35±0.02
Copy	Stack	Quantity	0.52±0.05
Copy	Stack	Phase	0.60±0.07
Copy	Match	Demo Quant	0.24±0.02
Copy	Match	Resp Quant	0.30±0.06

Causal Models

As previously described, DAS requires that the experimenter first has a testable, high level, symbolic, causal program. DAS then finds the best alignment of this program within the neural representation space. We tested for the existence of two different causal programs. The **Stack Program** is an algorithm in which the model uses a Quantity variable (Q in Figure 1) to track the count and a Phase variable (P in Figure 1) to track the phase of the sequence. It increments the Quantity during the demonstration phase and decrements it down during the response phase, knowing to stop when it hits 0. In contrast, the **Match Program** uses a Phase variable and two count variables—Demo Quant and Resp Quant. It increments these count variables during the demonstration and response phase respectively knowing to stop when Demo Quant is equal to Resp Quant during the response phase. We included Algorithm 1 in the Appendix to detail a step-by-step account of a single sequence step in the the Stack and Match programs.

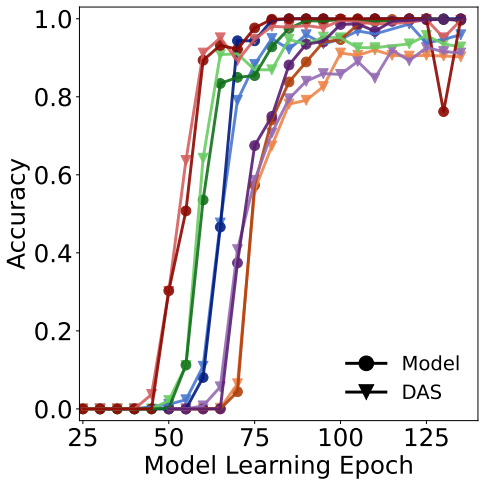

Figure 2: The relationship between model accuracy and DAS over training. Each point is from a model checkpoint trained on the Counting Task. The models' task performance is denoted by darker hues with circle markers. The DAS performance on the Quantity variable for the same checkpoint is in lighter hues with triangle markers. Here we see that the DAS performance correlates with task performance.

We note the existence of an infinite number of equivalent implementations of each of these programs respectively. For example, an equivalent program to Stack is one where the program immediately adds and subtracts 1 from the Quantity variable before carrying out the rest of the program as we have described. DAS only has the ability to discriminate between programs that are causally distinct from one another. We wish to acknowledge that there are many more experiments that we could do to further refine the symbolic program(s) in this work. We leave this exploration to future work.

Results/Discussion

We begin by addressing the symbol-like nature of the NN task solutions and by demonstrating that DAS can help us identify which symbolic program the NNs find. We can see from Table 1 the results of our DAS alignments. First, we note that we see relatively high alignment accuracies in the Quantity and Phase variables from the Stack program for the Counting Task models compared to the results from the Match program. We see this comparative difference when comparing the Quantity variable to both the Demo Quant and Resp Quant variables.

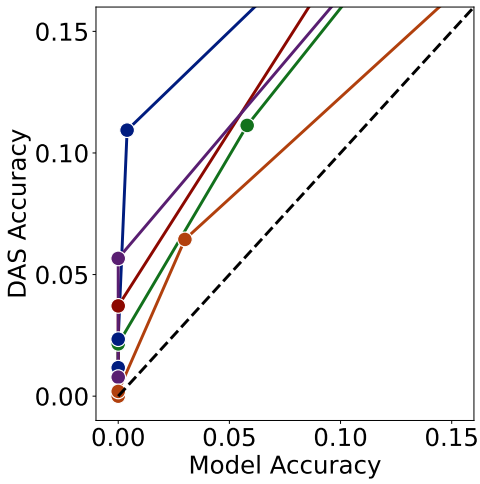

Figure 3: The x-axis shows task performance whereas the y-axis shows the DAS performance for the same model checkpoint on the Quantity variable. Colors denote model seed. We see that DAS performance tends precede the model's learning transition.

These comparisons suggest that the Stack model characterizes the solutions better than the Match program.

To further explore the meaning of our results, we compare models trained on the Copy and Counting tasks. The purpose of this comparison is to demonstrate that our findings for the Counting Task models are not merely performing an object dependent, copy operation, but rather performing an object independent numeric operation. We see from the comparatively low alignments of the Copy Task models that the two training variants have resulted in networks with different solutions. Although we cannot make claims about the exact nature of the model solutions from this result, we can use it to further delineate the interpretation of our alignment accuracies.

These results demonstrate that symbol-like representations of discrete numbers do emerge in neural models that have only been trained to produce the same number of items that they first observed. This is a proof of principle that neural systems do not need formal instruction for symbolic representations of number to emerge, nor do they need built in counting principles to inform their learning. This has implications for the mechanisms by which humans developed symbolic number systems. Perhaps the need to keep track of specific quantities arose in some cultures, creating task situations that depended on the ability to learn to represent exact quantities.

We turn now to the developmental trajectory of the causal alignments displayed in Figure 2. We can see from the performance curves that both the model accuracy and DAS performance begin a transition away from 0% at similar epochs. They also exhibit similar performance trajectories. We note that in all cases, the jump in DAS alignment precedes that of the model, as shown in Figure 3. This result can be contrasted with a hypothetical result in which the alignment curves lag behind the performance of the model. In this hypothetical, alternative case, a possible interpretation could have been that the network develops unique solutions for many different input-output pairs and progressively unifies these solutions. The results that we see in Figures 2 and 3 can be used as evidence for an early sign of an emergent, symbol-like solution in the NNs even at earlier training epochs. Perhaps this is to be expected in light of works like Saxe, McClelland, and Ganguli (2019) and Saxe, Sodhani, and Lewallen (2022) which show an inherent tendency for NNs to find solutions that share network pathways.

Despite the similarities between the model performance and alignment performance in Figure 2, we can see that the alignment performance often fails to achieve 100% alignment. Furthermore, the alignment performance continues to vary despite the models' ceiling task performance. We interpret these results as a reminder that representations in distributed systems exist on a continuum despite seemingly discrete, symbolic performance on the task. These results have an analogy to children's number cognition in which children may appear to possess a symbol-like understanding of exact numbers and their associated principles, but when probed deeper, the symbol-like picture falls apart. This can also be related to the Large Language Model (LLM) literature, in which the notion of sharp changes in performance as a function of LLM scale have been demonstrated to be a function of metric rather than a sudden change in innate ability Schaeffer, Miranda, and Koyejo (2023).

We use these results to highlight the nuances of symbolic interpretations in cognitive science. In one light, we have shown that we can find causal symbols within the distributed representations of the neural systems. We have also shown that these symbol-like representations emerge in a way that correlates with model performance. These results can inform our thoughts on the nature of distributed solutions in cognitive models—symbol-like representations are perhaps an inherent property of general distributed solutions, at least when the task is one that can be solved by relying on such representations. In another light, however, we see that it is easy to overly simplify our understanding of a distributed system based on task performance alone. There is a nuanced picture to distributed solutions that must be considered to understand the full picture of human cognition.

Lastly, we highlight the utility of DAS for interpreting distributed systems. In this work, DAS has revealed a crucial piece of the puzzle to understanding number representations in neural systems. DAS also demonstrated the subtle, graded nature of distributed solutions, which we see in human cognition. These findings serve as a bridge between connectionist and symbolic frameworks for understanding human cognition, suggesting that DAS will continue to be a useful tool for understanding the computations performed in cognitive models that rely on ANN systems.

References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. Retrieved from https://doi.org/10.1186/s40537-021-00444-8 doi: 10.1186/s40537-021-00444-8

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). *Language models are few-shot learners.*

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, *abs/1406.1078*. Retrieved from http://arxiv.org/abs/1406.1078

Davidson, K., Eng, K., & Barner, D. (2012). Does learning to count involve a semantic induction? *Cognition*, *123*(1), 162-173. doi: https://doi.org/10.1016/j.cognition.2011.12.013

Frank, M. C., Everett, D. L., Fedorenko, E., & Gibson, E. (2008). Number as a cognitive technology: Evidence from pirahã language and cognition. *Cognition*, *108*(3), 819-824. doi: https://doi.org/10.1016/j.cognition.2008.04.007

Geiger, A., Lu, H., Icard, T., & Potts, C. (2021). Causal abstractions of neural networks. *CoRR*, *abs/2106.02997*. Retrieved from https://arxiv.org/abs/2106.02997

Geiger, A., Wu, Z., Potts, C., Icard, T., & Goodman, N. D. (2023). *Finding alignments between interpretable causal variables and distributed neural representations.*

Gelman, R., & Gallistel, C. R. (1986). *The child's understanding of number*. Harvard University Press.

Gordon, P. (2004). Numerical cognition without words: Evidence from Amazonia. *Science*, *306*(5695), 496–499. doi: 10.1126/science.1094492

Hendrycks, D., & Gimpel, K. (2023). *Gaussian error linear units (gelus).*

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... Amodei, D. (2020). *Scaling laws for neural language models.*

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

McClelland, J. L., Rumelhart, D. E., & PDP Research Group (Eds.). (1986). *Parallel distributed processing. Volume 2: Psychological and biological models*. Cambridge, MA: MIT Press.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *CoRR*, *abs/1912.01703*. Retrieved from http://arxiv.org/abs/1912.01703

Pica, P., Lemer, C., & Izard, V., & Dehaene, S. (2004). Exact and approximate arithmetic in an amazonian indigene group. , *306*(5695), 499–503.

Pitt, B., Gibson, E., & Piantadosi, S. T. (2022, feb). Exact Number Concepts Are Limited to the Verbal Count Range. *Psychological Science*, 095679762110345. doi: 10.1177/09567976211034502

Sarnecka, B. W., & Carey, S. (2008). How counting represents number: What children must learn and when they learn it. *Cognition*, *108*(3), 662–674.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2019, May). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, *116*(23), 11537–11546. Retrieved from http://dx.doi.org/10.1073/pnas.1820226116 doi: 10.1073/pnas.1820226116

Saxe, A. M., Sodhani, S., & Lewallen, S. (2022). The neural race reduction: Dynamics of abstraction in gated networks.

Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are emergent abilities of large language models a mirage?

Smolensky, P. (1988). On the proper treatment of connectionism.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. Retrieved from http://arxiv.org/abs/1706.03762

Wynn, K. (1992). Children's acquisition of the number words and the counting system. *Cognitive psychology*, *24*(2), 220–251.

Algorithm 1 One sequence step of the Stack Program

$q \leftarrow \mathit{Quantity}$, $p \leftarrow \mathit{Phase}$, $y \leftarrow$ input token
if $y ==$ BOS **then** ▷ BOS is beginning of sequence token
  $q \leftarrow 0$, $p \leftarrow 0$
  return sample(DEMO) ▷ sample a demo token
else if $y \in$ DEMO **then** ▷ DEMO is set of demo tokens
  $q \leftarrow q + 1$
  return sample(DEMO)
else if $y ==$ T **then** ▷ T is trigger token
  $p \leftarrow 1$
else if $y ==$ RESP **then** ▷ RESP is response token
  $q \leftarrow q - 1$
end if
if $(q == 0)$ & $(p == 1)$ **then**
  return EOS ▷ EOS is end of sequence token
end if
return RESP