Emerging Representations for Counting in a Neural Network Agent Interacting with a Multimodal Environment

Silvester Sabathiel[1,3], James L. McClelland[2] and Trygve Solstad[3]

[1]Department of Computer Science, NTNU, NO-7491 Trondheim, Norway
[2]Department of Psychology, Stanford University, Stanford, CA, 94305, USA
[3] Department of Teacher Education, NTNU, NO-7491 Trondheim, Norway
trygve.solstad@ntnu.no

Abstract

Learning the procedure of counting represents a major step in children's development of the concept of the natural numbers. How children acquire generalized concepts of number and counting skills is still under debate. Here we investigate how a neural network agent develops representations for key concepts of counting while learning to perform several different counting tasks in a multimodal, interactive environment. We identify neural activity and connection patterns that realize a) a representation of the entity to count that was invariant to the task, b) a mapping from entity to number-word, and c) a representation of the number of entities that have been counted that was shared between tasks. The results support the notion that abstract representations of number can arise from integrating experiences across a range of number-related tasks.

Keywords: mathematical cognition; neural networks; learning to count; multimodal; representation;

Introduction

Learning to count involves acquiring several abstract concepts necessary to apply the procedure reliably and in a wide range of settings. Children usually reach basic competence with counting in the earliest stages of their education, although a full understanding of the generality of the relevant concepts is often only reached years later (Davidson et al., 2012). Even though key developmental stages of learning to count have been identified and analyzed (Clements and Sarama, 2009; Gelman and Gallistel, 1978), how children develop an ability to count that can be applied across modalities and contexts is still unclear. In particular, how the knowledge required for counting is represented and how this knowledge develops during learning are open questions.

One theory of knowledge and its origins holds that knowledge consists of explicit systems of rules or propositions, and learning is viewed as enriching or re-structuring such representations (Spelke et al., 1992). For the case of the counting concept, key principles that have been identified by Gelman and Gallistel (1978) include:

1. *"Correspond each entity to exactly one number word" (One-one principle)*
2. *"The order of the number words must follow a fixed sequence" (Stable order principle)*
3. *"The count word used on the last item in a set represents the number of items in the set" (Cardinality principle)*
4. *"The above principles can be applied to entities of any kind" (Abstraction principle)*

The theory that children represent their knowledge of numbers and counting in a rule-based structure is the basis for the *knower-level theory* where the acquisition of the *cardinality principle* is treated as reflecting a sudden rule induction (Sarnecka and Carey, 2008). A computational model developed in Piantadosi et al. (2012) shows how a system - given a defined set of primitive symbol processing operations - can *bootstrap* the meaning of number words. In their work it is the pre-specified symbolic primitives that give rise to the critical inductive leap/qualitative phenomena in the literature of the knower-level theory. However, studies presented in Davidson et al. (2012) indicate that children's ability to perform tasks thought to depend on the cardinality principle emerges gradually and is associated with their ability to perform other number related tasks. This suggests that learning to count might a) involve more gradual learning processes and b) involve the discovery of an integrated understanding through learning to perform several number related tasks.

The work we report here is part of a project in which we explore whether neural network-based learning models formulated within the parallel distributed processing framework (Rumelhart, 1986; Rogers and McClelland, 2014) can capture these aspects of children's number learning. We seek to understand how symbol-like processes could arise through a neural network's learning process. In Fang et al. (2018) it was shown that a recurrent neural network could learn to count squares arranged in a linear array from interacting with an external environment without explicit/symbolic representations or the assumption of given primitive cognitive operations. However, as many have argued, this specific task by itself might be performed by the neural network without acquiring a general or abstract understanding of number.

To explore the possibility that a concept of counting could

arise from integrating experiences across a range of number-related tasks, we recently extended this approach to a neural network agent that learned to solve several different counting-related tasks (Sabathiel et al., AfP). The tasks capture key conceptual aspects of the tasks humans are tested and educated on: recite N number words, count temporally and spatially distributed objects, and perform a give-n objects task by moving a given number of objects to a target location. The approach successfully exhibits cross-task generalization, in that the network learns a new task more quickly after previously learning other tasks. That work did not, however, explore the representations the network used to solve the task and support transfer learning – representations that might capture task-invariant aspects of the counting principles. In the current work, we were especially interested in exploring these representations. Specifically, we focus on the following research question:

Can abstract representations for the concepts of number and counting emerge in a neural network agent learning to perform different counting-related tasks?

Setup and Methods

The learning system and its environment is similar to that in (Sabathiel et al., AfP) and described in the following.

Learning Environment

The learning environment provides an artificial agent with a multimodal, interactive interface. The world is a 4x4 grid with two binary features at each grid point, one signaling the presence of an object at the grid location and another signaling the presence of the agent's hand (Fig. 1). The interface allows a set of motor and linguistic outputs to the environment and allows teaching signals corresponding to these outputs to guide learning. The set of motor actions are one-step movements of the hand in 2D space (left,right,up,down), as well as a touch, picking up and release action. The language output consists of the count words 'one' to 'nine' and the word *'Stop'*.

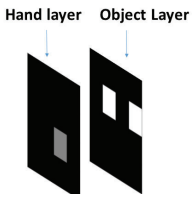
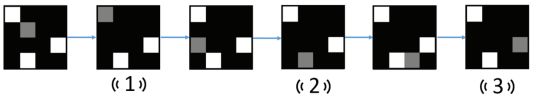
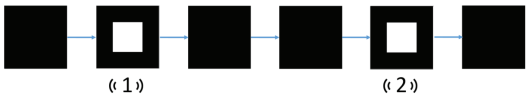


Figure 1: 2-layered visual input - the gray square represents the hand and the white squares represent the objects to be counted in the environment.

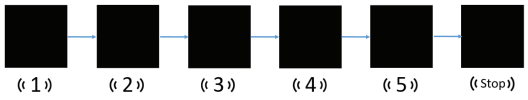
Task descriptions The agent was trained on four different tasks commonly used to investigate number comprehension in children described in Fig. 2.



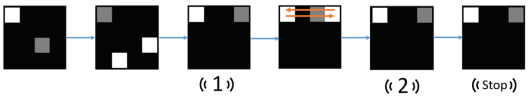
(a) *Count all objects:* The agent is to touch each object exactly once and say the corresponding number word as each object is touched.



(b) *Count all events:* The agent is to say the right number word after each flash of a white square in the visual input.



(c) *Recite-N:* The agent is given an arbitrary number, N, from 1 to 9 as input in the language channel and is to recite the number words from 1-N and say *'Stop'* afterwards.



(d) *Give-N*: The agent is given an arbitrary number, N, from 1 to 9 as input in the language channel and is to move N white squares from an infinite source of white squares in the upper left corner of the environment to the target in the upper right corner of the environment. The agent is to say the corresponding number word after each given square and say *'Stop'* after N objects have been moved.

Figure 2: Description of the four counting tasks and illustration of a solution process including the visual output and verbal input. Spoken words are denoted below the image. White squares represent manipulable objects and the gray square represents the movable hand. Green arrows denote the movement of the hand between the depicted time instances.

Learning system

Demonstration-driven learning We simulate a learning situation in which the agent observes and anticipates the actions of a teacher performing the counting tasks. For each time step, the agent receives feedback about the correct action of the teacher, and adjusts its connection weights based on the prediction error using backpropagation. The approach contrasts with reinforcement learning based approaches in which the environment only sets tasks and provides rewards. We argue that this approach captures important features of the environment in which children learn, allows the model to adopt culturally-defined counting habits, and accords with evidence that children's number learning is influenced by adults' use of number to refer to items present in the child's environment (Gunderson and Levine, 2011). In the subsequent testing situation, the agent acts autonomously without feedback, essentially learning to imitate the motor and language actions of the teacher. Importantly,

as discussed below, the scoring of performance during assessment is based on adherence to number principles, rather than the specific actions specified by the demonstration algorithm.

Neural Network Architecture The network architecture is illustrated in Fig. 4. The choice of architecture was motivated by the need to balance simplicity and generalizability on one hand with the computational capacity necessary to process the complexity of the multimodal sensory input, action space, and temporal dependencies inherent in the counting tasks on the other. The network architecture includes two channels, representing a visual modality and an auditory language modality:

Visual channel: The visual channel is a ConvLSTM Xingjian et al. (2015), which takes the 4x4 image as input and is designed to have the capacity to remember long term dependencies of the visual input. The ConvLSTM is a version of an LSTM whose internal structure uses replicated channels that tile the input space, as in a feed-forward Convolutional Neural Network (CNN), allowing the network to develop a spatially structured working memory. In our architecture the CNN consists of 5 kernels of size 3x3 applied with a stride of 1 and a zero padding of size 2. The concatenated, flattened 2D hidden state of the ConvLSTM and the task vector are fully connected to 70 units with ReLU activation function, constituting the output of the visual channel, which we denote as Visual representation.

Language channel: The language channel receives the task instructions encoded in a layer with a total of 20 binary units (Fig. 3), where the first 5 units encode the verb (action) of the task instruction, the subsequent 10 units encode the quantifier (*'1', '2', ..., '9', 'ALL'*) and the last 5 units encode the entity that is to be counted. Each of these 'words' is one-hot encoded. To allow for the possibility to extend the instruction vocabulary that can be tested with the same architecture, some of the units in the task instruction have been left without 'meaning' (and thus are not shown in Fig. 3). The language channel also receives the network's full output vector from the last time step, serving as an 'efference copy' of the linguistic activations in the network's output layer, independently of whether an overt action was emitted to the environment. These two vectors are concatenated to form the input to the LSTM. The data is then processed via a standard LSTM with an internal vector size of 33 units, with fully connected weights to learn the LSTM's gates (Hochreiter and Schmidhuber, 1997). The output of the LSTM is also a 33 unit vector.

Output: An intermediary, multimodal 'Visual-Language' representation with 103 units is produced by concatenating the output from the visual channel and the output from the language channel. A fully connected feedfoward network maps this representation to the action space where each unit represents one action and can take values from 0 to 1 independently (sigmoid activation function). The most active action is produced if the activation of the M unit exceeds 0.5, and the most active verbal output unit is produced if the activation of the V unit exceeds 0.5.

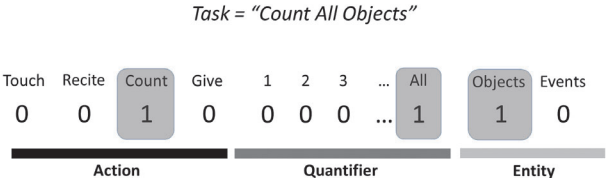


Figure 3: Concatenated task vector, which is given as input for the language channel. The vector corresponds to a statement consisting of an action, a quantifier and an entity and is used to instruct the agent about what task to perform.

Learning algorithm The network is trained via supervised learning with an automatic solving algorithm used to create the teaching signal. For each time step, the agent predicts the action $\mathbf{y}$ of the teaching signal $\hat{\boldsymbol{y}}$, where the teaching signal decides which action is executed and therefor the next state of the environment. After each whole trial, the weights θ of the neural network are updated using backpropagation to minimize the sum across items in a batch and steps of the action sequence of the mean-square-error between the output vector of the network and the encoded vector for the action from the teaching signal: $\mathcal{L}$. The batch-size is 8 times the number of tasks that are to be learned in the current learning schedule. An epoch of training is defined as one full forward and backward pass of the whole batch. After each trial a new batch is uniformly drawn from the tasks and the number of entities to be counted. The network was trained with the ADAM-optimizer Kingma and Ba (2014) and a learning rate of 10^{-2} in early learning stages (until the average loss dropped to 0.2) and 10^{-3} for the upcoming epochs in later learning stages. During the training dropout was applied to the layer, which is denoted as 'Visual-Language representation' in Fig. 4 where each unit in the corresponding layer was set to 0 with probability 0.4.

Representations In this work, the word representation is used to refer to the particular pattern of unit activity in the neural network in a specific situation, such as the situation in which the network has just encountered the third event in the 'count-the-events' task. We use analyses described below to determine whether these representations capture shared features of different situations, such as encountering the third item to be counted in different tasks.

Hinton diagrams For the visualization of the neural activity and the weights of the neural network we use *Hinton diagrams* (Hinton and Shallice, 1991), which allow for a graphical analysis of values in 2D arrays. In a Hinton di-

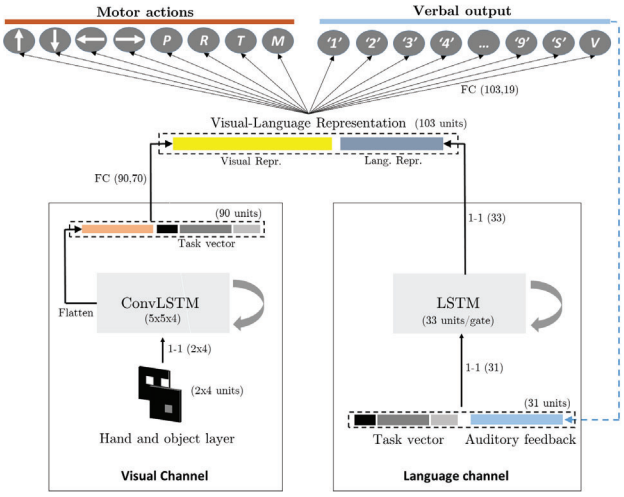


Figure 4: Neural Network architecture with multimodal channels: The output of the visual channel consists of the output of the ConvLSTM concatenated with the task input, and is fully connected (FC) to the next layer. The output of the language channel (LSTM) is concatenated with the output of the visual channel to form the multimodal 'Visual-Language' representation. The next motor action and the verbal output are computed via fully connected weights (FC) from the Visual-Language representation. FC(n,m) denotes fully connected weights from n to m units. 1-1(n) denotes a one-to-one mapping from n to n units. Dashed boxes denote the concatenation of their contained vectors and arrows represent the mapping between the layers according to their annotations. Annotations of the motor output nodes are arrows denoting the possible actions, including hand movements in the four directions, picking up (*P*), releasing (*R*) and touching (*T*) the object at the current position of the hand. For the language output, *S* represents the word *Stop* and the numbers represent the corresponding number words. The action represented by the most active output neuron is not executed unless the values of the two extra nodes *M* and *V* exceed the threshold of 0.5.

agram, positive values are shown as white squares and negative values by black squares. The sizes of the squares represents the magnitude of the weights or node activation. We use these diagrams as an exploratory analysis tool to qualitatively identify potential representations of the network, before performing quantitative analysis.

Representational Similarity Analyses We use representational similarity analysis (RSA) based on correlations between patterns of activity to make two types of arguments in this work: First, within a particular task, we use RSA to show that the network uses representations that allow it to distinguish between important concepts. That is, the representations for states related to a particular concept are similar to each other and dissimilar from representations of states not connected to the concept. Second, comparing between tasks, we use RSA to test for the abstractness of the representation. Here, we consider whether the representations of the concepts are relatively independent of context, by showing that states corresponding to instances of the same concept are similar across different class contexts.

For this purpose, we assess representational similarity on two subsets of nodes in the layer which we denoted as Visual-Language-Representation in Fig. 4: (1) the subset of nodes which receive their signals from the visual channel ('Visual representation') and (2) the subset of nodes receiving the signal from the language channel ('Language representation'). We also consider the connection weights from these nodes to illustrate the roles they play in determining the behavior of the network. Each of the three analyzed counting tasks involves a different kind of counted entity, either objects at different positions in the count-all-objects task; sequentially occurring events in the count-all-events task, or sequentially given items in the give-N task.

For the analysis in this work we recorded the neural activity in test runs, such that we obtained sets of node activities for all possible combinations of the representation, concept and context. To calculate a scalar measure of the similarity between any pair of the node vectors we used the Pearson correlation and show them in form of color-coded correlation matrices in the results.

Results

Training and testing the model

We trained a single instance of the neural network agent (see Methods) to solve all four counting tasks: Recite the list of number words (Recite N), count the number of flashes (Count-all-Events), move a given number of objects across the environment (Give-N), and count a given number of objects in the environment (Count-all-Objects). Within 15000 epochs the agent reached perfect performance in twenty consecutive trials on all tasks for the counting numbers one through nine.

To investigate if the agent learned a unified, abstract procedure for counting applicable to all tasks, or separate counting strategies specific to each task, we analyzed the representations in the different channels of the network as the agent solved the different tasks. Leaving the Recite N task out, in the following analysis we focus on the three tasks that involve counting entities.

Identification of entities

For the counting procedure to be widely applicable, it must operate on an abstract concept of object or entity that is not specific to any particular context or task. We asked if training the network to count different kinds of entities (such as

spatially distributed objects and temporal events) was sufficient to develop representations for such general entities, and looked for network representations that reliably correlated with the current presence of an entity to count.

Figure 5 shows the Visual representation of the neural network using a Hinton diagram (see Methods). To see if some nodes were preferentially responding when an entity to be counted was present in the visual field during the count-all-objects task, the neural activity is shown for different time points at which the neural network agent encountered an entity to count (numbers 1-9), and four time points at which it did not encounter an entity to count.

From the node activation pattern we see that some nodes were preferentially turned on when no entity was encountered and off only when an entity was to be counted. We refer to these nodes as *entity nodes* (highlighted in yellow). The same observation could be made on the population level for each task: We compared the Visual representation in each task with those of the other tasks both when an entity was to be counted (separately considering the first to the ninth entity) and when no entity was to be counted. The correlation matrices in Fig. 6 show that the population of visual nodes distinguished clearly between times when an entity was to be counted and times when no entity was to be counted. Correspondingly, a statistical analysis of 100 simulation runs showed that there was high correlation ($r = 0.89 \pm 0.11; Mean \pm STD$) between vectors of node activities in the Visual representation when an entity was to be counted and low correlation ($r = 0.14 \pm 0.11; Mean \pm STD$) between the corresponding vectors when entities were counted vs. when there was no entity to count.

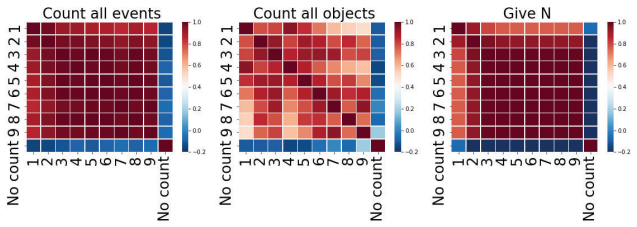


Figure 6: Similarity between representations in the visual layer for different numbers and the non-presence of an entity in different tasks, as measured by pairwise correlations of node activation vectors. The Visual representation distinguishes sharply between instances when there was an entity to count and when there was not, but did not distinguish between the number of entities that had been counted.

To see whether the representation of entity was *'abstract'* -in the sense that the representation was independent of the kind of entity, task, and the spoken number word - we identified three nodes in the Visual representation that were selectively activated when the agent did not encounter an entity to be counted for all three tasks. The activity of all these abstract entity nodes was recorded for 100 trials for each task (see Fig. 7. The nodes' activity was close to zero ($a = [0.03, 0.005, 0.004] \pm [0.01, 0.01, 0.02]; Mean \pm STD$) for all counted numbers and tasks when there was an entity to count, and substantially higher for all tasks when there was no countable entity present($a = [3.15, 2.88, 2.86] \pm [1.51, 1.38, 1.34]; Mean \pm STD$), indicating that the rep-

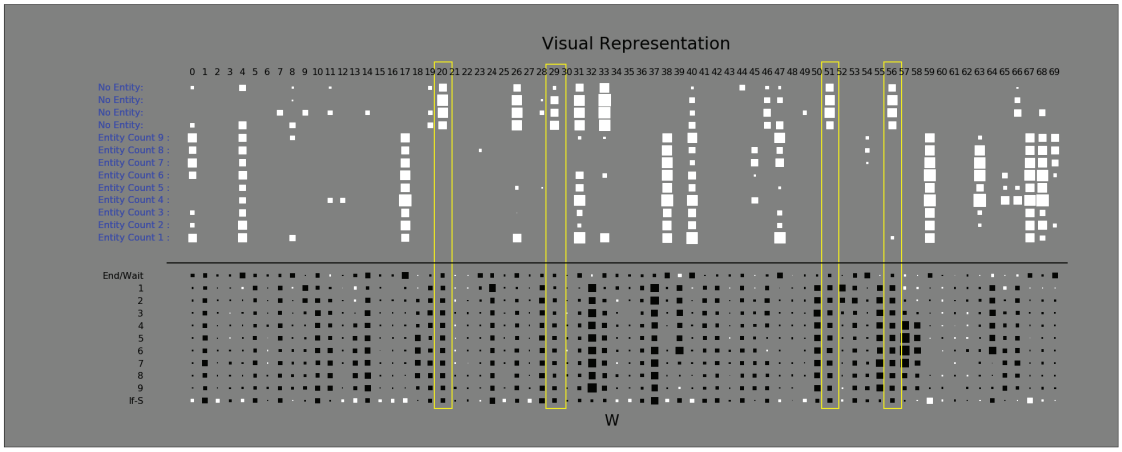


Figure 5: Hinton diagram (see Methods) of the enumerated node activations (above the horizontal black line) and connection weights (below the horizontal black line) in the Visual representation. White and black squares represent positive and negative activation values respectively. The size of the squares represents the magnitude of the weights and node activations. Nodes that are selectively active whenever there is no entity to count for all tasks are highlighted in yellow. The uniform connection weights of these nodes to the output of the number words indicate their number-independent inhibitory function when no entity is counted.

resentation of entity was independent of the kind of entity and task.

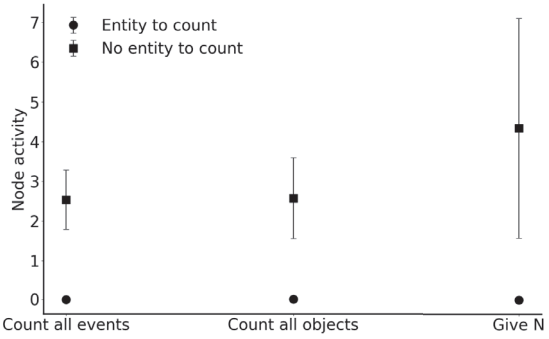


Figure 7: Mean and standard deviation of the activity of neuron number 20 in the Visual representation for time steps when an entity is to be counted vs. when there is no entity for different tasks. The plot shows that the activity encodes countable entities independent to the kind of entity.

One-to-one correspondence between entity and number words

Establishing 1-1 correspondence between entities and number words is an important developmental stage in learning to count, and consists of at least two components: i) avoiding counting the same object twice, and ii) moving to the next number word only when a new object is encountered, and not before. In previous work (Sabathiel et al., AfP), we identified the emergence of a memory mechanism in the visual layer of the network that would allow the agent to avoid counting the same object twice in the count-all-objects task. In the present analysis, we also identified a mechanism for the second component of 1-1 correspondence. Inspecting the weight matrix of the Hinton diagram of Fig. 5, we see that all entity nodes were connected to all the verbal number word nodes with equal inhibitory strength. This means that whenever there is no entity to count, speaking the next number word will be avoided through inhibitory control.

The state of the visual layer decided if a counting word was to be spoken or not. However, this layer did not distinguish between the different numbers. Keeping track of the number of entities counted was guided by the Language representation, as we will see in the next section.

Abstract number representations

A fully generalized counting procedure needs to operate on a concept of discrete numbers that is abstracted away from the specific context it was learned in. From the activity pattern in the Hinton diagram of all the nodes in the language channel (Fig.8) we see that each number had a distinct Language representation that was highly similar for different tasks.

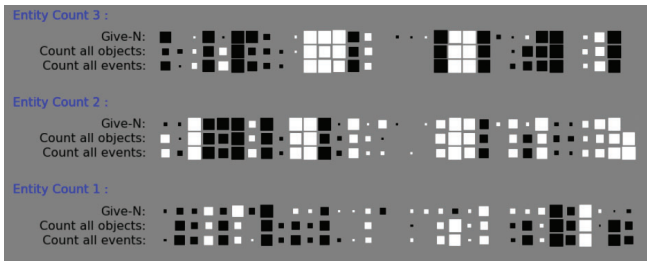


Figure 8: Hinton diagram of the node activation in the Language representation for all tasks. White and black squares represent positive and negative activation values respectively. The size of a square represents the magnitude of the node activation. The activity vectors for the same numbers show similar patterns for different tasks, while activity vectors for different numbers show different patterns.

To quantify the impression that the language layer contained an abstract representation of the current number of entities counted, we simulated 100 trials of each counting task and compared the similarity of the Language representations separately for each of the three tasks when a different number of entities had been counted. The similarity between representations for each pair of distinct numbers was low within each individual task ($r = -0.09 \pm 0.09; Mean \pm STD$), indicating that the language channel had developed distinct representations for each number (Fig.9).

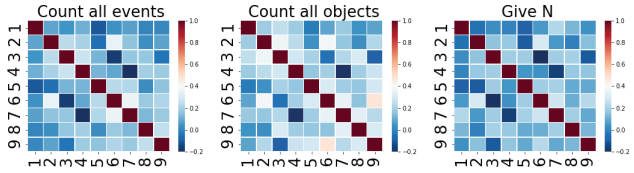


Figure 9: Similarity between representations in the language layer for different numbers in each task as measured by pairwise correlations of node activation vectors. Each number has a distinct language representation.

The Language representations for number were task-independent, as seen by the high correlation between representations of each number for different tasks ($r = 0.97 \pm 0.01; Mean \pm STD$; Fig.10). These results support the view that a unified or abstract representation of discrete numbers had emerged in the language layer of the network that was used to solve several counting-related tasks.

Discussion

To better understand potential mechanisms underlying the development of the concepts of counting and number, we analyzed the internal representations of a neural network that learned to solve multiple counting-related tasks. We found that the network developed specific representations for sev-

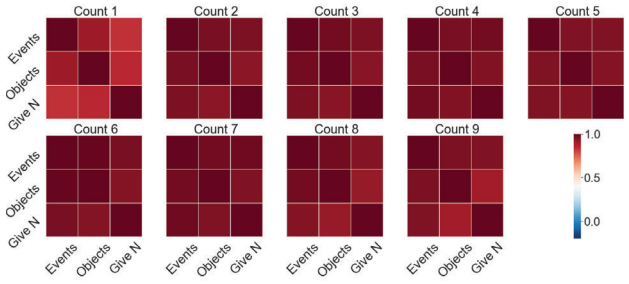


Figure 10: Similarity between representations in the language layer for the same number in different tasks as measured by pairwise correlations of node activation vectors.The Language representation of each number is the same for each task.

eral key components of counting, like the identification of an entity to count, the establishment of one-to-one correspondence between entities and number words, and the number of entities counted. These representations were highly similar between different counting tasks, suggesting that the network's knowledge about and representation of the counting procedure was shared across all of the tasks. In particular, the representation for the event of a given ordinality (first, second, third) was shared across the different task settings, suggesting that the network had acquired an abstract concept of number, independent of the particular entities being counted.

The identified entity and number nodes distinguish perceptually equivalent situations. We highlight two examples. First, when solving the Count-all-Objects task the entity nodes distinguish situations in which the agent's hand is situated 'on' an object when the object is yet to be counted from situations in which the object has been counted already and the agent is supposed to move on to the next countable object. Second, for the identified node activity patterns representing the number words, e.g. the second occurring event in the Count-all-Events task is perceptually equivalent to the third event, yet the language layer represents them differently. Even though not quantified in this work, these findings are in line with the work in Marstaller et al. (2013) where representation has been defined as 'that part of the shared entropy between environment states and internal states that goes beyond what is seen in the sensors'. Similarly, Haugeland (2013) understands representation as something that 'stands in' for specific features or aspects of the environment, even if these are currently not reflected in the perceptual system of the agent. In the neural network agent presented in this work, the representation corresponds to the activity pattern that 'stands in for' the number of items counted, which in itself is not present in the perceptual input.

The analysis we have described demonstrates how abstract representations can emerge for situations within our training regime. We have not shown that the neural network can benefit from these representations by applying them to new situations. However, our previous work has shown that the speed of learning each task is facilitated by prior learning of other tasks, and this is consistent with the idea that representations established for one task become available for use in other tasks that rely on the same conceptual structure.

We hope that this work inspires further projects to explore the issue of generalization with the help of these initial observations.

Acknowledgements

The authors would like to thank members of the research group for mathematical cognition and literacy at NTNU; the members of the PDP lab at Stanford University; and participants in the Digital Reasoning project for productive discussions. We would like to thank Keith Downing for invaluable guidance and comments on this work.

This work was supported by the Research Council of Norway.

References

Clements, D. H. and Sarama, J. (2009). Learning trajectories in early mathematics–sequences of acquisition and teaching. *Encyc. of language and literacy development*, 7:1–6.

Davidson, K., Eng, K., and Barner, D. (2012). Does learning to count involve a semantic induction? *Cognition*, 123(1).

Fang, M., Zhou, Z., Chen, S., and McClelland, J. (2018). Can a recurrent neural network learn to count things? In *Proceedings of the Meeting of the Cognitive Science Society*.

Gelman, R. and Gallistel, C. (1978). The child's concept of number. *Cambridge, MA: Harvard*.

Gunderson, E. A. and Levine, S. C. (2011). Some types of parent number talk count more than others: relations between parents' input and children's cardinal-number knowledge. *Developmental science*, 14(5):1021–1032.

Haugeland, J. (2013). Representational genera. In *Philosophy and connectionist theory*, pages 75–104. Psychology Press.

Hinton, G. E. and Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological review*, 98(1):74.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Marstaller, L., Hintze, A., and Adami, C. (2013). The evolution of representation in simple cognitive networks. *Neural computation*, 25(8):2079–2107.

Piantadosi, S. T., Tenenbaum, J. B., and Goodman, N. D. (2012). Bootstrapping in a language of thought: A formal model of conceptual change in number word learning.

Rogers, T. T. and McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive science*, 38(6):1024–1077.

Rumelhart, D. E. (1986). Parallel distributed processing: Explorations in the microstructure of cognition. *Learning internal representations by error propagation*, 1:318–362.

Sabathiel, S., McClelland, J. L., and Solstad, T. (AfP). A computational model of learning to count in a multimodal, interactive environment. In *Proceedings of the Meeting of the Cognitive Science Society*. Accepted for publication, 2020.

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

Spelke, E. S., Breinlinger, K., Macomber, J., and Jacobson, K. (1992). Origins of knowledge. *Psychological review*, 99(4):605.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network. In *Advances in neural information processing systems*, pages 802–810.