Can Generic Neural Networks Estimate Numerosity Like Humans?

[†]Sharon Y. Chen (syc2138@columbia.edu)[1], [†]Zhenglong Zhou (zzhou34@jhu.edu)[2], Mengting Fang (mtfang@mail.bnu.edu.cn)[3], and James L. McClelland (jlmcc@stanford.edu)[4]

[1]Department of Computer Science, Columbia University, New York, NY, 10027
[2]Department of Cognitive Science, Johns Hopkins University, Baltimore, MD, 21218
[3]Department of Mathematics, Beijing Normal University, Beijing, China
[4]Department Psychology, Stanford University, Stanford, CA, 94305

Abstract

Researchers exploring mathematical abilities have proposed that humans and animals possess an approximate number system (ANS) that enables them to estimate numerosities in visual displays. Experimental data shows that estimation responses exhibit a constant coefficient of variation (CV: ratio of variability of the estimates to their mean) for numerosities larger than four, and a constant CV has been taken as a signature characteristic of the innate ANS. For numerosities up to four, however, humans often produce error-free responses, suggesting the presence of estimation mechanisms distinct from the ANS specialized for this 'subitizing range'. We explored whether a constant CV might arise from learning in generic neural networks using widely-used neural network learning procedures. We find that our networks exhibit a flat CV for numerosities larger than 4, but do not do so robustly for smaller numerosities. Our findings are consistent with the idea that estimation for numbers larger than 4 may not require innate specialization for number, while also supporting the view that a process different from the one we model may underlie estimation responses for the smallest numbers.

Keywords: mathematical cognition; numerical cognition; neural networks; development; learning

[†]SYC and ZZ made equal contributions to this work.

Introduction

It is widely accepted that intuition (implicit knowledge) plays a strong role in mathematics. Here, and in a companion article (Fang, Zhou, Chen & McClelland, 2018) we consider alternatives to the widely held view (Feigenson, Dehaene & Spelke, 2004) that emphasizes innate, specialized systems as the foundation for these intuitions. Here, we focus on one putative system of this type, 'the approximate number system' (ANS), and explore the role of experience, rather than innate specialization, in shaping basic numerical abilities. Our work explores this possibility using neural network models and considers how the architecture and the details of the materials used to train and test the networks affect the outcome of learning and its developmental course.

Neural networks with very particular properties can arise during pre-natal development, shaped by genetically orchestrated processes, and it is possible to wire up a neural network that expressly exhibits properties attributed to the ANS (Dehaene & Changeux, 1993). We would not exclude the possibility that specialized pre-wiring may play a role in numerosity processing. However, it may still be worthwhile to explore neural networks lacking any specific specialization for number, similar to those that have been used to model a wide range of perceptual and cognitive processes, in both cognitive science (Rogers & McClelland, 2014) and artificial intelligence (LeCun, Bengio, & Hinton, 2015).

Recently, Stoianov & Zorzi (2012) pioneered an approach of the kind we are pursuing, applying it to numerosity comparison, in which a human observer views a pair of arrays and determines which one contains the larger number of separate items. The ability to make such comparisons appears to be present in non-human animals, newborn infants, and adults from primitive cultures that lack exact number words (Pica *et al.,* 2004). However, Stoianov and Zorzi found that a generic deep neural network can learn to perform the numerosity comparison task, using numerosity sensitive representations acquired through unsupervised learning from displays containing blobs varying in size and number. In work leading up to our present effort, Zou, Testolin & McClelland (in preparation) established that, prior to any learning, the representations in such networks can support numerosity discrimination at levels better than human neonates, and that unsupervised learning can then tune such representations. The gradual developmental refinement in numerosity acuity follows the power law pattern seen in human development (Odic *et al.,* 2015).

Here, we focus on another important aspect of numerical cognition: numerosity estimation. We choose this ability because it appears to depend on experience with number: Children's numerosity estimates can be uncorrelated with the actual number of presented items even for children who can perform counting tasks correctly, and the ability to estimate improves as children's other number abilities improve with experience (Davidson, Eng & Barner, 2012). Estimation, like numerosity discrimination, is *approximate,* in that for numerosities (N) larger than 3 or 4, estimates exhibit variability. This variability increases with N, such that the coefficient of variation (CV: ratio of the standard deviation to the mean estimate) is approximately constant (Revkin *et al*, 2008; Izard & Dehaene, 2008). Animals (Platt & Johnson, 1971) and humans (Whalen, Gallistel & Gelman, 1991) also exhibit an approximately constant coefficient of variation in tasks requiring the production of a target number of responses (for humans, brief presentation and pressure to respond quickly are used to prevent exact counting).

The present work focuses on estimation of small numerosities in visually-presented arrays, where educated human adults produce estimates whose mean value matches

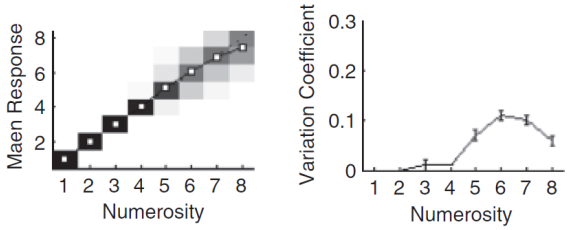

Figure 1. Mean response for different numerosities (Left) and coefficient of variation (right) from the experiment of Revkin et al (2008).

the presented numerosity N (Revkin et al, 2008; Figure 1). Their participants were virtually perfect for N up to 4, and showed CV's varying in a narrow range over the numbers 5 to 8 (a slight distortion in both the mean estimate and the CV for N=8 arises from the fact that the response options were restricted to the range from 1-8). The authors concluded that a distinct 'subitizing' mechanism is responsible for responses for N=4 or less, but that the ANS is responsible for responses for N greater than or equal to 5.

As we shall see, we find that, with sufficient training, neural networks with very different architectures trained and tested with varied materials can estimate numerosity with human level accuracy, exhibiting an approximately constant coefficient of variation across numerosities greater than or equal to 5. Our work supports the view that approximate number representation and processing may not require an innate system specialized for number; it may instead be a robust characteristic of relatively generic neural networks trained using generic training procedures.

Below we present two experiments. Experiment 1 explored estimation in both a complex contemporary network architecture designed for visual search and object identification and in a simpler, more standard, feed-forward architecture. We chose the former, the Differentiable Recurrent Attention Model (DRAM) of Gregor, Danihelka, Graves, Rezende, Wierstra (2015), as a first step toward a broader examination of number learning, including counting (see Fang *et al.*, 2018). The architecture has an attentional window similar to a camera that it can learn to move and zoom in and out, allowing it to process an input over a series of glimpses and to produce a sequence of responses, such as the names of the objects in the display. We chose the second, simpler and more generic architecture called the Feed-Forward (FF) network because we found that the complex features of the DRAM model (though important for counting, see Fang *et al*, 2018) were not used in estimation. For instance, the focus of attention remained constant over glimpses, and the accuracy of estimates was almost as good on the first as on the last glimpse. As we shall see, both networks exhibited a constant CV after training over numerosities greater than 4, capturing this signature feature of human and animal numerosity estimates.

Experiment 2 explored the effects of variation in the frequency of presentation of different numerosities during training and in the sizes of items in displays of different numerosities, using the FF network. The consideration of presentation frequency was motivated by findings and proposals of Piantadosi (2016), who observed that the frequency of occurrence of a numerosity in natural text decreases with N approximately in accordance with the function $f(N) \propto 1/(N^2)$. He further suggested that a constant CV independent of N arises as a rational consequence of the $1/(N^2)$ frequency distribution (the essence of the idea is that it is better to allow more variability in estimates of numbers that are used less frequently). We used this frequency distribution in our training materials for experiment 1. By comparing the outcome of training the network with the decreasing frequency distribution to a flat frequency distribution ($f(N)$ the same for all N) we tested whether the fact that our networks in Expt 1. exhibited a constant CV depended on the use of the $1/(N^2)$ frequency distribution. Importantly, as we shall see, a constant CV was observed after training with either a flat or a decreasing frequency distribution, indicating that a decreasing frequency distribution is not necessary to observe this signature property in numerosity estimation.

Expt. 2 also explored the effect of varying the relationship between N and the average area occupied by each item in the display, both during network training and testing. In Experiment 1, item area did not vary with N, either in training or testing, but area does tend to decrease with N in natural images (Zou *et al,* in preparation), and experimenters vary size inversely with N in a subset of their test stimuli to ensure numerosity estimates are not based on a simple estimate of total occupied area. Expt. 2 therefore explored whether a constant CV for numerosities larger than 4 would still be observed with training and testing materials in which item area decreased with increasing numerosity.

Materials and Methods

The DRAM model. We used a version of the Differentiable Recurrent Attention Model (DRAM, Gregor et al., 2015) implemented in TensorFlow (https://www.tensorflow.org/). The network is a recurrent neural network that can learn to direct its attention while attempting to estimate the number of dots in the display over a series of nine glimpses (the computational graph for one glimpse is shown in Figure 2A). As in Gregor *et al*, the network contains a selective attention mechanism, implemented by the *read* module, and a pair of LSTM modules (an *encoder* and a *decoder* LSTM, each containing 256 units). A linear bottleneck layer (100 units, simplified from Gregor *et al.*, designated as *z* in the figure) lies between the two LSTMs. At each glimpse *t*, the output from the *decoder* LSTM at glimpse *t-1* specifies the center position, spacing, and filter width of an 10x10 grid of Gaussian filters (the initial specification at $t = 0$ is determined by learned bias weights) used by the *read* operation, which convolves the image with the Gaussian filters. The filtered image is fed upward through the network as shown in Figure 2A. Each arrow corresponds to a *linear* mapping (multiplication with a matrix of modifiable connection

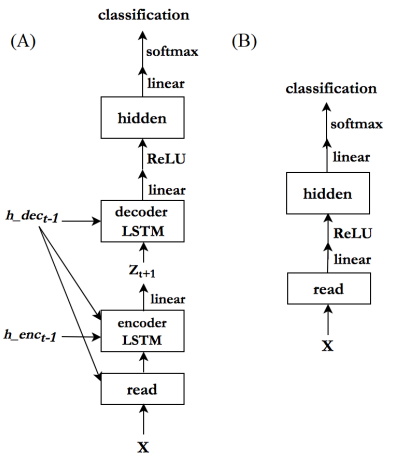

Figure 2. A: One glimpse from the DRAM network. B: the Feed-Forward Network. See text for details.

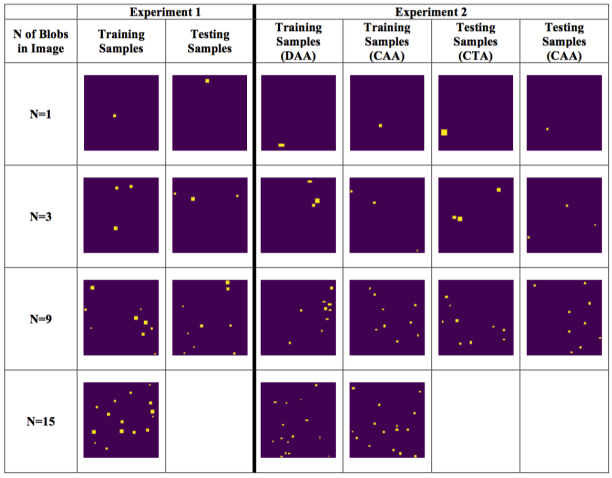

Figure 3. Examples of the stimuli used in the experiments.

weights plus biases), and corresponding weights are re-used across glimpses. The *encoder* and *decoder* LSTMs also receive their output at glimpse *t-1* as part of their input at glimpse *t*. The output of the *decoder* LSTM propagates through a linear mapping and a ReLU non-linearity to another hidden layer (256 units). Finally, the hidden layer output passes through another linear mapping followed by a *softmax* computation, producing output activations across a set of output units corresponding to numerosity responses ranging from 1 to 15. The unit with the largest activation at the last glimpse is the estimate of the number of items in the image.

The Feed-Forward Model. Our FF network has one hidden layer and uses a simplified *read* operation. The *read* operation performs filtering of the image based on a set of 15x15 Gaussian filters centered on the image and spaced to cover it completely. The filtered output of the *read* operation is then converted into a hidden layer (250 units) by the *linear* operation followed by a ReLU. A linear operation followed by a softmax determines the activations of the output units. The unit with the largest activation is treated as the network's numerosity estimate. Source code for both models is available at https://github.com/numberlearning/Estimation.)

Learning. Weights were modified based on the gradient of the cross-entropy loss computed by comparing the activations at the output of the softmax computation to a target pattern specifying the number of blobs in the input image. In DRAM, the gradient is computed at every glimpse and propagated backward through each arrow in the full 9-glimpse network graph. Learning occurs over a series of iterations. In each iteration, gradients are calculated over 100 patterns sampled with replacement from the training environment, then weights are updated using the Adam optimizer, with initial learning rate of .01 and *epsilon* =1.

Training and Test Data and Procedures. For both experiments, training and test images were 100x100 pixels, initialized to RGB value 0 (black). In each image, a certain number of blobs were generated. Each blob was a square of pixels of RGB value 255 (white). For training, each image contained a number N of blobs ranging from 1 to 15. A new set of 10,000 training images was generated after every 1,000 training iterations to prevent overfitting, and a new set of test images (1,000 for each N) was generated each time a network was tested. Networks were tested at pre-specified intervals as training progressed. During testing, input N ranged from 1 to 9, though responses could still range from 1 to 15. This prevented distortion of responses for larger numerosities in the test range (1-9) since responses both above and below this value were allowed. Gradient computations and weight updates did not occur during testing.

Experiment 1 compared the two architectures. We used 10 independent runs of the DRAM network (trained for 2,000,000 iterations) and of the FF network (trained for 3,000,000 iterations) to assess the final estimation performance and learning trajectory of each type of network. Each run used different random initial weights and different randomly generated training and testing patterns.

*Blob size and numerosity specifications***.** The edge length of each blob was randomly chosen from {2,3,4,5} independently for each blob, regardless of the number of blobs in the image. Blobs were placed randomly with the restriction that they could not overlap or touch. The number of images in each randomly generated set of training images containing N blobs was $10{,}000/N^2$ rounded to the nearest integer.

Experiment 2 investigated effects of varying aspects of the training and testing materials in the FF network, employing 10 independent runs in each of four conditions (40 runs total). Each condition combined one of two numerosity-frequency distributions with one of two numerosity-area distributions. The two numerosity frequency distributions were $f(N) \propto 1/(N^2)$, in which the number of images containing N blobs in each set of training images was $(10{,}000/N^2)$, rounded to the nearest integer and $f(N)$ constant, in which the number of images in each training set containing N blobs was 1000. The two types of numerosity-area distributions used

decreasing average area and constant average area. For the former, we approximated the decrease in object size with numerosity in a natural image data set (Zou *et al,* in preparation). A scale factor s(N) was chosen randomly from the interval a(N)*[0.8,1.2] for each blob, where a(N) decreased with N according to $a(N) = 2/(N^{0.3154})$. The width and height of each blob were determined by sampling a width variable k_1*s(N) and a height variable k_2*s(N); k_1 and k_2 were chosen independently from the range {2,3,4}, allowing blobs to be rectangles of different shapes. Values were then rounded probabilistically up or down so that the height and width of a blob was always an integer. In the constant average area condition, the size of each blob, independent of N, was chosen from the interval 3*[0.8, 1.2]. This ensured that the average blob area was approximately constant across different N. Probabilistic rounding resulted in square blobs whose size could range from 2x2 to 4x4.

For testing, as in Experiment 1, 1000 images were created for each N each time a network was tested. We followed procedures used in Revkin *et al* (2008) and in Izard and Dehaene (2008) to ensure that numerosity responses were based on the number of items in the display, rather than the total occupied area or the average area of items in the display. For each N, half of the test images had approximately constant *total* area (CTA), and the second half had approximately constant *average* area (CAA) occupied by individual blobs. Specifically, the total area occupied by blobs was approximately 81 for the CTA images regardless of N (so that individual blob size *decreased* with N), while the average area occupied by individual blobs was approximately constant at 9, independently of N in the CAA images (so that total blob area *increased* with N).

Results

Analysis Approach. We present results averaged over the 10 runs of each condition (DRAM vs FF in Experiment 1; training frequency and area conditions for Experiment 2). For each condition, we sought to quantify the relationship, displayed in Figure 4, between the true numerosity N (from 1-9) and both the mean numerosity estimate and the coefficient of variation (CV) of these estimates. For the mean estimates, the figure presents means and 95% confidence intervals over the 10 networks in each condition for the slopes (S), intercepts (I), and regression (R) of the relationship between the mean estimate and the true value of N. Similarly, for the relationship between the CV and the value of N, Figure 4 displays the slope, intercept, and regression values over values of N ranging from 5 to 9. To match the results from Revkin *et al.,* the slope for the mean estimates should be 1 and the intercept should be 0; the slope for the CV over the range 5-9 should be about 0 and the intercept (corresponding to the constant CV) should be a small number, close to 0.1. For experiment 2, we concentrate on results from the training conditions in which blob area decreased with N; results were similar when blob area during training did not vary with N, except where noted.

Experiment 1. *Final performance.* The final performance of our DRAM network (D) and our FF (F) network were both consistent with the results of Revkin *et al,* as shown in the top left panels for Figure 4. For both networks, mean estimates corresponded nearly perfectly to the true numerosity for all N, with slopes close to 1 and intercepts close to 0 (S_D=.979, S_F =.9904, I_D=0.02, I_F =0.0368; see Figure for confidence intervals). The coefficient of variation (CV) for each of the two networks was near 0 for displays with 1 blob and increased as numerosity increased. The CV flattened for numerosities over 4, with slopes approximately 0 in both networks (S_D =-0.0007, S_F =0.0031). For displays with 5 or more blobs, the CV for the FF network (I=0.0977), was slightly smaller than that of the DRAM network (I=0.1469).

Development. Both the DRAM and the FF network initially underestimated the number of items presented, and both showed decrease in the CV over the course of training (top right panels of Figure 4). However, the FF network learned faster and reached a lower CV, and showed a compression of the numerosity range, such that spacing between numbers decreases as N increases. Such a pattern is often seen in human estimation data, especially without calibration or feedback (Isard & Dehaene, 2008). The CV improved much more quickly for FF (note the factor of 10 scaling of iteration numbers chosen for presentation). Also, DRAM initially exhibited a linear relationship between mean estimate and N with a slope less than 1, while the FF network initially produced a nonlinear relationship that fit a power function with an exponent of less than 1 better than a linear function (for each run, R was slightly larger for the non-linear than the linear fit).

Experiment 2. Many interesting findings emerged from this study. We focus on three key findings.

Effect of area variation with N during testing. In all four conditions, we found that our FF model was sensitive to numerosity rather than a possible confounded area variable when tested at the end of training. Mean estimates of numerosity were completely unaffected by whether total area (CTA) or average area (CAA) stayed constant in the test stimuli. For example, for both the $f(N) \propto 1/(N^2)$ and $f(N)$ constant conditions, the mean estimates for CTA and CAA test stimuli lie on top of each other (see the 3rd and 4th rows, 1st column of Figure 4). Even early in development (3rd and 4th rows, 3rd column) the difference between CTA and CAA test stimuli makes very little difference to the mean estimates, though there are effects on the CV, especially for N = 1 early in development (3rd and 4th row, 4th column).

Effect of numerosity frequency distribution. To address the role of the numerosity frequency distribution during training, we first consider the CV slopes for numbers from 5-9. Surprisingly, the CV was strikingly constant over the range from 5-9 in the $f(N)$ constant conditions (S = .0014 ± .0008) as well as in the $f(N) \propto 1/(N^2)$ conditions (S = .0059 ± .001), indicating that the CV at the end of training is nearly constant for all four combinations of frequency distribution and area distribution. It thus appears that a decreasing numerosity

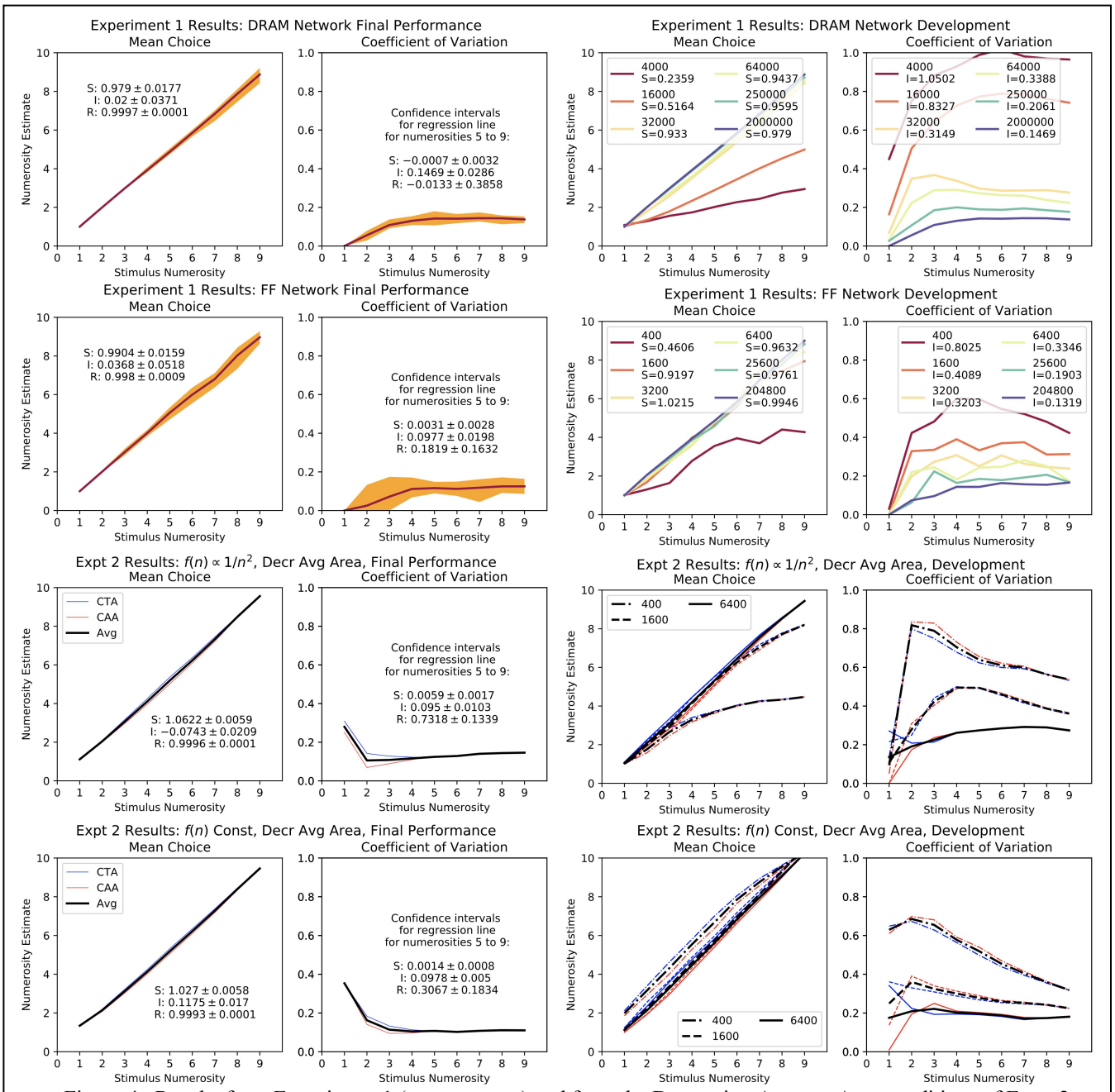

Figure 4. Results from Experiment 1 (top two rows) and from the Decreasing Average Area conditions of Expt. 2 (bottom 2 rows). Yellow bands signify range of values observed across runs. Numbers in legends for developmental results (right panels) correspond to iteration numbers. Abbreviations: S: Slope; I: Intercept; R: Regression coefficient (R); CTA: Constant total area test items; CAA: Constant Average Area Test items.

frequency function is not necessary to achieve a flat CV over this range in our feed-forward neural network.

That said, the numerosity frequency distribution did have a striking effect early in development. For the $f(N) \propto 1/(N^2)$ condition (row 3, column 3), there is a strong tendency to underestimate larger numerosities early during training, and the estimates appear to follow a logarithmic or power function with an exponent less than 1. On the other hand, early in training with a constant frequency distribution (row 4, col 3 of the figure) mean estimates tend to be too high, and increase almost linearly with N.

Effect of numerosity area distribution. The results already described indicate that the numerosity area distribution had little effect on the CV for numerosities greater than 4: However, the numerosity area distribution during training had a striking effect on the CV for small numerosities. Indeed, for N = 1, a constant numerosity area distribution during training (Experiment 1, top panels, and the constant average area training condition of Experiment 2, not shown)

led to a CV near 0, while a decreasing average area distribution led to a larger CV for N =1 than for any other numerosity. A similar tendency appears for N = 2, but it fades out for larger numerosities.

Discussion

In this work, we explored whether generic neural networks can exhibit characteristics of human number estimation through development. We trained and evaluated two distinct neural networks on datasets in which we manipulated relationships between numerosity and frequency and between numerosity and area. Our results demonstrated that generic neural networks can learn to estimate numerosities accurately, regardless of these factors. In particular, the fact that they estimated well for numerosities that were relatively infrequent in the training data (e.g., for N = 9, when $f(N) \propto 1/(N^2)$), and for the constant total area testing data, for which successful estimation based on area is impossible, showcased the robustness of their acquired numerical estimation ability. After sufficient development, across all combinations of training and testing sets, our networks' performance on the estimation task displayed a signature characteristic of the human approximate number system: constant CV for numerosities over 4. Our networks' acquisition of human-like estimations for numerosities in this range supports the idea that human estimation abilities may not require an innate approximate number system.

The finding that network performance eventually reached a stage in which its CV curve displayed the same levelling off for higher numerosities and independence from the influence of these extraneous factors has relevance for recent theories about the possible reason why numerosity judgments exhibit a constant CV. For instance, in Experiment 2, later in training, we obtained flat CV curves for higher numerosities even for networks trained with data in which all numerosities occurred with equal frequency. This suggests that the proposal of Piantadosi (2016), that there is less need for precision with less frequent numerosities, may not be the only way to explain why variability increases with N in numerosity estimation. In future work we hope to explore the idea, suggested by our developmental findings, that Piantadosi's suggestions may be relevant to estimation early in development, but experience may eventually level the playing field, an effect that we hope to explore in subsequent investigations.

It should be noted that our generic neural networks did not capture the human data for numerosities from 1 to 4. For these smaller numerosities, humans estimated the numerosity almost perfectly, with a CV close to 0, whereas we only observed such a low CV for N = 1, and then only in some conditions. Revkin *et al* (2008) and others have proposed that a distinct 'subitizing' mechanism, distinguished from the ANS, is responsible for estimating these numerosities. Our findings can be interpreted as falling in line with this possibility.

The possible 'subitizing' mechanisms that could explain human expertise in estimating smaller numerosities include the object individuation system (Feigenson *et al.,* 2004), in which humans are thought to be able to hold up to three objects in mind simultaneously. Another way these findings might be explained would be in terms of the idea that small numerosities give rise to distinctive emergent shapes, such as 'point', 'line', and 'triangle'. In future work, we hope to explore these possibilities.

References

Davidson, K., Eng, K. & Barner, D. (2012). Does learning to count involve a semantic induction? *Cognition*, *123*(1), 162-173.

Dehaene, S., & Changeux, J. P. (1993). Development of elementary numerical abilities: A neuronal model. *Journal of cognitive neuroscience*, *5*(4), 390-407.

Dehaene, S., & Mehler, J. (1992). Cross-linguistic regularities in the frequency of number words. *Cognition*, *43*(1), 1-29.

Feigenson, L., Dehaene, S., & Spelke, E. (2004). Core systems of number. *Trends in Cogn. Sci.*, *8*(7), 307-314.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

Izard, V., & Dehaene, S. (2008). Calibrating the mental number line. *Cognition*, *106*(3), 1221-1247.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Odic, D., Libertus, M. E., Feigenson, L., & Halberda, J. (2013). Developmental change in the acuity of approximate number and area representations. *Developmental psychology*, *49*(6), 1103.

Platt, J. R., & Johnson, D. M. (1971). Localization of position within a homogeneous behavior chain: Effects of error contingencies. *Learning and Motivation*, *2*(4), 386-414.

Revkin, S. K., Piazza, M., Izard, V., Cohen, L., & Dehaene, S. (2008). Does subitizing reflect numerical estimation?. *Psychological science*, *19*(6), 607-614.

Rumelhart, D. E., & McClelland, J. L. (1986). *Psychological and Biological Models* (Vol. 2). MIT Press.

Piantadosi, S. T. (2016). A rational analysis of the approximate number system. Psychonomic Bulletin & Review, 23(3), 877-886.

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

Stoianov, I., & Zorzi, M. (2012). Emergence of a'visual number sense'in hierarchical generative models. *Nature neuroscience*, *15*(2), 194.

Whalen, J., Gallistel, C. R., & Gelman, R. (1999). Nonverbal counting in humans: The psychophysics of number representation. *Psychological Science*, *10*(2), 130-137.

Zou, W., Testolin, A., & McClelland, J. L. (In preparation). Initial Competence and Developmental Refinement of a Sense of Number in a Deep Neural Network.