

Supporting information for “Maximization of learning speed in the motor cortex due to neuronal redundancy”

Recurrent connections

The motor cortex contains recurrent connections [1], so we needed to confirm that neuronal redundancy maximizes learning speed even when a neural network model includes adaptable recurrent connections. In this case, neural activities are determined by

$$\mathbf{A}_{k(t)}^t = (\mathbf{I} - \mathbf{U}^t)^{-1} \mathbf{W}^t \mathbf{t}_{k(t)}, \quad (1)$$

where $\mathbf{U}^t \in \mathbb{R}^{N \times N}$ represents the recurrent connections in the t th trial. Equation (1) is derived from the stationary solution of the equation $\mathbf{A}_{k(t)}^t = \mathbf{U}^t \mathbf{A}_{k(t)}^t + \mathbf{W}^t \mathbf{t}_{k(t)}$. \mathbf{U} is also learned to minimize the squared error as follows:

$$\mathbf{U}^{t+1} = \mathbf{U}^t + B_M (\mathbf{I} - \mathbf{U}^t)^{-1} (\mathbf{I} - \mathbf{U}^t)^{-1} \mathbf{Z}^T \mathbf{e}^t \mathbf{t}_{k(t)}^T (\mathbf{W}^t)^T, \quad (2)$$

where B_M is learning rate.

We numerically calculated learning speed, and figure S1A shows the learning speeds obtained when $N=4, 10, 50,$ or 100 , and $B_M=0, 0.025, 0.05, 0.075,$ or 0.1 . The whiter the colors, the faster the learning speed. Figures S1B and S1C show the learning curves produced when $N=10, 50,$ or 100 with $B_M=0.025$ and $N=10$ with $B_M=0, 0.05,$ or 0.1 , respectively. All of the results shown in these figures support our hypothesis that neuronal redundancy maximizes learning speed even when adaptable recurrent connections are included.

Recurrent connections have their own functional roles, however. By comparing the results obtained when $B_M = 0$ and otherwise, we confirmed that recurrent connections facilitate learning speed. Additionally, figure S1D shows the variance of the learning curves when $N = 100$, showing that the more B_M increases, the smaller the variance becomes. This figure suggests another functional role for recurrent connections: recurrent connections not only facilitate learning speed but also increase the stability of learning by decreasing the variance of the learning curve.

Nonlinear activation function

There is nonlinearity in the input-output relationship for neural responses. Therefore, we needed to confirm that neuronal redundancy maximizes learning speed even when the neural network included nonlinearity. In this case, neural activities were determined through a nonlinear function $\sigma(\cdot)$:

$$\mathbf{A}_{k(t)}^t = \sigma(\mathbf{W}^t \mathbf{t}_{k(t)}), \quad (3)$$

with σ causing the update rule of \mathbf{W} to change to

$$\mathbf{W}^{t+1} = \mathbf{W}^t + B \text{diag}(\sigma'(\mathbf{W}^t \mathbf{t}_{k(t)})) \mathbf{Z}^T \mathbf{R}^T \mathbf{e}^t \mathbf{t}_{k(t)}^T, \quad (4)$$

where $\text{diag}(\mathbf{a})$ denotes the diagonal matrix in which the (i, i) th component is the i th element of vector \mathbf{a} , and $\sigma'(\cdot)$ denotes the derivative of the nonlinear function. We use a sigmoid function:

$$\sigma(z) = \frac{K}{2} (1 + \tanh(\beta z)), \quad (5)$$

where K and β indicate the upper limit of the activation function and the slope of the sigmoid function, respectively ($K = 2$ and $\beta = 1$ throughout this section).

Figures S2A and S2B show learning speed when $N=10, 50, 100,$ or 1000 and the learning curve for $N=4, 10,$ or 100 , respectively. These figures suggest that neuronal redundancy maximizes learning speed, even in this nonlinear network.

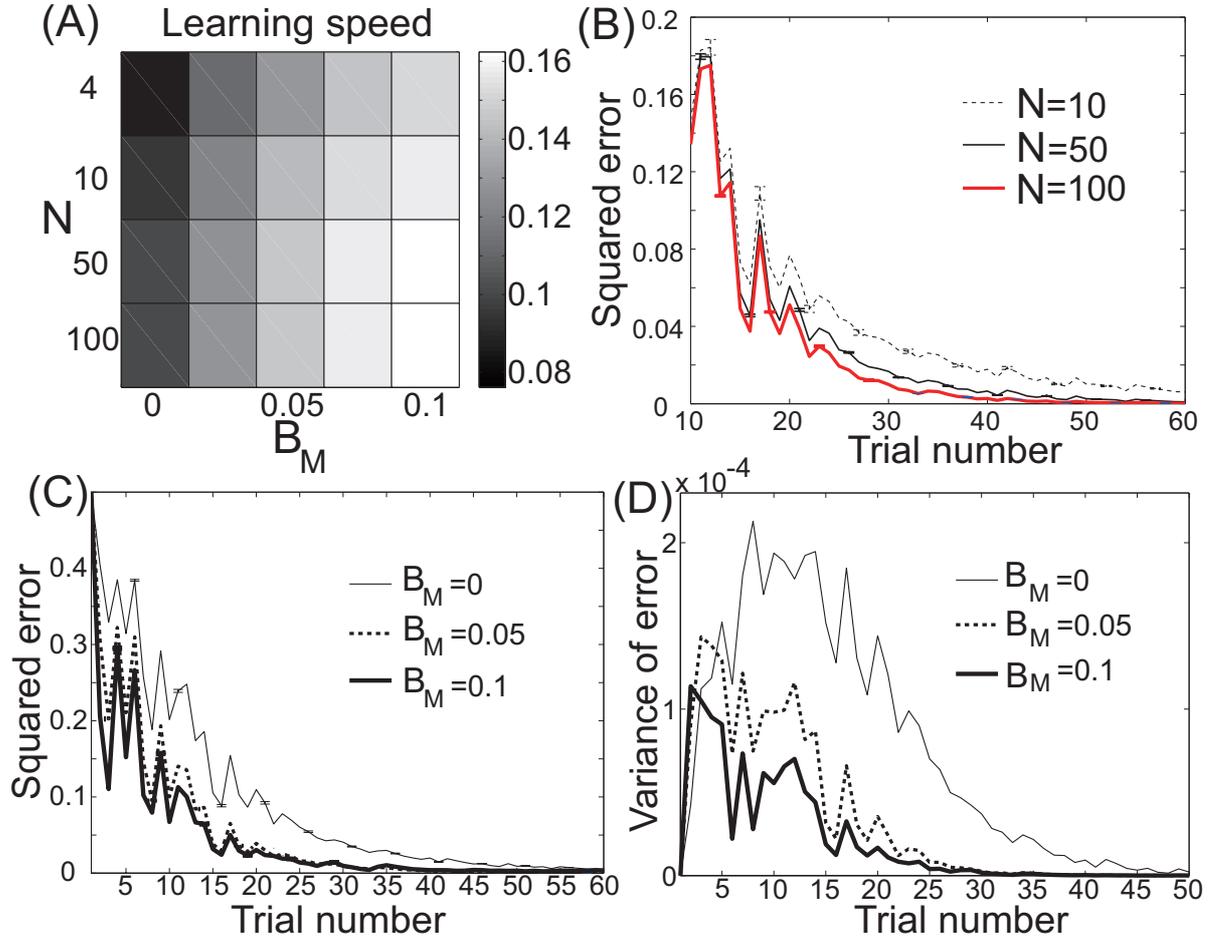


Figure S1: Relationship between learning speed, neuronal redundancy, and adaptable recurrent connections ($K = 8$). (A): Learning speed when $N = 4, 10, 50, 100$ and $B_M = 0, 0.025, 0.05, 0.075, 0.1$. The whiter the color, the faster the learning speed. (B): Learning curves obtained when $N=10, 50$, or 100 and $B_M = 0.025$. These curves show the average values of 1,000 randomly sampled sets of φ . Error bars represent the standard deviations of the errors. (C): Learning curves obtained when $B_M = 0, 0.05, 0.1$ and $N = 10$. These curves and error bars show average values and standard deviations. (D): Variance of the learning curve when $B_M = 0, 0.05, 0.1$ and $N = 100$ ($K = 8$). These variances are average values from 1,000 randomly sampled sets of φ .

Nonlinear task

This study has discussed the relationship between neuronal redundancy and learning speed using a linear rotational perturbation \mathbf{R} , but it is still uncertain whether our results hold when the constrained tasks are nonlinear. In fact, our motor system can solve nonlinear constrained tasks because the system needs to control nonlinear muscle units. We investigated whether our results hold even when a neural network needs to control nonlinear units.

In this case, motor commands are determined by

$$\mathbf{x}_{k(t)}^t = \mathbf{RZ}_M[\mathbf{M}_{k(t)}^t]_+, \quad (6)$$

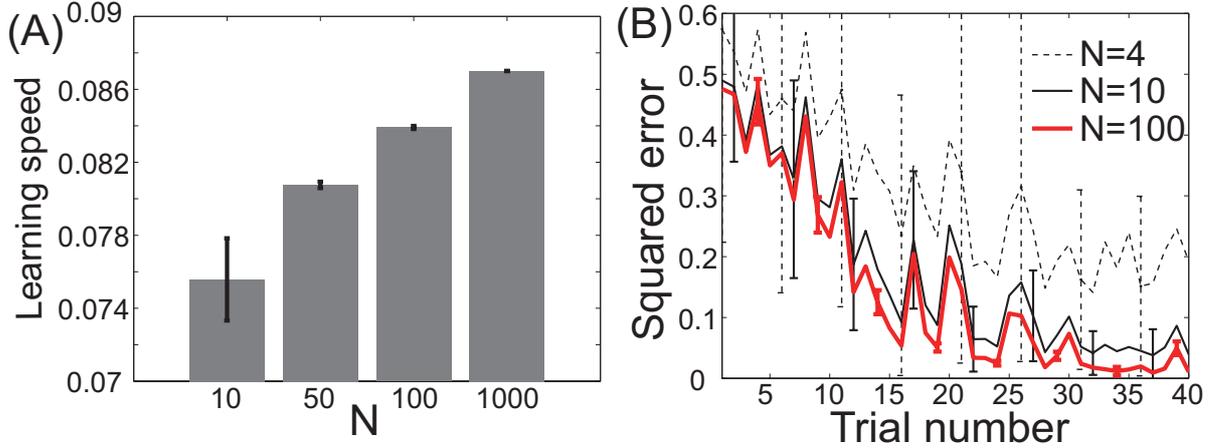


Figure S2: Relationship between learning speed and neuronal redundancy in the case of a nonlinear neural network ($K = 8$). (A): Learning speed when $N=10, 50, 100,$ and 1000 . The bar graphs and error bars depict sample means and standard deviations, both of which are calculated using the results of 1,000 randomly sampled sets of φ values. (B): Learning curves obtained when $N = 4, 10,$ or 100 . These curves and error bars show average values and the standard deviations of the errors.

where $\mathbf{M} \in \mathbb{R}^{Mu \times 1}$ is the activation of muscle units, $\mathbf{Z}_{Mu} \in \mathbb{R}^{2 \times Mu}$ is the decoder, and Mu represents the number of muscle units. The decoder is defined as $\mathbf{Z}_{Mu} = \frac{1}{Mu} \begin{pmatrix} \cos \varphi_1^m & \dots & \cos \varphi_{Mu}^m \\ \sin \varphi_1^m & \dots & \sin \varphi_{Mu}^m \end{pmatrix}$, where φ_i^m is the pulling direction of the i th muscle, the direction determined by assuming the wrist step tracking task in a midrange posture [2–4] ($i = 1, \dots, Mu$). $[\cdot]_+$ indicates the threshold linear function that satisfies $[a]_+ = a$ if $a > 0$ and $[a]_+ = 0$ in other cases (this is the function providing nonlinearity). Muscle activations are determined by

$$\mathbf{M}_{k(t)}^t = \mathbf{C} \mathbf{A}_{k(t)}^t, \quad (7)$$

where $\mathbf{C} \in \mathbb{R}^{M \times N}$ represents the cortico-muscle connections, and each component is randomly sampled from a normal Gaussian distribution. The synaptic weights \mathbf{W} are learned to minimize the squared error between \mathbf{t} and \mathbf{x} :

$$\mathbf{W}^{t+1} = \mathbf{W}^t + B[\mathbf{C}^T]_+ \mathbf{Z}^T \mathbf{R}^T \mathbf{e}_{k(t)}^t \mathbf{t}_{k(t)}^T, \quad (8)$$

where all elements of the m th row in $[\mathbf{C}]_+$ are set to 0 if the m th muscle activity equals 0. Because the nonlinear task is difficult, this task includes 10,000 baseline and 10,000 learning trials.

Figures S3A and S3B show the learning speed when $N=10, 20, 50, 100,$ or 1000 and the learning curve for $N=10$ or 100 . Although there was no significant difference in learning speed between the cases in which $N=100$ and $N=1,000$ for the 1% criterion, neuronal redundancy maximized learning speed, even in this nonlinear task.

Stochastic gradient methods

This study used the deterministic gradient descent method, which can explain the results of motor learning experiments. Although recent studies have suggested that this method is likely to be biologically plausible, other studies have suggested that stochastic gradient descent methods are more biologically plausible. We therefore confirmed that our results hold when a stochastic gradient method is used for the learning rule. We considered two representative stochastic methods: the weight perturbation and node perturbation algorithms [5].

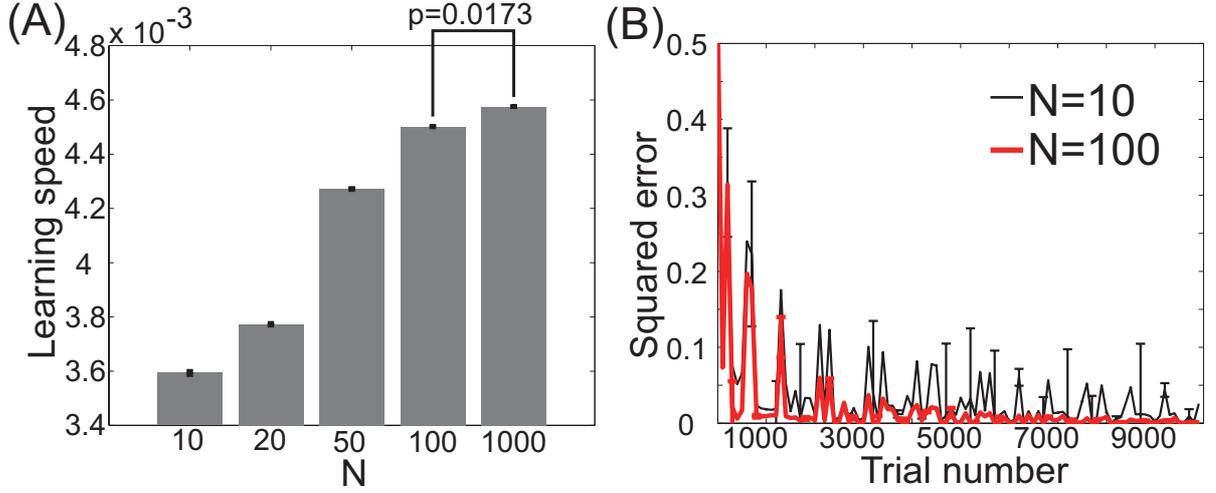


Figure S3: Relationship between learning speed and neuronal redundancy when the neural network includes nonlinear muscle units ($K = 8$). (A): The bar graphs and error bars depict sample means and standard deviations, both of which were calculated using the results of 1,000 randomly sampled sets of C values. (B): Learning curves obtained when $N = 10$ or 100 . These curves and error bars show average values and the standard deviations of the errors.

Weight perturbation

The weight perturbation algorithm estimates the gradient $\frac{\partial E}{\partial \mathbf{W}}$ by applying noise to synaptic weights. Synaptic weight \mathbf{W} is updated as

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \eta \sigma_w (E_{WP}^t - E^t) \boldsymbol{\psi}^t, \quad (9)$$

where $\boldsymbol{\psi}^t \in \mathbb{R}^{N \times 2}$ is a normal Gaussian random variable, σ_w is the standard deviation of the noise ($\sigma_w = 1$ throughout this section), E_{WP}^t is the squared error when the noise $\boldsymbol{\psi}^t$ is applied to synaptic weight \mathbf{W} , and E^t is the reference error when no noise is applied:

$$E_{WP}^t - E^t = \frac{1}{2} (\mathbf{e}_{k(t)}^t - \mathbf{RZ} \boldsymbol{\psi}^t \mathbf{t}_{k(t)}^t)^T (\mathbf{e}_{k(t)}^t - \mathbf{RZ} \boldsymbol{\psi}^t \mathbf{t}_{k(t)}^t) - \frac{1}{2} (\mathbf{e}^t)_{k(t)}^T \mathbf{e}_{k(t)}^t. \quad (10)$$

Noise averaging yields the update rule of synaptic weights:

$$\langle \mathbf{W}^{t+1} \rangle = \langle \mathbf{W}^t \rangle + B \sigma_w^2 \mathbf{Z}^T \mathbf{R}^T \mathbf{e}_{k(t)}^t \mathbf{t}_{k(t)}^T. \quad (11)$$

This is equivalent to deterministic gradient descent, except for the noise variance σ_w^2 . Because of this equivalence, neuronal redundancy also maximizes learning speed when the learning rule is weight perturbation.

We confirmed these analytical results using numerical simulations. Figures S4A and S4B show the learning speed when $N=10$, 50, 100, or 1,000 and the learning curve when $N=4$, 10, or 100 when the weight perturbation is used. Although there was no significant difference in learning speed between the cases in which $N=100$ and $N=1,000$, neuronal redundancy maximized learning speed, even in the case of weight perturbation.

Node perturbation

Node perturbation estimates the gradient $\frac{\partial E}{\partial \mathbf{W}}$ by applying noise to the output units. In the algorithm, the update rule of the synaptic weight \mathbf{W} is

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \mathbf{Z}^T \mathbf{R}^T \eta \sigma_n (E_{NP}^t - E^t) \boldsymbol{\epsilon}^t \mathbf{t}_{k(t)}^T, \quad (12)$$

where $\boldsymbol{\epsilon}^t \in \mathbb{R}^{2 \times 1}$ is a normal Gaussian random variable, σ_n is the standard deviation of the noise ($\sigma_n = 0.3$ throughout this study), E_{NP}^t is the squared error when the noise $\boldsymbol{\epsilon}^t$ is applied to the output units, and E^t is the reference error when no noise is applied:

$$E_{NP}^t - E^t = \frac{1}{2} (\mathbf{t}_{k(t)} - (\mathbf{x}_{k(t)} + \boldsymbol{\epsilon}^t))^T (\mathbf{t}_{k(t)} - (\mathbf{x}_{k(t)} + \boldsymbol{\epsilon}^t)) - \frac{1}{2} (\mathbf{t}_{k(t)} - \mathbf{x}_{k(t)})^T (\mathbf{t}_{k(t)} - \mathbf{x}_{k(t)}). \quad (13)$$

Noise averaging yields the following update rule for synaptic weights:

$$\langle \mathbf{W}^{t+1} \rangle = \langle \mathbf{W}^t \rangle + B \sigma_n^2 \mathbf{Z} \mathbf{R} \mathbf{e}_{k(t)}^t \mathbf{t}_{k(t)}^T. \quad (14)$$

This is equivalent to deterministic gradient descent, except for noise variance σ_n^2 . Because of this equivalence, neuronal redundancy also maximizes learning speed when node perturbation is used as the learning rule.

Figures S4C and S4D show the learning speed when $N=10, 50, 100$, or 1000 and the learning curve for $N=4, 10$, or 100 when node perturbation is used. Although there was no significant difference in learning speed between the cases in which $N=100$ and $N=1,000$, neuronal redundancy maximized learning speed, even in the case of node perturbation.

Synaptic decay

As described in the *Methods* section, when synaptic decay exists ($0 \ll A < 1$), our analytical calculations revealed that neuronal redundancy not only maximizes learning speed but also minimizes residual error. We confirmed this analytical result using numerical simulations. Figures S5A, S5D, and S5G show the residual error when $A = 0, 0.005$, and 0.01 , respectively. Figures S5B, S5E, and S5H show learning speed when $A = 0, 0.005$, and 0.01 , respectively. The results in these figures confirm our analytical results; neuronal redundancy not only maximizes learning speed but also minimizes residual error.

Optimal learning rate

We defined the learning rate as NB , but it remains unclear why the rate is multiplied by N . Werfel et al. reported that the optimal learning rate and speed decrease as the number of neurons increases [5], which is contradictory to our results. It should be noted that, in their model, the number of neurons N corresponds to the number of input units M . This section analytically reveals that learning rate should be $\mathcal{O}(N)$ and that our results do not contradict Werfel's results. In our study, the optimal learning rate is $\mathcal{O}(N)$, and learning speed is inversely proportional to the number of input units.

To analytically calculate the optimal learning rate and speed, each component of \mathbf{t} is assumed to be randomly sampled from a Gaussian distribution whose mean and variance are 0 and γ^2 , respectively. The optimal learning speed ideally should be investigated using all of the possible input sequences $\mathbf{t}^1, \dots, \mathbf{t}^{Trial}$. Although, in general, this calculation is analytically and numerically intractable, we can calculate the average learning speed across all the possible input sequences by calculating the average learning curve

$$\langle E^{t+1} \rangle = \int dt P(\mathbf{t}) E^{t+1} \quad (15)$$

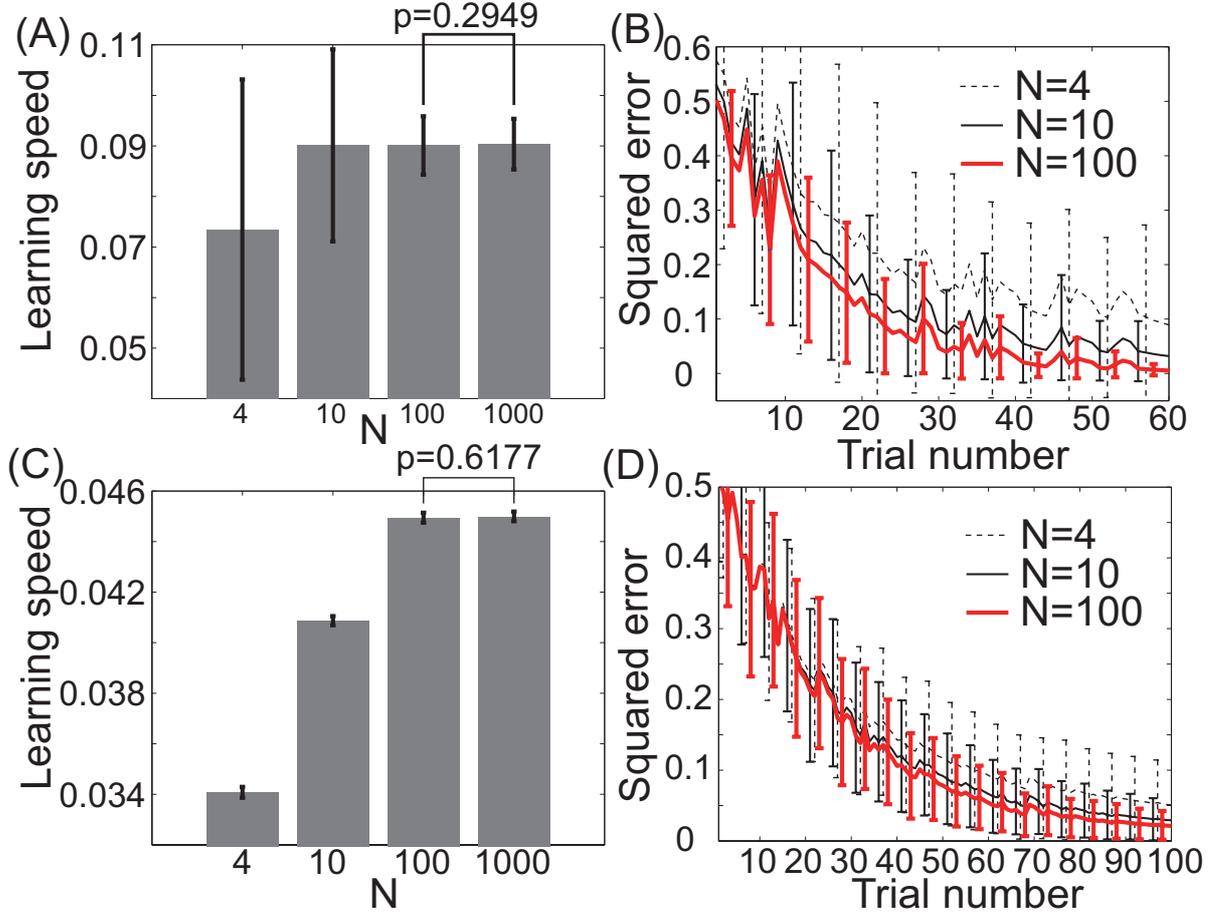


Figure S4: Relationship between learning speed and neuronal redundancy in the case of weight perturbation and node perturbation ($K = 8$). (A): Learning speed when $N = 4, 10, 100$, or 1000 , with weight perturbation as the learning rule. The bar graphs and error bars depict sample means and standard deviations, both of which are calculated using the results of 1,000 randomly sampled sets of φ . (B): Learning curves obtained when $N = 4, 10$, or 100 , with weight perturbation as the learning rule. These curves and error bars show the average values and the standard deviations of the errors. (C): Learning speed when $N = 4, 10, 100$, or 1000 , with node perturbation as the learning rule. The bar graphs and error bars depict sample means and standard deviations, both of which are calculated using the results of 1,000 randomly sampled sets of φ . (D): Learning curves obtained when $N = 4, 10$, or 100 , with node perturbation as the learning rule. These curves and error bars show average values and the standard deviations of the errors.

if $P(t)$ is Gaussian. Equation (15) can be written as

$$\begin{aligned}
 \langle E^{t+1} \rangle &= \frac{1}{2} \int dt^{t+1} P(t^{t+1}) dt^t P(t^t) (t^{t+1}) (\mathbf{I} - \mathbf{Z}\mathbf{W}^{t+1})^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^{t+1}) t^{t+1} \\
 &= \frac{\gamma^2}{2} \text{Tr}[(\mathbf{I} - \mathbf{Z}\mathbf{W}^{t+1})^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^{t+1})], \tag{16}
 \end{aligned}$$

where we use $\mathbf{x}^T \mathbf{A} \mathbf{x} = \text{Tr}[\mathbf{A} \mathbf{x} \mathbf{x}^T]$. Substituting the update rule of \mathbf{W}^{t+1} , equation (16) can be written

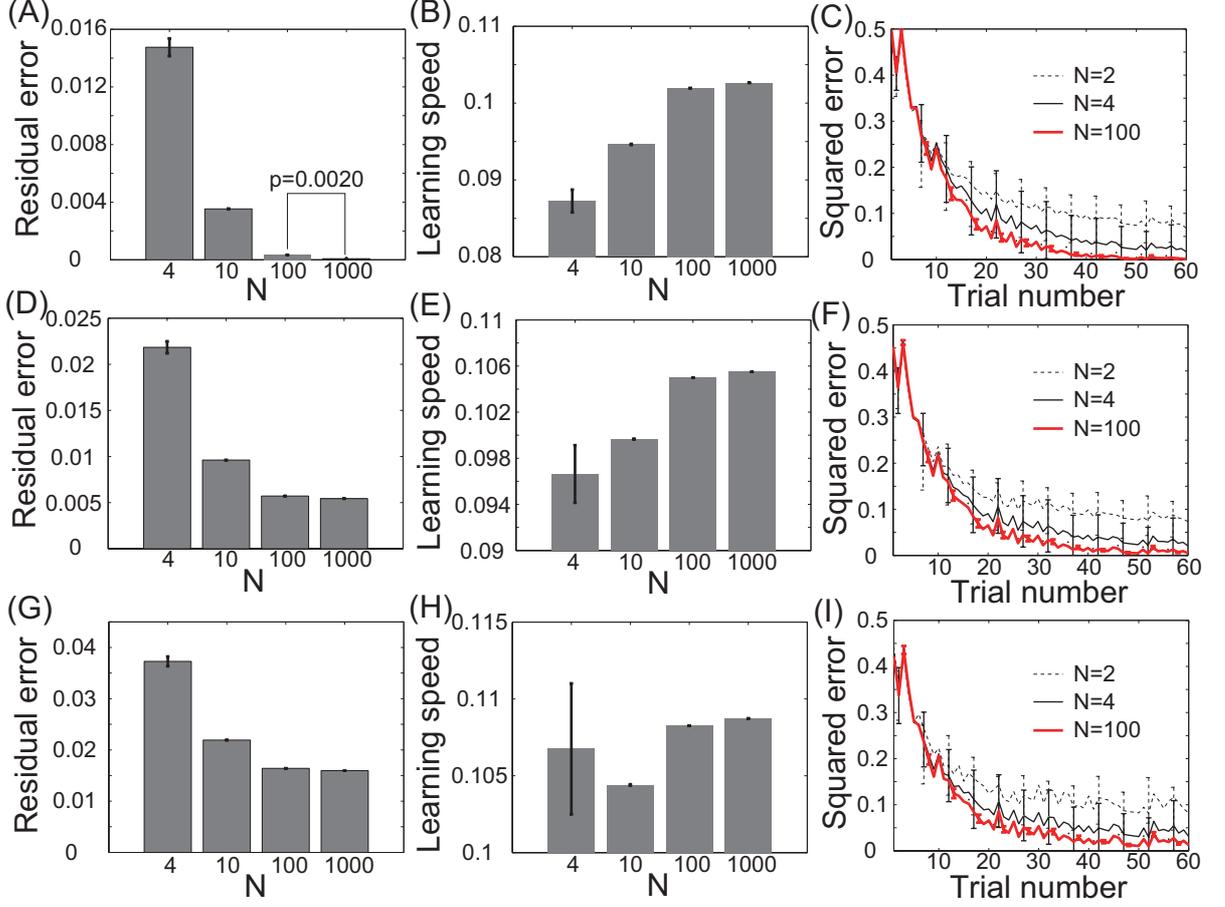


Figure S5: Relationship between residual error, learning speed, and neuronal redundancy with synaptic decay included ($K = 8$). (A): Residual error when $A = 0$. The bar graphs and error bars denote sample means and standard deviations, both of which are calculated using the results of 1,000 randomly sampled sets of φ values. (B): Learning speed when $A = 0$. The bar graphs and error bars depict sample means and standard deviations. (C): Learning curves obtained when $N = 4, 10$, and 100 and $A = 0$. These curves and error bars show average values and standard deviations. (D): Residual error when $A = 0.005$. (E): Learning speed when $A = 0.005$. (F): Learning curve when $A = 0.005$. (G): Residual error when $A = 0.01$. (H): Learning speed when $A = 0.01$. (I): Learning curve when $A = 0.01$.

as

$$\begin{aligned}
 \langle E^{t+1} \rangle &= \frac{\gamma^2}{2} \text{Tr}[(\mathbf{I} - \mathbf{Z}\mathbf{W}^t)^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^t) - 2\eta\gamma^2 (\mathbf{I} - \mathbf{Z}\mathbf{W}^t)^T \mathbf{Z}\mathbf{Z}^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^t) \\
 &+ 2\eta^2\gamma^4 (\mathbf{I} - \mathbf{Z}\mathbf{W}^t)^T \mathbf{Z}\mathbf{Z}^T \mathbf{Z}\mathbf{Z}^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^t) + \eta^2\gamma^4 \text{Tr}[(\mathbf{I} - \mathbf{Z}\mathbf{W}^t)^T \mathbf{Z}\mathbf{Z}^T \mathbf{Z}\mathbf{Z}^T (\mathbf{I} - \mathbf{Z}\mathbf{W}^t)]],
 \end{aligned} \tag{17}$$

which can be simplified as

$$(\bar{e}_{km}^{t+1})^2 = \gamma^2 (1 - 2\eta\gamma^2 \lambda_k + (M+2)\eta^2\gamma^4 \lambda_k^2) (\bar{e}_{km}^t)^2 = L_k(\eta) (\bar{e}_{km}^t)^2, \tag{18}$$

where we use $\langle \mathbf{t}\mathbf{t}^T \mathbf{A}\mathbf{t}\mathbf{t}^T \rangle = \gamma^4 (\mathbf{A} + \mathbf{A}^T) + \gamma^4 \text{Tr}[\mathbf{A}]\mathbf{I}$, $\mathbf{Z}^T \mathbf{Z} = \mathbf{V}^T \boldsymbol{\lambda}\mathbf{V}$, $\bar{\mathbf{e}} = \mathbf{V}(\mathbf{I} - \mathbf{Z}\mathbf{W})$, λ_k is the k th

eigenvalue of $\mathbf{Z}^T \mathbf{Z}$ and $\mathcal{O}(\frac{1}{N})$, and λ_k^2 is $\mathcal{O}(\frac{1}{N^2})$ ($k = 1, \dots, M$, $m = 1, \dots, M$). Learning speed is thus determined by $L_k(\eta)$ because

$$\langle E^{t+1} \rangle = \frac{\gamma^2}{2} \sum_k^M \sum_m^M (\bar{e}_{km}^{t+1})^2 = \frac{\gamma^2}{2} \sum_k^M \sum_m^M L_k(\eta) (\bar{e}_{km}^t)^2, \quad (19)$$

which means that, the smaller L_k becomes, the faster the learning speed becomes.

We can calculate the maximal learning speed with optimization of $L_k(\eta)$ with respect to η . We define the cost function as the average learning speed

$$L(\eta) = \frac{1}{M} \sum_k^M L_k(\eta) = \frac{1}{M} \sum_k^M \gamma^2 (1 - 2\eta\gamma^2 \lambda_k + (M+2)\eta^2\gamma^4 \lambda_k^2) \quad (20)$$

because k -independent η needs to be determined to minimize all the $L_k(\eta)$. Optimized η can be calculated as

$$\eta^* = \frac{1}{\gamma^4(M+2)} \frac{\sum_k^M \lambda_k}{\sum_k^M \lambda_k^2} \quad (21)$$

which means the optimized $L(\eta^*)$ is given by

$$L(\eta^*) = 1 - \frac{1}{M} \frac{1}{M+2} \frac{(\sum_k^M \lambda_k)^2}{\sum_k^M \lambda_k^2}. \quad (22)$$

Thus, learning speed decreases if $M(=T)$ increases, which is in agreement with Werfel's result. Numerical calculations support this agreement and our hypothesis: the more M grows, the slower learning speed becomes, and the more N grows, the faster learning speed becomes (figure S6A). In addition, when each component of \mathbf{Z} is $\mathcal{O}(\frac{1}{N})$, η^* is $\mathcal{O}(\frac{N}{M})$, which means that the optimal learning rate can be written as $\eta^* = NB^*$. It should be noted that neuronal redundancy maximizes learning speed even when each component of \mathbf{Z} is $\mathcal{O}(1)$ (figure S6B). In this case, η^* is $\mathcal{O}(\frac{1}{N})$. This study therefore considered a near-optimal learning rate, and our results do not contradict to Werfel's study.

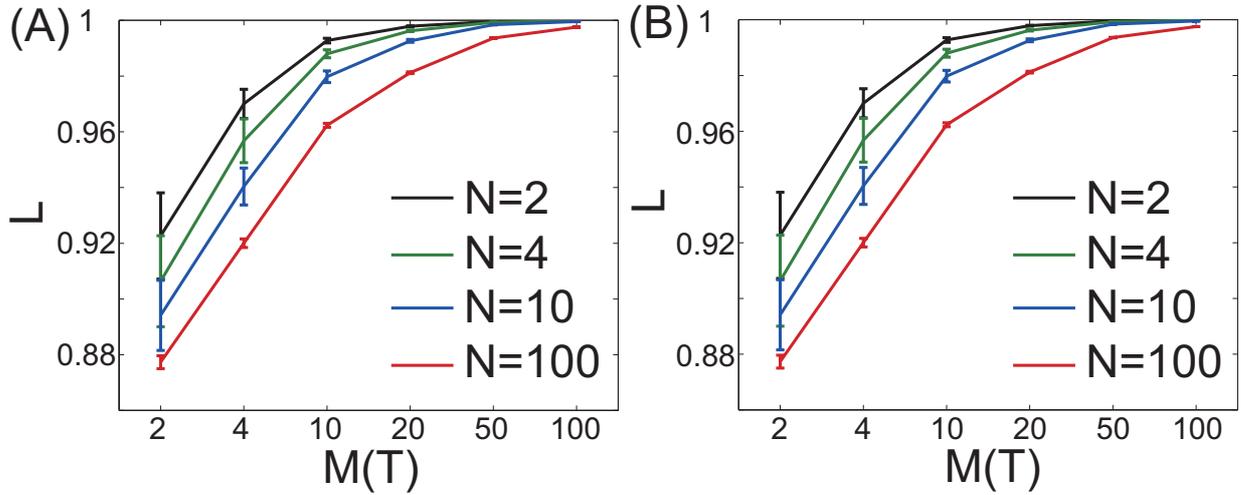


Figure S6: Calculated $L(\eta^*)$ values for $N = 2, 4, 10,$ or 100 and $T = 1, 2, 4, 10, 20, 50,$ or 100 . (A): Solid lines and error bars depict means and standard deviations of the values calculated using 100 sets of randomly sampled \mathbf{Z} values. Z_{ji} has a mean and variance of 0 and $\frac{1}{N^2}$, respectively. The smaller $L(\eta^*)$ becomes, the faster learning speed becomes. (B): Calculated $L(\eta^*)$ values obtained when each component of \mathbf{Z} is $\mathcal{O}(1)$, i.e., Z_{ji} has mean and variance that are 0 and 1, respectively.

References

1. Capaday C, Ethier C, Brizzi L, Sik A, van Vreewijk C, Gingras D. (2009) On the nature of the intrinsic connectivity of the cat motor cortex: evidence for a recurrent neural network topology. *J. Neurophysiol.* **102**: 2131-2141.
2. Hoffman D, Strick P, (1999) Step-tracking movements of the wrist. IV. Muscle activity associated with movements in different directions. *J. Neurophysiol.* **81**(1): 319-333.
3. Fagg A, Shah A, Barto A, (2002) A computational model of muscle recruitment for wrist movements. *J. Neurophysiol.* **88**(6): 3348-3358.
4. Shah A, Fagg A, Barto A, (2004) Cortical involvement in the recruitment of wrist muscles. *J. Neurophysiol.* **91**(6): 2445-2456.
5. Werfel J, Xie X, Seung S, (2005) Learning curves for stochastic gradient descent in linear feedforward networks. *Neural Comput.* **17**(12): 2699-2718.