Mechanisms of dimensionality reduction and decorrelation in deep neural networks

Haiping Huang[*]

School of Physics, Sun Yat-sen University, Guangzhou 510275, People's Republic of China
and Laboratory for Neural Computation and Adaptation, RIKEN Center for Brain Science, Wako-shi, Saitama 351-0198, Japan

(Received 30 October 2017; revised manuscript received 26 November 2018; published 14 December 2018)

Deep neural networks are widely used in various domains. However, the nature of computations at each layer of the deep networks is far from being well understood. Increasing the interpretability of deep neural networks is thus important. Here, we construct a mean-field framework to understand how compact representations are developed across layers, not only in deterministic deep networks with random weights but also in generative deep networks where an unsupervised learning is carried out. Our theory shows that the deep computation implements a dimensionality reduction while maintaining a finite level of weak correlations between neurons for possible feature extraction. Mechanisms of dimensionality reduction and decorrelation are unified in the same framework. This work may pave the way for understanding how a sensory hierarchy works.

DOI: 10.1103/PhysRevE.98.062313

I. INTRODUCTION

The sensory cortex in the brain encodes the structure of the environment in an efficient way. This is achieved by creating progressively better representations of sensory inputs, and these representations finally become easily decoded without any reward or supervision signals [1–3]. This kind of learning is called unsupervised learning, which has long been thought of as a fundamental function of the sensory cortex [4]. Based on the similar computational principle, many layers of artificial neural networks were designed to perform a nonlinear dimensionality reduction of high-dimensional data [5], which later triggered resurgence of deep neural networks. By stacking unsupervised modules on top of each other, one can produce a deep feature hierarchy, in which high-level features can be constructed from less abstract ones along the hierarchy. However, these empirical results do not have a principled understanding so far. Understanding what each layer exactly computes may shed light on how sensory systems work in general.

Recent theoretical efforts focused on the layerwise propagation of one input vector length, correlations between two inputs [6], and clustered noisy inputs of supervised classification tasks [7,8], generalizing a theoretical work of layered feedforward neural networks that studied the iteration of the overlap between the layer's activity and embedded random patterns [9]. However, these studies did not address the covariance of neural activity, one important feature of neural data modeling [10], which is directly related to the dimensionality and complexity of hierarchical representations. Therefore, a clear understanding of hierarchical representations has been lacking so far, which makes deep computation extremely nontransparent. Here, we propose a mean-field theory of input dimensionality reduction in deep neural networks. In this theory, we capture how a deep nonlinear transformation reduces the dimensionality of a data representation and, moreover, how the covariance level (redundancy) varies along the hierarchy. Both of these two features are fundamental properties of deep neural networks, and even information processing in vision [11].

Our theory helps to advance the understanding of deep computation in the following two aspects. (i) There exists an operating point where input and output covariance levels are equal. This point controls the level of covariance neither diverging nor decaying to zero, given sufficiently strong connections between layers. (ii) The dimensionality of data representation is reduced across layers, due to an additive positive term (contributed by the previous layer) affecting the dimensionality in a divisive way. These computational principles are revealed not only in deterministic deep networks with random weights but also in generative deep trained networks. Our analytical findings coincide with numerical simulations, demonstrating that the previous empirically observed dimensionality reduction [1,2,5] and the redundancy reduction hypothesis [12] could be theoretically explained within the same mean-field framework.

II. A DETERMINISTIC DEEP NETWORK

A deep network is a multilayered neural network performing hierarchical nonlinear transformations of sensory inputs (Fig. 1). The number of hidden layers is defined as the depth of the network, and the number of neurons at each layer is called the width of that layer. For simplicity, we assume an equal width (N). Weights between $l-1$ and lth layers are specified by a matrix $\mathbf{w}^l$, in which the ith row corresponds to incoming connections to the neuron i at the higher layer. Biases of neurons at the lth layer are denoted by $\mathbf{b}^l$. The input data vector is denoted by $\mathbf{v}$, and $\mathbf{h}^l$ ($l=1,\ldots,d$) denotes a hidden representation of the lth layer, in which each entry h_i^l defines a nonlinear transformation of its preactivation $\tilde{a}_i^l \equiv [\mathbf{w}^l\mathbf{h}^{l-1}]_i$, as $h_i^l = \phi(\tilde{a}_i^l + b_i^l)$. Without loss of generality, we choose the nonlinear transfer function as $\phi(x) = \tanh(x)$ and assume that

[*] physhuang@gmail.com; www.labxing.com/hphuang2018

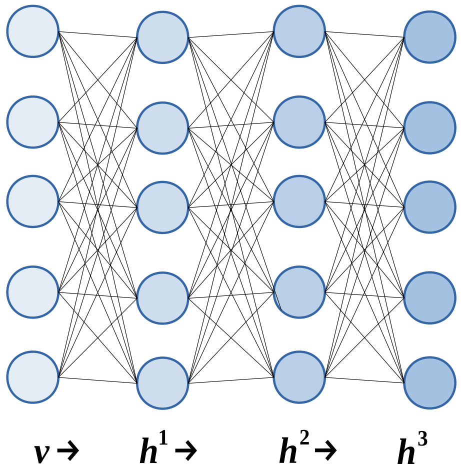

FIG. 1. Schematic illustration of a deep neural network. The deep neural network performs a layer-by-layer nonlinear transformation of the original input data (a high-dimensional vector $\mathbf{v}$). During the transformation, a cascade of internal representations $(\mathbf{h}^1, \ldots, \mathbf{h}^d)$ are created. Here, $d = 3$ denotes the depth of the deep network.

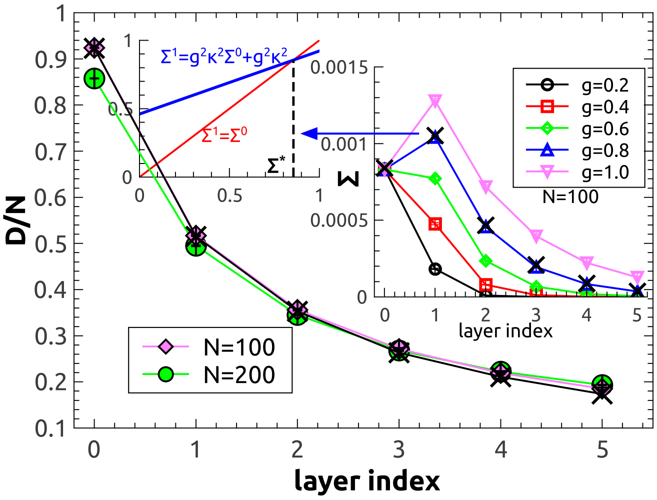

FIG. 2. Representation dimensionality versus depth in deterministic deep networks with random weights. Ten network realizations are considered for each network width. $\rho/\sqrt{N} = 0.05$, $g = 0.8$, and $\sigma_b = 0.1$. The right inset shows how the overall strength of covariance changes with depth and connection strength (g), and the left inset is a mechanism illustration ($\Sigma^{0,1,*}$ has been scaled by N). The crosses show simulation results ($g = 0.8$) obtained from 10^5 sampled configurations at each layer, compared with the theoretical predictions.

the weight follows a normal distribution $\mathcal{N}(0, g/N)$, and the bias follows $\mathcal{N}(0, \sigma_b)$. Random weight assumption plays an important role in recent studies of artificial neural networks [13–18], and the weight distribution of trained networks may appear random [19].

We consider a Gaussian input ensemble with zero mean, covariance $\langle v_i v_j \rangle = \frac{r_{ij}}{\sqrt{N}}$ for all $i \neq j$ (r_{ij} is a uniformly distributed random variable from $[-\rho, \rho]$), and variance $\langle v_i^2 \rangle = 1$. In the following derivations, we define the weighted-sum $\tilde{a}_i^l$ subtracted by its mean as $a_i^l = \sum_j w_{ij}^l (h_j^{l-1} - \langle h_j^{l-1} \rangle)$, thus a_i^l has zero mean. As a result, the covariance of $\mathbf{a}^l$ can be expressed as $\Delta_{ij}^l = \langle a_i^l a_j^l \rangle = [\mathbf{w}^l \mathbf{C}^{l-1} (\mathbf{w}^l)^T]_{ij}$, where $\mathbf{C}^{l-1}$ defines the covariance matrix of neural activity at the $l-1$-th layer (also called the connected correlation matrix in physics). Because the deep network defined in Fig. 1 is a fully connected feedforward network, where each neuron at an intermediate layer receives a large number of inputs, the central limit theorem implies that the means of the hidden neural activity $\mathbf{m}^l$ and the covariance $\mathbf{C}^l$ are given separately by

$$m_i^l = \langle h_i^l \rangle = \int Dt\phi\left(\sqrt{\Delta_{ii}^l}t + [\mathbf{w}^l \mathbf{m}^{l-1}]_i + b_i^l\right), \tag{1a}$$

$$C_{ij}^l = \int Dx\, Dy \phi\left(\sqrt{\Delta_{ii}^l}x + b_i^l + [\mathbf{w}^l \mathbf{m}^{l-1}]_i\right) \phi\left(\sqrt{\Delta_{jj}^l}\left(\Psi x + y\sqrt{1-\Psi^2}\right) + b_j^l + [\mathbf{w}^l \mathbf{m}^{l-1}]_j\right) - m_i^l m_j^l, \tag{1b}$$

where $Dx = e^{-x^2/2} dx/\sqrt{2\pi}$ and $\Psi = \frac{\Delta_{ij}^l}{\sqrt{\Delta_{ii}^l \Delta_{jj}^l}}$. To derive Eq. (1), we parametrize a_i^l and a_j^l by independent normal random variables (see Appendix A). Equation (1) forms an iterative mean-field equation across layers to describe the transformation of the activity statistics in deep networks.

To characterize the collective property of the entire hidden representation, we define an intrinsic dimensionality of the representation as $D = \frac{(\sum_{i=1}^N \lambda_i)^2}{\sum_{i=1}^N \lambda_i^2}$ [20], where $\{\lambda_i\}$ is the eigenspectrum of the covariance matrix $\mathbf{C}^l$. It is expected that $D = N$ if each component of the representation is generated independently with the same variance. Generally speaking, nontrivial correlations in the representation will result in $D < N$. Therefore, we can use the above mean-field equation together with the dimensionality to address how the complexity of hierarchical representations changes along the depth.

Based on this mean-field framework, we first study the aforementioned deterministic deep neural networks. Regardless of whichever network width is used, we find that the representation dimensionality progressively decreases across layers (Fig. 2). The theoretical results agree very well with numerical simulations (indicated by crosses in Fig. 2). This shows that, even in a random multilayered neural network, a more compact representation of the correlated input is gradually computed as the network becomes deeper, which is also one of the basic properties in biological hierarchical computations [2,21].

To get deeper insights about the hidden representation, we study how the overall strength of covariance at each layer changes with the network depth and the connection strength (g). The overall covariance-strength is measured by $\Sigma = \frac{2}{N(N-1)} \sum_{i<j} C_{ij}^2$, which is related to the dimensionality via $\Sigma = \frac{1}{N-1}[(\frac{1}{N}\sum_i C_{ii})^2/\tilde{D} - \frac{1}{N}\sum_i C_{ii}^2]$, where $\tilde{D} = D/N$, which is derived by noting that $\mathrm{tr}(\mathbf{C}) = \sum_i \lambda_i$ and $\mathrm{tr}(\mathbf{C}^2) = \sum_i \lambda_i^2$. We find that, to support an effective representation where neurons are not completely independent, the connection strength must be sufficiently strong (Fig. 2), such that weakly correlated neural activities are still maintained at later stages of processing. Otherwise, the information will

be blocked from passing through that layer where the neural activity becomes completely independent. High correlations imply strong statistical dependence and thus redundancy. An efficient representation must not be highly redundant [12], because a highly redundant representation cannot be easily disentangled and is thus not useful for computation; e.g., coadaptation of neural activities is harmful for feature extraction [22].

The dimensionality reduction results from the nested nonlinear transformation of input data. For a mechanistic explanation, by noting that Δ_{ij}^l is of the order $O(1/\sqrt{N})$, we expand $C_{ij}^l = K_{ij}\Delta_{ij}^l + O[(\Delta_{ij}^l)^2]$ in a large-N limit (Appendix B), where $K_{ij} \equiv \phi'(x_i^0)\phi'(x_j^0)$ and $x_{i,j}^0 \equiv b_{i,j}^l + [\mathbf{w}^l\mathbf{m}^{l-1}]_{i,j}$. Then $\Sigma^l \simeq g^2\kappa^2\Sigma^{l-1} + \frac{g^2\kappa^2}{N^2}\sum_i (C_{ii}^{l-1})^2$ where $\kappa \equiv \overline{[\phi'(x_i^0)]^2}$ (the average is taken over the random network parameters, see Appendix B). For the random model, $N\Sigma^1 = g^2\kappa^2(N\Sigma^0 + 1)$, which determines a critical $N\Sigma^* = \frac{g^2\kappa^2}{1-g^2\kappa^2}$ (so-called operating point), such that a first boost of the correlation strength is observed when $\Sigma^0 < \Sigma^*$ (Fig. 2); otherwise, the correlation level is maintained, or decorrelation is achieved. The iteration of Σ^l can be used to derive $\tilde{D}^1 = \frac{1}{(N-1)\Sigma^0+1+\Upsilon}$, where $\Upsilon > 0$ and its value depends on the layer's parameters. Thus Υ determines how significantly the dimensionality is reduced, and the dimensionality reduction is explained as $\tilde{D}^1 < \tilde{D}^0 = \frac{1}{(N-1)\Sigma^0+1}$. This relationship carries over to deeper layers (Appendix B), due to an additive positive term in the denominator of the dimension formula. Therefore, the decorrelation of redundant inputs together with the dimensionality reduction is theoretically explained.

III. A STOCHASTIC DEEP NETWORK

It is of practical interest to see whether a deep generative model trained in an unsupervised way has a similar collective behavior. We consider a deep belief network (DBN) as a typical example of stochastic deep networks [5], in which each neuron's activity at one hidden layer takes a binary value (±1) according to a stochastic function of the neuron's preactivation. Specifically, the DBN is composed of multiple restricted Boltzmann machines (RBMs) stacked on top of each other (Fig. 1). RBM is a two-layered neural network, where there are no lateral connections within each layer, and the bottom (top) layer is also named the visible (hidden) layer. Therefore, given the input $\mathbf{h}^l$ at the lth layer, the neural representation at a higher ($l+1$-th) layer is determined by a conditional probability:

$$P(\mathbf{h}^{l+1}|\mathbf{h}^l) = \prod_i \frac{e^{h_i^{l+1}\left([\mathbf{w}^{l+1}\mathbf{h}^l]_i + b_i^{l+1}\right)}}{2\cosh\left([\mathbf{w}^{l+1}\mathbf{h}^l]_i + b_i^{l+1}\right)}. \tag{2}$$

Similarly, $P(\mathbf{h}^l|\mathbf{h}^{l+1})$ is also factorized.

The DBN as a generative model, once network parameters (weights and biases) are learned (so-called training) from a data distribution, can be used to reproduce the samples mimicking that data distribution. With deep layers, the network becomes more expressive to capture high-order interdependence among components of a high-dimensional input, compared with a shallow RBM network. To study the expressive property of the DBN, we first specify a data distribution generated by a random RBM whose parameters follow the normal distribution $\mathcal{N}(0, g/N)$ for weights and $\mathcal{N}(0, \sigma_b)$ for biases. Using the random RBM as a data generator allows us to calculate analytically the complexity of the input data. In the random RBM, the hidden neural activity $\mathbf{h}$ at the top layer can be marginalized over using the conditional independence [Eq. (2)]; thus the distribution of the representation $\mathbf{v}$ at the bottom layer can be expressed as (Appendix C)

$$P(\mathbf{v}) = \frac{1}{Z}\prod_a \{2\cosh([\mathbf{w}^{l+1}\mathbf{v}]_a + b_a)\}\prod_i e^{v_i b_i}, \tag{3}$$

where a is the site index of hidden neurons, and Z is the partition function. Based on the Bethe approximation [23], which captures weak correlations among neurons, the covariance of neural activity (the same definition as before) under Eq. (3) can be computed from the approximate free energy using the linear response theory (Appendix D). The estimated statistics of the random-RBM representation are used as a starting point from which the mean-field complexity-propagation equation [Eq. (1)] iterates, for the investigation of the dimensionality and the redundancy reduction in the deep generative model.

Finally, we study the generative deep network where network parameters are learned in a bottom-up pass from the representations at lower layers. The network parameters for each stacked RBM in the DBN are updated by the contrast divergence procedure truncated to one step [24]. With this layerwise training, each layer learns a nonlinear transformation of the data, and upper layers are conjectured to learn more abstract (complex) concepts, which is a key step in object recognition problems [25]. One typical learning trajectory for each layer is shown in the top inset of Fig. 3(a), where the reconstruction error decreases with the learning epoch. The input data and subsequent representation complexity are captured very well by the theory [Fig. 3(b)]. We use the mean-field framework derived for deterministic networks to study the complexity propagation (starting from the statistics of the input data), which is reasonable, because to suppress the noise due to sampling, the mean activities at the intermediate layer are used as the input data when the next layer is learned [24]. Therefore, the stochasticity of the neural response is implicitly encoded into the learned parameters during training.

Compared to the initial input dimensionality, the representation dimensionality the successive layers create becomes lower [Fig. 3(a)], which coincides with observations in the deterministic random deep networks. This feature does not change when more neurons are used in each layer. During learning, the evolution of the dimensionality displays a nonmonotonic behavior [Fig. 3(c)]: the dimensionality first increases and then decreases to a stationary value. Moreover, the learning decorrelates the correlated input, whereas, after the first drop, the learning seems to preserve a finite level of correlations [the bottom inset of Fig. 3(a)]. These compact representations may remove some irrelevant factors in the input, which facilitates the formation of easily decoded representations at deeper layers. Our theoretical analysis in random neural networks will likely carry over to this unsupervised

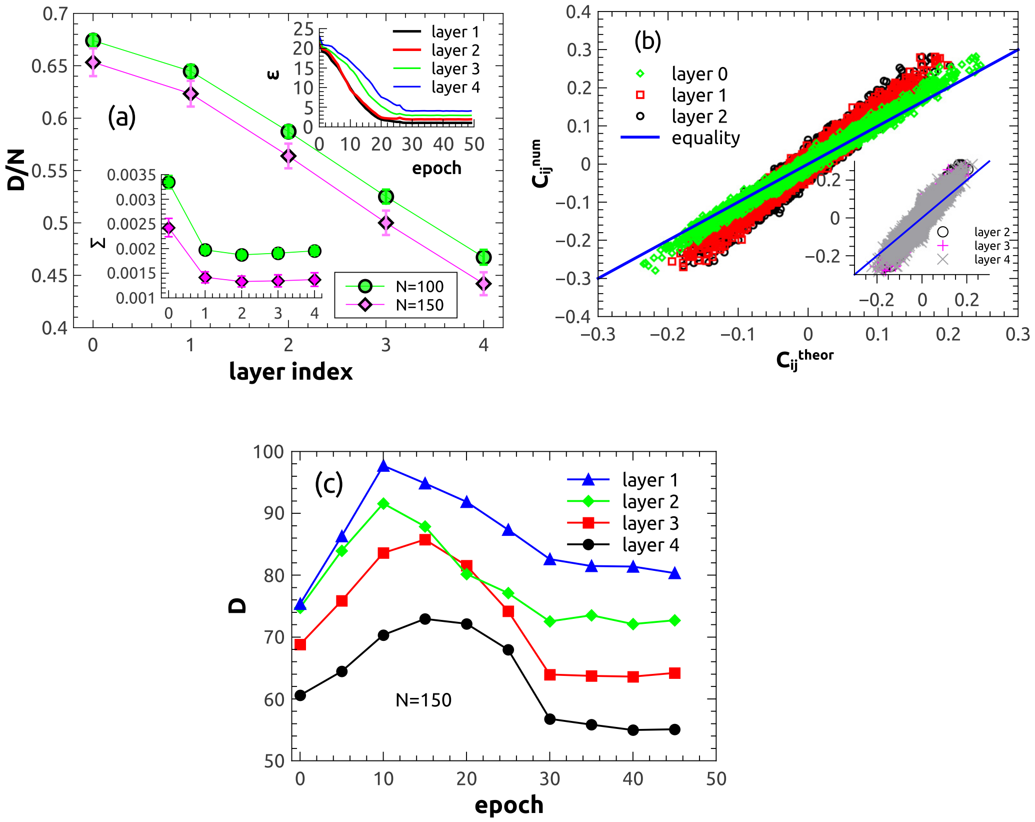

FIG. 3. (a) Representation behavior as a function of depth in generative deep networks. Ten network realizations are considered for each network width. The top inset shows an example ($N = 150$) of reconstruction errors ($\varepsilon \equiv \|\mathbf{h}' - \mathbf{h}\|_2^2$) between input $\mathbf{h}$ and reconstructed one $\mathbf{h}'$ for each layer during learning. The bottom inset shows the overall strength of covariance as a function of depth. (b) Numerically estimated off-diagonal correlation versus its theoretical prediction ($N = 150$). In these plots, we generate $M = 60\,000$ training examples (each example is an N-dimensional vector) from the random RBM whose parameters follow the normal distribution $\mathcal{N}(0, g/N)$ for weights and $\mathcal{N}(0, \sigma_b)$ for biases. $g = 0.8$ and $\sigma_b = 0.1$. Then these examples are learned by the DBN (see simulation details in Appendix C). (c) One typical learning trial shows how the estimated dimensionality evolves.

learning system; e.g., the operating point may explain the low-level preserved correlation.

By looking at the eigenvalue distribution of the covariance matrix, we find that the distribution for the unsupervised learning system deviates significantly from the Marchenko-Pastur law of a Wishart ensemble [26] (Appendix E). For the random neural networks, the eigenvalue distribution at deep layers seems to assign a higher probability density when the eigenvalue gets close to zero, yet a lower density at the tail of the distribution, compared with the Marchenko-Pastur law of a random-sample covariance matrix [27] (Appendix E). Therefore, the dimensionality reduction and its relationship with decorrelation are a nontrivial result of the deep computation.

IV. SUMMARY

Brain computation can be thought of as a transformation of internal representations along different stages of a hierarchy [3,21]. Deep artificial neural networks can also be interpreted as a way of creating progressively better representations of input sensory data. Our work provides mean-field evidence about this picture that compact representations of relatively low dimensionality are progressively created by deep computation, while a small level of correlations is still maintained to make feature extraction possible, in accord with the redundancy reduction hypothesis [12]. In the deep computation, more abstract concepts captured at higher layers along the hierarchy are typically built upon less abstract ones at lower layers, and high-level representations are generally invariant to local changes of the input [2], which thereby coincides with our theory that demonstrates a compact (compressed) representation formed by a series of dimensionality reduction. Unwanted variability may be suppressed in this compressed representation. It was hypothesized that neuronal manifolds at lower layers are strongly entangled with each other, while at later stages, manifolds are flattened to facilitate that relevant information can be easily decoded by downstream areas [1,2,21], which connects to the small level of correlations preserved in the network for a representation that may be maximally disentangled [28].

Our work thus provides a theoretical underpinning of the hierarchical representations, through a physics explanation of dimensionality reduction and decorrelation, which encourages several directions such as generalization of this theory to more complex architectures and data distributions, demonstration of how the compact representation helps generalization (invariance) or discrimination (selectivity) in a neural system [29,30], and using the revealed principles to control the complexity of internal representations for an engineering application.

ACKNOWLEDGMENTS

I thank Hai-Jun Zhou and Chang-Song Zhou for their insightful comments. This research was supported by AMED under Grant No. JP18dm020700 and the start-up budget Grant No. 74130-18831109 of the 100 Talents Program of Sun Yat-sen University.

APPENDIX A: DERIVATION OF MEAN-FIELD EQUATIONS FOR THE COMPLEXITY PROPAGATION IN DEEP RANDOM NEURAL NETWORKS

We first derive the mean-field equation for the mean activity m_i^l by noting that its pre-activation $\tilde{a}_i^l + b_i^l = a_i^l + [\mathbf{w}^l\mathbf{m}^{l-1}]_i + b_i^l$. a_i^l behaves like a Gaussian random variable with zero mean and variance Δ_{ii}^l, depending on the fluctuating input; thus the average operation for the mean activity can be computed as a Gaussian integral:

$$m_i^l = \langle h_i^l \rangle = \int Dt\phi\big(\sqrt{\Delta_{ii}^l}t + [\mathbf{w}^l\mathbf{m}^{l-1}]_i + b_i^l\big). \tag{A1}$$

Analogously, to compute the covariance C_{ij}^l, one first evaluates the statistics of the pre-activations of units i and j, i.e., a_i^l and a_j^l, which follows a joint Gaussian distribution with zero mean, variances Δ_{ii}^l and Δ_{jj}^l, respectively, and covariance Δ_{ij}^l. Therefore, one can use two independent standard Gaussian random variables with zero mean and unit variance to parametrize this joint distribution, which results in

$$\begin{aligned} C_{ij}^l = &\int DxDy\phi\big(\sqrt{\Delta_{ii}}x + b_i^l + [\mathbf{w}^l\mathbf{m}^{l-1}]_i\big) \\ &\times \phi\big[\sqrt{\Delta_{jj}}\big(\Psi x + y\sqrt{1-\Psi^2}\big) + b_j^l + [\mathbf{w}^l\mathbf{m}^{l-1}]_j\big] \\ &- m_i^l m_j^l, \end{aligned} \tag{A2}$$

where $Dx = e^{-x^2/2}dx/\sqrt{2\pi}$ and $\Psi = \frac{\Delta_{ij}}{\sqrt{\Delta_{ii}\Delta_{jj}}}$. The superscript l is omitted for the covariance of $\mathbf{a}^l$. It is easy to verify that the above parametrization of a_i^l and a_j^l follows the same statistics as mentioned above.

APPENDIX B: THEORETICAL ANALYSIS IN THE LARGE-N LIMIT

In the thermodynamic limit, the covariance of activation $\mathbf{a}^l$, Δ_{ij}^l, is of the order of $O(1/\sqrt{N})$, where N is the network width. This is because both the weights and the (connected) correlations are also of the order of $O(1/\sqrt{N})$. Thus, one can expand the covariance equation [Eq.(A2)] in the small Δ_{ij} limit. After the expansion and some simple algebra, one obtains

$$C_{ij}^l = K_{ij}\Delta_{ij} + O\big(\Delta_{ij}^2\big), \tag{B1}$$

where $K_{ij} = \phi'(x_i^0)\phi'(x_j^0)$ and $x_{i,j}^0 = b_{i,j}^l + [\mathbf{w}^l\mathbf{m}^{l-1}]_{i,j}$. It follows that

$$\begin{aligned} \Sigma^l &\simeq \frac{2}{N(N-1)}\sum_{i<j} K_{ij}^2\Delta_{ij}^2 \\ &\simeq g^2\overline{K_{ij}^2}\Sigma^{l-1} + \frac{g^2\overline{K_{ij}^2}}{N^2}\sum_i \big(C_{ii}^{l-1}\big)^2, \end{aligned} \tag{B2}$$

where $\overline{K_{ij}^2} \simeq \overline{[\phi'(x_i^0)]^2}^2$ in which the mean (overline) is taken over the quenched disorder, based on the facts that the correlation between different weights is negligible and that the covariance of the mean preactivations of different units is negligible as well in the thermodynamic limit.

Finally, $\overline{[\phi'(x_i^0)]^2} = \int Dt \int Du[\phi'(\sqrt{\sigma_b}u + \sqrt{gQ^{l-1}}t)]^2$, where one independent Gaussian random variable (u) corresponds to the randomness of the bias, and the other Gaussian random variable (t) corresponds to the random mean preactivation $[\mathbf{w}^l\mathbf{m}^{l-1}]_i$ because of the random weights. In addition, gQ specifies the variance of the mean preactivation, with the definition $Q = \frac{1}{N}\sum_i m_i^2$ (or spin glass order parameter in physics). Clearly, Q can be iteratively computed from one layer to its next layer as follows:

$$Q^l = \int Dt \int Du\phi^2\big(\sqrt{\sigma_b}u + \sqrt{gQ^{l-1}}t\big). \tag{B3}$$

Note that the initial $Q^0 = 0$ by the construction of the random model.

Looking at $l = 1$ for the random model, one finds $\overline{[\phi'(x_i^0)]^2} = \int Du[\phi'(\sqrt{\sigma_b}u)]^2 \equiv \kappa$ as defined in the main text. It follows that $\Sigma^1 = g^2\kappa^2\Sigma^0 + g^2\kappa^2/N$. Following the definition of the normalized dimensionality ($\tilde{D} = \frac{[\mathrm{Tr}(\mathbf{C})]^2}{N\mathrm{Tr}(\mathbf{C}^2)}$, an alternative definition of the dimensionality in the main text), one easily arrives at the relationship between $\tilde{D}^1 = \frac{1}{(N-1)\Sigma^0+1+\Upsilon}$ and $\tilde{D}^0 = \frac{1}{(N-1)\Sigma^0+1}$, noting that $\Upsilon \equiv \frac{\overline{[\phi'(x_i^0)]^4}}{\overline{[\phi'(x_i^0)]^2}^2} \geqslant 1$.

To derive the relationship between dimensionality of consecutive layers, we first define $K_1^l = \frac{1}{N}\sum_i C_{ii}^l$ and $K_2^l = \frac{1}{N}\sum_i (C_{ii}^l)^2$, and then we get the normalized dimensionality of layer l as

$$\tilde{D}^l = \frac{\big(K_1^l\big)^2}{(N-1)\Sigma^l + K_2^l}, \tag{B4}$$

which is compared with the counterpart at a higher layer $l+1$ given by

$$\tilde{D}^{l+1} = \frac{\big(K_1^l\big)^2}{(N-1)\Sigma^l + K_2^l + \big(K_1^l\big)^2\Upsilon}. \tag{B5}$$

To derive Eq. (B5), we used Eq. (B2). Because the additive term $(K_1^l)^2\Upsilon$ in the denominator is always positive, Eq. (B5) explains the dimensionality reduction across layers. The value of the additive term thus determines how significantly the dimensionality is reduced. Its behavior with increasing numbers of layers can also be analyzed within the large-N expansion. First, we derive the recursion equation for K_1^l. Using the fact

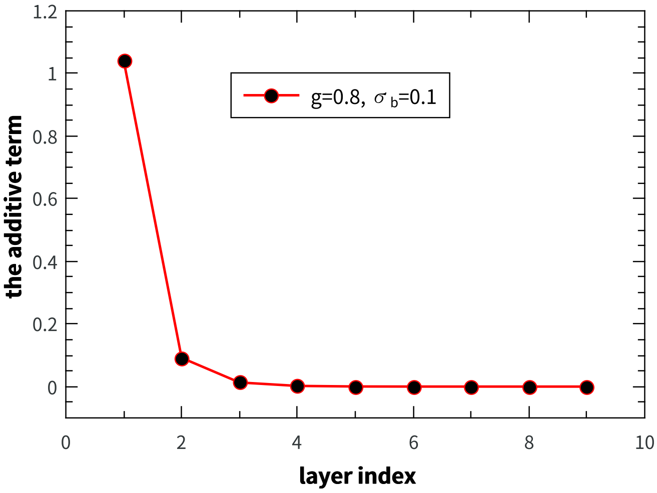

FIG. 4. The behavior of the additive term $(K_1^l)^2\Upsilon$ as a function of the network depth.

that $\Delta_{ii} \simeq gK_1^{l-1}$, one derives that

$$K_1^l = \int Du \int Dt\phi^2\big(\sqrt{g\big(K_1^{l-1} + Q^{l-1}\big)}t + \sqrt{\sigma_b}u\big) - Q^l, \tag{B6}$$

by following the same principle as mentioned above. Second, according to the definition, it is easy to write that

$$\Upsilon = \frac{\int Dt\int Du\big[\phi'\big(\sqrt{\sigma_b}u + \sqrt{gQ^{l-1}}t\big)\big]^4}{\big\{\int Dt\int Du\big[\phi'\big(\sqrt{\sigma_b}u + \sqrt{gQ^{l-1}}t\big)\big]^2\big\}^2}. \tag{B7}$$

Last, we find that the additive positive term tends to be a very small value as the number of layers increases (Fig. 4), which is consistent with the observation in a finite-N system. This implies that the estimated dimensionality at deep layers becomes nearly a constant, due to the nearly vanishing additive term.

APPENDIX C: TRAINING PROCEDURE OF DEEP BELIEF NETWORKS

A deep belief network (DBN) is composed of multiple restricted Boltzmann machines (RBMs) stacked on top of each other. It is a probabilistic deep generative model, because after network parameters (weights and biases) are learned (so-called training) from a data distribution, the model can be used to reproduce the samples mimicking the data distribution. With deep layers, the network becomes more expressive to capture high-order interdependence among components of a high-dimensional input, compared with a shallow RBM network.

Learning in a deep belief network can be achieved by layerwise training of each RBM in a bottom-up pass, which was justified to improve a variational lower bound on the data log-likelihood [24]. RBM is a two-layered neural network, where there are no lateral connections within each layer, and the bottom (top) layer is also named the visible (hidden) layer. Therefore, the RBM is described by the following energy function (also named Hamiltonian in physics):

$$E(\mathbf{s}, \boldsymbol{\sigma}) = -\sum_{i,a} s_a w_{ia}\sigma_i - \sum_a b_a^h s_a - \sum_i b_i^v \sigma_i, \tag{C1}$$

where $\mathbf{s}$ and $\boldsymbol{\sigma}$ are the hidden and visible activity vectors, respectively. $\mathbf{b}^{h,v}$ is the hidden (h) or visible (v) bias vector. In statistical mechanics, the neural activity follows a Boltzmann distribution, $P(\mathbf{s}, \boldsymbol{\sigma}) = \exp[-E(\mathbf{s}, \boldsymbol{\sigma})]/Z$, where Z is the partition function of the model. For a large network, Z can only be computed by approximated methods [16]. This distribution can be used to fit any arbitrary discrete distribution, following the maximal likelihood learning principle; i.e., the network parameters are updated according to gradient ascent of the data log-likelihood defined as follows:

$$\mathcal{L} = \sum_a \left\langle \ln\left[2\cosh\left(\sum_i w_{ia}\sigma_i + b_a^h\right)\right]\right\rangle_{\text{data}} + \sum_i b_i^v \langle\sigma_i\rangle_{\text{data}} - \ln Z, \tag{C2}$$

where the average is performed over all training data samples (or a mini-batch of the entire data set in the case where stochastic gradient ascent is used). The gradient ascent leads to the following learning equations for updating network parameters:

$$\Delta w_{ia} = \eta[\langle\sigma_i s_a p(s_a|\boldsymbol{\sigma})\rangle_{\text{data}} - \langle s_a\sigma_i\rangle_{\text{model}}], \tag{C3a}$$

$$\Delta b_i^v = \eta[\langle\sigma_i\rangle_{\text{data}} - \langle\sigma_i\rangle_{\text{model}}], \tag{C3b}$$

$$\Delta b_a^h = \eta[\langle s_a p(s_a|\boldsymbol{\sigma})\rangle_{\text{data}} - \langle s_a\rangle_{\text{model}}], \tag{C3c}$$

where η specifies a learning rate. Since there are no lateral connections between neurons at each layer, given one layer's activity, the other layer's activity is factorized as

$$p(\mathbf{s}|\boldsymbol{\sigma}) = \prod_a p(s_a|\boldsymbol{\sigma}) = \prod_a \frac{e^{s_a\left([\mathbf{w}\boldsymbol{\sigma}]_a + b_a^h\right)}}{2\cosh\left([\mathbf{w}\boldsymbol{\sigma}]_a + b_a^h\right)}. \tag{C4}$$

Similarly, $p(\boldsymbol{\sigma}|\mathbf{s})$ is also factorized.

There are many approximate methods to evaluate the model-dependent terms in the learning equations. Here, we use the most popular method, namely contrast divergence [24]. More precisely, RBMs are trained in a feedforward fashion using the contrast divergence algorithm [24], where Gibbs samplings of the model starting from each data point are truncated to a few steps and then used to compute model-dependent statistics for learning. The upper layer is trained with the lower layer's parameters being frozen. During the training of each RBM, the visible inputs are set to the mean activity of hidden neurons at the lower layer, while hidden neurons of the upper layer still adopt stochastic binary values according to Eq. (C4). With this layerwise training, each layer learns a nonlinear transformation of the data, and upper layers are conjectured to learn more abstract (complex) concepts, which is a key step in object and speech recognition problems [25].

The DBN learns a data distribution generated by a random RBM whose parameters follow the normal distribution

$\mathcal{N}(0, g/N)$ for weights and $\mathcal{N}(0, \sigma_b)$ for biases. $g = 0.8$ and $\sigma_b = 0.1$ unless otherwise specified. Using the RBM as a data generator allows us to control the complexity of the input data. In addition, the RBM has been used to model many real data sets (e.g., handwritten digits [24]). In numerical simulations (Fig. 3 in the main text), we generate $M = 60\,000$ training examples (each example is an N-dimensional vector) from the random RBM. Then these examples are learned by RBMs in the DBN. We divide the entire data set into mini-batches of size $B = 150$. One epoch corresponds to a sweep of the entire data set. Each RBM is trained for tens of epochs until the reconstruction error ($\varepsilon \equiv \|\mathbf{h}' - \mathbf{h}\|_2^2$) between input $\mathbf{h}$ and reconstructed input $\mathbf{h}'$ does not decrease. We use an initial learning rate of 0.12 divided by $\lceil t/10 \rceil$ at the tth epoch, and an ℓ_2 weight decay parameter of 0.0025.

APPENDIX D: ESTIMATING THE COVARIANCE STRUCTURE OF A RANDOM RESTRICTED BOLTZMANN MACHINE

To study the complexity propagation in the DBN, it is necessary to evaluate the statistics of the input data distribution, which is provided by a random RBM in the model setup of stochastic neural networks. This is because the estimated covariance can be used as a starting point from which the mean-field complexity-propagation equation iterates. In addition, characterizing the RBM representation may provide insights towards deep representations, since RBM is a building block for deep models and moreover a universal approximator of discrete distributions [31].

Given the RBM, the hidden neural activity at a higher layer (e.g., $\mathbf{s}$) can be marginalized over using the conditional independence [Eq. (C4)]; thus the distribution of the representation at a lower layer (e.g., $\boldsymbol{\sigma}$) can be expressed as

$$P(\boldsymbol{\sigma}) = \sum_{\mathbf{s}} P(\mathbf{s}, \boldsymbol{\sigma}) = \frac{1}{Z} \prod_a \left\{2\cosh\left([\mathbf{w}^{l+1}\boldsymbol{\sigma}]_a + b_a^h\right)\right\} \prod_i e^{\sigma_i b_i^v}, \tag{D1}$$

where Z is the partition function intractable for a large N. To study the statistics of the RBM representation, we need to compute the free energy function of Eq. (D1) defined as $F = -\ln Z$, where a unit inverse temperature is assumed. We use the Bethe approximation to compute an approximate free energy defined by F_{Bethe}. In physics, the Bethe approximation assumes $P(\boldsymbol{\sigma}) \approx \prod_a P_a(\boldsymbol{\sigma}_{\partial a}) \prod_i P_i(\sigma_i)^{1-N}$ [32], where a (∂a) indicates a factor node (its neighbors) representing the contribution of one hidden neuron to the joint probability [Eq. (D1)] in a factor graph representation [16]. P_a and P_i can be obtained from a variational principle of free energy optimization [16]. This approximation takes into account the correlations induced by nearest neighbors of each neuron in the factor graph, which thus improves the naive mean-field approximation where neurons are assumed independent.

Covariance of neural activity under Eq. (D1) (the so-called connected correlation in physics) can be computed from the approximate free energy using the linear response theory. However, due to the approximation, there exists a statistical inconsistency for diagonal terms computed under the Bethe approximation, i.e., $C_{ii} \neq 1 - m_i^2$. Therefore, we impose the statistical consistency of diagonal terms on a corrected free energy as $\tilde{F}_{\text{Bethe}} = F_{\text{Bethe}} - \frac{1}{2}\sum_i \Lambda_i (1 - m_i^2)$ [33–35]. Following the similar procedure in our previous work [16], we obtain the following mean-field iterative equation:

$$m_{i\to a} = \tanh\left(b_i^v - \Lambda_i m_i + \sum_{a' \in \partial i \backslash a} u_{a' \to i}\right), \tag{D2a}$$

$$u_{a' \to i} = \frac{1}{2} \ln \frac{\cosh\left(b_{a'}^h + G_{a' \to i} + w_{ia'}\right)}{\cosh\left(b_{a'}^h + G_{a' \to i} - w_{ia'}\right)}, \tag{D2b}$$

where $G_{a' \to i} \equiv \sum_{j \in \partial a' \backslash i} w_{ja'} m_{j \to a'}$ and the correction introduces an Onsager term $(-\Lambda_i m_i)$. The cavity magnetization $m_{i \to a}$ can be understood as the message passing from the visible node i to the factor node a, while the cavity bias $u_{a' \to i}$ is interpreted as the message passing from the factor node a' to the visible node i. In fact, Eq. (D2) is not closed. $\{\Lambda_i\}$ must be computed based on correlations. Therefore, we define the cavity susceptibility $\chi_{i \to a,k} \equiv \frac{\partial m_{i \to a}}{\partial b_k^v}$ [36]. According to this definition and the linear response theory, we close Eq. (D2) by obtaining the following susceptibility propagation equations:

$$\chi_{i \to a,k} = \left(1 - m_{i \to a}^2\right) \sum_{a' \in \partial i \backslash a} \Gamma_{a' \to i} \mathcal{P}_{a' \to i,k} + \delta_{ik}\left(1 - m_{i \to a}^2\right) - \Lambda_i C_{ik}, \tag{D3a}$$

$$C_{ik} = \frac{1 - m_i^2}{1 + \left(1 - m_i^2\right)\Lambda_i} \mathcal{F}_{ik}, \tag{D3b}$$

$$\Lambda_i = \frac{\mathcal{F}_{ii} - 1}{1 - m_i^2}, \tag{D3c}$$

where the full magnetization $m_i = \tanh(b_i^v - \Lambda_i m_i + \sum_{a' \in \partial i} u_{a' \to i})$, $\Gamma_{a \to i} \equiv \frac{\tanh(w_{ia})[1 - \tanh^2(b_a^h + G_{a \to i})]}{1 - \tanh^2(b_a^h + G_{a \to i}) \tanh^2(w_{ia})}$, $\mathcal{P}_{a \to i,k} \equiv \sum_{j \in \partial a \backslash i} \chi_{j \to a,k} w_{ja}$, and $\mathcal{F}_{ik} \equiv \sum_{a \in \partial i} \Gamma_{a \to i} \mathcal{P}_{a \to i,k} + \delta_{ik}$. It is easy to verify that Eq. (D3) leads to the consistency for the diagonal terms. Adding the diagonal constraint through the Lagrange multiplier $\boldsymbol{\Lambda}$ can not only solve the diagonal inconsistency problem but also improve the accuracy of estimating off-diagonal terms. After the RBM parameters (weights and biases) are specified, we run the above iterative equations [Eqs. (D2) and (D3)] from a random initialization of the messages and estimate the covariance and the associated representation dimensionality from the fixed point. These statistics are used as an initialization condition for the complexity-propagation equation [Eq. (1) in the main text] that is used to study the expressive property of the DBN with trained weights and biases.

We finally remark that, for a *trained* RBM, some components of the correlation matrix may lose the symmetry property ($C_{ij} \neq C_{ji}$), likely because of the above Bethe (cavity-based) approximation incapable of dealing with an irregular distribution of learned connection weights. The irregularity means that the distribution is divided into two parts: the bulk part is around zero, while the other part is dominated by a few large values of weights (as also observed recently in spectral

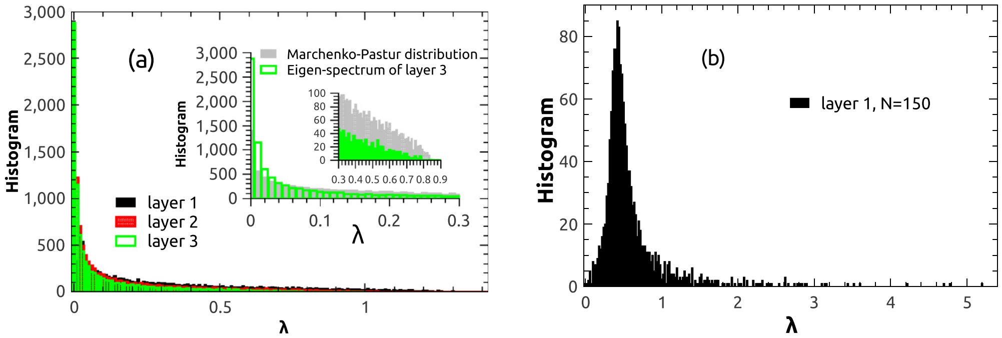

FIG. 5. The eigenvalue distribution of the covariance matrix estimated from the deep computation. (a) The distribution for the deep networks with random weights ($N = 100$). One hundred instances are used. In the inset, the distribution at deeper layers is compared with the Marchenko-Pastur law of a random-sample covariance matrix. The tail part is enlarged for comparison. (b) The distribution obtained from an unsupervised learning system of deep belief networks is strongly different from the Marchenko-Pastur law. Ten instances are used.

dynamics of learning in RBMs [37]). Our mean-field formula [Eqs. (D2) and (D3)] may offer a basis to be further improved to address this interesting special property, although one can enforce the symmetry by $[C_{ij} + C_{ji}]/2$ in our mean-field formula.

APPENDIX E: THE EIGENVALUE DISTRIBUTION OF THE COVARIANCE MATRIX

To analyze the eigenvalue distribution of the covariance matrix at each layer of deep networks, we first construct a random-sample covariance matrix. More precisely, we consider a real Wishart ensemble, where a random-sample covariance matrix is defined as $\frac{1}{N}\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}}$ in which $\boldsymbol{\xi}$ defines an $N \times P$ matrix whose entries follow independently a normal distribution $\mathcal{N}(0, \varsigma^2)$. In fact, $\boldsymbol{\xi}$ can be thought of as a random uncorrelated pattern matrix. To compare the real Wishart ensemble with the covariance matrix estimated from the mean-field theory of deep networks, we choose $P = N$, and ς^2 is obtained by matching the range of the eigenvalue. The designed random-sample covariance matrix has the Marchenko-Pastur law for the density of eigenvalues [26,27]:

$$\mu(\lambda) = \frac{1}{2\pi\lambda\varsigma^2}[(\lambda - \lambda_-)(\lambda_+ - \lambda)]^{1/2}, \tag{E1}$$

where $\lambda \in [\lambda_-, \lambda_+]$, $\lambda_- = 0$, and $\lambda_+ = 4\varsigma^2$.

For the random neural networks, the eigenvalue distribution at deep layers seems to assign a higher probability density when the eigenvalue gets close to zero, yet a lower density at the tail of the distribution [Fig. 5(a)], compared with the Marchenko-Pastur law of a random-sample covariance matrix. For the unsupervised learning system, we find that the distribution deviates significantly from the Marchenko-Pastur law of a Wishart ensemble. The distribution has a Gaussian-like bulk part together with a long tail [Fig. 5(b)]. Therefore, the dimensionality reduction and its relationship with decorrelation are a nontrivial result of the deep computation.

[1] J. J. DiCarlo and D. D. Cox, Trends Cognit. Sci. **11**, 333 (2007).

[2] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, Neuron **73**, 415 (2012).

[3] N. Kriegeskorte and R. A. Kievit, Trends Cognit. Sci. **17**, 401 (2013).

[4] D. Marr, Proc. R. Soc. London, Ser. B **176**, 161 (1970).

[5] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).

[6] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Red Hook, NY, 2016), pp. 3360–3368.

[7] B. Babadi and H. Sompolinsky, Neuron **83**, 1213 (2014).

[8] J. Kadmon and H. Sompolinsky, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Red Hook, NY, 2016), pp. 4781–4789.

[9] E. Domany, W. Kinzel, and R. Meir, J. Phys. A: Math. Gen. **22**, 2081 (1989).

[10] G. F. Elsayed and J. P. Cunningham, Nat. Neurosci. **20**, 1310 (2017).

[11] U. Guclu and M. A. J. van Gerven, J. Neurosci. **35**, 10005 (2015).

[12] H. Barlow, in *Sensory Communication*, edited by W. Rosenblith (MIT, Cambridge, MA, 1961), pp. 217–234.

[13] J. Tubiana and R. Monasson, Phys. Rev. Lett. **118**, 138301 (2017).

[14] S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, arXiv:1710.06570.

[15] A. Barra, G. Genovese, P. Sollich, and D. Tantari, Phys. Rev. E **97**, 022310 (2018).

[16] H. Huang and T. Toyoizumi, Phys. Rev. E **91**, 050101 (2015).

[17] H. Huang and T. Toyoizumi, Phys. Rev. E **94**, 062310 (2016).

[18] H. Huang, J. Stat. Mech.: Theory Exp. (2017) 053302.

[19] J. Pennington and Y. Bahri, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of Proceedings of Machine Learning Research, edited by D. Precup and

Y. W. Teh (PMLR, International Convention Centre, Sydney, Australia, 2017), pp. 2798–2806.

[20] K. Rajan, L. Abbott, and H. Sompolinsky, in *Advances in Neural Information Processing Systems 23*, edited by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Curran Associates, Red Hook, NY, 2010), pp. 1975–1983.

[21] D. L. Yamins and J. J. DiCarlo, Nat. Neurosci. **19**, 356 (2016).

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).

[23] H. A. Bethe, Proc. R. Soc. London, Ser. A **150**, 552 (1935).

[24] G. Hinton, S. Osindero, and Y. Teh, Neural Comput. **18**, 1527 (2006).

[25] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, Commun. ACM **54**, 95 (2011).

[26] V. Marcenko and L. Pastur, Math. USSR-Sb. **1**, 457 (1967).

[27] R. Cherrier, D. S. Dean, and A. Lefévre, Phys. Rev. E **67**, 046112 (2003).

[28] A. Achille and S. Soatto, arXiv:1706.01350.

[29] H. Hong, D. L. Yamins, N. J. Majaj, and J. J. DiCarlo, Nat. Neurosci. **19**, 613 (2016).

[30] M. Pagan, L. S. Urban, M. P. Wohl, and N. C. Rust, Nat. Neurosci. **16**, 1132 (2013).

[31] N. Le Roux and Y. Bengio, Neural Comput. **20**, 1631 (2008).

[32] M. Mézard and A. Montanari, *Information, Physics, and Computation* (Oxford University, Oxford, 2009).

[33] H. Huang and Y. Kabashima, Phys. Rev. E **87**, 062129 (2013).

[34] J. Raymond and F. Ricci-Tersenghi, Phys. Rev. E **87**, 052111 (2013).

[35] M. Yasuda and K. Tanaka, Phys. Rev. E **87**, 012134 (2013).

[36] S. Higuchi and M. Mezard, J. Phys.: Conf. Ser. **233**, 012003 (2010).

[37] A. Decelle, G. Fissore, and C. Furtlehner, Europhys. Lett. **119**, 60001 (2017).