Learning hierarchical category structure 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

Psychological experiments have revealed remarkable regularities in the developmental time course of cognition. Infants generally acquire broad categorical distinctions (i.e., plant/animal) before finer ones (i.e., bird/fish), and periods of little change are often punctuated by stage-like transitions. This pattern of progressive differentiation has also been seen in neural network models as they learn from exposure to training data. Our work explains *why* the networks exhibit these phenomena. We find solutions to the dynamics of error-correcting learning in linear three layer neural networks. These solutions link the statistics of the training set and the dynamics of learning in the network, and characterize formally how learning leads to the emergence of structured representations for arbitrary training environments. We then consider training a neural network on data generated by a hierarchically structured probabilistic generative process. Our results reveal that, for a broad class of such structures, the learning dynamics must exhibit progressive, coarse-to-fine differentiation with stage-like transitions 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 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 progressing through successive levels of 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?

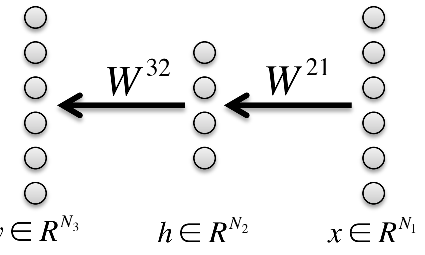

Figure 1: The three layer network analyzed in this work.

Here we analyze the learning dynamics of a *linear* three 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.

We outline our approach here in brief. We begin by decomposing the training set to identify important dimensions of variation using the singular value decomposition (SVD), which will turn out to be fundamental to our analysis. Next, we examine the equations governing gradient descent learning and show that they can be solved in terms of the SVD of the training set. This solution analytically expresses the weight values of the neural network at any point in time during learning as a function of the input training set. Finally, we consider generating the training set from a hierarchical probabilistic generative model. We analytically calculate the SVD of training sets so generated, which in combination with our previous results gives a formal grounding for how neural networks will learn about hierarchical categorical structure. We show that networks must exhibit progressive differentiation of categorical structure and stage-like transitions for any training set generated by a class of hierarchical generative models.

Decomposing the training set

Our fundamental goal is to understand the dynamics of learning in neural networks as a function of the training set. Toward this goal, in this section we introduce the singular value decomposition, which identifies important dimensions of variation in the training set. The SVD will turn out to be fundamentally linked to learning dynamics, a connection we develop in the next section. We wish to train a neural network to learn a particular input-output map from a set of P training examples $\{x^\mu, y^\mu\}, \mu = 1, \ldots, P$. These P pairs of vectors constitute the training set. In the model of semantic development introduced by Rumelhart and Todd (1993), for instance, elements of x^μ correspond to input units representing items such as *Canary* or *Rose*. The elements of y^μ correspond to output units representing possible predicates or attributes such as *can Fly* or *has Petals* that may or may not apply to each item. Hence each example links a particular item to a set of properties, and the training set contains the semantic content in the world to be learned by the network.

For concreteness, we consider a simple example dataset with four items (*Canary, Salmon, Oak,* and *Rose*) and five properties. 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. In a more natural data set, the plant-animal, bird-fish, and tree-flower distinctions are based on clusters of covarying properties, for which the single properties identified here serve as proxies.

An important function of the training set is the input-output correlation matrix

$$\Sigma^{31} \equiv \sum_{\mu=1}^{P} y^\mu x^\mu \equiv E[yx^T]. \quad (1)$$

For our example dataset, this matrix is shown in Fig. 2. Each column corresponds to an item, and denotes the properties possessed by that particular item.

Our example dataset contains important shared structure. The *Canary* and *Salmon*, for instance, both *can Move*, and hence may naturally be grouped together. Intuitively, they are both *animals*, and as a consequence have certain properties in common that are typical of animals. How can we identify these coherently covarying groups of items and their properties? We will show that the singular value decomposition of the input-output correlation matrix accomplishes exactly this.

The singular value decomposition (SVD)

$$\Sigma^{31} = U^{33} S^{31} V^{11^T} = \sum_{\alpha=1}^{N_1} s_\alpha u^\alpha v^{\alpha T}, \quad (2)$$

decomposes any matrix into the product of three matrices. Each of these matrices has an important real world interpretation. We call the $N_1 \times N_1$ orthogonal matrix V^{11} the *object analyzer*–it determines the position of a particular item along a number of important dimensions of the training set. The first row of V^{11^T}, for instance, 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. In our example dataset, the three dimensions identified by the SVD are *animal-plant*, *bird-fish*, and *flower-tree*.

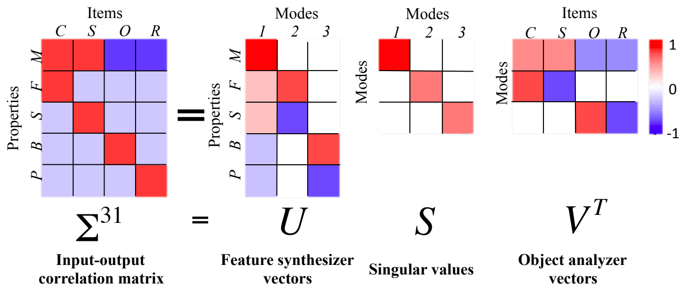

Figure 2: First three modes of the singular value decomposition of a toy dataset. Left: The learning environment is specified by an input-output correlation matrix. Right: The SVD decomposes Σ^{31} into *modes* that link a set of coherently covarying items (*object analyzer* vectors in the rows of V^T) to a set of coherently covarying properties (*feature synthesizer* vectors in the columns of U). 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.

The $N_3 \times N_3$ orthogonal matrix U^{33} can be interpreted as a *feature synthesizer*–it contains those features typical of a particular dimension in each column. Hence the feature synthesizer associated with the *animal-plant* dimension has positive values for *can Move*, since animals typically can move while plants cannot.

Finally the $N_3 \times N_1$ *association strength* matrix S^{31} captures the overall strength of the association between an input dimension and output dimension. It is nonzero only on the diagonal; these elements are the singular values $s_\alpha, \alpha = 1, \ldots, N_1$ ordered so that $s_1 \geq s_2 \geq \cdots \geq s_{N_1}$. The large association strength 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*.

In a larger training set, the SVD will extract modes that capture patterns of coherent covariation in the properties of items in the training set. The quantities defining each mode, $\{s_\alpha, u^\alpha, v^\alpha\}$, are connected to the learning dynamics of neural networks in the next section.

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). 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).

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 standard back propagation learning rule

$$\begin{aligned} \Delta W^{21} &= \lambda W^{32^T}(y^\mu - \hat{y}^\mu)x^{\mu T} && (3)\\ \Delta W^{32} &= \lambda(y^\mu - \hat{y}^\mu)h^{\mu T}, && (4)\end{aligned}$$

for each example μ, where $\hat{y}^\mu = W^{32}W^{21}x^\mu$ denotes the output of the network in response to input example x^μ, $h^\mu = W^{21}x^\mu$ is the hidden unit activity, and λ is a small learning rate. Here $W^{32^T}(y^\mu - \hat{y}^\mu)$ in (3) corresponds to the signal back-propagated to the hidden units through the hidden-to-output weights. These equations emphasize that the learning process works by comparing the network's current output $\hat{y}^\mu$ to the desired target output y^μ, and adjusting weights based on this error term.

By a substitution and rearrangement, however, we can equivalently write these equations as

$$\begin{aligned} \Delta W^{21} &= \lambda W^{32^T}\left(y^\mu x^{\mu T} - W^{32}W^{21}x^\mu x^{\mu T}\right) && (5)\\ \Delta W^{32} &= \lambda\left(y^\mu x^{\mu T} - W^{32}W^{21}x^\mu x^{\mu T}\right)W^{21^T}. && (6)\end{aligned}$$

This form emphasizes two crucial aspects of the learning dynamics. First, it highlights the importance of the statistics of the training set. In particular, the training set enters only through two terms, one related to the input-output correlations $y^\mu x^{\mu T}$ and the other related to the input correlations $x^\mu x^{\mu T}$. Indeed, if λ is sufficiently small so that weights change only a small amount per epoch, we can rewrite these equations in a batch update form by averaging over the training set to obtain the mean change in weights per learning epoch,

$$\begin{aligned} \tau\frac{d}{dt}W^{21} &= W^{32^T}\left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right) && (7)\\ \tau\frac{d}{dt}W^{32} &= \left(\Sigma^{31} - W^{32}W^{21}\Sigma^{11}\right)W^{21^T}, && (8)\end{aligned}$$

where $\Sigma^{11} \equiv \sum_{\mu=1} x^\mu x^{\mu T} \equiv E[xx^T]$ is an $N_1 \times N_1$ input correlation matrix, Σ^{31} is the $N_3 \times N_1$ input-output correlation matrix defined previously, and $\tau \equiv \frac{P}{\lambda}$. Hence we see that linear networks are sensitive only to the second order statistics of inputs and outputs. In general the learning process is driven by both the input and input-output correlation matrices. Here we take the simplifying assumption that these input correlations are insignificant; formally, we assume $\Sigma^{11} = I$, the identity matrix. Concretely, this assumption corresponds to the supposition that input representations for different items are highly differentiated from, or orthogonal to each other. While this is unlikely to hold exactly in any natural domain, we take this assumption for two reasons. First, it was used in prior simulation studies (Rogers & McClelland, 2004), and hence our attempt to understand their results is not limited by this assumption. Second, Rogers and McClelland (2004) have shown that relaxing this assumption to incorporate more complex input correlations leaves intact the basic phenomena of progressive differentiation and stage-like transitions in learning. Nevertheless, understanding the impact of input correlations is an important direction for further work.

Second, the form of Eqns. (7)-(8) highlights the *coupling* between the two equations: to know how to change W^{21} we must know W^{32}, and visa versa, since each appears in the update equation for the other. This coupling is the crucial element added by the addition of a hidden layer, and as we shall see, it qualitatively changes the learning dynamics of the network compared to a "shallow" network with no hidden layer. Intuitively, this coupling complicates the learning procedure since both weight matrices must cooperate to produce the correct answer; but crucially, it enables *knowledge sharing* between different items, by assigning them similar hidden unit representations. Without this coupling, the network would learn each item-property association independently, and would not be sensitive to shared structure in the training set.

The temporal dynamics of learning To understand the connection between learning dynamics and training set statistics, then, we can solve Eqns. (7)-(8). We have found a class of exact solutions (whose derivation will be presented elsewhere) that describe the weights of the network over time during learning, as a function of the training set. In particular, the composite mapping at any time t is given by

$$W^{32}(t)W^{21}(t) = \sum_{\alpha=1}^{N_2} a(t, s_\alpha, a_\alpha^0)\, u^\alpha v^{\alpha T}, \qquad (9)$$

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

$$a(t,s,a_0) = \frac{se^{2st/\tau}}{e^{2st/\tau} - 1 + s/a_0}. \qquad (10)$$

That is, the network learns about the N_2 strongest input-output modes identified by the singular value decomposition, progressively incorporating each mode into its representation. The coefficient $a(t,s^\alpha,a_0)$ describes how strongly input-output mode α has been learned by time t, starting from some small initial value of a_0. As can be seen from Fig. 3, this function is a sigmoidal curve, capturing the fact that the network initially knows nothing about a particular dimension (the *animal-plant* dimension, say), but over time learns the importance of this dimension and incorporates it into its representation, ultimately reaching the correct association strength s^α. At this point the network correctly maps items onto the *animal-plant* dimension using the object analyzer vector $v^{\alpha T}$, and generates the corresponding correct features using the feature synthesizer vector u^α.

Eqns. (9)-(10) describe the fundamental connection between the structure of a training set and learning dynamics. In particular, the dynamics depends on the singular value decomposition of the input-output correlation matrix of the

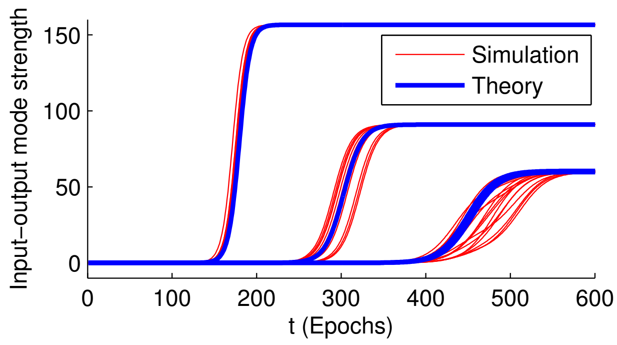

Figure 3: Close agreement between theoretically predicted time course and numerical simulations. Simulations were performed with a dataset sampled from the hierarchical diffusion process described in detail in a later section, with $D = 3$ hierarchical levels, binary branching, flip probability $\varepsilon = 0.1$, 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. (7)-(8) starting from different random initializations, and blue traces show theoretical curves obtained from (10).

training set. Further, they reveal important properties of these learning dynamics.

First, each input-output mode is learned on a different time scale, governed by its singular value s_α. To calculate this time scale, we can assume a small initial condition $a_0 = \varepsilon$ and ask when $a(t)$ in (10) rises to within ε of the final value s_α, i.e. $a(t) = s_\alpha - \varepsilon$; then the timescale of learning in the limit $\varepsilon \to 0$ is

$$t(s, \varepsilon) = \frac{\tau}{s_\alpha} \ln \frac{s_\alpha}{\varepsilon}. \qquad (11)$$

Hence up to a logarithmic factor, the time required to learn an input-output mode is inversely related to its association strength, quantified through its singular value.

Second, these dynamics reveal stage-like transitions in learning performance. Intuitively, this property arises from the sigmoidal transition in (10) 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. Because of the sigmoidal shape, the solution can remain very small for a long period of time before rapidly transitioning to mastery. To formalize this, we note that the time it takes to reach half mastery (i.e. $a(t_{half}) = s/2$) is

$$t_{half} = \frac{\tau}{2s} \log\left(\frac{s}{a_0} - 1\right). \qquad (12)$$

In contrast, the duration of the transition period in which the weights change rapidly is $t_{trans} = \frac{2\tau}{s}$ (using a linear approximation). Thus, by starting with a very small initial condition for the weights (i.e. $a_0 \approx 0$), it is clear that one can make the ratio t_{trans}/t_{half} arbitrarily small, i.e., the transition period can be very brief relative to the long initial period of dormancy. Hence the learning dynamics of (7)-(8) exhibit sharp stage-like transitions. Importantly, we can prove that networks with only direct input-output connections and no hidden layer are not capable of such stage-like transitions. Their existence is an emergent property of nonlinear learning dynamics in deep networks with at least one hidden layer.

The result in (9) is the solution to (7)-(8) 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.

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, 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. By analytically calculating the SVD of this data, 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 We propose a simple generative model of hierarchical data $\{x^\mu, y^\mu\}$, and compute for this model the input-output modes $(s_\alpha, u^\alpha, v^\alpha)$ which drive learning. The hierarchical structure in the generative model is represented by a tree (see e.g. Fig. 4). Each leaf node of this tree corresponds to an item in the dataset. Our generative process assigns features to these items such that items with more recent common ancestors are more likely to share features. For instance, our example dataset might have been generated by a three level binary tree with four leaf nodes. The top level would separate the animals from the plants, while the next level would separate the birds from the fish and the flowers from the plants.

In detail, to sample one feature's value across items, the root node is randomly set to ± 1 with equal probability $\frac{1}{2}$; next this value diffuses to children nodes, where its sign is flipped with a small probability ε. This process continues until the leaf nodes have been assigned values. These assignments yield the value of this feature on each item.

Under this process, the *can Move* feature, for example, might have arisen as follows: randomly the root node of the three level binary tree was assigned a value of 1. This value diffused down to the two second level nodes, maybe in this instance changing sign to -1 for the parent node of the plants, but not changing for the parent node of the animals. Then these values diffused down to the leaf nodes representing the individual items, perhaps not flipping sign for any of them. Hence the ultimate feature assignment would be $+1$ on the *Canary* and *Salmon* and -1 on the *Flower* and *Tree*. This is just one possible sample from the generative model, but serves to illustrate how hierarchical structure arises from the feature generation process. To generate more features, the process is repeated independently N times.

For simplicity, we consider trees with a regular branching structure. The tree has D levels indexed by $l = 0, \ldots, D-1$, with M_l total nodes at level l. 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 ($M_{D-1} = P$).

We have thus far described the output feature vectors y^μ. To complete the specification of the training set, we assume that the input vectors x^μ are simply chosen to be highly distinct (i.e., orthogonal). One such choice is a localist coding scheme in which a different element is active to represent the presence of each item.

Input-output modes of hierarchical data How will our neural network learn about training sets generated as just described? To understand this, we calculate the SVD of such training sets. We will see that the input-output modes identified by the SVD exactly mirror the tree structure used to generate the dataset. The feature generation process described in the previous section generates a training set with N features. In the limit of large numbers of features, we obtain the following (the full derivation to be presented elsewhere):

The *object analyzer* vectors exactly mirror the tree structure, as shown in Fig. 4. One mode will correspond to a broad, high level distinction (e.g., *animal-plant*) near the root of the tree, while another will correspond to a more detailed distinction (e.g., *bird-fish*). For binary trees, each object analyzer vector will have positive weights on all items on one side of a binary distinction, and negative weights on all items on the other side. The rest of the entries will be zero. Hence this object analyzer vector will only be able to tell items apart with respect to this one distinction. It contains no information about higher or lower level distinctions in the tree. For trees with other branching factors, the situation is the same: additional object analyzer vectors are introduced to permit distinctions between more than two options, but these vectors contain no information about distinctions at other levels in the tree.

The *association strength* or singular value s_l associated with level l of the binary tree is

$$s_l = \sqrt{NP\left(\sum_{k=l}^{D-1} \frac{\Delta_l}{M_l}\right)}, \tag{13}$$

where $q_k = (1-4\varepsilon(1-\varepsilon))^{D-1-k}$ and $\Delta_l \equiv q_l - q_{l-1}$, with the caveat that $q_{-1} \equiv 0$.

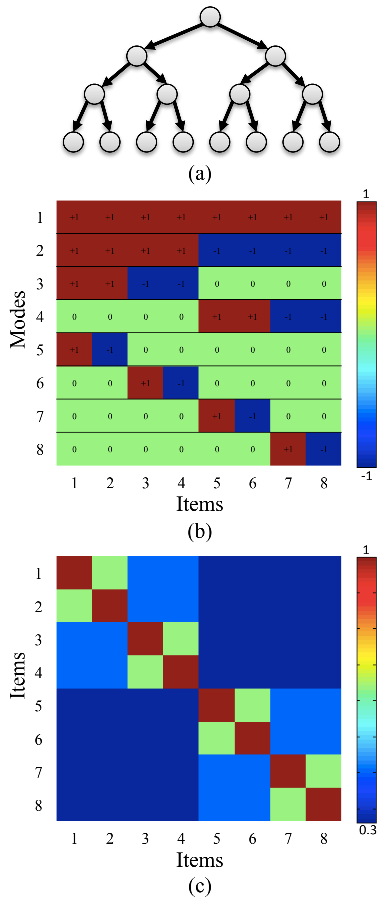

Figure 4: Statistical structure of hierarchical data. (a) Example hierarchical diffusion process with $D = 4$ levels and branching factor $B = 2$. (b) Analytically derived input singular vectors, or modes, (up to a scaling) of the resulting data, ordered top-to-bottom by singular value. Besides mode 0, each mode, or object analyzer, can discriminate objects, or leaves of the tree, whose first common ancestor arises at a given level of the tree. This level is 0 for mode 2, 1 for modes 3 and 4, and 3 for modes 5 through 8. Singular modes corresponding to broad distinctions (higher levels) have larger singular values, and hence will be learned earlier. (c) The covariance matrix between pairs of objects in the output feature space consists of hierarchically organized blocks.

While this equation gives the correct quantitative value for the association strength in terms of the parameters of the generative process, its most important property is its qualitative behavior: it is a decreasing function of the hierarchy level l (see, e.g., Fig. 5). Crucially, this means that the input-output modes corresponding to broader distinctions like *animal-plant* have a stronger *association strength* than those corresponding to finer distinctions like *bird-fish*. Since we have previously shown that modes with stronger association strengths are learned more quickly, this immediately implies that broader distinctions among examples will be learned faster than fine-grained distinctions among examples.

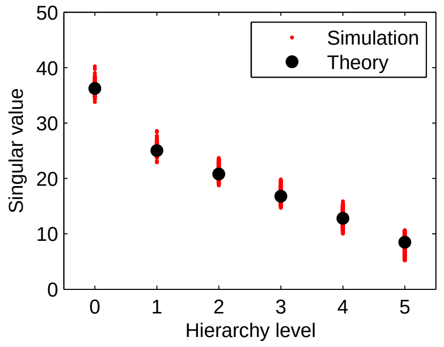

Figure 5: Agreement between theoretically computed singular values in the limit of large numbers of features (obtained from (13)) and simulation for hierarchically structured data. The simulations show singular values arising from sampling 200 features from a hierarchical generative model with six levels, 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.

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 hierarchically structured training data 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 neural network learning model with those of structured probabilistic learning models (e.g., Kemp and Tenenbaum (2008). Like structured models, neural networks can learn a range of different structure types, but unlike structured models, networks can learn without prior enumeration of such structures. Furthermore, networks can easily learn to represent data that are approximations or 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 Sloan 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.