Supplementary Information: A mathematical theory of semantic development in deep neural networks

Andrew M. Saxe [*], James L. McClelland [†], and Surya Ganguli [†] [‡]

[*]University of Oxford, Oxford, UK,[†]Stanford University, Stanford, CA, and [‡]Google Brain, Mountain View, CA

Acquiring Knowledge

We consider the setting where we are given a set of P examples $\left\{\mathbf{x}^i, \mathbf{y}^i\right\}, i = 1, \ldots, P$, where the input vector $\mathbf{x}^i$ identifies item i, and the output vector $\mathbf{y}^i$ is a set of features to be associated to this item. The network, defined by the weight matrices $\mathbf{W}^1, \mathbf{W}^2$ in the case of the deep network (or $\mathbf{W}^s$ in the case of the shallow network), computes its output as

$$\hat{\mathbf{y}} = \mathbf{W}^2\mathbf{W}^1\mathbf{x} \qquad \textbf{[S1]}$$

(or $\hat{\mathbf{y}} = \mathbf{W}^s\mathbf{x}$ for the shallow network). Training proceeds through stochastic gradient descent on the squared error

$$SSE(\mathbf{W}^1, \mathbf{W}^2) = \frac{1}{2}\left\|\mathbf{y}^i - \hat{\mathbf{y}}^i\right\|^2, \qquad \textbf{[S2]}$$

with learning rate λ, yielding the updates

$$\begin{aligned} \Delta\mathbf{W}^1 &= -\lambda\frac{\partial}{\partial\mathbf{W}^1}SSE(\mathbf{W}^1, \mathbf{W}^2) \\ \Delta\mathbf{W}^2 &= -\lambda\frac{\partial}{\partial\mathbf{W}^2}SSE(\mathbf{W}^1, \mathbf{W}^2). \end{aligned}$$

While the structure of the error surface in such deep linear networks is known [1], our focus here is on the dynamics of the learning process. Substituting Eqn. (S1) into Eqn. (S2) and taking derivatives yields the update rules specified in Eqn. (1) of the main text,

$$\begin{aligned} \Delta\mathbf{W}^1 &= \lambda{\mathbf{W}^2}^T\left(\mathbf{y}^i - \hat{\mathbf{y}}^i\right)\mathbf{x}^{iT}, \\ \Delta\mathbf{W}^2 &= \lambda\left(\mathbf{y}^i - \hat{\mathbf{y}}^i\right)\mathbf{h}^{iT}, \end{aligned}$$

where $\mathbf{h}^i = \mathbf{W}^1\mathbf{x}^i$ is the hidden layer activity for example i. We note that for simplicity these update equations describe the case in which all features of an item are observed whenever it is presented. However, this is not necessary: when a random subset of features is observed and error-corrective learning is applied only to the observed features, then if learning is gradual the average update will be equivalent to that from observing all features, up to a scale factor which can be absorbed into the learning rate.

This update is identical to that produced by the standard backpropagation algorithm, as can be seen by noting that the error $\mathbf{e}^i = \mathbf{y}^i - \hat{\mathbf{y}}^i$, and the backpropagated delta signal $\delta^i = {\mathbf{W}^2}^T\mathbf{e}$, such that these update equations can be rewritten as

$$\begin{aligned} \Delta\mathbf{W}^1 &= \lambda\delta^i\mathbf{x}^{iT}, \\ \Delta\mathbf{W}^2 &= \lambda\mathbf{e}^i\mathbf{h}^{iT}. \end{aligned}$$

We now derive the average weight change under these updates over the course of an epoch, when learning is gradual. We assume that all inputs $i = 1, \cdots, P$ are presented (possibly in random order), with updates applied after each. In the updates above, the weights change on each stimulus presentation and hence are functions of i, which we denote as $\mathbf{W}^1[i], \mathbf{W}^2[i]$. Our goal is to recover equations describing the dynamics of the weights across epochs, which we denote as $\mathbf{W}^1(t), \mathbf{W}^2(t)$. Here, $t = 1$ corresponds to viewing P examples, $t = 2$ corresponds to viewing $2P$ examples, and so on. In general throughout the main text and supplement we suppress this dependence for clarity where it is clear from context and simply write $\mathbf{W}^1, \mathbf{W}^2$.

When learning is gradual ($\lambda \ll 1$), the weights change minimally on each given example and hence $\mathbf{W}^1[i] \approx \mathbf{W}^1(t)$ for all patterns in epoch t. The total weight change over an epoch is thus

$$\begin{aligned} \Delta\mathbf{W}^1(t) &= \sum_{i=1}^{P}\lambda\mathbf{W}^2[i]^T\left(\mathbf{y}^i - \hat{\mathbf{y}}^i\right)\mathbf{x}^{iT}, \\ &= \sum_{i=i}^{P}\lambda\mathbf{W}^2[i]^T\left(\mathbf{y}^i - \mathbf{W}^2[i]\mathbf{W}^1[i]\mathbf{x}^i\right)\mathbf{x}^{iT}, \\ &\approx \sum_{i=1}^{P}\lambda\mathbf{W}^2(t)^T\left(\mathbf{y}^i - \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{x}^i\right)\mathbf{x}^{iT}, \\ &= \lambda P\mathbf{W}^2(t)^T(E[\mathbf{y}\mathbf{x}^T] - \mathbf{W}^2(t)\mathbf{W}^1(t)E[\mathbf{x}\mathbf{x}^T]), \\ &= \lambda P{\mathbf{W}^2}^T\left(\mathbf{\Sigma}^{yx} - \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{\Sigma}^x\right) \\ \Delta\mathbf{W}^2(t) &= \sum_{i=1}^{P}\lambda\left(\mathbf{y}^i - \hat{\mathbf{y}}^i\right)\mathbf{h}^{iT} \\ &= \sum_{i=i}^{P}\lambda\left(\mathbf{y}^i - \mathbf{W}^2[i]\mathbf{W}^1[i]\mathbf{x}^i\right)\mathbf{x}^{iT}\mathbf{W}^1[i]^T, \\ &\approx \sum_{i=1}^{P}\lambda\left(\mathbf{y}^i - \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{x}^i\right)\mathbf{x}^{iT}\mathbf{W}^1(t)^T, \\ &= \lambda P(E[\mathbf{y}\mathbf{x}^T] - \mathbf{W}^2(t)\mathbf{W}^1(t)E[\mathbf{x}\mathbf{x}^T])\mathbf{W}^1(t)^T, \\ &= \lambda P\left(\mathbf{\Sigma}^{yx} - \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{\Sigma}^x\right)\mathbf{W}^1(t)^T. \end{aligned}$$

where $\mathbf{\Sigma}^x \equiv E[\mathbf{x}\mathbf{x}^T]$ is an $N_1 \times N_1$ input correlation matrix, and $\mathbf{\Sigma}^{yx} \equiv E[\mathbf{y}\mathbf{x}^T]$ is an $N_3 \times N_1$ input-output correlation matrix. So long as λ is small, we can take the continuum limit of this difference equation to obtain Eqns. (2)-(3) of the main text,

$$\begin{aligned} \tau\frac{d}{dt}\mathbf{W}^1 &= {\mathbf{W}^2}^T\left(\mathbf{\Sigma}^{yx} - \mathbf{W}^2\mathbf{W}^1\mathbf{\Sigma}^x\right), & \textbf{[S3]} \\ \tau\frac{d}{dt}\mathbf{W}^2 &= \left(\mathbf{\Sigma}^{yx} - \mathbf{W}^2\mathbf{W}^1\mathbf{\Sigma}^x\right){\mathbf{W}^1}^T. & \textbf{[S4]} \end{aligned}$$

where the time constant

$$\tau \equiv \frac{1}{P\lambda}. \qquad \textbf{[S5]}$$

In the above, the weights are now a function of a continuous parameter that with slight abuse of notation we also denote as t, such that as t goes from 0 to 1 the network has seen P examples.

Explicit solutions from tabula rasa. To solve for the dynamics of $\mathbf{W}^1, \mathbf{W}^2$ over time [2, 3], we decompose the input-output correla-

www.pnas.org/cgi/doi/10.1073/pnas.1820226116

tions through the singular value decomposition (SVD),

$$\mathbf{\Sigma}^{yx} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_{\alpha=1}^{N_1} s_\alpha \mathbf{u}^\alpha \mathbf{v}^{\alpha T},$$

and then change variables to $\overline{\mathbf{W}}^1, \overline{\mathbf{W}}^2$ where

$$\begin{aligned} \mathbf{W}^1 &= \mathbf{R}\overline{\mathbf{W}}^1\mathbf{V}^T, & \textbf{[S6]} \\ \mathbf{W}^2 &= \mathbf{U}\overline{\mathbf{W}}^2\mathbf{R}^T, & \textbf{[S7]} \end{aligned}$$

and $\mathbf{R}$ is an arbitrary orthogonal matrix ($\mathbf{R}^T\mathbf{R} = I$). These variables analyze the dynamics in the basis defined by the SVD.

We additionally assume that the input correlations are white ($\mathbf{\Sigma}^x = \mathbf{I}$), such that the influence of perceptual similarity is minimal (but see a generalization to non-white input in Fig. S3). Our input, which is quite high-level, could be construed as the output of a 'conventional' image recognition deep neural network, or even further upstream as input to the hippocampus in the dentate gyrus. Indeed a major function of sparse representations in the dentate gyrus is thought to involve pattern separation, a process in which input representations that are similar are orthogonalized to form more dissimilar or whitened representations. From this whitened input with little perceptual similarity structure, the network must learn to map each item to its semantic properties, discovering deeper semantic relations that are unrelated to perceptual similarity.

Substituting into Eqns. (S3)-(S4) and using the assumption $\mathbf{\Sigma}^x = \mathbf{I}$ we have

$$\begin{aligned} \tau\frac{d}{dt}(\mathbf{R}\overline{\mathbf{W}}^1\mathbf{V}^T) &= \mathbf{R}\overline{\mathbf{W}}^{2^T}\mathbf{U}^T\left(\mathbf{\Sigma}^{yx} - \mathbf{U}\overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\mathbf{V}^T\mathbf{\Sigma}^x\right), \\ \tau\frac{d}{dt}\overline{\mathbf{W}}^1 &= \overline{\mathbf{W}}^{2^T}\mathbf{U}^T\left(\mathbf{U}\mathbf{S}\mathbf{V}^T - \mathbf{U}\overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\mathbf{V}^T\right)\mathbf{V}, \\ &= \overline{\mathbf{W}}^{2^T}\left(\mathbf{S} - \overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\right), & \textbf{[S8]} \\ \tau\frac{d}{dt}(\mathbf{U}\overline{\mathbf{W}}^2\mathbf{R}^T) &= \left(\mathbf{\Sigma}^{yx} - \mathbf{U}\overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\mathbf{V}^T\mathbf{\Sigma}^x\right)\mathbf{V}^T\overline{\mathbf{W}}^{1^T}\mathbf{R}^T \\ \tau\frac{d}{dt}\overline{\mathbf{W}}^2 &= \mathbf{U}^T\left(\mathbf{U}\mathbf{S}\mathbf{V}^T - \mathbf{U}\overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\mathbf{V}^T\right)\mathbf{V}\overline{\mathbf{W}}^{1^T} \\ &= \left(\mathbf{S} - \overline{\mathbf{W}}^2\overline{\mathbf{W}}^1\right)\overline{\mathbf{W}}^{1^T} & \textbf{[S9]} \end{aligned}$$

where we have made use of the orthogonality of the SVD bases, i.e., $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. Importantly, the change of variables is applied *after* deriving the gradient descent update equations in the untransformed coordinate system. Gradient descent is not invariant to reparametrization and so performing this change of variables *before* would correspond to analyzing potentially different dynamics.

Equations (S8)-(S9) have a simplified form because $\mathbf{S}$ is a diagonal matrix. Hence if $\overline{\mathbf{W}}^1$ and $\overline{\mathbf{W}}^2$ are also diagonal, the dynamics decouple into N_1 independent systems. We study the dynamics in this decoupled regime where $\overline{\mathbf{W}}^1(0)$ and $\overline{\mathbf{W}}^2(0)$ are diagonal. Off-diagonal elements represent coupling between different modes of the SVD, and decay to zero under the dynamics. Hence the decoupled solutions we find also provide good approximations to the full solution when $\overline{\mathbf{W}}^1(0)$ and $\overline{\mathbf{W}}^2(0)$ are initialized with small random weights, as shown through simulation (red lines in Fig. 3C of the main text).

In particular, let $c_\alpha = \overline{\mathbf{W}}^1{}_{\alpha\alpha}$ and $d_\alpha = \overline{\mathbf{W}}^2{}_{\alpha\alpha}$ be the α^{th} diagonal element of the first and second matrices, encoding the strength of mode α transmitted by the input-to-hidden and hidden-to-output weights respectively. We have the scalar dynamics

$$\begin{aligned} \tau\frac{d}{dt}c_\alpha &= d_\alpha(s_\alpha - c_\alpha d_\alpha) \\ \tau\frac{d}{dt}d_\alpha &= c_\alpha(s_\alpha - c_\alpha d_\alpha) \end{aligned}$$

for $\alpha = 1, \cdots, N_1$. In general, c_α can differ from d_α, but if weights are initialized to small initial values, these will be roughly equal. We therefore study *balanced* solutions where $c_\alpha = d_\alpha$. In particular, we will track the overall strength of a particular mode with the single scalar $a_\alpha = c_\alpha d_\alpha$, with dynamics

$$\begin{aligned} \tau\frac{d}{dt}a_\alpha &= c_\alpha\left(\tau\frac{d}{dt}d_\alpha\right) + \tau d_\alpha\left(\tau\frac{d}{dt}c_\alpha\right) \\ &= c_\alpha c_\alpha(s_\alpha - c_\alpha d_\alpha) + \tau d_\alpha d_\alpha(s_\alpha - c_\alpha d_\alpha) \\ &= 2a_\alpha(s_\alpha - a_\alpha). \end{aligned}$$

This is a separable differential equation which can be integrated to yield (here we suppress the dependence on α for clarity),

$$t = \frac{\tau}{2}\int_{a^0}^{a^f}\frac{da}{a(s-a)} = \frac{\tau}{2s}\ln\frac{a^f(s-a^0)}{a^0(s-a^f)} \qquad \textbf{[S10]}$$

where t is the time required to travel from an initial strength $a(0) = a^0$ to a final strength $a(t) = a^f$.

The entire time course of learning can be found by solving for a_f, yielding Eqn. (6) of the main text,

$$a_\alpha(t) = \frac{s_\alpha e^{2s_\alpha t/\tau}}{e^{2s_\alpha t/\tau} - 1 + s_\alpha/a_\alpha^0}.$$

Next, we undo the change of variables to recover the full solution. Define the time-dependent diagonal matrix $\mathbf{A}(t)$ to have diagonal elements $(\mathbf{A}(t))_{\alpha\alpha} = a_\alpha(t)$. Then by the definition of a_α, c_α, and d_α, we have $\mathbf{A}(t) = \overline{\mathbf{W}}^2(t)\overline{\mathbf{W}}^1(t)$. Inverting the change of variables in Eqns. (S6)-(S7), we recover Eqn. (5) of the main text, the overall input-output map of the network:

$$\mathbf{W}^2(t)\mathbf{W}^1(t) = \mathbf{U}\overline{\mathbf{W}}^2(t)\overline{\mathbf{W}}^1(t)\mathbf{V}^T = \mathbf{U}\mathbf{A}(t)\mathbf{V}^T.$$

This solution is not fully general, but rather provides a good account of the dynamics of learning in the network in a particular regime. To summarize our assumptions, the solution is applicable in the *gradual learning* regime ($\lambda \ll 1$), when initial mode strengths in each layer are roughly *balanced* ($c_\alpha = d_\alpha$), and approximately *decoupled* (off diagonal elements of $\overline{\mathbf{W}}^1, \overline{\mathbf{W}}^2 \ll 1$). These latter two conditions hold approximately when weights are initialized with small random values, and hence we call this solution the solution from *tabula rasa*. Notably, these solutions do not describe the dynamics if substantial knowledge is already embedded in the network when learning commences. When substantial prior knowledge is present, learning can have very different dynamics corresponding to unequal initial information in each layer ($c_\alpha \neq d_\alpha$) and/or strong coupling between modes (large off-diagonal elements in $\overline{\mathbf{W}}^1, \overline{\mathbf{W}}^2$).

How small must λ be to count as gradual learning? The requirement on λ is that the fastest dynamical timescale in (S3)-(S4) is much longer than 1, which is the timescale of a single learning epoch. The fastest timescale arises from the largest singular value s_1 and is $O(\tau/s_1)$ (cf Eqn. (S10)). Hence the requirement $\tau/s_1 \gg 1$ and the definition of τ yields the condition

$$\lambda \ll \frac{1}{s_1 P}.$$

Hence stronger structure, as measured by the SVD, or more training samples, necessitates a smaller learning rate.

The dynamics permit an explicit curve for the sum squared error over the course of learning. This is

$$\begin{aligned} SSE(t) &= \frac{1}{2}\sum_{i=1}^{P}||\mathbf{y}^i - \hat{\mathbf{y}}^i||_2^2 \\ &= \frac{1}{2}\mathrm{Tr}\sum_{i=1}^{P}\mathbf{y}^i{\mathbf{y}^i}^T - 2\hat{\mathbf{y}}^i{\mathbf{y}^i}^T + \hat{\mathbf{y}}^i{\hat{\mathbf{y}}^i}^T \\ &= \frac{P}{2}\mathrm{Tr}\boldsymbol{\Sigma}^y - P\mathrm{Tr}\boldsymbol{\Sigma}^{yx}{\mathbf{W}_{tot}}^T + \frac{P}{2}\mathrm{Tr}\mathbf{W}_{tot}\boldsymbol{\Sigma}^x{\mathbf{W}_{tot}}^T \\ &= \frac{P}{2}\mathrm{Tr}\boldsymbol{\Sigma}^y - P\mathrm{Tr}\mathbf{S}\mathbf{A}(t) + \frac{P}{2}\mathrm{Tr}\mathbf{A}(t)^2 \\ &= \frac{P}{2}\mathrm{Tr}\boldsymbol{\Sigma}^y - P\mathrm{Tr}\left[\left(\mathbf{S} - \frac{1}{2}\mathbf{A}(t)\right)\mathbf{A}(t)\right]. \end{aligned}$$

Early in learning, $\mathbf{A}(t) \approx 0$ and the error is proportional to $\mathrm{Tr}\boldsymbol{\Sigma}^y$, the variance in the output. Late in learning, $\mathbf{A}(t) \approx \mathbf{S}$ and the error is proportional to $\mathrm{Tr}\boldsymbol{\Sigma}^y - \mathrm{Tr}\mathbf{S}^2$, the output variance which cannot be explained by a linear model.

We emphasize that these solutions describe how the network converges to the linear least squares solution for the provided data. Because a deep linear network cannot represent nonlinear functions, this solution may have nonzero training error. In several of the settings we consider subsequently, we operate in the regime $N_1 = P$ such that the number of inputs is equal to the input dimension. In this case, even a linear network can always attain zero training error provided that the inputs are linearly independent (as they must be if they are white). However, if the ground truth input-output map is nonlinear then a network that achieves zero training error will not achieve good test error, depending on the degree of nonlinearity in the ground truth map. We restrict our attention to semantic settings in which the ground truth associations between objects and high level semantic features can indeed be implemented by a linear map. For analysis of generalization dynamics in linear networks see [2, 4, 5, 6].

In the *tabula rasa* regime, the individual weight matrices are given by

$$\begin{aligned} \mathbf{W}^1(t) &= \mathbf{R}\overline{\mathbf{W}}^1\mathbf{V}^T = \mathbf{R}\sqrt{\mathbf{A}(t)}\mathbf{V}^T, \\ \mathbf{W}^2(t) &= \mathbf{U}\overline{\mathbf{W}}^2\mathbf{R}^T = \mathbf{U}\sqrt{\mathbf{A}(t)}\mathbf{R}^T, \end{aligned}$$

due to the fact that $c_\alpha = d_\alpha = \sqrt{a_\alpha}$.

The full space of weights implementing the same input-output map is

$$\begin{aligned} \mathbf{W}^1(t) &= \mathbf{Q}\sqrt{\mathbf{A}(t)}\mathbf{V}^T, \\ \mathbf{W}^2(t) &= \mathbf{U}\sqrt{\mathbf{A}(t)}\mathbf{Q}^{-1} \end{aligned}$$

for any invertible matrix $\mathbf{Q}$.

Shallow network

Analogous solutions may be found for the shallow network. In particular, the gradient of the sum of square error

$$SSE(\mathbf{W}^s) = \frac{1}{2}\left\|\mathbf{y}^i - \hat{\mathbf{y}}^i\right\|^2,$$

yields the update

$$\Delta\mathbf{W}^s = \lambda(\mathbf{y}^i - \hat{\mathbf{y}}^i)\mathbf{x}^{iT}.$$

Averaging as before over an epoch yields the dynamics

$$\tau\frac{d}{dt}\mathbf{W}^s = \boldsymbol{\Sigma}^{yx} - \mathbf{W}^s\boldsymbol{\Sigma}^x,$$

a simple linear differential equation which may be solved explicitly. To make the solution readily comparable to the deep network dynamics, we change variables to $\mathbf{W}^s = \mathbf{U}\overline{\mathbf{W}}^s\mathbf{V}^T$,

$$\begin{aligned} \tau\frac{d}{dt}(\mathbf{U}\overline{\mathbf{W}}^s\mathbf{V}^T) &= \boldsymbol{\Sigma}^{yx} - \mathbf{U}\overline{\mathbf{W}}^s\mathbf{V}^T\boldsymbol{\Sigma}^x \\ \tau\frac{d}{dt}\overline{\mathbf{W}}^s &= \mathbf{S} - \overline{\mathbf{W}}^s. \end{aligned}$$

Defining ${\overline{\mathbf{W}}^s}_{\alpha\alpha} = b_\alpha$ and assuming decoupled initial conditions gives the scalar dynamics

$$\tau\frac{d}{dt}b_\alpha = s_\alpha - b_\alpha.$$

Integrating this simple separable differential equation yields

$$t = \tau\ln\frac{s_\alpha - b_\alpha^0}{s_\alpha - b_\alpha^f} \qquad \textbf{[S11]}$$

which can be inverted to find the full time course

$$b_\alpha(t) = s_\alpha\left(1 - e^{-t/\tau}\right) + b_\alpha^0 e^{-t/\tau}.$$

Undoing the change of variables yields the weight trajectory

$$\mathbf{W}^s = \mathbf{U}\mathbf{B}(t)\mathbf{V}^T$$

where $\mathbf{B}(t)$ is a diagonal matrix with elements $(\mathbf{B}(t))_{\alpha\alpha} = b_\alpha(t)$.

Simulation details for solutions from tabula rasa. The simulation results shown in Fig. 3 are for a minimal hand-crafted hierarchical dataset with $N_3 = 7$ features, $N_2 = 16$ hidden units, and $N_1 = P = 4$ items. Inputs were encoded with one-hot vectors, but we emphasize that the assumption of localist inputs is purely for ease of interpretation, and any orthogonal distributed representation would yield identical dynamics. The input-output correlations are

$$\begin{aligned} \boldsymbol{\Sigma}^{yx} &= 0.7P\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \boldsymbol{\Sigma}^x &= \mathbf{I}. \end{aligned}$$

We used $\tau = 1$, $\lambda = \frac{1}{P}$, and $a_0 = 0.0001$.

Rapid stage-like transitions due to depth. To understand the time required to learn a particular mode, we calculate the time t necessary for the learning process to move from an initial state with little knowledge, $a(0) = \epsilon$ for some small $\epsilon \ll 1$, to a state which has reached within ϵ of its final asymptote, $a(t_f) = s - \epsilon$. It is necessary to introduce this cutoff parameter ϵ because, first, deep networks initialized with weights exactly equal to zero have no dynamics, and second, because both shallow and deep networks do not reach their asymptotic values in finite time. Therefore we consider networks initialized a small distance away from zero, and consider learning to be complete when they arrive within a small distance of the correct answer. For the deep network, substituting these initial and final conditions into Eqn. (S10) yields

$$\begin{aligned} t &= \frac{\tau}{2s}\ln\frac{(s-\epsilon)^2}{\epsilon^2} \\ &\approx \frac{\tau}{s}\ln\frac{s}{\epsilon} \end{aligned}$$

for small ϵ.

For the shallow network, by contrast, substituting into Eqn. (S11) yields

$$\begin{aligned} t &= \tau \ln \frac{s-\epsilon}{\epsilon} \\ &\approx \tau \ln \frac{s}{\epsilon} \end{aligned}$$

for small ϵ. Hence these networks exhibit fundamentally different learning timescales, due to the $1/s$ term in the deep network, which strongly orders the learning times of different modes by their singular value size.

Beyond this difference in learning timescale, there is a qualitative change in the shape of the learning trajectory. Deep networks exhibit sigmoidal learning trajectories for each mode, while shallow networks undergo simple exponential approach. The sigmoidal trajectories in deep networks give a quasi-stage-like character to the learning dynamics: for much of the total learning time, progress is very slow; then in a brief transition period, performance rapidly improves to near its final value. How does the length of the transitional period compare to the total learning time? We define the transitional period as the time required to go from a strength $a(t_s) = \epsilon$ to within a small distance of the asymptote, $a(t_f) = s - \epsilon$, as before. Here t_s is the time marking the start of the transition period and t_f is the time marking the end. Then we introduce a new cutoff $\epsilon_0 < \epsilon$ for the starting strength of the mode, $a(0) = \epsilon_0$. The length of the transition period $t_{trans} = t_f - t_s$ is

$$t_{trans} = \frac{\tau}{2s} \ln \frac{(s-\epsilon)^2}{\epsilon^2},$$

while the total learning time $t_{tot} = t_f$ starting from the mode strength ϵ_0 is

$$t_{tot} = \frac{\tau}{2s} \ln \frac{(s-\epsilon)(s-\epsilon_0)}{\epsilon_0 \epsilon}.$$

Hence for a fixed ϵ defining the transition period, the total training time increases as the initial strength on a mode ϵ_0 decreases toward zero. In the limit $\epsilon_0 \to 0$, the ratio of the length of time in the transition period to the total training time is

$$\lim_{\epsilon_0 \to 0} t_{trans}/t_{tot} = 0,$$

such that the duration of the transition is exceptionally brief relative to the total training time. Hence deep networks can exhibit stage-like transitions.

By contrast, for the shallow network,

$$\begin{aligned} t_{trans} &= \tau \ln \frac{s-\epsilon}{\epsilon} \\ t_{tot} &= \tau \ln \frac{s-\epsilon_0}{\epsilon} \end{aligned}$$

and the ratio limits to $t_{trans}/t_{tot} = 1$ for fixed small ϵ and $\epsilon_0 \to 0$, indicating that the transition period is as long as the total training time and transitions are not stage-like.

Generalization to arbitrarily deep linear networks with task-aligned input correlations. To understand the impact of depth and input correlations, we study similar dynamics for arbitrary depth networks with a relaxation of the assumption of whitened inputs [3]. The results of this section show that, for deep linear networks, the main qualitative difference in learning dynamics is between shallow zero hidden layer networks and deep networks. Extra depth makes a quantitative change in learning speed, but preserves qualitative features like stage-like transitions, illusory correlations, and the formatting of internal representations.

Consider a deep linear network with D layers including the input and output layer, such that $D = 2$ corresponds to a shallow network with no hidden layers and $D = 3$ corresponds to a network with a single hidden layer. Gradient descent on the $D-1$ weight matrices in this network yields the dynamics

$$\tau \frac{d}{dt}\mathbf{W}^l = \left(\prod_{i=l+1}^{D-1}\mathbf{W}^i\right)^T \left[\mathbf{\Sigma}^{yx} - \left(\prod_{i=1}^{D-1}\mathbf{W}^i\right)\mathbf{\Sigma}^x\right]\left(\prod_{i=1}^{l-1}\mathbf{W}^i\right)^T \qquad \textbf{[S12]}$$

for $l = 1, \cdots, D-1$, where the product $\prod_{i=a}^{b}\mathbf{W}^i$ is taken to be the identity $\mathbf{I}$ when $a > b$. We now decouple these dynamics by making the change of variables

$$\mathbf{W}^l = \mathbf{R}_{l+1}\overline{\mathbf{W}}^l\mathbf{R}_l^T \quad l = 1, \cdots, D-1 \qquad \textbf{[S13]}$$

where the matrices $\mathbf{R}^l$ have orthonormal columns such that $\mathbf{R}_l^T\mathbf{R} = \mathbf{I}$ for all l. Additionally, we take $\mathbf{R}_{D-1} = \mathbf{U}$ and $\mathbf{R}_1 = \mathbf{V}$. With this change of variables, the dynamics decouple as before, provided that the input correlation matrix is diagonalized by the object analyzer vectors of the input-output covariance matrix, i.e., $\mathbf{\Sigma}^x = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ where $\mathbf{\Lambda}$ is a diagonal matrix of variances on each dimension. This yields the dynamics

$$\tau \frac{d}{dt}\overline{\mathbf{W}}^l = \left(\prod_{i=l+1}^{D-1}\overline{\mathbf{W}}^i\right)^T \left[\mathbf{S} - \left(\prod_{i=1}^{D-1}\overline{\mathbf{W}}^i\right)\mathbf{\Lambda}\right]\left(\prod_{i=1}^{l-1}\overline{\mathbf{W}}^i\right)^T. \qquad \textbf{[S14]}$$

If the initial conditions $\overline{\mathbf{W}}^l(0)$ start diagonal, they remain so. The overall network input-output map is again given by

$$\mathbf{W}^{tot} = \mathbf{U}\mathbf{A}(t)\mathbf{V}^T \qquad \textbf{[S15]}$$

where $\mathbf{A}(t)$ is a diagonal matrix of effective singular values. The effective singular value $a_\alpha(t) \equiv \mathbf{A}(t)_{\alpha\alpha}$ corresponding to mode α is the product of scalar layer strengths $b_l, l = 1, \cdots, D-1$ such that $a(t) = b_{D-1}(t)b_{D-2}(t)\cdots b_1(t)$ (where here we have suppressed the dependence on α for clarity). The continuous time gradient descent update on one specific b_l is

$$\tau \dot{b}_l = (s - a\lambda)\,\lambda \prod_{j=1, j\neq l}^{D-1} b_j(t), \quad l = 1, \cdots, D-1, \qquad \textbf{[S16]}$$

where s and λ are the corresponding singular value $\mathbf{S}_{\alpha\alpha}$ and input variance $\mathbf{\Lambda}_{\alpha\alpha}$ of the input mode, respectively. The overall update to a is therefore

$$\begin{aligned} \tau \dot{a} &= \sum_{i=1}^{D-1}\left[\prod_{j=1, j\neq i}^{D-1} b_j(t)\right]\dot{b}_i(t) & \textbf{[S17]} \\ &= (s - a\lambda)\,\lambda \sum_{i=1}^{D-1}\left[\prod_{j=1, j\neq i}^{D-1} b_j(t)\right]^2. & \textbf{[S18]} \end{aligned}$$

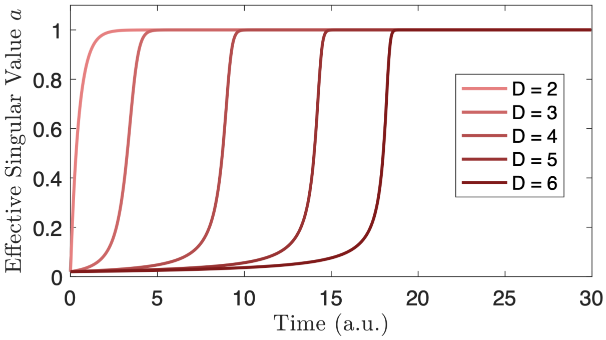

Fig. S1. Dynamics in arbitrarily deep linear networks. Effective singular value dynamics as a function of depth for a specific input-output mode with $s = 1$, $\lambda = 1$. Deeper networks train slightly slower, but all show a similar sigmoidal learning trajectory in contrast to the exponential approach trajectory of the shallow network ($D = 2$). Other parameters: $a(0) = 0.02, c = 2, s_m = \lambda_m = 1$.

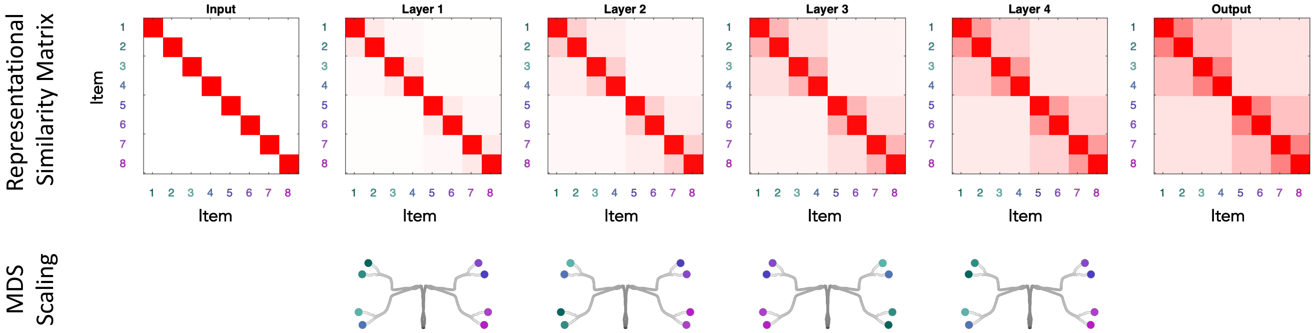

Fig. S2. Representations in arbitrarily deep linear networks. A $D = 6$ network with four hidden layers learns the hierarchically structured dataset from Fig. 4 of the main text. Top row: Representational similarity matrices across hidden layers show the emerging hierarchical structure in internal representations, through which the input correlations gradually transform into the required output correlations. Bottom row: Multidimensional scaling plots of internal representations show that every layer has a similar hierarchical representation and similar qualitative dynamics over the course of learning.

We now assume that each layer strength b_l begins roughly equal, $b_l(t) = b(t), \quad l = 1, \cdots, D-1$, such that $a(t) = b(t)^{D-1}$. This is a reasonable assumption for the limit of small random weights initialized with equal variance in each layer. The dynamics in the balanced regime simplify to

$$\tau \dot{a} = (s - a\lambda)\,\lambda \sum_{i=1}^{D-1} \left[b(t)^{D-2}\right]^2 \qquad \textbf{[S19]}$$

$$= (D-1)\left(s\lambda - a\lambda^2\right) a^{\frac{2D-4}{D-1}}. \qquad \textbf{[S20]}$$

Eqn. (S20) describes the evolution of each effective singular value as a function of its associated singular value s and input variance λ in the environment, the depth D of the network, and the inverse learning rate τ. While no general solution exists, this separable differential equation may be solved analytically for any specific depth (the results in the preceding sections perform this integration for $D = 2$ and $D = 3$ with $\lambda = 1$).

This continuous time analysis supposes that we use a fixed inverse learning rate τ for networks of different depth. However, the discrete updates will not remain stable as depth increases with a constant learning rate. To correctly estimate learning speed as a function of depth requires an estimate of how the maximum stable learning rate scales with depth. We estimate this by calculating the maximum eigenvalue of the Hessian over the relevant region. This gives,

$$\tau \ddot{a} = (2D-4)\left(s\lambda - a\lambda^2\right) a^{\frac{D-3}{D-1}} - (D-1)\lambda^2 a^{\frac{2D-4}{D-1}} \qquad \textbf{[S21]}$$

which in the interval $[0, s/\lambda]$ attains its maximum at the fixed point $a = s/\lambda$,

$$\tau \ddot{a} = (D-1)\lambda^2 \left(\frac{s}{\lambda}\right)^{\frac{2D-4}{D-1}}. \qquad \textbf{[S22]}$$

The learning rate is a global variable that cannot be chosen independently for each mode. Therefore it must be chosen such that the least-stable mode remains stable. Let s_m and λ_m be the singular value and input variance of the mode m which maximizes Eqn. (S22). We set the inverse learning rate τ proportional to this with proportionality constant c, giving

$$\tau = \frac{1}{c}(D-1)\lambda_m^2 \left(\frac{s_m}{\lambda_m}\right)^{\frac{2D-4}{D-1}}. \qquad \textbf{[S23]}$$

Substituting into the dynamics equation (S20), we have

$$\dot{a} = c\lambda_m^{-2}\left(\frac{s_m}{\lambda_m}\right)^{-\frac{2D-4}{D-1}}\left(s\lambda - a\lambda^2\right) a^{\frac{2D-4}{D-1}} \qquad \textbf{[S24]}$$

$$= c\lambda_m^{-2}\left(s\lambda - a\lambda^2\right)\left(\frac{\lambda_m a}{s_m}\right)^{\frac{2D-4}{D-1}} \qquad \textbf{[S25]}$$

Eqn. (S25) describes the mode dynamics for different depths when the learning rate has been approximately optimally scaled for each, permitting a comparison of learning speed across depth. Fig. S1 shows an example of the resulting dynamics. In essence, depth modestly slows training speeds but does not qualitatively change the dynamics. In deep linear networks, each mode undergoes a stage-like transition to its final asymptotic value.

Example learned internal representations are depicted in Fig. S2 for a network learning hierarchical correlations from white inputs. At layer l, each input-output mode is expressed with strength $b(t)^l$. The representational similarity matrix at layer l is therefore

$$\mathbf{VA}(t)^{\frac{l-1}{D-1}}\mathbf{V}^T, \qquad \textbf{[S26]}$$

such that the representational similarity progressively emerges across layers. Notably, all layers contain the same basic organization. That is, in deep networks, it is not the case that deeper layers represent broader distinctions in the hierarchy while shallower layers represent finer distinctions. All layers represent all distinctions, but the strength of the representation grows over the depth of the network. As shown in Fig. S2, this means that MDS scalings at every layer look qualitatively similar.

Learning dynamics with input correlations. For simplicity, we have assumed that the input correlations are white ($\mathbf{\Sigma}^x = \mathbf{I}$), corresponding to a setting in which there is little perceptual similarity between different items. In many settings, however, there are strong input correlations which can alter the learning dynamics. A given task may require boosting signal in low variance directions of the input, or ignoring signal from high variance directions of the input. Here we give explicit solutions for a single hidden layer deep linear network trained with certain input correlations. Specifically, we consider input correlations that are aligned with the object analyzer vectors of the input-output map. This generalization, among other applications, permits description of learning dynamics for autoencoders in which the input correlations match the output correlations.

In particular, integrating Eqn. (S20) for the special case of a single hidden layer network ($D = 3$) yields the learning time required to go from an initial mode strength a^0 to a final mode strength a^f of

$$t = \frac{\tau}{2s\lambda}\ln\frac{a^f(s - \lambda a^0)}{a^0(s - \lambda a^f)}. \qquad \textbf{[S27]}$$

Hence learning speed scales approximately as $O(\frac{1}{s\lambda})$. As the magnitude of the relevant input correlations diminish, the required learning time increases. Solving this equation for a^f yields the time course

$$a(t) = \frac{s/\lambda}{1 - \left(1 - \frac{s}{\lambda a_0}\right) e^{-\frac{2s\lambda}{\tau}t}}. \qquad \textbf{[S28]}$$

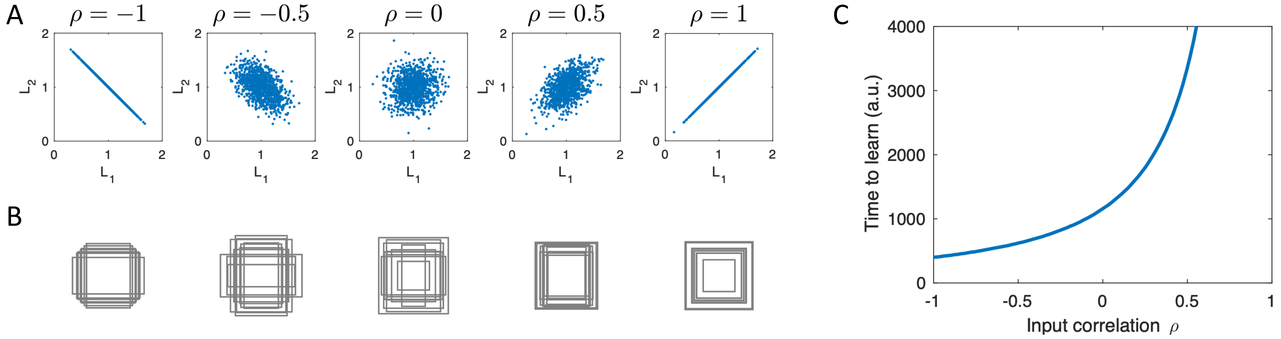

Fig. S3. Learning dynamics with input correlations. A deep linear network is tasked with reporting the difference in length $L_1 - L_2$ between two sides of a rectangle. (A) Example distributions of side lengths with different levels of input correlation ρ. (B) Example rectangle stimuli drawn from the distribution in the panel above. The difficulty of the discrimination increases with increasing correlation. (C) Time required to reach 90% of asymptotic error in a deep linear network as a function of the input correlation. Training time increases as the relevant variance in the input decreases.

Hence the dynamics asymptote to s/λ.

This solution allows a description of the dynamics of learning for autoencoders, for which $\mathbf{\Sigma}^{yx} = \mathbf{\Sigma}^{x}$ and $s = \lambda$. In particular the learning time scale is proportional to $O(\frac{1}{\lambda^2})$, such that directions of the input with smaller variance are learned more slowly.

As an example of the potential impact of input correlations on learning, in Fig. S3 we consider learning two semantic classes consisting of wide versus tall rectangles. In this task, a subject is shown a rectangle with side lengths L_1 and L_2, and asked to report the degree to which the rectangle is wider than tall. That is, the subject must report $L_1 - L_2$, and this difference is the dimension in input feature space that governs the task-relevant semantic distinction.

We take the lengths to be drawn from a multivariate normal distribution

$$\begin{bmatrix} L_1 \\ L_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} v_1 & v_2 \\ v_2 & v_1 \end{bmatrix}\right), \qquad \textbf{[S29]}$$

and require the network to produce the difference $L_1 - L_2$, scaled such that it has unit variance. This scaling reflects the fact that a particular association might need to be expressed at a fixed level in behavior, regardless of the magnitude of the input variance on which it is based. We therefore take the desired output to be $y = c(L_1 - L_2)$ where the constant $c = \frac{1}{\sqrt{2(v_1 - v_2)}}$ scales the output to unit variance. Straightforward calculations yield

$$\begin{aligned} \mathbf{\Sigma}^{x} &= \begin{bmatrix} 1+v_1 & 1+v_2 \\ 1+v_2 & 1+v_1 \end{bmatrix} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \\ \mathbf{\Sigma}^{yx} &= \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{v_1 - v_2} & -\sqrt{v_1 - v_2} \end{bmatrix} = \mathbf{U}\mathbf{S}\mathbf{V}^T \end{aligned}$$

with

$$\mathbf{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \textbf{[S30]}$$

$$\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{v_1 - v_2} \end{bmatrix} \qquad \textbf{[S31]}$$

$$\mathbf{\Lambda} = \begin{bmatrix} 2+v_1+v_2 & 0 \\ 0 & v_1 - v_2 \end{bmatrix} \qquad \textbf{[S32]}$$

$$\mathbf{U} = \begin{bmatrix} 1 \end{bmatrix}. \qquad \textbf{[S33]}$$

Therefore the relevant input-output mode describing the difference between side lengths has singular value $s = v_1 - v_2$, and input variance $\lambda = \sqrt{v_1 - v_2}$. To separate out the effect of input correlation, we compute the correlation coefficient $\rho = \frac{\mathrm{Cov}(L_1, L_2)}{\mathrm{STD}(L_1)\mathrm{STD}(L_2)} = \frac{v_2}{v_1}$ and rewrite the singular value and task-relevant input variance as $s = v_1(1-\rho)$ and $\lambda = \sqrt{v_1(1-\rho)}$.

From Eqn.(S27), the learning timescale is thus

$$t \approx O\left(\frac{1}{s\lambda}\right) \qquad \textbf{[S34]}$$

$$= O\left(\frac{1}{v_1^{3/2}(1-\rho)^{3/2}}\right). \qquad \textbf{[S35]}$$

Hence for a fixed independent input variance v_1, the learning time diverges as the task-relevant correlation in the input decreases ($\rho \to 1$). If instead the task-relevant input correlation is increased ($\rho \to -1$), learning speed improves relative to independent white inputs ($\rho = 0$). This tradeoff is shown in Fig. S3. These results show that perceptual similarity structure can help or hinder learning speed, depending on whether there is high or low perceptual variance on task-relevant input dimensions respectively.

Progressive differentiation of hierarchical structure. In this section we introduce a hierarchical probabilistic generative model of items and their attributes that, when sampled, produces a dataset that can be supplied to our simple linear network. Using this, we will be able to explicitly link hierarchical taxonomies to the dynamics of learning in our network. We show that our network will exhibit progressive differentiation with respect to any of the underlying hierarchical taxonomies allowed by our generative model.

A key result from the explicit solutions is that the time scale of learning of each input-output mode α of the correlation matrix $\mathbf{\Sigma}^{yx}$ is inversely proportional to the correlation strength s_α (i.e. singular value) of the mode. It is on this time scale, $O(\tau/s_\alpha)$, that the network learns to project perceptual representations onto internal representations using the right singular vector $\mathbf{v}^\alpha$ of $\mathbf{\Sigma}^{yx}$, and then expand this component of the internal representation into a contribution to the predicted feature output vector given by $\mathbf{u}^\alpha$, a left singular vector of $\mathbf{\Sigma}^{yx}$.

To understand the time course of learning of hierarchical structure, we analyze a simple generative model proposed in [7] of hierarchical data $\{\mathbf{x}^\mu, \mathbf{y}^\mu\}$, and compute for this model the statistical properties $(s_\alpha, \mathbf{u}^\alpha, \mathbf{v}^\alpha)$ which drive learning.

Hierarchical feature vectors from a branching diffusion process

We first address the output data $\mathbf{y}^\mu$, $\mu = 1, \ldots, P$. Each $\mathbf{y}^\mu$ is an N_3-dimensional feature vector where each feature i in example μ takes the value $\mathbf{y}_i^\mu = \pm 1$. The value of each feature i across all examples arises from a branching diffusion process occurring on a tree, as depicted in Fig. S4. Each feature i undergoes its own diffusion process on the tree, *independent* of any other feature j. This entire process, described below, yields a hierarchical structure on the set of examples $\mu = 1, \ldots, P$, which are in one-to-one correspondence with the leaves of the tree.

The tree has a fixed topology, with D levels indexed by $l = 0, \ldots, D-1$, with M_l total nodes at level l. We take for simplicity a regular branching structure, so that every node at level l has exactly B_l descendants. Thus $M_l = M_0 \Pi_{k=0}^{l-1} B_l$. The tree has a single root node at the top ($M_0 = 1$), and again P leaves at the bottom, one per example in the dataset ($M_{D-1} = P$).

Given a single feature component i, its value across P examples is determined as follows. First draw a random variable $\eta^{(0)}$ associated with the root node at the top of the tree. The variable $\eta^{(0)}$ takes the values ± 1 with equal probability $\frac{1}{2}$. Next, for each of the B_0 descendants below the root node at level 1, pick a random variable $\eta_i^{(1)}$, for $i = 1, \ldots, B_0$. This variable $\eta_i^{(1)}$ takes the value $\eta^{(0)}$ with probability $1 - \epsilon$ and $-\eta^{(0)}$ with probability ϵ. The process continues down the tree: each of B_{l-1} nodes at level l with a common ancestor at level $l-1$ is assigned its ancestor's value with probability $1-\epsilon$, or is assigned the negative of its ancestor's value with probability ϵ. Thus the original feature value at the root, $\eta^{(0)}$, diffuses down the tree with a small probability ϵ of changing at each level along any path to a leaf. The final values at the P leaves constitute the feature values $\mathbf{y}_i^\mu$ for $\mu = 1, \ldots, P$. This process is repeated independently for all feature components i.

In order to understand the dimensions of variation in the feature vectors, we would like to first compute the inner product, or overlap, between two example feature vectors. This inner product, normalized by the number of features N_3, has a well-defined limit as $N_3 \to \infty$. Furthermore, due to the hierarchical diffusive process which generates the data, the normalized inner product only depends on the level of the tree at which the first common ancestor of the two leaves associated with the two examples arises. Therefore we can make the definition

$$q_k = \frac{1}{N_3} \sum_{i=1}^{N_3} \mathbf{y}_i^{\mu_1} \mathbf{y}_i^{\mu_2},$$

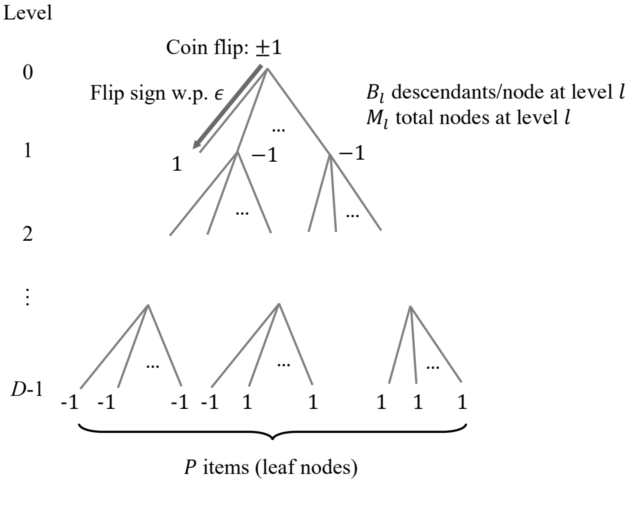
Level
0
1
2
⋮
D-1
Coin flip: ±1
Flip sign w.p. ϵ
B_l descendants/node at level l
M_l total nodes at level l
1
−1
−1
...
-1 -1 -1 -1 1 1 1 1 1
P items (leaf nodes)

Fig. S4. Generating hierarchically structured data through a branching diffusion process. To generate a feature, an initial binary value is determined through a coin flip at the top of the hierarchy. The sign of this value flips with a small probability along each link in the tree. At the bottom, this yields the value of one feature across items. Many features can be generated by repeatedly sampling from this process independently. The ± 1 values depicted are one possible sampling.

where again, the first common ancestor of leaves μ_1 and μ_2 arises at level k. It is the case that $1 = q_{D-1} > q_{D-2} > \cdots > q_0 > 0$. Thus pairs of examples with a more recent common ancestor have stronger overlap than pairs of examples with a more distant common ancestor. These $D-1$ numbers $q_0, \ldots, q_{D-2}$, along with the number of nodes at each level $M_0, \ldots, M_{D-1}$, are the fundamental parameters of the hierarchical structure of the feature vectors; they determine the correlation matrix across examples, i.e. the $P \times P$ matrix with elements

$$\mathbf{\Sigma}_{\mu_1 \mu_2} = \frac{1}{N_3} \sum_{i=1}^{N_3} \mathbf{y}_i^{\mu_1} \mathbf{y}_i^{\mu_2}, \qquad \textbf{[S36]}$$

and hence its eigenvectors and eigenvalues, which drive network learning, as we shall see below.

It is possible to explicitly compute q_k for the generative model described above. However, all that is really needed below is the property that $q_{D-2} > q_{D-1} > \cdots > q_0$. The explicit formula for q_k is

$$q_k = 1 - 2\Omega\left(D - 1 - k, 2\epsilon(1-\epsilon)\right),$$

where $\Omega(N, P)$ is the probability that a sum of N Bernoulli trials with probability P of being 1 yields an odd number of 1's. It is clear that the overlap q_k strictly decreases as the level k of the last common ancestor decreases (i.e. the distance up the tree to the last common ancestor increases).

Input-output correlations for orthogonal inputs and hierarchical outputs

We are interested in the singular values and vectors, $(s_\alpha, \mathbf{u}^\alpha, \mathbf{v}^\alpha)$ of $\mathbf{\Sigma}^{yx}$, since these drive the learning dynamics. We assume the P output feature vectors are generated hierarchically as in the previous section, but then assume a localist representation in the input, so that there are $N_1 = P$ input neurons and $\mathbf{x}_i^\mu = \delta_{\mu i}$. The input-output correlation matrix $\mathbf{\Sigma}^{yx}$ is then an $N_3 \times P$ matrix with elements $\mathbf{\Sigma}_{i\mu}^{yx} = \mathbf{y}_i^\mu$, with $i = 1, \ldots, N_3$ indexing feature components, and $\mu = 1, \ldots, P$ indexing examples. We note that

$$(\mathbf{\Sigma}^{yx})^T \mathbf{\Sigma}^{yx} = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T = N_3 \mathbf{\Sigma},$$

where $\mathbf{\Sigma}$, defined in (S36), is the correlation matrix across examples. From this we see that the eigenvectors of $\mathbf{\Sigma}$ are the same as the right singular vectors $\mathbf{v}^\alpha$ of $\mathbf{\Sigma}^{yx}$, and if the associated eigenvalue of $\mathbf{\Sigma}$ is λ_α, then the associated singular value of $\mathbf{\Sigma}^{yx}$ is $s_\alpha = \sqrt{N_3 \lambda_\alpha}$. Thus finding the singular values s_α of $\mathbf{\Sigma}^{yx}$, which determine the time scales of learning, reduces to finding the eigenvalues λ_α of $\mathbf{\Sigma}$.

We note that the localist assumption $\mathbf{x}_i^\mu = \delta_{\mu i}$ is not necessary. We could have instead assumed an orthogonal distributed representation in which the vectors $\mathbf{x}^\mu$ form an orthonormal basis $\mathbf{O}$ for the space of input-layer activity patterns. This would yield the modification $\mathbf{\Sigma}^{yx} \to \mathbf{\Sigma}^{yx}\mathbf{O}^T$, which would not change the singular values s_α at all, but would simply rotate the singular vectors, $\mathbf{v}^\alpha$. Thus distributed orthogonal perceptual representations and localist representations yield exactly the same time course of learning. For simplicity, we focus here on localist input representations.

We now find the eigenvalues λ_α and eigenvectors $\mathbf{v}^\alpha$ of the correlation matrix across examples, $\mathbf{\Sigma}$ in (S36). This matrix has a hierarchical block structure, with diagonal elements $q_{D-1} = 1$ embedded within blocks of elements of magnitude q_{D-2} in turn embedded in blocks of magnitude q_{D-3} and so on down to the outer-most blocks of magnitude $q_0 > 0$. This hierarchical block structure in turn endows the eigenvectors with a hierarchical structure.

To describe these eigenvectors we must first make some preliminary definitions. We can think of each P dimensional eigenvector as a function on the P leaves of the tree which generated the feature vectors $\mathbf{y}^\mu$, for $\mu = 1, \ldots, P$. Many of these eigenvectors will take constant values across subsets of leaves in a manner that respects the topology of the tree. To describe this phenomenon, let us define the notion of a level l funtion $f(\mu)$ on the leaves as follows: first consider a function g which takes M_l values on the M_l nodes at level l of the tree. Each leaf μ of the tree at level $D-1$ has a unique ancestor $\nu(\mu)$ at level l; let the corresponding level l function on the leaves induced by g be $f(\mu) = g(\nu(\mu))$. This function is constant across all subsets of leaves which have the same ancestor at level l. Thus any level l function cannot discriminate between examples that have a common ancestor which lives at any level $l' > l$ (i.e. any level lower than l).

Now every eigenvector of $\mathbf{\Sigma}$ is a level l function on the leaves of the tree for some l. Each level l yields a degeneracy of eigenvectors, but the eigenvalue of any eigenvector depends only on its level l. The eigenvalue λ_l associated with every level l eigenvector is

$$\lambda_l \equiv P\left(\sum_{k=l}^{D-1} \frac{\Delta_l}{M_l}\right),$$

where $\Delta_l \equiv q_l - q_{l-1}$, with the caveat that $q_{-1} \equiv 0$. It is clear that λ_l is a decreasing function of l. This immediately implies that finer scale distinctions among examples, which can only be made by level l eigenvectors for larger l, will be learned later than coarse-grained distinctions among examples, which can be made by level l eigenvectors with smaller l.

We now describe the level l eigenvectors. They come in M_{l-1} families, one family for each node at the higher level $l-1$ ($l=0$ is a special case–there is only one eigenvector at this level and it is a uniform mode that takes a constant value on all P leaves). The family of level l eigenvectors associated with a node ν at level $l-1$ takes nonzero values only on leaves which are descendants of ν. They are induced by functions on the B_{l-1} direct descendants of ν which sum to 0. There can only be $B_{l-1}-1$ such orthonormal eigenvectors, hence the degeneracy of all level l eigenvectors is $M_{l-1}(B_{l-1}-1)$. Together, linear combinations of all these level l eigenvectors can be used to assign different values to any two examples whose first common ancestor arises at level l but not at any lower level $l' > l$. Thus level l eigenvectors do not see any structure in the data at any level of granularity below level l of the hierarchical tree which generated the data. Recall that these eigenvectors are precisely the input modes which project examples onto internal representations in the multilayer network. Importantly, this automatically implies that structure below level l in the tree cannot arise in the internal representations of the network until after structure at level $l-1$ is learned.

We can now be quantitative about the time scale at which structure at level l is learned. We first assume the branching factors B_l are relatively large, so that to leading order, $\lambda_l = P\frac{\delta_l}{M_l}$. Then the singular values of $\mathbf{\Sigma}^{yx}$ at level l are

$$s_l = \sqrt{N\lambda_l} = \sqrt{NP\frac{\Delta_l}{M_l}}.$$

The time scale of learning structure at level l is then

$$\tau_l = \frac{\tau}{s_l} = \frac{1}{\lambda}\sqrt{\frac{P}{N}\frac{M_l}{\Delta_l}},$$

where we have used the definition of τ in (S5). The fastest time scale is τ_0 since $M_0 = 1$ and the requirement that $\tau_0 \gg 1$ yields the requirement $\lambda \ll \sqrt{P/N}$. If we simply choose $\lambda = \epsilon\sqrt{P/N}$ with $\epsilon \ll 1$, we obtain the final result

$$\tau_l = \frac{1}{\epsilon}\sqrt{\frac{M_l}{\Delta_l}}.$$

Thus the time scale for learning structure at a level of granularity l down the tree, for this choice of learning rate and generative model, is simply proportional to the square root of the number of ancestors at level l. For constant branching factor B, this time scale grows exponentially with l.

Illusory correlations. The dynamics of learning in deep but not shallow networks can cause them to exhibit illusory correlations during learning, where the prediction for a particular feature can be a U-shaped function of time. This phenomenon arises from the strong dependence of the learning dynamics on the singular value size, and the sigmoidal stage-like transitions in the deep network. In particular, a feature m for item i receives a contribution from each mode α of $a_\alpha(t)\mathbf{u}_m^\alpha\mathbf{v}_i^\alpha$. Looking at two successive modes k and $k+1$, these will cause the network's estimate of the feature to increase and decrease respectively if $\mathbf{u}_m^k\mathbf{v}_i^k > 0$ and $\mathbf{u}_m^{k+1}\mathbf{v}_i^{k+1} < 0$ (flipping these inequalities yields a symmetric situation where the feature will first decrease and then increase). The duration of the illusory correlation can be estimated by contrasting the time at which the first mode is learned compared to the second. In particular, suppose the second mode's singular value is smaller by an amount Δ, that is, $s_{k+1} = s_k - \Delta$. Then the illusory correlation persists for a time

$$\begin{aligned} t_{k+1} - t_k &= \frac{\tau}{s_k - \Delta}\ln\frac{s_k-\Delta}{\epsilon} - \frac{\tau}{s_k}\ln\frac{s_k}{\epsilon} \\ &= \frac{\tau s_k \ln\frac{s_k-\Delta}{s_k} + \tau\Delta\ln\frac{s_k}{\epsilon}}{s_k(s_k-\Delta)} \\ &\approx \frac{\tau\Delta\ln\frac{s_k}{\epsilon}}{s_k^2} \end{aligned}$$

in the regime where $\epsilon \ll \Delta \ll s_k$, or approximately a period of length $O(\Delta)$. While illusory correlations can cause the error on one specific feature to increase, we note that the total error across all features and items always decreases or remains constant (as is the case for any gradient descent procedure).

In contrast, the shallow network exhibits no illusory correlations. The prediction for feature m on item i is

$$\begin{aligned} \hat{\mathbf{y}}_m^i &= \sum_\alpha b_\alpha(t)\mathbf{u}_m^\alpha\mathbf{v}_i^\alpha \\ &= \sum_\alpha \left[s_\alpha\left(1-e^{-t/\tau}\right) + b_\alpha^0 e^{-t/\tau}\right]\mathbf{u}_m^\alpha\mathbf{v}_i^\alpha \\ &= \left(1-e^{-t/\tau}\right)\underbrace{\left[\sum_\alpha s_\alpha\mathbf{u}_m^\alpha\mathbf{v}_i^\alpha\right]}_{c_1} + e^{-t/\tau}\underbrace{\sum_\alpha b_\alpha\mathbf{u}_m^\alpha\mathbf{v}_i^\alpha}_{c_2} \\ &= c_1 - (c_1 - c_2)e^{-t/\tau} \end{aligned}$$

which is clearly monotonic in t. Therefore shallow networks never yield illusory correlations where the sign of the progress on a particular feature changes over the course of learning.

Organizing and Encoding Knowledge

Category membership, typicality, prototypes. The singular value decomposition satisfies a set of mutual constraints that provide consistent relationships between category membership and item and feature typicality. In particular, form the matrix $\mathbf{O} = [\mathbf{y}^1 \cdots \mathbf{y}^{N_1}]$ consisting of the features of each object in its columns. We assume that the input here directly codes for object identity using a one-hot input vector ($X = I$). Then the input-output correlation matrix which drives learning is $\mathbf{\Sigma}^{yx} = E[\mathbf{y}\mathbf{x}^T] = \frac{1}{P}\mathbf{O}$. The dynamics of learning are thus driven by the singular value decomposition of $\mathbf{O}$,

$$\frac{1}{P}\mathbf{O} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \qquad \textbf{[S37]}$$

where the matrices of left and right singular vectors are orthogonal ($\mathbf{U}^T\mathbf{U} = I$ and $\mathbf{V}^T\mathbf{V} = I$). Because of this orthogonality, multiplying both sides by $\mathbf{S}^{-1}\mathbf{U}^T$ from the left we have,

$$\begin{aligned} \frac{1}{P}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{O} &= \mathbf{S}^{-1}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T, \\ \frac{1}{P}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{O} &= \mathbf{V}^T \end{aligned}$$

Pulling out the element at the ith row and the αth column of $\mathbf{V}$ on both sides, we obtain Eqn. (13) of the main text,

$$\mathbf{v}_i^\alpha = \frac{1}{P s_\alpha} \sum_{m=1}^{N_3} \mathbf{u}_m^\alpha \mathbf{o}_m^i.$$

Similarly, multiplying Eqn. (S37) from the right by $\mathbf{V}\mathbf{S}^{-1}$ yields,

$$\begin{aligned} \frac{1}{P}\mathbf{O}\mathbf{V}\mathbf{S}^{-1} &= \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^{-1}, \\ \frac{1}{P}\mathbf{O}\mathbf{U}\mathbf{S}^{-1} &= \mathbf{U}. \end{aligned}$$

Extracting the elements at the ith row and αth column yields Eqn. (14) of the main text,

$$\mathbf{u}_m^\alpha = \frac{1}{P s_\alpha} \sum_{i=1}^{N_1} \mathbf{v}_i^\alpha \mathbf{o}_m^i.$$

Category coherence. Real world categories may be composed of a small number of items and features amid a large background of many items and possible features which do not possess category structure. Here we consider the task of identifying disjoint categories in the presence of such noise. We show that a single category coherence quantity determines the speed and accuracy of category recovery by a deep linear neural network, and compute the threshold category coherence at which deep linear networks begin to correctly recover category structure.

We consider a dataset of N_o objects and N_f features, in which a category of K_o objects and K_f features is embedded. That is, a subset C_f of $K_f = |C_f|$ features occur with high probability p for the subset C_i of $K_o = |C_i|$ items in the category. Background features (for which either the feature or item are not part of the category) occur with a lower probability q. Define the random matrix $\mathbf{R}$ of size $N_f \times N_o$ to have entries $\mathbf{R}_{ij} = 1$ with probability p and 0 with probability $1-p$ provided $i \in C_f$ and $j \in C_i$, and $\mathbf{R}_{ij} = 1$ with probability q and 0 with probability $1-q$ otherwise. A realization of this matrix yields one environment containing items and features with a category embedded into it. To access general properties of this setting, we study the behavior in the high-dimensional limit where the number of features and items is large, $N_f, N_o \to \infty$, but their ratio is constant, $N_o/N_f \to c \in (0, 1]$.

We suppose that the features are recentered and rescaled such that background features have zero mean and variance $1/N_f$ before being passed to the network. That is, we define the normalized, rescaled feature matrix

$$\tilde{\mathbf{R}} = \frac{1}{\sqrt{N_f q(1-q)}}(\mathbf{R} - q\mathbf{1}\mathbf{1}^{\mathbf{T}}) \qquad \textbf{[S38]}$$

where we have used the fact that $E[y_i] = q$ and $Var[y_i] = q(1-q)$ for a background feature i to derive the appropriate rescaling. With this rescaling we can approximately rewrite $\tilde{\mathbf{R}}$ as a random matrix perturbed by a low rank matrix corresponding to the embedded category,

$$\tilde{\mathbf{R}} \approx \mathbf{X} + \mathbf{P}. \qquad \textbf{[S39]}$$

Here each element of the noise matrix $\mathbf{X}$ is independent and identically distributed as $\mathbf{X}_{ij} = \frac{1}{\sqrt{N_f q(1-q)}}(x-q)$ where x is a Bernoulli random variable with probability q. The signal matrix $\mathbf{P}$ containing category information is low rank and given by

$$\mathbf{P} = \theta \frac{1}{\sqrt{K_f K_o}} \mathbf{1}_{C_f} \mathbf{1}_{C_i}^T \qquad \textbf{[S40]}$$

where $\mathbf{1}_C$ is a vector with ones on indices in the set C and zeros everywhere else, and θ is the associated singular value of the low rank category structure. In particular, elements of $\mathbf{R}$ for items and features within the category have a mean value of p. Hence using this and applying the mean shift and rescaling as before, we have

$$\theta = \frac{(p-q)\sqrt{K_f K_o}}{\sqrt{N_f q(1-q)}}. \qquad \textbf{[S41]}$$

To understand learning dynamics in this setting, we must compute the typical singular values and vectors of $\tilde{\mathbf{R}}$. Theorem 2.9 of [8] states that recovery of the correct singular vector structure only occurs for signal strengths above a threshold (an instance of the BBP phase transition [9]). In particular, let $\tilde{\mathbf{u}}, \tilde{\mathbf{v}}$ be the feature and object analyzer vectors of $\tilde{\mathbf{R}}$ respectively (left and right singular vectors respectively), and let

$$\mathbf{u}^{\text{Ideal}} = \frac{1}{\sqrt{K_f}} \mathbf{1}_{C_f}, \qquad \textbf{[S42]}$$

$$\mathbf{v}^{\text{Ideal}} = \frac{1}{\sqrt{K_o}} \mathbf{1}_{C_o} \qquad \textbf{[S43]}$$

be the ground truth feature and object analyzers arising from the category structure in Eqn. (S40). Then

$$\left(\tilde{\mathbf{u}}^T \mathbf{u}^{\text{Ideal}}\right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{c+\theta^2}{\theta^2(\theta^2+1)} & \text{for } \theta > c^{1/4} \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{[S44]}$$

$$\left(\tilde{\mathbf{v}}^T \mathbf{v}^{\text{Ideal}}\right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{c(1+\theta^2)}{\theta^2(\theta^2+c)} & \text{for } \theta > c^{1/4} \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{[S45]}$$

where $a.s.$ denotes almost sure convergence (i.e., with probability 1) in the high-dimensional limit ($N_f, N_o \to \infty$ and $N_o/N_f = c$).

In essence, for $\theta \leq c^{1/4}$, the learned feature and object analyzer vectors will have no overlap with the correct category structure. For $\theta > c^{1/4}$, the feature and object analyzer vectors will have positive dot product with the true category structure yielding at least partial recovery of the category. Using the definitions of θ and c and straightforward algebra, the recovery condition $\theta > c^{1/4}$ can be written as

$$\frac{(p-q)^2 K_f K_o}{q(1-q)\sqrt{N_f N_o}} > 1. \qquad \textbf{[S46]}$$

This motivates defining category coherence as

$$\mathcal{C} \equiv \frac{(p-q)^2 K_f K_o}{q(1-q)\sqrt{N_f N_o}} \qquad \textbf{[S47]}$$

$$= \text{SNR} \frac{K_f K_o}{\sqrt{N_f N_o}} \qquad \textbf{[S48]}$$

where we have defined the signal-to-noise ratio $\text{SNR} = \frac{(p-q)^2}{q(1-q)}$.

So defined, for a fixed item/feature ratio c, the category coherence $\mathcal{C}$ completely determines the performance of category recovery. To see this, we note that $\theta^2 = c^{1/2}\mathcal{C}$ such that Eqns. (S44)-(S45) can be written as

$$\left(\tilde{\mathbf{u}}^T \mathbf{u}^{\text{Ideal}}\right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{1+c^{-1/2}\mathcal{C}}{\mathcal{C}(\mathcal{C}+c^{-1/2})} & \text{for } \mathcal{C} > 1 \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{[S49]}$$

$$\left(\tilde{\mathbf{v}}^T \mathbf{v}^{\text{Ideal}}\right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{1+c^{1/2}\mathcal{C}}{\mathcal{C}(\mathcal{C}+c^{1/2})} & \text{for } \mathcal{C} > 1 \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{[S50]}$$

Hence recovery of category structure can be described by a single category coherence quantity that is sensitive to both the signal-to-noise ratio of individual features in the category relative to background feature variability, weighted by the size of the category. Finally, we reiterate the regime of validity for the analysis presented here: the theory applies in the limit where N_f and N_o are large, the ratio $c = N_o/N_f \in (0, 1]$ is finite (implying $N_f > N_o$), and the category size is on the order of the square root of the total number of items and features, $K_f K_o \sim \sqrt{N_f N_o}$.

Basic categories. To generalize the notion of category coherence further, we propose to simply define category coherence as the singular value associated with a categorical distinction in the SVD of the input-output correlations $\mathbf{\Sigma}^{yx}$. In this section we show that this definition can give rise to a basic level advantage depending on the similarity structure of the categories, and gives rise to an intuitive notion of category coherence based on within-category similarity and between-category difference. We additionally show that this definition makes category coherence dependent on the global structure of the dataset, through a well-known optimality condition.

Hierarchical singular values from item similarities. The hierarchical generative model considered previously has a simple structure of independent diffusion down the hierarchy. This results in singular values that are always a decreasing function of the hierarchy level. Here we show how more complex (but still hierarchical) similarity structures between items can give rise to a basic level advantage; and that defining category coherence as the associated singular value for a categorical distinction recovers intuitive notions of category coherence.

Suppose we have a set of items with input-output correlation matrix $\mathbf{\Sigma}^{yx}$. The singular values are the square root of the eigenvalues of the item similarity matrix,

$$\mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx} \equiv \mathbf{\Sigma}^{y}, \qquad \textbf{[S51]}$$

and the object analyzer vectors $\mathbf{v}^\alpha, \alpha = 1, \cdots, P$ are the eigenvectors of $\mathbf{\Sigma}^y$. We assume that the object analyzer vectors exactly mirror the hierarchical structure of the items, and for simplicity focus on the case of a regularly branching tree.

By assumption, the item similarity matrix has decomposition

$$\mathbf{\Sigma}^y = \sum_{\alpha=1}^{P} \lambda_\alpha \mathbf{v}^\alpha \mathbf{v}^{\alpha T}. \qquad \textbf{[S52]}$$

As described previously, eigenvectors come in groups corresponding to each hierarchical level k.

In this setting, the similarity matrix will have a hierarchical block structure (as can be seen in Fig. 8 of the main text). Each block corresponds to a subset of items, and blocks are either disjoint (containing different items) or nested (one block containing a subset of the items of the other). The blocks are in one to one correspondence with a rooted regularly branching tree, with leaves corresponding to each item and one block per internal node. Each block corresponding to a node of the tree at level k has constant entries of

$$q_k = \frac{1}{N_3}\sum_{i=1}^{N_3} \mathbf{y}_i^{\mu_1}\mathbf{y}_i^{\mu_2}, \qquad \textbf{[S53]}$$

the similarity between any two items μ_1, μ_2 with closest common ancestor at level k.

The eigenvalue associated with a category C at level k in the hierarchy can be written as

$$\lambda_k = \sum_{j\in C} \mathbf{\Sigma}^y_{ij} - \sum_{j\in S(C)} \mathbf{\Sigma}^y_{ij} \quad \text{for any } i \in C \qquad \textbf{[S54]}$$

where $S(C)$ is any sibling category of C in the tree (i.e. another category at the same hierarchical level). That is, take any member i of category C, and compute the sum of its similarity to all members of category C (including itself); then subtract the similarity between member i and all members of one sibling category $S(C)$. Hence this may directly be interpreted as the total within category similarity minus the between category difference.

A basic level advantage can thus occur if between category similarity is negative, such that items in different categories have anticorrelated features. This will cause the second term of Eqn. (S54) to be positive, boosting category coherence at that level of the hierarchy. The category coherence of superordinate categories will decrease (because within category similarity will decrease), and subordinate categories will be unaffected. If the anticorrelation is strong enough, an intermediate level can have higher category coherence, and be learned faster, than a superordinate level.

Global optimality properties of the SVD. Our proposal to *define* the category coherence $\mathcal{C}$ as the associated singular value for a particular object-analyzer vector makes category coherence fundamentally dependent on the interrelations between all items and their properties. To see this, we observe that the singular value decomposition obeys a well-known *global* optimality condition: if we restrict our representation of the environment to just k linear relations, then the first k modes of the SVD yield the lowest total prediction error of all linear predictors. In particular, suppose the network retains only the top k modes of the singular value decomposition, as would occur if training is terminated early before all modes have risen to their asymptote. The network predicts features $\tilde{\mathbf{O}} = \mathbf{U}\tilde{\mathbf{S}}\mathbf{V}^T$, where $\tilde{\mathbf{S}}$ contains just the first k singular values with the remaining diagonal elements set to zero (that is, $\tilde{\mathbf{S}}_{ii} = \mathbf{S}_{ii}$ for $i \leq k$ and $\tilde{\mathbf{S}}_{ii} = 0$ otherwise). The Eckart–Young–Mirsky theorem states that $\tilde{\mathbf{O}}$ is a solution to

$$\min_{\mathbf{B},\mathrm{rank}(\mathbf{B})\leq k} \|\mathbf{O} - \mathbf{B}\|_F\,. \qquad \textbf{[S55]}$$

Hence out of all rank k representations of the environment, the truncated SVD yields the minimum total error.

In the terminology of deep linear neural networks, out of all networks with $N_2 = k$ or fewer hidden neurons, networks with total weights $\tilde{\mathbf{W}}^2\tilde{\mathbf{W}}^1 = \tilde{\mathbf{O}} = \mathbf{U}\tilde{\mathbf{S}}\mathbf{V}^T$ are minimizers of the total sum squared error,

$$\min_{\mathbf{W}^1,\mathbf{W}^2,N_2\leq k} SSE(\mathbf{W}^1, \mathbf{W}^2).$$

We include a proof of this fact for completeness. First, note that

$$SSE(\tilde{\mathbf{W}}^1, \tilde{\mathbf{W}}^2) = \frac{1}{2}||\mathbf{U}(\mathbf{S} - \tilde{\mathbf{S}})\mathbf{V}||_F^2 = \frac{1}{2}\sum_{i=k+1}^{N_1} s_i(\mathbf{O})^2. \qquad \textbf{[S56]}$$

where here and in the following we will denote the ith largest singular value of the matrix A as $s_i(A)$. For two matrices $\mathbf{C} \in R^{N_3\times N_1}$ and $\mathbf{D} \in R^{N_3\times N_1}$, Weyl's theorem for singular values states that

$$s_{i+j-1}(\mathbf{C} + \mathbf{D}) \leq s_i(\mathbf{C}) + s_j(\mathbf{D})$$

for $1 \leq i, j \leq N_1$ and $i + j - 1 \leq N_1$. Taking $j = k + 1$, $\mathbf{C} = \mathbf{O} - \mathbf{W}^2\mathbf{W}^1$, and $\mathbf{D} = \mathbf{W}^2\mathbf{W}^1$ yields

$$\begin{aligned} s_{i+k}(\mathbf{O}) &\leq s_i(\mathbf{O} - \mathbf{W}^2\mathbf{W}^1) + s_{k+1}(\mathbf{W}^2\mathbf{W}^1) && \textbf{[S57]} \\ &\leq s_i(\mathbf{O} - \mathbf{W}^2\mathbf{W}^1) && \textbf{[S58]} \end{aligned}$$

for $1 \leq i \leq N_1 - k$. In the last step we have used the fact that $s_{k+1}(\mathbf{W}^2\mathbf{W}^1) = 0$ since $\mathbf{W}^2\mathbf{W}^1$ has rank at most k. We therefore have

$$\begin{aligned} \frac{1}{2}||\mathbf{O} - \mathbf{W}^2\mathbf{W}^1||_F^2 &= \frac{1}{2}\sum_{i=1}^{N_1} s_i(\mathbf{O} - \mathbf{W}^2\mathbf{W}^1)^2 \\ &\geq \frac{1}{2}\sum_{i=1}^{N_1-k} s_i(\mathbf{O} - \mathbf{W}^2\mathbf{W}^1)^2 && \textbf{[S59]} \\ &\geq \frac{1}{2}\sum_{i=k+1}^{N_1} s_i(\mathbf{O})^2 && \textbf{[S60]} \\ &= \frac{1}{2}||\mathbf{O} - \tilde{\mathbf{W}}^2\tilde{\mathbf{W}}^1||_F^2, \end{aligned}$$

where from (S59)-(S60) we have used (S57)-(S58) and the last equality follows from Eqn. (S56). Hence

$$SSE(\tilde{\mathbf{W}}^1, \tilde{\mathbf{W}}^2) \leq SSE(\mathbf{W}^1, \mathbf{W}^2) \qquad \textbf{[S61]}$$

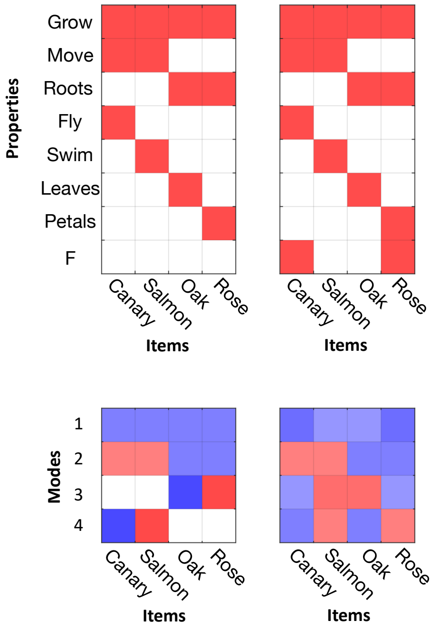

Fig. S5. Category structure is a nonlocal and nonlinear function of the features. Left column: a toy dataset with hierarchical structure (top) has object analyzer vectors that mirror the hierarchy (bottom). Right column: Adding a new feature F to the dataset (top) causes a substantial change to the category structure (bottom). In particular the features of the *Salmon* are identical in both datasets, yet the categorical groupings the *Salmon* participates in have changed, reflecting the fact that the SVD is sensitive to the global structure of the dataset.

as required.

As a simple example of how local changes to the features of a few items can cause global reorganization of categorical structure in the SVD, we consider the hierarchical dataset from Fig. 3 in the main text, but add a single additional feature. If this feature is not possessed by any of the items, then the categorical decomposition reflects the hierarchical structure of the dataset as usual. However if this feature is possessed by both the *Canary* and *Rose* (perhaps a feature like *Brightly colored*), the resulting categorical structure changes substantially, as shown in Fig. S5. While the highest two levels of the hierarchy remain similar, the lowest two levels have been reconfigured to group the *Canary* and *Rose* in one category and the *Salmon* and *Oak* in another. Consider, for instance, the *Salmon*: even though its own feature vector has not changed, its assignment to categories has. In the original hierarchy, it was assigned to a *bird-fish* distinction, and did not participate in a *tree-flower* distinction. With the additional feature, it now participates in both a *bright-dull* distinction and another distinction encoding the differences between the *Canary/Oak* and *Salmon/Rose*. Hence the mapping between features and categorical structure implied by the SVD can be non-local and nonlinear: small perturbations of the features of items can sometimes result in large changes to the singular vectors. This specific example is not intended to be an actual description of the property correlations for these items. Rather, we use it narrowly to demonstrate the point that the categorical structure arising from the SVD is a global property of all items and their features, and the categorical structure applied to one specific item can be altered by the features of other items.

Discovering and representing explicit structures. To investigate how datasets with certain underlying structural forms come to be represented in the neural network, we consider drawing datasets from probabilistic graphical models specified by graphs over items (and possibly hidden variables). To go from a graph to feature values for each item, we follow [10] and use a Gaussian Markov random field. Intuitively, this construction causes items which are nearby in the graph to have more similar features.

In particular, consider a graph consisting of a set of nodes $\mathcal{V}$ of size $K = |\mathcal{V}|$, connected by a set of undirected edges $\mathcal{E}$ with lengths $\{e_{ij}\}$, where e_{ij} is the length of the edge between node i and node j. Each item in the environment is associated with one node in the graph, but there can be more nodes than items. For instance, a tree structure has nodes for each branching point of the tree, but items are associated only with the leaves (in Fig. 9 of the main text, nodes associated with items are depicted as filled circles, while unassociated nodes lie at edge intersections). We construct the $K \times K$ weighted adjacency matrix $\mathbf{A}$ where $\mathbf{A}_{ij} = 1/e_{ij}$ and $\mathbf{A}_{ij} = 0$ if there is no edge between nodes i and j. Next, we form the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal weighted degree matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{K} \mathbf{A}_{ij}$. We take the value of a particular feature m across nodes in the graph to be distributed as

$$\tilde{\mathbf{f}} \sim \mathcal{N}\left(0, \tilde{\mathbf{\Phi}}^{-1}\right)$$

where $\tilde{\mathbf{f}}$ is a length K vector of feature values for each node, and $\tilde{\mathbf{\Phi}} = \mathbf{L} + 1/\sigma^2 \mathbf{I}$ is the precision matrix (inverse covariance matrix) of the Gaussian distribution. Here the parameter σ^2 instantiates graph-independent variation in the feature values which ensures the inverse exists. Finally, to obtain a length P vector $\mathbf{f}$ of feature values across items (rather than across all nodes in the graph) we take the subset of the vector $\tilde{\mathbf{f}}$ corresponding to nodes with associated items. This can be written as $\mathbf{f} = \mathbf{M}\tilde{\mathbf{f}}$ for an appropriate matrix $\mathbf{M} \in \mathbb{R}^{P \times K}$ which has $\mathbf{M}_{ij} = 1$ if item i is associated with node j and is zero otherwise. This is a linear transformation of a Gaussian, and hence $\mathbf{f}$ is Gaussian zero mean with covariance $\mathbf{\Phi}^{-1} = \mathbf{M}\tilde{\mathbf{\Phi}}^{-1}\mathbf{M}^T$,

$$\mathbf{f} \sim \mathcal{N}\left(0, \mathbf{\Phi}^{-1}\right). \qquad \mathbf{[S62]}$$

To obtain multiple features, we assume that features are drawn independently according to Eq. (S62).

This approach describes a generation process for a dataset: A set of N_3 features are drawn, yielding the dataset $\{\mathbf{x}^i, \mathbf{y}^i\}, i = 1, \ldots, P$ where for simplicity we assign one-hot input vectors $\mathbf{x}^i$ to each item such that $\mathbf{X} = [\mathbf{x}^1 \cdots \mathbf{x}^P] = \mathbf{I}$. This dataset is then presented to the neural network for training, and the dynamics are driven by the SVD of $\mathbf{\Sigma}^{yx} = \frac{1}{P}\sum_i^P \mathbf{y}^i \mathbf{x}^{iT} = \frac{1}{P}\mathbf{Y}\mathbf{X}^{\mathbf{T}}$ where $\mathbf{Y} = [\mathbf{y}^1 \cdots \mathbf{y}^P]$ is the concatenated matrix of features. From the definition of the SVD, the object analyzer vectors are the eigenvectors of the matrix

$$\begin{aligned} \mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx} &= \frac{1}{P^2}\mathbf{X}\mathbf{Y}^T\mathbf{Y}\mathbf{X} \\ &= \frac{1}{P^2}\mathbf{Y}^T\mathbf{Y} \equiv \mathbf{\Sigma}^y. \end{aligned}$$

Now we note that

$$\begin{aligned} \mathbf{\Sigma}^y_{ij} &= \frac{1}{P^2}\mathbf{y}^{iT}\mathbf{y}^j \\ &= \frac{1}{P^2}\sum_{m=1}^{N_3}\mathbf{y}^i_m\mathbf{y}^j_m \\ &= \frac{N_3}{P^2}\left(\frac{1}{N_3}\sum_{m=1}^{N_3}\mathbf{y}^i_m\mathbf{y}^j_m\right). \end{aligned}$$

As the number of features grows ($N_3 \to \infty$), this sample average converges to

$$\mathbf{\Sigma}^y_{ij} = \frac{N_3}{P^2}E\left[\mathbf{f}^i\mathbf{f}^j\right]$$

and hence from Eq. (S62),

$$\mathbf{\Sigma}^y = \frac{N_3}{P^2}\mathbf{\Phi}^{-1}.$$

Up to a scalar, the item covariance matrix is simply the covariance structure arising from the graph; and because matrix inversion preserves eigenvectors, the eigenvectors of the matrix $\mathbf{\Phi}$ are the object analyzer vectors. Finally, the singular values are $s_\alpha = \frac{\sqrt{N_3}}{P\sqrt{\zeta_\alpha}}$, where ζ_α is the α'th eigenvalue of $\mathbf{\Phi}$.

We now describe how the specific graph types considered in the main text result in structured matrices for which the SVD may be calculated analytically.

Clusters Here we consider partitioning P items into a set of $N_c \leq P$ clusters, yielding a graph in which each item in a cluster b is connected by a constant length edge e_b to a hidden cluster identity node. Let M_b be the number of items in cluster b. It is easy to see that the resulting item correlation structure is block diagonal with one block per cluster; and each block has the form $\mathbf{\Phi}_b^{-1} = c_1^b\mathbf{1} + c_2^b\mathbf{I}$ where $\mathbf{1} \in \mathbb{R}^{M_b \times M_b}$ is a constant matrix of ones, $\mathbf{I}$ is an identity matrix, $b = 1, \cdots, N_c$ is the block index, and c_1^b, c_2^b are scalar constants

$$\begin{aligned} c_1^b &= \frac{\sigma^2}{M_b+1} + \frac{M_b-1}{(1/e_b+1/\sigma^2)M_b} \\ &\quad + \frac{1}{((M_b+1)/e_b+1/\sigma^2)M_b(M_b+1)} \\ c_2^b &= \frac{\sigma^2}{M_b+1} - \frac{1}{M_b(1/e_b+1/\sigma^2)} \\ &\quad + \frac{1}{((M_b+1)/e_b+1/\sigma^2)M_b(M_b+1)} \end{aligned}$$

To understand learning dynamics in this setting, we must compute the eigenvalues and eigenvectors of this correlation structure. The eigenvalues and eigenvectors of a block diagonal matrix are simply the concatenated eigenvalues and eigenvectors of each of the blocks (where the eigenvectors from a block are padded with zeros outside of that block). Looking at one block b, the constant vector $\mathbf{v} = 1/\sqrt{M_b}\mathbf{1}$ is an object analyzer vector with eigenvalue

$$s_1 = \frac{1 + M_b\sigma^2/e_b}{(M_b+1)/e_b + 1/\sigma^2}.$$

The remaining $M_b - 1$ eigenvalues are all equal to

$$s_2 = \frac{1}{1/e_b + 1/\sigma^2}.$$

From these results we can draw several conclusions about the speed of learning simple category structure. First, we note that the shared structure in a category, encoded by the constant eigenvector, is always more prominent (and hence will be learned faster) than the item-specific information. That is, s_1 is always larger than s_2 in the relevant regime $M_b \geq 2$, $e_b > 0$, and $\sigma > 0$. To see this, we differentiate the difference $s_1 - s_2$ with respect to M_b and e_b and set the result to zero to find extremal points. This yields $M_b = 0$, and $1/e_b = 0$ or $1/e_b = -2/(M_b\sigma^2 + 2\sigma^2)$. Hence there are no critical points in the relevant region, and we therefore test the boundary of the constraints. For $e_b \to 0$, we have $s_1 - s_2 = \frac{\sigma^2}{1+1/M_b}$ which is increasing in M_b. For $M_b = 2$, we have $s_1 - s_2 = \frac{2\sigma^6}{3\sigma^4+4\sigma^2e_b+e_b^2}$ which is decreasing in e_b. The minimum along the boundary would thus occur at $M_b = 2, e_b \to \infty$, where the difference converges to zero but is positive at any finite value. Testing a point in the interior yields a higher value (for instance $M_b = 3$ and $e_b = 1$ yields $s_1 - s_2 = \frac{3\sigma^6}{4\sigma^4+5\sigma^2+1} \geq 0$), confirming that this is the global minimum and $s_1 > s_2$ in this domain. Hence categorical structure will typically be learned faster than idiosyncratic item-specific information.

We note that the graph we have constructed is only one way of creating categorical structure, which leaves different clusters independent. In particular, it establishes a scenario in which features of members in each category are positively correlated, but features of members of different categories are simply not correlated, rather than being anticorrelated. Hence the model considered instantiates within-cluster similarity, but does not establish strong between-cluster difference. We note that such anticorrelations can be readily incorporated by including negative links between hidden cluster nodes.

For the results presented in Fig. 9 we used $N_c = 3$ clusters with $M_b = \{4, 2, 3\}$ items per cluster, $e_b = 0.24$ for all clusters, and $\sigma = 4$.

Trees To construct a dataset with an underlying tree structure, in our simulations we make use of the hierarchical branching diffusion process described previously. Specifically, we used a three level tree with binary branching and flip probability $\epsilon = .15$. As shown, this gives rise to a hierarchically structured singular value decomposition.

To understand the generality of this result we can also formulate hierarchical structure in the Gaussian Markov random field framework. To implement a tree structure, we have a set of internal nodes corresponding to each branching point in the tree, in addition to the P leaf nodes corresponding to individual items. We form the adjacency graph $\mathbf{A}$ and compute the inverse precision matrix $\tilde{\Phi}$ as usual. To obtain the feature correlations on just the items of interest, we project out the internal nodes using the linear map $\mathbf{M}$. This ultimately imparts ultrametric structure in the feature correlation matrix $\mathbf{\Sigma}^y$. As shown in [11], such matrices are diagonalized by the ultrametric wavelet transform, which therefore respects the underlying tree structure in the dataset. An important special case is binary branching trees, which are diagonalized by the Haar wavelets [12].

Rings and Grids Items arrayed in rings and grids, such as cities on the globe or locations in an environment, yield correlation matrices with substantial structure. For a ring, correlation matrices are circulant, meaning that every row is a circular permutation of the preceding row. For a grid, correlation matrices are Toeplitz, meaning that they have constant values along each diagonal. Circulant matrices are diagonalized by the unitary Fourier transform [13], and so object analyzer vectors will be sinusoids of differing frequency. The associated singular value is the magnitude of the Fourier coefficient. If correlations are decreasing with distance in the ring, then the broadest spatial distinctions will be learned first, followed by progressive elaboration at ever finer scales, in an analogous process to progressive differentiation in hierarchical structure. Grid structures are not exactly diagonalized by the Fourier modes, but the eigenvalues of Circulant and Toeplitz matrices converge as the grid structure grows large and edge effects become small [13]. Our example is given in a 1D ring, but the same structure arises for higher dimensional lattices (yielding, for instance, doubly block circulant structure in a 2D ring [13, 14] which is diagonalized by the 2D Fourier transform).

In Fig. 9 , we used $P = 20$ items in a ring-structured GMRF in which items are only connected to their immediate neighbors. These connections have length $e_{ij} = 1/.7$ such that $\mathbf{A}_{ij} = 0.7$ if i, j are adjacent nodes. Finally we took the individual variance to be $1/\sigma^2 = 0.09$.

Orderings A simple version of data with an underlying transitive ordering is given by a 1D chain. In the GMRF framework, this will yield Toeplitz correlations in which the first dimension encodes roughly linear position as described above for grids. To instantiate a more complex example, in Fig. 9 we also consider a domain in which a transitive ordering is obeyed exactly: any feature possessed by a higher order entity is also possessed by all lower-order entities. This situation might arise in social dominance hierarchies, for example, with features corresponding to statements like "individual i dominates individual j" (see for example [10, 15]). To instantiate this, we use the input-output correlations

$$\boldsymbol{\Sigma}^{yx} = \frac{1}{P}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad \textbf{[S63]}$$

which realizes a scenario in which a group of items obeys a perfect transitive ordering. This structure yields feature correlations that take constant values on or below the diagonal in each column, and on or to the right of the diagonal in each row.

Cross-cutting Structure Real world datasets need not conform exactly to any one of the individual structures described previously. The domain of animals, for instance, might be characterized by a broadly tree-like structure, but nevertheless contains other regularities such as *male/female*, *predator/prey*, or *arctic/equatorial* which can cut across the hierarchy [16]. These will be incorporated into the hidden representation as additional dimensions which can span items in different branches of the tree. The example given in Fig. 9 instantiates a version of this scenario. The input-output correlation matrix is given by

$$\boldsymbol{\Sigma}^{yx} = \frac{1}{P}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1.1 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 1.1 \\ 1.1 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 1.1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad \textbf{[S64]}$$

This dataset has a hierarchical structure that is repeated for pairs of items, except for the final two features which encode categories that cut across the hierarchy. The feature values of 1.1 which occur in the finer levels of the hierarchy are to create a separation in singular values between the hierarchy modes and the cross-cutting structure.

Examples of this form can be cast in the GMRF framework by combining a tree structure with links to two categories representing the cross-cutting dimensions. The structure of the graph is depicted approximately in Fig. 9, but we note that it cannot be accurately portrayed in three dimensions: all members of each cross-cutting category should connect to a latent category node, and the length of the links to each category member should be equal. Additionally, the final links from the tree structure (depicted with dashed lines) should have length zero, indicating that without the cross-cutting structure the paired items would not differ.

Deploying Knowledge: Inductive Projection

In this section we consider how knowledge about novel items or novel properties will be extended to other items and properties. For instance, suppose that a novel property is observed for a familiar item (e.g., "a *pine* has property *x*"). How will this knowledge be extended to other items (e.g., "does a *rose* have property *x*")? Here we are interested in the interaction between two timescales of learning: the slow, gradual process of development that we describe with error correcting learning in a deep linear network; and the potentially rapid generalizations that can be made upon learning a new fact, based on the current background knowledge embedded in the network. To model this more rapid learning of a new fact, we add a new neuron representing the novel property or item, and following [17], apply error-correcting gradient descent learning only to the new synapses introduced between this new neuron and the hidden layer. In particular, if a novel property m is ascribed to item i, we instantiate an additional element $\hat{\mathbf{y}}_m$ in the output layer of the neural network and add an additional row of weights $\mathbf{w}_m^{2^T}$ to $\mathbf{W}^2$ representing new synaptic connections to this property neuron from the hidden layer. These weights are learned through gradient descent to attain the desired property value. Notably, we do not change other weights in the network (such as those from the input to the hidden layer), so as to prevent this fast learning of a single property from interfering with the broader bulk of knowledge already stored in the network (see [18, 19] for a discussion of the catastrophic interference that arises from rapid non-interleaved learning). This yields the weight update

$$\begin{aligned} \tau_f \frac{d}{dt}\mathbf{w}_m^{2^T} &= \frac{\partial}{\partial \mathbf{w}_m^{2^T}} \frac{1}{2}(\mathbf{y}_m^i - \hat{\mathbf{y}}_m^i)^2 \\ &= (1 - \mathbf{w}_m^{2^T}\mathbf{h}_i)\mathbf{h}_i^T \end{aligned}$$

where we have assumed the desired value for the feature $\mathbf{y}_m^i = 1$ and $\mathbf{h}_i = \mathbf{R}\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{x}^i$ is the hidden representation of item i. Here the time constant τ_f can be substantially faster than the time constant τ driving the development process. In this case, the dynamics above will converge rapidly to a steady state. If the new synaptic weights start at zero ($\mathbf{w}_m^{2^T}(0) = 0$), they converge to

$$\mathbf{w}_m^{2^T} = \mathbf{h}_i^T / ||\mathbf{h}_i||_2^2,$$

mirroring the hidden representation but with an appropriate rescaling. With these weights set, we may now ask how this knowledge will be extended to another item j with hidden representation $\mathbf{h}_j$. The network's prediction is

$$\begin{aligned} \hat{\mathbf{y}}_m^j &= (\mathbf{W}^2\mathbf{W}^1)_{mj}, \\ &= \mathbf{w}_m^{2^T}\mathbf{h}_j, \\ &= \mathbf{h}_i^T\mathbf{h}_j / ||\mathbf{h}_i||_2^2, \end{aligned}$$

yielding Eqn. (16) of the main text. Hence generalization occurs in proportion to the overlap in hidden representations between the familiar item i to which the new property m was ascribed and the familiar probe item j.

A parallel situation exists for learning that a novel item i possesses a familiar feature m. We add a new input node $\mathbf{x}_i$ to the network corresponding to the novel item. This input node is connected to the hidden layer through a new set of synaptic weights $\mathbf{w}_i^1$ which form a new column of $\mathbf{W}^1$. To leave the knowledge in the network intact, we perform gradient learning on only these new connections, corresponding to the "backpropagation to representation" procedure used by [17]. Define $\mathbf{h}_m$ to be the transpose of the mth row of $\mathbf{U}\sqrt{\mathbf{A}(t)}\mathbf{R}^T$, that is, the "backpropagated" hidden representation of feature m. Then

$$\begin{aligned} \tau_f \frac{d}{dt}\mathbf{w}_i^1 &= \frac{\partial}{\partial \mathbf{w}_i^1} \frac{1}{2}(\mathbf{y}_m^i - \hat{\mathbf{y}}_m^i)^2 \\ &= (1 - \mathbf{h}_m^T\mathbf{w}_i^1)\mathbf{h}_m \end{aligned}$$

where we have assumed that the familiar feature has value $\mathbf{y}_m^i = 1$ and the novel input $\mathbf{x}^i$ is a one-hot vector with its ith element equal to one and the rest zero. Solving for the steady state (starting from zero initial weights $\mathbf{w}_i^1(0) = 0$) yields weights

$$\mathbf{w}_i^1 = \mathbf{h}_m/||\mathbf{h}_m||_2^2.$$

With these weights configured, the extent to which the novel object i will be thought to have another familiar feature n is

$$\begin{aligned}\hat{\mathbf{y}}_n^i &= (\mathbf{W}^2\mathbf{W}^1)_{ni}, \\ &= \mathbf{h}_n^T\mathbf{w}_i^1, \\ &= \mathbf{h}_n^T\mathbf{h}_m/||\mathbf{h}_m||_2^2,\end{aligned}$$

yielding Eqn. (17) of the main text.

Linking Behavior and Neural Representations

Similarity structure is an invariant of optimal learning. Here we show that two networks trained on the same statistics starting from small random initial conditions will have identical similarity structure in their hidden layer representations. This relation does not hold generally, however: hidden activity similarity structure can vary widely between networks that still perform the same input-output task. We show that identical similarity structure arises only in networks that optimally implement the desired task in the sense that they use the minimum norm weights necessary to implement the input-output mapping.

The neural activity patterns in response to a set of probe items $\mathbf{X}$, concatenated columnwise into the matrix $\mathbf{H}$, is given by

$$\begin{aligned}\mathbf{H}_1 &= \mathbf{R}_1\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X} \\ \mathbf{H}_2 &= \mathbf{R}_2\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X}.\end{aligned}$$

Hence the similarity structure $\mathbf{H}^T\mathbf{H}$ is identical in both models, since

$$\begin{aligned}\mathbf{H}_1^T\mathbf{H}_1 &= \mathbf{X}^T\mathbf{V}\sqrt{\mathbf{A}(t)}\mathbf{R}_1^T\mathbf{R}_1\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X} \\ &= \mathbf{X}^T\mathbf{V}\mathbf{A}(t)\mathbf{V}^T\mathbf{X} \\ &= \mathbf{X}^T\mathbf{V}\sqrt{\mathbf{A}(t)}\mathbf{R}_2^T\mathbf{R}_2\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X} \\ &= \mathbf{H}_2^T\mathbf{H}_2.\end{aligned}$$

The key fact is simply that the arbitrary rotations are orthogonal, such that $\mathbf{R}_1^T\mathbf{R}_1 = \mathbf{R}_2^T\mathbf{R}_2 = \mathbf{I}$.

This invariance of the hidden similarity structure does not hold in general. Networks can perform the same input-output task but have widely different internal similarity structure. The full space of weight matrices that implement the desired input-output map is given by

$$\begin{aligned}\mathbf{W}^1(t) &= \mathbf{Q}\sqrt{\mathbf{A}(t)}\mathbf{V}^T, & \textbf{[S65]} \\ \mathbf{W}^2(t) &= \mathbf{U}\sqrt{\mathbf{A}(t)}\mathbf{Q}^{-1} & \textbf{[S66]}\end{aligned}$$

That is, the ambiguity in neural representations arising from degeneracy in the solutions is given by any invertible matrix $\mathbf{Q}$. In this more general case, two networks will no longer have identical similarity structure since

$$\begin{aligned}\mathbf{H}_1^T\mathbf{H}_1 &= \mathbf{X}^T\mathbf{V}\sqrt{\mathbf{A}(t)}\mathbf{Q}_1^T\mathbf{Q}_1\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X} \\ &\neq \mathbf{X}^T\mathbf{V}\sqrt{\mathbf{A}(t)}\mathbf{Q}_2^T\mathbf{Q}_2\sqrt{\mathbf{A}(t)}\mathbf{V}^T\mathbf{X} \\ &= \mathbf{H}_2^T\mathbf{H}_2,\end{aligned}$$

because $\mathbf{Q}^T\mathbf{Q} \neq \mathbf{I}$.

Why is the ambiguity in neural representations, encoded by the matrices $\mathbf{R}$, necessarily orthogonal in the learned solution from *tabula rasa*? A well-known optimality principle governs this behavior: among all weight matrices that implement the desired input-output map, these solutions have minimum norm. We prove this here for completeness.

Consider the problem

$$\begin{aligned}&\min_{\mathbf{W}^2,\mathbf{W}^1} ||\mathbf{W}^2||_F^2 + ||\mathbf{W}^1||_F^2 \\ &\text{s.t.} \quad \mathbf{W}^2\mathbf{W}^1 = \mathbf{U}\mathbf{S}\mathbf{V}^T\end{aligned}$$

in which we seek the minimum total Frobenius norm implementation of a particular input-output mapping. We can express the space of possible weight matrices as

$$\begin{aligned}\mathbf{W}^1 &= \mathbf{Q}\mathbf{A}\mathbf{V}^T, \\ \mathbf{W}^2 &= \mathbf{U}\mathbf{A}\mathbf{P}\end{aligned}$$

where $\mathbf{A} = \sqrt{\mathbf{S}}$ and we enforce the constraint $\mathbf{PQ} = \mathbf{I}$. This yields the equivalent problem

$$\begin{aligned}&\min_{\mathbf{P},\mathbf{Q}} ||\mathbf{W}^2||_F^2 + ||\mathbf{W}^1||_F^2 \\ &\text{s.t.} \quad \mathbf{PQ} = \mathbf{I}.\end{aligned}$$

We will show that a minimizer of this problem must have $\mathbf{P} = \mathbf{R}^T$ and $\mathbf{Q} = \mathbf{R}$ for some orthogonal matrix $\mathbf{R}$ such that $\mathbf{R}^T\mathbf{R} = \mathbf{I}$.

To solve this we introduce Lagrange multipliers $\mathbf{\Lambda}$ and form the Lagrangian

$$\begin{aligned}\mathcal{L} &= ||\mathbf{W}^2||_F^2 + ||\mathbf{W}^1||_F^2 + \mathrm{Tr}\left[\mathbf{\Lambda}^T(\mathbf{PQ} - \mathbf{I})\right] \\ &= \mathrm{Tr}\left[\mathbf{P}\mathbf{P}^T\mathbf{A}^2\right] + \mathrm{Tr}\left[\mathbf{Q}^T\mathbf{Q}\mathbf{A}^2\right] \\ &\quad + \mathrm{Tr}\left[\mathbf{\Lambda}^T(\mathbf{PQ} - \mathbf{I})\right].\end{aligned}$$

Differentiating and setting the result to zero we obtain

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{P}} &= 2\mathbf{A}^2\mathbf{P} + \mathbf{\Lambda}\mathbf{Q}^T = 0 \\ \frac{\partial\mathcal{L}}{\partial\mathbf{Q}} &= 2\mathbf{Q}\mathbf{A}^2 + \mathbf{P}^T\mathbf{\Lambda} = 0 \\ \implies \mathbf{\Lambda} &= -2\mathbf{A}^2\mathbf{P}\mathbf{Q}^{-T} = -2\mathbf{P}^{-T}\mathbf{Q}\mathbf{A}^2.\end{aligned}$$

Now note that since $\mathbf{PQ} = \mathbf{I}$, we have $\mathbf{Q} = \mathbf{P}^{-1}$ and $\mathbf{P}^T = \mathbf{Q}^{-T}$, giving

$$\begin{aligned}-2\mathbf{A}^2\mathbf{P}\mathbf{Q}^{-T} &= -2\mathbf{P}^{-T}\mathbf{Q}\mathbf{A}^2 \\ \mathbf{A}^2\mathbf{P}\mathbf{P}^T &= (\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{A}^2 \\ \mathbf{SM} &= \mathbf{M}^{-1}\mathbf{S} & \textbf{[S67]}\end{aligned}$$

where we have defined $\mathbf{M} \equiv \mathbf{P}\mathbf{P}^T$. Decomposing $\mathbf{W}^2$ with the singular value decomposition,

$$\begin{aligned}\mathbf{W}^2 &= \mathbf{U}\tilde{\mathbf{A}}\tilde{\mathbf{V}}^T = \mathbf{UAP} \\ \implies \mathbf{P} &= \mathbf{A}^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{V}}^T \\ &= \mathbf{D}\tilde{\mathbf{V}}^T\end{aligned}$$

where $\mathbf{D} \equiv \mathbf{A}^{-1}\tilde{\mathbf{A}}$ is a diagonal matrix. Hence $\mathbf{M} = \mathbf{P}\mathbf{P}^T = \mathbf{D}^2$, so $\mathbf{M}$ is also diagonal. Returning to Eqn. (S67), we have

$$\begin{aligned}\mathbf{MS} &= \mathbf{M}^{-1}\mathbf{S} \\ \mathbf{M}^2\mathbf{S} &= \mathbf{S}\end{aligned}$$

where we have used the fact that diagonal matrices commute. To satisfy this expression, elements of $\mathbf{M}$ on the diagonal must be ± 1 for any nonzero elements of $\mathbf{S}$, but since $\mathbf{M} = \mathbf{D}^2$ we must select the positive solution. For elements of $\mathbf{S}$ equal to zero, $\mathbf{M}_{ii} = 1$ still satisfies the equation (weights in these directions must be zero). This yields $\mathbf{M} = \mathbf{I}$, and so $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. Therefore $\mathbf{P}$ is orthogonal. Finally $\mathbf{Q} = \mathbf{P}^{-1} = \mathbf{P}^T$, and so is orthogonal as well.

Minimum norm implementations of a network's input-output map thus have the form

$$\begin{aligned}\mathbf{W}^1(t) &= \mathbf{R}\sqrt{\mathbf{A}(t)}\mathbf{V}^T, \\ \mathbf{W}^2(t) &= \mathbf{U}\sqrt{\mathbf{A}(t)}\mathbf{R}^T\end{aligned}$$

where the ambiguity matrix $\mathbf{R}$ is orthogonal, $\mathbf{R}^T\mathbf{R} = \mathbf{I}$. This is identical to the form of the weights found under *tabula rasa* learning dynamics, showing that gradient learning from small initial weights naturally finds the optimal norm solution.

When the brain mirrors behavior. The behavioral properties attributed to each item may be collected into the matrix $\mathbf{Y} = \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{X}$. Its similarity structure $\mathbf{Y}^T\mathbf{Y}$ is thus

$$\begin{aligned}\mathbf{Y}^T\mathbf{Y} &= \mathbf{X}^T\mathbf{W}^1(t)^T\mathbf{W}^2(t)^T\mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{X} \\ &= \mathbf{X}^T\mathbf{V}\mathbf{A}(t)\mathbf{U}^T\mathbf{U}\mathbf{A}(t)\mathbf{V}^T\mathbf{X} \\ &= \mathbf{X}^T\mathbf{V}\mathbf{A}(t)^2\mathbf{V}^T\mathbf{X} \\ &= \left(\mathbf{H}^T\mathbf{H}\right)^2,\end{aligned}$$

where in the last step we have used the assumption that the probe inputs are white ($\mathbf{X}^T\mathbf{X} = \mathbf{I}$), such that they have similar statistics to those seen during learning (recall $\mathbf{\Sigma}^x = \mathbf{I}$ by assumption). This yields Eqn. (18) of the main text. We note that, again, this link between behavior and neural representations emerges only in optimal minimum norm implementations of the input-output map.

Hence the behavioral similarity of items shares the same object-analyzer vectors, and therefore the same categorical structure, as the neural representation; but each semantic distinction is expressed more strongly (according to the square of its singular value) in behavior relative to the neural representation. Intuitively, this greater distinction in behavior is due to the fact that half of the semantic relation is encoded in the output weights $\mathbf{W}^2$, which do not influence the neural similarity of the hidden layer, as it depends only on $\mathbf{W}^1$.

Simulation details for linking behavior and neural representations. Here we describe the experimental parameters for Fig. 11 of the main text. We trained networks on a minimal hand-crafted hierarchical dataset with $N_3 = 7$ features, $N_2 = 32$ hidden units, and $N_1 = P = 4$ items. Inputs were encoded with one-hot vectors. The dataset was given by

$$\begin{aligned}\mathbf{\Sigma}^{yx} &= 0.7P\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \\ \mathbf{\Sigma}^x &= \mathbf{I}.\end{aligned}$$

The full-batch gradient descent dynamics were simulated for four networks with $\lambda = 0.01$ for a thousand epochs. Networks were initialized with independent random Gaussian weights in both layers,

$$\begin{aligned}\mathbf{W}^1(0)_{ij} &\sim \mathcal{N}(0, a_0^2/N_1) \\ \mathbf{W}^2(0)_{ij} &\sim \mathcal{N}(0, a_0^2/N_3).\end{aligned}$$

The two small-initialization networks (panels A-B) had $a_0 = 0.0002$ while the two large initialization networks (panels C-D) had $a_0 = 1$. Individual neural responses and representational similarity matrices from the hidden layer and behavior were calculated at the end of learning, using probe inputs corresponding to the original inputs ($\mathbf{X} = \mathbf{I}$).

1. P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw*, 2(1):53–58, 1989.
2. K. Fukumizu. Effect of Batch Learning In Multilayer Neural Networks. In *Proc 5th International Conference on Neural Information Processing*, 1998.
3. A.M. Saxe, J.L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Y. Bengio and Y. LeCun, editors, *International Conference on Learning Representations*, Banff, Canada, 2014.
4. P. Baldi and Y. Chauvin. Temporal Evolution of Generalization during Learning in Linear Networks. *Neural Computation*, 3:589–603, 1991.
5. M. Advani and A.M. Saxe. High-dimensional dynamics of generalization error in neural networks. In *arXiv*, 2017.
6. A.K. Lampinen and S. Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In T. Sainath, editor, *International Conference on Learning Representations*, 2019.
7. C. Kemp, A. Perfors, and J.B. Tenenbaum. Learning domain structures. In *Proc Ann Meet Cogn Sci Soc*, volume 26, pages 672–7, January 2004.
8. F. Benaych-Georges and R.R. Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *J Multivar Anal*, 111:120–135, 2012.
9. J. Baik, G.B. Arous, and S. Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Ann Probab*, 33(5):1643–1697, 2005.
10. C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proc Natl Acad Sci USA*, 105(31):10687–92, August 2008.
11. A.Y. Khrennikov and S.V. Kozyrev. Wavelets on ultrametric spaces. *Appl Comput Harmon A*, 19:61–76, 2005.
12. F. Murtagh. The haar wavelet transform of a dendrogram. *J Classif*, 24(1):3–32, 2007.
13. R.M. Gray. Toeplitz and Circulant Matrices: A Review. *Found Trends Commun Inf Theory*, 2(3):155–239, 2005.
14. G.J. Tee. Eigenvectors of block circulant and alternating circulant matrices. *Res Lett Inf Math Sci*, 8:123–142, 2005.
15. C. Kemp and J.B. Tenenbaum. Structured statistical models of inductive reasoning. *Psychol Rev*, 116(1):20–58, 2009.
16. J.L. McClelland, Z. Sadeghi, and A.M. Saxe. A Critique of Pure Hierarchy: Uncovering Cross-Cutting Structure in a Natural Dataset. *Neurocomputational Models of Cognitive Development and Processing*, pages 51–68, 2016.
17. T.T. Rogers and J.L. McClelland. *Semantic cognition: A parallel distributed processing approach.* MIT Press, Cambridge, MA, 2004.
18. M. McCloskey and N.J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In G.H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.
19. J.L. McClelland, B.L. McNaughton, and R.C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol Rev*, 102(3):419–57, 1995.