

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/328528381>

A mathematical theory of semantic development in deep neural networks

Preprint · October 2018

CITATIONS

0

READS

137

3 authors, including:



James L. McClelland
Stanford University

347 PUBLICATIONS 58,272 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Semantics [View project](#)



Simulating N400 amplitudes with neural network models [View project](#)

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

An extensive body of empirical research has revealed remarkable regularities in the acquisition, organization, deployment, and neural representation of human semantic knowledge, thereby raising a fundamental conceptual question: what are the theoretical principles governing the ability of neural networks to acquire, organize, and deploy abstract knowledge by integrating across many individual experiences? We address this question by mathematically analyzing the nonlinear dynamics of learning in deep linear networks. We find exact solutions to this learning dynamics that yield a conceptual explanation for the prevalence of many disparate phenomena in semantic cognition, including the hierarchical differentiation of concepts through rapid developmental transitions, the ubiquity of semantic illusions between such transitions, the emergence of item typicality and category coherence as factors controlling the speed of semantic processing, changing patterns of inductive projection over development, and the conservation of semantic similarity in neural representations across species. Thus, surprisingly, our simple neural model qualitatively recapitulates many diverse regularities underlying semantic development, while providing analytic insight into how the statistical structure of an environment can interact with nonlinear deep learning dynamics to give rise to these regularities.

semantic cognition | neural networks | hierarchical generative models

Abbreviations: SVD, singular value decomposition

Human cognition relies on a rich reservoir of semantic knowledge enabling us to organize and reason about our complex sensory world [1–4]. This semantic knowledge allows us to answer basic questions from memory (i.e. "Do birds have feathers?"), and relies fundamentally on neural mechanisms that can organize individual items, or entities (i.e. *Canary*, *Robin*) into higher order conceptual categories (i.e. *Birds*) that include items with similar features, or properties. This knowledge of individual entities and their conceptual groupings into categories or other ontologies is not present in infancy, but develops during childhood [1, 5], and in adults, it powerfully guides the deployment of appropriate inductive generalizations.

The acquisition, organization, deployment, and neural representation of semantic knowledge has been intensively studied, yielding many well-documented empirical phenomena. For example, during acquisition, broader categorical distinctions are generally learned before finer-grained distinctions [1, 5], and long periods of relative stasis can be followed by abrupt conceptual reorganization [6, 7]. Intriguingly, during these periods of developmental stasis, children can strongly believe illusory, incorrect facts about the world [2].

Also, many psychophysical studies of performance in semantic tasks have revealed empirical regularities governing the organization of semantic knowledge. In particular, category membership is a *graded* quantity, with some items being more or less typical members of a category (i.e. a sparrow is a more typical bird than a penguin). The notion of item typicality is both highly reproducible across individuals [8, 9] and correlates with performance on a diversity of semantic tasks [10–14]. Moreover, certain categories themselves are thought to be highly coherent (i.e. the set of things that are *Dogs*), in contrast to less coherent categories (i.e. the set of things that are *Blue*). More coherent categories play a privileged role in the organization of our semantic knowledge; coherent categories are the ones that are most easily learned and represented [8, 15, 16]. Also, the or-

ganization of semantic knowledge powerfully guides its deployment in novel situations, where one must make inductive generalizations about novel items and properties [2, 3]. Indeed, studies of children reveal that their inductive generalizations systematically change over development, often becoming more specific with age [2, 3, 17–19].

Finally, recent neuroscientific studies have begun to shed light on the organization of semantic knowledge in the brain. The method of representational similarity analysis [20, 21] has revealed that the similarity structure of neural population activity patterns in high level cortical areas often reflect the semantic similarity structure of stimuli, for instance by differentiating inanimate objects from animate ones [22–26]. And strikingly, studies have found that such neural similarity structure is preserved across human subjects, and even between humans and monkeys [27, 28].

This wealth of empirical phenomena raises a fundamental conceptual question about how neural circuits, upon experiencing many individual encounters with specific items, can over developmental time scales extract abstract semantic knowledge consisting of useful categories that can then guide our ability to reason about the world and inductively generalize. While a diverse set of theories have been advanced to explain human semantic development, there is currently no analytic, mathematical theory of neural circuits that can account for the diverse phenomena described above. Interesting non-neural accounts for the discovery of abstract semantic structure include for example the conceptual "theory-theory" [2, 16–18], and computational Bayesian [29] approaches. However, neither currently proposes a neural implementation that can infer abstract concepts from a stream of specific examples. In contrast, much prior work has shown, through simulations, that neural networks can gradually extract semantic structure by incrementally adjusting synaptic weights via error-corrective learning [4, 30–37]. However, in contrast to the computational transparency enjoyed by Bayesian approaches, the theoretical principles governing how even simple artificial neural networks extract semantic knowledge from their ongoing stream of experience, embed this knowledge in their synaptic weights, and use these weights to perform inductive generalization, remains obscure.

In this work, our fundamental goal is to fill this gap by employing an exceedingly simple class of neural networks, namely deep linear networks. Surprisingly, we find that this model class can qualitatively account for a diversity of phenomena involving semantic cognition described above. Indeed, we build upon a considerable neural network literature [30–37] addressing such phenomena through simulations of more complex nonlinear networks. We build particularly on the integrative, simulation-based treatment of semantic cognition in [4], often using the same modeling strategy in a simpler linear set-

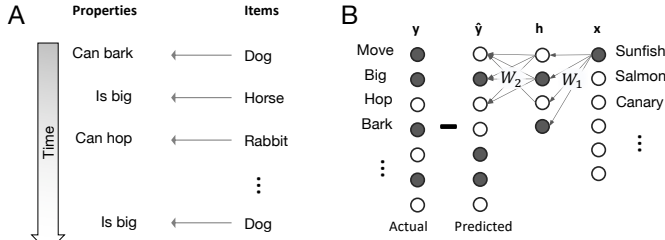


Fig. 1. (A) During development, the network experiences sequential episodes with items and their properties. (B) After each episode, the network adjusts its synaptic weights to reduce the discrepancy between actual observed properties y and predicted properties \hat{y} .

ting, to obtain similar results but with additional analytical insight. Thus, in contrast to prior work, whether conceptual, Bayesian, or connectionist, our simple model is the first to permit exact analytical solutions describing the entire developmental trajectory of knowledge acquisition and organization, and its subsequent impact on the deployment and neural representation of semantic structure. In the following, we describe each of these aspects of semantic knowledge acquisition, organization, deployment, and neural representation in sequence, and we summarize our main findings in the discussion.

A Deep Linear Neural Network Model

Here we consider a framework for analyzing how neural networks extract abstract semantic knowledge by integrating across many individual experiences of items and their properties, across developmental time. In each experience, given an item as input, the network is trained to correctly produce its associated properties or features as output. Consider for example, the network's interaction with the semantic domain of living things, schematized in Fig. 1A. If the network encounters an item, such as a *Canary*, perceptual neural circuits produce an activity vector $\mathbf{x} \in \mathbb{R}^{N_1}$ that identifies this item and serves as input to the semantic system. Simultaneously, the network observes some of the item's properties, for example that a canary *Can Fly*. Neural circuits produce an activity feature vector $\mathbf{y} \in \mathbb{R}^{N_3}$ of that item's properties which serves as the desired output of the semantic network. Over time, the network experiences many individual episodes with a variety of different items and their properties. The total collected experience furnished by the environment to the semantic system is thus a set of P examples $\{\mathbf{x}^i, \mathbf{y}^i\}$, $i = 1, \dots, P$, 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's task is to predict an item's properties \mathbf{y} from its perceptual representation \mathbf{x} . These predictions are generated by prop-

agating activity through a three layer linear neural network (Fig. 1B). The input activity pattern \mathbf{x} in the first layer propagates through a synaptic weight matrix \mathbf{W}^1 of size $N_2 \times N_1$, to create an activity pattern $\mathbf{h} = \mathbf{W}^1 \mathbf{x}$ in the second layer of N_2 neurons. We call this layer the "hidden" layer because it corresponds neither to input nor output. The hidden layer activity then propagates to the third layer through a second synaptic weight matrix \mathbf{W}^2 of size $N_3 \times N_2$, producing an activity vector $\hat{\mathbf{y}} = \mathbf{W}^2 \mathbf{h}$ which constitutes the output prediction of the network. The composite function from input to output is thus simply $\hat{\mathbf{y}} = \mathbf{W}^2 \mathbf{W}^1 \mathbf{x}$. For each input \mathbf{x} , the network compares its predicted output $\hat{\mathbf{y}}$ to the observed features \mathbf{y} and it adjusts its weights so as to reduce the discrepancy between \mathbf{y} and $\hat{\mathbf{y}}$.

To study the impact of depth, we will contrast the learning dynamics of this deep linear network to that of a shallow network that has just a single weight matrix, \mathbf{W}^s of size $N_3 \times N_1$ linking input activities directly to the network's predictions $\hat{\mathbf{y}} = \mathbf{W}^s \mathbf{x}$. At first inspection, it may seem that there is no utility whatsoever in considering deep linear networks, since the composition of linear functions remains linear. Indeed, the appeal of deep networks is thought to lie in the increasingly expressive functions they can represent by successively cascading many layers of nonlinear elements [38, 39]. In contrast, deep linear networks gain no expressive power from depth; a shallow network can compute any function that the deep network can, by simply taking $\mathbf{W}^s = \mathbf{W}^2 \mathbf{W}^1$. However, as we see below, the learning dynamics of the deep network is highly *nonlinear*, while the learning dynamics of the shallow network remains linear. Strikingly, many complex, nonlinear features of learning appear even in deep linear networks, and do not require neuronal nonlinearities.

As an illustration of the power of deep linear networks to capture learning dynamics even in nonlinear networks, we compare the two learning dynamics in Fig. 2. Fig. 2A shows a low dimensional visualization of the simulated learning dynamics of a multilayered nonlinear neural network trained to predict the properties of a set of items in a semantic domain of animals and plants (details of the neural architecture and training data can be found in [4]). The nonlinear network exhibits a striking, hierarchical progressive differentiation of structure in its internal hidden representations, in which animals versus plants are first distinguished, then birds versus fish, and trees versus flowers, and finally individual items. This remarkable phenomenon raises important questions about the theoretical principles governing the hierarchical differentiation of structure in neural networks. In particular, how and why do the network's dynamics and the statistical structure of the input conspire to generate this phenomenon? In Fig. 2B we mathematically *derive* this phenomenon by finding *analytic* solutions to the nonlinear dynamics of learning in a deep linear network, when that network is exposed to a hierarchically structured semantic domain, thereby shedding considerable theoretical insight onto the origins of hierarchical differentiation in a deep network. We present the derivation below, but for now, we note that the resemblance between Fig. 2A and Fig. 2B suggests that deep linear networks can form an excellent, analytically tractable model for shedding conceptual insight into the learning dynamics, if not the expressive power, of their nonlinear counterparts.

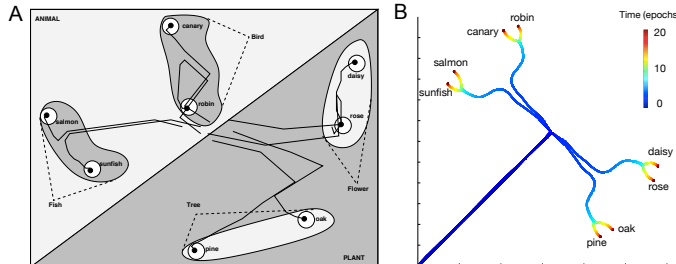


Fig. 2. (A) A two dimensional multi-dimensional scaling (MDS) visualization of the temporal evolution of internal representations, across developmental time, of a deep nonlinear neural network studied in [4]. Reprinted from Figure 3.9, p. 113 of Ref [4]. Copyright © MIT Press, Permission Pending. (B) An MDS visualization of analytically derived learning trajectories of the internal representations of a deep linear network exposed to a hierarchically structured domain.

Acquiring Knowledge

We now begin an outline of the derivation that leads to Fig. 2B. The incremental error corrective process described above can be formalized as online stochastic gradient descent; each time an example i is presented, the weights \mathbf{W}^2 and \mathbf{W}^1 are adjusted by a small amount in the direction that most rapidly decreases the squared error $\|\mathbf{y}^i - \hat{\mathbf{y}}^i\|^2$, yielding the standard back propagation learning rule

$$\Delta \mathbf{W}^1 = \lambda \mathbf{W}^{2T} (\mathbf{y}^i - \hat{\mathbf{y}}^i) \mathbf{x}^{iT}, \quad \Delta \mathbf{W}^2 = \lambda (\mathbf{y}^i - \hat{\mathbf{y}}^i) \mathbf{h}^{iT}, \quad [1]$$

where λ is a small learning rate. This incremental update depends *only* on experience with a *single* item, leaving open the fundamental

conceptual question of how and when the accumulation of such incremental updates can extract over developmental time, abstract structures, like hierarchical taxonomies, that are emergent properties of the *entire* domain of items and their features.

We show the extraction of such abstract domain structure is possible provided learning is gradual, with a small learning rate λ . In this regime, many examples are seen before the weights appreciably change, so learning is driven by the statistical structure of the domain. We imagine training is divided into a sequence of learning epochs. In each epoch the above rule is followed for all P examples in random order. Then averaging [1] over all P examples and taking a continuous time limit gives the mean change in weights per learning epoch,

$$\tau \frac{d}{dt} \mathbf{W}^1 = \mathbf{W}^{2T} (\Sigma^{yx} - \mathbf{W}^2 \mathbf{W}^1 \Sigma^x), \quad [2]$$

$$\tau \frac{d}{dt} \mathbf{W}^2 = (\Sigma^{yx} - \mathbf{W}^2 \mathbf{W}^1 \Sigma^x) \mathbf{W}^{1T}, \quad [3]$$

where $\Sigma^x \equiv E[\mathbf{x}\mathbf{x}^T]$ is an $N_1 \times N_1$ input correlation matrix, $\Sigma^{yx} \equiv E[\mathbf{y}\mathbf{x}^T]$ is an $N_3 \times N_1$ input-output correlation matrix, and $\tau \equiv \frac{1}{P\lambda}$ (see SI for detailed derivation). Here, t measures time in units of learning epochs; as t varies from 0 to 1, the network has seen P examples corresponding to one learning epoch. These equations reveal that learning dynamics in even in our simple linear network can be highly complex: the second order statistics of inputs and outputs drives synaptic weight changes through coupled *nonlinear* differential equations with up to cubic interactions in the weights.

Explicit solutions from *tabula rasa*. These nonlinear dynamics are difficult to solve for arbitrary initial conditions on synaptic weights. However, we are interested in a particular limit: learning from a state of essentially no knowledge, which we model as small random synaptic weights. To further ease the analysis, we shall assume that the influence of perceptual correlations is minimal ($\Sigma^x \approx \mathbf{I}$). Our fundamental goal, then, is to understand the dynamics of learning in (2)-(3) as a function of the input-output correlation matrix Σ^{yx} . The learning dynamics is closely related to terms in the singular value decomposition (SVD) of Σ^{yx} (Fig. 3A),

$$\Sigma^{yx} = \mathbf{U} \mathbf{S} \mathbf{V}^T = \sum_{\alpha=1}^{N_1} s_{\alpha} \mathbf{u}^{\alpha} \mathbf{v}^{\alpha T}, \quad [4]$$

which decomposes any matrix into the product of three matrices. Each of these matrices has a distinct semantic interpretation.

For example, the α 'th column \mathbf{v}^{α} of the $N_1 \times N_1$ orthogonal matrix \mathbf{V} can be thought of as an object analyzer vector; it determines the position of item i along an important semantic dimension α in the training set through the component v_i^{α} . To illustrate this interpretation concretely, we consider a simple example dataset with four items (*Canary*, *Salmon*, *Oak*, and *Rose*) and five properties (Fig. 3). The two animals share the property *can Move*, while the two plants do not. Also each item has a unique property: *can Fly*, *can Swim*, *has Bark*, and *has Petals*. For this dataset, while the first row of \mathbf{V}^T is a uniform mode, the second row, or the second object analyzer vector \mathbf{v}^2 , determines where items sit on an *animal-plant* dimension, and hence has positive values for the *Canary* and *Salmon* and negative values for the plants. The other dimensions identified by the SVD are a *bird-fish* dimension, and a *flower-tree* dimension.

The corresponding α 'th column \mathbf{u}^{α} of the $N_3 \times N_3$ orthogonal matrix \mathbf{U} can be thought of as a *feature synthesizer* vector for semantic distinction α . Its components u_m^{α} indicate the extent to which feature m is present or absent in distinction α . Hence the feature synthesizer \mathbf{u}^2 associated with the *animal-plant* semantic dimension has positive values for *Move* and negative values for *Roots*, as animals typically can move and do not have roots, while plants behave oppositely. Finally the $N_3 \times N_1$ singular value matrix \mathbf{S} has nonzero elements s_{α} , $\alpha = 1, \dots, N_1$ only on the diagonal, ordered

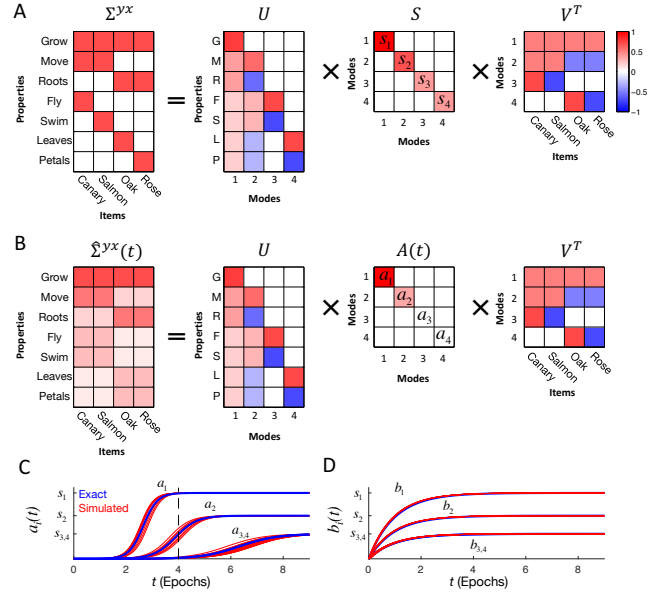


Fig. 3. (A) Singular value decomposition (SVD) of input-output correlations. Associations between items and their properties are decomposed into modes. Each mode links a set of coherently covarying properties (a column of \mathbf{U}) with a set of coherently covarying items (a row of \mathbf{V}^T). The strength of the mode's covariation is encoded by the singular value of the mode (diagonal element of \mathbf{S}). (B) Network input-output map, analyzed via the SVD. The effective singular values (diagonal elements of $\mathbf{A}(t)$) evolve over time during learning. (C) Time-varying trajectories of the deep network's effective singular values $a_i(t)$. Black dashed line marks the point in time depicted in panel B. (D) Time-varying trajectories of a shallow network's effective singular values $b_i(t)$.

so that $s_1 \geq s_2 \geq \dots \geq s_{N_1}$. s_{α} captures the overall strength of the association between the α 'th input and output dimensions. The large singular value for the *animal-plant* dimension reflects the fact that this one dimension explains more of the training set than the finer-scale dimensions like *bird-fish* and *flower-tree*.

Given the SVD of the training set's input-output correlation matrix in (4), we can now explicitly describe the network's learning dynamics. The network's overall input-output map at time t is a time-dependent version of this SVD decomposition (Fig. 3B); it shares the object analyzer and feature synthesizer matrices of the SVD of Σ^{yx} , but replaces the singular value matrix \mathbf{S} with an effective singular value matrix $\mathbf{A}(t)$,

$$\mathbf{W}^2(t) \mathbf{W}^1(t) = \mathbf{U} \mathbf{A}(t) \mathbf{V}^T = \sum_{\alpha=1}^{N_2} a_{\alpha}(t) \mathbf{u}^{\alpha} \mathbf{v}^{\alpha T}, \quad [5]$$

where the trajectory of each effective singular value $a_{\alpha}(t)$ obeys

$$a_{\alpha}(t) = \frac{s_{\alpha} e^{2s_{\alpha} t / \tau}}{e^{2s_{\alpha} t / \tau} - 1 + s_{\alpha} / a_{\alpha}^0}. \quad [6]$$

Eqn. 6 describes a sigmoidal trajectory that begins at some initial value a_{α}^0 at time $t = 0$ and rises to s_{α} as $t \rightarrow \infty$, as plotted in Fig. 3C. This solution is applicable when the network begins as a *tabula rasa*, or an undifferentiated state with little initial knowledge, corresponding small random initial weights (see SI for derivation), and it provides an accurate description of the learning dynamics in this regime, as confirmed by simulation in Fig. 3C.

This solution also gives insight into how the internal representations in the hidden layer of the deep network evolve. An exact solution for \mathbf{W}^2 and \mathbf{W}^1 is given by

$$\mathbf{W}^1(t) = \mathbf{Q} \sqrt{\mathbf{A}(t)} \mathbf{V}^T, \quad \mathbf{W}^2(t) = \mathbf{U} \sqrt{\mathbf{A}(t)} \mathbf{Q}^{-1}, \quad [7]$$

where \mathbf{Q} is an arbitrary $N_2 \times N_2$ invertible matrix (SI Appendix). If initial weights are small, then the matrix \mathbf{Q} will be close to orthogonal, i.e., $\mathbf{Q} \approx \mathbf{R}$ where $\mathbf{R}^T \mathbf{R} = \mathbf{I}$. Thus the internal representations are specified up to an arbitrary rotation \mathbf{R} . Factoring out the rotation, the hidden representation of item i is

$$h_i^\alpha = \sqrt{a^\alpha(t)} \mathbf{v}_i^\alpha. \quad [8]$$

Thus internal representations develop over time by projecting inputs onto more and more input-output modes as they are learned.

The shallow network has a solution of analogous form, $\mathbf{W}^s(t) = \sum_{\alpha=1}^{\min(N_1, N_3)} b_\alpha(t) \mathbf{u}^\alpha \mathbf{v}^{\alpha T}$, but now each singular value evolves as

$$b_\alpha(t) = s_\alpha \left(1 - e^{-t/\tau}\right) + b_\alpha^0 e^{-t/\tau}. \quad [9]$$

In contrast to the deep network's sigmoidal trajectory, Eqn. 9 describes a simple exponential approach from the initial value b_α^0 to s_α , as plotted in Fig. 3D. Hence depth fundamentally changes the dynamics of learning, yielding several important consequences below.

Rapid stage like transitions due to depth. We first compare the time-course of learning in deep versus shallow networks as revealed in Eqns. (6) and (9). For the deep network, beginning from a small initial condition $a_\alpha^0 = \epsilon$, each mode's effective singular value $a_\alpha(t)$ rises to within ϵ of its final value s_α in time

$$t(s_\alpha, \epsilon) = \frac{\tau}{s_\alpha} \ln \frac{s_\alpha}{\epsilon} \quad [10]$$

in the limit $\epsilon \rightarrow 0$ (SI Appendix). This is $O(1/s_\alpha)$ up to a logarithmic factor. Hence modes with stronger explanatory power, as quantified by the singular value, are learned more quickly. Moreover, when starting from small initial weights, the sigmoidal transition from no knowledge of the mode to perfect knowledge can be arbitrarily sharp. Indeed the ratio of time spent in the sigmoidal transition regime to the ratio of time spent before making the transition can go to infinity as the initial weights go to zero (see SI Appendix). Thus rapid stage like transitions in learning can be prevalent even in deep linear networks.

By contrast, the timescale of learning for the shallow network is

$$t(s_\alpha, \epsilon) = \tau \ln \frac{s_\alpha}{\epsilon}, \quad [11]$$

which is $O(1)$ up to a logarithmic factor. Hence in a shallow network, the timescale of learning a mode depends only weakly on its associated singular value. Essentially all modes are learned at the same time, with an exponential rather than sigmoidal learning curve.

Progressive differentiation of hierarchical structure. We are now almost in a position to explain how we analytically derived the result in Fig. 2B. The only remaining ingredient is a mathematical description of the training data. Indeed the numerical results in Fig. 2A arose

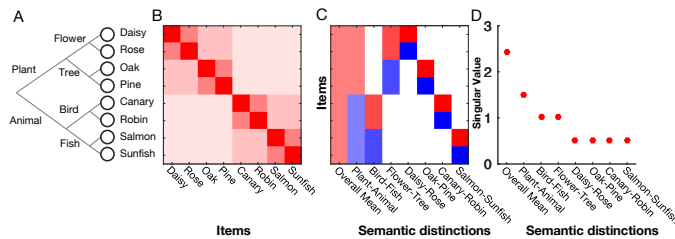


Fig. 4. Hierarchy and the SVD. (A) A domain of eight items with an underlying hierarchical structure. (B) The correlation matrix of the features of the items. (C) Singular value decomposition of the correlations reveals semantic distinctions that mirror the hierarchical taxonomy. This is a general property of the SVD of hierarchical data. (D) The singular values of each semantic distinction reveal its strength in the dataset, and control when it is learned.

from a toy-training set, making it difficult to understand which aspects the data were essential for the hierarchical learning dynamics. Here, we introduce a probabilistic generative model for hierarchically structured data, in order to move beyond toy datasets to extract general principles of how statistical structure impacts learning.

We use a generative model (introduced in [40]) that mimics the process of evolution to create a dataset with explicit hierarchical structure (see SI Appendix). In the model, each feature diffuses down an evolutionary tree (Fig. 4A), with a small probability of mutating along each branch. The items lie at the leaves of the tree, and the generative process creates a hierarchical similarity matrix between items such that items with a more recent common ancestor on the tree are more similar to each other (Fig. 4B). We analytically computed the SVD of this hierarchical dataset and we found that the object analyzer vectors, which can be viewed as functions on the leaves of the tree in Fig. 4C respect the hierarchical branches of the tree, with the larger (smaller) singular values corresponding to broader (finer) distinctions. Moreover, in Fig. 4A we have artificially labelled the leaves and branches of the evolutionary tree with organisms and categories that might reflect a natural realization of this evolutionary process.

Now, inserting the singular values in Fig. 4D (and SI Appendix) into the deep learning dynamics in Eq. 6 to obtain the time-dependent singular values $a^\alpha(t)$, and then inserting these along with the object analyzers vectors \mathbf{v}^α in Fig. 4C into Eq. 8, we obtain a complete analytic derivation of the evolution of internal representations over developmental time in the deep network. An MDS visualization of these evolving hidden representation then yields Fig. 2B, which qualitatively recapitulates the much more complex network and dataset that led to Fig. 2A. In essence, this analysis completes a mathematical proof that the striking progressive differentiation of hierarchical observed in Fig. 2 is an inevitable consequence of deep learning dynamics, even in linear networks, when exposed to hierarchically structured data. The essential intuition is that dimensions of feature variation across items corresponding to broader (finer) hierarchical distinctions have stronger (weaker) statistical structure, as quantified by the singular values of the training data, and hence these dimensions are learned faster (slower), leading to waves of differentiation in a deep, but not a shallow network. Such a pattern of hierarchical differentiation has long been argued to apply to the conceptual development of infants and children [1, 5–7].

Illusory Correlations. Another intriguing aspect of semantic development is that children sometimes attest to false beliefs (i.e. worms have bones [2]) that could not have been learned through direct experience. These errors challenge simple associationist accounts of semantic development that would predict a steady, monotonic accumulation of information about individual properties [2, 16, 17, 41]. Yet as shown in Fig. 5, the network's knowledge of individual properties exhibits complex, non-monotonic trajectories over the course of learning. The overall prediction for a property is a sum of contri-

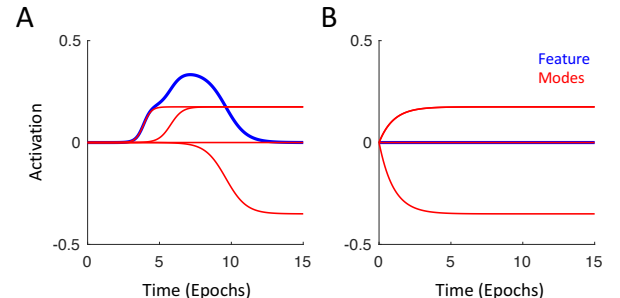


Fig. 5. Illusory correlations during learning. (A) Predicted value (blue) of feature “Can Fly” for item “Salmon” over the course of learning in a deep network (dataset as in Fig. 3). The contributions from each input-output mode are shown in red. (B) The predicted value and modes for the same feature in a shallow network.

butions from each mode, where the specific contribution of mode α to an individual feature m for item i is $a_\alpha(t) \mathbf{u}_m^\alpha \mathbf{v}_i^\alpha$. In the example of Fig. 5A, the first two modes make a positive contribution while the third makes a negative one, yielding the inverted U-shaped trajectory.

Indeed, any property-item combination for which $\mathbf{u}_m^\alpha \mathbf{v}_i^\alpha$ takes different signs across different α will exhibit a non-monotonic learning curve, making such curves a frequent occurrence even in a fixed, unchanging environment. In a deep network, two modes with singular values that differ by Δ will have an interval in which the first is learned but the second is not. The length of this developmental interval is roughly $O(\Delta)$ (SI Appendix). Moreover, the rapidity of the deep network’s stage-like transitions further accentuates the non-monotonic learning of individual properties. This behavior, which may seem hard to reconcile with an incremental, error-corrective learning process, is a natural consequence of minimizing global rather than local prediction error: the fastest possible improvement in predicting all properties across all items sometimes results in a transient increase in the size of errors on specific items and properties. Every property in a shallow network, by contrast, monotonically approaches its correct value and therefore shallow networks provably never exhibit illusory correlations (SI Appendix).

Organizing and Encoding Knowledge

We now turn from the dynamics of learning to its final outcome. When exposed to a variety of items and features interlinked by an underlying hierarchy, for instance, what categories naturally emerge? Which items are particularly representative of a categorical distinction? And how is the structure of the domain internally represented?

Category membership, typicality, and prototypes. A long observed empirical finding is that category membership is not simply a logical, binary variable, but rather a *graded* quantity, with some objects being more or less typical members of a category (i.e. a *sparrow* is a more typical bird than a *penguin*). Indeed, such graded judgements of category membership are both highly reproducible across individuals [8, 9] and moreover correlate with performance on a range of tasks: subjects more quickly verify the category membership of more typical items [10, 11], more frequently recall typical examples of a category [12], and more readily extend new information about typical items to all members of a category [13, 14]. Our theoretical framework provides a natural mathematical definition of item typicality that both explains how it emerges from the statistical structure of the environment and improves task performance.

Indeed, a natural notion of the typicality of an item i for a categorical distinction α is simply the quantity \mathbf{v}_i^α in the corresponding object analyzer vector. To see why this is natural, note that after learning, the neural network’s internal representation space has a semantic distinction axis α , and each object i is placed along this axis at a coordinate proportional to \mathbf{v}_i^α , as seen in Eq. (8). Thus according to our definition, extremal points along this axis are the most typical members of a category. For example, if α corresponds to the bird-fish axis, objects i with large positive \mathbf{v}_i^α are typical birds, while objects i with large negative \mathbf{v}_i^α are typical fish. Also, the contribution of the network’s output to feature neuron m in response to object i , from the hidden representation axis α alone is given by

$$\hat{\mathbf{y}}_m^\alpha \leftarrow \mathbf{u}_m^\alpha s_\alpha \mathbf{v}_i^\alpha. \quad [12]$$

Hence under our definition of typicality, an item i that is more typical than another other item j will have $|\mathbf{v}_i^\alpha| > |\mathbf{v}_j^\alpha|$, and thus will necessarily have a larger response magnitude under Eq. (12). Any performance measure which is monotonic in the response will therefore increase for more typical items under this definition. Thus our definition of item typicality is both a mathematically well defined function of the statistical structure of experience, through the SVD, and proveably correlates with task performance in our network.

Several previous attempts at defining the typicality of an item involve computing a weighted sum of category specific features present or absent in the item [8, 15, 42–44]. For instance, a *sparrow* is a more typical bird than a *penguin* because it shares more relevant features (*can fly*) with other birds. However, the specific choice of which features are relevant—the weights in the weighted sum of features—has often been heuristically chosen and relied on prior knowledge of which items belong to each category [8, 43]. Our definition of typicality can also be described in terms of a weighted sum of an object’s features, but the weightings are *uniquely* fixed by the statistics of the entire environment through the SVD (see SI Appendix):

$$\mathbf{v}_i^\alpha = \frac{1}{P_{S_\alpha}} \sum_{m=1}^{N_3} \mathbf{u}_m^\alpha \mathbf{o}_m^i, \quad [13]$$

which holds for all i, m , and α . Here, item i is defined by its feature vector $\mathbf{o}^i \in R^{N_3}$, where component \mathbf{o}_m^i encodes the value of its m^{th} feature. Thus the typicality \mathbf{v}_i^α of item i in distinction α can be computed by taking a weighted sum of the components of its feature vector \mathbf{o}^i , where the weightings are precisely the coefficients of the corresponding feature synthesizer vector \mathbf{u}^α (scaled by the reciprocal of the singular value). The neural geometry of Eq. 13 is illustrated in Fig. 6 when α corresponds to the bird-fish categorical distinction.

In many theories of typicality, the particular weighting of object features corresponds to a prototypical object [3, 15], or the best example of a particular category. Such object prototypes are often obtained by a weighted average over the feature vectors for the objects in a category (i.e. averaging together the features of all birds, for instance, will give a set of features they share). However, such an average relies on prior knowledge of the extent to which an object belongs to a category. Our theoretical framework also yields a natural notion of object prototypes as a weighted average of object feature vectors, but unlike many other frameworks, it yields a *unique* prescription for the object weightings in terms of the statistical structure of the environment through the SVD (SI Appendix):

$$\mathbf{u}_m^\alpha = \frac{1}{P_{S_\alpha}} \sum_{i=1}^{N_1} \mathbf{v}_i^\alpha \mathbf{o}_m^i. \quad [14]$$

Thus the feature synthesizer \mathbf{u}^α , can itself be thought of as a category prototype for distinction α , as it can be obtained through a weighted average of *all* the object feature vectors \mathbf{o}^i , where the weighting of object i is none other than the *typicality* \mathbf{v}_i^α of object i in distinction α . In essence, each element \mathbf{u}_m^α of the prototype vector signifies how important feature m is for distinction α (Fig. 6).

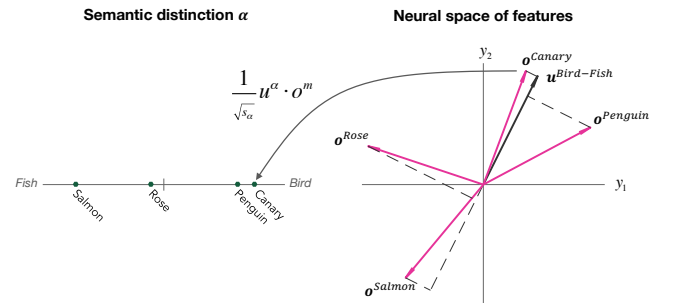


Fig. 6. The geometry of item typicality. For a semantic distinction α (in this case α is the bird-fish distinction) the object analyzer vector \mathbf{v}_i^α arranges objects i along an internal neural representation space where the most typical birds take the extremal positive coordinates, and the most typical fish take the extremal negative coordinates. Objects like a rose, that is neither a bird nor a fish, are located near the origin on this axis. Positions along the neural semantic axis can also be obtained by computing the inner product between the feature vector \mathbf{o}^i for object i and the feature synthesizer \mathbf{u}^α as in (13). Moreover \mathbf{u}^α can be thought of as a category prototype for semantic distinction α through (14).

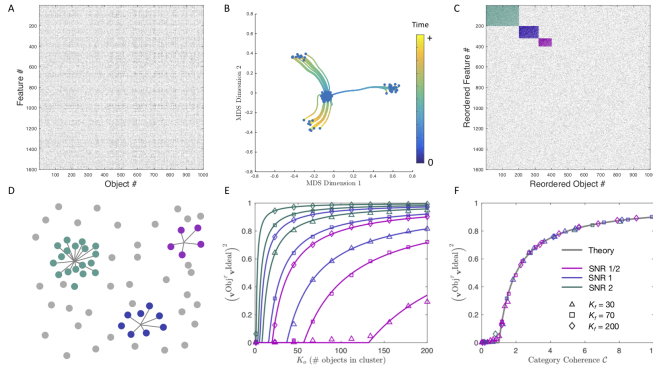


Fig. 7. The discovery of disjoint categories buried in noise. (A) A data set of $N_0 = 1000$ items and $N_f = 1600$ features, with no discernible visible structure. (B) Yet when a deep linear network learns to predict the features of items, an MDS visualization of the evolution of its internal representations reveals 3 clusters. (C) By computing the SVD of the product of synaptic weights $\mathbf{W}^2\mathbf{W}^1$, we can extract the network’s object analyzers \mathbf{v}^α and feature synthesizers \mathbf{u}^α , finding 3 with large singular values s_α , for $\alpha = 1, \dots, 3$. Each of these 3 object analyzers \mathbf{v}^α and feature synthesizers \mathbf{u}^α takes large values on a subset of items and features respectively, and we can use them to reorder the rows and columns of (A) to obtain (C). This re-ordering reveals underlying structure in the data corresponding to 3 disjoint categories, such that if a feature and item belong to a category, the feature is present with a high probability p , whereas if it does not, it appears with a low probability q . (D) Thus intuitively, the dataset corresponds to 3 clusters buried in a noise of irrelevant objects and features. (E) Performance in recovering one such category can be measured by computing the correlation coefficients between the object analyzer and feature synthesizer returned by the network to the ideal object analyzer $\mathbf{v}^{\text{Ideal}}$ and ideal feature synthesizer $\mathbf{u}^{\text{Ideal}}$ that take nonzero values on the items and features, respectively, that are part of the category, and are zero on the rest of the items and features. This learning performance, for the object analyzer, is shown for various parameter values. Solid curves are analytical theory derived from a random matrix analysis (SI Appendix) and data points are obtained from simulations. (F) All performance curves in (E) collapse onto a *single* theoretically predicted, universal learning curve, when measured in terms of the category coherence defined in Eq. 15.

In summary, a beautiful and simple duality between item typicality and category prototypes arises as an emergent property of the learned internal representations of the neural network. The typicality of an item is determined by the projection of that item’s feature vector onto the category prototype in (13). And the category prototype is an average over all object feature vectors, weighted by their typicality in (14). Moreover, in any categorical distinction α , the most typical items i and the most important features m are determined by the *extremal* values of \mathbf{v}_i^α and \mathbf{u}_m^α .

Category coherence. The categories we naturally learn are not arbitrary, but instead are in some sense coherent, and efficiently represent the structure of the world [8, 15, 16]. For example, the set of things that are *Red* and *cannot Swim*, is a well defined category, but intuitively is not as coherent as the category of *Dogs*; we naturally learn, and even name, the latter category, but not the former. When is a category learned at all, and what determines its coherence? An influential proposal [8, 15] suggested that coherent categories consist of tight clusters of items that share many features, and moreover are highly distinct from other categories with different sets of shared features. Such a definition, as noted in [3, 16, 17] can be circular: to know which items are category members, one must know which features are important for that category, and conversely, to know which features are important, one must know which items are members. Thus a mathematical definition of category coherence, as a function of the statistical structure of the environment, that is provably related to the learnability of categories by neural networks, has remained elusive.

Here we provide such a definition for a simple model of disjoint categories, and demonstrate how neural networks can cut through the

Gordian knot of circularity. Our definition and theory is motivated by, and consistent with, prior network simulations exploring notions of category coherence through the coherent covariation of features [4].

Consider for example, a dataset consisting of N_o objects and N_f features. Now consider a category consisting of a subset of K_f features that tend to occur with high probability p in a subset of K_o items, whereas a background feature occurs with a lower probability q in a background item p when either are not part of the category. For what values of K_f , K_o , p , q , N_f and N_o can such a category be learned, and if so, how accurately? Fig. 7A-D illustrates, for example, how a neural network can extract 3 such categories buried in the background noise. We see in Fig. 7E that the performance of category extraction increases as the number of items K_o and features K_f in the category increases, and also as the signal-to-noise ratio, or $\text{SNR} \equiv \frac{(p-q)^2}{q(1-q)}$ increases. Through random matrix theory (SI Appendix), we show that performance depends on the various parameters *only* through a category coherence variable

$$C = \text{SNR} \frac{K_o K_f}{\sqrt{N_o N_f}}. \quad [15]$$

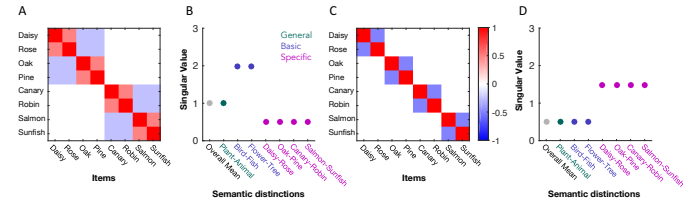


Fig. 8. From hierarchical similarity structure to category coherence. (A) A hierarchical similarity structure over objects in which categories at the basic level are very different from each other due to a negative similarity. (B) For such similarity structure, basic level categorical distinctions acquire larger singular values, or category coherence, and therefore gain an advantage in both learning and in task performance. (C) Now subordinate categories are very different from each other through negative similarity. (D) Consequently, subordinate categories gain a coherence advantage. See SI Appendix for analytic formulas relating similarity structure to category coherence.

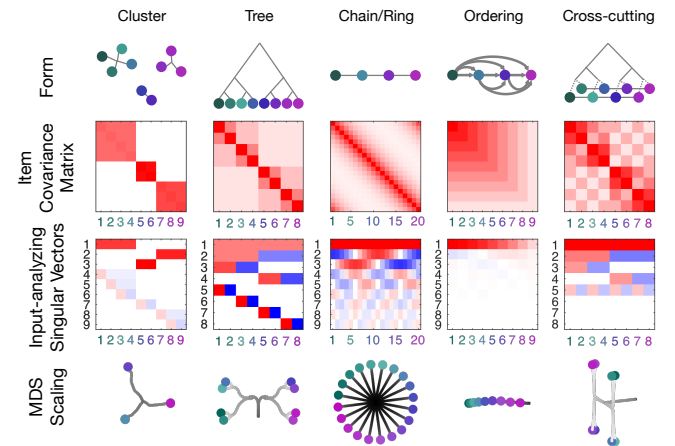


Fig. 9. Representation of explicit structural forms in a neural network. Each column shows a different structure. The first four columns correspond to pure structural forms, while the final column has cross-cutting structure. First row: The structure of the data generating probabilistic graphical model (PGM). Second row: The resulting item covariance matrix arising from either data drawn from the PGM (first four columns) or designed by hand (final column). Third row: The input-analyzing singular vectors that will be learned by the linear neural network. Each vector is scaled by its singular value, showing its importance to representing the covariance matrix. Fourth row: MDS view of the development of internal representations.

When the performance curves in Fig. 7E are re-plotted with category coherence \mathcal{C} on the horizontal axis, all the curves collapse onto a single *universal* performance curve shown in Fig. 7F. We derive a mathematical expression for this curve in SI Appendix. It displays an interesting threshold behavior: if the coherence $\mathcal{C} \leq 1$, the category is not learned at all, and the higher the coherence above this threshold, the better the category is learned.

This threshold is strikingly permissive. For example, at $\text{SNR} = 1$, it occurs at $K_o K_f = \sqrt{N_o N_f}$. Thus in a large environment of $N_o = 1000$ objects and $N_f = 1600$ features, as in Fig. 7A, a small category of 40 objects and 40 features can be easily learned, even by a simple deep linear network. Moreover, this analysis demonstrates how the deep network solves the problem of circularity described above by simultaneously bootstrapping the learning of object analyzers and feature synthesizers in its synaptic weights. Finally, we note that the definition of category coherence in Eq. (15) is qualitatively consistent with previous notions; coherent categories consist of large subsets of items possessing, with high probability, large subsets of features that tend not to co-occur in other categories. However, our quantitative definition has the advantage that it proveably governs category learning performance in a neural network.

Basic categories. Closely related to category coherence, a variety of studies of naming have revealed a privileged role for *basic* categories at an intermediate level of specificity (i.e. *Bird*), compared to superordinate (i.e. *Animal*) or subordinate (i.e. *Robin*) levels. At this basic level, people are quicker at learning names [45, 46], prefer to generate names [46], and are quicker to verify the presence of named items in images [11, 46]. We note that basic level advantages typically involve naming tasks done at an older age, and so need not be inconsistent with progressive differentiation of categorical structure from superordinate to subordinate levels as revealed in preverbal cognition [1, 4–7, 47]. Moreover, some items are named more frequently than others, and these frequency effects could contribute to a basic level advantage [4]. However in artificial category learning experiments where frequencies are tightly controlled, basic level categories are still often learned first [48]. What statistical properties of the environment could lead to this basic level effect? While several properties have been put forth in the literature [11, 42, 44, 48], a mathematical function of environmental structure that proveably confers a basic level advantage to neural networks has remained elusive.

Here we provide such a function by generalizing the notion of category coherence \mathcal{C} in the previous section to hierarchically structured categories. Indeed, in any dataset containing strong categorical structure, so that its singular vectors are in one to one correspondence with categorical distinctions, we simply propose to *define* the coherence of a category by the associated singular value. This definition has the advantage of obeying the theorem that more coherent categories are learned faster, through Eq. (6). Moreover, we show in SI Appendix that this definition is consistent with that of category coherence \mathcal{C} defined in Eq. (15) for the special case of disjoint categories. However, for hierarchically structured categories as in Fig. 4, this singular value definition always predicts an advantage for superordinate categories, relative to basic or subordinate.

Is there an alternate statistical structure for hierarchical categories that confers high category coherence at lower levels in the hierarchy? We exhibit two such structures in Fig. 8. More generally, in the SI Appendix, we analytically compute the singular values at each level of the hierarchy in terms of the similarity structure of items. We find these singular values are a weighted sum of within cluster similarity minus between cluster similarity for all levels below, weighted by the fraction of items that are descendants of that level. If at any level, between cluster similarity is negative, that detracts from the coherence of superordinate categories, contributes strongly to the coherence of categories at that level, and does not contribute to subordinate categories.

Thus the singular value based definition of category coherence is qualitatively consistent with prior intuitive notions. For instance, paraphrasing Keil (1991), coherent categories are clusters of tight bundles of features separated by relatively empty spaces [17]. Also, consistent with [3, 16, 17], we note that we cannot judge the coherence of a category without knowing about its relations to all other categories, as singular values are a complex emergent property of the entire environment. But going beyond past intuitive notions, our quantitative definition of category coherence based on singular values enables us to prove that coherent categories are most easily and quickly learned, and also proveably provide the most accurate and efficient linear representation of the environment, due to the global optimality properties of the SVD (see SI Appendix for details).

Discovering and representing explicit structures. While we have focused on hierarchical structure, the world may contain many different types of abstract structures. How are these different structures learned and encoded by neural networks? A convenient formalization of environmental structure can be specified in terms of a probabilistic graphical model (PGM), defined by a graph over items (Fig. 9 top) that can express a variety of structural forms underlying a domain, including clusters, trees, rings, grids, orderings, and hierarchies. Features are assigned to items by independently sampling from the PGM (see [29] and SI Appendix), such that nearby items in the graph are more likely to share features. For each of these structural forms, in the limit of a large number of features, we computed the item-item covariance matrices (Fig. 9 second row), object analyzer vectors (Fig. 9 third row) and singular values of the resultant input-output correlation matrix, and we employed them in our learning dynamics in Eq. 6 to compute the development of the network’s internal representations through Eq. 8. These evolving hidden representations are shown in (Fig. 9 bottom). Overall, this approach yields several insights into how distinct structural forms, through their different statistics, drive learning in a deep network, as summarized below:

Clusters. Graphs that break items into distinct clusters give rise to block-diagonal constant matrices, yielding object-analyzer vectors that pick out cluster membership.

Trees. Tree graphs give rise to ultrametric covariance matrices, yielding object-analyzer vectors that are tree-structured wavelets that mirror the underlying hierarchy [49, 50].

Rings and Grids. Ring-structured graphs give rise to circulant covariance matrices, yielding object-analyzer vectors that are Fourier modes ordered from lowest to highest frequency [51].

Orderings. Graphs that transitively order items yield highly structured, but non-standard, covariance matrices whose object analyzers encode the ordering.

Cross-cutting Structure. Real-world domains need not have a single underlying structural form. For instance, while some features of animals and plants generally follow a hierarchical structure, other features like *male* and *female* can link together hierarchically disparate items. Such cross-cutting structure can be orthogonal to the hierarchical structure, yielding object-analyzer vectors that span hierarchical distinctions.

In essence, these results reflect an analytic link between two very popular, but different, methods of capturing structure: PGM’s and deep networks. This general analysis transcends the particulars of any one dataset, and shows how different abstract structures become embedded in the internal representations of a deep neural network.

Deploying Knowledge: Inductive Projection

Over the course of development, the knowledge acquired by children powerfully reshapes their inductions upon encountering novel items and properties [2, 3]. For instance, upon learning a novel fact (e.g., “a canary is warm-blooded”) children extend this new knowledge

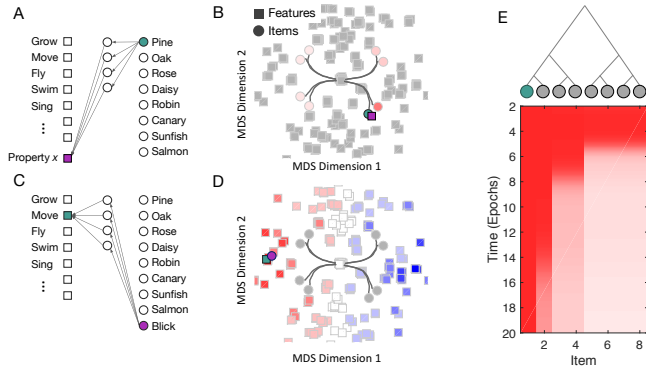


Fig. 10. The neural geometry of inductive generalization. (A) A novel feature (property x) is observed for a familiar item (i.e., "a pine has property x "). (B) Learning assigns the novel feature a neural representation in the hidden layer of the network that places it in semantic similarity space near the object which possesses the novel feature. The network then inductively projects that novel feature to other familiar items (e.g., "Does a rose have property x ?" only if their hidden representation is close in neural space). (C) A novel item (a *blink*) possesses a familiar feature (i.e., "a blink can move"). (D) Learning assigns the novel item a neural representation in the hidden layer that places it in semantic similarity space near the feature possessed by the novel item. Other features are inductively projected to that item (e.g., "Does a blink have wings?") only if their hidden representation is close in neural space. (E) Inductive projection of a novel property ("a pine has property x ") over learning. As learning progresses, the neural representations of items become progressively differentiated, yielding progressively restricted projection of the novel feature to other items. Here the pine can be thought of as the left-most item node in the tree.

to related items, as revealed by their answers to questions like "is a robin warm-blooded?" Studies of inductive projection have shown that children's answers to such questions change over the course of development [2, 3, 17–19], generally becoming more specific with age. For example, young children may readily project the novel property of warm-blooded to distantly related items, while older children will only project it to more closely related items. How could such changing patterns of inductive generalization arise in a neural network? Here, building upon previous network simulations of inductive projection [4, 31], we show analytically that deep networks exposed to hierarchically structured data, naturally yield progressively narrowing patterns of inductive projection across development.

Consider the act of learning that a familiar item has a novel feature (e.g., "a pine has property x "). To accommodate this knowledge, new synaptic weights must be chosen between the familiar item *pine* and the novel property x (Fig. 10A), without disrupting prior knowledge of items and their properties already stored in the network. This may be accomplished by adjusting *only* the weights from the hidden layer to the novel feature so as to activate it appropriately. With these new weights established, inductive projections of the novel feature to other familiar items (e.g., "does a rose have property x ?" naturally arise by querying the network with other inputs. If a novel property m is ascribed to a familiar item i , the inductive projection of this property to any other item j is given by (see SI Appendix),

$$\hat{y}_m = \mathbf{h}_j^T \mathbf{h}_i / \|\mathbf{h}_i\|^2. \quad [16]$$

This equation implements a similarity-based inductive projection of the novel property to other items, where the similarity metric is precisely the Euclidean similarity of hidden representations of pairs of items (Fig. 10B). In essence, being told "a pine has property x ," the network will more readily project the novel property x to those familiar items whose hidden representations are close to that of the pine.

A parallel situation arises upon learning that a novel item possesses a familiar feature (e.g., "a *blink* can move," Fig. 10C). Encoding this knowledge requires new synaptic weights between the item and the hidden layer. Appropriate weights may be found through

standard gradient descent learning of the item-to-hidden weights for this novel item, while holding the hidden-to-output weights fixed to preserve prior knowledge about features. The network can then inductively project other familiar properties to the novel item (e.g., "Does a blink have legs?") by simply generating a feature output vector in response to the novel item as input. Under this scheme, a novel item i with a familiar feature m will be assigned another familiar feature n through the equation (SI Appendix),

$$\hat{y}_n = \mathbf{h}_n^T \mathbf{h}_m / \|\mathbf{h}_m\|^2, \quad [17]$$

where the α^{th} component of \mathbf{h}_n is $\mathbf{h}_n^\alpha = \mathbf{u}_n^\alpha \sqrt{a_\alpha(t)}$. $\mathbf{h}_n \in \mathbf{R}^{N_2}$ can be thought of as the hidden representation of feature n at developmental time t . In parallel to (16), this equation now implements similarity based inductive projection of familiar features to a novel item. In essence, being told "a *blink* can move," the network will more readily project other familiar features to a *blink*, if those features have a similar internal representation as that of the feature move.

Thus the hidden layer of the deep network furnishes a common, semantic representational space into which both features and items can be placed. When a novel feature m is assigned to a familiar item i , that novel feature is placed close to the familiar item in the hidden layer, and so the network will inductively project this novel feature to other items close to i in neural space. In parallel, when a novel item i is assigned a familiar feature m , that novel item is placed close to the familiar feature, and so the network will inductively project other features close to m in neural space, onto the novel item.

This principle of similarity based generalization encapsulated in Eqns. 16 and 17, when combined with the progressive differentiation of internal representations over developmental time as the network is exposed to hierarchically structured data, as illustrated in Fig. 2B, then naturally explains the shift in patterns of inductive projection from broad to specific across development, as shown in Fig. 10E. For example, consider specifically the inductive projection of a novel feature to familiar items (Fig. 10AB). Earlier (later) in developmental time, neural representations of all items are more similar to (different from) each other, and so the network similarity based inductive projection will extend the novel feature to many (fewer) items, thereby exhibiting progressively narrower patterns of inductive projection that respect the hierarchical similarity structure of the environment (Fig. 10E). Thus remarkably, even a deep linear network can provably exhibit the same broad to specific changes in patterns of inductive projection that are empirically observed in many works [2, 3, 17, 18].

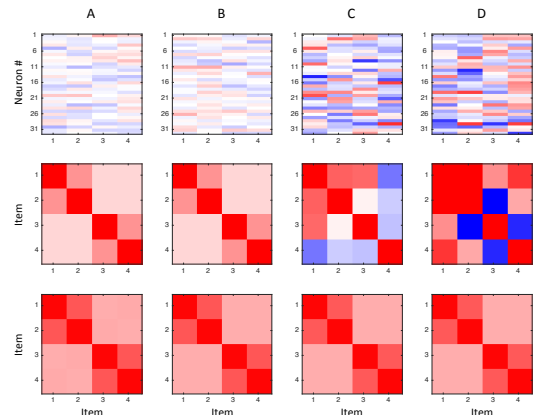


Fig. 11. Neural representations and invariants of learning. Columns A-B depict two networks trained from small norm random weights. Columns C-D depict two networks trained from large norm random weights. Top row: Neural tuning curves \mathbf{h}_i at the end of learning. Neurons show mixed selectivity tuning, and individual tuning curves are different for different trained networks. Middle row: Representational similarity matrix Σ^h . Bottom row: Behavioral similarity matrix Σ^y . For small-norm, but not large-norm weight initializations, representational similarity is conserved and behavioral similarity mirrors neural similarity.

Linking Behavior and Neural Representations

Compared to previous models which have primarily made behavioral predictions, our theory has a clear neural interpretation. Here we discuss implications for the neural basis of semantic cognition.

Similarity structure is an invariant of optimal learning. An influential method for probing neural codes for semantic knowledge in empirical measurements of neural activity is the representational similarity approach (RSA) [20, 21, 28, 52], which examines the similarity structure of neural population vectors in response to different stimuli. This technique has identified rich structure in high level visual cortices, where, for instance, inanimate objects are differentiated from animate objects [22–26]. Strikingly, studies have found remarkable constancy between neural similarity structures across human subjects, and even between humans and monkeys [27, 28]. This highly conserved similarity structure emerges despite considerable variability in neural activity patterns across subjects [53, 54]. Indeed, exploiting similarity structure enables more effective across-subject decoding of fMRI data relative to transferring a decoder based on careful anatomical alignment [55]. Why is representational similarity conserved, both across individuals and species, despite highly variable tuning of individual neurons and anatomical differences?

Remarkably, we show that two networks trained in the same environment *must* have *identical* representational similarity matrices despite having detailed differences in neural tuning patterns, *provided* that the learning process is optimal, in the sense that it yields the smallest norm weights that solve the task (see SI Appendix for a derivation). One way to get close to the optimal manifold of synaptic weights of smallest norm after learning, is to start learning from small random initial weights. We show in Fig. 11AB that two networks, each starting from different sets of small random initial weights, will after training learn very different internal representations (Fig. 11AB top row) but will have nearly identical representational similarity matrices (Fig. 11AB middle row). Such a result is however, not obligatory. Two networks starting from large random initial weights not only learn different internal representations, but also learn different representational similarity matrices (Fig. 11CD top and middle rows). This pair of networks both learn the same composite input output map, but with suboptimal large-norm weights. Hence our theory, combined with the empirical finding that similarity structure is preserved across humans and species, may speculatively suggest that all these disparate neural circuits may be implementing an approximately optimal learning process in a common environment.

When the brain mirrors behavior. In addition to matching neural similarity patterns across subjects, experiments using fMRI and single unit responses have also documented a correspondence between neural similarity patterns and behavioral similarity patterns [21]. When does neural similarity mirror behavioral similarity? We show this correspondence again emerges only in optimal networks.

In particular, denote by $\hat{\mathbf{y}}_i$ the behavioral output of the network in response to item i . These output patterns yield the behavioral similarity matrix $\Sigma^{\hat{\mathbf{y}}}_{ij} = \hat{\mathbf{y}}_i^T \hat{\mathbf{y}}_j$. In contrast, the neural similarity matrix is $\Sigma^{\mathbf{h}}_{ij} = \mathbf{h}_i^T \mathbf{h}_j$ where \mathbf{h}_i is the hidden representation of stimulus i . We show in the SI Appendix that if the network learns optimal smallest norm weights, then these two similarity matrices obey the relation

$$\Sigma^{\hat{\mathbf{y}}} = \left(\Sigma^{\mathbf{h}}\right)^2. \quad [18]$$

Moreover, we show the two matrices share the same singular vectors. Hence behavioral similarity patterns share the same structure as neural similarity patterns, but with each semantic distinction expressed more strongly (according to the square of its singular value) in behavior relative to the neural representation. While this precise math-

ematical relation is yet to be tested in detail, some evidence points to this greater category separation in behavior [27].

Given that optimal learning is a prerequisite for neural similarity mirroring behavioral similarity, as in the previous section, there is a match between the two for pairs of networks trained from small random initial weights (Fig. 11AB middle and bottom rows), but not for pairs of networks trained from large random initial weights (Fig. 11CD middle and bottom rows). Thus again, speculatively, our theory suggests that the experimental observation of a link between behavioral and neural similarity may in fact indicate that learning in the brain is finding optimal network solutions that efficiently implement the requisite transformations with minimal synaptic strengths.

Discussion

In summary, the main contribution of our work is the analysis of a simple model, namely a deep linear neural network, that can, surprisingly, qualitatively capture a diverse array of phenomena in semantic development and cognition. Our exact analytical solutions of nonlinear learning phenomena in this model yield conceptual insights into why such phenomena also occur in more complex nonlinear networks [4, 31–37] trained to solve semantic tasks. In particular, we find that the hierarchical differentiation of internal representations in a deep, but not a shallow, network (Fig. 2) is an inevitable consequence of the fact that singular values of the input-output correlation matrix drive the timing of rapid developmental transitions (Fig. 3 and Eqns. (6) and (10)), and hierarchically structured data contains a hierarchy of singular values (Fig. 4). In turn, semantic illusions can be highly prevalent between these rapid transitions simply because global optimality in predicting all features of all items necessitates sacrificing correctness in predicting some features of some items (Fig. 5). And finally, this hierarchical differentiation of concepts is intimately tied to the progressive sharpening of inductive generalizations made by the network (Fig. 10).

The encoding of knowledge in the neural network after learning also reveals precise mathematical definitions of several aspects of semantic cognition. Basically, the synaptic weights of the neural network extract from the statistical structure of the environment a set of paired object analyzers and feature synthesizers associated with every categorical distinction. The bootstrapped, simultaneous learning of each pair solves the apparent Gordian knot of knowing both which items belong to a category, and which features are important for that category: the object analyzers determine category membership, while the feature synthesizers determine feature importance, and the set of extracted categories are *uniquely* determined by the statistics of the environment. Moreover, by defining the typicality of an item for a category as the strength of that item in the category’s object analyzer, we can prove that typical items must enjoy enhanced performance in semantic tasks relative to atypical items (Eq. (12)). Also, by defining the category prototype to be the associated feature synthesizer, we can prove that the most typical items for a category are those that have the most *extremal* projections onto the category prototype (Fig. 6 and Eq. 13). Finally, by defining the coherence of a category to be the associated singular value, we can prove that more coherent categories can be learned more easily and rapidly (Fig. 7) and explain how changes in the statistical structure of the environment determine what level of a category hierarchy is the most basic or important (Fig. 8). All our definitions of typicality, prototypes and category coherence are broadly consistent with intuitions articulated in a wealth of psychology literature, but our definitions imbue these intuitions with enough mathematical precision to prove theorems connecting them to aspects of category learnability, learning speed and semantic task performance in a neural network model.

More generally, beyond categorical structure, our analysis provides a principled framework for explaining how the statistical structure of diverse structural forms associated with different probabilistic graphical models gradually become encoded in the weights of a neural network (Fig. 9). Also, with regards to neural representa-

tion, our theory reveals that across different networks trained to solve a task, while there may be no correspondence at the level of single neurons, the similarity structure of internal representations of any two networks will both be *identical* to each other, and closely related to the similarity structure of behavior, *provided* both networks solve the task optimally, with the smallest possible synaptic weights. Neither correspondence is obligatory, in that both need not hold for suboptimal networks (Fig. 11). This result suggests, but by no means proves, that the neural and behavioral alignment of similarity structure in human and monkey IT may be a consequence of each circuit finding optimal solutions to similar tasks.

While our simple neural network surprisingly captures this diversity of semantic phenomena in a mathematically tractable manner, because of its linearity, the phenomena it can capture still barely scratch the surface of semantic cognition. Some fundamental semantic phenomena that require complex nonlinear processing and mem-

ory include context dependent computations, dementia in damaged networks, theory of mind, the deduction of causal structure, and the binding of items to roles in events and situations. While it is inevitably the case that biological neural circuits exhibit all of these phenomena, it is not clear how our current generation of artificial nonlinear neural networks can recapitulate all of them. However, we hope that a deeper mathematical understanding of even the simple network presented here can serve as a springboard for the theoretical analysis of more complex neural circuits, which in turn may eventually shed much needed light on how the higher level computations of the mind can emerge from the biological wetware of the brain.

ACKNOWLEDGMENTS. S.G. thanks the Burroughs-Wellcome, Sloan, McKnight, James S. McDonnell and Simons Foundations for support. J.L.M. was supported by AFOSR. A.S. was supported by Swartz, NDSEG, & MBC Fellowships. We thank Juan Gao and Madhu Advani for useful discussions.

1. F.C. Keil. *Semantic and conceptual development: An ontological perspective*. Harvard University Press, Cambridge, MA, 1979.
2. S.E. Carey. *Conceptual Change In Childhood*. MIT Press, Cambridge, MA, 1985.
3. G.L. Murphy. *The Big Book of Concepts*. MIT, Cambridge, 2002.
4. T.T. Rogers and J.L. McClelland. *Semantic cognition: A parallel distributed processing approach*. MIT Press, Cambridge, MA, 2004.
5. J.M. Mandler and L. McDonough. Concept Formation in Infancy. *Cognitive Development*, 8:291–318, 1993.
6. B. Inhelder and J. Piaget. *The growth of logical thinking from childhood to adolescence*. Basic Books, New York, 1958.
7. R. Siegler. Three aspects of cognitive development. *Cogn Psychol*, 8:481–520, 1976.
8. E. Rosch and C.B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573–605, 1975.
9. L.W. Barsalou. Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories. *J Exp Psychol Learn Mem Cogn*, 11(4):629–654, 1985.
10. L.J. Rips, E.J. Shoben, and E.E. Smith. Semantic distance and the verification of semantic relations. *J Verbal Learning Verbal Behav*, 12:1–20, 1973.
11. G.L. Murphy and H.H. Brownell. Category differentiation in object recognition: typicality constraints on the basic category advantage. *J Exp Psychol Learn Mem Cogn*, 11(1):70–84, 1985.
12. C.B. Mervis, J. Catlin, and E. Rosch. Relationships among goodness-of-example, category norms, and word frequency. *Bull Psychon Soc*, 7(3):283–284, 1976.
13. L.J. Rips. Inductive Judgments about Natural Categories. *J Verbal Learning Verbal Behav*, 14(6):665–681, 1975.
14. D.N. Osherson, E.E. Smith, O. Wilkie, A. López, and E. Shafir. Category-based induction. *Psychological Review*, 97(2):185–200, 1990.
15. E. Rosch. Principles of Categorization. In E. Rosch and B.B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum, Hillsdale, NJ, 1978.
16. G.L. Murphy and D.L. Medin. The role of theories in conceptual coherence. *Psychol Rev*, 92(3):289–316, 1985.
17. F.C. Keil. The Emergence of Theoretical Beliefs as Constraints on Concepts. In S. Carey and R. Gelman, editors, *The Epigenesis of Mind: Essays on Biology and Cognition*. Psychology Press, 1991.
18. S. Carey. Précis of 'The Origin of Concepts'. *Behav Brain Sci*, 34(3):113–24, 2011.
19. S.A. Gelman and J.D. Coley. The Importance of Knowing a Dodo Is a Bird: Categories and Inferences in 2-Year-Old Children. *Dev Psychol*, 26(5):796–804, 1990.
20. S. Edelman. Representation is representation of similarities. *Behav Brain Sci*, 21(4):449–467, 1998.
21. N. Kriegeskorte and R.A. Kievit. Representational geometry: integrating cognition, computation, and the brain. *Trends Cogn Sci*, 17(8):401–12, 2013.
22. T.A. Carlson, R.A. Simmons, N. Kriegeskorte, and L.R. Slevc. The emergence of semantic meaning in the ventral temporal pathway. *J Cogn Neurosci*, 26(1):120–31, 2014.
23. T. Carlson, D.A. Tovar, A. Alink, and N. Kriegeskorte. Representational dynamics of object vision: The first 1000 ms. *J Vis*, 13(10):1–19, 2013.
24. B.L. Giordano, S. McAdams, R.J. Zatorre, N. Kriegeskorte, and P. Belin. Abstract encoding of auditory objects in cortical activity patterns. *Cereb Cortex*, 23(9):2025–37, 2013.
25. N. Liu, N. Kriegeskorte, M. Mur, F. Haddj-Bouziane, W.M. Luh, R.B.H. Tootell, and L.G. Ungerleider. Intrinsic structure of visual exemplar and category representations in macaque brain. *J Neurosci*, 33(28):11346–60, 2013.
26. A.C. Connolly, J.S. Guntupalli, J. Gors, M. Hanke, Y.O. Halchenko, Y.C. Wu, H. Abdi, and J.V. Haxby. The representation of biological classes in the human brain. *J Neurosci*, 32(8):2608–18, 2012.
27. M. Mur, M. Meys, J. Bodurka, R. Goebel, P.A. Bandettini, and N. Kriegeskorte. Human Object-Similarity Judgments Reflect and Transcend the Primate-IT Object Representation. *Front Psychol*, 4:128, 2013.
28. N. Kriegeskorte, M. Mur, D.A. Ruff, R. Kiani, J. Bodurka, H. Esteky, K. Tanaka, and P.A. Bandettini. Matching Categorical Object Representations in Inferior Temporal Cortex of Man and Monkey. *Neuron*, 60(6):1126–1141, 2008.
29. C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proc Natl Acad Sci USA*, 105(31):10687–92, 2008.
30. G.E. Hinton. Learning distributed representations of concepts. In R.G.M. Morris, editor, *Parallel Distributed Processing: Implications for Psychology and Neurobiology*. Oxford University Press, Oxford, UK, 1986.
31. D.E. Rumelhart and P.M. Todd. Learning and connectionist representations. In D.E. Meyer and S. Kornblum, editors, *Attention and performance XIV*. MIT Press, Cambridge, MA, 1993.
32. J.L. McClelland. A Connectionist Perspective on Knowledge and Development. In T.J. Simon and G.S. Halford, editors, *Developing cognitive competence: New approaches to process modeling*. Erlbaum, Hillsdale, NJ, 1995.
33. K. Plunkett and C. Sinha. Connectionism and developmental theory. *Br J Dev Psychol*, 10(3):209–254, 1992.
34. P.C. Quinn and M.H. Johnson. The emergence of perceptual category representations in young infants. *J Exp Child Psychol*, 66:236–263, 1997.
35. E. Colunga and L.B. Smith. From the lexicon to expectations about kinds: A role for associative learning. *Psychol Rev*, 112(2):347–382, 2005.
36. B. McMurray, J.S. Horst, and L.K. Samuelson. Word learning emerges from the interaction of online referent selection and slow associative learning. *Psychol Rev*, 119(4):831–877, 2012.
37. P. Blouw, E. Solodkin, P. Thagard, and C. Eliasmith. Concepts as semantic pointers: A framework and computational model. *Cogn Sci*, 40(5):1128–1162, 2016.
38. G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–7, 2006.
39. Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*. MIT Press, 2007.
40. 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.
41. R. Gelman. First Principles Organize Attention to and Learning About Relevant Data: Number and the Animate-Inanimate Distinction as Examples. *Cognitive Science*, 14:79–106, 1990.
42. C.B. Mervis and M.A. Crisafi. Order of Acquisition of Subordinate-, Basic-, and Superordinate-Level Categories. *Child Dev*, 53(1):258–266, 1982.
43. T. Davis and R.A. Poldrack. Quantifying the internal structure of categories using a neural typicality measure. *Cereb Cortex*, 24(7):1720–1737, 2014.
44. E. Rosch, C. Simpson, and R.S. Miller. Structural bases of typicality effects. *J Exp Psychol Hum Percept Perform*, 2(4):491–502, 1976.
45. J.M. Anglin. *Word, object, and conceptual development*. Norton, 1977.
46. E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Braem. Basic Objects in Natural Categories. *Cogn Psychol*, 8:382–439, 1976.
47. J.M. Mandler, P.J. Bauer, and L. McDonough. Separating the sheep from the goats: Differentiating global categories. *Cogn Psychol*, 23(2):263–298, 1991.
48. J.E. Corter and M.A. Gluck. Explaining basic categories: Feature predictability and information. *Psychol Bull*, 111(2):291–303, 1992.
49. A.Y. Khrennikov and S.V. Kozyrev. Wavelets on ultrametric spaces. *Appl Comput Harmon Anal*, 19(1):61–76, 2005.
50. F. Murtagh. The haar wavelet transform of a dendrogram. *J Classif*, 24(1):3–32, 2007.
51. R.M. Gray. Toeplitz and Circulant Matrices: A Review, 2005.
52. A. Laakso and G. Cottrell. Content and cluster analysis: Assessing representational similarity in neural systems. *Philosophical Psychology*, 13(1):47–76, 2000.
53. J.V. Haxby, M.I. Gobbini, M.L. Furey, A. Ishai, J.L. Schouten, and P. Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.
54. S.V. Shinkareva, V.L. Malave, M.A. Just, and T.M. Mitchell. Exploring commonalities across participants in the neural representation of objects. *Hum Brain Mapp*, 33(6):1375–1383, 2012.
55. R.D.S. Raizada and A.C. Connolly. What Makes Different People's Representations Alike: Neural Similarity Space Solves the Problem of Across-subject fMRI Decoding. *J Cogn Neurosci*, 24(4):868–877, 2012.

Supplementary Material

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 $\{\mathbf{x}^i, \mathbf{y}^i\}$, $i = 1, \dots, P$, 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} in the case of the shallow network), computes its output as

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

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

$$SSE(\mathbf{W}^1, \mathbf{W}^2) = \frac{1}{2} \|\mathbf{y}^i - \hat{\mathbf{y}}^i\|^2, \quad [\text{S2}]$$

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,

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

where $\mathbf{h}^i = \mathbf{W}^1 \mathbf{x}^i$ is the hidden layer activity for example i .

This update is identical to that produced by the standard back-propagation 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 $\delta^i = \mathbf{W}^{2T} \mathbf{e}^i$, 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 $i = 1, \dots, P$ 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

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

where $\Sigma^x \equiv E[\mathbf{x} \mathbf{x}^T]$ is an $N_1 \times N_1$ input correlation matrix, and $\Sigma^{yx} \equiv E[\mathbf{y} \mathbf{x}^T]$ 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,

$$\tau \frac{d}{dt} \mathbf{W}^1 = \mathbf{W}^{2T} (\Sigma^{yx} - \mathbf{W}^2 \mathbf{W}^1 \Sigma^x), \quad [\text{S3}]$$

$$\tau \frac{d}{dt} \mathbf{W}^2 = (\Sigma^{yx} - \mathbf{W}^2 \mathbf{W}^1 \Sigma^x) \mathbf{W}^{1T}. \quad [\text{S4}]$$

where the time constant

$$\tau \equiv \frac{1}{P\lambda}. \quad [\text{S5}]$$

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, we decompose the input-output correlations through the singular value decomposition (SVD),

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

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

$$\mathbf{W}^1 = \mathbf{R} \bar{\mathbf{W}}^1 \mathbf{V}^T, \quad [\text{S6}]$$

$$\mathbf{W}^2 = \mathbf{U} \bar{\mathbf{W}}^2 \mathbf{R}^T, \quad [\text{S7}]$$

and \mathbf{R} is an arbitrary orthogonal matrix ($\mathbf{R}^T \mathbf{R} = \mathbf{I}$). These variables analyze the dynamics in the basis defined by the SVD. Substituting into Eqns. (S3)-(S4) and using the simplifying assumption $\Sigma^x = \mathbf{I}$ we have

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

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 $\bar{\mathbf{W}}^1$ and $\bar{\mathbf{W}}^2$ are also diagonal, the dynamics decouple into N_1 independent systems. We study the dynamics in this decoupled regime where $\bar{\mathbf{W}}^1(0)$ and $\bar{\mathbf{W}}^2(0)$ 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 $\bar{\mathbf{W}}^1(0)$ and $\bar{\mathbf{W}}^2(0)$ are initialized with small random weights, as shown through simulation (red lines in Fig. 3C of the main text).

In particular, let $c_\alpha = \bar{\mathbf{W}}^1_{\alpha\alpha}$ and $d_\alpha = \bar{\mathbf{W}}^2_{\alpha\alpha}$ 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 $\alpha = 1, \dots, N_1$. 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),

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

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 $\mathbf{A}(t) = \bar{\mathbf{W}}^2(t) \bar{\mathbf{W}}^1(t)$. 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:

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

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 $\bar{\mathbf{W}}^1, \bar{\mathbf{W}}^2 \ll 1$). 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 $\bar{\mathbf{W}}^1, \bar{\mathbf{W}}^2$).

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

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

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

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

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

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

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

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

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

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 $\mathbf{W}^s = \mathbf{U} \bar{\mathbf{W}}^s \mathbf{V}^T$,

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

Defining $\bar{\mathbf{W}}_{\alpha\alpha}^s = b_\alpha$ 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

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

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. The input-output correlations are

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

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 $\epsilon_0 \rightarrow 0$, the ratio of the length of time in the transition period to the total training time is

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

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 $\epsilon_0 \rightarrow 0$, indicating that the transition period is as long as the total training time and transitions are not stage-like.

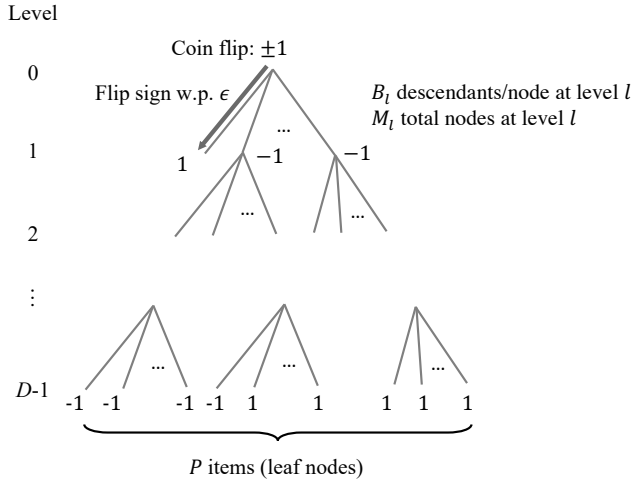


Fig. S1. Generating hierarchically structured data through a branching diffusion process. To generate a feature, an initial binary value is determined through a coin flip at the top of the hierarchy. The sign of this value flips with a small probability along each link in the tree. At the bottom, this 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.

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 Σ^{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}^α of Σ^{yx} , and then expand this component of the internal representation into a contribution to the predicted feature output vector given by \mathbf{u}^α , a left singular vector of Σ^{yx} .

To understand the time course of learning of hierarchical structure, we analyze a simple generative model proposed in [2] 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}^μ , $\mu = 1, \dots, P$. Each \mathbf{y}^μ is an N_3 -dimensional feature vector where each feature i in example μ takes the value $y_i^\mu = \pm 1$. The value of each feature i across all examples arises from a branching diffusion process occurring on a tree, as depicted in Fig. S1. 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 $\mu = 1, \dots, P$, which are in one-to-one correspondence with the leaves of the tree.

The tree has a fixed topology, with D levels indexed by $l = 0, \dots, D-1$, 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 $M_l = M_0 \prod_{k=0}^{l-1} B_k$. 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 $i = 1, \dots, B_0$. 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 y_i^μ for $\mu = 1, \dots, P$. 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 $N_3 \rightarrow \infty$. 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

$$q_k = \frac{1}{N_3} \sum_{i=1}^{N_3} y_i^{\mu_1} y_i^{\mu_2},$$

where again, the first common ancestor of leaves μ_1 and μ_2 arises at level k . It is the case that $1 = q_{D-1} > q_{D-2} > \dots > q_0 > 0$. 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, \dots, q_{D-2} , along with the number of nodes at each level M_0, \dots, 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

$$\Sigma_{\mu_1 \mu_2} = \frac{1}{N_3} \sum_{i=1}^{N_3} y_i^{\mu_1} y_i^{\mu_2}, \quad [\text{S12}]$$

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 $q_{D-2} > q_{D-1} > \dots > q_0$. The explicit formula for q_k is

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

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 Σ^{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 Σ^{yx} is then an $N_3 \times P$ matrix with elements $\Sigma_{i\mu}^{yx} = y_i^\mu$, with $i = 1, \dots, N_3$ indexing feature components, and $\mu = 1, \dots, P$ indexing examples. We note that

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

where Σ , defined in (S12), is the correlation matrix across examples. From this we see that the eigenvectors of Σ are the same as the right singular vectors \mathbf{v}^α of Σ^{yx} , and if the associated eigenvalue of Σ is λ_α , then the associated singular value of Σ^{yx} is $s_\alpha = \sqrt{N_3 \lambda_\alpha}$. Thus finding the singular values s_α of Σ^{yx} , which determine the time scales of learning, reduces to finding the eigenvalues λ_α of Σ .

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}^μ form an orthonormal basis O for the space of input-layer activity patterns. This would yield the modification $\Sigma^{yx} \rightarrow \Sigma^{yx} O^T$, which would not change the singular values s_α at all, but would simply rotate the singular vectors, \mathbf{v}^α . 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}^α of the correlation matrix across examples, Σ in (S12). 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}^μ , for $\mu = 1, \dots, P$. 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 function $f(\mu)$ 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 Σ 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

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

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 multi-

layer 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 Σ^{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

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

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 $\Sigma^{yx} = E[\mathbf{y}\mathbf{x}^T] = \frac{1}{P}\mathbf{O}$. The dynamics of learning are thus driven by the singular value decomposition of \mathbf{O} ,

$$\frac{1}{P}\mathbf{O} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \quad [\text{S13}]$$

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 i th row and the α th column of \mathbf{V} on both sides, we obtain Eqn. (13) of the main text,

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

Similarly, multiplying Eqn. (S13) from the right by $\mathbf{V}\mathbf{S}^{-1}$ yields,

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

Extracting the elements at the i th row and α th column yields Eqn. (14) of the main text,

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

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, $N_f, N_o \rightarrow \infty$, but their ratio is constant, $N_o/N_f \rightarrow c \in (0, 1]$.

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

$$\tilde{\mathbf{R}} = \frac{1}{\sqrt{N_f q(1-q)}}(\mathbf{R} - q\mathbf{1}\mathbf{1}^T) \quad [\text{S14}]$$

where we have used the fact that $E[y_i] = q$ and $\text{Var}[y_i] = q(1-q)$ 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,

$$\tilde{\mathbf{R}} \approx \mathbf{X} + \mathbf{P}. \quad [\text{S15}]$$

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

$$\mathbf{P} = \theta \frac{1}{\sqrt{K_f K_o}} \mathbf{1}_{C_f} \mathbf{1}_{C_i}^T \quad [\text{S16}]$$

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

$$\theta = \frac{(p-q)\sqrt{K_f K_o}}{\sqrt{N_f q(1-q)}}. \quad [\text{S17}]$$

To understand learning dynamics in this setting, we must compute the typical singular values and vectors of $\tilde{\mathbf{R}}$. Theorem 2.9 of [3] states that recovery of the correct singular vector structure only occurs for signal strengths above a threshold (an instance of the BBP phase transition [4]). 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

$$\mathbf{u}^{\text{Ideal}} = \frac{1}{\sqrt{K_f}} \mathbf{1}_{C_f}, \quad [\text{S18}]$$

$$\mathbf{v}^{\text{Ideal}} = \frac{1}{\sqrt{K_o}} \mathbf{1}_{C_o} \quad [\text{S19}]$$

be the ground truth feature and object analyzers arising from the category structure in Eqn. (S16). Then

$$\left(\tilde{\mathbf{u}}^T \mathbf{u}^{\text{Ideal}} \right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{c+\theta^2}{\theta^2(\theta^2+1)} & \text{for } \theta > c^{1/4} \\ 0 & \text{otherwise} \end{cases} \quad [\text{S20}]$$

$$\left(\tilde{\mathbf{v}}^T \mathbf{v}^{\text{Ideal}} \right)^2 \xrightarrow{a.s.} \begin{cases} 1 - \frac{c(1+\theta^2)}{\theta^2(\theta^2+c)} & \text{for } \theta > c^{1/4} \\ 0 & \text{otherwise} \end{cases} \quad [\text{S21}]$$

where *a.s.* denotes almost sure convergence (i.e., with probability 1) in the high-dimensional limit ($N_f, N_o \rightarrow \infty$ 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

$$\frac{(p-q)^2 K_f K_o}{q(1-q)\sqrt{N_f N_o}} > 1. \quad [\text{S22}]$$

This motivates defining category coherence as

$$\mathcal{C} \equiv \frac{(p-q)^2 K_f K_o}{q(1-q)\sqrt{N_f N_o}} \quad [\text{S23}]$$

$$= \text{SNR} \frac{K_f K_o}{\sqrt{N_f N_o}} \quad [\text{S24}]$$

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. (S20)-(S21) can be written as

$$\left(\tilde{\mathbf{u}}^T \mathbf{u}^{\text{Ideal}}\right)^2 \xrightarrow{\text{a.s.}} \begin{cases} 1 - \frac{1+c^{-1/2}\mathcal{C}}{c(\mathcal{C}+c^{-1/2})} & \text{for } \mathcal{C} > 1 \\ 0 & \text{otherwise} \end{cases} \quad [\text{S25}]$$

$$\left(\tilde{\mathbf{v}}^T \mathbf{v}^{\text{Ideal}}\right)^2 \xrightarrow{\text{a.s.}} \begin{cases} 1 - \frac{1+c^{1/2}\mathcal{C}}{c(\mathcal{C}+c^{1/2})} & \text{for } \mathcal{C} > 1 \\ 0 & \text{otherwise} \end{cases} \quad [\text{S26}]$$

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 Σ^{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 Σ^{yx} . The singular values are the square root of the eigenvalues of the item similarity matrix,

$$\Sigma^{yxT} \Sigma^{yx} \equiv \Sigma^y, \quad [\text{S27}]$$

and the object analyzer vectors $\mathbf{v}^\alpha, \alpha = 1, \dots, P$ are the eigenvectors of Σ^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

$$\Sigma^y = \sum_{\alpha=1}^P \lambda_\alpha \mathbf{v}^\alpha \mathbf{v}^{\alpha T}. \quad [\text{S28}]$$

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

$$q_k = \frac{1}{N_3} \sum_{i=1}^{N_3} \mathbf{y}_i^{\mu_1} \mathbf{y}_i^{\mu_2}, \quad [\text{S29}]$$

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

$$\lambda_k = \sum_{j \in C} \Sigma_{ij}^y - \sum_{j \in S(C)} \Sigma_{ij}^y \quad \text{for any } i \in C \quad [\text{S30}]$$

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. (S30) 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 $\hat{\mathbf{O}} = \mathbf{U} \tilde{\mathbf{S}} \mathbf{V}^T$, where $\tilde{\mathbf{S}}$ contains just the first k singular values with the remaining diagonal elements set to zero (that is, $\tilde{S}_{ii} = S_{ii}$ for $i \leq k$ and $\tilde{S}_{ii} = 0$ otherwise). The Eckart–Young–Mirsky theorem states that $\hat{\mathbf{O}}$ is a solution to

$$\min_{\mathbf{B}, \text{rank}(\mathbf{B}) \leq k} \|\mathbf{O} - \mathbf{B}\|_F. \quad [\text{S31}]$$

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 $\mathbf{W}^2 \mathbf{W}^1 = \tilde{\mathbf{O}} = \mathbf{U} \tilde{\mathbf{S}} \mathbf{V}^T$ 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

$$SSE(\tilde{\mathbf{W}}^1, \tilde{\mathbf{W}}^2) = \frac{1}{2} \|\mathbf{U}(\mathbf{S} - \tilde{\mathbf{S}})\mathbf{V}\|_F^2 = \frac{1}{2} \sum_{i=k+1}^{N_1} s_i(\mathbf{O})^2. \quad [\text{S32}]$$

where here and in the following we will denote the i th 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

$$s_{i+k}(\mathbf{O}) \leq s_i(\mathbf{O} - \mathbf{W}^2 \mathbf{W}^1) + s_{k+1}(\mathbf{W}^2 \mathbf{W}^1) \quad [\text{S33}]$$

$$\leq s_i(\mathbf{O} - \mathbf{W}^2 \mathbf{W}^1) \quad [\text{S34}]$$

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

$$\begin{aligned} \frac{1}{2} \|\mathbf{O} - \mathbf{W}^2 \mathbf{W}^1\|_F^2 &= \frac{1}{2} \sum_{i=1}^{N_1} s_i(\mathbf{O} - \mathbf{W}^2 \mathbf{W}^1)^2 \\ &\geq \frac{1}{2} \sum_{i=1}^{N_1-k} s_i(\mathbf{O} - \mathbf{W}^2 \mathbf{W}^1)^2 \quad [\text{S35}] \\ &\geq \frac{1}{2} \sum_{i=k+1}^{N_1} s_i(\mathbf{O})^2 \quad [\text{S36}] \\ &= \frac{1}{2} \|\mathbf{O} - \tilde{\mathbf{W}}^2 \tilde{\mathbf{W}}^1\|_F^2, \end{aligned}$$

where from (S35)-(S36) we have used (S33)-(S34) and the last equality follows from Eqn. (S32). Hence

$$SSE(\tilde{\mathbf{W}}^1, \tilde{\mathbf{W}}^2) \leq SSE(\mathbf{W}^1, \mathbf{W}^2) \quad [\text{S37}]$$

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. S2. 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.

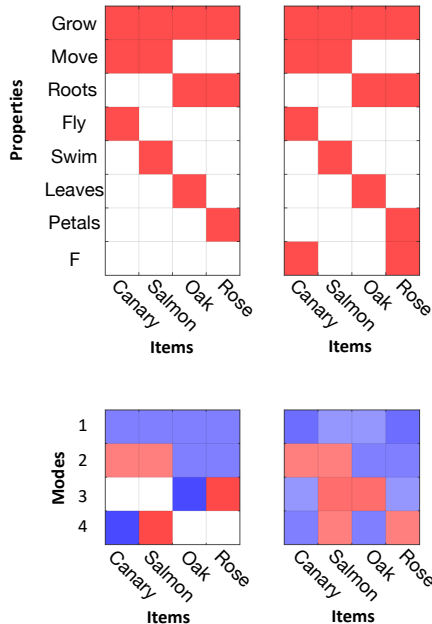


Fig. S2. Category structure is a nonlocal and nonlinear function of the features. Left column: a toy dataset with hierarchical structure (top) has object analyzer vectors that mirror the hierarchy (bottom). Right column: Adding a new feature F to the dataset (top) causes a substantial change to the category 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.

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 [5] 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

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

where $\tilde{\mathbf{f}}$ is a length K vector of feature values for each node, and $\tilde{\Phi} = \mathbf{L} + 1/\sigma^2 \mathbf{I}$ 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 $M_{ij} = 1$ 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 $\Phi^{-1} = \mathbf{M}\tilde{\Phi}^{-1}\mathbf{M}^T$,

$$\mathbf{f} \sim \mathcal{N}(0, \Phi^{-1}). \quad [\text{S38}]$$

To obtain multiple features, we assume that features are drawn independently according to Eq. (S38).

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

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

Now we note that

$$\begin{aligned} \Sigma_{ij}^y &= \frac{1}{P^2} \mathbf{y}^{iT} \mathbf{y}^j \\ &= \frac{1}{P^2} \sum_{m=1}^{N_3} \mathbf{y}_m^i \mathbf{y}_m^j \\ &= \frac{N_3}{P^2} \left(\frac{1}{N_3} \sum_{m=1}^{N_3} \mathbf{y}_m^i \mathbf{y}_m^j \right). \end{aligned}$$

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

$$\Sigma_{ij}^y = \frac{N_3}{P^2} E[\mathbf{f}^i \mathbf{f}^j]$$

and hence from Eq. (S38),

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

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 Φ 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 Φ .

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 $\Phi_b^{-1} = c_1^b \mathbf{1} + c_2^b \mathbf{I}$ where $\mathbf{1} \in \mathbb{R}^{M_b \times M_b}$ is a constant matrix of ones, \mathbf{I} is an identity matrix, $b = 1, \dots, N_c$ 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 $e_b \rightarrow 0$, 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 $s_1 - s_2 = \frac{2\sigma^6}{3\sigma^4 + 4\sigma^2 e_b + e_b^2}$ which is decreasing in e_b . The minimum along the boundary would thus occur at $M_b = 2, e_b \rightarrow 0$, 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 Σ^y . As shown in [6], 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 [7].

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 [8], 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 [8]. Our example is given in a 1D ring, but the same structure arises for higher dimensional structure (yielding, for instance, doubly block circulant structure in a 2D ring [8, 9] 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 [5, 10]). To instantiate this, we use the input-output correlations

$$\Sigma^{yx} = \frac{1}{P} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad [\text{S39}]$$

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 [11]. 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

$$\Sigma^{yx} = \frac{1}{P} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1.1 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 1.1 \\ 1.1 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.1 & 1.1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \quad [\text{S40}]$$

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 [12], 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 [13, 14] for a discussion of the catastrophic interference that arises from rapid non-interleaved learning). This yields the weight update

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

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 ($\mathbf{w}_m^{2T}(0) = 0$), they converge to

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

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

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

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 [12]. Define \mathbf{h}_m to be the transpose of the m th 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 i th 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

$$\mathbf{W}^1(t) = \mathbf{Q} \sqrt{\mathbf{A}(t)} \mathbf{V}^T, \quad [\text{S41}]$$

$$\mathbf{W}^2(t) = \mathbf{U} \sqrt{\mathbf{A}(t)} \mathbf{Q}^{-1} \quad [\text{S42}]$$

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

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

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

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

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

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

To solve this we introduce Lagrange multipliers Λ and form the Lagrangian

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

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{P}} &= 2\mathbf{A}^2\mathbf{P} + \mathbf{\Lambda}\mathbf{Q}^T = 0 \\ \frac{\partial \mathcal{L}}{\partial \mathbf{Q}} &= 2\mathbf{Q}\mathbf{A}^2 + \mathbf{P}^T\mathbf{\Lambda} = 0 \\ \Rightarrow \mathbf{\Lambda} &= -2\mathbf{A}^2\mathbf{P}\mathbf{Q}^{-T} = -2\mathbf{P}^{-T}\mathbf{Q}\mathbf{A}^2.\end{aligned}$$
$$\begin{aligned} -2\mathbf{A}^2\mathbf{P}\mathbf{Q}^{-T} &= -2\mathbf{P}^{-T}\mathbf{Q}\mathbf{A}^2 \\ \mathbf{A}^2\mathbf{P}\mathbf{P}^T &= (\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{A}^2 \\ \mathbf{S}\mathbf{M} &= \mathbf{M}^{-1}\mathbf{S} \end{aligned} \quad [\text{A43}]$$
$$\begin{aligned} \mathbf{W}^2 &= \mathbf{U}\tilde{\mathbf{A}}\tilde{\mathbf{V}}^T = \mathbf{U}\mathbf{A}\mathbf{P} \\ \Rightarrow \mathbf{P} &= \mathbf{A}^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{V}}^T \\ &= \mathbf{D}\tilde{\mathbf{V}}^T \end{aligned}$$
$$\begin{aligned} \mathbf{MS} &= \mathbf{M}^{-1}\mathbf{S} \\ \mathbf{M}^2\mathbf{S} &= \mathbf{S} \end{aligned}$$

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

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

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

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

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 .

$$\begin{aligned} \Sigma^{yx} &= 0.7P \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \Sigma^x &= \mathbf{I}. \end{aligned}$$
$$\begin{aligned}\mathbf{W}^1(0)_{ij} &\sim \mathcal{N}(0, a_0^2/N_1) \\ \mathbf{W}^2(0)_{ij} &\sim \mathcal{N}(0, a_0^2/N_3).\end{aligned}$$

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. 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.
3. 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.
4. J. Baik, G.B. Arous, and S. Pécché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Ann Probab*, 33(5):1643–1697, 2005.

5. C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proc Natl Acad Sci USA*, 105(31):10687–92, August 2008.
6. A.Y. Khrennikov and S.V. Kozyrev. Wavelets on ultrametric spaces. *Appl Comput Harmon A*, 19:61–76, 2005.
7. F. Murtagh. The haar wavelet transform of a dendrogram. *J Classif*, 24(1):3–32, 2007.
8. R.M. Gray. Toeplitz and Circulant Matrices: A Review. *Found Trends Commun Inf Theor*, 2(3):155–239, 2005.

9. G.J. Tee. Eigenvectors of block circulant and alternating circulant matrices. *Res Lett Inf Math Sci*, 8:123–142, 2005.
10. C. Kemp and J.B. Tenenbaum. Structured statistical models of inductive reasoning. *Psychol Rev*, 116(1):20–58, 2009.
11. 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.
12. T.T. Rogers and J.L. McClelland. *Semantic cognition: A parallel distributed processing approach*. MIT Press, Cambridge, MA, 2004.
13. 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.
14. 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.