COGNITIVE NEUROSCIENCE

Number detectors spontaneously emerge in a deep neural network designed for visual object recognition

Khaled Nasr*, Pooja Viswanathan†, Andreas Nieder‡

Humans and animals have a "number sense," an innate capability to intuitively assess the number of visual items in a set, its numerosity. This capability implies that mechanisms to extract numerosity indwell the brain's visual system, which is primarily concerned with visual object recognition. Here, we show that network units tuned to abstract numerosity, and therefore reminiscent of real number neurons, spontaneously emerge in a biologically inspired deep neural network that was merely trained on visual object recognition. These numerosity-tuned units underlay the network's number discrimination performance that showed all the characteristics of human and animal number discriminations as predicted by the Weber-Fechner law. These findings explain the spontaneous emergence of the number sense based on mechanisms inherent to the visual system.

INTRODUCTION

Humans and animals have a "number sense," an innate capability to intuitively assess the number of visual items in a set, its "numerosity" (*1*, *2*). This capacity allows newborn human infants (*3*) and animals (*4*) to assess the number of items in a visual scene. Human psychophysics (*5*, *6*), brain imaging studies in humans (*7*, *8*), and single-neuron recordings in animals support the direct and automatic assessment of numerosity in the brain. In animals that had not been trained to judge number, single neurons spontaneously responded to numerosity and were tuned to preferred numerosities (*9*, *10*). These "number neurons" that also exist in the human brain (*11*) are regarded as the neuronal foundation of numerical information processing (*12*).

The innate presence of the number sense implies that mechanisms to extract numerosity indwell the brain's visual system, although it is by nature primarily concerned with visual objects. In recent years, biologically inspired deep neural networks have provided valuable insights into the workings of the visual system. Generative neural networks, a class of deep networks that learn to form an internal model of the sensory input, have been shown to become sensitive to numerosity but could not explain the emergence of real number neurons (*13*). Here, we use a hierarchical convolutional neural network (HCNN), a class of biologically inspired models that have recently achieved great success in computer vision applications (*14*, *15*) and in the modeling of the ventral visual stream (*16*, *17*). Like the brain, these models comprise several feedforward and retinotopically organized layers containing individual network units that mimic different types of visual neurons. The training procedure autonomously determines selectivity for individual features in each unit to maximize the network's performance on a given task. Here, we built such a network and trained it on a visual object recognition task unrelated to numbers to explore whether and how sensitivity to numbers would spontaneously emerge.

Animal Physiology Unit, Institute of Neurobiology, Auf der Morgenstelle 28, University of Tübingen, 72076 Tübingen, Germany.
*Present address: Clinical Neurotechnology Lab, Charité–Berlin University of Medicine, Charitéplatz 1, 10117 Berlin, Germany.
†Present address: Laboratory of Neural Systems, The Rockefeller University, 1230 York Avenue, New York, NY 10065, USA.
‡Corresponding author. Email: andreas.nieder@uni-tuebingen.de

RESULTS

Numerosity selectivity spontaneously emerges in a deep neural network trained for object classification

We trained a deep neural network to classify objects in natural images. The network model was an instance of HCNNs (*18*), originally inspired by the discovery of simple and complex cells in early visual cortex (*19*). The network model (Fig. 1A and Table 1; see Materials and Methods for details) can be conceptually divided into two parts: a feature extraction network that learned to convert natural images into a high-level representation suitable for object classification and a classification network that produced object-class probabilities based on this representation. The network consisted mainly of convolutional layers and pooling layers. Network units in convolutional layers performed local filtering operations analogous to simple cells in the visual cortex, while the units in pooling layers aggregated responses in local patches in their input, similar to complex cells. Network units that had the same receptive fields in convolutional layers competed with each other using a simple form of lateral inhibition (*14*).

We trained the network on object recognition using the ILSVRC2012 ImageNet dataset [(*14*); see Materials and Methods for details]. This dataset contains around 1.2 million images that have been classified into 1000 categories based on the most prominent object depicted in each image. After training, the network was tested on object classification with 50,000 new images that the network had never seen before. The network achieved a highly significant object classification accuracy of 49.9% (chance level = 0.1%; $P < 0.001$, binomial test) on this dataset. Figure 1B shows examples of the test images and the predictions made by the network.

To explore whether the network trained on object classification with natural images could spontaneously assess the number of items in dot displays (their numerosity), we investigated whether different numerosities elicit different activations in the network units. To that aim, we discarded the classification network and presented only the feature extraction network with newly generated images of dot patterns depicting various numerosities ranging from 1 to 30, following (*20*) for monkey experiments. Figure 2A shows examples of those images. To control for the effect that the visual appearance of the dot displays might have on unit activations, we used 21 images for each numerosity across three different stimulus sets. The first stimulus set (standard set) showed circular dots of random size and spacing. The second stimulus set (control set 1) displayed dots of equal total dot area and dot density

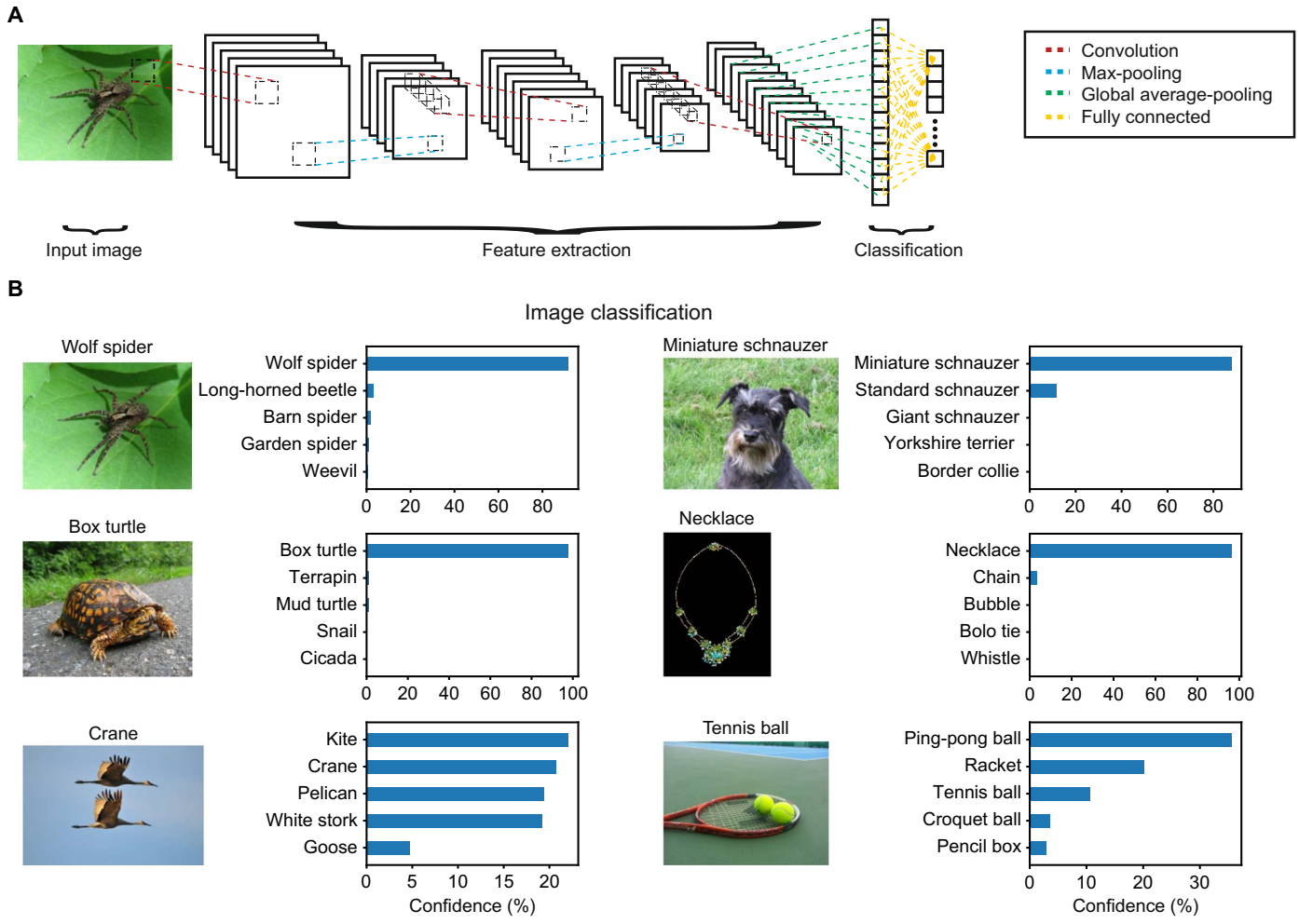

Fig. 1. An HCNN for object recognition. (**A**) Simplified architecture of the HCNN. The feature extraction network consists of convolutional layers that compute multiple feature maps. Each feature map represents the presence of a certain visual feature at all possible locations in the input and is computed by convolving the input with a filter and then applying a nonlinear activation function. Max-pooling layers aggregate responses by computing the maximum response in small nonoverlapping regions of their input. The classification network consists of a global average-pooling layer that computes the average response in each input feature map, and a fully connected layer where the response of each unit represents the probability that a specific object class is present in the input image. (**B**) Successful classification of a wolf spider by the network from other arthropods is shown as an example. Example images representative of those used in the test set and the top 5 predictions made by the network for each image ranked by confidence. Ground-truth labels are shown above each image. Images shown here are from the public domain (Wikimedia Commons).

across numerosities. The third stimulus set (control set 2) consisted of items of different geometric shapes with equal overall convex hull across numerosities (see Materials and Methods for details).

We presented a total of 336 images to the network and recorded the responses of the final layer. A two-way analysis of variance (ANOVA) with numerosity and stimulus set as factors was performed to detect network units selective to the number of items ($P < 0.01$) but without significant effects for stimulus set or interaction. Of the 37,632 network units in the final layer, 3601 (9.6%) were found to be numerosity-selective network units. The responses of numerosity-selective units exhibited a clear tuning pattern (Fig. 2B) that was virtually identical to those of real neurons [Fig. 2C; real neurons from (*20*)]: Each network unit responded maximally to a presented numerosity, its preferred numerosity, and progressively decreased its response as the presented numerosity deviated from the preferred numerosity. The distribution of preferred numerosities covered the entire range (1 to 30) of presented numerosities, with more network units preferring smaller than larger numerosities (Fig. 2D), similar to the distribution observed in real neurons (Fig. 2E) (*20*).

Tuning properties of numerosity-selective network units

If the numerosity-selective network units are analogous to numerosity-selective neurons found in the brain, then they should exhibit the same tuning properties. To investigate this, we averaged the responses from numerosity-selective network units that have the same preferred numerosity and normalized them to the 0 to 1 activation range to create the pooled network tuning curves (Fig. 3). The pooled network units' tuning curves revealed characteristics of real neurons (*12*): The shape of the units' tuning curves was asymmetric peak functions on a linear number scale, with more sharply decaying slopes toward smaller than larger numerosities. This pattern suggests that the network units' tuning was better represented on a nonlinearly compressed, possibly

Table 1. Description of the layers in the HCNN.

Role	Layer	Type	Number of feature maps	Spatial size	Kernel size
Feature extraction	0	Input image	3	224 × 224	–
	1	Convolutional	32	224 × 244	9 × 9
	2	Max-pooling	32	224 × 244	2 × 2
	3	Convolutional	48	112 × 112	9 × 9
	4	Max-pooling	48	112 × 112	2 × 2
	5	Convolutional	96	56 × 56	7 × 7
	6	Max-pooling	96	56 × 56	2 × 2
	7	Convolutional	192	28 × 28	5 × 5
	8	Max-pooling	192	28 × 28	2 × 2
	9	Convolutional	384	14 × 14	5 × 5
	10	Max-pooling	384	14 × 14	2 × 2
	11	Convolutional	768	7 × 7	5 × 5
	12	Convolutional	768	7 × 7	5 × 5
	13	Convolutional	768	7 × 7	5 × 5
Classification	14	Average-pooling	768	1 × 1	7 × 7
	15	Softmax classifier	1000	1 × 1	1 × 1

logarithmic, scale, where large numerosities occur closer together than small numerosities.

To verify this, we first plotted the pooled network tuning curves once on a linear scale and again on a logarithmic scale (Fig. 4A). The network tuning curves became more symmetric and had a near-constant tuning width across preferred numerosities on the logarithmic scale. To quantify this effect, we fit Gaussian functions to the network tuning curves plotted on a linear scale and on three different nonlinearly compressed scales, namely, two power scales and a logarithmic scale [$f(x) = x^{0.5}$, $f(x) = x^{0.33}$, $f(x) = \log_2(x)$]. These scales represent different levels of nonlinear compression such that the level of compression progressively increases when going from the linear scale to the logarithmic scale. The Gaussian function was chosen because it is a standard symmetric function. If a scale is suited to the tuning curves, they should become symmetric around preferred numerosities when plotted on that scale, and therefore, the goodness of fit (r^2 score) of the Gaussian function to the tuning curves should be increased (*21*). We found that the Gaussian function proved a significantly better fit for the data on any of the nonlinear scales than on the linear scale ($P < 0.05$, paired t test) (Fig. 4B). The goodness of fit was not significantly different between any of the nonlinear scales ($P > 0.05$). Furthermore, we plotted the SD of the Gaussian fit as a measure of the tuning curve width for each of the tuning curves against the preferred numerosity associated with each curve (Fig. 4C). The clear and positive slope of the Gaussian widths on the linear scale ($r = 0.96$, $P = 2.1 \times 10^{-9}$) indicated that tuning width systematically increased with numerosity. In contrast, the slope had values close to zero for the logarithmic scale ($r = 0.20$, $P = 0.47$), indicating that tuning widths were invariant with lognormal tuning curves of increasing numerosity.

Previous network models of number coding postulated summation units, units that monotonically increase or decrease responses with increasing numbers, either as necessary precursors to tuned number detectors (*22*, *23*) or as actual output units (*13*). In our network, however, summation units were negligible in both respects. In the output layer (layer 13), only 0.5% of all units were summation units, in stark contrast to 9.6% tuned number units. The preceding intermediate layers (layers 12 and 11) contained only 0.9 and 2.3% summation units, respectively. Crucially, when we eliminated the responses of all summation units before testing the model, the proportions of tuned neurons, their distribution, and average tuning curves were qualitatively unchanged. Therefore, summation units were not necessary for the network to develop number detectors.

Relevance of numerosity-selective network units to performance

We then investigated whether numerosity-selective network units would be sufficient to solve a matching task that required abstracting the absolute numerosity from the low-level visual features of the stimuli. For this purpose, we constructed a numerosity matching task (see Materials and Methods for details) that was comparable to the tasks developed for monkeys and humans (*24*). In each trial, the feature extraction network was presented with two images of dot patterns, and for each image, the responses of the numerosity-selective units in the final layer of the network were recorded. The responses of the selective units were fed into a small two-layer neural network, which was trained to identify whether the two images contained the same number of dots. The feature extraction network was fixed during training and was not allowed to adapt based on the labeled examples

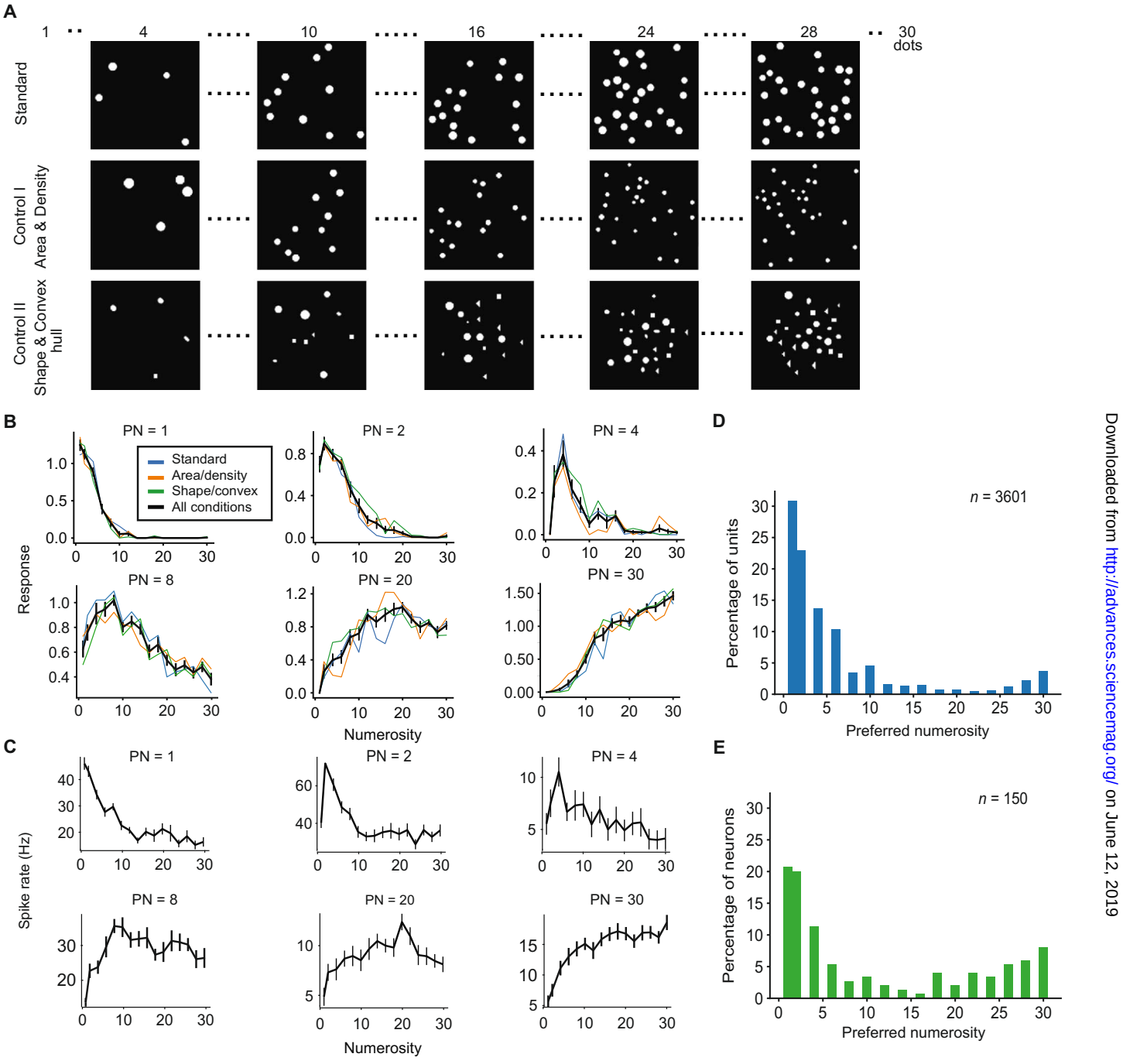

Fig. 2. Numerosity-tuned units emerging in the HCNN. (**A**) Examples of the stimuli used to assess numerosity encoding. Standard stimuli contain dots of the same average radius. Dots in Area & Density stimuli have a constant total area and density across all numerosities. Dots in Shape & Convex hull stimuli have random shapes and a uniform pentagon convex hull (for numerosities >4). (**B**) Tuning curves for individual numerosity-selective network units. Colored curves show the average responses for each stimulus set. Black curves show the average responses over all stimulus set. Error bars indicate SE measure. PN, preferred numerosity. (**C**) Same as (B), but for neurons in monkey prefrontal cortex (*20*). Only the average responses over all stimulus sets are shown. (**D**) Distribution of preferred numerosities of the numerosity-selective network units. (**E**) Same as (D), but for real neurons recorded in monkey prefrontal cortex [data from (*20*)].

and therefore remained at the numerosity-naïve stage. After training, we measured the accuracy of the network on pairs of dot images that were not used during training and found it to be 81% (chance level was 50%). The numerosity-selective network units allowed reliable numerosity discrimination.

If tuning of the network units to their respective preferred numerosities were performance relevant, the tuning quality is expected to suffer in cases where the network makes erroneous numerosity judgments. That is, performance success would correlate with whether the network units show maximal activity to their preferred numerosities;

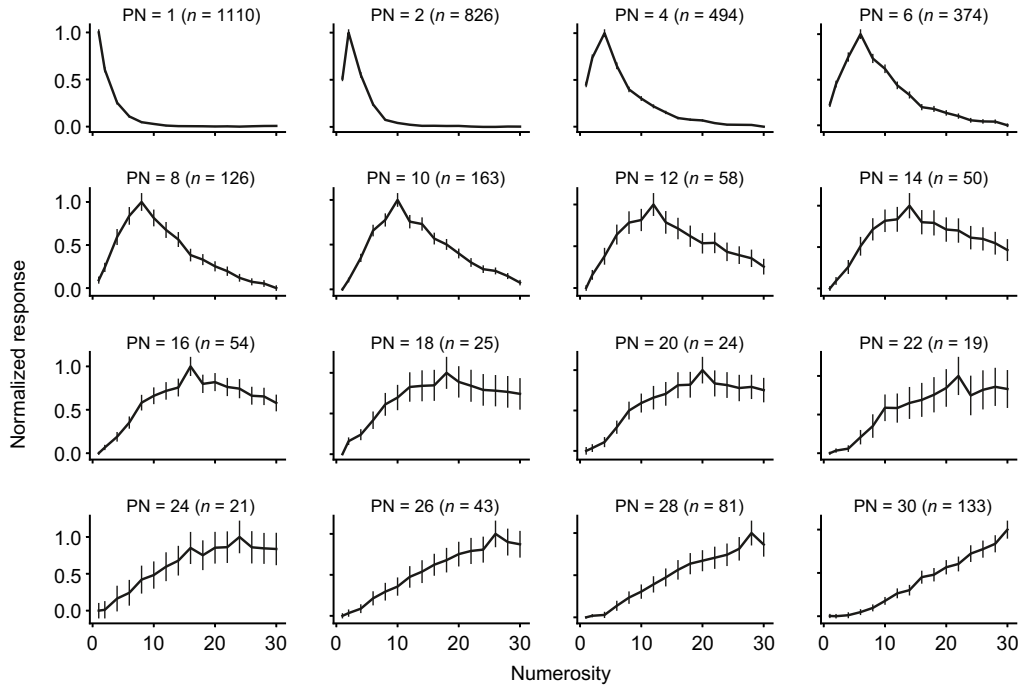

Fig. 3. Tuning curves of numerosity-selective network units. Average tuning curves of numerosity-selective network units tuned to each numerosity. Each curve is computed by averaging the responses of all numerosity-selective units that have the same preferred numerosity. The pooled responses are normalized to the 0 to 1 range. Preferred numerosity and number of numerosity-selective network units are indicated above each curve. Error bars indicate SE measure.

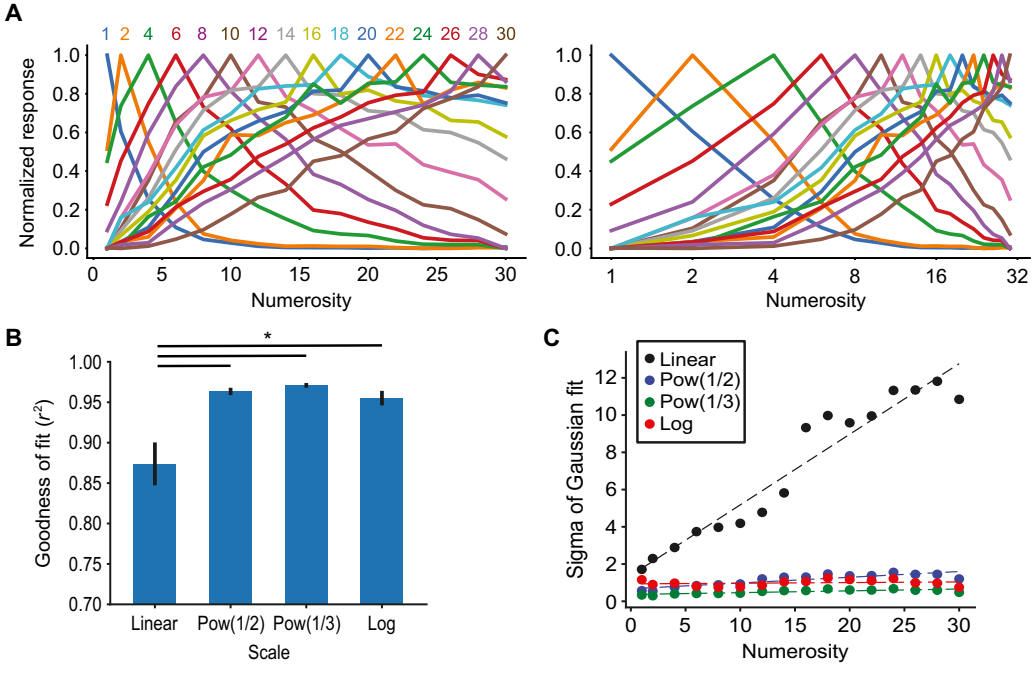

Fig. 4. Tuning properties of numerosity-selective network units. (**A**) Left: Average tuning curves for network units preferring each numerosity plotted on a linear scale. Right: Same tuning curves plotted on a logarithmic scale. (**B**) Average goodness-of-fit measure for fitting Gaussian functions to the tuning curves on different scales [$P_{\text{linear-log}} = 0.009$; $P_{\text{linear-pow(1/2)}} = 0.003$; $P_{\text{linear-pow(1/3)}} = 0.001$]. (**C**) SD of the best-fitting Gaussian function for each of the tuning curves of numerosity-selective network units for different scales.

if activation to the preferred numerosity is decreased, the network would be error prone. We therefore compared the tuning of the network numerosity units between correct and error trials. To that aim, we plotted the normalized unit responses in correct and error trials as a function of numerical distance from the preferred numerosity (Fig. 5A). We found that, in error trials, the average response to the preferred numerosity significantly dropped to 91% of that in correct trials. The tuning of the units was significantly worse during error

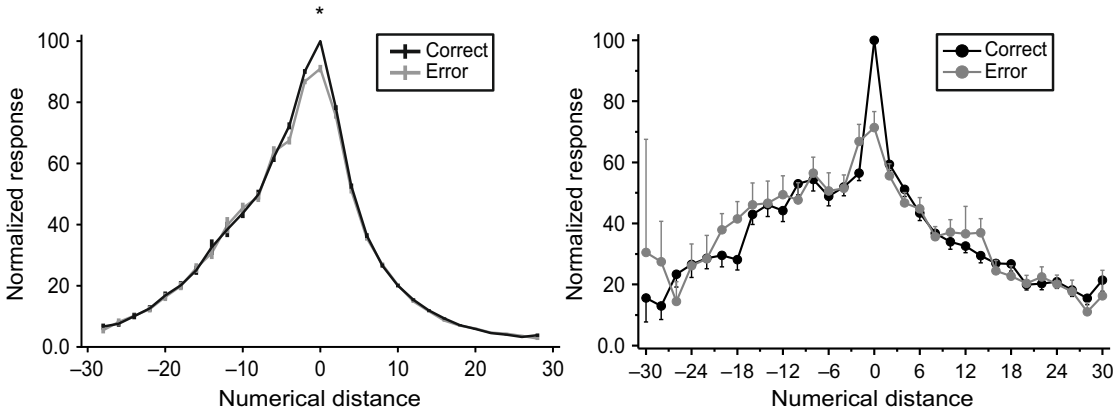

Fig. 5. Relevance of numerosity-selective units to network performance. Average activity of numerosity-selective network units shown as a function of numerical distance between preferred numerosities and sample numerosities in the matching task. Left: Data from network units. Responses were average separately for correct trials (black) and error trials (gray). Responses during error trials are normalized to the maximum average response during correct trials ($P = 0.019$). Right: Same plot but for real neurons recorded from monkey prefrontal cortex [data from (*20*)].

trials, causing the network to judge numerosity wrongly. The very same effect has also been observed for tuning curves in monkeys (Fig. 5B) (*20*).

Moreover, removing the summation units in the network hardly affected the network's number discrimination performance. After removing the summation units from the last three layers (layers 11 to 13), accuracy in network performance dropped only mildly, from 81 to 77%. Note that a mild reduction in accuracy was expected, because the removed summation units in the final layer were also classified as numerosity-selective units tuned to 1 and 30 and were thus part of the input to the numerosity matching network. The maintenance of accurate matching performance without summation units demonstrates that the network's performance did not depend on the summation units, even the ones in the output layers.

To further confirm the relevance of the tuned numerosity-selective units for the network's performance, we investigated whether the matching by the network showed the same characteristics as the behavioral performance functions of humans and animals. The first important characteristic is the "numerical distance effect," the observation that behavioral discrimination is progressively enhanced as numerical distance between two quantities increases (*24*, *25*). This is reflected in better performance in cases where the two numerosities presented are remote from each other, giving rise to a bell-shaped performance function. The network also made more errors when the numerical distance between the two presented numerosities was small than when the distance was large, thus showing a numerical distance effect as reflected by the bell-shaped performance functions in Fig. 6A.

The second important characteristic is the "numerical size effect," the finding that discrimination of numerosities with constant numerical distance worsens as the numerical magnitude increases. As a consequence, the behavioral performance functions widen with increasing magnitude (*24*). In agreement with the numerical size effect, the network had more difficulty comparing large numerosities of a given numerical distance than small numerosities with the same distance (Fig. 6A; see Materials and Methods for details). As a consequence, the network's performance functions got wider as numerosity increased, thus mirroring the network units' tuning curves (Fig. 4). The performance functions also became symmetrical with a near-constant width when plotted on a logarithmic scale (Fig. 6B). To quantify this, we again fitted Gaussian functions to the network's performance functions and observed a significantly better fit on the nonlinear scales as compared to the linear scale ($P < 0.05$, paired t test; Fig. 6C), a significant positive slope when the SD of the Gaussian fits on the linear scale was plotted against the preferred numerosities ($r = 0.92$, $P < 0.0001$; Fig. 6D) and a flat slope when the logarithmic scale was used ($r = -0.36$, $P = 0.17$; Fig. 6D). The stark similarity between the results summarized in Figs. 4 and 6 indicates that, as observed in animals (*20*, *21*), the units' tuning curves and the network's performance output were tightly linked.

DISCUSSION

Number sense mechanisms inherent to the visual system

Compared to other network models of number processing, the main advance offered by the HCNN we implemented is that its architecture and function closely mimic the visual system, such as hierarchical layers in which network units with receptive fields and exhibiting lateral inhibition form topographically organized maps of visual features. Although our model was merely trained to classify natural images in a task that was unrelated to numerosity, its spontaneously emergent numerosity-tuned units allowed reliable categorization of the number of items in dot displays. These findings suggest that the spontaneous emergence of the number sense is based on mechanisms inherent to the visual system. The workings of the visual system seem to be sufficient to arrive at a visual sense of number. Numerosity selectivity can emerge simply as a by-product of exposure to natural visual stimuli, without requiring any explicit training for numerosity estimation. The basic number sense may not depend on the development of a certain specialized domain but seem to capitalize on already existing cortical networks. This could explain why numerically naïve subjects, such as newborns (*3*) and animals in the wild (*26*), are innately endowed with numerical capabilities. Of course, this is not to say that numerical competence, both nonsymbolic and particularly symbolic, would not be enhanced and shaped by experience and task demands later in life.

Beyond providing an explanation of the neuroscience of the number sense, our approach also highlights how artificial neural networks give rise to unexpected feature selectivity that helps to understand emergent properties of the brain. Our results show that artificial neural networks seem to extract many more higher-order features from

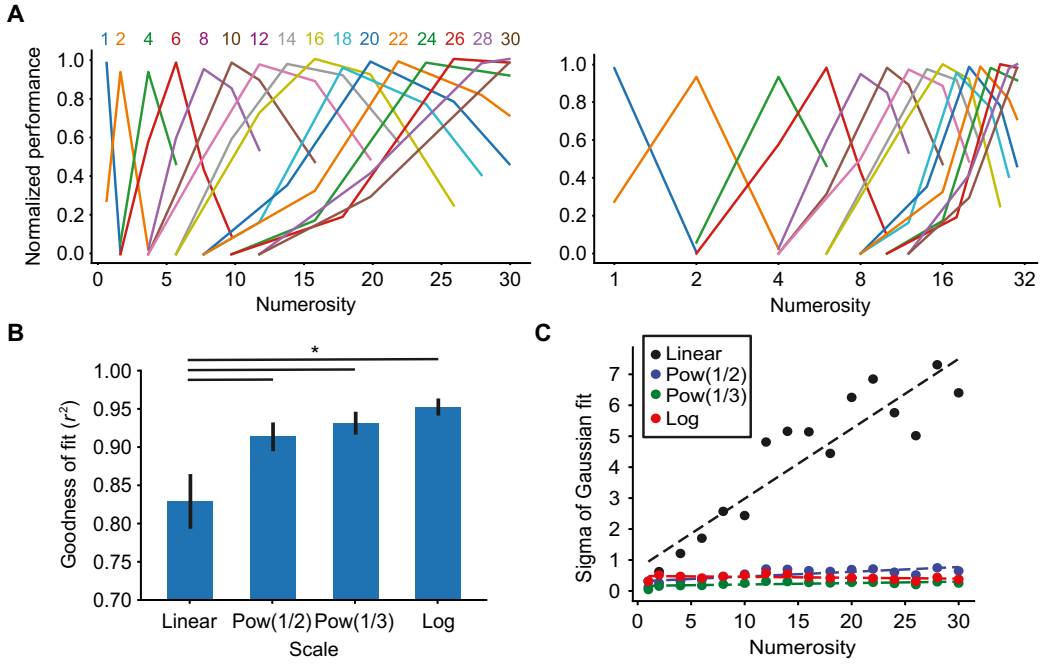

Fig. 6. Performance of the HCNN model in the numerosity matching task. (**A**) Left: Performance functions resulting from the discrimination of numerosities plotted on a linear scale. Each curve shows the probability of the model predicting that the sample image contains the same number of items as the test image (peak of the function). Sample numerosity is indicated above each curve. Right: Same performance functions but plotted on a logarithmic scale. (**B**) Average goodness-of-fit measure for fitting Gaussian functions to the performance tuning curves on different scales [$P_{\text{linear-log}} = 0.003$; $P_{\text{linear-pow(1/2)}} = 0.049$; $P_{\text{linear-pow(1/3)}} = 0.016$]. *$P < 0.05$. (**C**) SD of the best-fitting Gaussian function for each of the performance tuning curves for different scales.

natural images than previously believed. Of particular interest is the level of abstraction and the generalization these network units display. These aspects of neural networks can be exploited to better understand their ability to generalize across different tasks.

Our approach with a biologically inspired deep neural network lends insights into the putative cellular mechanisms that give rise to a number sense. Unlike previous models for visual number coding that relied on either hard-coded connection strengths (*22*, *23*) or training on non-naturalistic dot pattern stimuli (*13*), our model was merely trained to classify natural images in a task that was unrelated to numerosity. In addition, while previous models (*13*, *22*) only controlled for the most obvious non-numerical stimulus parameter, namely, the total area of the objects, we verify the abstract nature of numerosity coding by using an extensive set of controls that address not only the total area of the objects to be enumerated but also their density, individual shapes, and overall convex hull.

Comparison with the neurophysiological number code

Despite these controls, a significant portion of network units in the topmost layers of our model spontaneously developed numerosity encoding that was virtually identical to the encoding observed in real neurons. Just like neurons in the brains of numerically naïve animals, about 10% of the network units exhibited numerosity selectivity (*9*, *10*). Moreover, and in agreement with real number neurons, the network units were tuned to preferred numerosities, exhibited approximate tuning that decreased in precision with increasing numbers, and were best described on a logarithmically compressed number line. The activity of the network's numerosity-selective units obeyed the Weber-Fechner law known to be followed by neurons in the human (*11*), monkey (*12*, *21*), and crow (*27*, *28*) brain. A logarithmically compressed number line has also been retraced indirectly in the human cerebral cortex using blood oxygen level–dependent activity in human functional imaging studies (*29*, *30*).

Previous network models of number coding postulated the existence of "summation units" that monotonically increase or decrease responses with increasing numbers. These summation units were either implemented as necessary precursors to tuned number detectors (*22*, *23*) or emerged as actual output units (*13*). However, summation units were irrelevant with respect to both functions in our deep network. Neither the output layer nor the preceding intermediate layers contained a meaningful proportion of summation units. In all relevant network layers, tuned number neurons dominated by far. To evaluate whether even a small proportion of summation units might have played meaningful role in our network's functionality and the emergence of tuned number neurons, we eliminated the responses of all summation units before testing the model. However, the proportions of tuned neurons, their distribution, and average tuning curves were unaffected. We therefore conclude that summation units are not necessary for the network to develop number detectors. These computational results are in agreement with findings in extensive single-cell recordings in humans (*11*), monkeys (*12*, *20*, *21*, *31*), and crows (*27*, *28*), in both numerically trained and numerically naïve animals (*9*, *10*), in which exclusively tuned number neurons were reported. Our network results therefore indicate that summation units are not relevant for the number sense.

Numerical discrimination performance

The network's numerosity-selective units were sufficient to explain numerical discrimination performance seen in humans and animals. They underlay the performance of the network on a task requiring

abstraction of absolute numerosity from the low-level features of visual stimuli. The network's performance reflected characteristics that are well known from the psychophysical literature, such as the numerical distance effect, the numerical size effect, and logarithmic scaling. The network's performance thus obeyed the Weber-Fechner law known to exist for human and animal number discriminations (*24*).

We used the responses of numerosity-selective units that spontaneously emerged in the numerosity-naïve object recognition network to train a small two-layer neural network to judge whether two images contained the same number of dots. In generalization tests, viewing novel pairs of dot images, this network matched the number of dots with a significant accuracy of 81% correct responses (with 50% as chance level). This accuracy was quantitatively similar to the performance of monkeys and even humans in an analogous delayed matching-to-numerosity task, particularly with large numerosities (*24*). Because adjacent numerical values in dot displays are difficult to discriminate for humans and monkeys due to the numerical distance effect (*12*, *32*), the network is not expected to reach perfect discrimination. Only with ample training, the performance of monkeys slightly increased over time (*24*), and also in agreement with this behavioral enhancement, more and more selective prefrontal cortex neurons represented numerosity (*33*). We therefore suspect that the accuracy of our network would increase with an implementation of reinforcement learning that allows the numerosity-selective units to adapt while performing the numerosity matching task.

Outlook

Our network was designed to process objects shown simultaneously on displays. Such a simultaneous assessment of the number of items is the most common presentation format to investigate the nonsymbolic number sense in humans (*5*, *6*, *24*, *25*, *29*) and animals (*24*, *34*, *35*), and our deep network spontaneously derived the number of items from these multi-dot patterns. It would be interesting to know whether and how our network could be extended to also deal with quantities of items that are presented sequentially. Sequential enumeration requires an assessment of number across time, rather than across space as for dot displays, and the neuronal mechanisms between these two processes differ (*36*, *37*). True counting is a sequential process. Children arrive at this symbolic counting stage once they understand numeral lists based on the successor principle, i.e., the idea that each number is generated by adding one to its predecessor (*38*, *39*). How this key concept paves the way toward exact symbolic numbers represented in the brain is currently unknown and needs to be explored in further experimental and computational studies.

MATERIALS AND METHODS

Neural network model

We used an HCNN (*18*) that consisted of a feedforward hierarchy of layers, in which visual input was received by network units in the first layer and propagated through multiple layers along the hierarchy. The architecture of the model is shown in Fig. 1A and detailed in Table 1. Two main types of layers were used in the model: convolutional layers and pooling layers. In total, our network comprised 13 layers: 8 convolutional layers and 5 pooling layers. Network units in a convolutional layer computed a weighted sum of their inputs, normalized it to a standard range, and passed it through a nonlinear activation function. Network units in pooling layers aggregated responses by computing an average or a maximum over local nonoverlapping patches in their input. This process provided a degree of translational invariance and reduced the spatial size of the input. Network units in each layer were organized topographically into multiple feature maps, and network units in each feature map detected the presence of a certain visual feature at all possible locations in the input. Units in the same feature map shared weights. Therefore, each feature map can be seen as collectively computing a convolution between its inputs and a weight kernel. The weight kernels were adapted for a specific task by optimizing an objective function that measured the performance of the model on that task. The network architecture and hyperparameters (i.e., number of layers, number of kernels in each layer, and kernel sizes) were chosen to provide a reasonable accuracy on the object recognition task while remaining similar to the networks used as models for V4 and IT (*16*, *17*) and to ensure that neurons in the final layer of the feature extraction stage have receptive fields that cover the entire input image. However, the findings reported in this work were not sensitive to the exact choice of hyperparameters.

We trained the HCNN model to classify color images into objects. The network can be conceptually divided into two parts: a feature extraction network and a classification network (Fig. 1A). The input to the feature extraction network was a color image of size 224 × 224 pixels. The feature extraction network consisted of convolutional and max-pooling layers. Inputs to a convolutional layer were padded with zeros along the edges so that its input and output had the same size. The activation function used in convolutional layers was the rectified linear function $f(x) = \max(x, 0)$. Before applying the activation function, the outputs of the convolution operation were normalized to have zero mean and unit SD [batch normalization; (*40*)]. In a convolutional layer, network units that received the same input (i.e., network units at the same spatial location in different feature maps) inhibited each other using the local response normalization function introduced by Krizhevsky *et al.* (*14*)

$$b^i_{x,y} = a^i_{x,y} \Bigg/ \left(k + \alpha \sum_{\max(0,i-n/2)}^{\min(N-1,i+n/2)} \left(a^j_{x,y}\right)^2 \right)$$

where $a^i_{x,y}$ is the unnormalized response for the network unit at location x, y in the ith feature map, N is the total number of feature maps in the layer, $b^i_{x,y}$ is the normalized response, and the rest of the variables are constants set to the values $k = 2$, $\alpha = 10^{-4}$, $\beta = 0.75$, and $n = 15$, which were based on the values used in (*14*). This normalized the activity of each network unit by dividing by a measure of the total activity of n network units at the same spatial location in adjacent feature maps. Normalizing the local responses enforced competition among these network units, thereby mimicking the effects of lateral inhibition. Max-pooling layers aggregated responses in nonoverlapping regions of 2 × 2 network units. The classification network consisted of a global average pooling layer (*41*) that computed the average response over all spatial locations in each of the final feature maps produced by the feature extraction network and an output layer that performed the classification and contained 1000 network units, one network unit per object category. The response of each network unit in this layer represented the probability that the corresponding object category was present in the image. To ensure that the responses in the final layer represented a valid probability distribution, the Softmax activation function $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ was applied, where x_i is the response of the ith network unit in the layer. The weights of the model were

initialized randomly [Xavier initialization; (*42*)] and then optimized by minimizing the cross-entropy between the predicted object category probabilities and the ground-truth labels. The minimization was performed using mini-batch gradient descent (*43*) with a batch size of 256 images, a learning rate of 0.1, and a momentum of 0.9. The model was trained for 10 epochs (complete presentations of the training data). The model was implemented in Python using the PyTorch framework (*44*). Training was performed on two NVIDIA K80 graphics processing units.

Stimulus datasets

The neural network model was trained to perform object classification on the ILSVRC2012 ImageNet dataset (*45*), which contains around 1.2 million images. Each image was labeled with the category of the most prominent object depicted in the image. The dataset contains images of objects belonging to 1000 categories. The object classification accuracy of the model was evaluated on 50,000 images that were not seen by the model during training.

To examine its response to different numbers of items (i.e., numerosities), the network was presented with randomly generated images containing n = 1, 2, 4, 6, …30 dots. The network was tested under three different stimulus sets: a standard set and two control sets that controlled for non-numerical visual stimulus cues. In the standard condition, all the dots had about the same radius (standard set, $r = 7 \pm 0.7\epsilon$ pixels, where ϵ was randomly drawn for a standard normal distribution separately for each dot). In the first control condition (control set 1), the total area of the dots and the average distance between pairs of dots were kept constant at 1200 pixels and 90 to 100 pixels, respectively. In the second control condition (control set 2), the convex hull of the dots was the same (a pentagon of constant circumference) regardless of numerosity (for numerosities larger than 4), and the shapes of the individual dots varied (possible shapes: circle, rectangle, ellipse, and triangle). The network's responses were evaluated over n = 336 different images with an equal number of images (n = 7) for each numerosity and stimulus set combination. The sample sizes of total images and images of the same numerosity were adjusted to those applied in electrophysiological monkey experiments (*20*). For the numerosity matching task, the matching model was trained on a similarly generated but larger dataset of 4800 images and tested on a separate dataset of the same size.

Analysis of network units

After being trained for object classification, the network was presented with images depicting different numbers of items (numerosities 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, and 30), and the units' responses in the final layer of the feature extraction network (see Table 1) were analyzed. Analogous to the approach used to detect numerosity-selective neurons in monkeys and humans (11, 20, 31), a two-way ANOVA with numerosity (16 numerosities) and stimulus sets (three numerosity sets: one standard and two control) as factors was applied to find network units that exhibited a significant effect for numerosity ($P < 0.01$), but no significant effect for stimulus set or interaction. These network units with a main effect for numerosity, but no main effect for stimulus set or any interaction, were labeled as "numerosity-selective network units." Tuning curves for selective network units were calculated by averaging the responses of each unit for all presentations of the same numerosity. For each network unit, the numerosity that induced the largest average response was defined as the preferred numerosity. To summarize the responses of the numerosity-selective network units, tuning curves for neurons that had the same preferred numerosity were pooled by averaging and normalized to the 0 to 1 range. Gaussian functions were fit to the pooled network tuning curves plotted on four different scales: $f(x) = x$, $f(x) = x^{0.5}$, $f(x) = x^{0.33}$, and $f(x) = \log_2(x)$. Given the symmetry of the Gaussian function, the scale on which it best fits the tuning curves is expected to be the one that best describes the data (*21*). The SD of the Gaussian fits was taken as a measure of the width of the tuning curves.

Summation (or monotonic) units were defined as those numerosity-selective units that had a preferred numerosity of 1 or 30 and whose tuning curves could be fit with a straight line with a coefficient of determination larger than 0.5. To test the performance relevance of summation units, summation units were eliminated by setting their responses to zero; the model was then tested again with all analyses as described before.

Numerosity matching task for the network

To test the relevance of the numerosity-selective network units on the network's performance, a simple model was trained to use their activity to solve a numerosity matching task. In each trial, the network was presented with two images of dot patterns, a sample image and a test image, and the responses of the numerosity-selective network units to each image were recorded. The matching model was trained to use these responses to discriminate between trials showing matching numerosities as opposed to nonmatching numerosities. Similar to the approach used by (*20*) for monkey experiments, the sample numerosities covered the entire 1 to 30 range, and the test numerosities were randomly chosen to be 0.4, 0.7, 1, 1.3, or 1.6 times the sample numerosity.

The matching model consisted of a small feedforward neural network with an output layer that contains two network units (indicating either a numerosity match or nonmatch) and an intermediate layer that contained 16 network units. The output layer used the Softmax activation function, while the intermediate layer used the rectified linear activation function. The model was trained in a manner similar to the original network. The dropout procedure (*46*) was used in the input layer to prevent overfitting (75% of input units were randomly silenced each training iteration). The model was tested on new images, and the matching accuracy was computed. Then, we averaged the responses of each network unit toward the sample numerosities across correct trials and normalized each network unit's responses to have a maximum of 1. We created a population tuning curve by centering each network unit on its preferred numerosity such that the responses to the other numerosities could be expressed as the numerical distance from its preferred numerosity. Using the same preferred numerosity for each network unit, we also computed this curve whenever the network erroneously classified numerosity, i.e., during error trials. Performance tuning curves were constructed by computing, for each possible pair of sample and test numerosities, the percentage of trials in which the network judged the two numerosities to be the same. Similar to the tuning curves of real neurons (*21*), Gaussian functions were fit to the behavioral tuning curves plotted on four different scales, and the SD of the Gaussian functions was taken as a measure of the width of the network's performance tuning curves.

REFERENCES AND NOTES

1. T. Dantzig, *Number: The Language of Science* (The Free Press, 1930).
2. S. Dehaene, *The Number Sense* (Oxford Univ. Press, 1999).
3. V. Izard, C. Sann, E. S. Spelke, A. Streri, Newborn infants perceive abstract numbers. *Proc. Natl. Acad. Sci. U.S.A.* **106**, 10382–10385 (2009).

4. R. Rugani, L. Regolin, G. Vallortigara, Discrimination of small numerosities in young chicks. *J. Exp. Psychol. Anim. Behav. Process.* **34**, 388–399 (2008).
5. D. Burr, J. Ross, A visual sense of number. *Curr. Biol.* **18**, 425–428 (2008).
6. D. C. Burr, G. Anobile, R. Arrighi, Psychophysical evidence for the number sense. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **373**, 20170045 (2017).
7. J. Park, N. K. DeWind, M. G. Woldorff, E. M. Brannon, Rapid and direct encoding of numerosity in the visual stream. *Cereb. Cortex* **26**, 748–763 (2016).
8. E. Castaldi, D. Aagten-Murphy, M. Tosetti, D. Burr, M. C. Morrone, Effects of adaptation on numerosity decoding in the human brain. *Neuroimage* **143**, 364–377 (2016).
9. P. Viswanathan, A. Nieder, Neuronal correlates of a visual "sense of number" in primate parietal and prefrontal cortices. *Proc. Natl. Acad. Sci. U.S.A.* **110**, 11187–11192 (2013).
10. L. Wagener, M. Loconsole, H. M. Ditz, A. Nieder, Neurons in the endbrain of numerically naive crows spontaneously encode visual numerosity. *Curr. Biol.* **28**, 1090–1094.e4 (2018).
11. E. F. Kutter, J. Bostroem, C. E. Elger, F. Mormann, A. Nieder, Single neurons in the human brain encode numbers. *Neuron* **100**, 753–761.e4 (2018).
12. A. Nieder, The neuronal code for number. *Nat. Rev. Neurosci.* **17**, 366–382 (2016).
13. I. Stoianov, M. Zorzi, Emergence of a 'visual number sense' in hierarchical generative models. *Nat. Neurosci.* **15**, 194–196 (2012).
14. A. Krizhevsky, I. Sulskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2012).
15. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV] (4 September 2014).
16. C. F. Cadieu, H. Hong, D. L. K. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, J. J. DiCarlo, Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLOS Comput. Biol.* **10**, e1003963 (2014).
17. D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, J. J. DiCarlo, Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 8619–8624 (2014).
18. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time-series, in *The Handbook of Brain Theory and Neural Networks* (MIT Press, 1995), vol. 3361, pp. 255–258.
19. D. H. Hubel, T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160**, 106–154 (1962).
20. A. Nieder, K. Merten, A labeled-line code for small and large numerosities in the monkey prefrontal cortex. *J. Neurosci.* **27**, 5986–5993 (2007).
21. A. Nieder, E. K. Miller, Coding of cognitive magnitude: Compressed scaling of numerical information in the primate prefrontal cortex. *Neuron.* **37**, 149–157 (2003).
22. S. Dehaene, J.-P. Changeux, Development of elementary numerical abilities: A neuronal model. *J. Cogn. Neurosci.* **5**, 390–407 (1993).
23. T. Verguts, W. Fias, Representation of number in animals and humans: A neural model. *J. Cogn. Neurosci.* **16**, 1493–1504 (2004).
24. K. Merten, A. Nieder, Compressed scaling of abstract numerosity representations in adult humans and monkeys. *J. Cogn. Neurosci.* **21**, 333–346 (2009).
25. P. B. Buckley, C. B. Gillman, Comparisons of digits and dot patterns. *J. Exp. Psychol.* **103**, 1131–1136 (1974).
26. S. Benson-Amram, G. Gilfillan, K. McComb, Numerical assessment in the wild: Insights from social carnivores. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **373**, 20160508 (2017).
27. H. M. Ditz, A. Nieder, Neurons selective to the number of visual items in the corvid songbird endbrain. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 7827–7832 (2015).
28. H. M. Ditz, A. Nieder, Sensory and working memory representations of small and large numerosities in the crow endbrain. *J. Neurosci.* **36**, 12044–12052 (2016).
29. M. Piazza, V. Izard, P. Pinel, D. Le Bihan, S. Dehaene, Tuning curves for approximate numerosity in the human intraparietal sulcus. *Neuron* **44**, 547–555 (2004).
30. S. N. Jacob, A. Nieder, Tuning to non-symbolic proportions in the human frontoparietal cortex. *Eur. J. Neurosci.* **30**, 1432–1442 (2009).
31. A. Nieder, D. J. Freedman, E. K. Miller, Representation of the quantity of visual items in the primate prefrontal cortex. *Science* **297**, 1708–1711 (2002).
32. A. Nieder, S. Dehaene, Representation of number in the brain. *Annu. Rev. Neurosci.* **32**, 185–208 (2009).
33. P. Viswanathan, A. Nieder, Differential impact of behavioral relevance on quantity coding in primate frontal and parietal neurons. *Curr. Biol.* **25**, 1259–1269 (2015).
34. E. M. Brannon, H. S. Terrace, Ordering of the numerosities 1 to 9 by monkeys. *Science* **282**, 746–749 (1998).
35. H. M. Ditz, A. Nieder, Numerosity representations in crows obey the Weber–Fechner law. *Proc. Biol. Sci.* **283**, 20160083 (2016).
36. A. Nieder, I. Diester, O. Tudusciuc, Temporal and spatial enumeration processes in the primate parietal cortex. *Science* **313**, 1431–1435 (2006).
37. A. Nieder, Supramodal numerosity selectivity of neurons in primate prefrontal and posterior parietal cortices. *Proc. Natl. Acad. Sci. U.S.A.* **109**, 11860–11865 (2012).
38. M. Le Corre, S. Carey, One, two, three, four, nothing more: An investigation of the conceptual sources of the verbal counting principles. *Cognition* **105**, 395–438 (2007).
39. S. Carey, *The Origin of Concepts: Oxford Series in Cognitive Development* (Oxford Univ. Press, 2009).
40. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv: 1502.03167[cs.LG] (11 February 2015).
41. M. Lin, Q. Chen, S. Yan, Network in network. arXiv:1312.4400[cs.NE] (16 December 2013).
42. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. *Proc. Mach. Learn. Res.* **9**, 249–256 (2010).
43. S. Ruder, An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG] (15 September 2016).
44. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch. *Adv. Neural Inf. Process. Syst.* **30**, 1–4 (2017).
45. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
46. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).

Acknowledgments: We acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC, and the German Research Foundation (DFG) through grant no. INST 37/935-1 FUGG. **Funding:** This work was supported by a DFG grant to A.N. (NI 618/10-1). **Author contributions:** K.N., P.V., and A.N. designed the research. K.N. designed and implemented the model. K.N., P.V., and A.N. discussed all aspects of the implementation of the model, analysis, and figures. K.N., P.V., and A.N. wrote the paper. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper. Additional data related to this paper may be requested from the authors.

Submitted 19 October 2018
Accepted 26 March 2019
Published 8 May 2019
10.1126/sciadv.aav7903

Citation: K. Nasr, P. Viswanathan, A. Nieder, Number detectors spontaneously emerge in a deep neural network designed for visual object recognition. *Sci. Adv.* **5**, eaav7903 (2019).

Number detectors spontaneously emerge in a deep neural network designed for visual object recognition

Khaled Nasr, Pooja Viswanathan and Andreas Nieder

Sci Adv **5** (5), eaav7903.
DOI: 10.1126/sciadv.aav7903