Temporal Sparseness of the Premotor Drive Is Important for Rapid Learning in a Neural Network Model of Birdsong

Ila R. Fiete, Richard H.R. Hahnloser, Michale S. Fee, and H. Sebastian Seung

INTRODUCTION

Birdsong is a complex, learned motor behavior driven by a discrete set of premotor brain nuclei with well-studied anatomy. Neural activity, too, has been characterized in these nuclei, through recordings in awake singing birds, making the birdsong circuit a uniquely rich and accessible system for the study of motor coding and learning.

Juvenile male songbirds learn their songs from adult male tutors of the same species. Singing is used for courtship and territorial displays, and in evolutionary terms is an important tutor of the same species. Singing is used for courtship and territorial displays, and in evolutionary terms is an important
tutors of the same species. Singing is used for courtship and territorial displays, and in evolutionary terms is an important

Song learning is thought to involve plasticity of synapses from HVC to RA. This is because these synapses display anatomical evidence of extensive synaptic growth and redistribution and physiological evidence of synaptic change and maturation during the critical period. The temporal sparseness of HVC activity implies that these HVC–RA synapses are used in a very special manner during song: that is, each synapse is used during only one instant in the motif. Is there any functional significance to this way of using synapses? Here we investigate the possibility that it facilitates song learning.

Intuitively, the situation where each synapse participates in the production of just one short part of the motif seems ideally suited for minimizing interference between different synapses during learning. In this paper we make the intuitive argument more concrete through both computer simulations and mathematical analysis of a simple neural network model of birdsong learning.

It has been observed that interference between synapses can hinder learning in artificial recurrent neural networks. Because of the multilayered architecture of the song motor system, we are here motivated to study the effect in a feedforward multilayer network. Experiments indicate that the young bird uses the mismatch or error between its own vocalizations and a desired song template (an internally stored copy of a tutor song) to iteratively modify its song to match the template (Brainard and
tutors of the same species. Singing is used for courtship and territorial displays, and in evolutionary terms is an important

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked "advertisement" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.
Doupe 2000; Konishi 1965). Thus, the goal of learning in our feedforward model network (of HVC, RA, and an output motor layer), is to alter the initial output sequence of motor activity by gradual adjustment of the HVC-to-RA weights, until the output sequence matches a specified desired sequence (Doya and Sejnowski 1995; Troyer and Doupe 2000).

It is not known how the brain translates goal-directed problems such as song imitation into prescriptions for synaptic change, although it is thought that if distance to the goal is quantified in a reward (error) function, neural and synaptic changes may occur in directions that increase the reward (decrease error), thus performing hill-climbing on the function. A common computational approach in modeling this phenomenon is to define such an error function, then move on the error surface toward the minimum along the gradient, or direction of steepest descent. This can be done by direct gradient calculation in single-layer networks, or by backpropagation, a simple technique for gradient descent in multilayer networks. Hill climbing can also be achieved by more biologically plausible learning algorithms that perform a stochastic approximation of gradient following without needing to explicitly compute the gradient (Bartlett and Baxter 1999; Seung 2003; Williams 1992). For simplicity, we apply learning by direct gradient following (backpropagation). Because the various gradient-based learning rules described above are in a mathematically similar class, we expect sparseness arguments made in the context of one learning rule to generalize to the others in the same class.

**Methods**

**General framework**

We study a multilayer feedforward network (Fig. 1) with an HVC layer that provides sequential inputs to the network and drives activity in the hidden layer RA; the output layer of motor units is driven by activity in RA. HVC activities are written as \( h(t) \), RA activities as \( r_j(t) \), and output activities as \( o_j(t) \), with

\[
 r_j(t) = f(\sum_{i=1}^{N_h} W_{ji} h(t) - \theta_j) \quad (1)
\]

and

\[
 o_j(t) = \sum_{i=1}^{N_h} A_{ji} r_i(t) \quad (2)
\]

where \( N_h \), \( N_r \), and \( N_o \) are the numbers of units in the HVC, RA, and motor layers, respectively; \( f \) is the activation function of RA neurons; and \( \theta_j \) is the threshold for the \( j \)th RA neuron. The plastic weights from HVC to RA are given by the matrix \( W \); because there is no direct evidence of plasticity in the connections from RA to the motor neurons, we take these weights to be fixed, and represent them by a fixed weight matrix \( A \).

Observational evidence suggests that vocal motor learning in the zebra finch segments roughly into 2 phases: first, a temporal motor sequence is established, and later the notes and syllables occurring in that motor sequence become more distinct, diversified, and refined (Tchernichovski et al. 2001). In that the goal of this work is to study the effects of HVC sparseness on the learning of feedforward premotor representations, we do not deal with the formation of sequences within HVC; instead we focus on the formation and refinement of HVC–motor representations as seen in the latter phase. The sequential patterns of HVC activities and desired output activities are externally imposed (see below for numerical details) in our simulations, and do not change throughout learning; the goal of the network is to learn to match the actual outputs \( o_j(t) \) of the network, driven by HVC activity, with the desired outputs \( d_j(t) \), through adjustment of the plastic weights. In one pass through the song motif, called an epoch, the network outputs are computed from Eqs. 1 and 2. The total network error for that epoch is determined from the objective function

\[
 C = \int_0^T dt \sum_{i=1}^{N_o} (d_i(t) - o_i(t))^2 \quad (3)
\]

For learning, network weights \( W \) are adjusted after each epoch to minimize this cost function according to the backpropagation gradient-descent rule

\[
 \Delta W_j = -\eta \frac{\partial C}{\partial W_j} = \eta \int_0^T dt \sum_{i=1}^{N_o} 2(d_i(t) - o_i(t)) A_{ji} o_i(t) \quad (4)
\]

where \( f' \) is the derivative of the activation function of RA neuron \( j \), and the parameter \( \eta \) scales the overall size of the weight update.

**Numerical details of nonlinear network simulations**

We simulate learning in the network described above, with \( N_h = 500 \) HVC neurons, \( N_r = 800 \) RA neurons, and \( N_o = 2 \) output units. Assuming that each HVC neuron bursts \( B \) times per motif, activity for the \( j \)th HVC neuron is fixed by choosing \( B \) onset times \( \{ t_1^j, t_2^j, \ldots, t_B^j \} \) at random from the entire time interval \( T \). A burst is then modeled as a simple binary pulse of duration \( \tau_s \) so that \( h(t) = 1 \) for \( t_i^j \leq t \leq t_i^j + \tau_s \) and \( h(t) = 0 \) otherwise (Fig. 2A). We use values of \( B = 1, 2, 4, 8 \), and based on numerical observations of the HVC burst length (Hahnloser et al. 2002), use \( \tau_s = 6 \) ms. We assume a nonlinear form for the RA activation function, given by the sigmoid \( f(x) = r_{max}(1 + e^{-x})^{-1} \), so \( f'(x) = f(x)(r_{max} - f(x))(2/r_{max}) \), with \( r_{max} = 600 \) Hz and \( s = 5 \) (a parameter that stretches the analog part of the response; large values of \( s \) produce analog neurons with a linear regime and saturation, whereas the \( s \to 0 \) limit produces binary neurons. In experimental current-injection studies, RA neurons show a range of linear response up to at least 100 Hz (Spiro et al. 1999), and routinely fire bursts of spikes at 500 Hz during song, motivating our choice of \( s = 5 \) and \( r_{max} = 600 \) Hz. In all simulations, the total duration of the simulated song motif is \( T = 150 \) ms, and time is discretized with a grain of \( dr = 0.1 \) ms. The initial HVC-to-RA weights \( W_j \) are picked randomly from the interval \([0, 1/B]\) (scaling with \( B \) to keep the summed drive to RA fixed as the number of bursts per neuron per motif is varied in HVC), with 40% of them \( (P_{d} = 0.4) \) randomly diluted to zero. The threshold for RA neurons is given by \( \theta = 1.2(1 - P_{d}) N_r \tau_s / T \), where \( N_r \tau_s / T \) is the average input received by the average RA neuron from HVC at each time in the song; the factor 1.2 is chosen to keep RA activity low initially. Each RA neuron projects to one output neuron (i.e., the RA-to-output weight matrix \( A \) is
block-diagonal), and equal numbers of RA neurons project to each output. The nonzero entries of $A$ are chosen from a Gaussian distribution with mean 1 and SD 1/4. Desired sequences $d_k(t)$ for the output units are fixed by choosing a sequence of steps of 12-ms duration and random heights chosen from the interval $[0, N_o/(8N_H)]$, and are smoothed with a 2-ms linear low-pass filter. The gradient-following rule, Eq. 4, is used to update the weights $W$ after each epoch.

To study the effects of sparse HVC activity on learning speed, we performed 4 groups of simulations where $B$, the number of bursts-per-HVC-neuron-per-song motif, was fixed at $B = 1, 2, 4, \text{or} 8$, respectively. For each $B$, we performed several sets of learning trials with a separate, systematically varied value of the overall learning step-size $\eta$ for each set (more details below). Within each set of simulations, consisting of 15 trials each with fixed $\eta$, the weights $A$ and $W$ were drawn randomly and independently for every trial, as described above. All other parameters, including the desired outputs $d_k(t)$, were kept fixed for all $B$ and all $\eta$. Initially 25 evenly spaced values of $\eta$ were chosen for each $B$, always in a range where some of the values were too large and resulted in divergence of the learning curve, whereas most values resulted in decreasing errors. The (15-trial) averaged learning curves for each $B$ were judged to be rapidly or slowly converging based on the number of epochs taken to cross a preselected, reasonably small error value (see below); only learning curves with nonincreasing error over the length of the simulation were considered. Typically, very small values of $\eta$ result in very slow learning, whereas very large values lead to divergence. Thus, the best learning speeds could be obtained by a choice of $\eta$ away from both extremes. To make sure the learning curves chosen for comparison as a function of $B$ were reasonably close to the best possible curve for each $B$, we picked 2 values of $\eta$ for each $B$ that resulted in the 2 fastest averaged learning curves, and used these as endpoints in another set of learning trials with 10 values of $\eta$ spaced between the endpoints. For each $\eta$, we again averaged 15 trials. By this process, a value of $\eta = \eta^B(B)$ was found that resulted in the fastest learning for each $B$.

The threshold error value at which we consider the network to have learned the task is when it reached an error of 0.02 or better [corresponding to $\int dt \Sigma_k (d_k - \alpha_k)^2 < 1\% \int dt \Sigma_k \alpha_k^2$, thin horizontal line in Fig. 3; for an example of the output performance in what we consider to be a well-learned task, see Fig. 2c where $\int dt \Sigma_k (d_k - \alpha_k)^2 = 0.15\% \int dt \Sigma_k \alpha_k^2$; learning speeds are judged by the number of epochs taken for the learning curves to reach this value.

**Parameter variations and ranges**

The network converged to produce outputs close to the desired outputs over a large range of parameters, so long as a sufficiently small value of the learning rate parameter, $\eta$, was used. This is expected, because with small $\eta$, the learning rule follows the gradient of the error function, and will converge to a local minimum of the error surface; more interestingly, the dependency of learning time on $B$ (see RESULTS) was also consistent across a large parameter range. In simulation, we tried variations where $W$ was drawn from a Gaussian, instead of uniform, random distribution; the initial weight dilution, $P_{dil}$, ranged from 0 to 0.6 (0–60% of the initial weights initially diluted to 0); half of all nonzero weights from RA to each output unit (in $A$) were made negative, mimicking push-pull rather than just pull control over the outputs; the numbers of HVC, RA, and output units were independently varied by factors of 0.5 and 2; the simulated song length ranged from 80 to 400 ms; RA unit activation functions were taken to be linear or sigmoidal. In all of these cases, it was possible to find $\eta$ so that the simulations converged to the desired output, and the dependency of learning time on $B$ was found to be qualitatively the same as for the specific parameters described here.

The results shown here are with parameters chosen according to the following priorities. 1) Simulate the largest network that would run in a reasonable amount of time. We used $N_h = 500$, $N_r = 800$, and $N_o = 2$, in place of $N_h \approx 20,000$, $N_r \approx 7,000$, and $N_o \approx 7$ in the actual bird, where $N_o$ is taken to be the number of individual vocal muscles controlled by RA. The simulated song length $T$ had to be scaled down

**FIG. 3.** Four curves track error as a function of epoch while learning with $B = 1, 2, 4, \text{and} 8$ bursts per HVC neuron per simulated song segment. For each $B$, the overall weight update step size was optimized to give the fastest possible monotonic convergence toward zero error. Number of epochs taken to reach a prespecified learning criterion (thin horizontal line) grows sharply with $B$, nearly doubling each time $B$ doubles.
to compensate for the reduced HVC and RA model populations driving song; thus $T = 150$ ms instead of the approximately 600- to 1,000-ms duration of a typical zebra finch song motif. 2) Initiate (before learning) the HVC-RA weights and RA neural thresholds so that the initial activity in RA is low and nonuniform. This was done because we noticed that, interestingly, if initiated in this way, the postlearning activity in the model RA neurons reliably resembles that of RA neurons in the actual songbird (see RESULTS).

**Numerical eigenvalue computation**

For each of $B = 1, 2, 4$, or 8, we randomly generated a matrix of HVC activity (as described above) with $N_h = 3000, T = 300$ ms, $\tau_a = 6$ ms, and $dt = 0.1$ ms. For each $B$, the HVC equal-time cross-correlation matrix $Q_{ij} = \sum_{t=0}^{T} h_i(t)h_j(t)$ was computed, and its eigenvalues computed numerically.

**RESULTS**

**Simulations**

We simulated learning by gradient following (as described in Eqs. 1–4 and METHODS) in a feedforward network consisting of an HVC, an RA, and a motor output layer (Fig. 1). Sample input (HVC activity) and the initial and desired outputs (for one of 2 output units) are shown in Fig. 2, A and B, respectively. In the simulation of Fig. 2, each HVC neuron is active exactly once in the song motif. After several epochs of learning (gradient descent on the mismatch between actual and desired outputs), activity in the output units closely matches the desired outputs; Fig. 2C. Note that in our model, the RA neurons act as hidden units and their patterns of activity are not explicitly constrained. The activities of 3 randomly selected RA neurons from the model network after learning is complete are shown in Fig. 2, D–F. It is interesting to note that with sigmoid RA activation functions, if initial connections between HVC and RA are weak and random and if initial RA activity is low, the emergent activity patterns of RA neurons in the trained network qualitatively resemble the behavior of real RA neurons recorded in vivo during singing (Yu and Margoliash 1986; A. Leonardo and M. S. Fee, unpublished observations): for example, individual RA unit activity is not well correlated with the outputs, the distribution of single-burst durations of RA neurons resembles that of RA neurons in the singing zebra finch, and similar patterns of output activity may be driven by rather different patterns of activity in RA.

Our goal is to examine the effects of the sparseness of HVC drive on the learning speed of the network. We repeated the learning simulations, as pictured in Fig. 2, with fixed values for the song length, single-burst duration of HVC neurons, and network size, but varied $B$, the number of bursts fired per HVC neuron per song motif (see METHODS). Figure 3 shows the results of this study; the 4 learning curves correspond to simulations where the number of bursts per HVC neuron is varied to be $B = 1, 2, 4$, or 8, respectively. Each curve in Fig. 3 is an average over 15 trials that start with different random initial weights $W$ and $A$ but with a single fixed $B$. The network is considered to have learned the task when the error drops below a prespecified error tolerance, signified by the thin horizontal line. For each value of $B$, the task of learning was realizable (i.e., the network could successfully learn the desired outputs). Also for each $B$, the overall coefficient controlling the weight-update step size was optimized to give the fastest learning possible (see METHODS); thus the learning speed comparison is between the best-case multtrial average curves for each $B$.

In going from $B = 1$ burst per HVC neuron per motif to 2 bursts, we see in Fig. 3 that the learning time (number of iterations for the learning curve to intersect the learning criterion line) nearly doubles; the same happens in going from 2 bursts to 4, or 4 to 8. This apparently strong dependence of learning time on the number of HVC bursts is surprising, considering that in all cases (all $B$) the learning task was realizable, and that the premotor HVC drive in going from $B = 1$ to $B = 4$, for example, was still relatively sparse. The effect, that increasing $B$ leads to increased learning time, persisted over a wide range of network parameters (see note on parameter choices in METHODS). To better understand the process by which more densely distributed HVC bursts per motif lead to slower learning, and why this effect is robust across a broad range of parameters, we turn to an analysis of learning in a linearized version of the network.

**Linear analysis**

We found the basic effect of the slowdown of learning with temporally denser HVC codes to be present regardless of many changes in network properties, such as network size, length of simulated motif, and choice of RA activation function. To isolate the critical factors involved in the learning slowdown, we study the learning curves of a network with the same architecture and learning rule as above, but with linear RA activation functions, $f(y) = y$. Although this is a simplification, a linear network permits us to analytically derive the dependence of the learning curves on $B$, the number of times each HVC neuron bursts during a song motif. Moreover, linear analysis lends itself to a convenient geometric interpretation of the learning process.

**Relation between learning speed and HVC activity.** If RA units are linear, the error function $C$ of Eq. 3 becomes a quadratic surface over the multidimensional space $\{W\}$ of the HVC-to-RA weights (see Appendix)

$$C = \text{Tr} \{AWQW^2A^\prime\}$$

(5)

In geometric terms, $Q$ is a matrix that determines the shape of the quadratic surface, because its eigenvalues specify the overall shape (steepness or flatness) of the quadratic surface along the various directions in weight space. Large eigenvalues correspond to steep directions, and small eigenvalues to shallow ones. In terms of network activities, $Q$ is the zero-time-lag correlation matrix of HVC activity: element $Q_{ij}$ reflects the equal-time cross-correlations in the activity of neurons $i$ and $j$, summed over all times in the motif. For example, if the 2 neurons are always coactive, $Q_{ij}$ is large, and if they are never coactive, $Q_{ij} = 0$. The importance of $Q$ in shaping the error surface emerges from the fact that HVC activity determines which synapses $W$ are active in driving the output, how often they are used, and thus whether and when they must be modified to reduce error.

Learning corresponds to moving downward on the paraboloid quadratic surface by adjustment of the underlying network weights $W$. Learning by gradient descent, Eq. 4, means that the downward movement follows along the direction of the gradi-

---

*J Neurophysiol* • VOL 92 • OCTOBER 2004 • www.jn.org
ent or path steepest descent toward the minimum of the error surface. The total error may be broken down into components of error along different directions in the weight space \(\{W\}\), and it is well known that in a linear system, these component errors decrease as decaying exponentials with different decay rates; these decay rates are determined by the shape of the error surface in that direction. Specifically, with certain assumptions on the distribution of fixed weights \(A\), the optimal (leading to fastest learning) decay rates are given by ratios of the eigenvalues of \(Q\) \(\{\lambda_1, \lambda_2, \ldots, \lambda_N\}\), with the largest eigenvalue, \(\lambda_1\) (see APPENDIX). The learning speed along the direction parallel to eigenvector \(\alpha\) can be defined as the decay rate along that direction

\[
\nu_\alpha = \frac{\lambda_1}{\lambda_1}
\]

For learning to converge, all \(\nu_\alpha\) values must be less than 1 and greater than 0; this is necessarily true here because all eigenvalues of \(Q\) are guaranteed to be positive, and the factor of \(1/\lambda_1\) effectively sets the maximum learning speed to be less than 1. Within these limits, the larger all the \(\nu_\alpha\), the larger the decay rate, and so the total error will decrease more rapidly. It is instructive to consider two cases: 1) all eigenvalues are essentially equal, and 2) all eigenvalues are equal but one, which is very much larger. In case 1, we see from Eq. 6 that the learning speeds along all \(\alpha\) are equal and equal to 1; the geometric interpretation is that the error surface is isotropic, Fig. 4A, and learning can proceed (equally) rapidly in all directions of the error surface. In case 2, the error surface is strongly anisotropic Fig. 4B. Learning will still be fast along the (steep) direction corresponding to \(\lambda_1\), given that \(\nu_1 = 1\). However, learning along all other directions will be much slower because all remaining \(\nu_\alpha \ll 1\). Geometrically speaking, the maximum weight-update step size is constrained by the steepest direction, since small steps in weight space lead to large changes in error and can quickly lead to divergent error. Because the remaining directions are much shallower, the small weight-space step size constraint leads to much smaller decreases in error per epoch along all other directions, resulting in a sharp slowdown in the overall learning.

Hence, a narrowly distributed range of eigenvalues leads to faster learning, whereas singularly large eigenvalues that stand out from the rest broaden the range and cause a slowdown.

**MEAN-FIELD DERIVATION: LEARNING TIME GROWTH WITH SYNAPTIC INTERFERENCE.** With Eq. 6, the problem of deriving learning curves is essentially reduced to the problem of computing the eigenvalues of the correlation matrix \(Q\). Certain important features of the eigenvalue distribution can be derived from a mean-field matrix \(Q\), obtained by replacing each element of the correlation matrix with its ensemble-averaged expectation value (see APPENDIX); moreover, \(Q\) elucidates the relationship between features of HVC activity and features of the eigenvalue spectrum. As \(B\) is increased, the HVC autocorrelations (diagonal elements of \(Q\)) increase as \(B\), whereas the cross-correlations (off-diagonal) increase as a small factor times \(B^2\). The cross-correlations contribute to only the largest eigenvalue of \(Q\), causing it to scale as \(B^2\), whereas all remaining eigenvalues scale as \(B\). Therefore the largest eigenvalue of \(Q\) is a direct reflection of cross-correlations in HVC activity. Because cross-correlations in HVC activity lead to interference or cross-correlations in the use of different HVC–RA synapses in driving the song motif, the size of the largest eigenvalue equivalently reflects the degree of synaptic interference in the HVC–RA synapses. Let \(\nu_\alpha(B)\) designate the learning speed along the \(\alpha\) eigenvector of \(Q\) as a function of \(B\). The mean-field eigenvalue calculation yields (see APPENDIX)

\[
\nu_1(B) = \nu_1(1) \quad \text{in steepest direction} \ (\alpha = 1)
\]

\[
\nu_\alpha(B) = \frac{\nu_1(1)}{B} \quad \text{all other directions} \ (\alpha > 1)
\]

In other words, as \(B\) is increased, the optimal learning speeds decreases as \(1/B\) along all directions in weight space except along the direction corresponding to \(\lambda_1\), whose optimal learning speed remains unchanged. Because the cumulative initial error will generically have significant error components in several directions, the cumulative learning speeds will noticeably decrease as \(B\) is increased. According to the mean-field results above, the learning time with \(B = 2\) will be approximately twice as long as for \(B = 1\) because of synaptic interference. It is important to note, also, that the effects of increasing \(B\) on learning speed should be noticeable soon after learning has begun and the first transients (corresponding to learning along the first eigenvector) have passed, and not just toward the end of learning, where only the fine features remain to be learned. That is, the effects of multiple bursts on learning speed are manifest whether the output is learned relatively crudely or to great final precision.

This is all in good agreement with the overall decrease in learning speeds observed in the nonlinear network simulations of the last section. In the linear analysis, we see moreover that the scaling of learning time with \(B\) is an essential one (see APPENDIX): given a fixed network size, motif length, and HVC...
single-burst duration, increasing the number of bursts per HVC neuron per motif necessarily leads to a reduction in the optimal learning speed for the network, with no adjustable parameters to remove this dependency. In other words: learning with multiple bursts per HVC neuron per motif will be slower than learning with fewer bursts, independent of the HVC-RA network size, the motif length, and the single-burst duration, so long as these parameters are kept fixed while the number of bursts is varied in the comparison of learning time.

The mean-field analysis also sheds light on the identity of the eigenvector with the largest eigenvalue $\lambda_1$: it is the common mode eigenvector, with all positive entries, that corresponds to a simultaneous increase or decrease, for all parts of the motif, in the summed drive from HVC to the motor outputs. It is intuitive that this is the most "volatile" mode, leading to explosive growth of network activity. The remaining modes are differential, allowing rearrangements of the motor drive from moment to moment in the song without a large net change in the mean strength of the drive.

**NUMERICAL VERIFICATION OF MEAN-FIELD CALCULATION.** The vastly simplified mean-field derivation of the scaling of learning speed with $B$ (from the eigenvalues of $\langle Q \rangle$) neglected variance and other higher-order statistics of $Q$. To check the results of the analysis, therefore, we numerically compute the eigenvalues of $Q$ from randomly generated HVC activity matrices (see METHODS). The results are shown in Fig. 5, and agree well with the mean-field analysis. In Fig. 5a, we plot the top $300$ $B = 1$ eigenvalues, together with the top $300$ $B = 8$ eigenvalues scaled by $1/8$. All the eigenvalues for $B = 1$ form a continuum, and the scaled $B = 8$ eigenvalues sit on the same continuum, except for the top eigenvalue, which is much larger than the rest. The gap between the topmost eigenvalue and all the rest for $B > 1$ is better seen in the inset of Fig. 5a, where the largest eigenvalue scales as $B^2$, whereas the second-largest scales as $B$. This causes learning speeds to scale as $1/B$ (Fig. 6), as derived in Eq. 8.

![FIG. 5. Top 300 eigenvalues of the correlation matrix Q, divided by B, for B = 1 bursts per HVC neuron per song segment (black circles), and for B = 8 (gray circles). Inset: scaling of $\lambda_1$ (▼) and $\lambda_2$ (▲) with B, from numerical calculations. We see that $\lambda_i/B = B$, whereas $\lambda_2/B = const$. Solid lines show the same scaling, derived from $\langle Q \rangle$.](image-url)

**DISCUSSION**

**Summary**

We have built a simplified framework to analyze the learning of premotor representations in the songbird premotor circuit, given a sparse premotor drive from HVC, a set of plastic connections between HVC and RA, and a gradient learning rule that minimizes the mismatch between the tutor and pupil songs. Within this framework, we have demonstrated how temporally sparse activity allows the fast learning of premotor representations, and have quantified, in a network of linear neurons, the dependency of learning rate on the number of times an HVC neuron is active during a motif. Sparsely active HVC neurons have small cross-correlations: increasing the number of HVC bursts per motif increases cross-correlations in HVC activity, which leads to correlated changes of synaptic weights. To keep network activity from diverging because of the correlated weight changes, the maximum allowable weight-update size must be constrained; this normalization decreases the step size for other, uncorrelated weight changes that are required for learning. Thus the overall learning speed decreases with increasing numbers of HVC bursts per motif. Although our analytical description is based on linear units, the simulations (Fig. 3) of learning in nonlinear units and the heuristic...
Relation to past work

Several motor and sensory brain areas display sparse neural codes. This work augments other theoretical studies that argue in favor of the utility of sparse codes in various contexts, such as information theory, coding fidelity, decoding ease, and learning efficiency (Foldiak 1995). We have presented a quantitative analysis of the relationship between sparse representations in layers coding high-level activity (in this case, abstract sequential activity in HVC) and learning speed in a multilayer feedforward network.

Questions about training time in networks such as this one have been studied in the machine learning community, resulting in prescriptions to speed up learning by rescaling the learning rate parameter (overall weight update step size) differentially along the different eigenvectors, or by reparameterizing neuronal activities to make the error surface more isotropic. In a work closely related to this one, LeCun et al. (1991) in particular recommend that the eigenvector associated with the largest eigenvalue be subtracted from the learning updates, or that symmetrically active \{-1, 1\} neuronal units be used in the input layer instead of asymmetric \{0, 1\} units, thus reducing the anisotropy of the error surface by reducing the mean of the off-diagonal entries of the input-unit correlation matrix and so bringing the largest eigenvalue closer to the remaining ones. Given that neural firing rates are zero or positive, the activity of individual neurons in biological networks is necessarily asymmetric. Furthermore, although the learning rate parameter (overall step size) may easily be tuned at the individual synaptic level, it is not obvious how to apply separate learning rates to separate eigenvectors in a biologically plausible way, since individual synapses participate in multiple eigenvectors. Therefore, we suggest that with the use of unary HVC activity in birdsong learning, biology may have found its own solution to these very problems.

Different learning rules

We have also performed simulations of learning by stochastic weight perturbation, a reinforcement algorithm that drives learning by making stochastic estimates of the gradient without explicitly computing it; we obtain preliminary results from simulation that are qualitatively similar to the ones stated for direct gradient learning in this paper, finding that learning is faster when the number of HVC bursts is small. In fact, if biology does indeed make use of stochastic reinforcement algorithms to perform goal-related learning, the impetus to increase learning speed through sparse coding may be considerably greater because such stochastic gradient algorithms are typically much slower overall than algorithms that can directly compute gradients and move along them.

Correlations in HVC activity

In this work, each HVC model neuron can equivalently be viewed as a subpopulation of perfectly correlated (i.e., always coactive) neurons. We studied the case where each strongly self-correlated subpopulation bursts one or multiple times, but where the individual subpopulations are independent of each other. This picture is fully consistent both with the HVC data on RA-projecting neurons (Hahnloser et al. 2002), and with recurrent synfire chainlike models for the generation of sequential activity in populations of neurons.

Nevertheless, it is possible to imagine a case where the subpopulations are correlated with each other: if for example, the simultaneous bursting of 2 subpopulations in one part of the motif makes it more likely that, when they each burst again in other parts of the motif, they will burst together. Such correlations between neural subpopulations would enhance the correlations in the overall population activity at different times in the song, increasing synaptic interference compared to the independent subpopulation case, and increasing the overall anisotropy of the error surface. In this case, our qualitative results on the advantage of sparse coding for learning would be the same; in detail, the slowdown resulting from nonsparse coding would be more pronounced, from the additional contribution of intergroup cross-correlations, than described for the independent subpopulation case.

Juvenile HVC activity

Single-unit HVC recordings have been made only in adult birds, where the coding is seen to be unary (single burst of activity per neuron per motif). We wondered what the role of such extreme sparseness in HVC might be if it were present during the learning process, and found that it could confer a great advantage in terms of learning speed. On this basis, we predict that if songbirds acquire their songs under pressure to learn quickly, then sparseness of HVC activity could be integral to the learning process and should thus already be present in the HVC of juvenile birds in the early and mid sensorimotor period, instead of arising as an emergent property late in song learning.

Relevance to other sensory and motor systems

The aspect of motor learning we have explored here is the mapping of a set of sparse, high-level neural (HVC) patterns onto a denser set of low-level motor activations, in a multilayer feedforward network model of song generation. Because biological sensory and motor processing areas tend to be multilayered with important feedforward components, these results relating sparseness to learning speed in the formation of feedforward maps should be relevant in a broad range of systems. Examples of sparse coding can be seen in rat auditory cortex neurons responding to tone pips (Deweese et al. 2003); temporally and spatially sparse responses to natural scenes in ferret visual cortex (Weliky et al. 2003); sparse representations of location in hippocampal place cells; highly selective corticostriatal activity in macaque motor cortex (Turner and Delong 2000); and sparse coding of odor identity in Kenyon cells of the locust mushroom body (Perez-Orive et al. 2002; Theunissen 2003). The results of our study suggest that such sparse sensory and motor codes may facilitate the learning of feedforward representations.

On the other hand, one might wonder why, if sparse coding confers a significant advantage in terms of learning speed, are not most neural representations ultrasparse or unary? One reason is that sparse coding carries a cost: the representational
capacity of a very sparsely coded network is low. Thus, the advantages of sparse coding must be balanced against capacity constraints. Such capacity constraints may dominate, or at least play a more important role, in systems other than the zebra finch HVC. For example, songbirds that memorize much larger song repertoires may be subject to HVC volume constraints, and in these cases, we expect the coding to be sparse, for fast learning, but not necessarily unary, as in the finch.

Other implications of sparse coding

We do not intend to imply that the only role of sparse coding in the zebra finch HVC is the reduction of synaptic interference in the learning of feedforward HVC-to-RA weights. Temporally sparse coding could play an important role in mitigating the problem of temporal credit assignment in learning, which is encountered when feedback about a performance arrives significantly later than the neural activities that generated it. Moreover, sparse codes in HVC may play an important role not just in motor aspects of song learning and production but in song recognition as well (Lewicki and Konishi 1995; Margoliash 1986; Margoliash and Fortune 1992; Volman 1993).

APPENDIX

Learning curve

With linear RA neurons, we define the network equations to be $r = W h, o = A W h = X h$, where $h$ is the $N_h \times N_r$ matrix of HVC activity, $r$ is the $N_r \times N_s$ matrix of RA activity, $o$ is a $N_s \times N_t$ matrix of output activity, $A$ is the matrix of fixed weights from RA to the outputs, and $W$ is the matrix of plastic weights from HVC to RA. $N_r = T d h$ is the number of discrete time bins in the motif, where $T$ is the motif length and $d h$ is the grain size. With a change of variables $Q = h h^T$ and $x = X - X^*$, where $X^*$ is defined by $X^* h = d (X^* h$ exists if a solution exists, i.e., if the learning task is realizable), the cost function is $C = \frac{1}{2} \text{Tr}[x Q^T Q]$. Applying a gradient descent update, Eq. 4, on $C$, we have that $x \rightarrow (x - \eta A^T Q) Q$, where $\eta$ is a positive scalar that scales the overall learning step size. If each RA neuron projects to one output unit, and if the summed synaptic weights to each output are approximately equal, $AA^T$ becomes a scalar matrix that can be absorbed into $\eta$. Thus, the multilayer perceptron problem with 2 layers of weights becomes effectively a single-layer perceptron, and we have that after $n$ iterations $x^{(n)} = x^{(0)} (1 - \eta Q)^n$, so

$$C^{(n)} = \frac{1}{2} \text{Tr} \{x^{(n)} (1 - \eta Q)^n Q (1 - \eta Q^T)^n x^{(0)}\}.$$  \hspace{1cm} (A1)

In the eigenvector basis of the Hessian matrix $Q$ (eigenvalues $\{\lambda_{n-1}, \ldots, \lambda_{N_r}\}$ arranged in nonincreasing order, $\lambda_1 \geq \cdots \geq \lambda_{N_r}$), with projection of the $k$th row of $x^{(n)}$ along the $o$th eigenvector given by $x_{k o, o}$, the error after $n$ learning iterations is given by

$$C^{(n)} = \frac{1}{2} \sum_{o=1}^{N_r} (1 - \eta \lambda_o)^n \lambda_o |x_{k o, o}|^2.$$  \hspace{1cm} (A2)

Let $c_o = \lambda_{o, o} \Sigma_k |x_{k o, o}|^2$ represent the initial error along the $o$th eigenvector. The total error evolves iteratively by multiplication of the initial errors $c_o$ by a factor $(1 - \eta \lambda_o)^n$ per iteration; $\eta$ must be chosen small enough so that $|1 - \eta \lambda_o| < 1$ for each $a$, to allow error to decrease and for the learning curve of Eq. A2 to converge. Hence, $\eta$ must be less than $2/\lambda_1$, and it is easy to see that the optimal choice for $\eta$ is $\eta^* = 1/\lambda_1$ ($\eta < 1/\lambda_1$ leads to overdamped convergence, whereas $1/\lambda_1 < \eta < 2\lambda_1$ displays underdamped oscillatory convergence).

Analysis of eigenvalues

The mean-field matrix $Q$ is formed by replacing all elements of $Q$ by their ensemble-averaged expectation values (i.e., generate $Q$ and average, element by element, over several trials). Therefore, $Q = BN_2 + (B^T N_2^T N_2) I^T$, where $N_2 = \tau_d / d t$. There are only 2 distinct eigenvalues, $\lambda_1 = B N_2 + B^T N_2^T (N_2 - 1) / N_2 = B^2 N_2 N_2 / T$ (provided $N_2$ has 1 eigenvalue per column), and $\lambda_2 = B N_2 + B^T N_2^T 2 N_2 - B N_2$ (provided $T > B_0, B_0$ is a common mode eigenvector, and $\lambda_2 = B N_2 + B^T N_2^T (N_2 - 1)$ differential modes. [This effect, of an eigenvalue spectrum with one ‘large’ eigenvalue, is generic for $N \times N$ matrices with random entries of mean $a$ on the diagonal and $b$ on the off-diagonal, if $b > a N$ (Edwards and Jones 1976).] Hence the learning rate for all modes $\alpha > 1$ is given by $\nu_n = (1/B) (T N_2 / T_f - 1/B, as in Eq. 8 (Fig. 6)."

ACKNOWLEDGMENTS

We are grateful to E. Tredale for help in setting up a system to run parallel simulations, to M. Mehta and J. Werfel for comments on the manuscript, to X. Xie and M. Tresch for helpful discussions, and to the referees for suggestions.

GRANTS

This work was supported by grants from National Institutes of Health, Howard Hughes Medical Institute, and National Science Foundation (PHY 99-07949).

REFERENCES


