PNAS RESEARCH ARTICLE | COMPUTER SCIENCES NEUROSCIENCE

Data-driven emergence of convolutional structure in neural networks

Alessandro Ingrosso[a,1] and Sebastian Goldt[b,1]

Edited by Scott Kirkpatrick, The Hebrew University of Jerusalem, Jerusalem, Israel; received February 3, 2022; accepted August 12, 2022 by Editorial Board Member Terrence J. Sejnowski

Exploiting data invariances is crucial for efficient learning in both artificial and biological neural circuits. Understanding how neural networks can discover appropriate representations capable of harnessing the underlying symmetries of their inputs is thus crucial in machine learning and neuroscience. Convolutional neural networks, for example, were designed to exploit translation symmetry, and their capabilities triggered the first wave of deep learning successes. However, learning convolutions directly from translation-invariant data with a fully connected network has so far proven elusive. Here we show how initially fully connected neural networks solving a discrimination task can learn a convolutional structure directly from their inputs, resulting in localized, space-tiling receptive fields. These receptive fields match the filters of a convolutional network trained on the same task. By carefully designing data models for the visual scene, we show that the emergence of this pattern is triggered by the non-Gaussian, higher-order local structure of the inputs, which has long been recognized as the hallmark of natural images. We provide an analytical and numerical characterization of the pattern formation mechanism responsible for this phenomenon in a simple model and find an unexpected link between receptive field formation and tensor decomposition of higher-order input correlations. These results provide a perspective on the development of low-level feature detectors in various sensory modalities and pave the way for studying the impact of higher-order statistics on learning in neural networks.

neural networks | convolution | receptive fields | invariance

Exploiting invariances in data is crucial for neural networks to learn efficient representations and to make accurate predictions. Translation invariance is a key symmetry in image processing and lies at the heart of feed-forward (1, 2) and recurrent (3, 4) models of the visual system. In the early sensory stage, the feature maps obtained by convolving a set of filters with an input arise from the collective action of localized receptive fields (RFs) organized in a tessellation pattern. The importance of RFs for understanding neural networks was recognized in the seminal work of Hubel and Wiesel (5) on the early stages of the visual system. RFs remain a key building block in theoretical neuroscience (6–8), from the statistical formulation of single-neuron encoding (9, 10) to hierarchical models of cortical processing in various sensory modalities (11, 12). A key question in neuroscience is how these RFs are developed and what mechanism drives their spatial organization. The computational inquiry into how RFs can originate from image statistics goes back to the seminal work of Olshausen and Field (13), who showed that a specific unsupervised learning algorithm maximizing sparseness of neural activity was sufficient for developing localized RFs, similar to those found in primary visual cortex.

In machine learning, convolutional neural networks (CNNs) (14) were inspired by the ideas of Hubel and Wiesel (5) and rely on linear convolutions, followed by nonlinear functions and pooling operations (15) that encourage translation invariance of the network output (16–18). CNNs classify images significantly better than vanilla, fully connected (FC) networks, which do not take this symmetry explicitly into account (19). Since their success in computer vision (15, 20–22), deep CNNs have served as a prime example for how encoding prior knowledge about data invariances into the network architecture can improve both sample and parameter efficiency of learning.

Subsequent work has since engineered architectures and representations capable of dealing with data characterized by different invariances and geometries, such as social or gene regulatory networks (23–31). These invariances, however, are not always known beforehand. Deep scattering networks (32–34) have been proposed as architectures that are invariant to a rich class of transformations. Another approach altogether would be to learn low-level feature detectors that take basic symmetries into account directly from data. In the case of images, the question thus becomes, Can we learn convolutions from scratch?

Significance

The interplay between data symmetries and network architecture is key for efficient learning in neural networks. Convolutional neural networks perform well in image recognition by exploiting the translation invariance of images. However, learning convolutional structure directly from data has proven elusive. Here we show how a neural network trained on translation-invariant data can autonomously develop a convolutional structure. Our work thus shows that neural networks can learn representations that exploit the data symmetries autonomously, by exploiting higher-order data statistics. We finally identify the maximization of non-Gaussianity as a guiding principle for representation learning in our model, linking discriminative vision tasks and unsupervised feature extraction.

Author affiliations: [a]Quantitative Life Sciences, The Abdus Salam International Centre for Theoretical Physics, 34151 Trieste, Italy; and [b]Department of Physics, International School of Advanced Studies, 34136 Trieste, Italy

Author contributions: A.I. initiatied the study; and A.I. and S.G. designed research, performed research, contributed new reagents/analytic tools, analyzed data, and wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. S.K. is a guest editor invited by the Editorial Board.

Copyright © 2022 the Author(s). Published by PNAS. This article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

[1]To whom correspondence may be addressed. Email: ingrosso@ictp.it or sgoldt@sissa.it.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2201854119/-/DCSupplemental.

Published September 26, 2022.

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

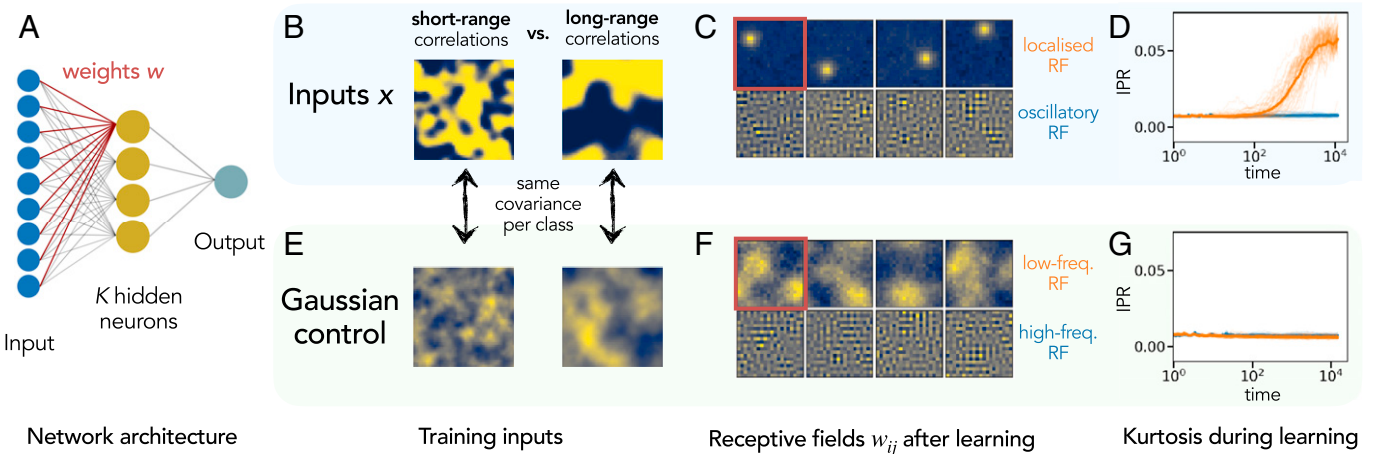

Fig. 1. The emergence of convolutional structure in FC neural networks is driven by higher-order input correlations. (*A*) Two-layer, FC neural network with K neurons in the hidden layer. (*B*) Networks are trained on a binary classification task with two-dimensional inputs $\boldsymbol{x} = (x_{ij})$ of size $D = L \times L$ drawn from a translation-invariant random process (Eq. **1**) with $L = 28$. The network has to discriminate inputs with different correlation lengths, $\xi^- = 0.1L$ (*Left*) and $\xi^+ = 0.2L$ (*Right*). (*C*) RFs of some representative neurons taken from a network with $K = 100$ neurons after training. The elements of the each weight vector are arranged in a $L \times L$ grid. Half the neurons develop localized RFs: the magnitude of their weights is significantly different from zero only in a small region of the input space. The other neurons converge to superpositions of two-dimensional Fourier components. (*D*) IPR (Eq. **2**) of each neuron during training. The IPR is large for localized RFs but remains small for oscillatory RFs. (*E*) Gaussian control dataset: the network is trained on a mixture of two Gaussians, each having zero mean and the same covariance as inputs in *B*. (*F*) RFs after training the network on the Gaussian control data. (*G*) IPR (Eq. **2**) of the RFs of a network trained on Gaussian data.

The hallmarks of convolutional structure that we are looking for are local connectivity, resulting in localized RFs, and the sharing of weights between neurons. Furthermore, we require that the local filters have to be applied across the whole image; i.e., the filters have to tile the sensory space. Uniform tiling of sensory space is crucial in our understanding of input processing in biological circuits, and a number of theoretical justifications have been given in terms of coding efficiency (35, 36).

FC layers are expressive enough to implement such convolutional structure, with weights that are sparse (due to locality) and redundant (due to weight sharing). The emergence of localized RFs has been recently shown in unsupervised models such as autoencoders (37, 38) and restricted Boltzmann machines (39) or with the use of similarity-preserving learning rules (40). However, learning convolutions directly from data by training an initially FC network on a discriminative task has so far proven elusive: FC networks do not develop any of the hallmarks of convolutions without tailor-made regularization techniques, and they perform significantly worse than convolutional networks (19, 41, 42). The problem thus lies in the learnability of the convolutional structure through the standard paradigm of machine learning (optimization of a cost function via first-order methods).

Here we show that FC neural networks can indeed learn a convolutional structure directly from their inputs if trained on data with non-Gaussian, higher-order local structure. We design a supervised classification task that fulfils these criteria and show that the higher-order statistics of the inputs can drive the emergence of localized, space-tiling RFs.

Results

Fully Connected Networks Can Learn Localized RFs from Scratch. In our first experiment, we trained a simple two-layer neural network with K neurons in the hidden layer (Fig. 1*A*) on a synthetic data set with two-dimensional inputs $\boldsymbol{x} = (x_{ij})$ of size $D = L \times L$ as in Fig. 1*B*. We generated inputs by first drawing a random vector $\boldsymbol{z} = (z_{ij})$ from a centered Gaussian distribution with a covariance that renders the input distribution translation invariant along both dimensions. Each pixel in the synthetic image x_{ij} is then computed as

$$x_{ij} = \frac{\psi\left(g z_{ij}\right)}{Z(g)}, \qquad [1]$$

where $\psi(\cdot)$ is a symmetric, saturating nonlinear function such as the error function, $g > 0$ is a gain factor, and the normalization constant $Z(g)$ ensures that pixels have unit variance for all values of g (see *Materials and Methods* for details). Intuitively, the gain factor controls the sharpness in the images: a large gain factor results in images with sharp edges and important non-Gaussian statistics (Fig. 1*B*), while images with a small gain factor are close to Gaussians in distribution.

Inputs are divided in $M = 2$ classes, labeled $y = \pm 1$, that differ by the correlation length $\xi^{\pm}$ between pixels: the image shown in Fig. 1 *B*, Left, has a shorter correlation length than the one in Fig. 1 *B*, *Right*; hence, the input in Fig. 1 *B*, *Left*, varies more rapidly in space. The learning task consists in discriminating inputs based on these correlation lengths.

A network with $K = 100$ hidden neurons reaches $>98\%$ prediction accuracy on this task when trained using online stochastic gradient descent (SGD), where a new sample $(\boldsymbol{x}, y)$ is drawn from the input distribution at each step of the algorithm. This limit allows us to focus on the impact of the data distribution; we discuss the case of finite training data in Fig. 2. After learning, the hidden neurons have split into two groups, with about half the neurons acting as detectors for inputs with long-range correlation. We plot the weight vector, or the RF, of four of these neurons in Fig. 1 *C*, Top. The RFs of these neurons are localized: they only have a few synaptic weights whose magnitude is significantly larger than zero in a small region of input space. On the other hand, neurons that detect short-range correlations develop very different representations: they converge to highly oscillatory patterns, i.e., sparse superpositions of higher-frequency Fourier modes.

Beyond the visual inspection of the RFs, we can quantify their localization by computing the inverse participation ratio (IPR) of their weight vector $\boldsymbol{w} = (w_i)$,

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

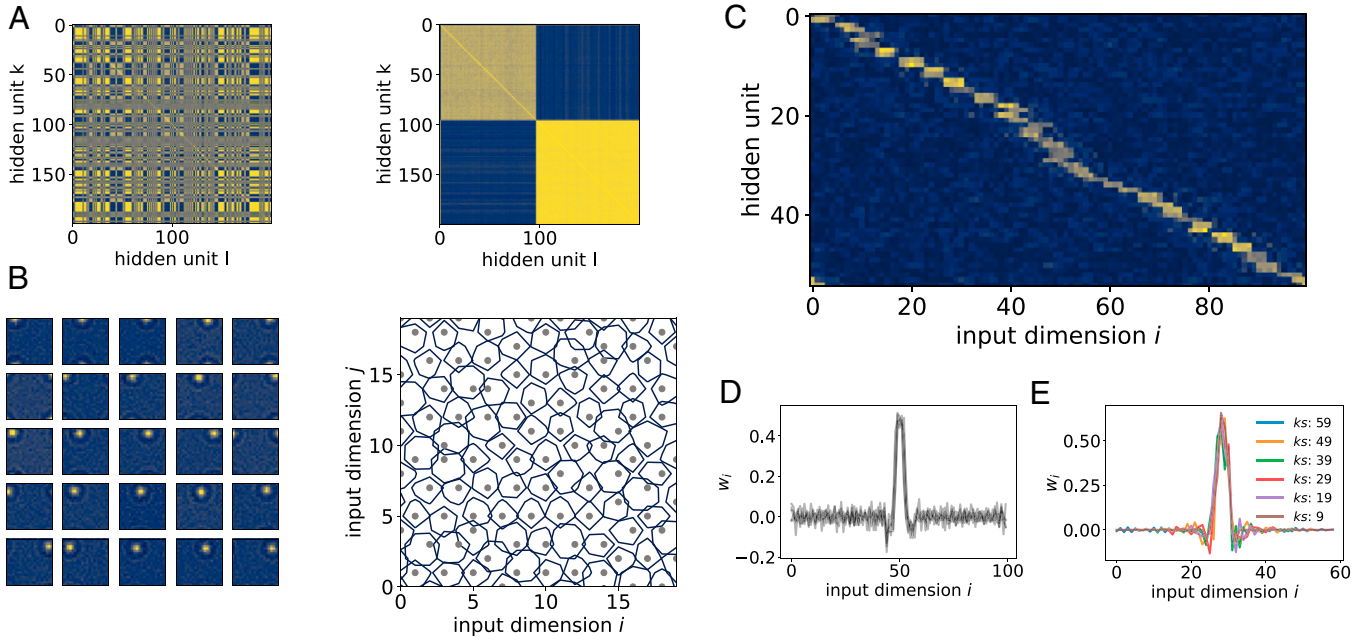

Fig. 2. RFs of FC networks tile input space and resemble the filters learned by a CNN. (*A*) (*Left*) Color plot of the translation-independent distance matrix (d_{kl}) (Materials and Methods) in a network with $K = 200$, trained on a two-dimensional binary classification task with $(\xi^-, \xi^+) = \left(\sqrt{2}, 2\right)$, $L = 20$. (*Right*) Permuted distance matrix using hierarchical clustering, showing how synaptic weight vectors cluster into two groups. (*B*) (*Left*) Weight intensity of localized RFs of a subset of neurons from the network in *A*. (*Right*) Centers (gray) and contour lines (blue) of the whole set of localized RFs plotted over the two-dimensional input space. (*C*) Localized RFs in a network with $K = 301$ trained on a one-dimensional task with $(\xi^-, \xi^+) = \left(\sqrt{10}, \sqrt{20}\right)$, $D = L = 100$. Weight intensity of each neuron is plotted along the rows, showing that RFs are arranged so as to tile the input space. Hidden units were sorted according to their center, in view of the permutation symmetry. (*D*) Overlay of five randomly selected RFs from *C*, after centering. (*E*) Filters of a two-layer convolutional network trained on the same task as *C* and *D*. Different colors correspond to different kernel sizes k_S, ranging from 9 to 59 pixels. Additional parameters are as follows: gain $g = 3$, batch learning with $P = \alpha D$ inputs, $\alpha = 10^5$, and SGD with batch size 1,000.

$$\mathrm{IPR}(\boldsymbol{w}) = \frac{\sum_{i=1}^{D} w_i^4}{\left(\sum_{i=1}^{D} w_i^2\right)^2}. \qquad [2]$$

The IPR quantifies the amount of nonzero components of a vector. It is commonly used to distinguish localized from extended eigenstates in quantum mechanics and random matrix theory (43) and is related to the kurtosis of the weights. We can successfully employ the IPR to measure the localization of RF in space throughout learning. We plot the IPR for the RFs of all neurons in Fig. 1*D* as a function of learning time, which is defined as the number of SGD steps divided by the total input size D. Localized neurons develop a large IPR over the course of training, while the IPR of neurons with oscillatory RFs remains very small.

Higher-Order Input Correlations Induce Localized RFs. To determine which of the characteristics of the dataset drive the emergence of localized RFs, we trained the same network on a Gaussian control task (Fig. 1*E*). For each class of inputs, we drew a new set of control images $\boldsymbol{c}$ from a Gaussian distribution with the same covariance as the inputs $\boldsymbol{x}$ from that class. We will sometimes refer to these inputs as the Gaussian process (GP) and denote the nonlinear inputs as NLGP. While both the inputs $\boldsymbol{x}$ and the Gaussian controls $\boldsymbol{c}$ from a given class have the same covariance by construction and are thus both translation-invariant, the original inputs $\boldsymbol{x}$ have increasingly sharp edges as we increase the gain factor g. These edges are a visual manifestation of the higher-order spatial correlations that cannot be captured by the simple Gaussian model. Indeed, the Gaussian samples appear blurry in comparison to the original data.

The same network with $K = 100$ neurons achieved a slightly inferior prediction accuracy on the Gaussian dataset. After learning, the neurons have again split evenly into two populations, detecting short- and long-range correlations. However, neurons learn very different representations from the data, with example RFs shown in Fig. 1*F*. There are no more localized fields; instead, neurons' weights converge to two-dimensional superpositions of low- and high-frequency Fourier components. This qualitative observation is borne out by the measurement of the IPR (Eq. **2**) of the RFs, which stays flat around zero throughout learning (cf. Fig. 1*G*).

Taken together, the results summarized in Fig. 1 show that localized RFs, the first hallmark of convolutions, emerge autonomously when training two-layer FC networks on a task with translation-invariant inputs that crucially possess non-Gaussian, higher-order local structure. This is to be contrasted with other recent studies that focused on the learnability of tasks that can be expressed as convolutions in a teacher–student setup (44–46).

RFs Tile Input Space and Resemble Filters of Convolutional Networks. The FC networks we trained also implement weight sharing, the second hallmark of convolutions, where the same filter is applied across the whole input. As shown in Fig. 2*A*, hidden units tend to cluster in two distinct groups. These clusters, which are identified by computing similarities between neurons using a translation-invariant measure (*Materials and Methods*), correspond to neurons with localized and oscillatory RFs. These RFs were obtained from a network that was trained using SGD on a finite dataset with $P = \alpha D$ samples, $\alpha = 10^5$.

We show a representative set of neurons with localized RF in Fig. 2 *B*, *Left*. The centers of these RF are spread over the input dimensions (Fig. 2 *B*, *Right*). The tiling is more striking in the one-dimensional case: we show in Fig. 2*C* all the localized RF's by plotting the weight vectors along the rows of the matrix. We see that as the number of hidden neurons K becomes comparable

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

to the input size D, the RFs tile the input space. A similar tiling has been observed in unsupervised learning with restricted Boltzmann machines by Harsh et al. (39).

We also compared the RF learned from scratch with the filters learned in a two-layer CNN with different filter sizes trained on the same task (see *Materials and Methods* for details). We found that the learned convolutional filters are stable across filter sizes (Fig. 2*E*). Strikingly, when a convolutional network is trained on the same task, the obtained filters strongly resemble the RFs learned by the FC network, as can be seen from a comparison of the filters in Fig. 2*E* with Fig. 2*D*, where we show RF of five randomly chosen neurons from Fig. 2*C*.

Current Theories of Learning Break Down during the Formation of RFs. How can we capture the formation of RFs theoretically? There exist precise theories for learning in neural networks with linear activation functions (47–52). However, the dynamics of even a deep linear network with several layers will only depend on the input–input and the input–label covariance matrices, i.e., the first two moments of the data (50). This formalism thus cannot capture the formation of RFs, which is driven by non-Gaussian fluctuations in the inputs. An exact theory describing the learning dynamics is available for nonlinear two-layer neural networks with large input size $D \to \infty$ and a few neurons $K \sim \mathcal{O}(1)$ in the hidden layer (53, 54). We verified that networks in this limit also form RFs (*SI Appendix*, Fig. S4). In this limit, one can derive a set of ordinary differential equations that predict the evolution of the prediction mean-squared test error (pmse) of a network (Eq. **10**) when training on Gaussian mixture classification (55). In Fig. 3, we show the pmse of a network with $K = 8$ neurons trained on the Gaussian control task (blue lines) and verify that this theory yields matching predictions (blue crosses; full details in *Materials and Methods*).

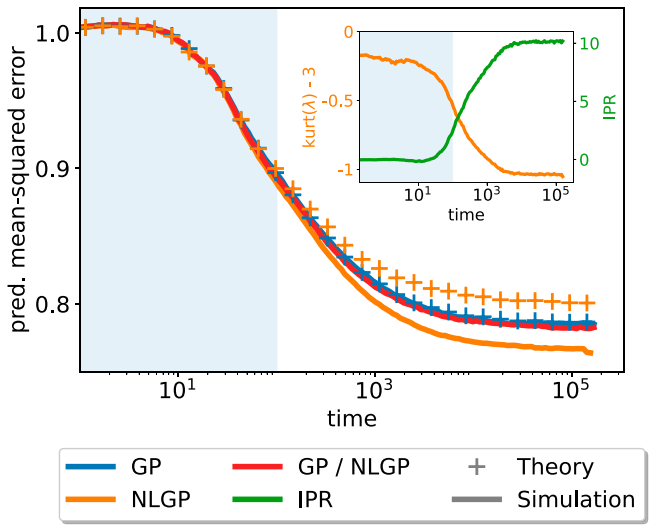

Fig. 3. Existing theories of learning in neural networks break down during the formation of RFs. pmse (10) of a network with $K = 8$ neurons trained on nonlinear Gaussian inputs (NLGP [Eq. **1**], orange) and on the Gaussian control task (GP, blue) with length scales $\xi^+ = 2\xi^- = 16$. The pmse is calculated using held-out test data during the simulation (solid lines). We also show the test error of the network trained on GP but evaluated on NLGP data (GP/NLGP, red). The crosses give the pmse obtained from evaluating an analytical expression describing the error of an equivalent Gaussian model (Materials and Methods). While the analytical expression accurately predicts the error in the beginning of training (blue shaded area), it breaks down for the network trained on NLGP around time 10^2. This is precisely the time at which the weights start to localize, as measured by the average IPR (2) of the localized weights (*Inset*, green). Simultaneously, the excess kurtosis of the preactivations of the network decreases (*Inset*, orange). Additional parameters are as follows: one-dimensional task with $D = L = 400$ and learning rate $\eta = 0.05$. Curves are averaged over 20 runs.

This type of analysis has recently been extended from mixtures of Gaussians to more complex input distributions thanks to the phenomenon of Gaussian equivalence, whereby the performance of a network trained on non-Gaussian inputs is still well captured by an appropriately chosen Gaussian model for the data. This Gaussian equivalence was used successfully to analyze random features (56–58) and neural networks with one or two layers, even when inputs were drawn from pretrained generative models (59–62). In Fig. 3, we plot the test error of a network trained on NLGP data together with the theoretical prediction obtained from applying the Gaussian equivalence theorem (GET) (61) (details are given in *Materials and Methods*). Initially, the theoretical predictions from the GET (orange crosses) agree with the test error measured in the simulation (orange line), but the theory breaks down around time $\approx 10^2$, when predictions start deviating from simulations.

The breakdown of the Gaussian theory coincides with the localization of the RFs, as measured by their IPR (Eq. **2**; green line in Fig. 3, *Inset*). The increased localization of the weights also coincides with a change in the statistics of the preactivations of the hidden neurons, $\lambda \sim \sum_i w_i x_i$: the excess kurtosis of λ (orange line) is initially close to zero, meaning that λ is approximately Gaussian, but decreases as the weights localize, indicating a transition to a non-Gaussian distribution.

We can finally see from Fig. 3 that the network is only influenced by the second-order fluctuations in both the NLGP and the GP at the beginning of training since the pmse values for models trained on NLGP and GP initially coincide. Likewise, a network trained on GP and evaluated on NLGP test data has the same test accuracy as the network trained directly on NLGP in the early stages of learning (red line). The higher-order moments of the NLGP inputs start influencing learning only at a later stage, when the IPR of the weight vectors increases and the Gaussian theory breaks down. This sequential learning of increasingly higher-order statistics of the inputs is reminiscent of how neural networks learn increasingly complex functions during training. Simplicity biases of this kind have been analyzed in simple models of neural networks (51, 53, 63–65) and have been demonstrated in modern convolutional networks (66). The sequential learning of increasingly higher-order statistics and the ensuing breakdown of the GET to describe learning is a result of independent interest which we will investigate further in future work.

The failure of the Gaussian theory to describe the emergence of RFs forces us to develop another theoretical approach. We make a first step in this direction by introducing a simplified model, which allows us to analyze the impact of the non-Gaussian statistics.

A Simplified Model Highlights the Importance of Non-Gaussian Statistics. The analysis of a reduced model with a single neuron reveals an interesting connection between the higher-order statistics of the data and the pattern formation mechanism driving the emergence of convolutional structure. We consider a single neuron with a polynomial activation function $\tilde{\sigma}$ of order 3 and study the weight updates of SGD in the limit of small learning rate. This leads us to consider the gradient flow (GF) dynamics of the neuron's weight vector $\boldsymbol{w}$ after averaging over the data distribution, which takes the form

$$\dot{\boldsymbol{w}} = \frac{1}{M}\sum_{\mu=1}^{M}\left[c_2^{\mu}\left(y^{\mu}, v, b\right) C^{\mu}\boldsymbol{w} + c_4\left(v, b\right) T^{\mu}\boldsymbol{w}^{\otimes 3}\right], \quad [3]$$

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

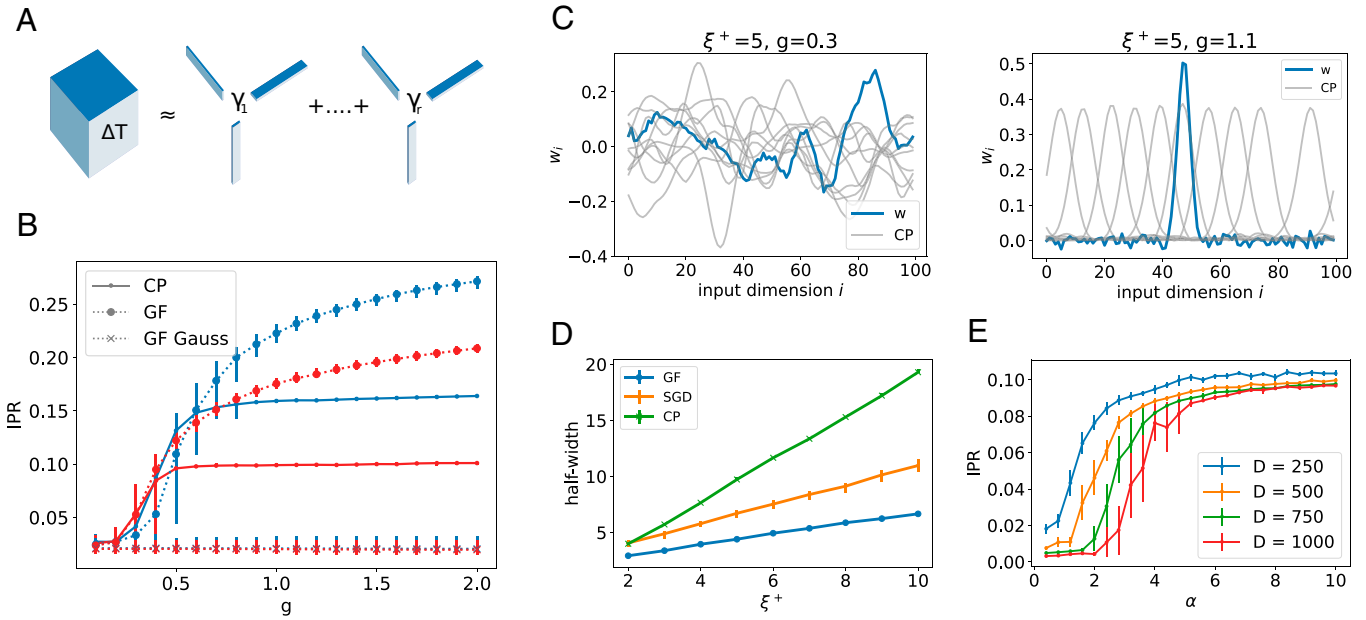

Fig. 4. Non-Gaussianity drives pattern formation in a simplified model of gradient descent dynamics. (*A*) Pictorial illustration of CP decomposition (67, 68), a tensor decomposition technique where a tensor (here a three-way tensor) is decomposed into a weighted sum of rank 1 tensors. (*B*) IPR as the gain factor g of the data is increased, thereby increasing the non-Gaussianity of the inputs (cf. Eq. **1**). The IPR is shown for 1) the leading CP factor of the fourth-order cumulant ΔT^2 (solid), 2) the weight vector of a single neuron obtained by integrating the GF equation (Eq. **5**; dots), and 3) the weight vector obtained by integrating the GF equation where higher-order moments have been replaced with Gaussian moments of the inputs (crosses). Error bars indicate interquartile range around median across 30 samples. (*C*) Synaptic weight vectors $\boldsymbol{w}$ (blue) obtained from integrating the GF equation for (*Left*) small and (*Right*) large values of the gain parameter. In gray, we show the 10 leading CP factors $\boldsymbol{u}_k$ of the fourth-order cumulant ΔT^2 for both datasets. (*D*) Half-width of the weight vector obtained from integrating the GF equation (Eq. **5**; blue) and of the leading CP factor of the fourth-order cumulant (green), as the correlation length ξ^+ of the inputs is increased. We also show the half-width of the weight vector obtained by training the simplified model directly using SGD updates with finite step size (orange). (*E*) Maximal IPR among the first 20 CP factors for a dataset containing $P = \alpha D$ inputs with $g = 3$, $\xi = 5$, for increasing α and size D. Error bars indicate ± 3 SE around the average across 30 samples. Additional parameters in *A*–*D* are as follows: one-dimensional inputs, $D = L = 100$, $K = 1$, $\xi^- = 0$, cumulants estimated from a dataset with $P = \alpha D$ inputs, $\alpha = 100$, learning rate $\eta = 0.01$, and bias fixed at $b = -1$.

where $C^\mu_{ij} = \langle x^\mu_i x^\mu_j \rangle$ and $T^\mu_{ijk\ell} = \langle x^\mu_i x^\mu_j x^\mu_k x^\mu_\ell \rangle$ are the second- and fourth-order joint moments of the inputs in the μ th class, respectively ($M = 2$). The notation $\boldsymbol{w}^{\otimes 3}$ indicates a threefold outer product of the vector $\boldsymbol{w}$ with itself (Eq. **15**). In Eq. **3**, we discarded the fifth-order term and introduced the coefficients c_2, c_4, as described in *Materials and Methods*. The same steps can be used to derive a similar GF equation for the bias of the neuron. Since the results do not depend on the exact value of the bias or on its dynamics, here we simplify the discussion by fixing the bias at $b = -1$. We verified that following the GF dynamics of the model in Eq. **3** yields a localized RF (cf. the blue lines in Fig. 4).

While a complete analysis of the synaptic dynamics for generic fourth-order tensors T^μ is very complicated and beyond our scope, we can gain insight by rewriting T^μ as a sum of the cumulant ΔT^μ plus the contribution from the second-order moment

$$T^\mu_{ijk\ell} = \Delta T^\mu_{ijk\ell} + C^\mu_{ij} C^\mu_{k\ell} + C^\mu_{ik} C^\mu_{j\ell} + C^\mu_{i\ell} C^\mu_{jk}. \qquad [4]$$

The cumulant ΔT^μ has the useful property that it is exactly zero for Gaussian inputs; in other words, it quantifies the non-Gaussian part of the fourth-order input statistics. We can then rewrite the synaptic dynamics as

$$\dot{\boldsymbol{w}} = \frac{1}{M} \sum_{\mu=1}^{M} \left(c_2^\mu + c_4 q^\mu \right) C^\mu \boldsymbol{w} + \frac{c_4}{M} \sum_{\mu=1}^{M} \Delta T^\mu \boldsymbol{w}^{\otimes 3}, \qquad [5]$$

where $q^\mu = \boldsymbol{w}^T C^\mu \boldsymbol{w}$ is the so-called self-overlap of the synaptic weight, and we dropped the dependence in c_2 and c_4 for brevity.

In our data model, the relative importance of the non-Gaussian statistics in the inputs is controlled by the gain factor g introduced in Eq. **1**: for small values of g, the error function is almost linear, and the inputs are almost Gaussian. For Gaussian inputs, the cumulant $\Delta T^\mu = 0$ and the synaptic dynamics $\dot{\boldsymbol{w}}$ are thus given only by the first term on the right-hand side. It can be shown that the fixed point equations imply a very sparse power spectrum (*Materials and Methods*); in other words, the weights converge to a superposition of only a few Fourier components, in line with our general finding for $K \geq 1$.

We train the single neuron on a task with nonlinear inputs (NLGP) at various values of the gain factor g. We set the correlation length of inputs in one class to zero, $\xi^- = 0$, and vary the correlation length for the second class. Integrating Eq. **5** yields the weight vector of the neuron at the end of training. We plot the IPR of this weight as a function of the gain factor g with the dotted line in Fig. 4*B* (blue and red for $\xi^+ = 3$, $\xi^+ = 5$, respectively), and show the weight for two values of the gain factor (Fig. 4*C*). The single neuron develops an RF that is increasingly localized as the gain factor, and hence the non-Gaussianity of the inputs, increases. We also integrated Eq. **5** keeping only the first term, so as to only retain the influence of the Gaussian part of the data on the learning dynamics. Integrating the reduced equation yields a synaptic weight that is not localized—its IPR is negligible for all values of the gain (crosses in Fig. 4*B*). The driving force behind the emergence of localized RFs in the reduced model is thus the fourth-order cumulant ΔT^μ.

We can gain insight into the structure of these higher-order correlations by means of tensor decomposition. Just like a matrix (which is a tensor of order 2) can be decomposed into a sum of outer products between eigenvectors, higher-order tensors can be expressed as a sum of a relatively small number of outer products of vectors, which are called factors in this context. There

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

exist several ways to decompose a tensor; here we focus on the CANDECOMP/PARAFAC (CP) decomposition of the fourth-order cumulant, whereby

$$\Delta T = \sum_{k=1}^{r} \gamma_k \boldsymbol{u}_k \otimes \boldsymbol{u}_k \otimes \boldsymbol{u}_k \otimes \boldsymbol{u}_k \qquad [6]$$

and r is the rank of the decomposition (see Fig. 4*A* for an illustration of a third-order tensor). Tensor decomposition has been successfully applied to supervised and unsupervised machine learning (68, 69), and its relevance in the context of unsupervised synaptic plasticity has also been recently recognized: Ocker and Buice (70) showed that a polynomial version of Hebbian learning can recover the dominant tensor eigenvectors of higher-order correlations, in a way that is similar to how the classic Oja rule recovers the leading eigenvector of the input covariance (71).

We find that the progressive localization of the RFs mirrors the localization of the dominant CP factors of the fourth-order cumulant (full curves in Fig. 4 *B* and *C*). For large enough datasets and high-rank r, the CP factors of the inputs tile the input space in a manner similar to the RFs obtained on the supervised task with a large number of hidden neurons K (Fig. 4 *C*, *Right*). The precise shape of the RFs is controlled by the CP factor obtained from a rank $r = 1$ decomposition of the fourth-order cumulant: as we show in Fig. 4*D*, the half-width of the localized RF obtained using both the reduced model (GF) and the full SGD dynamics closely follows the half-width of the CP factor when the correlation length ξ^+ is varied. Since the CP factor is computed over the correlated inputs, while the perceptron sees both correlated and uncorrelated inputs, the half-width of the RFs obtained from learning are smaller than that of the CP factor. Furthermore, the half-width of the RF obtained from SGD is closer to the value of the CP factor as the GF dynamics only has access to the first four moments of the inputs (cf. Eq. **5**), while the perceptron trained with SGD sees all the moments of its inputs.

It is also attractive to relate pattern formation in weight space with bump attractor dynamics in models with nonlinear local interactions (72–74). The dynamics of Eq. **5** in the presence of a low-rank CP decomposition of ΔT^{μ} is instructive, in that it manifests attractor-like phenomenology (*SI Appendix*, Fig. S2). This kind of dynamics in weight space is reminiscent of memory retrieval in continuous Hopfield models, where a third-order interaction among spin variables mediated by the fourth-order moment tensor is necessary for retrieval (75). When the previously introduced polynomial activation function $\tilde{\sigma}$ is used in conjunction with finite batch training and a plastic readout weight, both drifting periods and transitions between localized fields are apparent over the course of learning, as a result of effective noise induced by finite batch size (*SI Appendix*, Fig. S3). Although abrupt transitions are accompanied by transient sweeps of the readout weight v, both v and the bias b remain approximately constant while the position or sign of the localized field change, as predicted by the symmetry in the training data. Drifting localized fields have also been observed when a generative model (restricted Boltzmann machine) is trained using contrastive divergence (39) to reproduce configurations from a one-dimensional Ising chain.

Discussion

CNNs achieve better performance and need fewer samples than FC networks when trained with SGD, especially in vision, even though sufficiently wide FC networks can express convolutions. However, FC networks do not develop a convolutional structure autonomously when trained on a supervised image classification task. d'Ascoli et al. (41) recently highlighted the dynamical nature of this problem when they showed that convolutional solutions are not reachable by SGD starting from random FC initial conditions. The training has to be augmented by techniques like weight pruning (76) or complex regularization schemes (42) in order to learn weight matrices that display local connectivity and are organized in patterns reminiscent of convolutional networks.

Here we showed that convolutional structure in FC neural networks can emerge during training on a supervised learning task. We designed a minimal model of the visual scene whose non-Gaussian, higher-order statistics are the crucial ingredient for the emergence of RFs characterized by both localization and weight sharing. We further highlighted the dynamical nature of the learning phenomenon: the progression from second-order to higher-order statistics during learning is an example of how neural networks learn functions of increasing complexity.

We studied the pattern formation mechanism of localized RFs using a reduced model with a single neuron. A similar approach has recently been used in the context of unsupervised learning of configurations generated by lattice models in physics (39), where weight localization was interpreted in terms of a Turing instability mechanism. Our work follows the legacy of earlier pioneering studies on single neurons that analyzed storage and memory retrieval of spatially correlated (77–80) and invariant datasets (81). At variance with these classical works, here we focused on the dynamics of learning and studied the role of higher-order statistics. Encapsulating these higher-order information in appropriate order parameters presents itself as a crucial next step, in that it will allow studying the typical structure of the optimal solution to supervised learning problems with complex datasets.

The analysis of the single-neuron model led us to relate the emergence of structural properties of network connectivity to the tensor decomposition of higher-order input cumulants. For a neural network to be able to capture this structure, a large amount of data must be processed: one indeed expects the structure in the dominant CP factors (or tensor eigenvectors) of higher-order moments to depend on the number of samples. We demonstrate this point numerically in Fig. 4*E*, where we plot the localization of the dominant CP factors of the fourth-order cumulant (measured by their IPR) as a function of the number of samples for various input sizes D; details on how we performed tensor decomposition for large D are given in *SI Appendix*, section D. We note that as D increases (for constant correlation length ξ), the sample fluctuations increase at the transition. A better understanding of this transition and similar other properties of higher-order cumulants represents an interesting direction for further study. We expect that typical-case studies of the decomposition of large random tensors (82–86) and approaches based on random matrix theory (87) will prove fruitful in this direction, similar to the progress in understanding the spectral properties of random covariance matrices (88–90).

The extension of our single-neuron model to the multi-neuron case proves complicated due to the effective repulsive interactions between different weight vectors that appear even for Gaussian inputs. However, studying these effective repulsive interactions in the general case is an interesting future direction for our work as it could shed light on the mechanism of space-tiling.

While in this work we focused on translation symmetry, recent developments in deep learning have dealt with a variety of symmetry groups (23, 25, 29), and a general framework has been introduced for constructing equivariant layers capable of dealing with input invariances (31). It is thus natural to consider the impact of a generic symmetry group on the higher-order statistics of the data and ask for the conditions under which such a structure

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

is learnable and whether there exists a minimal amount of data necessary to detect such an invariance.

In the interest of tractability, here we employed a synthetic data model and a simple two-layer network. A full dynamic analysis of deeper architectures is complicated by the highly nonlinear dynamics of gradient descent, already evident even in linear networks (50–52). Understanding how invariances in data interact with depth in a neural network is an interesting direction for future investigation, both at the analytical and numerical level.

Studying the formation of RFs in recurrent networks is an interesting future direction for two reasons: recurrent networks provide an effective tool for capturing the spatiotemporal dynamics of the visual scene (4, 91), and they are promising models for the processing stages in the visual system (3, 92, 93). The simplest test bed for our approach would be the study of recurrent networks solving classification tasks (94) in the presence of data invariances.

Finally, we note that our work establishes an intriguing connection between supervised and unsupervised learning. We found that the gradient updates drive the weights in directions that increase the non-Gaussianity of the preactivations of the hidden neurons, as measured by their excess kurtosis (cf. Fig. 3). The representations found in this way perform better than the ones obtained from Gaussian inputs. Maximizing non-Gaussianity has long been recognized as a powerful mechanism to extract meaningful representations from images, e.g., kurtosis maximization in independent component analysis (95, 96). This work represents a step toward linking generative model approaches to vision with task-relevant feature extraction carried out by supervised learning rules.

Materials and Methods

Data Models. Our dataset consists of inputs $\boldsymbol{x}$ that can be one- or two-dimensional, divided in M distinct classes. Here we illustrate the different types of inputs in one dimension.

A data vector of the NLGP is given by $\boldsymbol{x}^\mu = Z^{-1}(g)\psi(g\boldsymbol{z}^\mu)$, where $\boldsymbol{z}^\mu$ is a zero-mean Gaussian vector of length L and covariance matrix $C_{ij}^\mu = \langle z_i^\mu z_j^\mu \rangle = e^{-(|i-j|/\xi^\mu)^2}$, with $i, j = 1, 2, \ldots, L$. The covariance thus only depends on the distance between sites i and j, given by $|i-j|$. The normalization factor $Z(g)$ is chosen such that $\mathrm{Var}(x) = 1$. Throughout this work, we took ψ to be a symmetric saturating function $\psi(z) = \mathrm{erf}(z/\sqrt{2})$, for which $Z(g)^2 = 2/\pi \arcsin\left(g^2/(1+g^2)\right)$. We also enforce periodic boundary conditions.

We create the Gaussian clone (GP) by drawing inputs from a Gaussian distribution with mean zero and the same covariance as the corresponding NLGP. The covariance of the NLGP can be evaluated analytically for $\psi(z) = \mathrm{erf}(z/\sqrt{2})$ and reads

$$\langle x_i^\mu x_j^\mu \rangle = \frac{2}{\pi Z(g)} \arcsin\left(\frac{g^2}{1+g^2} C_{ij}^\mu\right), \qquad [7]$$

where we have used that fact that $C_{ii} = 1$. The experiments on GPs are thus not performed on the Gaussian variables $\boldsymbol{z}$; they are performed on Gaussian random variables with covariance given in Eq. **7**. In this way, we exclude the possibility that the change in the two-point correlation function from applying the nonlinearity ψ is responsible for the emergence of RFs.

For one-dimensional inputs, the fact that the covariances of the NLGP and the GP depend only on the distances between pixels $|i-j|$ implies that they are circulant matrices (97). These matrices display a number of useful properties: they can be diagonalized using discrete Fourier transform, and thus, any two circulant matrices of the same size can be jointly diagonalized and commute with each other. We use this fact in the analysis of the reduced model to diagonalise the dynamics of the synaptic weights (*Gaussian inputs*).

We obtain the covariance for two-dimensional inputs by taking the Kronecker product of the one-dimensional covariance matrix with itself. For any dimension, we indicate the total input size by D.

Details on Neural Network Training. We trained a two-layer FC network with K hidden units and activation function σ. The output of the network to an input $\boldsymbol{x}$ is

$$\phi(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \sigma\left(\sum_{i=1}^{D} w_{ki} x_i + b_k\right), \qquad [8]$$

with $W \in \mathbb{R}^{K\times D}$ the matrix of first-layer weights and b_k the hidden unit biases. We initialized W with independent identically distributed (i.i.d.) zero-mean Gaussian entries with variance $1/D$. To obtain a minimal model of developing convolutions, we fixed the second-layer weights of the network to the value $v_k = 1/K$. We show that the emergence of RFs also occurs in networks where both layers are trained from scratch in *SI Appendix*, Fig. S1. We employed the sigmoidal activation function $\sigma(h) = \mathrm{erf}(h/\sqrt{2})$ for the results shown in Figures 1, 2, and 3 and verified that localized RFs also emerge with rectified linear unit (ReLU) activation $\sigma(x) = \max(0, x)$.

We trained the network using vanilla SGD, using both standard minibatch learning from a finite dataset and online learning. In the latter, a new sample $(\boldsymbol{x}, y)$ is drawn from the input distribution for each step of SGD. This limit is widely used in the theory of neural networks as it permits focusing on the statistical properties of the inputs, without effects that could arise due to scarce amounts of data. It has furthermore been shown that online learning is quite close to the practice of deep learning, where heavy data augmentation schemes lead to very large effective dataset sizes (98).

For binary discrimination tasks, we used $\{-1, +1\}$ output for the two classes. We focus our analysis on mean-square loss for simplicity of mathematical treatment. We verified that cross-entropy loss does not alter our main results. For the comparison with convolutional networks, we employed a two-layer network composed of a convolutional layer with circular padding, followed by an FC layer with linear output.

Invariant Overlap and Clustering. In order to compare different weight vectors $\boldsymbol{w}_k$ (rows of the first-layer weight matrix W), we introduce a similarity measure that is invariant to translation. Given two normalized weight vectors $\boldsymbol{w}_k$ and $\boldsymbol{w}_l$, the overlap $\tilde{q}_{kl}$ reads

$$\tilde{q}_{kl} = \frac{1}{D} |\tilde{\boldsymbol{w}}_k| \cdot |\tilde{\boldsymbol{w}}_\ell|, \qquad [9]$$

where $\tilde{w}_{k\tau}$ stands for the τ th Fourier components of the vector $\boldsymbol{w}_k$ and the absolute value is computed entrywise. The latter operation makes $\tilde{q}_{kl}$ invariant with respect to translation by removing the phase information. To help identify the set of localized RFs, we cluster the weight vectors $\boldsymbol{w}_k$ using a distance matrix $d_{kl} = 1 - \tilde{q}_{kl}$ and (average-linkage) hierarchical clustering. The same procedure is employed in both one and two dimensions.

Gaussian Equivalence. The pmse for a given network (as shown in Fig. 3) is defined as

$$\mathrm{pmse} \equiv \langle (\phi(\boldsymbol{x}) - y)^2 \rangle_{\boldsymbol{x},y}. \qquad [10]$$

The average is taken over the data distributions $(\boldsymbol{x}, y)$. We compute this error during the simulation by evaluating the performance of the model on held-out test data. A crucial observation is that the inputs $\boldsymbol{x}^\mu$ only affect the network output (Eq. **8**) via the dot product with the network's weights preactivations

$$\lambda_k^\mu = \sum_{i=1}^{D} w_{ki} x_i^\mu. \qquad [11]$$

The high-dimensional average over inputs in Eq. **10** can thus be replaced by a low-dimensional average over the preactivations. This approach to studying the learning dynamics of two-layer networks was pioneered by Saad and Solla (53) and Riegler and Biehl (99), who studied neural networks learning random functions of i.i.d. Gaussian inputs, and was recently made rigorous (100, 101). To obtain the theoretical predictions in Fig. 3, we built on a recent extension of this analysis to the case of mixtures of Gaussian inputs with nontrivial input correlations (55).

For Gaussian inputs (GP), the K preactivations λ_k^μ are jointly Gaussian for each class μ. For non-Gaussian inputs (NLGP), one can invoke the GET, which stipulates that for a wide class of input distributions, the preactivations λ_k^μ remain Gaussian (60, 61). In Fig. 3, we evaluate the test error of a network trained on NLGP by

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

replacing the actual preactivations λ_k^μ with Gaussian random variables $\tilde{\lambda}_k^\mu$ of mean zero and covariance $\langle \tilde{\lambda}_k^\mu \tilde{\lambda}_\ell^\mu \rangle = \boldsymbol{w}_k^T C^\mu \boldsymbol{w}_\ell$ for each input class μ. The GET prediction for the test error is then obtained by evaluating the average in Eq. **10** over the Gaussian variables $\tilde{\lambda}_k^\mu$. As we show in Fig. 3, the predictions based on Gaussian equivalence match the simulation initially but break down when the IPR of the localized weights increases.

The influence of the localization of the weight $\boldsymbol{w}_k$ on the higher-order statistics of the local fields λ_k^μ can be seen by computing the statistical excess kurtosis

$$\text{kurtosis}(\lambda_k^\mu) = \frac{\langle \lambda_k^4 \rangle_\mu}{\langle \lambda_k^2 \rangle_\mu{}^2} - 3, \qquad [12]$$

where the average is taken over the μ th input class. In Fig. 3, *Inset*, we plot the excess kurtosis averaged over the neurons with localized weights and over the two input classes.

Reduced Model and Tensor Decomposition. In order to analyze the learning dynamics in the presence of higher-order statistics, we expand the activation function to third order $\sigma(h) \approx \tilde{\sigma}(h) = \alpha_1 h - \frac{\alpha_3}{3} h^3$. In particular, for $\sigma(h) = \text{erf}\left(h/\sqrt{2}\right)$, one has $\alpha_1 = \sqrt{2/\pi}$ and $\alpha_3 = 1/\sqrt{2\pi}$. Upon presentation of a pattern from the μ th class, the gradient update reads $\Delta w_i^\mu = \eta v \left(y^\mu - v g\left(h^\mu\right)\right) g'\left(h^\mu\right) x_i^\mu$, with η a small learning rate. We thus get

$$\Delta w_i^\mu \propto c_2^\mu\left(y^\mu, v, b\right) \sum_{a=1}^{D} x_a^\mu w_a + c_4\left(v, b\right) \sum_{abc}^{D} x_a^\mu x_b^\mu x_c^\mu x_i^\mu w_a w_b w_c, \qquad [13]$$

where we discarded the fifth-order term in $\boldsymbol{w}$ and set

$$c_2^\mu\left(y^\mu, v, b\right) = -v\left[\left(\alpha_1 - \alpha_3 b^2\right)^2 + 2\alpha_3 b\left(y^\mu - \alpha_1 v b + \frac{\alpha_3}{3} v b^3\right)\right]$$

$$c_4\left(v, b\right) = v\left[\frac{4}{3}\alpha_3\left(\alpha_1 - \alpha_3 b^2\right) - 2\alpha_3^2 b^2\right].$$

Averaging over the inputs $\boldsymbol{x}^\mu$ and summing across the M classes, we get

$$\langle \Delta \boldsymbol{w} \rangle = \frac{\eta}{M} \sum_{\mu=1}^{M} \left(c_2^\mu C^\mu \boldsymbol{w} + c_4 T^\mu \boldsymbol{w}^{\otimes 3}\right), \qquad [14]$$

where we dropped the dependence in c_2 and c_4 for brevity. Recall that $C_{ij}^\mu = \langle x_i^\mu x_j^\mu \rangle$ and $T_{ijk\ell}^\mu = \langle x_i^\mu x_j^\mu x_k^\mu x_\ell^\mu \rangle$ are the second- and fourth-order joint moments of the inputs in the μ th class, respectively. Here and in Eqs. **3** and **5** we used the notation $T\boldsymbol{w}^{\otimes 3}$ to indicate the $3\times$ product of the tensor T with the vector $\boldsymbol{w}$:

$$\left(T\boldsymbol{w}^{\otimes 3}\right)_i = \sum_{abc}^{D} T_{abci} w_a w_b w_c. \qquad [15]$$

Gaussian inputs. Using Wick's theorem for centered data, $T_{abcd} = C_{ab}C_{cd} + C_{ac}C_{bd} + C_{ad}C_{bc}$, we can express the third-order term using the respective covariance matrices:

$$\sum_{abc}^{D} T_{abci}^\mu w_a w_b w_c = 3q^\mu \sum_{c=1}^{D} C_{ci}^\mu w_c, \qquad [16]$$

with $q^\mu = \boldsymbol{w}^T C^\mu \boldsymbol{w}$ the single-unit definition of the overlap. In the limit of small learning rate η, the full update up to third order in the weights thus reads

$$\dot{\boldsymbol{w}} = \frac{1}{M} \sum_{\mu=1}^{M} \left(c_2^\mu + c_4 q^\mu\right) C^\mu \boldsymbol{w}. \qquad [17]$$

We can Fourier transform Eq. **17**, exploiting the fact that all the C^μ are jointly diagonalizable. Eq. **17** then implies that at the steady state,

$$\tilde{w}_\tau = 0 \qquad \text{or} \qquad \sum_{\mu=1}^{M} \lambda_\tau^\mu \left(c_2^\mu + c_4 q^\mu\right) = 0, \qquad [18]$$

where $\tilde{w}_\tau$ are the components of $\boldsymbol{w}$ in the Fourier basis. We thus have a set of D equations with $M + 2$ unknown. It follows that $\tilde{w}_\tau = 0$ for most τ values.

Generic inputs and CP decomposition. We decompose the fourth-order moment of the μ th class as $T^\mu = T_g^\mu + \Delta T^\mu$, with T_g^μ and ΔT^μ the Gaussian component and the fourth-order cumulant, respectively. The full update thus reads

$$\dot{\boldsymbol{w}} = \frac{1}{M} \sum_{\mu=1}^{M} \left(c_2^\mu + c_4 q^\mu\right) C^\mu \boldsymbol{w} + \frac{c_4}{M} \sum_{\mu=1}^{M} \Delta T^\mu \boldsymbol{w}^{\otimes 3}. \qquad [19]$$

We employ CP decomposition (68) in order to find a low-rank approximation of the cumulant tensor ΔT, i.e., a set of r real coefficients γ_a and vectors $\boldsymbol{u}_a$ such that

$$\Delta T_{ijk\ell} \approx \sum_{a=1}^{r} \gamma_a u_{ia} u_{ja} u_{ka} u_{\ell a}. \qquad [20]$$

Note that the symmetry of the cumulant tensor implies that the vectors $\boldsymbol{u}_a$ are the same across the four modes. For moderate input size D, we use the Tensorly package in Python (102). For large D (typically for $D > 100$), construction and storage of large tensors of higher order become prohibitive. We thus built upon the framework recently introduced in ref. 103, which uses an implicit representation of high-order moment tensors coupled to a gradient-based optimization. We generalized the method in ref. 103 to deal with the low-rank approximation of cumulant tensors. A detailed description is given in *SI Appendix*.

Data, Materials, and Software Availability. Code for data generation and network training can be found at GitHub (https://github.com/sgoldt/conv_emerge) (104).

ACKNOWLEDGMENTS. A.I. thanks L. F. Abbott for his support and fruitful discussions while at the Zuckerman Institute, Columbia University, where research included in this work was partly performed. We thank J. Barbier, M. Marsili, and M. Refinetti for fruitful discussions.

1. J. J. DiCarlo, D. Zoccolan, N. C. Rust, How does the brain solve visual object recognition? *Neuron* **73**, 415–434 (2012).
2. D. L. K. Yamins *et al.*, Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 8619–8624 (2014).
3. K. Kar, J. J. DiCarlo, Fast recurrent processing via ventrolateral prefrontal cortex is needed by the primate ventral stream for robust core visual object recognition. *Neuron* **109**, 164–176.e5 (2021).
4. C. J. Spoerer, P. McClure, N. Kriegeskorte, Recurrent convolutional neural networks: A better model of biological object recognition. *Front. Psychol.* **8**, 1551 (2017).
5. 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).
6. E. I. Knudsen, M. Konishi, Center-surround organization of auditory receptive fields in the owl. *Science* **202**, 778–780 (1978).
7. J. P. Jones, A. Stepnoski, L. A. Palmer, The two-dimensional spectral structure of simple receptive fields in cat striate cortex. *J. Neurophysiol.* **58**, 1212–1232 (1987).
8. J. J. DiCarlo, K. O. Johnson, S. S. Hsiao, Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey. *J. Neurosci.* **18**, 2626–2645 (1998).
9. L. Paninski, J. Pillow, J. Lewi, Statistical models for neural encoding, decoding, and optimal stimulus design. *Prog. Brain Res.* **165**, 493–507 (2007).
10. M. Vidne *et al.*, Modeling the impact of common noise inputs on the network activity of retinal ganglion cells. *J. Comput. Neurosci.* **33**, 97–121 (2012).
11. M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex. *Nat. Neurosci.* **2**, 1019–1025 (1999).
12. F. E. Theunissen, J. E. Elie, Neural processing of natural sounds. *Nat. Rev. Neurosci.* **15**, 355–366 (2014).
13. B. A. Olshausen, D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
14. Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network" in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed. (Morgan-Kaufmann, 1990), pp. 396–404.
15. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
16. D. Scherer, A. Müller, S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition" in *International Conference on Artificial Neural Networks*, K. Diamantaras, W. Duch, L. S. Iliadis, Eds. (Springer, 2010), pp. 92–101.
17. J. Schmidhuber, Deep learning in neural networks: An overview. *Neural Netw.* **61**, 85–117 (2015).
18. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016).
19. G. Urban *et al.*, "Do deep convolutional nets really need to be deep and convolutional?" in *International Conference on Learning Representations*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2017).
20. A. Krizhevsky, I. Sutskever, G. Hinton, "Imagenet classification with deep convolutional neural networks" in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2012), pp. 1097–1105.

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

21. K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition" in *International Conference on Learning Representations*, Y. Bengio, Y. LeCun, Eds. (2015). https://dblp.org/db/conf/iclr/iclr2015.html. Accessed 1 January 2021.
22. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
23. T. Cohen, M. Welling, "Group equivariant convolutional networks" in *Proceedings of The 33rd International Conference on Machine Learning*, eds. M. F. Balcan, K. Q. Weinberger (Proceedings of Machine Learning Research, New York, NY, 2016), vol. 48, pp. 2990–2999.
24. T. Cohen, M. Weiler, B. Kicanaoglu, M. Welling, "Gauge equivariant convolutional networks and the icosahedral CNN" in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri, R. Salakhutdinov, Eds. (Proceedings of Machine Learning Research, 2019), vol. 97, pp. 1321–1330.
25. R. Kondor, S. Trivedi, "On the generalization of equivariance and convolution in neural networks to the action of compact groups" in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy, A. Krause, Eds. (Proceedings of Machine Learning Research, 2018), Vol. 80, pp. 2747–2755.
26. D. J. Rezende, S. Racanière, I. Higgins, P. Toth, Equivariant Hamiltonian flows. arXiv [Preprint] (2019). https://arxiv.org/abs/1909.13739. Accessed 1 April 2021.
27. T. S. Cohen, M. Geiger, J. Köhler, M. Welling, "Spherical CNNs" in *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=Hkbd5xZRb. Accessed 1 January 2021.
28. F. Monti *et al.*, "Geometric deep learning on graphs and manifolds using mixture model cnns" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, 2017), pp. 5425–5434.
29. M. Weiler, M. Geiger, M. Welling, W. Boomsma, "TS Cohen, 3d steerable cnns: Learning rotationally equivariant features in volumetric data" in *Advances in Neural Information Processing Systems*, S. Bengio *et al.*, Eds., (Curran Associates, Inc., 2018), vol. 31.
30. M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.* **34**, 18–42 (2017).
31. M. M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv [Preprint] (2021). https://arxiv.org/abs/2104.13478. Accessed 1 October 2021.
32. S. Mallat, Group invariant scattering. *Commun. Pure Appl. Math.* **65**, 1331–1398 (2012).
33. J. Bruna, S. Mallat, Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1872–1886 (2013).
34. S. Mallat, Understanding deep convolutional networks. *Philos. Trans.- Royal Soc., Math. Phys. Eng. Sci.* **374**, 20150203 (2016).
35. E. Doi *et al.*, Efficient coding of spatial information in the primate retina. *J. Neurosci.* **32**, 16256–16264 (2012).
36. Y. Karklin, E. Simoncelli, "Efficient coding of natural images with a population of noisy linear-nonlinear neurons" in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2011), vol. 24, pp. 999–1007.
37. M. K. Benna, S. Fusi, Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence. *Proc. Natl. Acad. Sci.* **118**, e2018422118 (2021).
38. M. Farrell, S. Recanatesi, R. C. Reid, S. Mihalas, E. Shea-Brown, Autoencoder networks extract latent variables and encode these variables in their connectomes. *Neural Netw.* **141**, 330–343 (2021).
39. M. Harsh, J. Tubiana, S. Cocco, R. Monasson, 'Place-cell' emergence and learning of invariant data with restricted Boltzmann machines: Breaking and dynamical restoration of continuous symmetries in the weight space. *J. Phys. A Math. Theor.* **53**, 174002 (2020).
40. A. Sengupta, C. Pehlevan, M. Tepper, A. Genkin, D. Chklovskii, "Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks" in *Advances in Neural Information Processing Systems*, S. Bengio *et al.*, Eds., (Curran Associates, Inc., 2018), vol. 31, pp. 7080–7090.
41. S. d'Ascoli, L. Sagun, G. Biroli, J. Bruna, "Finding the needle in the haystack with convolutions: On the benefits of architectural bias" in *Advances in Neural Information Processing Systems*, H. Wallach *et al.*, Eds., (Curran Associates, Inc., 2019), vol. 32. https://dl.acm.org/doi/10.5555/3454287.3455124. Accessed 1 January 2021.
42. B. Neyshabur, "Towards learning convolutions from scratch" in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin, Eds. (Curran Associates, Inc., 2019), vol. 33, pp. 8078–8088.
43. F. L. Metz, I. Neri, D. Bollé, Localization transition in symmetric random matrices. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **82**, 031135 (2010).
44. S. Mei, T. Misiakiewicz, A. Montanari, Learning with invariances in random features and kernel models. arXiv [Preprint] (2021). https://arxiv.org/abs/2102.13219. Accessed 1 October 2021.
45. T. Misiakiewicz, S. Mei, Learning with convolution and pooling operations in kernel methods. arXiv [Preprint] (2021). https://arxiv.org/abs/2111.08308. Accessed 1 January 2022.
46. A. Favero, F. Cagnetta, M. Wyart, "Locality defeats the curse of dimensionality in convolutional teacher-student scenarios" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, J. Wortman Vaughan, Eds. (Curran Associates, Inc., 2021), vol. 34, pp. 9456–9467.
47. P. Baldi, K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.* **2**, 53–58 (1989).
48. Y. L. Cun, I. Kanter, S. A. Solla, Eigenvalues of covariance matrices: Application to neural-network learning. *Phys. Rev. Lett.* **66**, 2396–2399 (1991).
49. A. Krogh, J. A. Hertz, Generalization in a linear perceptron in the presence of noise. *J. Phys. Math. Gen.* **25**, 1135 (1992).
50. A. Saxe, J. McClelland, S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks" in *International Conference on Learning Representations (ICLR)*, Yoshua Bengio, Yann LeCun, Eds. (2014). https://dblp.org/rec/journals/corr/SaxeMG13.html?view=bibtex or https://iclr.cc/archive/2014/conference-proceedings/. Accessed 1 January 2021.
51. A. M. Saxe, J. L. McClelland, S. Ganguli, A mathematical theory of semantic development in deep neural networks. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 11537–11546 (2019).
52. M. S. Advani, A. M. Saxe, H. Sompolinsky, High-dimensional dynamics of generalization error in neural networks. *Neural Netw.* **132**, 428–446 (2020).
53. D. Saad, S. A. Solla, Exact solution for on-line learning in multilayer neural networks. *Phys. Rev. Lett.* **74**, 4337–4340 (1995).
54. M. Biehl, H. Schwarze, Learning by on-line gradient descent. *J. Phys. Math. Gen.* **28**, 643–656 (1995).
55. M. Refinetti, S. Goldt, F. Krzakala, L. Zdeborova, "Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed" in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila, T. Zhang, Eds. (Proceedings of Machine Learning Research, 2021), vol. 139, pp. 8936–8947.
56. Z. Liao, R. Couillet, "On the spectrum of random features maps of high dimensional data" in *International Conference on Machine Learning*, J. Dy, A. Krause, Eds. (Proceedings of Machine Learning Research, 2018), vol. 80, pp. 3063–3071.
57. M. Seddik, M. Tamaazousti, R. Couillet, "Kernel random matrices of large concentrated data: The example of gan-generated images" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, S. Sanei, L. Hanzo, Eds. (IEEE, 2019), pp. 7480–7484.
58. S. Mei, A. Montanari, The generalization error of random features regression: Precise asymptotics and the double descent curve. *Commun. Pure Appl. Math* **75**, 667–766 (2022).
59. S. Goldt, M. Mézard, F. Krzakala, L. Zdeborová, Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Phys. Rev. X* **10**, 041044 (2020).
60. H. Hu, Y. M. Lu, Universality laws for high-dimensional learning with random features. arXiv [Preprint] (2020). https://arxiv.org/abs/2009.07669. Accessed 31 March 2021.
61. S. Goldt *et al.*, The Gaussian equivalence of generative models for learning with two-layer neural networks in mathematical and scientific machine learning, Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference, in *Proceedings of Machine Learning Research* (2021), vol. 145, pp. 426–471.
62. B. Loureiro *et al.*, "Learning curves of generic features maps for realistic datasets with a teacher-student model" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, J. Wortman Vaughan, Eds. (Curran Associates, Inc., 2021), vol. 34, pp. 18137–18151.
63. H. Schwarze, J. Hertz, Generalization in a large committee machine. *EPL* **20**, 375 (1992).
64. A. Engel, C. Van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, 2001).
65. N. Rahaman *et al.*, "On the spectral bias of neural networks" in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, K. Chaudhuri, R. Salakhutdinov, Eds. (PMLR, 2019), pp. 5301–5310.
66. D. Kalimeris *et al.*, "SGD on neural networks learns functions of increasing complexity" in *Advances in Neural Information Processing Systems*, H. Wallach *et al.*, Eds., (Curran Associates, Inc., 2019), vol. 32.
67. H. A. L. Kiers, A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity. *J. Chemometr.* **12**, 155–171 (1998).
68. T. G. Kolda, B. W. Bader, Tensor decompositions and applications. *SIAM Rev.* **51**, 455–500 (2009).
69. A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, M. Telgarsky, Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.* **15**, 2773–2832 (2014).
70. G. K. Ocker, M. A. Buice, "Tensor decomposition of higher-order correlations by nonlinear Hebbian plasticity" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, J. Wortman Vaughan (2021), vol. 34, pp. 11326–11339.
71. E. Oja, A simplified neuron model as a principal component analyzer. *J. Math. Biol.* **15**, 267–273 (1982).
72. S. Amari, Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.* **27**, 77–87 (1977).
73. A. D. Redish, A. N. Elga, D. S. Touretzky, A coupled attractor model of the rodent head direction system. *Network* **7**, 671–685 (1996).
74. A. Compte, N. Brunel, P. S. Goldman-Rakic, X. J. Wang, Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cereb. Cortex* **10**, 910–923 (2000).
75. D. Bollé, T. M. Nieuwenhuizen, I. P. Castillo, T. Verbeiren, A spherical Hopfield model. *J. Phys. Math. Gen.* **36**, 10269–10277 (2003).
76. F. Pellegrini, G. Biroli, "Neural Network Pruning Denoises the Features and Makes Local Connectivity Emerge in Visual Tasks" in *Proceedings of the 39th International Conference on Machine Learning* (PMLR, 2022), vol. 162, pp. 17601–17626.
77. R. Monasson, Storage of spatially correlated patterns in autoassociative memories. *J. Phys. I* **3**, 1141–1152 (1993).
78. R. Monasson, Properties of neural networks storing spatially correlated patterns. *J. Phys. Math. Gen.* **25**, 3701–3720 (1992).
79. W. Tarkowski, M. Lewenstein, Storage of sets of correlated data in neural network memories. *J. Phys. Math. Gen.* **26**, 2453–2469 (1993).
80. W. Tarkowski, M. Lewenstein, Learning from correlated examples in a perceptron. *J. Phys. Math. Gen.* **26**, 3669–3679 (1993).
81. W. Tarkowski, M. Komarnicki, M. Lewenstein, Optimal storage of invariant sets of patterns in neural network memories. *J. Phys. Math. Gen.* **24**, 4197–4217 (1991).
82. E. Richard, A. Montanari, "A statistical model for tensor PCA" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2014), vol. 27.
83. A. Montanari, D. Reichman, O. Zeitouni, "On the limitation of spectral methods: From the Gaussian hidden clique problem to rank-one perturbations of gaussian tensors" in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett, Eds. (Curran Associates, Inc., 2015), vol. 28.
84. T. Lesieur, L. Miolane, M. Lelarge, F. Krzakala, L. Zdeborová, "Statistical and computational phase transitions in spiked tensor estimation" in *2017 IEEE International Symposium on Information Theory (ISIT)*, Gerhard Kramer, Rudolf Mathar, Eds. (IEEE, 2017), pp. 511–515.
85. W. K. Chen, Phase transition in the spiked random tensor with rademacher prior. *Ann. Stat.* **47**, 2734–2756 (2019).
86. A. Perry, A. S. Wein, A. S. Bandeira, Statistical limits of spiked tensor models. *Ann. Inst. Henri Poincare Probab. Stat.* **56**, 230–264 (2020)..
87. J. H. de Morais Goulart, R. Couillet, P. Comon, A random matrix perspective on random tensors. arXiv [Preprint] (2021). https://arxiv.org/abs/2108.00774. Accessed 30 March 2021.
88. J. Baik, G. B. Arous, S. Péché, Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Ann. Probab.* **33**, 1643–1697 (2005).
89. A. Auffinger, G. Ben Arous, J. Cerny, Random matrices and complexity of spin glasses. *Commun. Pure Appl. Math.* **66**, 165–201 (2013).
90. M. Potters, J. P. Bouchaud, *A First Course in Random Matrix Theory: For Physicists, Engineers and Data Scientists* (Cambridge University Press, 2020).

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.

91. M. Liang, X. Hu, "Recurrent convolutional neural network for object recognition" in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Curran Associates, 2015), pp. 3367–3375.
92. C. J. Spoerer, T. C. Kietzmann, J. Mehrer, I. Charest, N. Kriegeskorte, Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLOS Comput. Biol.* **16**, e1008215 (2020).
93. R. S. van Bergen, N. Kriegeskorte, Going in circles is the way forward: The role of recurrence in visual inference. *Curr. Opin. Neurobiol.* **65**, 176–193 (2020).
94. M. Farrell, S. Recanatesi, T. Moore, G. Lajoie, E. Shea-Brown, Recurrent neural networks learn robust representations by dynamically balancing compression and expansion. bioRxiv [Preprint] (2019). https://www.biorxiv.org/content/10.1101/564476v2. Accessed 8 August 2021.
95. A. Bell, T. J. Sejnowski, "Edges are the 'independent components'of natural scenes" in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, T. Petsche, Eds., (MIT Press, 1996), vol. 9.
96. A. Hyvärinen, E. Oja, Independent component analysis: Algorithms and applications. *Neural Netw.* **13**, 411–430 (2000).
97. R. A. Horn, C. R. Johnson, *Matrix Analysis* (Cambridge University Press, 2012).
98. P. Nakkiran, B. Neyshabur, H. Sedghi, "The bootstrap framework: Generalization through the lens of online optimization" in *International Conference on Learning Representations* (2021). https://openreview.net/forum?id=guetrIHLFGI. Accessed 1 December 2021.
99. P. Riegler, M. Biehl, On-line backpropagation in two-layered neural networks. *J. Phys. Math. Gen.* **28**, L507–L513 (1995).
100. S. Goldt, M. Advani, A. M. Saxe, F. Krzakala, L. Zdeborová, "Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup" in *Advances in Neural Information Processing Systems*, H. Wallach *et al.*, Eds., (Curran Associates, Inc., 2019), vol. 32.
101. R. Veiga, L. Stephan, B. Loureiro, F. Krzakala, L. Zdeborová, Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks. arXiv [Preprint] (2022). https://arxiv.org/abs/2202.00293. Accessed 2 May 2022.
102. J. Kossaifi, Y. Panagakis, A. Anandkumar, M. Pantic, Tensorly: Tensor learning in Python. *J. Mach. Learn. Res.* **20**, 1–6 (2019).
103. S. Sherman, T. G. Kolda, Estimating higher-order moments using symmetric tensor decomposition. *SIAM J. Matrix Anal. Appl.* **41**, 1369–1387 (2020).
104. A. Ingrosso, S. Goldt, Data from "Data-driven emergence of convolutional structure in neural networks." GitHub. https://github.com/sgoldt/conv_emerge. Deposited 8 September 2022.

Downloaded from https://www.pnas.org by SISSA-BIBLIOTECA INTL SCHOOL FOR ADVANCED STUDY on December 11, 2023 from IP address 147.122.98.68.