**Fundamental bounds on learning performance in neural circuits**

Dhrusa Venkita Raman*†, Adriana Perez Rotondo*, and Timothy O’Leary*†

*Department of Engineering, University of Cambridge, Cambridge CB21PZ, United Kingdom

Edited by Terrence J. Sejnowski, Salk Institute for Biological Studies, La Jolla, CA, and approved March 4, 2019 (received for review August 3, 2018)

How does the size of a neural circuit influence its learning performance? Larger brains tend to be found in species with higher cognitive function and learning ability. Intuitively, we expect the learning capacity of a neural circuit to grow with the number of neurons and synapses. We show how adding apparently redundant neurons and connections to a network can make a task more learnable. Consequently, large neural circuits can either devise connectivity to generating complex behaviors or exploit this connectivity to achieve faster and more precise learning of simpler behaviors. However, we show that in a biologically relevant setting where synapses introduce an unavoidable amount of noise, there is an optimal size of network for a given task. Above the optimal network size, the addition of neurons and synaptic connections starts to impede learning performance. This suggests that the size of brain circuits may be constrained by the need to learn efficiently with unreliable synapses and provides a hypothesis for why some neurological learning deficits are associated with hyperconnectivity. Our analysis is independent of specific learning rules and uncovers fundamental relationships between learning rate, task performance, network size, and intrinsic noise in neural circuits.

In the brain, computations are distributed across circuits that can include many millions of neurons and synaptic connections. Maintaining a large nervous system is expensive energetically and productively (1–3), suggesting that the cost of additional neurons is balanced by an increased capacity to learn and process information. Empirically, a “bigger is better” hypothesis is supported by the correlation of brain size with higher cognitive function and learning capacity across animal species (4–6). Within and across species, the volume of a brain region often correlates with the importance or complexity of the tasks it performs (7–9). These observations make sense from a theoretical perspective because larger artificial neural networks can solve more challenging computational tasks than smaller networks (10–15). However, we still lack a firm theoretical understanding of how network size improves learning performance.

Biologically it is not clear that there is always a computational advantage to having more neurons and synapses engaged in learning a task. During learning, larger networks face the problem of tuning greater numbers of synapses using limited and potentially corrupted information on task performance (16, 17). Moreover, no biological component is perfect, so unavoidable noise arising from the molecular machinery in individual synapses might sum unfavorably as the size of a network grows. Intriguingly, a number of well-studied neurodevelopmental disorders exhibit cortical hyperconnectivity at the same time as learning deficits (18–21). It is therefore a fundamental question whether learning capacity can grow indefinitely with the number of neurons and synapses in a neural circuit or whether there is some law of diminishing returns that eventually leads to a decrease in performance beyond a certain network size.

We address these questions with a general mathematical analysis of learning performance in neural circuits that is independent of specific learning rules and circuit architectures. For a broad family of learning tasks, we show how the expected learning rate and steady-state performance are related to the size of a network. The analysis reveals how connections can be added to intermediate layers of a multi-layer network to reduce the difficulty of learning a task. This gain in overall learning performance is accompanied by slower per-synapse rates of change, predicting that synaptic turnover rates should vary across brain areas according to the number of connections involved in a task and the typical task complexity.

If each synaptic connection is intrinsically noisy, we show that there is an optimal network size for a given task. Above the optimal network size, adding neurons and connections degrades learning and steady-state performance. This reveals an important disparity between synapses in artificial neural networks, which are not subject to unavoidable intrinsic noise, and those in biology, which are necessarily subject to fluctuations at the molecular level (22–25).

For networks that are beneath the optimal size, it turns out to be advantageous to add apparently redundant neurons and connections. We show how additional synaptic pathways reduce the impact of imperfections in learning rules and uncertainty in the task error. This provides a potential theoretical explanation for recent, counterintuitive experimental observations in mammalian cortex (26, 27), which show that neurons frequently make multiple, redundant synaptic connections to the same postsynaptic cell. A nonobvious consequence of this result is that the size

Significance

We show how neural circuits can use additional connectivity to achieve faster and more precise learning. Biologically, internal synaptic noise imposes an optimal size of network for learning a given task. Above the optimal size, addition of neurons and synaptic connections starts to impede learning and task performance. Overall brain size may therefore be constrained by pressure to learn effectively with unreliable synapses and may explain why certain neurological learning deficits are associated with hyperconnectivity. Beneath this optimal size, apparently redundant connections are advantageous for learning. Such apparently redundant connections have recently been observed in several species and brain areas.

Author contributions: D.V.R. and T.O. designed research; D.V.R. and A.P.R. performed research; D.V.R., A.P.R., and T.O. analyzed data; D.V.R. and T.O. wrote the paper; and T.O. interpreted results.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

Data deposition: Code for figure simulations in this paper is available at https://github.com/olearylab/raman_et_al_2018.

†To whom correspondence may be addressed. Email: tso24@cam.ac.uk or dvr23@cam.ac.uk.

1This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1813416116/-/DCSupplemental.

Published online May 6, 2019.
of a neural circuit can either reflect the complexity of a fixed task or instead deliver greater learning performance on simpler, arbitrary tasks.

Results

Modeling the Effect of Network Size on Learning. Our goal is to analyze how network size affects learning and steady-state performance in a general setting depicted in Fig. 1, which is independent of specific tasks, network architectures, and learning rules. We assume that there is some error signal that is fed back to the network via a learning rule that adjusts synaptic weights. We also assume that the error signal is limited both by noise and by a finite sampling rate quantified by some time interval \( T \) (Fig. 1A). In addition to the noise in the learning rule, we also consider noise that is independently distributed across synapses (“intrinsic synaptic noise”). This models molecular noise in signaling and structural apparatus in a biological synapse that is uncorrelated with learning processes and with changes in other synapses. Network size is adjusted by adding synapses and neurons (Fig. 1B).

Before analyzing the general case, we motivate the analysis with simulations of fully connected, multilayer nonlinear feedforward neural networks that we trained to learn input–output mappings (Fig. 2A). We used the so-called student–teacher framework to generate tasks (e.g., refs. 28 and 29). A “teacher” network is initialized with random fixed weights. The task is for a randomly initialized “student” network to learn the input–output mapping of the teacher. This framework models learning of any task that can be performed by a feedforward neural network by setting the teacher as the network optimized to perform the task.

The sizes of the student networks were set by incrementally adding neurons and connections to internal layers of a network with the same initial connection topology as that of the teacher (Fig. 2B and Materials and Methods). This generated student networks of increasing size with the guarantee that each student can in principle learn the exact input–output mapping of the teacher.

Learning was simulated by modifying synapses with noise-corrupted gradient descent to mimic an imperfect biological learning rule. We emphasize that we do not assume learning in a biological network occurs by explicit gradient descent. However, any error-based learning rule must induce synaptic changes that approximate gradient descent, as we show below (Eq. 1). We assume that learning must be performed online; that is, data arrive one sample at a time. We believe this captures a typical biological learning scenario where a learner gets intermittent examples and feedback.

The phenomena we wish to understand are shown in Fig. 2C and D. We trained networks of varying sizes on the same task, with the same amount of learning-rule noise. Larger networks learn more quickly and to a higher steady-state performance than smaller networks when there is no intrinsic synaptic noise (Fig. 2C). This is surprising because the only difference between the smallest network and larger networks is the addition of redundant synapses and neurons, and the task is designed so that all networks can learn it perfectly in principle. Moreover, as shown in Fig. 2D, adding intrinsic noise to the synapses of the student networks results in a nonmonotonic relationship between performance and network size. Beyond a certain size, both learning and asymptotic performance start to worsen.

The simulations in Fig. 2 provide evidence of an underlying relationship between learning rate, task performance, network size, and intrinsic noise. To understand these observations in a rigorous and general way, we mathematically analyzed how network size and imperfections in feedback learning rules impact learning in a general case.

We note that in machine learning, noise processes such as dropout and stochastic regularization (e.g., refs. 30–32) can be applied to improve generalization from finite training data. Intrinsic synaptic noise is qualitatively different from these regularization processes. In particular, the per-synapse magnitude of intrinsic noise remains constant, independent of network size or training level. Moreover, our simulations use online learning, which is distinct from the common machine-learning paradigm where data are divided into training and test sets. The implications of our paper for this paradigm are considered in SI Appendix, Regularization and Generalization Error and Online Learning and Generalization Error, where we also show that regularization can be incorporated as learning-rule noise.

Learning Rate and Task Difficulty. We define task error as a smooth function \( F(w) \) which depends on the vector \( w \) of \( N \) synaptic weights in a network. We assume that learning rules use some (potentially imperfect) estimate of this error to adjust synaptic weights.

Biologically, it is reasonable to assume that learning-related synaptic changes occur due to old information. For example, a task-related reward or punishment may be supplied only at the end of a task, which itself takes time to execute. Similarly, even if error feedback is ongoing in time, there will always be some biochemically induced delay between acquisition of this error signal and its integration into plastic changes at each synapse.

Thus, there will be a maximum rate at which task error information can be transmitted to synapses during learning, which for mathematical convenience can be lumped into discrete time points. Suppose feedback on task error occurs at time points 0 and \( T \), but not in between, for some \( T > 0 \) (Fig. 3A). If the
but each synapse is subject to internal noise in addition to noise in the learning rule (red), reflecting imperfect sensory feedback, imperfect learning rules, and task-irrelevant changes in synaptic strengths. Synaptic strengths therefore controls the learning rate during this interval as the value of $T$.

Informally, the curvature term in Eq. 1 involves balancing error surface curvature and per-synapse rate of plasticity. This is illustrated in Fig. 3.

Note that we have made no assumptions on the size of $T$, so the $O(T^2)$ term in Eq. 2 is not necessarily small. Nonetheless, we can gain useful insight for how error surface geometry affects learning by examining the other terms on the right-hand side of Eq. 2.

The learning rate, $k$, is likely to remain positive during learning if the gradient direction changes gradually as the error surface is traversed (i.e., the error surface is almost linear). In this case a high rate of plasticity—due to a high gain between feedback error and synaptic change—will result in a high learning rate. However, if the descent direction changes rapidly due to the curvature of the error surface (i.e., the surface is crinkled up), then correlation with $-\nabla F[w(0)]$ becomes a weaker predictor of learning over the entire time interval $T$. Effective learning therefore involves balancing error surface curvature and per-synapse rate of plasticity. This is illustrated in Fig. 3.d, where the length of the leaps along the error surface indicates the rate of plasticity.

We next decompose the contributions to the overall synaptic change during a learning increment. First we assume that synapses are perfectly reliable, with no intrinsic noise fluctuations affecting their strengths. In this case, we can decompose

$$F[w(T)] = (1 - kT)F[w(0)],$$

with $k < 1$. A larger positive value of $k$ implies a faster rate of learning. We can write the total change in error over the interval $T$ (Fig. 3.d) as

$$F[w(T)] - F[w(0)] = \int_0^T \langle \nabla F[w(t)], \dot{w}(t) \rangle \, dt = T \mathbb{E}_t [\langle \nabla F[w(t)], \dot{w}(t) \rangle],$$

where expectation is taken across a uniform distribution of time points in $[0, T]$, dots denote time derivatives, and angle brackets denote the (standard) inner product. Eq. 1 shows that synaptic changes, on average, must anticorrelate with the gradient for learning to occur. We can thus decompose net learning rate during the interval $T$ into contributions as follows (further details in SI Appendix, Learning Rate and Local Task Difficulty):

$$k = \frac{\|\nabla F[w(0)]\|_2}{F[w(0)]} \left[ \frac{\text{contribution from gradient}}{\text{gradient strength}} \right] + O(T^2),$$

where

$$G_F[\hat{\omega}_T] := \frac{1}{2\|\nabla F[w(0)]\|_2} \left\langle \hat{\omega}_T, \nabla^2 F[w(0)]\hat{\omega}_T \right\rangle.$$

Hats indicate unit length normalized vectors (i.e., $\hat{\omega} = \frac{\omega}{\|\omega\|_2}$) and $\hat{\omega}_T$ denotes the average synaptic change over the time interval $[0, T]$, normalized by $T$:

$$\hat{\omega}_T = \frac{\dot{w}(T) - w(0)}{T}.$$
and
−
G
Task-Relevant Plasticity
+
w
allow us to
∥
γ
describes the average degree of intrinsic noise per
[0,
[0
γ
γ
O
])
.
If we assume that each component is independent (see
γ
T
= γ
Raman et al.
∇
+ γ
A
(blue line) between
over the interval
denoting the
γ
∥
and
γ
T
γ
.
Note that a learning rule could theoretically induce task-
relevant plasticity and task-irrelevant plasticity,

). We can decompose this interpolated trajectory into task-relevant
and task-irrelevant plasticity components. The former is the initial steepest-descent
direction  − ∇F(w(0)), and the latter is the remaining orthogonal component, which corresponds to the direction n\_2 + n\_3.

![Fig. 3. Geometry of error-based learning in arbitrary networks. (A) Schematic of task error as a function of (two) synaptic weights. Learning rule receives and processes task-relevant feedback to provide direction for each synapse to move in weight space. Direction must correlate with steepest-descent direction, resulting in initial improvement of task error. If no new feedback is received over a long time period T, this initially good direction may eventually go uphill, thus becoming bad. Frequent error feedback, a less “curvy” error surface, and a good correlation with the initial steepest-descent direction make learning faster. Local task difficulty captures these factors. (B) Schematic of changes in three weights over interval [0, T]. The true weight trajectory w(t) over time (red line) is summarized by an interpolated, linear trajectory  (blue line) between w(0) and w(T). We can decompose this interpolated trajectory into task-relevant and task-irrelevant plasticity components. The former is the initial steepest-descent direction  − ∇F(w(0)), and the latter is the remaining orthogonal component, which corresponds to the direction n\_2 + n\_3.](image-url)

\[ w(T) - w(0) = -\gamma_1 T \nabla \hat{F}[w(0)] + \gamma_2 T \mathbf{n}_2 + \gamma_3 \sqrt{T} \sqrt{N} \mathbf{n}_3 \]

Note that  γ\_1 describes the average degree of intrinsic noise per synapse, whereas  γ\_1 and  γ\_2 describe components of synaptic change over the entire network. This in turn gives the following expression for the average weight velocity over the learning interval:

\[ \omega_T = -\gamma_1 \nabla \hat{F}[w(0)] + \gamma_2 \mathbf{n}_2 + \gamma_3 \frac{1}{\sqrt{T}} \mathbf{n}_3. \]

Note that the per-synapse fluctuation due to the  γ\_3 term is independent of network size. This is because \|n\_3\| = 1, which implies that the expected magnitude of the t\_th component of  \mathbf{n}_3 is \sqrt{\frac{1}{T}}. If we assume that each component is independent (see SI Appendix, Decomposition of Local Task Difficulty for justification), we can also write the magnitude of total synaptic rate of change across the network in a convenient form:

\[ \|\omega_T\|^2 = \gamma_1^2 + \gamma_2^2 + \gamma_3^2 \frac{N}{T}. \]

We see that the  γ\_3 term, which is a measure of per-synapse fluctuation magnitude, scales with the number of neurons. This stands in contrast to  γ\_1 and  γ\_2, which measure fluctuation sources over the network. Eqs. 5 and 6 allow us to rewrite Eq. 2:

\[ E[k] = \frac{-\|\nabla F[w(0)]\|_2}{F[w(0)]} \left[ -\gamma_1 + G_F \|\omega_T\| + O(T^2) \right] + O(T^2). \]

For given values of the  γ\_1 and  T, we see from Eq. 7 that  G_F controls the learning rate of a network: A higher value of  G_F[\|\omega_T\|] leads to slower or negative learning. For this reason, we refer to  G_F as the local task difficulty. Again, the O(T^2)
term may not be small. However, as mentioned, it is reasonable to assume this term cannot be controlled by the learning rule because it depends on higher-order derivatives of $F$ that synapses are unlikely to be able to compute. Therefore, we can reasonably say that learning requires the first term of Eq. 7 to be negative and ceases to occur when this term is zero. This implies

$$G_F[\hat{w}_T] \leq \frac{\gamma_1}{T(\gamma_1^2 + \gamma_2^2 + \gamma_3^2 \gamma_2^2 + \frac{\gamma_3^2}{\gamma}^2)}.$$  \[8\]

This inequality relates the intrinsic “learnability” of a task (local task difficulty, $G_F$), the rate of information on task error ($T$), the quality of the learning rule (relative magnitudes of $\gamma_1$ and $\gamma_2$), the network size ($N$), and the intrinsic noisiness of synapses ($\gamma_3$).

If inequality Eq. 8 is broken, then learning stops entirely. At some point in learning, this breakage is inevitable: As $F[w]$ approaches a local minimum, the gradient $\nabla F[w]$ approaches zero, and the Hessian $\nabla^2 F[w]$ is guaranteed positive semidefinite. At a precise minimum of error, $G_F$ becomes unbounded. This means that for a nonzero $T$, cessation of learning is preceded by an increase in local task difficulty, and learning stops just as inequality 8 above is broken.

To validate our analysis we numerically computed the quantities in Eq. 7 in simulations (Fig. 4). In the case of a linear network with quadratic error the $O(T^2)$ terms disappear, allowing us to verify that equality in Eq. 8 indeed predicts the steady-state value of $G_F$. This agreement is confirmed in Fig. 4A.

For more general error functions, we have observed that Eq. 8 is always conservative in numerical simulations: Learning stops before local task difficulty reaches the critical value, implying that the $O(T^2)$ term of Eq. 7 is usually negative. This is demonstrated in Fig. 4A and B.

In summary, we have shown that local task difficulty $G_F$ determines the learning rate as well as the steady-state learning performance of a network.

Note that in Fig. 4 (as well as in subsequent numerical simulations) we define the entire distribution of input–output pairs to be a finite set or “batch” generated from a fixed, random set of inputs. This a technical necessity that allowed us to numerically calculate true task error (i.e., the error over all inputs) and the true task gradient $\nabla F[w]$) and thus define specific values of $\gamma_i$ when applying the update of Eq. 5. We emphasize that this finite batch is considered to be the entire distribution, not a sample from a true (unknown) distribution. It is not possible to numerically specify the $\gamma_i$ in the simulations of Fig. 2, because we cannot explicitly calculate the true gradient in this case. Instead, learning occurred online by a single sample from an infinite number of potential inputs, producing a stochastic gradient estimate. Fortunately, conclusions drawn from both simulation paradigms are interchangeable. Online learning using a stochastic gradient is mathematically equivalent to adding task-irrelevant ($\gamma_2$) plasticity to the batch learning setup of Figs. 4 and 6. The amount by which $\gamma_2$ is increased depends on the task, but not the network size/architecture (see SI Appendix, Online Learning and Generalization Error for further details).

**Local Task Difficulty as a Function of Network Size.** We next show precisely how network size influences the local task difficulty and thus learning rate and steady-state performance when other factors such as noise and the task itself remain the same.

Recall that $n_2$ represents the direction in which synapses are perturbed due to error in the learning rule and other task-irrelevant changes that affect all synapses. Meanwhile $n_3$ represents the direction of weight change due to intrinsic white-noise fluctuations at each synapse. For arbitrary tasks, networks, and learning trajectories we can model these terms as coming from mutually independent probability distributions that are independent of task error $F[w]$ and its derivatives. Thus, we assume $E[n_2^2 n_3^2] = 0$, which allows us to write an expression for expected local task difficulty:

$$\mathbb{E}[G_F[\hat{w}_T]] = \mathbb{E}[n_2] = \mathbb{E}[n_3] = \mathbb{E}[n_2 n_3] = 0,$$

where $\mathbb{E}$ denotes the expected value. This term almost always decreases learning rate, as $k_{pred}$ now consistently overestimates $k$. Thus, predicted steady-state local task difficulty (e.g., A, Bottom Right, dashed line) is consistently an overestimate.

### Fig. 4. Numerical validation of learning-rate calculations in simulated neural networks. (A) Local task difficulty and mean squared error over time for a linear network (Left) with quadratic error function and a nonlinear network (Right). Local task difficulty is low when the networks are in an untrained state. As performance improves, it rises, until reaching some steady-state level (black dashed lines). We can predict this steady state a priori, exactly for the quadratic error, and conservatively for the nonlinear error, using Eq. B. Both networks are trained using a corrupted learning rule ($\gamma = [0.2, 1, 0], T = 2$; Materials and Methods). Network sizes are 200 synaptic weights (linear) and 220 synaptic weights (nonlinear, one hidden layer). (B) We use the same linear (Left) and nonlinear (Right) networks as in A. We compare the predicted value of the learning rate $k_{pred}$ (using Eq. 7 and $\gamma = \hat{\gamma}$) with the actual value, under low-noise (Top, $\hat{\gamma} = [1, 0.05, 0.05]$) and high-noise (Bottom, $\hat{\gamma} = [0.2, 0.5, 0.1]$) conditions. Dashed lines represent $k = k_{pred}$. Density plots of $(k, k_{pred})$ are shown. Two sources of discrepancy exist. First, $k_{pred}$ is calculated from the mean values $\gamma$ (Materials and Methods). Transient correlations between task-irrelevant sources of plasticity and the gradient lead to unbiased fluctuations of $\gamma$ around $\hat{\gamma}$. This is the only source of discrepancy in the linear case (Left). Thus, the density distributes equally on either side of the dashed line. In the nonlinear case (Right), there is an unknown, nonzero $O(T^2)$ term (Eq. 7) unaccounted for in calculation of $k_{pred}$. This term almost always decreases learning rate, as $k_{pred}$ now consistently overestimates $k$. Thus, predicted steady-state local task difficulty (e.g., A, Bottom Right, dashed line) is consistently an overestimate.
that determines the learning rate.

\[
\nabla_{t}(\frac{1}{2}\|\nabla F(0)\|^2) + \frac{1}{2}(n, \nabla^2 F(0)[n] + O(\|n\|^2).
\]

[11]

We can gain intuition into how Eq. 9 is derived without going through additional technical details (SI Appendix, Decomposition of Local Task Difficulty). Suppose that the weights were perturbed by a randomly chosen direction \(n\) over the time interval \([0, t]\). This gives

\[
F[w(T)] = F[w(0) + n] = F[w(0)] + \langle \nabla F[w(0)], n \rangle + \frac{1}{2}(n, \nabla^2 F[w(0)][n] + O(\|n\|^2).
\]

[11]

If the direction \(n\) is drawn independently of task error and its derivatives, then

\[E[\langle \nabla F[w(0)], n \rangle] = 0.\]

Therefore, it is the quadratic term of Eq. 11 that determines the effect of the perturbation on task error. Its contribution is

\[
e[n, \nabla^2 F[w(0)][n]] = \frac{\|n\|^2\|\nabla F[w(0)]\|^2}{N}.
\]

So the effect of random perturbations on learning grows with the ratio of the number of synapses, \(N\), Eq. 9 tells us explicitly how local task difficulty (and thus expected learning rate and steady-state performance) can be modified by changing the size of a network, provided the size change leaves \(\frac{\|\nabla^2 F[w(0)]\|^2}{N}\) unchanged. For different network architectures there are many possible ways of adding neurons and connections while satisfying these constraints. This explains why the naive size increases in Fig. 2C generically increased learning performance and provides a general explanation for enhanced learning performance in larger networks.

**Network Expansions That Increase Learning Performance.** We next give detailed examples of network expansions that increase learning rate and use the theory developed so far to compute the optimal size of a network when intrinsic noise is present. We first analyze a linear network and then apply insights from this to a more general nonlinear feedforward case.

Consider a linear network (i.e., a linear map, as shown in Fig. S.4) that transforms any input \(u\) into an output \(y = Wu\) for a matrix \(W \in \mathbb{R}^{n \times n}\) of synaptic weights. The input-dependent error of the network is taken as a simple mean square error,

\[
F[W] = \int F[W, u]P(u) du = \int \|y(u) - Wu\|^2 P(u) du,
\]

where the input vectors are drawn from some distribution \(P(u)\) (e.g., a Gaussian) and \(y(u)\) is a target output generated by a linear mapping of the same rank and dimension.

We next embed this network in a larger network with \(c_1 i\) inputs, \(c_2 o\) outputs, and a synaptic weight matrix \(W'\), for some integers \(c_1, c_2 \geq 1\). We define the total number of weights as \(N = c_1 c_2 o\). We take the transformation \(u' = Bu \in \mathbb{R}^{c_1 i}\), where \(B \in \mathbb{R}^{c_1 i \times c_1 i}\) is an arbitrary semiorthogonal matrix. Geometrically, \(B\) therefore represents the composition of a projection into the higher-dimensional space \(\mathbb{R}^{c_1 i}\) with a rotation. Note that this is an invertible mapping: If \(u' = Bu\), then \(B^{-1}u' = u\). Similarly, we can take \(y'(u') = Dy'(u) \in \mathbb{R}^{c_2 o}\), where \(D^T D = I_o\). This is illustrated in Fig. S.4A.

The expanded neural network with weights \(W' \in \mathbb{R}^{c_2 o \times c_1 i}\) has to learn the same mapping as the original, but with respect to the higher-dimensional inputs. So the network receives inputs \(u' \in Bu\) and transforms them to outputs \(y' = W'u'\), with input-dependent error

\[
F'[W', u'] = \|y'(u') - W'u'\|^2
= \|Dy'(u') - W' Bu'\|^2
= \|y'(u) - D^T W' Bu\|^2.
\]

[12]

For some weight configuration \(W'\) in an expanded network, Eq. 12 tells us that if these weights are related to the original network weights by \(W = D^T W' B \in \mathbb{R}^{n \times n}\), then we have

\[F'[W'] = F[W].\]

Explicit differentiation of \(F'[W']\) (see SI Appendix, Learning in a Linear Network) yields:

\[
\|\nabla F'[W']\|^2 = \|\nabla F[W]\|^2
\]

\[
Tr(\nabla^2 F'[W']) = 2c_2 o \int_{u \in \mathbb{E}} \|u\|^2 P(u)
= c_2 Tr(\nabla^2 F[W]).
\]

[13]
We will assume that $\nabla \hat{F}[\mathbf{w}]$ (which is a normalized vector) projects approximately equally onto the different eigenvectors of $\nabla^2 \hat{F}[\mathbf{w}]$. The latter is constant, whereas the former is a linear function of the (randomly chosen) $\mathbf{W}$, which justifies this assumption. In this case, Eq. 13 implies

$$\nabla \hat{F}[\mathbf{w}]^T \nabla^2 \hat{F}[\mathbf{w}] \nabla \hat{F}[\mathbf{w}] \approx c_2 \nabla \hat{F}[\mathbf{w}]^T \nabla^2 \hat{F}[\mathbf{w}] \nabla \hat{F}[\mathbf{w}]. \tag{14}$$

Bringing together Eqs. 13 and 14 and the formula Eq. 10 for $G_F^1$, we see that

$$G_F^1[\mathbf{w}] \approx c_2 G_F[\mathbf{w}], \tag{15a}$$

and similarly

$$\text{Tr}(\nabla^2 F[\mathbf{w}]) \approx c_2 \text{Tr}(\nabla^2 F[\mathbf{w}]). \tag{15b}$$

We can therefore write local task difficulty of the expanded network in terms of quantities of the smaller network:

$$E[G_F^1[\hat{\omega}_T]] \|\hat{\omega}_T\|^2_2 \approx c_2 G_F[\mathbf{w}] + c_2 \frac{\text{Tr}(\nabla^2 F[\mathbf{w}])}{2 \|\nabla^2 F[\mathbf{w}]\|_2} \left[ \frac{\gamma_2^2}{N} + \frac{\gamma_3^2}{T} \right]. \tag{16}$$

As long as the ratio $\frac{\gamma_3}{\gamma_2}$ is fixed, we can rewrite $\hat{N}$ in terms of $c_2$ and alter $c_2$ as an independent parameter in Eq. 16. Indeed this allows us to optimize the steady-state error of the network by changing $N$. To see how, recall that

$$E[k] = -\frac{\|\nabla^2 F[\mathbf{w}(0)]\|_2}{\|\nabla^2 F[\mathbf{w}]\|_2} \left[ -\gamma_1 + \delta \right] + O(T^2),$$

where $\delta = G_F[\hat{\omega}_T] \|\hat{\omega}_T\|^2_2 T$,

with $O(T^2) \equiv 0$ for quadratic error. Suppose the network has reached steady-state error; i.e., $E[k] = 0$. If we decreased $\delta$, then $E[k]$ would also decrease, and the network would learn further. Therefore, to derive the optimal $N^*$, we should minimize the expression for $\delta$ in $N$ (or equivalently, $c_2$). We differentiate $\delta$ in $c_2$ and note that a single stationary point exists, satisfying the equation

$$N^* = \frac{\gamma_2^2}{C \gamma_2^2 + \gamma_3^2},$$

where $C = \frac{(\nabla^2 F[\mathbf{w}(0)], \nabla^2 F[\mathbf{w}])}{\text{Tr}(\nabla^2 F[\mathbf{w}(0)])}$. Note that $C$ is unknown in general because it depends on the weight configuration of the network. However, we can take the heuristic $C \approx \frac{1}{\mathbf{F}}$, since if the gradient $\nabla^2 F[\mathbf{w}(0)]$ is uncorrelated with the Hessian $\nabla^2 F[\mathbf{w}(0)]$, then it would project equally onto each of the eigenvectors of the latter, and thus the numerator of $C$ would be the mean eigenvalue, i.e., $\frac{1}{N} \text{tr}(\nabla^2 F[\mathbf{w}(0)])$. This results in an approximate expression for $N^*$, the stationary point of $\delta$ in $N$:

$$N^* \approx \frac{T \gamma_2^2}{\gamma_3^2} (1 - \frac{\gamma_2^2}{\gamma_3^2}). \tag{17}$$

Since $\lim_{N \to \infty} \delta = \infty$, we have that $\delta(\hat{N})$ is monotonically increasing in $\hat{N}$, for $\hat{N} \geq N^*$. So if $N^* < N$, the size of the original network, then any extra redundancy hurts learning performance. If $N^* > N$, then the optimal network size $N_{opt}$ satisfies $N_{opt} = N^*$. Our formula is verified numerically in Fig. 6A, by evaluating the learning performance of transformed neural networks of different sizes, with different $\gamma_i$ values.

This estimate of the optimal network size is plotted in Fig. 5B, which shows the dependence on intrinsic synaptic noise levels. As noise decreases to zero, we see that the optimal network size grows arbitrarily. In addition, the optimal network size is smaller for a lower amount of task-irrelevant plasticity (i.e., a “better” learning rule). We validate the optimal network size estimate in Fig. 6A in simulations.
We next consider nonlinear multilayer, feedforward networks. Again, we use the student–teacher framework to generate learning tasks. We consider learning performance of a nominal and an expanded network, both with $l$ layers, and both using the same learning rule. The only difference between the two networks is the larger number of neurons in each hidden layer of the expanded network. We will use our theory to predict an optimal number of synapses (and consequently optimally hidden layer sizes) for the transformed network. As before, this size will depend on the learning rule used by the networks, which is defined by levels of task-relevant plasticity, task-irrelevant plasticity, per-synapse white-noise intensity, and frequency of task error feedback. Our predictions are validated in simulations in Fig. 6B.

We first describe the nominal network architecture. Given a vector $h^{(k-1)}$ of neural activities at layer $k-1$, the neural activity at layer $h^{(k)}$ is

$$h^{(k)} = \sigma(W^{(k)} h^{(k-1)}).$$

Here, $W^{(k)}$ is the matrix of synaptic weights at the $k$th layer. The concatenation of the synaptic weight matrices across all layers is denoted $W$ and has $N$ elements. We interchangeably denote it as a vector $w \in \mathbb{R}^N$. The function $\sigma$ passes its arguments elementwise through some nonlinearity (sigmoidal in simulations; Materials and Methods). The first layer of neurons receives an input vector $u$ in place of neural activities $h^{(0)}$. The output $y(W, u)$ is defined as neural activity at the final hidden layer.

For any given state $w$ of the nominal network, we can construct a state $\phi(w)$ of the expanded network with the same input–output properties; i.e., $y(w, u) = y'(\phi(w), u)$, where $y'$ denotes expanded network output. We do this by setting synaptic weights of the added neurons in the expanded network to zero, so the neurons do not contribute at state $\phi(w)$. Nevertheless the extra neurons can affect expanded network behavior because once they are perturbed by the learning rule they contribute to the error gradient and higher derivatives.

Suppose the nominal network is at state $w(0)$, and a learning rule picks the direction $\omega_T$ for weight change over the time interval $[0, T]$. This direction will have some local task difficulty $G^{\omega_T}[\omega_T]$. If we map the state $w(0)$ and the direction $\omega_T$ to the transformed network via the transformation $\phi$, then we can estimate local task difficulty $G^{\phi}[\phi(\omega_T)]$ of the transformed network (see SI Appendix, Learning in a Nonlinear, Feedforward Network for additional details). We get

$$E[G^{\omega_T}[\phi(\omega_T)]||\phi(\omega_T)]^2 \approx \sqrt{\frac{N}{\gamma^2} G^{\omega_T}_B[w(0)]} + \sqrt{\frac{N}{2} \frac{T}{\gamma^2} \gamma^2 F[w(0)] || F[w(0)] ||_2} \left[ \frac{\gamma^2}{N} + \frac{\gamma^2}{T} \right].$$  \tag{18}$$

We can use Eq. 18 to minimize

$$\delta = G^{\omega_T}_B[\omega_T]||\omega_T||^2 T$$

in $\tilde{N}$, the number of synapses. There is a single global minimum of $\delta$ in $\tilde{N}$, for $\tilde{N} > 0$. It satisfies

$$\tilde{N}^* \approx \frac{N^* T \gamma^2}{\gamma^2} \frac{\gamma^2}{T} \frac{N^* + 1}{\gamma^2}.$$

This gives an optimal size of the transformed network for minimizing steady-state task performance (validated in Fig. 6B). Note the dependence of $\tilde{N}^*$ on $N^*$, the number of weights in the nominal and the teacher networks. The teacher networks can generate arbitrary nonlinear mappings whose complexity grows with $N$. In this way Eq. 19 reflects the intrinsic difficulty of the task.

**Discussion**

It is difficult to disentangle the physiological and evolutionary factors that determine the size of a brain circuit (33–35). Previous studies focused on the energetic cost of sustaining large numbers of neurons and connecting them efficiently (2, 34–36). Given the significant costs associated with large circuits (3), it is clear that some benefit must offset these costs, but it is currently unclear whether other inherent tradeoffs constrain network size. We showed under broad assumptions that there is an upper limit to the learning performance of a network which depends on its size and the intrinsic volatility of synaptic connections.

Neural circuits in animals with large brains were presumably shaped on an evolutionary timescale by gradual addition of neurons and connections. Expanding a small neural circuit into a larger one can increase its dynamical repertoire, allowing it to generate more complex behaviors (37, 38). It can improve the quality of locally optimal behaviors arrived at after learning (39, 40). Less obviously, as we show here, circuit expansion can also allow a network to learn simpler tasks more quickly and to greater precision.

By directly analyzing the influence of synaptic weight configurations on task error we derived a quantity we called “local task difficulty” that determines how easily an arbitrary network can learn. We found that local task difficulty always depends implicitly on the number of neurons and can therefore be decreased by adding neurons according to relatively unrestricted constraints. In simple terms, adding redundancy flattens out the mapping between synaptic weights and task error, reducing the local task difficulty on average. This flattening makes learning faster and steady-state task error lower because the resulting error surface is less contorted and easier to descend using intermittent task error information. Biological learning rules are unlikely to explicitly compute gradients. Regardless, any learning rule that uses error information must effectively approximate a gradient as the network learns.

As an analogy, imagine hiking to the base of a mountain without a map and doing so using intermittent and imperfect estimates of the slope underfoot. An even slope will be easier to descend because slope estimates will remain consistent and random errors in the estimates will average out over time. An undulating slope will be harder to descend because the direction of descent necessarily changes with location. Now consider the same hike in a heavy fog at dusk. The undulating slope will become far harder to descend. However, if it were possible to somehow smooth out the undulations (that is, reduce local task difficulty), the same hike would progress more efficiently. This analogy illustrates why larger neural circuits are able to achieve better learning performance in a given task when error information is corrupted.

In specific examples we show that adding neurons to intermediate layers of a multilayer, feedforward network increases the magnitude of the slope (gradient) of the task error function relative to its curvature. From this we provide a template for scaling up network architecture such that both quantities increase approximately equally. This provides hypotheses for the organizing principles in biological circuits which, among other things, predict a prevalence of apparently redundant connections in networks that need to learn new tasks quickly and to high accuracy. Recent experimental observations reveal such apparently redundant connections in a number of brain areas across species (26, 27, 41, 42).

Even if neurons are added to a network in a way that obeys the architectural constraints we derive, intrinsic synaptic noise
eventually defeats the benefits conferred to learning. All synapses experience noisy fluctuations due to their molecular makeup (23–25, 43–45). These sources of noise are distinct from shared noise in a feedback signal that is used in learning. Such independent noise sources accumulate as a network grows in size, outcompeting the benefit of size on learning performance. An immediate consequence is an optimal network size for a given task and level of synaptic noise.

Furthermore, our results show that different noise sources in nervous systems impact learning in qualitatively different ways. Noise in the learning rule as well as external noise in the task error, which may arise from sensory noise or fluctuations in the task, can be overcome in a larger circuit. On the other hand, the impact of intrinsic noise in the synaptic connections only worsens as network size grows. Our results demonstrate the intuitive fact that insufficient connections impair learning performance. Conversely, and less obviously, excessive numbers of connections impair learning once the optimal network size is exceeded. This provides a hypothesis for why abnormalities in circuit connectivity may lead to learning deficits (18–21).

Our analysis allowed us to predict the optimal size of a network in theoretical learning tasks where we can specify the levels of noise in the learning rule and in synapses. Fig. 5 shows that the optimal network size decreases rapidly as the intrinsic noise in synapses increases. We speculate that the emergence of large neural circuits therefore depended on evolutionary modifications to synapses that reduce intrinsic noise. In particular, optimal network size increases explosively as intrinsic synaptic noise approaches zero. An intriguing and challenging goal for future work would be to infer noise parameters in synapses across different nervous systems and test whether overall network size obeys the relationships our theory predicts.

Materials and Methods

Full details of all simulations are provided in SI Appendix. We provide an overview here. Code related to paper is publicly accessible in ref. 46.

Network Architectures. All tested neural networks have fully connected feedforward architectures. Unless otherwise specified networks are nonlinear with multiple hidden layers. Each neuron in these networks passes inputs through a sigmoidal nonlinearity σ: R → R of the form σ(x) = 1/(1 + e−x) .

Details of the Learning Tasks. All training tasks used in simulation use the student–teacher framework. The basic setup is as follows: We first make a teacher network, which has the same basic network architecture as the student. We then initialize the teacher weights at fixed values, which are generated as follows (unless otherwise specified): At the kth layer of the teacher network, weights are distributed uniformly on the interval [−a k, a k]. Here a k is set so that the SD of weights on the layer is \( \sqrt{2/\pi} \), where i is the number of inputs to the layer. This scaling with i ensures that the magnitude of hidden layer outputs does not increase with the size of hidden layer (47).

We then specify a set \( \mathcal{U} \) consisting of 1,000 input vectors. We generate each vector \( u \in \mathcal{U} \) componentwise, randomly drawing each component \( u_k \) from the distribution \( u_k \sim \mathcal{N}(0,1) \). Note that in Fig. 2 this acts as a test set, which is not used when training the network (details in the next section). The output vector of the teacher, given an input vector u, is denoted \( y^T(u) \). The task of the student network, with weights \( w \), is then to match its output to \( y^T(u) \) for all \( u \in \mathcal{U} \). Therefore, the task error is

\[
F(w) = \sum_{u \in \mathcal{U}} \left\| y^T(u) - y^T(w, u) \right\|^2;
\]

where \( y(w, u) \) denotes the output of the student given input \( u \) and weights \( w \).

For the linear networks studied at the beginning of the section Network Expansions That Increase Learning Performance, the dimensionalities of the input and output vectors differ between the teacher and the students. However, fixed matrix transformations lift the teacher inputs/outputs into the appropriate dimensionality (Eq. 12). For all other networks, the number of network inputs/outputs is shared between the students and the teacher. Teacher and students also have the same number of hidden layers. At the ith hidden layer, each student has at least as many neurons as the teacher. This ensures that the teacher network forms a subset of each student network and therefore that each student network is theoretically capable of exactly recreating the input–output mapping \( y^T(u) \) of the teacher.

Network Training. The theoretical analysis in this paper and the simulations in Fig. 2 pertain to online learning, where training data are sampled continuously from an (infinite) distribution defined by the input–output mapping of the teacher networks. However, in cases where we needed to compute a true gradient (specifically Figs. 4 and 6), we needed to define finite distributions to numerically evaluate the true gradient. Having the true gradient allows us to precisely specify values of task-relevant and task-irrelevant components of plasticity. We emphasize (as we emphasized in the main text) that this differs from treating the finite set as a sample—or batch—from an infinite distribution, which would incur generalization issues because any finite sample will be necessarily biased. The relationship between the results of our paper and the latter scenario is described in SI Appendix, Regularization and Generalization Error.

In Figs. 4 and 6 learning is conducted on the finite input set \( \mathcal{U} \) described in the previous section. At each learning cycle, we apply the weight update

\[
w_{t+1} = w_t - \nabla_{\mathcal{T}} F(w_t) + T \hat{v}^T \hat{s} + \sqrt{T} \hat{\nu} \hat{n}^T 
\]

[21]

The parameters \( \{ \gamma \} \) and \( T \) specify the quality of the learning rule, the feedback delay, and the intrinsic, per-synapse noise (main text). \( N \) is the number of synapses in the network. We calculate the gradient \( \nabla_{\mathcal{T}} F(w_t) \) by taking the gradient of Eq. 21 using backpropagation. The normalized vectors \( \hat{v} \) and \( \hat{n} \) represent sources of task-relevant plasticity. The dynamics of the unnormalized vector \( v \) satisfy \( \dot{v} = \sqrt{\nu T} \hat{v} + \sqrt{\nu} \hat{\nu} \), where \( \nu \) is a Gaussian random variable, normalized such that \( \| \hat{n} \|_2 = 1 \).

The network in Fig. 2 C and D is conducted online from an infinite distribution. At each learning cycle, we randomly draw a single input vector \( u \) of Gaussian components; i.e., \( u_i \sim \mathcal{N}(0,1) \). We replace the term \( \nabla_{\mathcal{T}} F(w_t) \) in the weight update Eq. 21 with \( \nabla_{\mathcal{T}} F(w_t, u) \), the (normalized) stochastic gradient, where \( F(w, u) := \| y^T(u) - y^T(w, u) \|^2 \).

The overall error \( F(w) \) is then the expected error on the next input; i.e.,

\[
F(w) = \int_{\mathcal{U}} F(w, u) \rho(u) \, du,
\]

where \( \rho(u) \) is the componentwise Gaussian probability density function from which \( u \) is drawn, with support \( \Omega \). We cannot exactly calculate \( F(w) \) in this setting; we therefore use the error function described above, which is constructed from 1,000 inputs, providing an estimate of \( F(w) \).

ACKNOWLEDGMENTS. We thank Fulvio Forni, Rodrigo Echeveste, and Aoife McMahon for careful readings of the manuscript, and Rodolphe Sepulchre and Stephen Boyd for helpful discussions. This work is supported by European Research Council Grant 712016-FLXENEURO (716643).


PNAS | May 21, 2019 | vol. 116 | no. 21 | 10545


