Learning hierarchical categories in deep neural networks

Andrew M. Saxe (asaxe@stanford.edu)
Department of Electrical Engineering
James L. McClelland (mcclelland@stanford.edu)
Department of Psychology
Surya Ganguli (sganguli@stanford.edu)
Department of Applied Physics
Stanford University, Stanford, CA 94305 USA

Abstract

A wide array of psychology experiments have revealed remarkable regularities in the developmental time course of human cognition. For example, infants generally acquire broad categorical distinctions (i.e., plant/animal) before finer-scale distinctions (i.e., dog/cat), often exhibiting rapid, or stage-like transitions. What are the theoretical principles underlying the ability of neuronal networks to discover categorical structure from experience? We develop a mathematical theory of hierarchical category learning through an analysis of the learning dynamics of multilayer networks exposed to hierarchically structured data. Our theory yields new exact solutions to the nonlinear dynamics of error correcting learning in deep, three layer networks. These solutions reveal that networks learn input-output covariation structure on a time scale that is inversely proportional to its statistical strength. We further analyze the covariance structure of data sampled from hierarchical probabilistic generative models, and show how such models yield a hierarchy of input-output modes of differing statistical strength, leading to a hierarchy of time-scales over which such modes are learned. Our results reveal that even the second order statistics of hierarchically structured data contain powerful statistical signals sufficient to drive complex experimentally observed phenomena in semantic development, including progressive, coarse-to-fine differentiation of concepts and sudden, stage-like transitions in performance punctuating longer dormant periods.

Keywords: neural networks; hierarchical generative models; semantic cognition; learning dynamics

Introduction

Our world is characterized by a rich, nested hierarchical structure of categories within categories, and one of the most remarkable aspects of human semantic development is our ability to learn and exploit this rich structure. Experimental work has shown that infants and children acquire broad categorical distinctions before fine categorical distinctions (Keil, 1979; Mandler & McDonough, 1993), suggesting that human category learning is marked by a progressive differentiation of concepts from broad to fine. Furthermore, humans can exhibit stage-like transitions as they learn, rapidly moving from ignorance to mastery (Inhelder & Piaget, 1958; Siegler, 1976).

Many neural network simulations have captured aspects of these broad patterns of semantic development (Rogers & McClelland, 2004; Rumelhart & Todd, 1993; McClelland, 1995; Plunkett & Sinha, 1992; Quinn & Johnson, 1997). The internal representations of such networks exhibit both progressive differentiation and stage like transitions.

However the theoretical basis for the ability of neuronal networks to exhibit such strikingly rich nonlinear behavior remains elusive. What are the essential principles that underly such behavior? What aspects of statistical structure in the input are responsible for driving such dynamics? For example, must networks exploit nonlinearities in their input-output map to detect higher order statistical regularities to drive such learning?

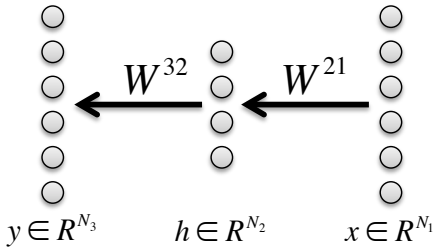

Figure 1: The three layer network analyzed in this work.

Here we analyze the learning dynamics of a *linear* 3 layer network and find, surprisingly, that it can exhibit highly nonlinear learning dynamics, including rapid stage-like transitions. Furthermore, when exposed to hierarchically structured data sampled from a hierarchical probabilistic model, the network exhibits progressive differentiation of concepts from broad to fine. Since such linear networks are sensitive only to the second order statistics of inputs and outputs, this yields the intriguing result that merely second order patterns of covariation in hierarchically structured data contain statistical signals powerful enough to drive certain nontrivial, high level aspects of semantic development in deep networks.

Gradient descent dynamics in multilayer neural networks

We examine learning in a three layer network (input layer 1, hidden layer 2, and output layer 3) with linear activation functions, simplifying the network model of Rumelhart and Todd (1993), in which input units correspond to items e.g, *Canary, Rose* and output units correspond to possible predicates or attributes *Can Fly, Has Petals* that may or may not apply to each item. Let N_i be the number of neurons in layer i, W^{21} be an $N_2 \times N_1$ matrix of synaptic connections from layer 1 to 2, and similarly, W^{32} an $N_3 \times N_2$ matrix of connections from layer 2 to 3. The input-output map of the network is $y = W^{32}W^{21}x$, where x is an N_1 dimensional column vector representing inputs to the network, and y is an N_2 dimensional column vector representing the network output (see Fig. 1).

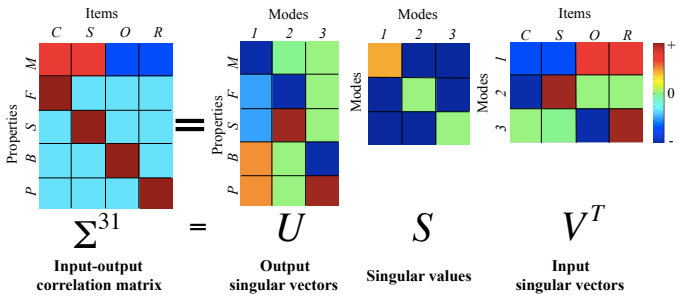

Figure 2: Example singular value decomposition for a toy dataset. Left: The learning environment is specified by an input-output correlation matrix. This example dataset has four items: *Canary, Salmon, Oak, and Rose.* The two animals share the property that they can *Move*, while the two plants cannot. In addition each item has a unique property: can *Fly*, can *Swim*, has *Bark*, and has *Petals*, respectively. Right: The SVD decomposes Σ^{31} into input-output *modes* that link a set of coherently covarying properties (*output singular vectors* in the columns of U) to a set of coherently covarying items (*input singular vectors* in the rows of V^T). The overall strength of this link is given by the *singular values* lying along the diagonal of S. In this toy example, mode 1 distinguishes plants from animals; mode 2 birds from fish; and mode 3 flowers from trees.

We wish to train the network to learn a particular input-output map from a set of P training examples $\{x^\mu, y^\mu\}, \mu = 1, \ldots, P$. The input vector x^μ, identifies item μ while each y^μ is a set of attributes to be associated to this item. Training is accomplished in an online fashion via stochastic gradient descent; each time an example μ is presented, the weights W^{32} and W^{21} are adjusted by a small amount in the direction that minimizes the squared error $\left\|y^\mu - W^{32}W^{21}x^\mu\right\|^2$ between the desired feature output, and the network's feature output. This gradient descent procedure yields the learning rule

$$\Delta W^{21} = \lambda W^{32^T}\left(y^\mu x^{\mu T} - W^{32}W^{21}x^\mu x^{\mu T}\right) \quad (1)$$

$$\Delta W^{32} = \lambda\left(y^\mu x^{\mu T} - W^{32}W^{21}x^\mu x^{\mu T}\right)W^{21^T}, \quad (2)$$

for each example μ, where λ is a small learning rate. We imagine that training is divided into a sequence of learning epochs, and in each epoch, the above rules are followed for all P examples in random order. As long as λ is sufficiently small so that the weights change by only a small amount per learning epoch, we can average (1)-(2) over all P examples and take a continuous time limit to obtain the mean change in weights per learning epoch,

$$\tau\frac{d}{dt}W^{21} = W^{32^T}\left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right) \quad (3)$$

$$\tau\frac{d}{dt}W^{32} = \left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right)W^{21^T}, \quad (4)$$

where $\Sigma^{11} \equiv E[xx^T]$ is an $N_1 \times N_1$ input correlation matrix,

$$\Sigma^{31} \equiv E[yx^T] \quad (5)$$

is an $N_3 \times N_1$ input-output correlation matrix, and $\tau \equiv \frac{P}{\lambda}$. 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. We note that, although the network we analyze is completely linear with the simple input-output map $y = W^{32}W^{21}x$, the gradient descent learning dynamics given in Eqns. (3)-(4) are highly nonlinear.

Decomposing the input-output correlations Our fundamental goal is to understand the dynamics of learning in (3)-(4) as a function of the input statistics Σ^{11} and Σ^{31}. In general, the outcome of learning will reflect an interplay between the perceptual correlations in the input patterns, described by Σ^{11}, and the input-output correlations described by Σ^{31}. To begin, though, we consider the case of orthogonal input representations where each item is designated by a single active input unit, as used by (Rumelhart & Todd, 1993) and (Rogers & McClelland, 2004). In this case, Σ^{11} corresponds to the identity matrix. Under this scenario, the only aspect of the training examples that drives learning is the second order input-output correlation matrix Σ^{31}. We consider its singular value decomposition (SVD)

$$\Sigma^{31} = U^{33}S^{31}V^{11^T} = \sum_{\alpha=1}^{N_1} s_\alpha u^\alpha v^{\alpha T}, \quad (6)$$

which will play a central role in understanding how the examples drive learning. The SVD decomposes any rectangular matrix into the product of three matrices. Here V^{11} is an $N_1 \times N_1$ orthogonal matrix whose columns contain *input-analyzing* singular vectors v^α that reflect independent modes of variation in the input, U^{33} is an $N_3 \times N_3$ orthogonal matrix whose columns contain *output-analyzing* singular vectors u^α that reflect independent modes of variation in the output, and S^{31} is an $N_3 \times N_1$ matrix whose only nonzero elements are on the diagonal; these elements are the singular values $s_\alpha, \alpha = 1, \ldots, N_1$ ordered so that $s_1 \geq s_2 \geq \cdots \geq s_{N_1}$. An example SVD of a toy dataset is given in Fig. 2. As can be seen, the SVD extracts coherently covarying items and properties from this dataset, with various modes picking out the underlying hierarchy present in the toy environment.

The temporal dynamics of learning A central result of this work is that we have described the full time course of learning by solving the nonlinear dynamical equations (3)-(4) for orthogonal input representations ($\Sigma^{11} = I$), and arbitrary input-output correlation Σ^{31}. In particular, we find a class of exact solutions (whose derivation will be presented elsewhere) for $W^{21}(t)$ and $W^{32}(t)$ such that the composite mapping at any time t is given by

$$W^{32}(t)W^{21}(t) = \sum_{\alpha=1}^{N_2} a(t, s_\alpha, a_\alpha^0)\, u^\alpha v^{\alpha T}, \quad (7)$$

where the function $a(t, s, a^0)$ governing the strength of each input-output mode is given by

$$a(t, s, a_0) = \frac{s e^{2st/\tau}}{e^{2st/\tau} - 1 + s/a_0}. \quad (8)$$

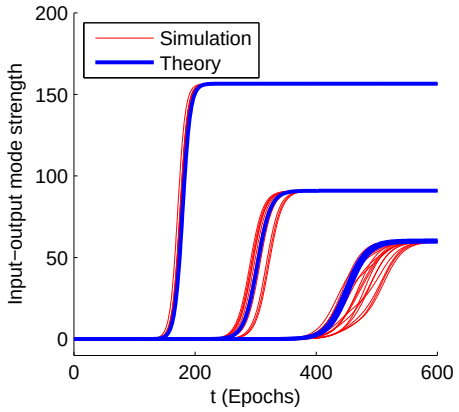

Figure 3: Close agreement between theoretically predicted time course and numerical simulations. Simulations were performed with a dataset sampled from the hierarchical diffusion process described in detail in a later section, with $D = 3$ hierarchical levels, binary branching, flip probability $\varepsilon = 0.1$, and $N = 10,000$ sampled features. This data set had 3 unique singular values. Red traces show ten simulations of the singular value dynamics of $W^{32}(t)W^{21}(t)$ in Eqns. (3)-(4) starting from different random initializations, and blue traces show theoretical curves obtained from (8).

As can be seen from Fig. 3, for $a_0 < s$, this function is a sigmoidal curve that starts at a_0 when $t = 0$, and asymptotically rises to s as $t \to \infty$. Thus for small initial conditions a^0_α, the weight trajectory (7) describes an evolving network whose input-output mapping successively builds up the first N_2 modes of the SVD of Σ^{31} in (6). This result is the solution to (3)-(4) for a special class of initial conditions on the weights W^{21} and W^{32}. However this analytic solution is a good approximation to the time evolution the network's input-output map for random small initial conditions, as confirmed in Fig. 3.

Eqns. (7)-(8) reveal a number of important properties of the learning dynamics. What is the final outcome of learning? As $t \to \infty$, the weight matrices converge to the best rank N_2 approximation of Σ^{31}.

More importantly, what is the timescale of learning? Each pair of output (u^α) and input (v^α) modes are learned in (7) on a different time scale, governed by the singular value s_α. To estimate this time scale, we can assume a small initial condition $a_0 = \varepsilon$ and ask when $a(t)$ in (8) is within ε of the final value s, i.e. $a(t) = s - \varepsilon$; then the timescale of learning in the limit $\varepsilon \to 0$ is

$$t(s,\varepsilon) = \frac{\tau}{s} \ln \frac{s}{\varepsilon}. \tag{9}$$

This is $O(\tau/s)$ up to a logarithmic factor. Thus the time required to learn an input-output mode is inversely related to its statistical strength, quantified through its singular value.

Finally, these dynamics reveal stage-like transitions in learning performance. Intuitively, this property arises from the sigmoidal transition in (8) from a state in which the network does not represent a particular input-output relation at all, to a state in which the network fully incorporates that relation. To formalize this, we begin with the sigmoidal learning curve in (8). If we assume the initial strength of the mode a_0 satisfies $a_0 < s/2$, where s is its final learned value, we can define the *transition time* to be the time at which the mode is half learned (i.e. $a(t_{half}) = s/2$). This yields

$$t_{half} = \frac{\tau}{2s} \log\left(\frac{s}{a_0} - 1\right). \tag{10}$$

We can then define the transition period t_{trans} as the time required for a linear approximation to $a(t,s,a^0)$ at t_{half} to rise from zero to s. This yields a transition time to go from a state of no learning to almost full learning given by $t_{trans} = \frac{2\tau}{s}$. Thus, by starting with a very small initial condition in the weights (i.e. a_0), it is clear that one can make the ratio t_{trans}/t_{half} arbitrarily small. Hence the learning dynamics of (3)-(4) can indeed exhibit sharp stage-like transitions consisting of long periods of dormancy ended by an abrupt transition to mastery. Interestingly, we can prove that single layer networks are not capable of such stage-like transitions. Thus their existence is an emergent property of nonlinear learning dynamics in deep networks with at least one hidden layer, and does not require nonlinearity in the input-output map of the network.

Summary of learning dynamics The preceding analyses have established a number of crucial features of gradient descent learning in a simple linear network, making explicit the relationship between the statistical structure of training examples and the dynamics of learning. In particular, for an arbitrary input-output task the network will ultimately come to represent the closest rank N_2 approximation to the full input-output correlation matrix. Furthermore, the learning dynamics depend crucially on the singular values of the input-output correlation matrix. Each input-output mode is learned in time inversely proportional to its associated singular value, yielding the intuitive result that stronger input-output associations are learned before weaker ones.

The singular values and vectors of hierarchically generated data

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 neural network. Using this, we will be able to explicitly link hierarchical taxonomies of categories to the dynamics of network learning. A key result in the following is that our network must exhibit progressive differentiation with respect to any of the underlying hierarchical taxonomies allowed by our generative model.

Hierarchical feature vectors from a branching diffusion process To understand the time course of learning of hierarchical structure, we propose a simple generative model of hierarchical data $\{x^\mu, y^\mu\}$, and compute for this model the statistical properties $(s_\alpha, u^\alpha, v^\alpha)$ which drive learning. We first

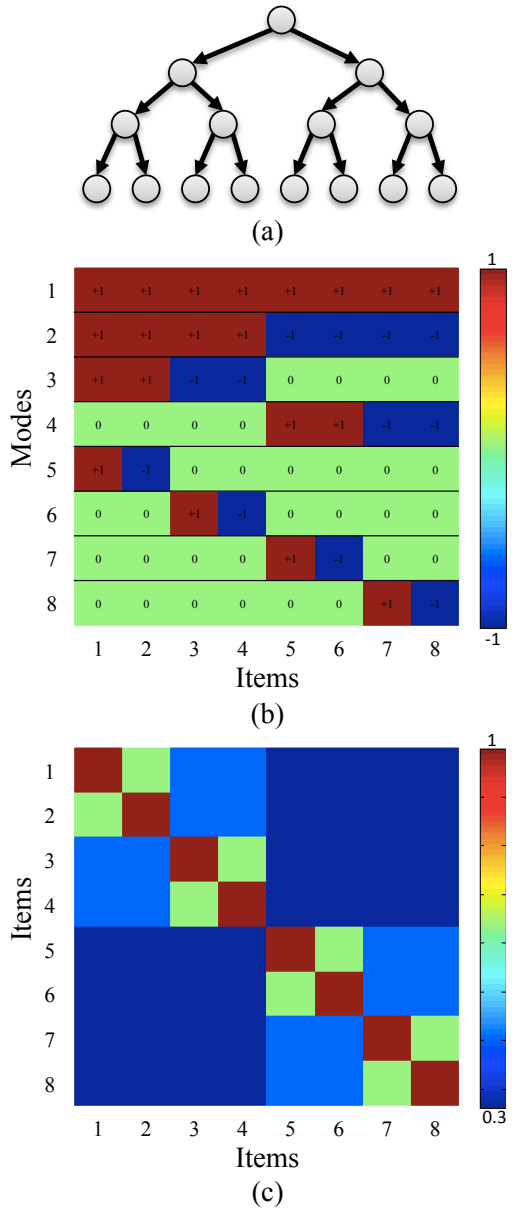

Figure 4: Statistical structure of hierarchical data. (a) Example hierarchical diffusion process with $D = 4$ levels and branching factor $B = 2$. To sample one feature's value across items, the root node is randomly set to ± 1; next this value diffuses to children nodes, where its sign is flipped with a small probability ε. The leaf node assignments yield the value of this feature on each item. To generate more features, the process is repeated independently N times. (b) Analytically derived input singular vectors (up to a scaling) of the resulting data, ordered top-to-bottom by singular value. Mode 1 is a level 0 function on the tree, mode 2 is level 1, 3 and 4 are level 2, while modes 5 through 8 are level 3. Singular modes corresponding to broad distinctions (higher levels) have the largest singular values, and hence will be learned first. (c) The output covariance of the data consists of hierarchically organized blocks.

address the output data $y^\mu, \mu = 1, \ldots, P$. Each y^μ is an N-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 (see e.g. Fig. 4A). Each feature i undergoes its own diffusion process on the tree, *independent* of any other feature j. This entire process, described below, yields a hierarchical structure on the set of examples $\mu = 1, \ldots, P$, which are in one-to-one correspondence with the leaves of the tree.

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

Given a single feature component i, its value across P examples is determined as follows. First draw a random variable $\eta^{(0)}$ associated with the root node at the top of the tree. The variable $\eta^{(0)}$ takes the values ± 1 with equal probability $\frac{1}{2}$. Next, for each of the B_0 descendants below the root node at level 1, pick a random variable $\eta_i^{(1)}$, for $i = 1, \ldots, B_0$. This variable $\eta_i^{(1)}$ takes the value $\eta^{(0)}$ with probability $1 - \varepsilon$ 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 $1 - \varepsilon$, 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 $\mu = 1, \ldots, P$. This process is repeated independently for N feature components.

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

$$q_k = \frac{1}{N} \sum_{i=1}^{N} y_i^{\mu_1} y_i^{\mu_2}, \tag{11}$$

where again, the first common ancestor of leaves μ_1 and μ_2 arises at level k. It is possible to explicitly compute q_k for the generative model described above, which yields

$$q_k = (1 - 4\varepsilon(1-\varepsilon))^{D-1-k}. \tag{12}$$

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). Thus pairs of examples with a more recent common ancestor have stronger overlap than pairs of examples with a more distant

common ancestor (see e.g. Fig. 4C). These $D-1$ numbers $q_0,\ldots,q_{D-2}$, along with the number of nodes at each level $M_0,\ldots,M_{D-1}$, are the fundamental parameters of the hierarchical structure of the feature vectors; they determine the correlation matrix across examples, i.e. the $P\times P$ matrix with

$$\Sigma_{\mu_1\mu_2}=\frac{1}{N}\sum_{i=1}^{N}y_i^{\mu_1}y_i^{\mu_2}, \tag{13}$$

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

Input-output correlations for orthogonal inputs and hierarchical outputs We are interested in the singular values and vectors $(s_\alpha,u^\alpha,v^\alpha)$ of Σ^{31} defined in (5), since these were shown previously to drive the learning dynamics. We assume the P output feature vectors are generated hierarchically as in the previous section, and use a localist representation in the input with $N_1=P$ input neurons and $x_i^\mu=\delta_{\mu i}$ (though we note that the localist assumption is not necessary–all that is required is orthogonal inputs). The input-output correlation matrix Σ^{31} is then an $N\times P$ matrix with elements ${\Sigma^{31}}_{i\mu}=y_i^\mu$, with $i=1,\ldots,N$ indexing feature components, and $\mu=1,\ldots,P$ indexing examples. We note that

$${\Sigma^{31}}^T\Sigma^{31}=V^{11}{S^{31}}^T S^{31}{V^{11}}^T=N\Sigma, \tag{14}$$

where Σ, defined in (13), is the correlation matrix across examples. From this we see that the eigenvectors of Σ are the same as the right singular vectors v^α of Σ^{31}, and if the associated eigenvalue of Σ is λ_α, then the associated singular value of Σ^{31} is $s_\alpha=\sqrt{N\lambda_\alpha}$. Thus finding the singular values s_α of Σ^{31}, which determine the time scales of learning, through (9), reduces to finding the eigenvalues λ_α of Σ.

Moreover, the eigenvectors v^α of Σ are of great interest precisely because they are the input-analyzing singular vectors in (7). At any point in developmental time t, if the collection of input modes v^α that have been learned so far (i.e. have large $a(t,s_\alpha,a_\alpha^0)$ in (7)), cannot discriminate across (i.e. take different values on) a subset of input examples, then neither the network's input-output map, nor the network's internal representations, can discriminate across that subset of examples.

We now find the eigenvalues λ_α and eigenvectors v^α of the correlation matrix across examples, Σ in (13). 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 y^μ, for $\mu=1,\ldots,P$ (see e.g. Fig. 4B). Many of these eigenvectors will take constant values across subsets of leaves in a manner that respects the topology of the tree. To describe this phenomenon, let us define the notion of a level l funtion $f(\mu)$ on the leaves as follows: first consider a function g which takes M_l values on the M_l nodes at level l of the tree. Each leaf μ of the tree at level $D-1$ has a unique ancestor $\nu(\mu)$ at level l; let the corresponding level l function on the leaves induced by g be $f(\mu)=g(\nu(\mu))$. This function is constant across all subsets of leaves which have the same ancestor at level l. Thus any level l function cannot discriminate between examples that have a common ancestor which lives at any level $l'>l$ (i.e. any level lower than l).

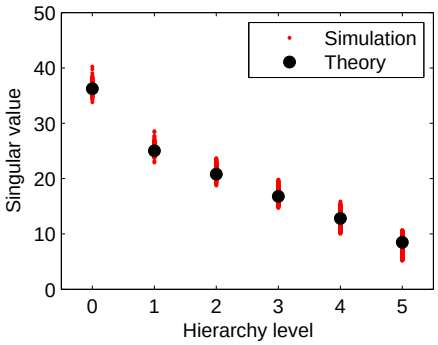

Figure 5: Agreement between theoretically predicted singular values (obtained from (12) and (15)) and simulation for hierarchically structured data. The simulations show singular values arising from sampling 200 features from a hierarchical generative model with six levels, binary branching, and $\varepsilon=0.1$. The singular values are a decreasing function of the hierarchy level, implying that finer distinctions among examples will be learned more slowly.

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

$$\lambda_l\equiv P\left(\sum_{k=l}^{D-1}\frac{\Delta_l}{M_l}\right), \tag{15}$$

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 (see e.g. Fig. 5). 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 next describe the level l eigenvectors. They come in M_{l-1} families, one family for each node at the higher level $l-1$ ($l=0$ is a special case–there is only one eigenvector at this level and it is a uniform mode that takes a constant value on all P leaves). The family of level l eigenvectors associated with a node ν at level $l-1$ takes nonzero values only on leaves which are descendants of ν. They are induced by functions on the B_{l-1} direct descendants of ν which sum to 0. There can only be $B_{l-1}-1$ such orthonormal eigenvectors, hence the degeneracy of all level l eigenvectors is $M_{l-1}(B_{l-1}-1)$. Together, linear combinations of all these level l eigenvectors can be used to assign different values to

any two examples whose first common ancestor arises at level l but not at any lower level $l' > l$. Thus level l eigenvectors do not see any structure in the data at any level of granularity below level l of the hierarchical tree which generated the data. Recall that these eigenvectors are precisely the input modes which project examples onto internal representations in the multilayer network. Importantly, this automatically implies that structure below level l in the tree cannot arise in the internal representations of the network until after structure at level $l-1$ is learned. Indeed, quantitatively, the time scale for learning input structure at level l can be computed (in the limit of large branching ratios) through (9) to be

$$\tau_l = O\left(\sqrt{\frac{M_l}{\Delta_l}}\right). \tag{16}$$

This time scale is proportional to the square root of the number of ancestors at level l, and interestingly, for constant branching factor B, it grows exponentially with l.

Summary of the statistics of hierarchical data Thus we have shown that the singular vectors of data from a hierarchical diffusion process correspond exactly to the hierarchical distinctions in the underlying tree, and furthermore, that singular vectors corresponding to broader hierarchical distinctions have larger singular values than those corresponding to finer distinctions (Fig. 4AB). In combination with the preceding analysis of neural network learning dynamics, this result shows that our deep neural network must exhibit progressive differentiation on any dataset generated by an instance of this class of hierarchical, branching diffusion processes.

Discussion

Our results explore the rich dynamics arising from gradient descent learning in a deep neural network, despite a completely linear input-output mapping. We have shown that these dynamics, driven solely by second order statistics, identify coherently covarying input and output modes in the learning environment, and we expressed the full time course of learning in terms of these modes. Finally, we moved beyond particular datasets to extract general principles by analyzing the covariance structure of hierarchical probabilistic models, showing that progressive differentiation is a general feature of learning in deep neural networks.

We have focused our analysis on a few notable features of the learning dynamics–progressive differentiation and stage-like transitions–but our framework yields insights (to be presented elsewhere) into many other phenomena in semantic development such as, erroneous "illusory correlations" early in learning, familiarity and typicality effects, inductive property judgements, and the impact of perceptual correlations on learning dynamics. Moreover, this approach enables quantitative definitions of important intuitive notions like "category coherence", and yields precise theorems delineating how category coherence controls network learning rates.

By connecting probabilistic models and neural networks, our framework quantitatively links structured environments to learning dynamics. In future work, it will be important to compare the features of our learning model with those of structured probabilistic models (e.g., Kemp and Tenenbaum (2008). Like structured probabilistic models, our model can learn a range of different structure types, but unlike their model, it does so without prior enumeration of such structures. Furthermore, our models can easily learn to represent data that are approximations of hybrids of different structure types – features that, we believe, characterize natural domains, such as the domain of living things considered here.

Acknowledgments

S.G. thanks DARPA, BioX, Burroughs-Wellcome, and Swartz foundations for support. J.L.M. was supported by AFOSR. A.S. was supported by a NDSEG Fellowship and MBC Traineeship. We thank Juan Gao and Jeremy Glick for useful discussions.

References

Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. New York: Basic Books.

Keil, F. (1979). *Semantic and conceptual development: An ontological perspective*. Cambridge, MA: Harvard University Press.

Kemp, C., & Tenenbaum, J. B. (2008, August). The discovery of structural form. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(31), 10687–92.

Mandler, J. M., & McDonough, L. (1993). Concept Formation in Infancy. *Cognitive Development*, *8*, 291–318.

McClelland, J. L. (1995). A Connectionist Perspective on Knowledge and Development. In T. Simon & G. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling.* Hillsdale, NJ: Erlbaum.

Plunkett, K., & Sinha, C. (1992). Connectionism and developmental theory. *British Journal of Developmental Psychology*, *10*(3), 209–254.

Quinn, P., & Johnson, M. (1997). The emergence of perceptual category representations in young infants: A connectionist analysis. *Journal of Experimental Child Psychology*, *66*, 236–263.

Rogers, T., & McClelland, J. (2004). *Semantic cognition: A parallel distributed processing approach*. Cambridge, MA: MIT Press.

Rumelhart, D., & Todd, P. (1993). Learning and connectionist representations. In D. Meyer & S. Kornblum (Eds.), *Attention and performance xiv: Synergies in experimental psychology, artifical intelligence, and cognitive neuroscience.* Cambridge, MA: MIT Press.

Siegler, R. (1976). Three aspects of cognitive development. *Cognitive Psychology*, *8*, 481–520.