Cognition 215 (2021) 104815

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/cognit

Learning exact enumeration and approximate estimation in deep neural network models

Celestino Creatore[a,1], Silvester Sabathiel[a,b,1], Trygve Solstad[a,*]

[a] Department of Teacher Education, Faculty of Social and Educational Sciences, NTNU—Norwegian University of Science and Technology, Norway
[b] Department of Computer Science, Faculty of Information Technology and Electrical Engineering, NTNU—Norwegian University of Science and Technology, Norway

ARTICLE INFO

Keywords:
Number sense
Approximate number
Exact number
Deep neural networks
Computational modelling
Representations

ABSTRACT

A system for approximate number discrimination has been shown to arise in at least two types of hierarchical neural network models—a generative Deep Belief Network (DBN) and a Hierarchical Convolutional Neural Network (HCNN) trained to classify natural objects. Here, we investigate whether the same two network architectures can learn to recognise exact numerosity. A clear difference in performance could be traced to the specificity of the unit responses that emerged in the last hidden layer of each network. In the DBN, the emergence of a layer of monotonic 'summation units' was sufficient to produce classification behaviour consistent with the behavioural signature of the approximate number system. In the HCNN, a layer of units uniquely tuned to the transition between particular numerosities effectively encoded a thermometer-like 'numerosity code' that ensured near-perfect classification accuracy. The results support the notion that parallel pattern-recognition mechanisms may give rise to exact and approximate number concepts, both of which may contribute to the learning of symbolic numbers and arithmetic.

1. Introduction

What is the foundation for the conceptual development of natural numbers and elementary arithmetic? Although counting is our only procedure for exactly determining the size of large sets of items, both humans and non-human animals have a natural 'number sense' that consists of two components; for sets smaller than five, we can directly perceive the *exact* number of items in a process called 'subitizing' (Agrillo, Piffer, Bisazza, & Butterworth, 2012; Clements, Sarama, & Macdonald, 2019; Jevons, 1871; Tomonaga & Matsuzawa, 2002). Beyond this 'subitizing range', we can make *approximate* judgements about (i) the numerosity of a single set of items (estimation task), and (ii) the relative size of two sets of items (discrimination task), with an accuracy that decreases logarithmically with the size of the set or the difference between sets (Dehaene, 2011; Izard, Sann, Spelke, & Streri, 2009; Rugani, Regolin, & Vallortigara, 2008). Recently, our understanding of how this approximate number sense may be grounded in perception has been substantially advanced through neurophysiological experiments and computational modelling (Nieder & Dehaene, 2009). However, what cognitive mechanisms underlie and differentiate approximate estimation and exact enumeration is still unclear.

Understanding natural numbers and arithmetic requires a notion of exact numerosity and the association of exact numerosity to number-word labels. A system representing discrete numerosity should allow (i) the establishment of one-to-one correspondence between numerosity and objects and (ii) the distinguishing of transformations that are invariant to numerosity, such as the spatial translation of objects (Butterworth, 2010). An example of a representation system which satisfies these requirements is the 'magnitude code'—a thermometer-like representation of numerosity in which the numerosity 'one' is represented by the activity of one particular group of neural units, 'two' by the activity of an additional group of units, etc. (Testolin, Zou, & McClelland, 2020; Verguts & Fias, 2004; Verguts, Fias, & Stevens, 2005; Zorzi & Butterworth, 1999). Building on this representation of numerosity enabled a network model, trained with a one-shot Hebbian learning rule, to make exact number judgements about the largest of two sets that accounted for several aspects of human performance on the task (Zorzi & Butterworth, 1999). Subsequently, Verguts and Fias (2004) successfully trained a model to classify one-dimensional vectors by their digit sum with supervised learning and observed that a magnitude code emerged in the hidden layer. These results suggest an important role of symbols and direct feedback in the learning of representations which support

[*] Corresponding author at: Department of Teacher Education, NTNU, NO-7491 Trondheim, Norway.
E-mail addresses: cele.creatore@gmail.com (C. Creatore), silvester.sabathiel@ntnu.no (S. Sabathiel), trygve.solstad@ntnu.no (T. Solstad).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.cognition.2021.104815
Received 1 November 2020; Received in revised form 12 June 2021; Accepted 15 June 2021
Available online 26 June 2021

exact number concepts.

Recently, deep neural network models trained on two-dimensional dot-pattern images that more closely resemble experimental conditions have shed light on the potential mechanisms underlying approximate number sense. Some studies have used a strategy based on *unsupervised learning* in a class of hierarchical generative networks called deep belief networks (DBNs). Rather than explicitly recognising or classifying the input, these networks are trained with the objective of building an internal representation of the input data through minimising the error when reconstructing the input data (Hinton, Osindero, & Teh, 2006; Hinton & Salakhutdinov, 2006). Based on the internal representation, the networks can learn to discriminate between numerosities in an approximate manner that is comparable to empirical studies with humans (Stoianov & Zorzi, 2012; Testolin, Dolfi, Rochus, & Zorzi, 2020; Testolin, Zou, & McClelland, 2020; Zorzi & Testolin, 2018). These modelling results highlight the relevance of unsupervised sensory experience in developing an approximate number sense. Other studies, growing out of the latest developments in neural network models for visual pattern recognition (Krizhevsky, Sutskever, & Hinton, 2012; LeCun et al., 1989), have shown that approximate sensitivity to numerosity can also arise in hierarchical neural networks trained with *supervised learning*, either as a result of classifying dot-pattern images (Chen, Zhou, Fang, & McClelland, 2018) or as a byproduct of classifying objects in natural scenes (Nasr, Viswanathan, & Nieder, 2019). All of these deep network models differed in their architectural design, learning policy (supervised learning via backpropagation versus unsupervised learning via reconstruction error), learning objective (classification of objects versus reconstruction of input images), and input data. Yet, they yield very similar results which mirror those obtained both in behavioural experiments in a wide range of species and electrophysiological recordings from single neurons in the primate brain (Nieder, 2005, 2016).

In this paper, we ask to what extent two prominent deep network models of the approximate number sense—a DBN and an HCNN—can learn to recognise *exact* numerosity and what learning mechanisms and representations might underlie this ability.

2. Methods

2.1. Hierarchical convolutional neural network model

We used a supervised HCNN, similar to that of Nasr and coworkers (Nasr et al., 2019). The HCNN is a standard feed-forward network architecture for classifying images (Krizhevsky et al., 2012; LeCun et al., 1989) which is loosely inspired by the architecture of the visual cortex. The model's weights are updated through the backpropagation learning algorithm for which a biologically plausible implementation is currently lacking. However, the principle of learning from labels is both biologically relevant and highly effective, and the search for biologically realistic approximations to the backpropagation algorithm and alternative supervised learning algorithms is ongoing (e.g. Lillicrap, Santoro, Marris, Akerman, & Hinton, 2020). The architecture of the network is depicted in Fig. 1A and the details of its dimensions, hyperparameters, and the implementation can be found in Table 1.

The HCNN consisted of four alternating layers of convolution and pooling operations responsible for the extraction of visual features. The

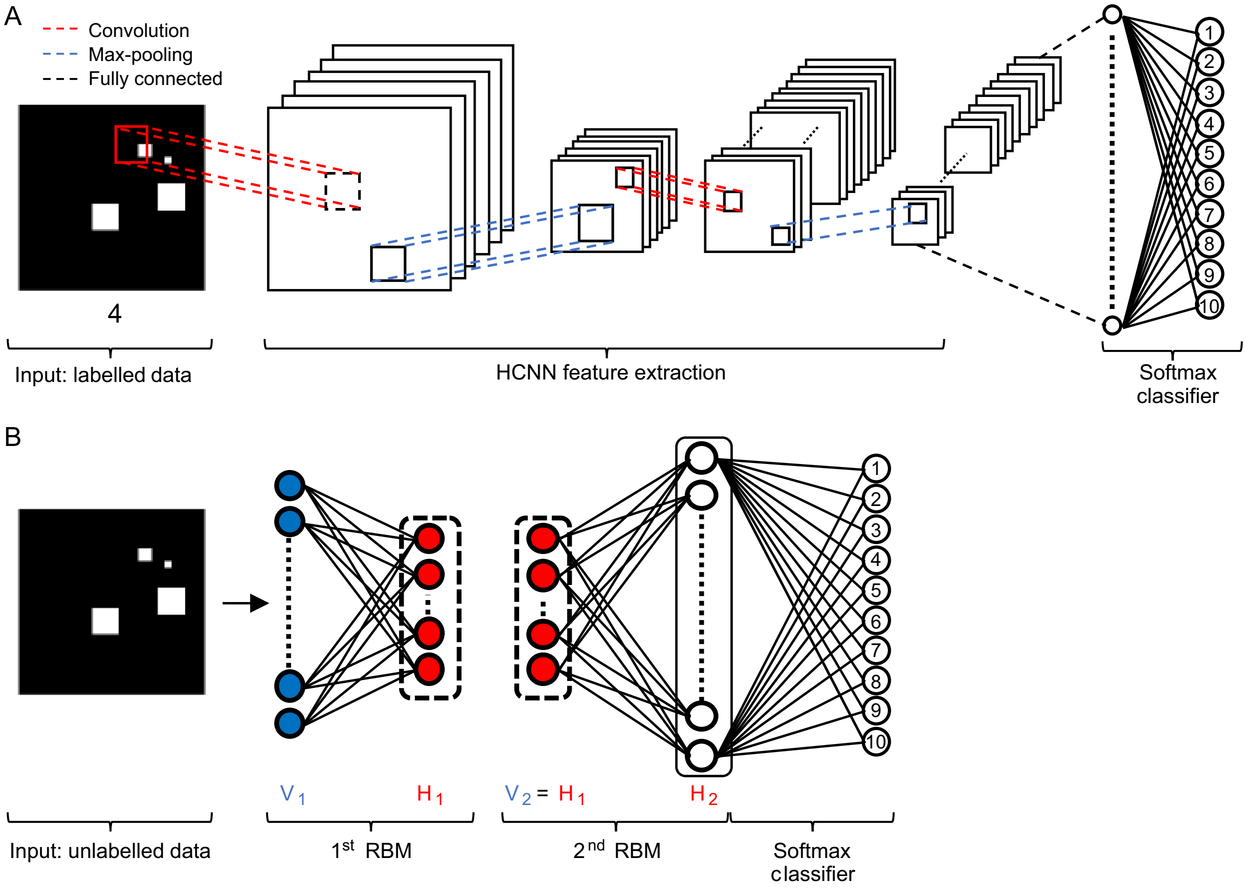

Fig. 1. Simplified architecture of the two deep learning models. A: The Hierarchical Convolutional Neural Network (HCNN) used to classify the labelled input images by their numerosities from 1 to 10 with a fully supervised learning strategy. B: The Deep Belief Network consisted of two layers of Restricted Boltzmann Machines (RBM) and extracted abstract representations of unlabelled input data with unsupervised learning. These abstract representations were used to classify the numerosity of the input data into one of ten classes using a linear Softmax classifier. V_n denotes visible layer number n and H_m denotes hidden layer number m.

Table 1
Description of the layers in the HCNN.

Layer	Type	Spatial Size	Parameters	Activation
0	Input	28 × 28		
1	Convolutional	28 × 28	filters = 8, kernel size = 5	ReLU
2	Max Pooling	14 × 14	pool size = 2, stride = 2	
3	Convolutional	14 × 14	filters = 64, kernel size = 3	ReLU
4	Max Pooling	7 × 7	pool size = 2, stride = 2	
5	Flatten	3136		
6	Fully connected	90		Sigmoid
7	Dropout (For training)	90	dropout rate = 0.25	
8	Fully connected	10		Softmax

extracted features were then passed on to the classification layer of the network, which generated numerosity-label probabilities via a Softmax activation function. The architecture was chosen to be as simple as possible while still reliably solving the numerosity classification task. Models with 30 or 10 units in the last feature extraction layer did not reliably show the same level of performance as the model with 90 units.

The weights were adapted based on the gradient of the cross-entropy loss function, given the Softmax of the network's output and the one-hot encoded numerosity labels. The neural network was trained for 100 epochs, with each epoch consisting of gradient updates over sampled batches of size 128 iterated through the whole training set. The gradient updates were calculated using the RMSProp optimiser with an initial learning rate of 0.0005 and a learning decay rate of 0.7.

The model was implemented in Python 3.0 using TensorFlow 1.15.2. All the trainable variables were initialised by the default initialiser of TensorFlow.

2.2. Deep belief network model

A DBN (Hinton et al., 2006; Hinton & Salakhutdinov, 2006) is an unsupervised, hierarchical neural network model with the learning objective of developing an internal representation adequate to reconstruct its own sensory input, rather than to directly classify the input data. This learning principle can be interpreted as a form of learning by observation with no specific behavioural task to perform and no feedback from the environment about the quality of the representation that develops. The network updates its weights through a Hebbian-like associative learning rule, which is a familiar principle of neural plasticity that adds to the model's biological plausibility (Hinton et al., 2006; Stoianov & Zorzi, 2012; Testolin, Dolfi, et al., 2020; Testolin, Zou, & McClelland, 2020; Zorzi & Testolin, 2018).

The DBN was composed by stacking together two Restricted Boltzmann Machines (RBMs) (Fig. 1B), in a similar manner as in Testolin, Dolfi, Rochus, and Zorzi (2020). The RBM realises a hierarchical generative model where two layers of non-linear processing units are trained with the scope of minimising the discrepancy (reconstruction error) between the empirical distribution of the input data and the generative model distribution, i.e. the internal representation generated by the network in response to the input. The neural units in the first hidden layer (H_1 in Fig. 1B) encode simple visual properties of the input nodes (V_1) to which they are fully connected via a matrix of symmetric weights. These encoded features (the activation of the hidden neurons) represent the first, abstract (or internal) representation of the external stimuli generated by the model. These features are then combined into more complex features in the subsequent layer (H_2 in Fig. 1B).

The RBMs were trained using the one-step contrastive divergence method (Hinton, 2002) for 400 epochs. The connection weights between the visible and hidden units were randomly initialised using a normal distribution with zero mean and standard deviation of 0.1. The source code of the DBN was implemented in Python 3.0 using Pytorch and was based on the implementation of the original Hinton's DBN model developed in Python by Testolin, Stoianov, De Grazia and Zorzi (Testolin, Stoianov, De Filippo De Grazia, & Zorzi, 2013).

In order to evaluate how efficiently the emergent representations of the DBN could determine the numerosity of the input images, we trained a 10-class Softmax classifier, corresponding to the one in the HCNN model, to classify the input images according to their numerosities based on input from the second layer of the DBN.

2.3. Numerosity datasets

We generated a dataset of visual stimuli consisting of 60,000 two-dimensional binary images of size 28 × 28 pixels. The dataset contained ten labelled classes of images corresponding to the numerosities 1 through 10. Each image was composed of between one and ten non-overlapping white squares (pixel intensity equal to one) drawn on a black background (pixel intensity equal to zero). The side length of the white squares varied from 2 to 6 pixels, for a total of five different sizes of squares. Each image was created by placing squares, drawn uniformly from the range of possible side lengths, at random positions with the requirement that the squares have a minimum of 2 pixels distance to other squares and to the edges of the image. The dataset was split into a training set with 75% of the images and a test set with 25% of the images. Another dataset of 100,000 images was produced with the same statistics to create the tuning curves of the trained networks. As a consequence of the white squares varying in size and position, the number of white squares in the images correlated with other variables such as the total area of the white squares ($r = 0.89$) and the convex hull of the white squares ($r = 0.94$). Statistics for the dataset are visualised in Fig. S1.

To investigate the influence of non-numerical features of the images on the performance of the networks, we generated three additional datasets with distinctive shape statistics. The first dataset consisted of images with identically sized squares of side length 3 pixels. In this dataset, numerosity perfectly correlated with area (Fig. S2), and the correlation between numerosity and convex hull was $r = 0.94$. The second dataset consisted of randomly sized rectangles with side length between 1 and 5 pixels (Fig. S3). In this dataset the correlation between numerosity and area was $r = 0.87$ and between numerosity and convex hull was $r = 0.93$. The third dataset consisted of images where the total area of rectangles was fixed to 64 pixels for all numerosities. In this dataset the correlation between numerosity and area was $r = 0$ (Fig. S4). The correlation between numerosity and convex hull of white rectangles was $r = 0.9$. Properties of the additional datasets, other than the shapes and sizes of the objects, such as the restrictions on the objects' relative distance or spatial distributions, were the same as for the original dataset of randomly sized squares described above.

2.4. Tuning curves

In order to analyse the functionality of single units in the network, we produced neural tuning curves by averaging the neural activities over all the images depicting the same numerosity. Assigning each unit to the numerosity it encoded was carried out algorithmically. A unit was said to encode numerosity n if the unit was only active for numerosities '$\geq n$' (and inactive for '$<n$') or only active for numerosities '$\leq n$' (and inactive for '$>n$'). A unit was defined as *active* when its activity level was above 0.5.

2.5. Representational similarity analysis

We used representational similarity analysis (Kriegeskorte, Mur, & Bandettini, 2008; Testolin, Dolfi, et al., 2020) to measure the similarity between internal representations associated with different numerosities.

An activity vector was defined as the vector of activity levels of all neural units in a network layer in response to a given input image. The Pearson correlation between any pair of activity vectors was used as a metric of similarity and visualised in a colour-coded representational similarity matrix. In a representational similarity matrix, the element $r_{i,\ j}$ represents the Pearson correlation between the population vectors computed for input image i and input image j. High positive correlation was shown in red and high negative correlation was shown in blue.

2.6. Hinton diagrams

A Hinton diagram (Hinton & Shallice, 1991) was used to visualise the connections and connection strengths (weights) between the last hidden layer of the HCNN and the output classifier. In a Hinton diagram, positive connection weights are represented by white squares and negative connection weights by black squares. The size of a square represents the magnitude of the corresponding connection weight.

2.7. Area-adjusted baseline

In order to investigate to what extent the neural networks could base their inference of numerosity on the total area of squares in each image, we calculated an area-adjusted baseline for each dataset. The area-adjusted baseline corresponds to the average probability of guessing the right numerosity when the total area of shapes in an image in the dataset, as well as the distribution of total area in the dataset, is known. We first calculated the proportion of images belonging to each numerosity for each distinct value of total area in the dataset. Next, the area-adjusted baseline accuracy was calculated as the weighted average of these proportions over the set of total areas in each numerosity.

3. Results

3.1. Numerosity classification performance

We first asked to what degree the two network architectures, a DBN and an HCNN (Fig. 1), could learn to classify square-pattern images by their numerosity from one through ten. The networks were trained and tested on the very same datasets: an input dataset of 45,000 synthetic square-pattern images depicting different numbers of white squares that varied in size and position, and a separate test set of 15,000 square-pattern images (see section 2). After training, the HCNN classified the test set images by the number of squares with an overall accuracy of 99% (chance level = 10%). The performance was close to uniform over all numerosities (Fig. S5). This result demonstrates that a neural network can learn to recognise exact numerosity even beyond the standard subitizing range of up to four items.

The DBN was trained with an unsupervised, layer-wise policy until the convergence of the reconstruction error at each hidden layer of the network (see Section 2). Following the procedure of Stoianov and Zorzi (2012) we next used the abstract features learnt at the second and final hidden layer of the DBN as input to train a simple Softmax classifier for the classification of the images into ten numerosity categories (Fig. 1B). We evaluated the classification performance of several DBN architectures on the test images by varying the number of neural units in the first hidden layer (h_1) and the second hidden layer (h_2) within the range of $100 - 1000$ (Fig. S6). A DBN architecture with $h_1 = 200$ units in the first hidden layer and $h_2 = 1000$ units in the second hidden layer produced the highest overall accuracy of approximately 60%. This accuracy was substantially higher than the accuracy obtained using the same Softmax classifier trained and tested using either the raw/original test images as input (max 33%, see also Fig. S6) or the abstract features learned with the first hidden layer as input (max 47%), showing that the network had learned representations of the input images that were increasingly sensitive to numerosity for successive layers of processing.

Most previous models of the approximate number system have used a binary numerosity discrimination task (Nasr et al., 2019; Stoianov & Zorzi, 2012; Zorzi & Testolin, 2018). Here, we followed Chen et al. (2018) and Testolin, Zou, and McClelland (2020) and used a numerosity estimation task. The numerosity estimation task allowed us to produce a distribution of classification responses across all numerosities that is more directly comparable to experimental data (Viswanathan & Nieder, 2013) and the theoretical prediction of a number line code (Dehaene, 2011). In a number line code, the approximate representation of numerosity is conceived as a distribution of activation on a mental number line (Dehaene, Piazza, Pinel, & Cohen, 2003) where numbers are sorted by their proximity and the overlap between the distributions of activation increases with numerosity. We found that DBN accuracy was highest for small numerosities, starting with 100% for images with a single square and gradually decreased for larger numerosities (Fig. 2). This numerosity response distribution qualitatively reproduced the classic behavioural variability signature often attributed to the approximate number system and a model that instantiates a number line code (Dehaene, 2001; Gallistel & Gelman, 1992). The graded performance apparent in Fig. 2 is also in qualitative agreement with empirical studies assessing pre-counters' enumeration of up to six items (Sella, Berteletti, Lucangeli, & Zorzi, 2016).

The network performance results show that exact numerosity perception can be learned by a hierarchical neural network trained with supervised learning and consolidate the view that a network model trained under unsupervised learning, like the DBN, can capture key aspects of approximate number perception. We next asked whether the qualitative difference in classification performance, exact versus approximate enumeration of squares in an image, could be explained by the neural representations emerging from the two networks' learning processes.

3.2. Population level representations of numerosity

Although the two network models differed in many regards, they implemented the same Softmax classifier to determine the correct numerosity label. We therefore expected that, after training, classification performance would depend primarily on the representations that had developed in the networks and were fed into the output classifier, rather than the specifics of the architectures or learning algorithms. To provide mechanistic insight into why the two networks performed differently, we investigated the representations of the last layer before the Softmax classifier of each network. We exposed the trained network models to a subset of 1000 images (100 images per numerosity) from the

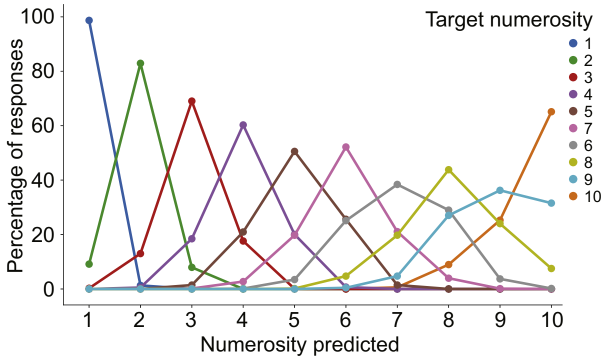

Fig. 2. Numerosity response distribution of the deep belief network. Different colours represent the actual numerosity from one through ten in the input image, as specified in the figure legend. The x-axis denotes the numerosity predicted by the Softmax classifier. The y-axis denotes the percentage of classification responses for each predicted numerosity. Note the decreasing accuracy and broader response variability with increasing target numerosity. The higher performance for numerosity 10 was an artifact of the limited range of numerosities tested.

original test set of square-pattern images and recorded the neural activities triggered by each test image across the 90 and 1000 units in the last layer of the HCNN and DBN, respectively.

A representational similarity analysis (see Section 2) of the neural activities revealed that the two architectures solved the numerosity classification task in qualitatively different ways (Fig. 3). In the last layer of the HCNN, ten separate network states were selectively activated by a specific input numerosity (Fig. 3A). These states exhibited neural activity patterns whose pairwise similarity was high for input images with the same numerosity, and systematically decreased with the difference in numerosity between input images. That is, input images with the same number of squares, but in different sizes and locations, triggered neural activity patterns that were highly correlated (yellow blocks along the diagonal of the representational similarity matrix where $r_{i,j} \approx 1$ for images i and j depicting the same numerosity), while input images with different numbers of squares triggered neural activity patterns with correlations which decreased as a function of the difference between the number of squares in the input images. This population level representation corresponds to a local neural code where the numerosity of the input image can be decoded from which particular subpopulation of the network units are activated by the input.

In contrast to the HCNN, the DBN did not develop a uniquely identifiable activity pattern for each numerosity. Correlations between activity states for input images of both the same and different numerosities were small, and the representational similarity matrix did not show sharp network states or a clear code with discrete transitions in network activity between different numerosities (Fig. 3B). However, summing up the activity of all units for input images of each numerosity showed that the total neural activity generated at the second hidden layer of the DBN increased with increasing numerosity (Fig. 4). Thus, the DBN as a whole was *sensitive* to different visual numerosities in a manner that supported approximate numerosity judgement comparable to biological data.

3.3. Single unit representations of numerosity

A more detailed picture of the representations developed by the networks can be obtained by tracing the activity of single neural units in the final network layer as a function of the input numerosity. We produced neural tuning curves by averaging the neural activities over all the images depicting the same numerosity obtaining a total of 1000 and 90 tuning curves for the DBN (Fig. S7) and the HCNN (Fig. S8), respectively.

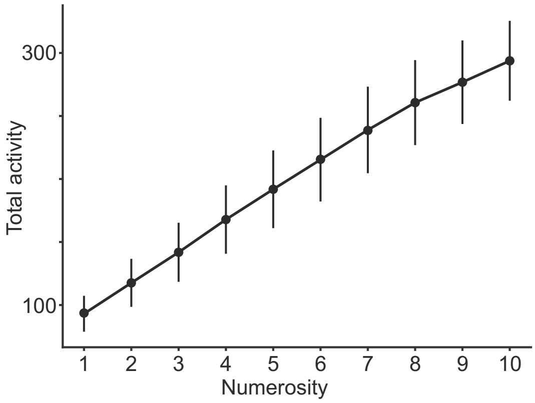

Fig. 4. Total activity of all 1000 units in the second hidden layer of the DBN as a function of the numerosity of the input image. The total activity was calculated by first summing up the 1000 single-neuron activities generated by each image across the second hidden layer and then averaging these quantities across all images belonging to the same numerosity. Vertical bars represent the standard deviation associated with these averages.

In the case of the DBN, single neural units at the second hidden layer exhibited graded 'summation unit' responses with high response variability rather than units selectively responding to particular numerosities (Fig. 5). Given that the DBN was not trained with the specific objective of classifying the visual stimuli by their numerosity labels, the lack of numerosity-selective units may not be surprising. It may also be an indication that the network estimated numerosity indirectly from an encoding of associated statistical features such as the size or area of shapes in the image (see section 3.4). Regardless, while numerosity-

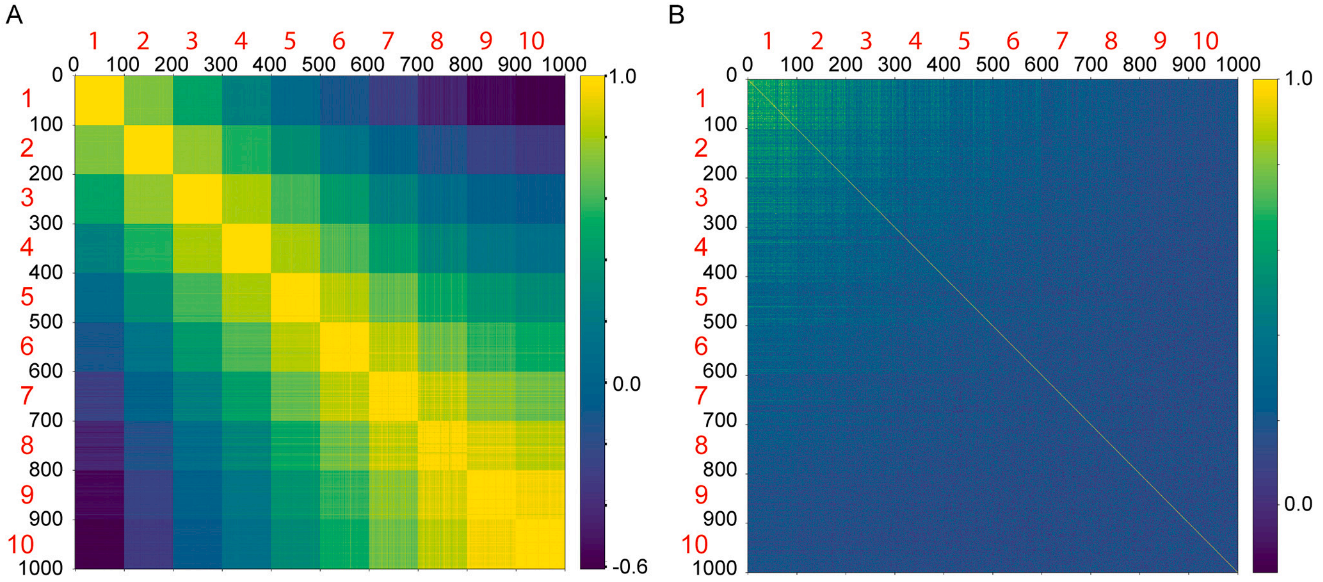

Fig. 3. Representational similarity matrix for the HCNN (A) and the DBN (B). The representational similarity matrix plot shows the correlation $r_{i,j}$, colour coded from low values in dark blue to high values in yellow, between the neural activities developed at the last layer of the network in response to test images i and j. The total number of test images was 1000 (black number labels), with 100 images per numerosity (red number labels). A: The block pattern shows that the population activity was very similar for different images with the same number of squares in them, $r_{i,j} \approx 1$ when images i and j had the same numerosity (red labels); in contrast the activity was gradually more dissimilar for images depicting different numbers of squares. B: In the case of the DBN, no clear network states could be identified and the correlation $r_{i,j}$ was low both when images contained the same number of items and when they depicted different numerosities. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

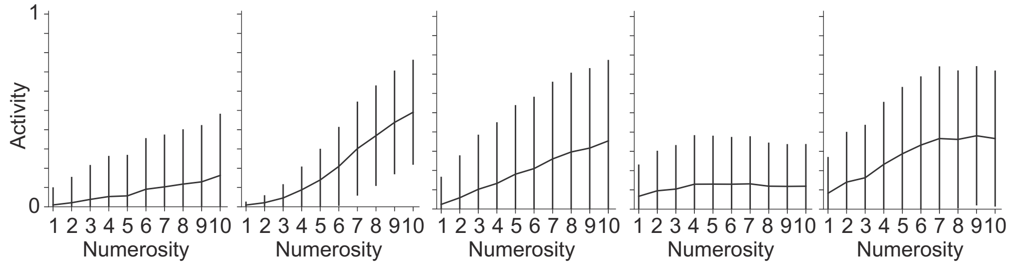

Fig. 5. Tuning curves for 5 of 1000 units in the last hidden layer of the deep belief network. Graphs denote mean activity and error bars denote standard deviation. See also Fig. S7. Note the gradual increase in activity as a function of numerosity for most units and the large variability in all units.

selective tuning has been reported in other models of approximate number sense (e.g. Nasr et al., 2019), these results extend previous work (e.g. Stoianov & Zorzi, 2012) in showing that monotonic tuning curves are sufficient to produce approximate number judgements and the behavioural performance profile predicted by a more sophisticated number-line code (Fig. 2).

Fig. 6 shows tuning curves extracted from the last layer of the HCNN for 19 neural units preferentially responding to distinct numerosities. Half of the sharply selective units were activated for numerosities larger than a particular cardinal number, while the other half was activated for numerosities smaller than a particular cardinal number. About 10% of the units were not selective for numerosity. Collectively, the units encoded the exact numerosity of the input image. The tuning curve analyses indicate that the HCNN learned to extract and represent numerosity in a form consistent with the *numerosity code* proposed by Zorzi and Butterworth (1999), also described as a ‘thermometer code’ or ‘cardinality code’. The representation reflects the property of cardinal numbers that each preceding number is contained in the succeeding number, i.e. 1 is part of 2, which is part of 3, and so on. Tuning curves for the entire population of units in the last layer of the HCNN are shown in Fig. S8.

Having identified that numerosity was represented in the HCNN as units responding to numerosities ‘larger than’ or ‘smaller than’ different preferred cardinal numbers, we asked exactly how this code was used to compute the correct numerosity labels in the output. The pattern of connections between different units in the final hidden layer and the output units Softmax classifier provides insight into this mechanism. Fig. 7 shows a visual representation of the connection weights from the last hidden layer to the output layer using a Hinton diagram (see Section 2). The weight matrix shows that units in the last hidden layer that

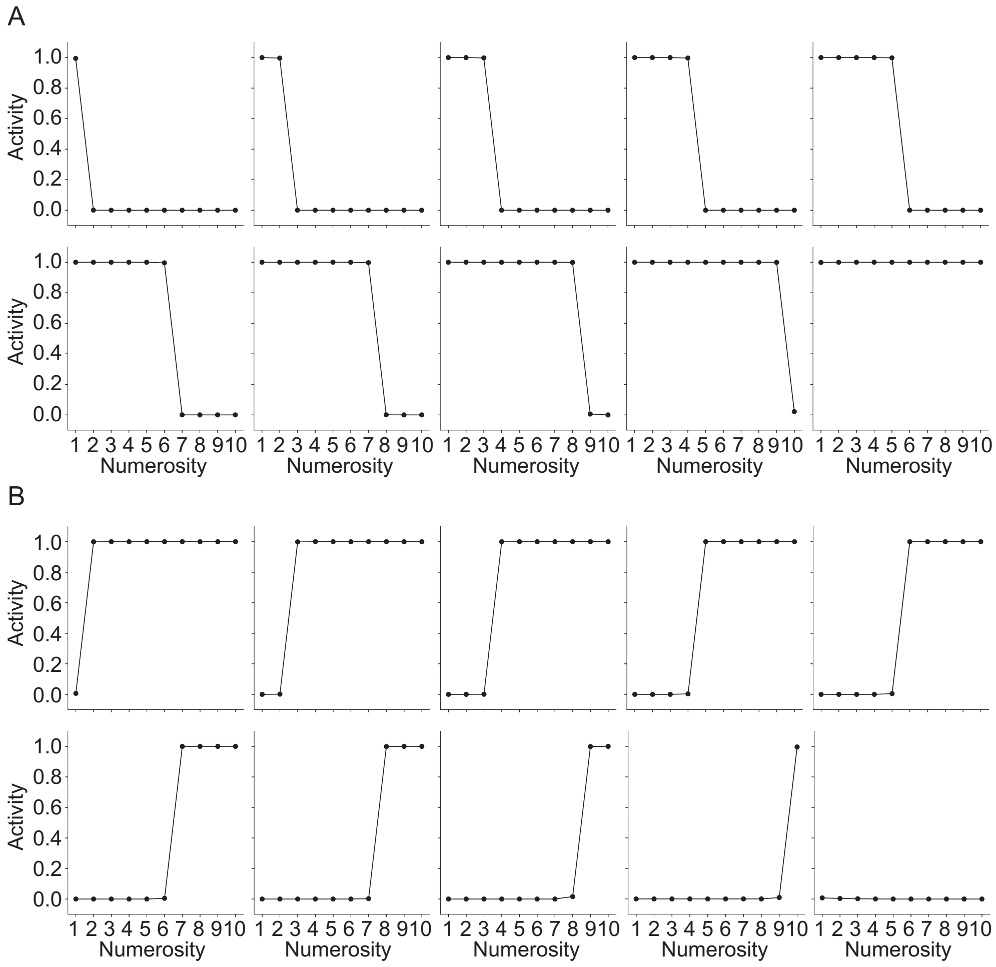

Fig. 6. Hierarchical convolutional neural network (HCNN) tuning curves for 19 of 90 of the units in the last hidden layer of the HCNN which respond to different numerosities. Graphs show the neural activity averaged over 100 images of each numerosity (y-axis) as a function of numerosity depicted in the input images (x-axis). A: Nine units selectively activated by images representing less than a particular numerosity, and one non-selective unit. B: Nine units selectively activated by images representing more than a particular numerosity, and one non-selective unit.

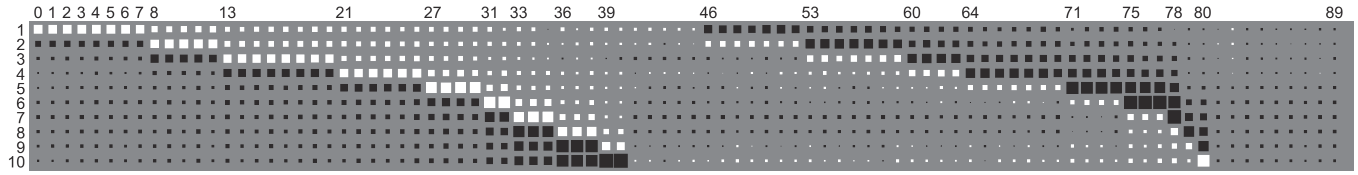

Fig. 7. Hinton diagram (see Section 2) of the connection weights that connect the last hidden layer with the output units of the hierarchical convolutional neural network (HCNN). Numbers in the left column (1−10) indicate the output labels corresponding to different output numbers of the Softmax classifier. Numbers in the top row represent units of the final hidden layer of the HCNN connecting to the output layer, and corresponding tuning curves are shown in Fig. S8.

represented ‘≤*n*’ objects in the input image activated output units representing numerosity labels ‘≤*n*’ and inhibited output units representing numerosity labels ‘>*n*’. Units in the final hidden layer that represented ‘≥*n*’ objects in the input image inhibited output units representing numerosity labels ‘<*n*’. The resulting computation activated the appropriate output cardinality units via a superposition principle, which is a familiar computational principle of many brain systems.

As an example, assume that the network has been given an image with 3 squares. Units that respond to numerosity ‘≤3’, such as unit $N=14$ in Fig. S8, will excite output units 1–3 and inhibit output units 4–10. Units that respond to numerosity ‘≥3’, such as unit $N=54$ in Fig. S8, will inhibit output units 1–2. Thus, only output unit 3 would receive a positive net activation.

This qualitative analysis of the network's functionality can be tested on the behavioural outcome it predicts. To test the functional relevance of the subset of units tuned to a specific numerosity, we silenced all units whose tuning curves switched activity state for each of the cardinal numbers in turn (3–15 units per numerosity; see Section 2). The classification performance then dropped to 0% for the silenced number without affecting performance for other numbers (Fig. 8). Note that silencing the only unit responding selectively to images of numerosity 10 caused only a marginal drop in the network's classification performance. However, when we silenced all ‘≥*n*’ units, classification performance dropped to 0% exclusively for numerosity 10, confirming the impression from the Hinton diagram (Fig. 7) that the corresponding output unit was activated when all other output units were inhibited by ‘≤*n*’ units.

3.4. Non-numerical features

In light of a recent and hot debate concerning the role of non-numerical perceptual cues in numerosity judgements (see e.g. Gebuis, Cohen Kadosh, & Gevers, 2016; Wilkey & Ansari, 2019), we asked whether the network models were primarily sensitive to numerosity per se, or, rather, to non-numerical features that co-vary with numerosity, such as the total area of white shapes in the images. We first analysed how single unit activity was associated with numerosity, and next we directly tested how network performance was affected by the total area of white shapes in the images in different datasets.

On the level of single units, plotting neural activity in the final HCNN layer as a function of total surface area of the squares (Fig. S9) or as a function of convex hull of the squares (Fig. S10) resulted in broad tuning curves with gradual rather than abrupt transitions. Moreover, units responding to the same numerosities (Fig. S8) also responded identically when input images varied as a function of either area (Fig. S9) or convex hull (Fig. S10), essentially distinguishing 10 discrete network states. This encoding would not be expected if the network coded for continuous variables such as total area or convex hull of the squares. The sharp tuning curves with abrupt transitions at a preferred numerosity (Fig. 6) were distinctively associated with tracing activity as a function of numerosity and suggest that the HCNN was primarily selective to numerical information.

For the DBN, both individual tuning curves (Fig. 5) and cumulative network activity (Fig. 4) displayed high variability in response to images of the same numerosity, indicating that variables other than numerosity affected the representation.

In the dataset of square-pattern images, the number of squares correlated with non-numerical variables such as the total area of squares (the total number of white pixels) in the images (see Section 2). To investigate the extent to which network performance could be explained by the total area of squares in the images, we calculated the probability that a square-pattern image belonged to each numerosity given the total area of squares in the image. From Fig. 9A we see that while the performance of the HCNN seemed unrelated to this area-adjusted baseline, the DBN performed only slightly better than the level expected if classification was based exclusively on the total area of squares in the

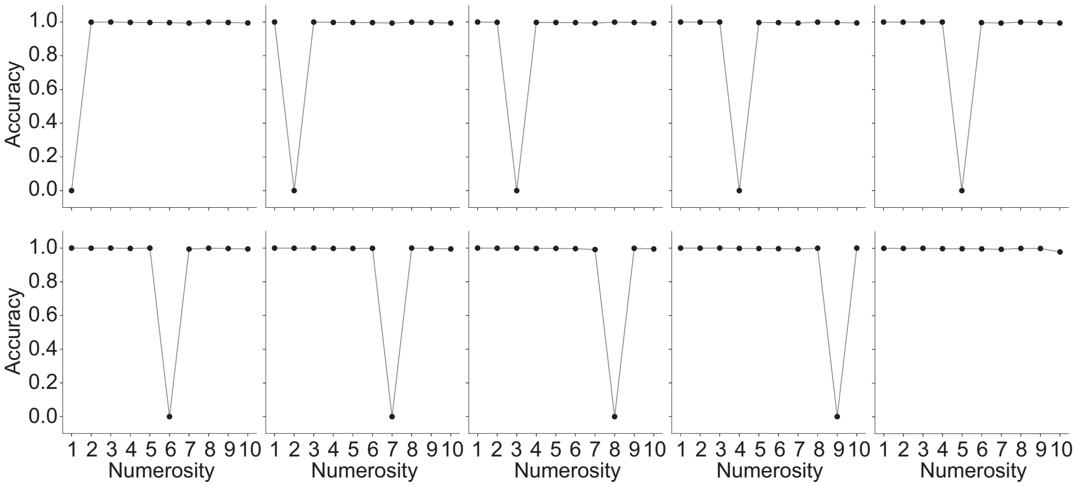

Fig. 8. Effect of silencing units in the hierarchical convolutional neural network (HCNN) selectively activating a particular output numerosity label. Single-numerosity classification accuracy of the HCNN is shown on the y-axis. When all the neural units selectively activating a single number label were silenced, the accuracy dropped to 0% for the *silenced* number, while the accuracy for all other numbers remained virtually unaffected in all cases.

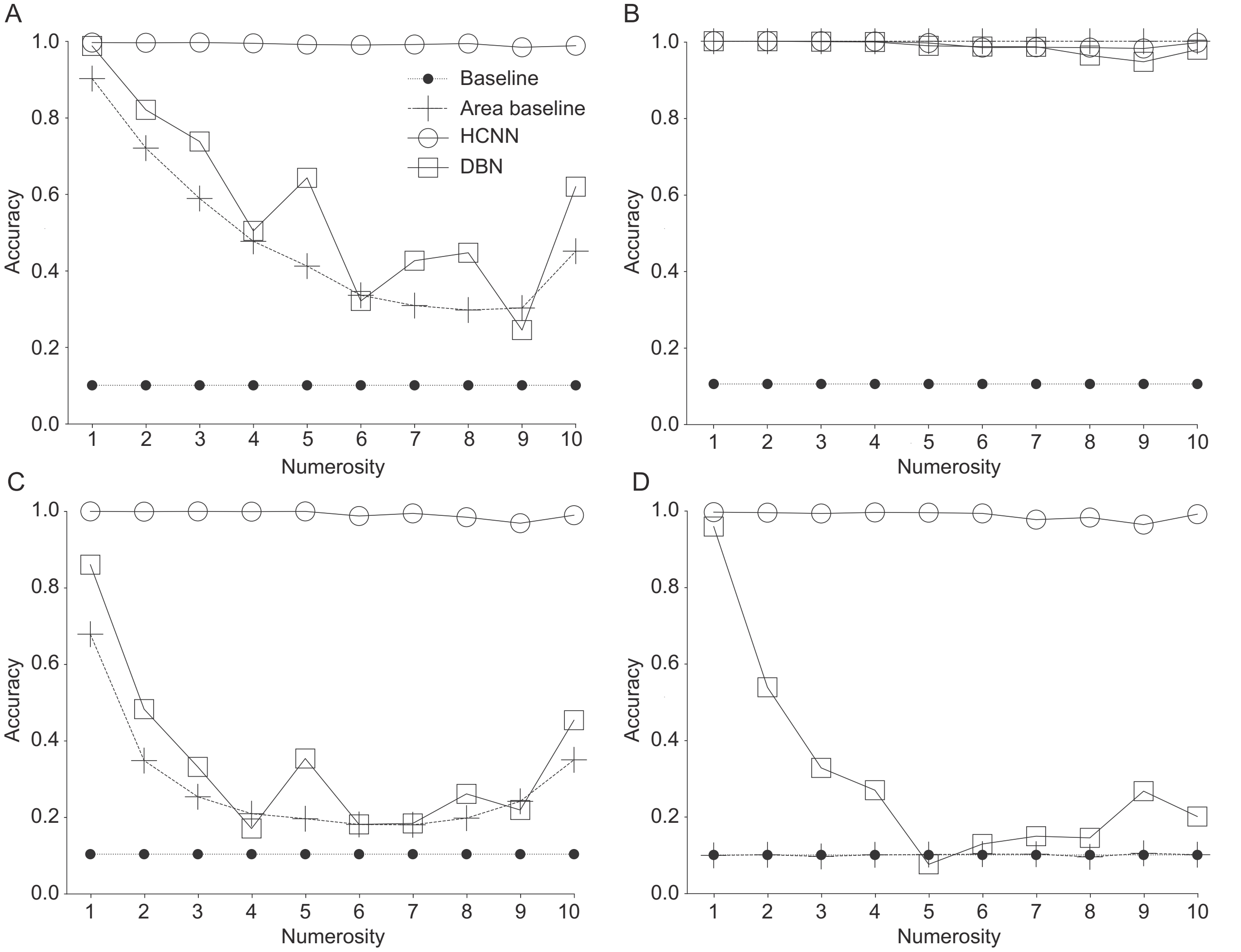

Fig. 9. Network performance compared to baseline and area-adjusted baseline. The baseline corresponds to inferring numerosity only from the number of numerosity categories (10% chance level). The area-adjusted baseline corresponds to inferring numerosity only from the total area of shapes in each image and depends on the distribution of images in the dataset (See Section 2). The networks were trained and tested on four different datasets: A: squares of random sizes (Fig. S1); B: squares of identical size (9 pixels per square), with perfect correlation between numerosity and area (Fig. S2); C: rectangles of random sizes (Fig. S3); and D: rectangles of identical total area (64 pixels per image), with no correlation between numerosity and area (Fig. S4).

images. These results consolidate the impression that the DBN's performance could, to a large degree, be explained by a mapping of area to numerosity and is consistent with the perspective that the system developed sensitivity to numerosity through performing sensory integration rather than extracting numerosity per se (Gebuis et al., 2016).

When trained and tested on a new dataset consisting of white squares of identical size, and thus a one-to-one mapping between the total area of white squares and the numerosity of an image, both networks reached near-perfect performances (Fig. 9B). This result shows that the DBN could perform at a level comparable to the HCNN for less abstract, non-numerical variables but does not show whether the DBN could also develop sensitivity to numerosity without relying on area.

The effect of area on network performance can be more directly tested in a dataset where all images have the same total area of white shapes. However, for square shapes, it is not possible to make images with a common total area for all numerosities. We therefore generated two additional datasets of rectangles: one dataset with images of rectangles of random sizes, and one dataset with images in which the total area of rectangles was fixed at 64 pixels for all numerosities. The identical total area of all images ensured that there was no correlation between area and numerosity in this dataset (see Section 2 and supplementary material).

When trained and tested on the dataset with randomly sized rectangles, network performance did not deviate much from that of the dataset with randomly sized squares for either network (Fig. 9C). When trained and tested on the dataset with rectangles of identical total area, the HCNN preserved its near-perfect performance. The accuracy of the DBN was close to chance level for numerosities larger than four, but still identified smaller numerosities (Fig. 9D). Similar results were obtained when the networks were trained on the dataset with rectangles of random sizes and tested on the dataset with rectangles with a fixed, identical area (Fig. S11).

We conclude that while the HCNN learned an area-invariant representation of numerosity, the DBN relied more strongly on non-numerical features, especially for large numerosities. It is still possible that other non-numerical features supported the performance of either network.

4. Discussion

In this paper we investigated if and how learning of small symbolic numbers can be supported by general learning principles of hierarchical neural networks. We trained two networks of different architectures and learning algorithms to classify the same input dataset of dot-pattern images using the same output classifier.

A deep belief network (DBN) trained to generate dot-pattern images without associated labels learned to estimate the number of squares at an overall accuracy of 60% and a single-number accuracy that declined for increasing numerosity. This result is in qualitative agreement with behavioural data from pre-counting toddlers (Sella et al., 2016) as well as with other recent proposals simulating the task of discriminating between pairs of numerosities with similar network models (Chen et al., 2018; Testolin, Zou, & McClelland, 2020; Zorzi & Testolin, 2018). The hierarchical convolutional neural network (HCNN), trained to directly classify the input images by numerosity labels, learned to exactly enumerate the squares with near-perfect performance. This behaviour is reminiscent of subitizing, i.e. the direct perception of the number of items in a set, whose neurobiological underpinnings are currently largely unknown. Underlying the difference in function and performance of the two networks, qualitatively distinct mechanisms for recognising numerosity emerged at the level of neural representations. The final layer of both networks encoded statistics of numerosity in the input images in such a way that numerosity could be extracted to a much higher degree than from the raw images directly. However, the encoding of numerosity information differed radically between the networks.

The DBN developed a dense population representation of numerosity-sensitive neural units with monotonically increasing response profiles called summation units (Dehaene & Changeux, 1993). Contrary to previous models of approximate number discrimination (Nasr et al., 2019; Zorzi & Testolin, 2018), we did not find symmetric, numerosity-selective neural units in the DBN. However, the summation code was sufficient for producing the characteristics of a logarithmic number-line code (Dehaene, 2011) in the output units of the simple Softmax classification layer (Fig. 2). Using a 10-class classification task rather than a binary discrimination task allowed us to directly demonstrate the behavioural response profile of the approximate number system in the output classification layer of the DBN, which also qualitatively matched single neuron activity in primates (Viswanathan & Nieder, 2013). The logarithmic representation of number is efficient in the sense that a large range of numerosities can be represented with relatively few neural resources. However, it lacks the single-object resolution for large numerosities that is necessary for establishing one-to-one correspondences between objects and number words and for the exact enumeration required for the concept of natural numbers.

In contrast to the DBN, the HCNN developed a sparse population code for numerosity in which single neurons abruptly switched their responses for a particular, preferred numerosity (Fig. 6). This representation effectively implements a thermometer-like numerosity code (Zorzi & Butterworth, 1999), that captures the compositional principle of natural numbers: that each numerosity (natural number) includes the smaller numerosities. The single-unit response profiles were distinct from the Gaussian-like tuning curves of number-selective neurons reported in experimental data (Viswanathan & Nieder, 2013) and models of approximate number perception (Nasr et al., 2019; Zorzi & Testolin, 2018). However, the numerosity code enables the exact numerosity of the input image to be clearly distinguished by the population of neural units and allows two important properties necessary for the conceptual development of natural numbers: i) establishing one-to-one correspondence between numerosity categories and objects in the real world, and ii) distinguishing transformations that are invariant to numerosity (Butterworth, 2010).

A potential strength of the summation code that developed in the DBN is that it might more easily generalise to an extended range of numerosities. Training a new output classifier on top of the summation code would likely suffice to approximately estimate an extended range of numerosities, e.g. from one through twenty. In comparison, the representation that developed in the HCNN was closely tailored to the input training set, having groups of neural units uniquely representing single numerosities. Exactly classifying an extended range of input numerosities with the same HCNN would require restructuring of the existing representation through retraining of the entire network. Alternatively, supporting an extended range of input numerosities could be achieved by retraining a subgroup of neural units or recruiting additional neural units into the code. Such plasticity processes might be reasonable from the point of view of a neurobiological mechanism or of an extended neural network model with a mechanism for continual learning (e.g. Kirkpatrick et al., 2017).

Although the brain is unlikely to implement details of the network architectures and learning algorithms of the simulations in this study, the results of such simulations are still relevant to understanding cognitive function. For example, the current simulations contribute to our understanding of how specific behaviours are supported by the development of specific neural representations. In this paper we showed how exact number perception can emerge from developing neural representations that correspond to the cardinality of a set of items and that language in the form of number labels may be key to developing exact number concepts. Whether humans and non-human animals that can classify and label exact numerosities also develop a precise neural numerosity code remains to be settled by neurophysiological experiments. Directly comparing different computational models trained on the same input data provides a useful approach to investigating what architectural components and learning processes are important to numerosity perception and will ultimately allow us to better understand the basic principles that shape our numerical cognition.

While human subitizing is typically limited to four objects, the HCNN learned to perfectly classify numerosities up to ten. If a general neural network mechanism underlies human subitizing, one might expect an effect of practice on the subitizing range. There is currently limited empirical data on the effect of extended practice on human subitzing performance. However, in some schools, subitizing is routinely practiced and extended to a form of rapid arithmetic reasoning called ‘conceptual subitizing’ or ‘groupitizing’ (Clements, 1999; Clements et al., 2019; Starkey & Mccandliss, 2014; Wender & Rothkegel, 2000). By learning to recognise the numerosity of an image as a combination of sub-patterns of smaller numerosities, elementary school children can rapidly perceive numerosities up to 10 and beyond without counting (Clements, 1999). Indeed, some early 20th century curricula considered subitizing a prerequisite to learning the counting algorithm, with the benefit of being faster and less error-prone in the initial phases of number learning (Clements et al., 2019). Conceptual subitizing is rapidly gaining interest in primary education for its potential to strengthen arithmetic fluency and foster a conceptual understanding of the compositional nature of numbers (Clements et al., 2019; Clements & Sarama, 2014; Özdem & Olkun, 2019). A natural further step towards a computational understanding of mathematical cognition would be to investigate the development of a similar rudimentary form of mathematical reasoning in neural networks or other computational model systems.

An ongoing debate about whether and how perceptual number sense contributes to mathematics achievement is influencing educational policy (e.g. Butterworth, 2010; Dehaene, 2011; Siegler & Braithwaite, 2017). Children's performances in approximate number judgement tasks have been conjectured to influence, and reported to correlate with, mathematical proficiency (e.g. Feigenson, Libertus, & Halberda, 2013; Halberda, Mazzocco, & Feigenson, 2008; Siegler & Braithwaite, 2017; Starr, Libertus, & Brannon, 2013). However, studies taking other variables, such as attention, into account suggest that exact enumeration, but not approximate estimation, is directly related to mathematical performance (Libertus, 2019; Soltész, Szücs, & Szücs, 2010; Szűcs, Devine, Soltész, Nobes, & Gabriel, 2014). It is possible that the direct perception and classification of exact numerosity through subitizing constitutes a cognitive foundation for the conceptual development of natural numbers as well as learning elementary arithmetic and more advanced symbolic mathematics (Carey, 2001; Dehaene, 2011; Siegler & Braithwaite, 2017). However, the causal relationship between number sense and mathematics achievement is likely to be more complex. Developing proficiency with symbolic numbers extends over many years and likely involves an intricate interplay between subitizing, analogue and approximate number processing, and higher cognitive processes such as working memory and attention which might change considerably over time (e.g. Szűcs et al., 2014).

In conclusion, we have compared two neural network models and seen that they develop different internal representations that support qualitatively different numerosity perception systems—exact enumeration and approximate estimation. The results support the theory that parallel neural-level mechanisms underlie exact and approximate number sense (Piazza, 2010). Generic learning mechanisms in hierarchical neural networks are sufficient to reproduce key features of the approximate visual perception of numbers, consistent with the view that our approximate number sense may arise naturally from sensory integration (Gebuis et al., 2016). At the same time, rather than being a Kantian a priori concept (Siegler & Braithwaite, 2017), and consistent with the observation that subitizing is not reserved for the visual modality (Camos & Tillmann, 2008; Riggs et al., 2006), exact enumeration can also emerge from sensory experience through a general pattern recognition mechanism that is efficiently supported by language in the form of labelled examples.

Funding

This work was funded by the Research Council of Norway under grant 283441.

Author note

This work only includes original figures.

Ethical approval

No particular ethics approval was required for this work.

Declaration of Competing Interest

The authors have no known conflict of interest to disclose.

Acknowledgements

We would like to thank Alberto Testolin, James L. McClelland, Gerald Goldin, and Keith Downing for discussions and comments on an earlier draft of this manuscript.

Appendix A. Supplementary data

Supplementary material to this article can be found online at https://doi.org/10.1016/j.cognition.2021.104815.

References

Agrillo, C., Piffer, L., Bisazza, A., & Butterworth, B. (2012). Evidence for two numerical systems that are similar in humans and guppies. *PLoS One, 7*. https://doi.org/10.1371/journal.pone.0031923.

Butterworth, B. (2010). Foundational numerical capacities and the origins of dyscalculia. *Trends in Cognitive Sciences, 14*, 534–541. https://doi.org/10.1016/j.tics.2010.09.007.

Camos, V., & Tillmann, B. (2008). Discontinuity in the enumeration of sequentially presented auditory and visual stimuli. *Cognition, 107*, 1135–1143. https://doi.org/10.1016/j.cognition.2007.11.002.

Carey, S. (2001). Cognitive foundations of arithmetic: Evolution and ontogenisis. *Mind & Language, 16*, 37–55. https://doi.org/10.1111/1468-0017.00155.

Chen, S., Zhou, Z., Fang, M., McClelland, J. L., Rogers, T. T., Rau, M., … Rogers, T. T. (2018). Can generic neural networks estimate numerosity like humans?. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Clements, D. (1999). Subitizing: What is it? Why teach it? *Teaching Children Mathematics, 5*, 400–405.

Clements, D., Sarama, J., & Macdonald, B. (2019). *Subitizing: The neglected quantifier: merging perspectives from psychology and mathematics education* (pp. 13–45). https://doi.org/10.1007/978-3-030-00491-0_2.

Clements, D. H., & Sarama, J. (2014). *Learning and teaching early math: The learning trajectories approach* (2nd ed.). New York: Taylor & Francis.

Dehaene, S. (2001). Subtracting pigeons: Logarithmic or linear? *Psychological Science, 12*, 244–246. https://doi.org/10.1111/1467-9280.00343.

Dehaene, S. (2011). *The number sense: How the mind creates mathematics, revised and (Updated ed.)*. USA: Oxford University Press.

Dehaene, S., & Changeux, J. P. (1993). Development of elementary numerical abilities: A neuronal model. *Journal of Cognitive Neuroscience, 5*, 390–407. https://doi.org/10.1162/jocn.1993.5.4.390.

Dehaene, S., Piazza, M., Pinel, P., & Cohen, L. (2003). Three parietal circuits for number processing. *Cognitive Neuropsychology, 20*, 487–506. https://doi.org/10.1080/02643290244000239.

Feigenson, L., Libertus, M. E., & Halberda, J. (2013). Links between the intuitive sense of number and formal mathematics ability. *Child Development Perspectives, 7*, 74–79. https://doi.org/10.1111/cdep.12019.

Gallistel, C. R., & Gelman, R. (1992). Preverbal and verbal counting and computation. *Cognition, 44*, 43–74. https://doi.org/10.1016/0010-0277(92)90050-R.

Gebuis, T., Cohen Kadosh, R., & Gevers, W. (2016). Sensory-integration system rather than approximate number system underlies numerosity processing: A critical review. *Acta Psychologica, 171*, 17–35. https://doi.org/10.1016/j.actpsy.2016.09.003.

Halberda, J., Mazzocco, M., & Feigenson, L. (2008). Individual differences in non-verbal number acuity correlate with maths achievement. *Nature, 455*, 665–668. https://doi.org/10.1038/nature07246.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*, 1771–1800. https://doi.org/10.1162/089976602760128018.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*, 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*, 504–507. https://doi.org/10.1126/science.1127647.

Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review, 98*, 74. https://doi.org/10.1037/0033-295x.98.1.74.

Izard, V. R., Sann, C., Spelke, E. S., & Streri, A. (2009). Newborn infants perceive abstract numbers. *Proceedings of the National Academy of Sciences of the United States of America, 106*, 10382–10385. https://doi.org/10.1073/pnas.0812142106.

Jevons, W. (1871). The power of numerical discrimination. *Nature, 3*, 281–282. https://doi.org/10.1038/003281a0.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences, 114*, 3521–3526. https://doi.org/10.1073/pnas.1611835114.

Kriegeskorte, N., Mur, M., & Bandettini, P. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience, 2*, 4. https://doi.org/10.3389/neuro.06.004.2008.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. Technical Report. URL http://code.google.com/p/cuda-convnet/.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*, 541–551. https://doi.org/10.1162/neco.1989.1.4.541.

Libertus, M. E. (2019). Understanding the link between the approximate number system and math abilities. In *Cognitive Foundations for Improving Mathematical Learning* (pp. 91–106). Elsevier. https://doi.org/10.1016/b978-0-12-815952-1.00004-9.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews. Neuroscience, 21*, 335–346. https://doi.org/10.1038/s41583-020-0277-3.

Nasr, K., Viswanathan, P., & Nieder, A. (2019). Number detectors spontaneously emerge in a deep neural network designed for visual object recognition. *Science Advances, 5*, 1–11. https://doi.org/10.1126/sciadv.aav7903.

Nieder, A. (2005). Counting on neurons: The neurobiology of numerical competence. *Nature Reviews Neuroscience, 6*, 177–190. https://doi.org/10.1038/nrn1626.

Nieder, A. (2016). The neuronal code for number. *Nature Reviews Neuroscience, 17*, 366–382. https://doi.org/10.1038/nrn.2016.40.

Nieder, A., & Dehaene, S. (2009). Representation of number in the brain. *Annual Review of Neuroscience, 32*, 185–208. https://doi.org/10.1146/annurev.neuro.051508.135550.

Özdem, S., & Olkun, S. (2019). Improving mathematics achievement via conceptual subitizing skill training. *International Journal of Mathematical Education in Science and Technology*. https://doi.org/10.1080/0020739X.2019.1694710.

Piazza, M. (2010). Neurocognitive start-up tools for symbolic number representations. *Trends in Cognitive Sciences, 14*, 542–551. https://doi.org/10.1016/j.tics.2010.09.008.

Riggs, K., Ferrand, L., Lancelin, D., Fryziel, L., Dumur, G., & Simpson, A. (2006). Subitizing in tactile perception. *Psychological Science, 17*, 271–272. https://doi.org/10.1111/j.1467-9280.2006.01696.x.

Rugani, R., Regolin, L., & Vallortigara, G. (2008). Discrimination of small numerosities in young chicks. *Journal of Experimental Psychology. Animal Behavior Processes, 34*, 388–399. https://doi.org/10.1037/0097-7403.34.3.388.

Sella, F., Berteletti, I., Lucangeli, D., & Zorzi, M. (2016). Spontaneous non-verbal counting in toddlers. *Developmental Science, 19*, 329–337. https://doi.org/10.1111/desc.12299.

Siegler, R. S., & Braithwaite, D. W. (2017). Numerical development. *Annual Review of Psychology, 68*, 187–213. https://doi.org/10.1146/annurev-psych-010416-044101.

Soltész, F., Szücs, D., & Szücs, L. (2010). Relationships between magnitude representation, counting and memory in 4-to 7-year-old children: A developmental study. *Behavioral and Brain Functions, 6*, 13. https://doi.org/10.1186/1744-9081-6-13.

Starkey, G., & Mccandliss, B. (2014). The emergence of "groupitizing" in children's numerical cognition. *Journal of Experimental Child Psychology, 126*, 120–137. https://doi.org/10.1016/j.jecp.2014.03.006.

Starr, A., Libertus, M. E., & Brannon, E. M. (2013). Number sense in infancy predicts mathematical abilities in childhood. *Proceedings of the National Academy of Sciences, 110*, 18116–18120. https://doi.org/10.1073/pnas.1302751110.

Stoianov, I., & Zorzi, M. (2012). Emergence of a "visual number sense" in hierarchical generative models. *Nature Neuroscience, 15*, 194–196. https://doi.org/10.1038/nn.2996.

Szűcs, D., Devine, A., Soltész, F., Nobes, A., & Gabriel, F. (2014). Cognitive components of a mathematical processing network in 9-year-old children. *Developmental Science*, 506–624. https://doi.org/10.1111/desc.12144.

Testolin, A., Dolfi, S., Rochus, M., & Zorzi, M. (2020). Visual sense of number vs. sense of magnitude in humans and machines. *Scientific Reports, 10*, 1–13. https://doi.org/10.1038/s41598-020-66838-5.

Testolin, A., Stoianov, I., De Filippo De Grazia, M., & Zorzi, M. (2013). Deep unsupervised learning on a desktop PC: A primer for cognitive scientists. *Frontiers in Psychology, 4*, 251. https://doi.org/10.3389/fpsyg.2013.00251.

Testolin, A., Zou, W. Y., & McClelland, J. L. (2020). Numerosity discrimination in deep neural networks: Initial competence, developmental refinement and experience statistics. *Developmental Science*. https://doi.org/10.1111/desc.12940.

Tomonaga, M., & Matsuzawa, T. (2002). Enumeration of briefly presented items by the chimpanzee (pan troglodytes) and humans (homo sapiens). *Animal Learning & Behavior, 30*, 143–157. https://doi.org/10.3758/BF03192916.

Verguts, T., & Fias, W. (2004). Representation of number in animals and humans: A neural model. *Journal of Cognitive Neuroscience, 16*, 1493–1504. https://doi.org/10.1162/0898929042568497.

Verguts, T., Fias, W., & Stevens, M. (2005). A model of exact small-number representation. *Psychonomic Bulletin and Review, 12*, 66–80. https://doi.org/10.3758/BF03196349.

Viswanathan, P., & Nieder, A. (2013). Neuronal correlates of a visual "sense of number" in primate parietal and prefrontal cortices. *Proceedings of the National Academy of Sciences, 110*, 11187–11192. https://doi.org/10.1073/pnas.1308141110.

Wender, K., & Rothkegel, R. (2000). Subitizing and its subprocesses. *Psychological Research, 64*, 81–92. https://doi.org/10.1007/s004260000021.

Wilkey, E. D., & Ansari, D. (2019). Challenging the neurobiological link between number sense and symbolic numerical abilities. *Annals of the New York Academy of Sciences, 1462*, 76–98. https://doi.org/10.1111/nyas.14225.

Zorzi, M., & Butterworth, B. (1999). *A computational model of number comparison, in: Proceedings of the twenty first annual conference of the cognitive science society* (pp. 772–777). Mahwah, NJ: Erlbaum.

Zorzi, M., & Testolin, A. (2018). An emergentist perspective on the origin of number sense. *Philosophical Transactions of the Royal Society, B: Biological Sciences, 373*. https://doi.org/10.1098/rstb.2017.0043.