PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Parallel synapses with transmission nonlinearities enhance neuronal classification capacity

Yuru Song[1], **Marcus K. Benna**[2]*

1 Neurosciences Graduate Program, University of California, San Diego, La Jolla, California, United States of America, **2** Department of Neurobiology, School of Biological Sciences, University of California, San Diego, La Jolla, California, United States of America

* mbenna@ucsd.edu

Citation: Song Y, Benna MK (2025) Parallel synapses with transmission nonlinearities enhance neuronal classification capacity. PLoS Comput Biol 21(5): e1012285. https://doi.org/10.1371/journal.pcbi.1012285

Editor: Jonathan David Touboul, Brandeis University, UNITED STATES OF AMERICA

Received: June 30, 2024

Accepted: March 13, 2025

Published: May 9, 2025

Data availability statement: Code to reproduce these results is available at https://github.com/london-and-tequila/parallel_synapse_capacity (for the capacity calculations of the restricted and unrestricted models) and at https://github.com/london-and-tequila/parallel_synapse_MNIST (for neural network simulations with parallel synapses). No new datasets were recorded for this work.

Funding: The author(s) received no specific funding for this work.

Abstract

Cortical neurons often establish multiple synaptic contacts with the same postsynaptic neuron. To avoid functional redundancy of these parallel synapses, it is crucial that each synapse exhibits distinct computational properties. Here we model the current to the soma contributed by each synapse as a sigmoidal transmission function of its presynaptic input, with learnable parameters such as amplitude, slope, and threshold. We evaluate the classification capacity of a neuron equipped with such nonlinear parallel synapses, and show that with a small number of parallel synapses per axon, it substantially exceeds that of the Perceptron. Furthermore, the number of correctly classified data points can increase superlinearly as the number of presynaptic axons grows.

When training with an unrestricted number of parallel synapses, our model neuron can effectively implement an arbitrary aggregate transmission function for each axon, constrained only by monotonicity. Nevertheless, successful learning in the model neuron often requires only a small number of parallel synapses.

We also apply these parallel synapses in a feedforward neural network trained to classify MNIST images, and show that they can increase the test accuracy. This demonstrates that multiple nonlinear synapses per input axon can substantially enhance a neuron's computational power.

Introduction

Synaptic connections play a crucial role in information transmission within neural systems. There is mounting evidence to suggest that cortical neurons are frequently connected by more than a single synapse. Initial findings predominantly focused on the somatosensory cortex [1–7], particularly the barrel cortex. Subsequent research has indicated that multiple connections between the same pair of (pre- and postsynaptic) neurons, which we will refer to as parallel synapses here, also exist in other brain areas, such as the hippocampus [8,9] and visual cortex [4]. The number of parallel synapses varies across different brain regions and research methodologies. For instance, in the rat barrel cortex, 4 to 6 synaptic contacts between layer 4

Competing interests: The authors have declared that no competing interests exist.

and layer 2/3 neuron pairs have been reported [7], while later in silico reconstructions have identified 5 to 25 parallel synapses in the rat somatosensory cortex [10].

Synapses not only transmit signals between neurons, but they also undergo plasticity. The observed types of plasticity include short-term plasticity [11], homeostatic plasticity [12–14] and long-term plasticity [15–17]. Even synapses from the same neuron can express different forms of plasticity [18–20]. Thus, when two neurons are connected by multiple parallel synapses, each of these synapses might have different synaptic strengths and signal transmission properties. However, the impact of these potential variations in synapse properties on the computational power of neural circuits remains to be explored. Recent studies have begun to provide an understanding of the potential benefits of parallel synapses. In [21], a phenomenological model was developed to explain the formation of parallel synapses. One stochastic model [22] aims to explain the distribution of the number of synapses between two neurons. It considers the synapses between two neurons a result of structural plasticity interacting with neuronal activity. Prior research has also approached multi-synaptic connections through the framework of Bayesian learning, proposing that multiple synapses on the same dendrite optimally estimate the input distribution and facilitate faster learning [23]. However, faster learning does not necessarily imply an increased capacity for memory or task execution. Other related work [24] explores multiple synaptic contacts in the context of temporal pattern learning, where each synapse is characterized by a distinct temporal filter. This allows a single presynaptic spike to generate postsynaptic potentials with temporal profiles composed of a mixture of the individual filters for a set of parallel synapses.

Previous works have also explored the concept of multi-weight connections between pairs of neurons in machine learning contexts [25,26]. In [25], the authors introduced a model featuring multi-weight connections between neuron pairs, with each weight representing synaptic strength through a specific type of neurotransmitter. Meanwhile, the neuron model from [26] assumes repeated axonal inputs to the postsynaptic dendritic trees. In both models, inputs from various axons interact through dendritic nonlinearity before reaching the soma.

In our work, we focus specifically on the functional benefits of parallel synapses in the absence of dendritic nonlinearities that combine different inputs [27–32], in order to understand the computational advantages they can convey without additional mechanisms that would further complicate our models and conflate our results with the known benefits of such mechanisms.

To elucidate the computational benefits of parallel synapses, we examine the memory capacity of a single neuron connected to presynaptic neurons through parallel synapses. Specifically, we consider each synapse to be parameterized by a nonlinear monotonic function, with parameters optimized to store a large number of presynaptic activity patterns. The synapses are assumed to be all excitatory, meaning that larger inputs result in increased synaptic responses for each synapse. We systematically investigate how memory capacity scales with the number of parallel synapses from presynaptic axons. Our model does not include dendritic nonlinearities, and inputs from different axons interact only at the soma. Structurally, this setup resembles a Perceptron [33–35], allowing us to compare the memory capacity of our model to that of a sign-constrained Perceptron [36–39]. Our model demonstrates a large memory capacity that increases with the number of presynaptic axons. Even a moderate number of parallel synapses can substantially enhance the memory capacity of the postsynaptic neuron, surpassing the memory capacity of the sign-constrained Perceptron.

We also extend our model to a limiting case in which the number of parallel synapses is unbounded. Here, the aggregate synaptic transmission function (summarizing the combined effect of a set of parallel synapses) becomes essentially an arbitrary monotonic function. In this scenario, we again observe a large memory capacity that increases with respect to the

number of presynaptic axons. Interestingly, despite the availability of an unlimited number of parallel synapses, the model only requires a small number of them to achieve this memory capacity.

Additionally, to test the model's ability to generalize (to inputs not used for parameter optimization), we apply parallel synapses in a feedforward neural network and train the model to perform digit classification on the MNIST dataset [40]. We show that parallel synapses can significantly increase the testing accuracy compared to a neural network without parallel synapses. This is the case even when both models have an equal number of parameters.

Results

Mathematical characterization of parallel synapses

We consider the memory capacity of a single neuron, which receives inputs from different presynaptic axons. As illustrated in Fig 1a, each axon can establish several synaptic connections with the neuron's dendrites. We assume that the input from a particular axon is the same for these parallel synapses, and that they contribute additively to the current into the soma, i.e., we neglect dendritic nonlinearities. If the synaptic transmission functions for these parallel synapses are identical, they merely serve to boost the overall dynamic range [41] of the postsynaptic current and thus could be deemed redundant. Therefore, we assume that the synaptic transmission functions can vary and be independently learned. In particular, if the synaptic transmission functions of the parallel synapses were linear, the overall synaptic transmission function of these synapses would also be merely a linear function (so that additional parallel synapses would not enlarge the class of input-output mappings that the neuron can implement). Thus we presuppose that each synaptic transmission function is nonlinear (Fig 1b), and more specifically, we model it as a sigmoid function $h_{i,j}(x_i)$ of the input activity with three parameters:

$$h_{i,j}(x_i) = \frac{(a_{i,j})^2}{1 + \exp\left(-s_{i,j}(x_i - t_{i,j})\right)}. \tag{1}$$

In Eq (1) and the following, we use i to index the i-th input axon, whose activity is denoted by x_i, and j to index the j-th parallel synapse on that axon (Fig 1c). The first parameter, $(a_{i,j})^2$, is the amplitude of the transmission function, i.e., the maximum value of the synaptic response (which we write as a square to ensure it remains nonnegative during learning). The second parameter, $t_{i,j}$, is the threshold of the transmission function, which represents the value of presynaptic input at which the postsynaptic response increases most rapidly. The third parameter, $s_{i,j}$, is proportional to the slope of the transmission function, which characterizes how steep it is at the threshold. Biological synapses on a given presynaptic neuron are typically either all excitatory or all inhibitory, which is known as Dale's law [42]. They usually do not switch between excitation and inhibition, although evidence of neurotransmitter switching has been observed [43]. Given that the response of excitatory synapses generally increases with larger input, we constrain the synaptic transmission function to increase monotonically (nonnegative slope). As a result, the range of the effective synaptic weights is restricted to positive values or zero. The definition of effective synaptic weight is illustrated in Fig C a and C d in S1 Text.

In contrast to models with nontrivial dendritic computation [27–32], which are often functionally analogous to hidden layer neural networks that nonlinearly combine multiple inputs at an intermediate stage of processing, our model features nonlinearities that only act on individual input pathways (in addition to the final somatic output nonlinearity shared by all of

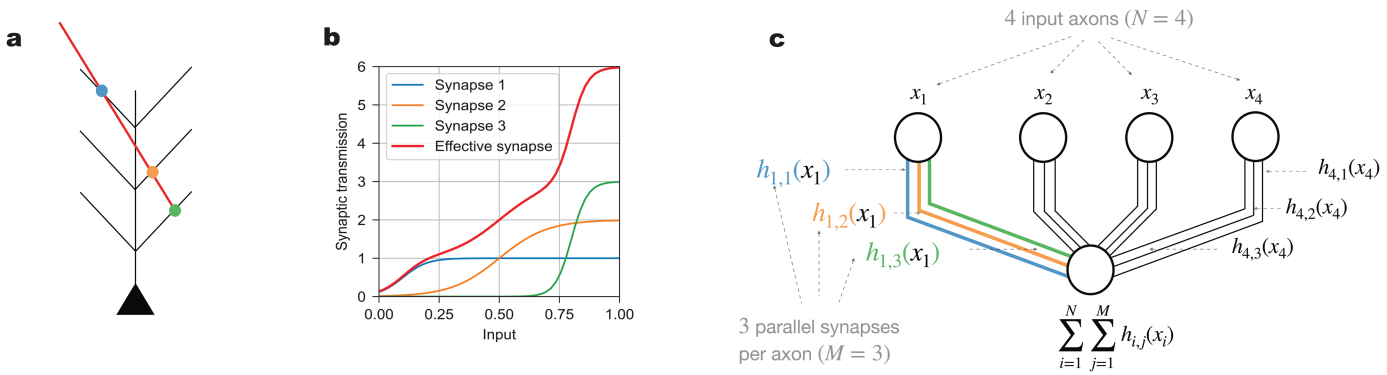

Fig 1. Neuronal connection via parallel synapses. (a). Schematic of an axon (red) making multiple synaptic contacts with a postsynaptic neuron (black). The dots indicate three parallel synapses (blue, orange, and green). **(b)**. Each synapse has its own synaptic transmission function, parameterized as a sigmoid function. The effective, aggregate synaptic transmission function (red line) from the presynaptic axon to the postsynaptic neuron additively combines the parallel synapses' transmission functions (blue, orange and green lines). **(c)**. Schematic of a neuron receiving inputs of presynaptic neuronal activity via parallel synapses. In the case shown, there are four axonal inputs ($N = 4$) and three parallel synapses ($M = 3$) per axon. Each presynaptic neuron ($i = 1, 2, 3, 4$) (upper four circles) connects to the postsynaptic neuron (lower circle) via three parallel synapses ($j = 1, 2, 3$), illustrated as a set of three parallel lines. The presynaptic neurons generally have different input activity values (x_1, x_2, x_3 and x_4), but the parallel synapses from a given presynaptic neuron convey the same input activity (e.g., the three synapses on the left have the same input value x_1 from the first presynaptic neuron). The axon in (a) with three parallel synapses can be viewed as the first presynaptic input in (c). The blue, orange and green dots in (a) would then correspond to the blue, orange and green lines in (c), with synaptic transmission functions $h_{1,1}(x_1)$, $h_{1,2}(x_1)$ and $h_{1,3}(x_1)$, respectively. These three synaptic transmission functions also correspond to the blue, orange and green lines in (b). The effective, aggregate synaptic transmission function from the first presynaptic neuronal input is $h_{1,1}(x_1) + h_{1,2}(x_1) + h_{1,3}(x_1)$, plotted as the red line in (b). Since the model has no dendritic nonlinearities, the total input to the soma of the postsynaptic neuron is the sum of the activations from all parallel synapses of all presynaptic neurons, i.e., $\sum_{i=1}^{N} \sum_{j=1}^{M} h_{i,j}(x_i)$.

https://doi.org/10.1371/journal.pcbi.1012285.g001

these models). Thus our model has the same single-neuron architecture as the (standard) Perceptron [33] in which the structure of the dendritic tree of a real neuron plays no functional role, since the currents (into the soma) contributed by each synapse are assumed to simply add regardless of the location of the synapse. The only nonlinearity that integrates multiple inputs and facilitates their interaction is the nonlinear activation function at the soma, which implements the binary classification, modeling the neuron's decision whether to spike or not in response to a certain input pattern.

Parallel synapses can substantially increase the memory capacity of a neuron

To systematically quantify the memory capacity of our model neuron, we use the classification of random patterns, a commonly used benchmark for measuring a model's capacity [44]. In our setup, successful classification is defined as correctly classifying all patterns in one dataset (Fig 2a and 2d). The patterns to be classified are random; values from different input dimensions (i.e., presynaptic neuronal activities) are independently drawn from a uniform distribution between 0 and 1. The binary labels of these patterns are also random, sampled from a Bernoulli distribution with an equal probability of being either +1 or −1. For a given number of input axons N, the probability of successful classification decreases as the problem size P grows (Fig E in S1 Text). Here we assume that the number of parallel synapses M is the same across all input axons. We want to determine the maximum problem size P a neuron can solve with 50% probability, denoted as P^*, normalized by the number of input axons N. We focus on the scaling behavior of the critical P^*/N, which describes the capacity of our model. For different N values, we test a set of problems with various sizes P to find the critical P^* We repeat the process for different numbers of parallel synapses, i.e., values of M (Fig E a-d and

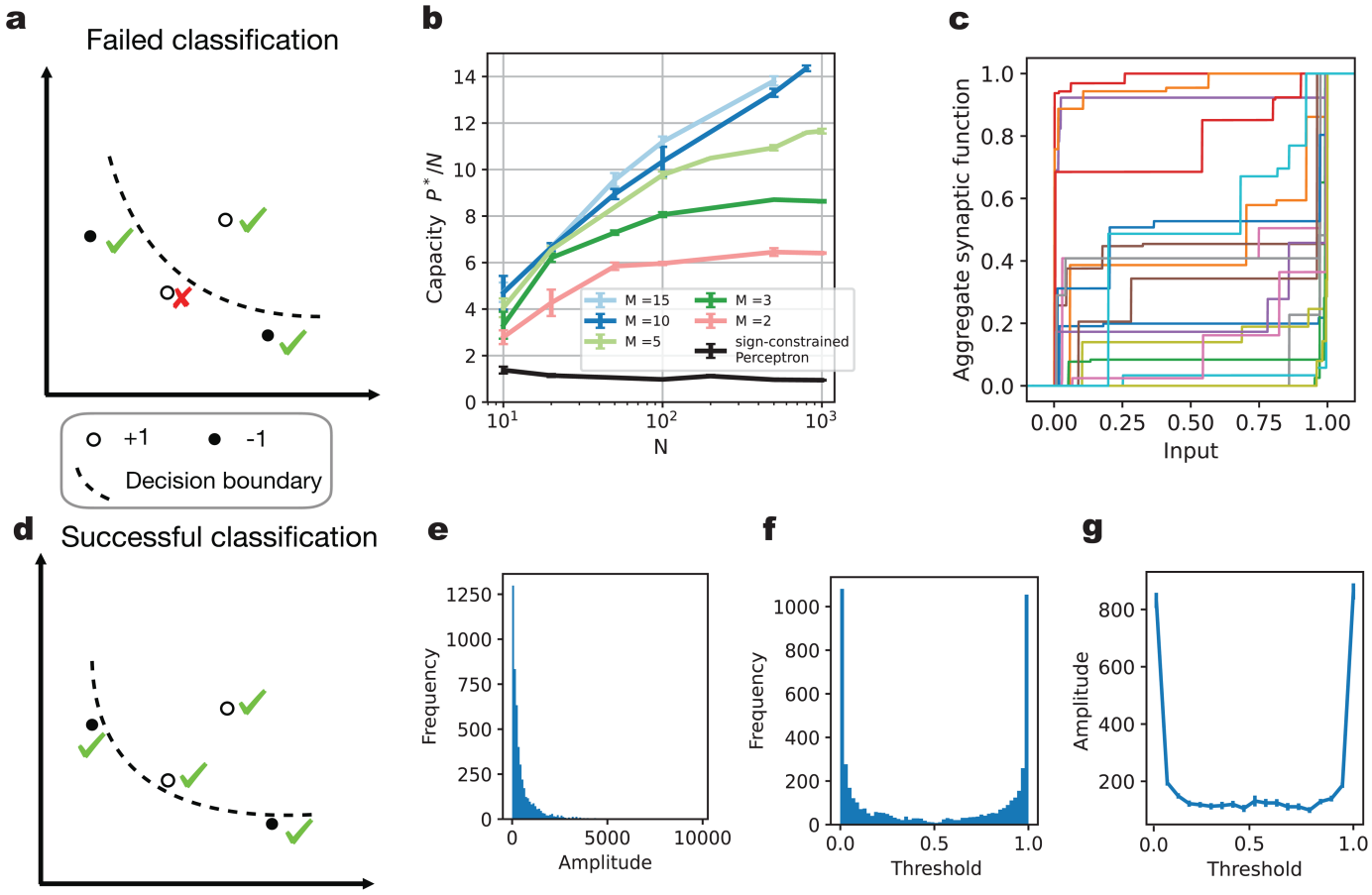

Fig 2. Increased memory capacity with parallel synapses. (a & d). Illustration of binary classification task. In a two-dimensional input space, four random patterns are labeled with +1 (white dots) or −1 (black dots) labels. The goal is to find a decision boundary (dashed line) that correctly classifies all data points. In (a), the lower left data point with +1 label is misclassified, and thus the classification problem has not been solved yet. In (d), all data points are correctly classified, and thus the problem has been solved successfully. **(b)**. The capacity P^*/N of a neuron with different numbers of input axons and different numbers of parallel synapses. With increasing N, the capacity keeps increasing, and thus P^* grows faster than linearly. **(c)**. Examples of learned effective synaptic transmission functions, normalized by their maximum value. Here, N is 1000, M is 10 and P is 11000. **(e)**. Histogram of amplitude values for all synapses in (c). **(f)**. Histogram of threshold values for all synapses in (c). **(g)**. The relationship between parallel synapses' amplitudes and their threshold values. The thresholds for all parallel synapses in (c) are divided into the same bins as in (f). We plot the average amplitude of the parallel synapses with thresholds in each bin.

https://doi.org/10.1371/journal.pcbi.1012285.g002

E h in S1 Text). The details of the gradient descent training algorithm used to optimize the model parameters and of the capacity estimation procedure for a given M and N are described in the Methods.

Our comparison benchmark is the capacity of a neuron with only linear synaptic connections of nonnegative efficacy and a step-function somatic nonlinearity, which corresponds to the Perceptron model [33] with an additional constraint on the sign of the synaptic weights. This model has a linear scaling of the critical P^* with the number of presynaptic axons, with theoretical capacity P^*/N asymptotically equal to one for large N [34–36]. Since this capacity result of the sign-constrained Perceptron is an asymptotic statement, we characterize the algorithmic model capacity for finite N numerically by running simulations using the learning rule of [36] (Fig E e in S1 Text). The detailed setup of the simulations is described in S1 Text.

As shown in Fig 2b, our model achieves a critical P^*/N substantially larger than one. Even with just two parallel synapses per axon ($M = 2$), the capacity P^*/N is about 6 with 100 different presynaptic neurons ($N = 100$). When the number of parallel synapses M increases, the capacity also grows. This shows that nonlinear parallel synapses can substantially enhance the classification capacity of a neuron. For a small number of parallel synapses per axon, such as $M = 2$, the capacity appears to saturate when the input axon number is large (Fig 2b). With more parallel synapses, the capacity increases further and the saturation of the capacity seems to occur at larger N values. We also tested our model using input patterns sampled from a Gaussian distribution $\mathcal{N}(0, 1)$, in which case the classification capacity is close to that observed with uniform inputs and follows the same trend (Figs A a and E g in S1 Text).

Notably, the learned parallel synapses have a wide variety of response profiles, as illustrated by their aggregate synaptic transmission functions (Fig 2c). For example, while most synapses have small amplitudes, there are some synapses whose amplitudes are much larger (Fig 2e). The synaptic function thresholds are more densely distributed near the edges of the input range (Fig 2f). The amplitude of the synaptic response is also related to the synapse's threshold (Fig 2g). The synapses that have large amplitude tend to be sensitive to inputs near one of the edges of the input range, where their learned thresholds are preferentially located.

We observe similar properties in the learned parallel synapses when using a Gaussian input distribution (Fig A in S1 Text). The learned synaptic thresholds (Fig A d in S1 Text) are also somewhat concentrated near the tails of the input distribution, although the amplitude versus threshold curve in Fig A e in S1 Text peaks less sharply near the edges of the sampled input range compared to Fig 2g. The difference likely arises from the input distribution, which has smooth tails in the Gaussian case, leading to synaptic responses that are less discriminative for inputs close to any particular value in these regions (as opposed to the case of a uniform input distribution whose support has clear boundaries).

Previous studies have investigated optimal synaptic weight distributions at maximum storage capacity for sign-constrained Perceptrons [45–47], and we can similarly interrogate the effective synaptic weights learned in our model. We visualized the histogram of synapse amplitudes on a logarithmic scale in Fig C b in S1 Text and the histogram of effective synaptic weights derived from the aggregate synaptic transmission function in Fig C e in S1 Text. Here, the effective synaptic weight of a set of parallel synapses is defined as the ratio of the aggregate synaptic response to the axonal input. Because the aggregate synaptic functions in our model are nonlinear, the effective synaptic weights vary with the magnitude of the presynaptic input (Fig C d in S1 Text).

Considering that synaptic wiring incurs resource costs and impacts neuronal organization [48–51], we can introduce synapse pruning during the training of our model. While the classification capacity decreases with more aggressive synapse pruning, it remains higher than that of the sign-constrained Perceptron (Fig D in S1 Text). Since noise impacts the reliability of stored patterns and the corresponding storage capacity, we also examined the case where the input patterns are corrupted by noise. Under these conditions, the classification capacity is again reduced compared to the case without noise, particularly when using the strict definition of success that requires all (noisy) patterns to be correctly classified (Fig B in S1 Text), but the model continues to outperform the baseline (sign-constrained) Perceptron.

Achieving high capacity only requires few synapses per axon

From the results of the previous section, it is clear that the presence of several nonlinear parallel synapses can increase the neuronal classification capacity. Does this mean that more

parallel synapses are always better? Conversely, we can approach this from a theoretical perspective by asking: In the extreme case in which the number of parallel synapses on each input axon is unlimited, how large can the memory capacity become? To answer this question, we parameterize the effective aggregate transmission function of a set of parallel synapses as a staircase-like function of the axonal input (see Methods). The height of each step can take an arbitrary nonnegative value, and steps can appear at any value of the input. When the problem size is P, each aggregate transmission function for an axon can take at most P different values on this dataset, so with P steps we obtain a very flexible parametrization. For reasons of biological plausibility, we again constrain each aggregate transmission function to be monotonically increasing (by demanding nonnegative step sizes). In order to clearly differentiate them, we will refer to the model with a limited number of parallel synapses on each input axon as a restricted neuron, and the model with an unlimited number of parallel synapses on each input axon as an unrestricted neuron. Again, we use binary classification problems with random patterns and random labels to assess the capacity of this system.

In the unrestricted neuron model, synaptic transmission functions involve step functions with discontinuities described by a large number of parameters. We have developed an alternative training algorithm for this model. This algorithm is based on a straightforward intuition: we use a loss function that increases the function values at input points with positive labels and decreases the function values for those with negative labels, pushing the model closer towards correct classification. This approach can be applied independently to each input dimension, and data points that remain misclassified can then be assigned increased importance for subsequent iterations. A detailed description of the algorithm is provided in the Methods.

Since there is no interaction between synaptic currents until the final somatic nonlinearity, we can construct a loss function for each input dimension (axon). We denote the value of the aggregate transmission function for the μ-th data point and the i-th axon by $I_{i,\mu}$, which depends on the input $x_{i,\mu}$. If the label y_μ of this data point is +1, our algorithm aims to ensure a sufficiently large value of the synaptic function, $I_{i,\mu}$ at $x_{i,\mu}$. On the other hand, if y_μ is –1, the algorithm adjusts $I_{i,\mu}$ to be small. To achieve this we initially define our cost function as $-\sum_{\mu=1}^{P} y_\mu I_{i,\mu} + \lambda \sum_{\mu=1}^{P} (I_{i,\mu})^2$, where we include a regularization term with parameter λ to discourage unrealistically large currents into the soma. Since some data points may be more challenging to classify than others, we introduce an importance weight w_μ for the loss of each data point. In summary, our loss function can be expressed as the following objective

$$\min_{I_{i,\mu}} -\sum_{\mu=1}^{P} w_\mu y_\mu I_{i,\mu} + \lambda \sum_{\mu=1}^{P} (I_{i,\mu})^2. \tag{2}$$

This single-axon objective can be used in an iterative algorithm to solve the classification problem. During each iteration, we first find a solution to Eq (2) for all axons i (see Methods for details). Next, we evaluate the model's predictions for the binary classification task. For misclassified data points, we increase their importance weights by one unit, i.e., $w_\mu^{(t+1)} \leftarrow w_\mu^{(t)} + 1$. We then repeat this procedure until all data points are correctly classified or the maximum number of iterations is reached.

The unrestricted neuron model also has a memory capacity larger than that of the sign-constrained Perceptron (Fig 3a). This capacity was determined using a procedure similar to that for the restricted neuron, where the problem size P is incrementally increased to identify the critical P^*/N (Fig E l in S1 Text). The scaling of the critical P^*/N value appears to increase

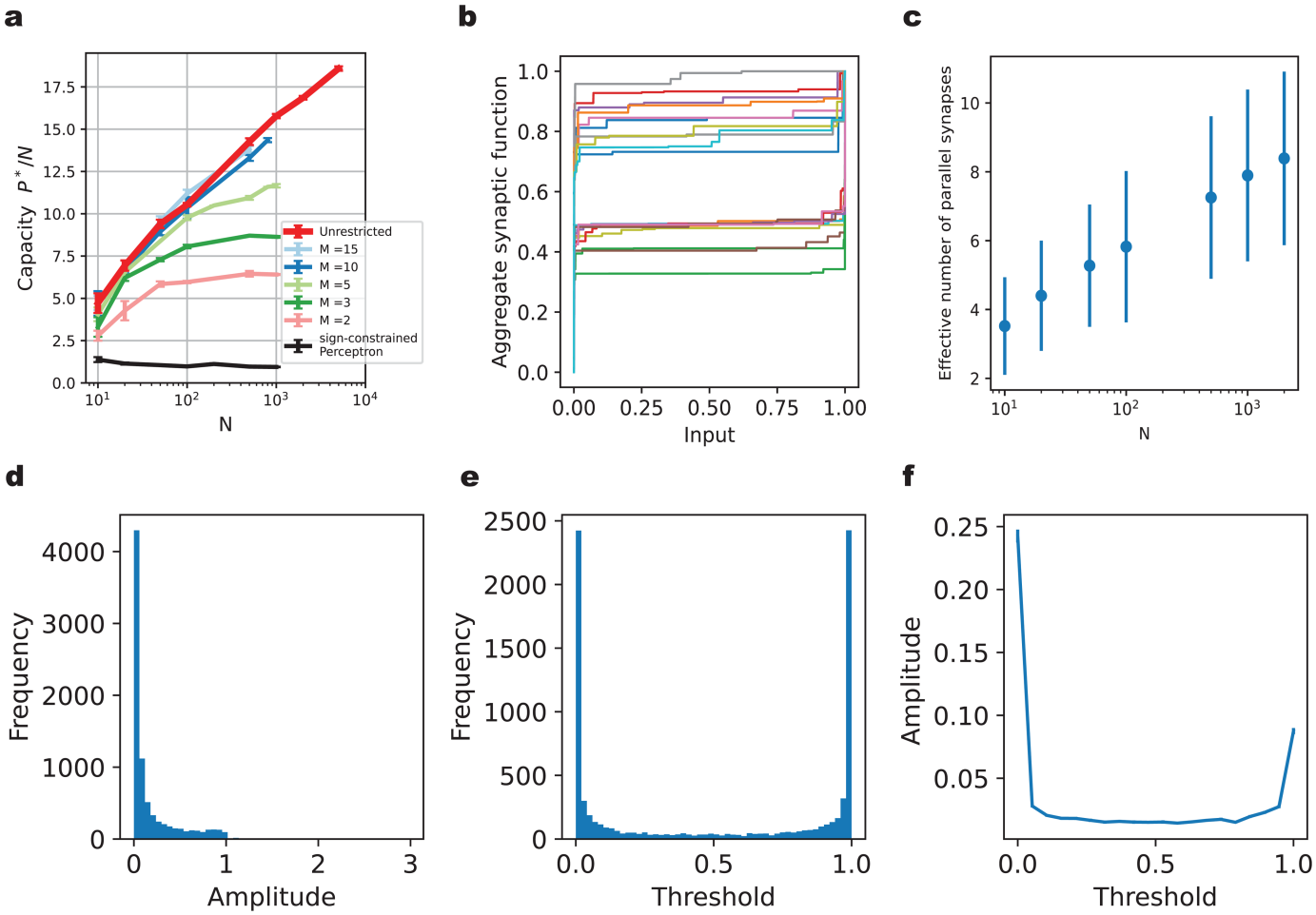

Fig 3. High memory capacity of the unrestricted model only requires few parallel synapses per axon. **(a)**. The capacity P^*/N of the unrestricted neuron (red line) compared to restricted neurons with different numbers of parallel synapses per axon. **(b)**. Examples of learned aggregate synaptic transmission functions for the unrestricted neuron, normalized by their maximum value. Here, $N = 1000$ and $P = 16000$. **(c)**. The effective number of parallel synapses on each input axon for the unrestricted neuron model, which varies with the number of input axons N. The problem size is near the critical number of patterns P^*. The effective number of parallel synapses is defined by counting the synapses with amplitude larger than 1/1000 of the maximum amplitude. **(d)**. Histogram of the amplitudes of all effective synapses from the example shown in (b) with $N = 1000$ and $P = 16000$. **(e)**. Histogram of the thresholds of all effective synapses from (b). **(f)**. Relationship between the amplitude of a synapse and its threshold. The thresholds for all effective parallel synapses in (b) are divided into the same bins as in (e). We plot the average amplitude of the synapses with thresholds in each bin.

https://doi.org/10.1371/journal.pcbi.1012285.g003

with N (approximately in proportion to $\log N$), rather than plateau when N is large. However, for the input dimensions (number of axons) relevant to realistic neural systems, say $N \sim 1000$, the unrestricted neuron has a capacity comparable to the restricted neuron model with $M = 10$ or $M = 15$. Interestingly, the unrestricted neuron does not utilize all available potential synapses (Fig 3b and 3c). The majority of the utilized parallel synapses have thresholds near the edges of the input range (Fig 3e). Also, the majority of the utilized parallel synapses have a small amplitude (Fig 3d) and those parallel synapses with large amplitude are often tuned to the edges of the input range (Fig 3b and 3f), in agreement with the restricted model, but with a bias towards the lower end of the input range (Fig 3f). This bias arises from the regularization and the fact that we constrained the synaptic function to be nonnegative. The histogram

of the effective synaptic weights derived from the aggregate synaptic transmission functions is shown in Fig C f in S1 Text.

Enhanced classification accuracy through parallel synapses in neural networks

So far, we have used random patterns to evaluate the memory capacity of a single neuron equipped with parallel synapses. However, real-world problems often involve structured input distributions, and require generalization to unseen patterns. To evaluate the model's ability to generalize, we incorporate parallel synapses into a feedforward neural network. We train this neural network to perform a classification task using the MNIST dataset, which consists of handwritten digits [40].

Specifically, we use a two-layer fully connected neural network, with variable number of neurons in the hidden layer (denoted as D_{hidden}, between 5 and 30) and $D_{\text{out}} = 10$ neurons in the output layer, reflecting the one-hot encoding of the labels to be predicted. The input layer consists of $D_{\text{in}} = 28 \times 28$ neurons, corresponding to the pixels of the images. Each neuron in the hidden layer connects to each output layer neuron via a fixed number (M) of parallel synapses (Fig 4a). The connections between input layer and hidden layer are established through standard, linear synapses. We use smooth rectified linear (Softplus) activation functions for neurons in the both hidden layer. Parallel synapses are applied exclusively between the hidden and output layer neurons, adding $D_{\text{hidden}} D_{\text{out}} (3M - 1)$ extra parameters compared to a network with the same architecture but only single, linear synapses. We do not add parallel synapses between the input and hidden layers, because the input dimensionality is high (28×28), and this would therefore result in a significantly larger number of additional parameters. Overall, such a two-layered neural network with parallel synapses has a total of $(D_{\text{in}} + 1) D_{\text{hidden}} + (3MD_{\text{hidden}} + 1) D_{\text{out}}$ parameters, including the bias terms for the activities of each neuron in the hidden and output layers. For example, with $D_{\text{hidden}} = 20$ and $M = 3$ parallel synapses, the network has a total of 17510 parameters.

For comparison, we also train a conventional two-layered neural network with $D_{\text{hidden}} = 22$ neurons in the hidden layer on the same MNIST task, using single, linear synapses for all neuronal connections (Fig 4b). This network is referred to as '$D_{\text{hidden}} = 22$, linear synapse' in Fig 4c, and has roughly the same parameter count as a network configuration with $D_{\text{hidden}} = 20$ and 3 parallel synapses, totaling 17500 learnable parameters. Similarly, for each of the $D_{\text{hidden}} = 5,\ 10$ and 30 cases, we train a corresponding network with linear synapses that has a roughly equal number of parameters. The total parameter count for both network types is detailed in Table A in S1 Text. It's important to note that because our parallel synapses have monotonically increasing transmission functions, i.e., are excitatory, we also apply this constraint to the linear synapses in the hidden to output layer weights of the standard (comparison) network, which thus also have to be nonnegative. We assess model performance by the testing accuracy on the held-out MNIST data.

As illustrated in Fig 4c, networks with parallel synapses ($D_{\text{hidden}} = 5, 10, 20$ and 30, with 3 parallel synapses), show higher accuracy after 50 epochs of training compared to their counterparts with linear synapses and an equivalent number of parameters. Specifically, the improvement in accuracy ranges from 0.75% to 2.0%. This shows that multiple nonlinear parallel synapses can indeed improve the generalization performance on a supervised learning task with nontrivially structured data. The testing accuracy values for each network are listed

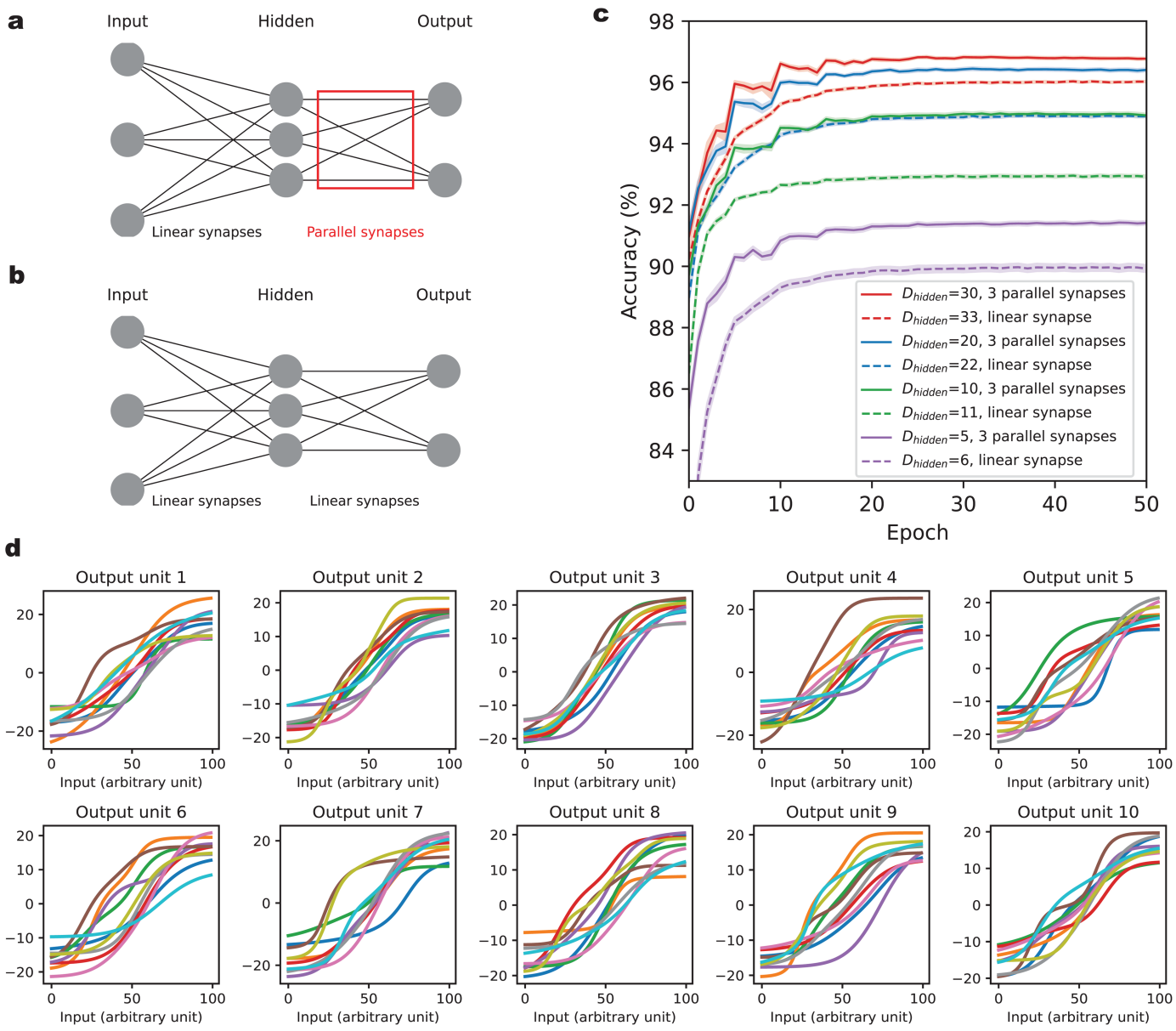

Fig 4. Enhancing classification accuracy for hand-written digits using parallel synapses. **(a)**. The feedforward network consists of an input layer, a hidden layer and an output layer. There are 784 and 10 neurons in the input and output layers, respectively, corresponding to the image pixels and the digit classes to be predicted. The number of neurons D_{hidden} in the hidden layer varies across different models. The connections between input and hidden layers are regular, linear synapses (no multiple parallel connections). The connections between hidden and output layers are nonlinear parallel synapses, with 3 parallel synapses per connection. **(b)**. The comparison benchmark model of a standard feedforward neural network with only linear synapses. There are 784 and 10 neurons in the input and output layers, respectively. The number of neurons in the hidden layer is slightly larger than in (a), in order to keep the total number of parameters in the two networks approximately the same. **(c)**. Classification accuracy on held-out data for neural network models with parallel synapses and standard neural networks with linear synapses for the MNIST dataset. The x-axis indicates training epochs. The y-axis corresponds to the percentage of correctly classified test data points. Each model is initialized with 20 distinct random seeds, with results averaged across these seeds. **(d)**. Learned aggregate synaptic transmission functions for an example network with $D_{\text{hidden}} = 10$. For each pair of hidden neuron and output neuron, there are 3 parallel synapses connecting them. The aggregate synaptic transmission functions are grouped together according to the output neuron to which they connect, each corresponding to one panel. The x-axis is the percentile of the input from the corresponding hidden unit.

https://doi.org/10.1371/journal.pcbi.1012285.g004

in Table B in S1 Text. In Fig 4d, we show the learned aggregate synaptic transmission functions of an example network, grouped by the output neuron to which they connect. Compared to the models trained on the random pattern classification task (Figs 2c and 3b), the aggregate synaptic functions learned in the MNIST task have less steep slopes and smoother transitions. In the random pattern classification task, the model is trained to maximize the number of memorized patterns drawn from a uniform distribution, which results in learned synaptic functions exhibiting sharper transitions and greater sensitivity near the edges of the input range. In contrast, for the MNIST task the parallel synapses have to deal with more general input distributions (Fig F in S1 Text), and the improved generalization performance of the trained model appears to rely on smoother aggregate synaptic functions with thresholds less concentrated near the boundaries of the input distribution.

Discussion

In this study, we have demonstrated that multiple nonlinear synapses between two neurons can enhance the memory capacity compared to a single linear connection. We have modeled the nonlinear transmission functions of each of these parallel synapses as a simple sigmoidal nonlinearity. Given different parameters (e.g. thresholds of the sigmoids), this allows the parallel synapses to contribute in a non-redundant fashion to the required computations, and thus may offer a functional explanation of the observation of multiple synaptic contacts between a pair of neurons in terms of enhanced classification capacity. The synaptic nonlinearity is essential here, since linear parallel synapses would be degenerate in the sense that their combined effect on the current into the postsynaptic neuron would still be linear.

Unlike in models incorporating nontrivial dendritic computation [27–31] or hidden layer neural networks [25,26,32], the nonlinearities in our model exclusively act on individual inputs. Consequently, our model's architecture mirrors a Perceptron [33]. The sole nonlinearity that integrates multiple inputs and allows interactions between them is the step-like activation function at the soma, which implements the binary classification.

Specifically, we have systematically investigated the increase in memory capacity using random binary classification tasks, observing an increase of the capacity P^*/N with the number of input axons. We showed that even a small number of parallel synapses between two neurons leads to much larger capacity than in the classical sign-constrained Perceptron model [35–37,39,44,45]. Furthermore, we examined a model with an a priori unlimited number of nonlinear synapses between two neurons. In this case the growth of the number of stored memory patterns with the number of input axons showed a superlinear behavior (similar to the previous model with sufficiently many parallel synapses). However, we found that only a relatively small number of potential synapses were utilized. The effective number of parallel synapses fell within the range observed in the cortex [4,5,9,10,21], suggesting that in the presence of a fixed number of available input axons, neurons in the brain may form parallel synapses until diminishing returns limit their capacity improvement. We also evaluated the generalization ability of a neural network equipped with parallel synapses. Using the MNIST dataset, we showed that parallel synapses enhance the test classification accuracy compared to networks with linear synapses, even when both models have the same number of parameters.

A fundamental assumption in our model is the nonlinearity and learnability of the synaptic transmission function. Synaptic transmission mechanisms involving stochastic vesicle release, the resulting resource depletion, and short-term synaptic plasticity are unlikely to result in postsynaptic currents that grow exactly linearly with the presynaptic firing rate. However, relating our models to these mechanisms will require a more biophysically detailed implementation involving spiking neurons. Also, while certain synaptic parameters such as

the physical size of synapses, which is often considered as a proxy for the overall synaptic efficacy, can certainly change as a consequence of long-term potentiation or depression events, it is less clear to what extent more subtle synaptic properties, corresponding to the detailed shape of our synaptic transmission functions, can be reliably modified during biological learning.

The existing literature has studied the impact of nonlinearities in the context of dendritic computation or hidden-layer neural networks. In [32], the effects of nonlinear activation functions on model capacity are explored in a tree-like committee machine, whose structure resembles dendritic branching. A hidden unit in this network receives inputs from multiple units that are combined nonlinearly, with each input taking distinct values, analogous to independent axons. The computational benefits of nonlinear dendritic activation functions in a two-layer neuron model with sign-constrained synaptic weights are examined in [28], using both theoretical and numerical methods. In contrast to these two works, our study focuses on nonlinearities in the synaptic transmission function (acting on individual inputs) and specifically investigates the impact of parallel synapses (which receive identical inputs from the same presynaptic axon). In [52], Kolmogorov-Arnold Networks replace linear synaptic weights with learnable activation functions, resulting in improved model performance and interpretability. A key difference compared to our work is our focus on a more biologically plausible model, specifically our use of monotonic (sigmoidal) synaptic transmission functions. A promising direction for further research would be to investigate how the type of nonlinearity in parallel synapses can influence the neuronal classification capacity.

This study has focused exclusively on offline learning, in which the algorithm has access to the whole dataset at any point during the learning process. However, a more biologically relevant situation would restrict the learning to occur one data point at a time. Adapting our algorithms to such an online learning scenario presents an interesting future direction for our research.

Methods

Binary classification task

In order to numerically quantify the memory capacity of our model, we utilize binary classification tasks of randomly generated input activity patterns. The binary labels associated with each pattern are also randomized. To generate an N-dimensional pattern $\mathbf{x}_\mu$, we sample the input of the i-th axon ($i = 1, 2, \ldots, N$) from a uniform distribution between 0 and 1, represented by $\mathcal{U}(0, 1)$, independently for each input dimension. The label y_μ associated with the pattern is also independently drawn from a Bernoulli distribution with equal probabilities for +1/ − 1.

Estimating the capacity from numerical simulations

To evaluate the memory capacity of our model, we adopt a methodology similar to that used to measure the memory capacity of the Perceptron. We train a model neuron to classify random patterns. When the number of random patterns, denoted as P, is relatively small, the model neuron is likely to achieve successful classification. However, as the number of patterns increases, at some point the model neuron can no longer classify all the patterns correctly. We define a critical number of patterns, denoted as P^*, at which the model neuron has a 50% chance of successfully classifying all the patterns at the end of training.

The capacity (P^*/N) of the restricted model neuron depends on the number of input axons (N) and the number of parallel synapses per axons (M). We test various P values to assess the

critical number of pattern (P^*) for a restricted model neuron. For each problem size P, we repeat the numerical experiments at least 5 times to estimate the successful classification rate for the model neuron (Fig E in S1 Text). The capacity (P^*/N) is defined as the critical problem size P^* normalized by the number of input axons. To find P^*/N, we interpolate the success rate between different P/N values using a sigmoid function. Fitting a single sigmoid function to the success rates obtained for a model neuron provides one point estimate of its capacity P^*/N. To determine the confidence interval of the capacity P^*/N, we repeatedly sample subsets of the numerical simulations and fit new sigmoid functions to the results from these subsets. The confidence interval estimation is as follows: 1. For a model with a given value of N, half of the simulation results are randomly sampled without replacement for each problem size P; 2. For each resampled dataset, a sigmoid function (ranging from 0 to 1) is fitted to the success rates as a function of P/N; 3. The critical capacity P^*/N is estimated by finding the P/N value corresponding to a success rate of 0.5; 4. This resampling and fitting process is repeated 100 times, providing an estimated capacity and its confidence interval. The whole procedure is performed separately for every model (restricted model neurons with distinct M values, unrestricted models and baseline Perceptron models) for different numbers of input axons N. The error bars in the capacity plots (Figs 2b and 3a) represent the standard deviation of the estimated capacity values obtained from 100 resamplings.

For the unrestricted model neuron, the estimation method for the capacity is similar to that of the restricted model neuron. The only difference is that the capacity depends solely on the number of input axons (N). Therefore, we only need to determine the capacity for various N values. As above, the resampling technique is employed to estimate the confidence interval of the capacity, for each unrestricted model neuron with distinct N value.

Restricted neuron trained with gradient descent

Model. In this version of the model neuron, the transmission function of each individual synapse is characterized by a sigmoid function (Eq 1). The j-th ($j = 1 \ldots M$) parallel synapse on the i-th ($i = 1 \ldots N$) axon has three parameters. The first parameter is $a_{i,j}$. We take its squared value, $(a_{i,j})^2$, as the amplitude of the transmission function, i.e., the maximum value of the response. The second parameter, $t_{i,j}$, is the threshold of the transmission function, which represents the value of the presynaptic input at which the postsynaptic response increases most rapidly. The third parameter, $s_{i,j}$, is the slope of the transmission function, which characterizes the sensitivity to input changes near the threshold. Since, in general, the synaptic response for excitatory synapses increases with larger input, we constrain the synaptic transmission function to increase monotonically (positive slope). That is, $s_{i,j}$ is constrained to be nonnegative. The input for the μ-th pattern on the i-th axon is denoted by $x_{i,\mu}$.

Unlike models with nontrivial dendritic computation or hidden layer neural networks, the synaptic nonlinearities in our model only act on individual input channels, and thus the model architecture is essentially that of a simple Perceptron. The only nonlinearity that combines multiple inputs and allows them to interact is the step-like activation function at the soma that implements the binary classification (or Softplus somatic nonlinearity for the MNIST simulations).

For each input $\mathbf{x}_\mu$, the total current into the soma is $z_\mu = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{i,j}(x_{i,\mu})$. The prediction for μ-th data point is $\hat{y}_\mu = \text{sign}\left(z_\mu - \theta\right)$, where θ is a threshold parameter regulating the excitability of the neuron. This parameter can also be trained with gradient descent.

During simulations, we are using the hyperbolic tangent instead of the (positive) sigmoid function to model the individual synaptic transmission functions $h_{i,j}$. Since $\tanh(x)$ is centered around 0, this makes it slightly easier to learn the final threshold θ, but otherwise

this approach is mathematically equivalent, since the two functions are related as $\tanh(x) = 2\sigma(2x) - 1$. Denote the parameters of the hyperbolic tangent function as $a'_{i,j}$, $s'_{i,j}$, $t'_{i,j}$ and θ'. The hyperbolic tangent function that we use in simulations is then

$$h'_{i,j}(x) = \frac{2(a'_{i,j})^2}{1 + \exp\Big(-2s'_{i,j}(x - t'_{i,j})\Big)} - (a'_{i,j})^2. \tag{3}$$

Comparing Eqs (1) and (3), the learned parameters using hyperbolic tangent and sigmoid function can be linked using the following equations: $(a_{i,j})^2 = 2(a'_{i,j})^2$, $s_{i,j} = 2s'_{i,j}$, $t_{i,j} = t'_{i,j}$ and $\theta = \theta' - \sum_{i,j}(a'_{i,j})^2$. In Fig 2e–g, we show the parameters of the sigmoid function used in the main text, which are essentially scaled versions of parameters in the hyperbolic tangent function.

Training. For the binary classification task, we use the hinge loss as the cost function (Eq 4). The parameter ϵ is the margin of the hinge loss, which we set to 0.1 during training:

$$L = \sum_{\mu=1}^{P} \max\Big(0, \epsilon - (z_\mu - \theta)y_\mu\Big). \tag{4}$$

Based on gradient descent, the update rules of the parameters are as follows (Eq 5):

$$\begin{aligned}
a_{i,j} &\longleftarrow a_{i,j} + \eta_a \sum_{\mu\in\Omega} \frac{2y_\mu a_{i,j}}{1 + \exp\Big(-s_{i,j}(x_{i,\mu} - t_{i,j})\Big)}, \\
s_{i,j} &\longleftarrow s_{i,j} + \eta_s \sum_{\mu\in\Omega} \Big(1 - \frac{1}{1 + \exp\big(-s_{i,j}(x_{i,\mu} - t_{i,j})\big)}\Big) y_\mu h_{i,j}(x_{i,\mu})\big(x_{i,\mu} - t_{i,j}\big), \\
t_{i,j} &\longleftarrow t_{i,j} - \eta_t \sum_{\mu\in\Omega} \Big(1 - \frac{1}{1 + \exp\big(-s_{i,j}(x_{i,\mu} - t_{i,j})\big)}\Big) y_\mu h_{i,j}(x_{i,\mu}) s_{i,j}, \\
\theta &\longleftarrow \theta - \eta_\theta \sum_{\mu\in\Omega} y_\mu,
\end{aligned} \tag{5}$$

where the set Ω includes all data points that are misclassified before the update, or correctly classified by a margin less than ϵ, i.e., $\Omega : \{\mu | \epsilon - (z_\mu - \theta)y_\mu > 0\}$.

During training, we observe that the amplitude of certain synapses will become very small, which leads to a vanishing gradient of all parameters of these synapses, according to Eq (5). Those synapses with extremely small amplitude will become ineffective, and their parameters will no longer be updated. To utilize those ineffective synapses again during training, we increase their amplitude $(a_{i,j})^2$ to a minimum value of 0.01. Their threshold parameters are also randomly shuffled to different values. The motivation is that the thresholds of those ineffective synapses may not be helpful for learning (causing their amplitude to become small), so randomly changing their thresholds to different values might increase the chance of better classification. The values of the shuffled thresholds are drawn from a distribution with support on the input range. This distribution has higher density near the edges of the input range, which mimics the learned threshold distribution (Fig 2f).

Training of the unrestricted neuron model

To parameterize the arbitrary monotonic transmission function from each axon of the unrestricted model, we assume that it takes the form of a staircase composed of step-like functions of the input value (Fig 5). The model neuron is parameterized by N aggregate transmission functions in total, each for one axon. There are P data points in the dataset, thus there will be at most P different values the arbitrary monotonic function can take on this dataset for each

input dimension. In other words, we assume that there are up to P (step-like) parallel synapses on each axon, which limits the biological plausibility of this model.

We constrain each aggregate synaptic function to be monotonic, i.e, for excitatory synapses the function values have to be increasing with increasing input, for each input dimension. We label the synaptic function value of the α-th input value on the i-th axon as $I_{i,\alpha}$, where the α index labels the data points in order of increasing input values (for that axon).

To correctly classify all data points, our training algorithm has to construct a suitable monotonic function, $I_{i,\alpha}$, for each axon. For the μ-th data point, the input is $x_{i,\mu}$ on the i-th axon. If its label y_μ is +1, the algorithm tends to increase the value at $x_{i,\mu}$, whereas if y_μ is −1, the algorithm pushes $I_{i,\mu}$ towards the negative direction. Note that there are two types of ordering of the data points involved. The first one is the natural ordering of the data points, which is indexed by μ and shared by all axons. The other one is the order of increasing inputs $x_{i,\alpha}$, which is indexed by α. Since the increasing order of $x_{i,\alpha}$ differs for each dimension i, this ordering described by α depends on i. The two types of ordering can be linked by a permutation matrix for each axon, which is completely determined by the input data. We will use $x_{i,\alpha}$ (with input values increasing with α) in the following for simplicity.

We can use $-\sum_\alpha y_\alpha I_{i,\alpha}$ as part of our cost function for the i-th axon, which will push the aggregate synaptic function in the correct direction corresponding to the label y_α of each data point with a given input value $x_{i,\alpha}$. Note that the computation of $I_{i,\alpha}$ can be performed in parallel for different input dimensions i. Therefore, we drop the index of i in the remainder of this section for simplicity. Since there could be data points that are more difficult to classify than other data points, we introduce an importance weight w_α for each data point in the loss function by writing $-\sum_\alpha y_\alpha w_\alpha I_\alpha$. This leads to an efficient, iterative way to calculate w_α, such that more difficult data points can have higher importance weights in the next iteration of learning.

The complete cost function we employ for each input dimension is $C = -\sum_\alpha y_\alpha w_\alpha I_\alpha + \lambda \sum_\alpha (I_\alpha)^2$. Here $\sum_\alpha (I_\alpha)^2$ is an L2 regularization term with coefficient λ that discourages

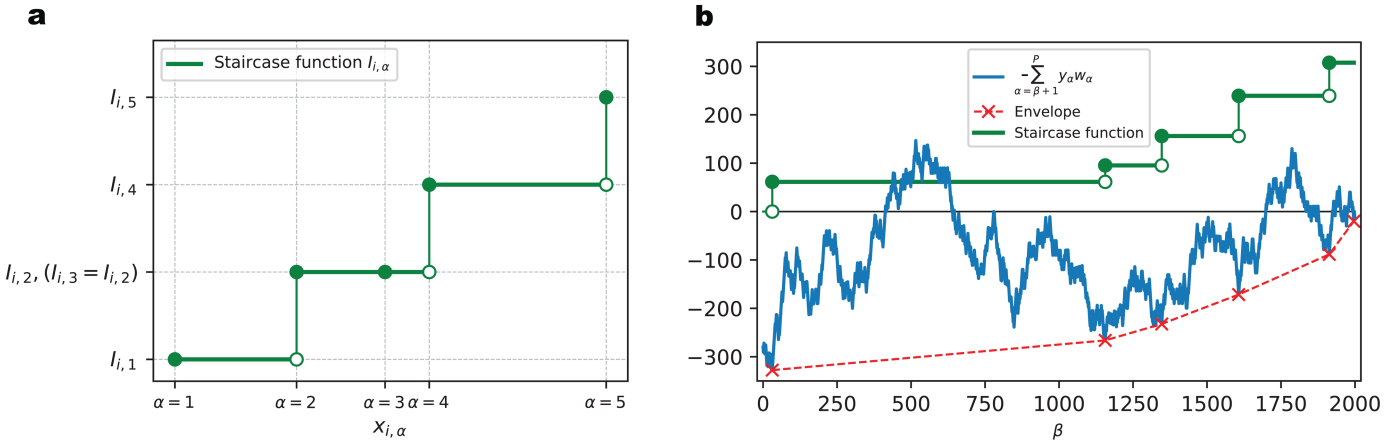

Fig 5. Illustration of the unrestricted model and its training algorithm. (a). For the i-th axon, its aggregate transmission function value at input $x_{i,\alpha}$ is parameterized by the height of the staircase function $I_{i,\alpha}$. The example shown here demonstrates 5 different input values and their corresponding aggregate synaptic function values on the i-th axon. The staircase function either increases (at $\alpha = 1, 2, 4$ and 5) or stays flat (at $\alpha = 3$) at each input value, in order to maintain monotonicity (nondecreasing). **(b)**. Envelope method to find the solution to Eqs (7) and (8), for each input dimension. Staircase functions (green solid line) are either flat or increase at different input values. Solid green dots indicate the increased function value. A (cumulative) random walk (blue solid line) is constructed from the current importance weights of the data points and their labels. A piece-wise linear envelope (red dashed line) is constructed such that it is always below or touching the (cumulative) random walk. The slopes of the envelope segments are increasing from left to right.

https://doi.org/10.1371/journal.pcbi.1012285.g005

postsynaptic currents from becoming too large. To satisfy the monotonicity constraint of the aggregate transmission function, it has to obey $I_1 \leq I_2 \leq \cdots \leq I_P$. To achieve this, we introduce the step size parameters ρ_β, with $\beta = 1 \ldots P - 1$, such that $I_\alpha = \sum_{\beta=1}^{\alpha-1} \rho_\beta^2$, where without loss of generality we have imposed $I_1 = 0$. Here ρ_β^2 represents the step size of the staircase function at the $\beta + 1$-th input value. Then we formulate an optimization problem as follows

$$\min_{I_\alpha, \alpha=1,\ldots,P} -\sum_\alpha y_\alpha w_\alpha I_\alpha + \lambda \sum_\alpha (I_\alpha)^2. \tag{6}$$

Taking the first and second derivative of $C = -\sum_\alpha y_\alpha w_\alpha I_\alpha + \lambda \sum_\alpha (I_\alpha)^2$ with respect to ρ_β, we find

$$\frac{\partial C}{\partial \rho_\beta} = -2\rho_\beta \sum_{\alpha=\beta+1}^{P} (y_\alpha w_\alpha - 2\lambda I_\alpha), \tag{7}$$

$$\frac{\partial^2 C}{\partial \rho_\beta^2} = 8\lambda \rho_\beta^2 (P - \beta) - 2 \sum_{\alpha=\beta+1}^{P} (y_\alpha w_\alpha - 2\lambda I_\alpha). \tag{8}$$

Setting the first derivative to zero, ρ_β and I_α have to satisfy either $\rho_\beta = 0$ or $\sum_{\alpha=\beta+1}^{P} (y_\alpha w_\alpha - 2\lambda I_\alpha) = 0$ for each β. The case $\rho_\beta = 0$ means that the step function is flat at the $\beta + 1$-th input value (no step). The alternative case $\sum_{\alpha=\beta+1}^{P} y_\alpha w_\alpha = 2\lambda \sum_{\alpha=\beta+1}^{P} I_\alpha$ means that the function increases at the $\beta + 1$-th input value. However, to keep $\partial^2 C / \partial \rho_\beta^2 > 0$, the two conditions cannot be satisfied at the same time, so our algorithm has to decide which one should hold for each β.

Solutions to the optimization problem specified by Eqs (7) and (8) can be found by using the envelope method illustrated in Fig 5. The problem is equivalent to determining values of all $I_{\beta+1}$ where the step sizes are nonzero (such that $I_{\beta+1} > I_\beta$). The envelope method iteratively determines $I_{\beta+1}$ with nonzero staircase step size from right to left ($\beta = P$ to $\beta = 1$). Suppose $\beta + 1$ is the rightmost location (β is closest to P) where the staircase step size is nonzero, i.e., $I_\alpha = I_{\beta+1}$ for all $\alpha \geq \beta + 1$, then from Eq (7), we have

$$\sum_{\alpha=\beta+1}^{P} (y_\alpha w_\alpha - 2\lambda I_{\beta+1}) = 0, \tag{9}$$

and thus

$$I_{\beta+1} = \frac{1}{2\lambda(P - \beta)} \sum_{\alpha=\beta+1}^{P} y_\alpha w_\alpha. \tag{10}$$

Based on Eq (10), the value of $I_{\beta+1}$ can be determined using all the y_α and w_α to its right (for $\alpha \geq \beta + 1$). If we consider $y_\alpha w_\alpha$ as the step of a random walk at each α (depending on the random labels y_α and the importance weights), then $\sum_{\alpha=\beta+1}^{P} y_\alpha w_\alpha$ is the cumulative distance traversed between $\beta + 1$ and P.

For the remaining locations with nonzero staircase step sizes at smaller β (corresponding to data points with smaller input values), we can use Eqs (7) and (8) in a similar fashion. Suppose $\beta_1 + 1$ and $\beta_2 + 1$ are two adjacent locations where the staircase step sizes are nonvanishing (with $\beta_2 > \beta_1$), then we have

$$\sum_{\alpha=\beta_1+1}^{P} (y_\alpha w_\alpha - 2\lambda I_\alpha) = 0, \tag{11}$$

$$\sum_{\alpha=\beta_2+1}^{P} (y_\alpha w_\alpha - 2\lambda I_\alpha) = 0. \tag{12}$$

Subtracting Eq (12) from Eq (11), we obtain

$$\sum_{\alpha=\beta_1+1}^{\beta_2} y_\alpha w_\alpha = 2\lambda \sum_{\alpha=\beta_1+1}^{\beta_2} I_\alpha. \tag{13}$$

Since β_1 and β_2 are two adjacent steps, all I_α between them (with $\alpha = \beta_1 + 1, \ldots, \beta_2$) have the same value, i.e., $I_\alpha = I_{\beta_1+1}\ \forall \alpha \in \{\beta_1 + 1, \ldots, \beta_2\}$. This leads to

$$I_{\beta_1+1} = \frac{1}{2\lambda(\beta_2 - \beta_1)} \sum_{\alpha=\beta_1+1}^{\beta_2} y_\alpha w_\alpha. \tag{14}$$

As above, if considering $y_\alpha w_\alpha$ as the step of a one-dimensional random walk at each α, the quantity $\sum_{\alpha=\beta_1+1}^{\beta_2} y_\alpha w_\alpha$ is the cumulative distance traveled between β_1 and β_2. The intuition that follows from Eqs (10) and (14) leads us to construct an envelope of the random walk, as in Fig 5, such that the slopes of each segment are positive. The steps sizes of our staircase function are only nonzero at locations where the envelope touches the random walk. The value of the staircase function is proportional to the slope of the envelope. Since the synaptic functions are monotonic (nondecreasing in this case), the slopes of the envelope have to be increasing as α grows.

Having solved the above optimization problem separately for each axon, we can now tackle the overall classification problem using the following iterative process: 1. Given the ordered input data $x_{i,\alpha}$, labels y_α and the current importance weights w_α (whose values are all initialized to unity), construct the envelope and calculate the aggregate synaptic transmission function I_α; 2. Given the new I_α, evaluate the model's prediction for all the data points; 3. Re-weight the data points by updating w_α if the data point α is misclassified, incrementing $w_\alpha \leftarrow w_\alpha + 1$. We repeat steps 1 to 3, until all data points are correctly classified or a maximum iteration number is reached.

Training neural network models with parallel synapses on the MNIST classification task

For the MNIST classification task, we employ a fully connected neural network with one hidden layer. This network processes images of handwritten digits from the MNIST dataset, each being 28 × 28 pixels. The network's input layer consists of 784 neurons, corresponding to the pixel count, and the output layer has 10 neurons, representing a one-hot encoding of digit identities ranging from 0 to 9. Linear (single) synapses form the feedforward connections from the input layer to the hidden layer. The hidden layer's nonlinearity is a Softplus function, which is a smooth approximation to the Rectified Linear (ReLU) function. The feedforward connections from the hidden layer to the output layer are made through either nonlinear parallel synapses or single linear synapses (for comparison). In both types of networks, we apply batch normalization of inputs to the hidden layer. We employ a multi-label cross-entropy loss for training. The hyperparameter settings are described in Table C in S1 Text.

In networks with parallel synapses, these synapses are used exclusively from the hidden layer to the output layer. Each set of parallel synapses, connecting a pair of neurons, is constrained to have a monotonically increasing aggregate transmission function. For the purpose of fair comparison, the linear synapses connecting hidden layer and output layer in networks without parallel synapses are also constrained to be monotonically increasing, i.e., have positive weights.

The MNIST dataset [40] comprises 60,000 images in the training set and 10,000 images in the testing set. During each training epoch, we train the networks on all images in the training set and subsequently test their classification accuracy on the testing set.

Acknowledgments

The numerical simulations were performed on the Nautilus platform, which is supported in part by National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC-2112167, CNS-2100237, CNS-2120019, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100Gbps networks. M.K.B was supported by R01NS125298 (NINDS) and the Kavli Institute for Brain and Mind.

Supporting information

S1 Text. Supplementary materials including Figures and Tables.
(PDF)

Author contributions

Conceptualization: Marcus K. Benna.

Formal analysis: Yuru Song, Marcus K. Benna.

Funding acquisition: Marcus K. Benna.

Investigation: Yuru Song, Marcus K. Benna.

Methodology: Yuru Song, Marcus K. Benna.

Software: Yuru Song.

Supervision: Marcus K. Benna.

Visualization: Yuru Song.

Writing – original draft: Yuru Song, Marcus K. Benna.

Writing – review & editing: Yuru Song, Marcus K. Benna.

References

1. Deuchars J, West DC, Thomson AM. Relationships between morphology and physiology of pyramid-pyramid single axon connections in rat neocortex in vitro. J Physiol. 1994;478(3):423–35. https://doi.org/10.1113/jphysiol.1994.sp020262 PMID: 7965856
2. Feldmeyer D, Lübke J, Sakmann B. Efficacy and connectivity of intracolumnar pairs of layer 2/3 pyramidal cells in the barrel cortex of juvenile rats. J Physiol. 2006;575(Pt 2):583–602. https://doi.org/10.1113/jphysiol.2006.105106 PMID: 16793907
3. Holler S, Köstinger G, Martin KA, Schuhknecht GF, Stratford KJ. Structure and function of a neocortical synapse. Nature. 2021;591(7848):111–6. https://doi.org/10.1038/s41586-020-03134-2 PMID: 33442056
4. Lee W-CA, Bonin V, Reed M, Graham BJ, Hood G, Glattfelder K, et al. Anatomy and function of an excitatory network in the visual cortex. Nature. 2016;532(7599):370–4. https://doi.org/10.1038/nature17192 PMID: 27018655
5. Markram H, Lübke J, Frotscher M, Roth A, Sakmann B. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. J Physiol. 1997;500(2):409–40. https://doi.org/10.1113/jphysiol.1997.sp022031 PMID: 9147328

6. Shepherd GM, Stepanyants A, Bureau I, Chklovskii D, Svoboda K. Geometric and functional organization of cortical circuits. Nat Neurosci. 2005;8(6):782–90. https://doi.org/10.1038/nn1447 PMID: 15880111
7. Silver RA, Lubke J, Sakmann B, Feldmeyer D. High-probability uniquantal transmission at excitatory synapses in barrel cortex. Science. 2003;302(5652):1981–4. https://doi.org/10.1126/science.1087160 PMID: 14671309
8. Chicurel ME, Harris KM. Three-dimensional analysis of the structure and composition of CA3 branched dendritic spines and their synaptic relationships with mossy fiber boutons in the rat hippocampus. Journal of Comp Neurol. 1992;325(2):169–82.
9. Schmidt H, Gour A, Straehle J, Boergens KM, Brecht M, Helmstaedter M. Axonal synapse sorting in medial entorhinal cortex. Nature. 2017;549(7673):469–75. https://doi.org/10.1038/nature24005 PMID: 28959971
10. Markram H, Muller E, Ramaswamy S, Reimann MW, Abdellah M, Sanchez CA, et al. Reconstruction and simulation of neocortical microcircuitry. Cell. 2015;163(2):456–92. https://doi.org/10.1016/j.cell.2015.09.029 PMID: 26451489
11. Zucker RS, Regehr WG. Short-term synaptic plasticity. Annu Rev Physiol. 2002;64:355–405. https://doi.org/10.1146/annurev.physiol.64.092501.114547 PMID: 11826273
12. Turrigiano GG, Leslie KR, Desai NS, Rutherford LC, Nelson SB. Activity-dependent scaling of quantal amplitude in neocortical neurons. Nature. 1998;391(6670):892–6. https://doi.org/10.1038/36103 PMID: 9495341
13. Turrigiano GG, Nelson SB. Homeostatic plasticity in the developing nervous system. Nat Rev Neurosci. 2004;5(2):97–107. https://doi.org/10.1038/nrn1327 PMID: 14735113
14. Burrone J, Murthy VN. Synaptic gain control and homeostasis. Curr Opin Neurobiol. 2003;13(5):560–7. https://doi.org/10.1016/j.conb.2003.09.007 PMID: 14630218
15. Brown RE, Milner PM. The legacy of Donald O. Hebb: more than the Hebb Synapse. Nat Rev Neurosci. 2003;4(12):1013–19. https://doi.org/10.1038/nrn1257
16. Lynch MA. Long-term potentiation and memory. Physiol Rev. 2004;84(1):87–136. https://doi.org/10.1152/physrev.00014.2003 PMID: 14715912
17. Morris RGM. Long-term potentiation and memory. Philos Trans R Soc Lond B Biol Sci. 2003;358(1432):643–7. https://doi.org/10.1098/rstb.2002.1230 PMID: 12740109
18. Reyes A, Lujan R, Rozov A, Burnashev N, Somogyi P, Sakmann B. Target-cell-specific facilitation and depression in neocortical circuits. Nat Neurosci. 1998;1(4):279–85. https://doi.org/10.1038/1092 PMID: 10195160
19. Markram H, Wang Y, Tsodyks M. Differential signaling via the same axon of neocortical pyramidal neurons. Proc Natl Acad Sci USA. 1998;95(9):5323–28. https://doi.org/10.1073/pnas.95.9.5323 PMID: 9560274
20. Trommersh̓auser J, Schneggenburger R, Zippelius A, Neher E. Heterogeneous presynaptic release probabilities: functional relevance for short-term plasticity. Biophys J. 2003;84(3):1563– 79. https://doi.org/10.1016/s0006-3495(03)74967-4
21. Fares T, Stepanyants A. Cooperative synapse formation in the neocortex. Proc Natl Acad Sci USA. 2009;106(38):16463–68. https://doi.org/10.1073/pnas.0813265106 PMID: 19805321
22. Fauth M, Wörgötter F, Tetzlaff C. The formation of multi-synaptic connections by the interaction of synaptic and structural plasticity and their functional consequences. PLOS Comput Biol. 2015;11(1):e1004031. https://doi.org/10.1371/journal.pcbi.1004031 PMID: 25590330
23. Hiratani N, Fukai T. Redundancy in synaptic connections enables neurons to learn optimally. Proc Natl Acad Sci USA. 2018;115(29):E6871–9. https://doi.org/10.1073/pnas.1803274115 PMID: 29967182
24. Beniaguev D, Shapira S, Segev I, London M. Multiple synaptic contacts combined with dendritic filtering enhance spatio-temporal pattern recognition capabilities of single neurons. bioRxiv. 2022. p. 2022–01.
25. Zhang J, Hu J, Liu J. Neural network with multiple connection weights. Pattern Recognit. 2020;107:107481. https://doi.org/10.1016/j.patcog.2020.107481
26. Jones IS, Kording KP. Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree? Neural Comput. 2021;33(6):1554–71.
27. London M, Häusser M. Dendritic computation. Annu Rev Neurosci. 2005;28:503–32. https://doi.org/10.1146/annurev.neuro.28.061604.135703 PMID: 16033324
28. Lauditi C, Malatesta EM, Pittorino F, Baldassi C, Brunel N, Zecchina R. Impact of dendritic non-linearities on the computational capabilities of neurons. arXiv Preprint 2024. arXiv:240707572.
29. Polsky A, Mel BW, Schiller J. Computational subunits in thin dendrites of pyramidal cells. Nat Neurosci. 2004;7(6):621–7. https://doi.org/10.1038/nn1253 PMID: 15156147

30. Poirazi P, Brannon T, Mel BW. Pyramidal neuron as two-layer neural network. Neuron. 2003;37(6):989–99. https://doi.org/10.1016/s0896-6273(03)00149-1 PMID: 12670427
31. Poirazi P, Papoutsi A. Illuminating dendritic function with computational models. Nat Rev Neurosci. 2020;21(6):303–21. https://doi.org/10.1038/s41583-020-0301-7 PMID: 32393820
32. Zavatone-Veth JA, Pehlevan C. Activation function dependence of the storage capacity of treelike neural networks. Phys Rev E. 2021;103(2):L020301. https://doi.org/10.1103/PhysRevE.103.L020301 PMID: 33736039
33. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev. 1958;65(6):386–408. https://doi.org/10.1037/h0042519 PMID: 13602029
34. Gardner E, Derrida B. Optimal storage properties of neural network models. J Phys A Math Gen. 1988;21(1):271.
35. Gardner E. The space of interactions in neural network models. J Phys A Math Gen. 1988;21(1):257.
36. Amit DJ, Wong KYM, Campbell C. Perceptron learning with sign-constrained weights. J Phys A Math Gen. 1989;22(12):2039. https://doi.org/10.1088/0305-4470/22/12/009
37. Amit DJ, Campbell C, Wong KYM. The interaction space of neural networks with sign-constrained synapses. J Phys A Math Gen. 1989;22(21):4687. https://doi.org/10.1088/0305-4470/22/21/030
38. Nadal JP. On the storage capacity with sign-constrained synaptic couplings. Netw Comput Neural Syst. 1990;1(4):463–6.
39. Legenstein R, Maass W. On the classification capability of sign-constrained perceptrons. Neural Comput. 2008;20(1):288–309. https://doi.org/10.1162/neco.2008.20.1.288 PMID: 18045010
40. Deng L. The MNIST database of handwritten digit images for machine learning research. IEEE Signal Process Mag. 2012;29(6):141–2. https://doi.org/10.1109/msp.2012.2211477
41. Fusi S, Abbott LF. Limits on the memory storage capacity of bounded synapses. Nat Neurosci. 2007;10(4):485–93. https://doi.org/10.1038/nn1859 PMID: 17351638
42. Eccles JC, Fatt P, Koketsu K. Cholinergic and inhibitory synapses in a pathway from motor-axon collaterals to motoneurones. J Physiol. 1954;126(3):524–62. https://doi.org/10.1113/jphysiol.1954.sp005226 PMID: 13222354
43. Spitzer NC. Neurotransmitter switching in the developing and adult brain. Annu Rev Neurosci. 2017;40:1–19. https://doi.org/10.1146/annurev-neuro-072116-031204 PMID: 28301776
44. Cover TM. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans Electron Comput. 1965;EC-14(3):326–34. https://doi.org/10.1109/pgec.1965.264137
45. Brunel N, Hakim V, Isope P, Nadal J-P, Barbour B. Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. Neuron. 2004;43(5):745–57. https://doi.org/10.1016/j.neuron.2004.08.023 PMID: 15339654
46. Clopath C, Nadal J-P, Brunel N. Storage of correlated patterns in standard and bistable Purkinje cell models. PLoS Comput Biol. 2012;8(4):e1002448. https://doi.org/10.1371/journal.pcbi.1002448 PMID: 22570592
47. Brunel N. Is cortical connectivity optimized for storing information? Nat Neurosci. 2016;19(5):749–55. https://doi.org/10.1038/nn.4286 PMID: 27065365
48. Bullmore E, Sporns O. The economy of brain network organization. Nat Rev Neurosci. 2012;13(5):336–49. https://doi.org/10.1038/nrn3214 PMID: 22498897
49. Chen BL, Hall DH, Chklovskii DB. Wiring optimization can relate neuronal structure and function. Proc Natl Acad Sci USA. 2006;103(12):4723–8. https://doi.org/10.1073/pnas.0506806103 PMID: 16537428
50. Rubinov M, Ypma RJF, Watson C, Bullmore ET. Wiring cost and topological participation of the mouse brain connectome. Proc Natl Acad Sci USA. 2015;112(32):10032–7. https://doi.org/10.1073/pnas.1420315112 PMID: 26216962
51. Harris JJ, Jolivet R, Attwell D. Synaptic energy use and supply. Neuron. 2012;75(5):762–77. https://doi.org/10.1016/j.neuron.2012.08.019 PMID: 22958818
52. Liu Z, Wang Y, Vaidya S, Ruehle F, Halverson J, Soljačić M, et al. Kan: Kolmogorov-arnold networks. arXiv Preprint. arXiv:240419756. 2024. https://arxiv.org/abs/2404.19756