SIAM J. MATH. DATA SCI.
Vol. 2, No. 4, pp. 1004–1033

Memory Capacity of Neural Networks with Threshold and Rectified Linear Unit Activations*

Roman Vershynin†

Abstract. Overwhelming theoretical and empirical evidence shows that mildly overparametrized neural networks—those with more connections than the size of the training data—are often able to memorize the training data with 100% accuracy. This was rigorously proved for networks with sigmoid activation functions [M. Yamasaki, *Proceedings of the International Conference on Artificial Neural Networks*, 1993, pp. 546–549; G.-B. Huang, *IEEE Trans. Neural Netw.*, 14 (2003), pp. 274–281] and, very recently, for rectified linear unit (ReLU) activations [C. Yun, S. Sra, and A. Jadbabaie, *Proceedings of the Conference on Neural Information Processing Systems*, 2019, pp. 15532–15543]. Addressing a open question of Baum [*J. Complexity*, 4 (1988), pp. 193–215], we prove that this phenomenon holds for general multilayered perceptrons, i.e., neural networks with threshold activation functions, or with any mix of threshold and ReLU activations. Our construction is probabilistic and exploits sparsity.

Key words. neural networks, memory, capacity, multilayer perceptron

AMS subject classifications. 68Q32, 92B20

DOI. 10.1137/20M1314884

1. Introduction.

This paper continues the long study of the memory capacity of neural architectures. How much information can a human brain learn? What are fundamental memory limitations, and how should the "optimal brain" be organized to achieve maximal capacity? These questions are complicated by the fact that we do not sufficiently understand the architecture of the human brain. But suppose that a neural architecture is known to us. Consider, for example, a given artificial neural network. Is there a general formula that expresses the memory capacity in terms of the network's architecture?

1.1. Neural architectures.

In this paper we study a general layered, feedforward, fully connected neural architecture with arbitrarily many layers, arbitrarily many nodes in each layer, with either threshold or rectified linear unit activation functions between all layers, and with the threshold activation function at the output node.

Readers unfamiliar with this terminology may think of a neural architecture as a computational device that can compute certain compositions of linear and nonlinear maps. Let us describe precisely the functions computable by a neural architecture. Some of the best

*Received by the editors January 24, 2020; accepted for publication (in revised form) July 22, 2020; published electronically October 20, 2020.
https://doi.org/10.1137/20M1314884
Funding: This research was supported in part by U.S. Air Force grant FA9550-18-1-0031.
†Department of Mathematics, University of California at Irvine, Irvine, CA 92617 USA (rlvershynin@gmail.com).

studied and most popular nonlinear functions $\phi : \mathbb{R} \to \mathbb{R}$, or "activation functions," include the *threshold function* and the *rectified linear unit* (ReLU), defined by

$$\phi(t) = \mathbf{1}_{\{t>0\}} \quad \text{and} \quad \phi(t) = \max(0, t) = t_+, \tag{1.1}$$

respectively.[1] We call a map *pseudolinear* if it is a composition of an affine transformation and a nonlinear transformation ϕ applied coordinatewise. Thus, $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ is pseudolinear map if it can be expressed as

$$\Phi(x) = \phi(Vx - b), \quad x \in \mathbb{R}^n,$$

where V is a $m \times n$ matrix of "weights," $b \in \mathbb{R}^m$ is a vector of "biases," and ϕ is either the threshold or ReLU function (1.1), which we apply to each coordinate of the vector $Wx - b$.

A *neural architecture* computes compositions of pseudolinear maps, i.e., functions $F : \mathbb{R}^{n_1} \to \mathbb{R}$ of the type

$$F = \Phi_L \circ \cdots \circ \Phi_2 \circ \Phi_1,$$

where $\Phi_1 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$, $\Phi_2 : \mathbb{R}^{n_2} \to \mathbb{R}^{n_3}$, $\ldots$, $\Phi_{L-1} : \mathbb{R}^{n_{L-1}} \to \mathbb{R}^{n_L}$, $\Phi_L : \mathbb{R}^{n_L} \to \mathbb{R}$ are pseudolinear maps. Each of the maps Φ_i may be defined using either the threshold or ReLU function, and mix and match is allowed. However, for the purpose of this paper, *we require the output function $\Phi_L : \mathbb{R}^{n_L} \to \mathbb{R}$ to have the threshold activation.*[2]

We regard the matrices V and b in the definition of each pseudolinear map Φ_i as free parameters of the given neural architecture. Varying these free parameters, one can make a given neural architecture compute different functions $F : \mathbb{R}^{n_1} \to \{0, 1\}$. Let us denote the class of such functions computable by a given architecture by

$$\mathcal{F}(n_1, \ldots, n_L, 1).$$

As Figure 1 illustrates, a neural architecture can be visualized as a directed graph, which consists of L layers each having n_i nodes (or *neurons*), and one output node. Successive layers are connected by bipartite graphs, each of which represents a pseudolinear map Φ_i.

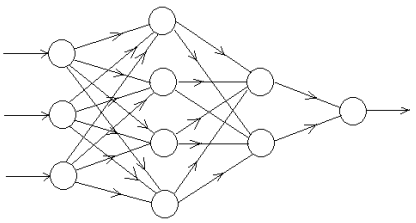

Figure 1. *A neural architecture with an input layer, two hidden layers, and an output node. The class of functions $F : \mathbb{R}^3 \to \mathbb{R}$ this architecture can compute is denoted $\mathcal{F}(3, 4, 2, 1)$.*

[1] It should be possible to extend our results for other activation functions. To keep the argument simple, we shall focus on the threshold and ReLU nonlinearities in this paper.

[2] General neural architectures used by practitioners and considered in the literature may have more than one output node and have other activation functions at the output node.

Each neuron is a little computational device. It sums all inputs from the neurons in the previous layer with certain weights, applies the activation function ϕ to the sum, and passes the output to neurons in the next layer. More specifically, the neuron determines if the sum of incoming signals from the previous layers exceeds a certain firing threshold b. If so, the neuron fires with strength 1 (if ϕ is the threshold activation function) or with strength proportional to the incoming signal (if ϕ is the ReLU activation).

1.2. Memory capacity. When can a given neural architecture remember a given data? Suppose, for example, that we have K digital pictures of cats and dogs, encoded as as vectors $x_1, \dots, x_K \in \mathbb{R}^{n_1}$, and labels $y_1, \dots, y_K \in \{0, 1\}$, where 0 stands for a cat and 1 for a dog. Can we train a given neural architecture to memorize which images are cats and which are dogs? Equivalently, does there exist a function $F \in \mathcal{F}(n_1, \dots, n_L, 1)$ such that

$$F(x_k) = y_k \quad \text{for all} \quad k = 1, \dots, K? \tag{1.2}$$

A common belief is that this should happen for any sufficiently *overparametrized* network—an architecture that that has significantly more free parameters than the size of the training data. The free parameters of our neural architecture are the $n_{i-1} \times n_i$ weight matrices V_i and the bias vectors $b_i \in \mathbb{R}^{n_i}$. The number of biases is negligible compared to the number of weights, and the number of free parameters is approximately the same as the *number of connections*[3]

$$\overline{W} = n_1 n_2 + \cdots + n_{L-1} n_L.$$

Thus, one can wonder whether a general neural architecture is able to memorize the data as long as the number of connections is bigger than the size of the data, i.e., as long as

$$\overline{W} \gtrsim K. \tag{1.3}$$

Motivated by this question, one can define *memory capacity* of a given architecture as the largest size of general data the architecture is able to memorize. In other words, the memory capacity is the largest K such that for a general[4] set of points $x_1, \dots, x_K \in \mathbb{R}^{n_1}$ and for any labels $y_1, \dots, y_K \in \{0, 1\}$ there exists a function $F \in \mathcal{F}(n_1, \dots, n_L)$ that satisfies (1.2).

The memory capacity is clearly bounded above by the vc-dimension,[5] which is $O(\overline{W} \log \overline{W})$ for neural architectures with threshold activation functions [7] and $O(L\overline{W} \log \overline{W})$ for neural architectures with ReLU activation functions [5]. Thus, our question is whether these bounds are tight—is memory capacity (approximately) proportional to $\overline{W}$?

1.3. The history of the problem. A version of this question was raised by Baum [6] in 1988. Building on the earlier work of Cover [8], Baum studied the memory capacity of *multilayer perceptrons*, i.e., feedforward neural architectures with threshold activation functions. He first looked at the network architecture $[n, m, 1]$ with one hidden layer consisting of m

[3] We dropped the number of output connections, which is negligible compared to $\overline{W}$.

[4] One may sometimes wish to exclude some kinds of pathological data, so natural assumptions can be placed on the set of data points x_k. In this paper, for example, we consider unit and separated points x_k.

[5] This is because the vc-dimension is the maximal K for which *there exist* points $x_1, \dots, x_K \in \mathbb{R}^{n_1}$ so that for any labels $y_1, \dots, y_K \in \{0, 1\}$ there exists a function $F \in \mathcal{F}(n_1, \dots, n_L)$ that satisfies (1.2). The memory capacity requires *any* general set of points $x_1, \dots, x_K \in \mathbb{R}^{n_1}$ to succeed as above.

nodes (and, as notation suggests, n nodes in the hidden layer and one output node). Baum noticed that for data points x_k in general position in $\mathbb{R}^n$, the memory capacity of the architecture $[n, m, 1]$ is about nm, i.e., it is proportional to the number of connections. This is not difficult: general position guarantees that the hyperplane spanned by any subset of n data points misses any other data points; this allows one to train each of the m neurons in the hidden layer on its own batch of n data points.

Baum then asked if the same phenomenon persists for deeper neural networks. He asked whether for large K there exists a deep neural architecture with a total of $O(\sqrt{K})$ neurons in the hidden layers and with memory capacity at least K. Such a result would demonstrate the benefit of depth. Indeed, we just saw that shallow architecture $[n, O(\sqrt{K}), 1]$ has capacity just $n\sqrt{K}$, which would be smaller than the hypothetical capacity K of deeper architectures for $n \ll K$.

There was no significant progress on Baum's problem. As Mitchison and Durbin noted in 1989, "one significant difference between a single threshold unit and a multilayer network is that, in the latter case, the capacity can vary between sets of input vectors, even when the vectors are in general position" [17]. Attempting to count different functions F that a deep network can realize on a given data set (x_k), Kowalczyk wrote in 1997, "One of the complications arising here is that in contrast to the single neuron case even for perceptrons with two hidden units, the number of implementable dichotomies may be different for various n-tuples in general position... Extension of this result to the multilayer case is still an open problem" [15].

The memory capacity problem is more tractable for neural architectures in which the threshold activation is replaced by one of its continuous proxies such as ReLU, sigmoid, tanh, or polynomial activation functions. Such activations allow neurons to fire with variable, controllable amplitudes. Heuristically, this ability makes it possible to encode the training data very compactly into the firing amplitudes.

Yamasaki claimed without proof in 1993 that for the sigmoid activation $\phi(t) = 1/(1+e^{-t})$ and for data in general position, the memory capacity of a general deep neural architecture is lower bounded $\overline{W}$, the number of connections [24]. A version of Yamasaki's claim was proved in 2003 by Huang for arbitrary data and neural architectures with two hidden layers [13].

In 2017, Zhang et al. [26] gave a construction of an arbitrarily large (but not fully connected) neural architecture with ReLU activations and whose memory capacity is proportional to both the number of connections and the number of nodes. Hardt and Ma [12] gave a different construction of a residual network with similar properties.

Very recently, Yun, Sra, and Jadbabaie [25] removed the requirement that there be more nodes than data, showing that the memory capacity of networks with ReLU and tanh activation functions is proportional to the number of connections. Ge, Wang, and Zhao [11] proved a similar result for polynomial activations.

Significant efforts were made in the last two years to justify why for overparametrized networks, the gradient descent and its variants could achieve 100% capacity on the training data [10, 9, 16, 27, 1, 14, 28, 18, 21, 2, 19]; see [22] for a survey of related developments.

1.4. Main result. Meanwhile, the original problem of Baum [6]—to determine memory capacity of networks with threshold activations—has remained open. In contrast to the neurons

with continuous activation functions, neurons with threshold activations either do not fire at all or fire with the same unit amplitude. The strength of the incoming signal is lost when transmitted through such neurons, and it is not clear how the data can be encoded.

This is what makes Baum's question hard. In this paper, we (almost) give a positive answer to this question.

Why "almost"? First, the size of the input layer n_1 should not affect the capacity bound and should be excluded from the count of the free parameters $\overline{W}$. To see this, consider, for example, the data points $x_k \in \mathbb{R}^{n_1}$ all lying on one line; with respect to such data, the network is equivalent to one with $n_1 = 1$. Next, ultranarrow bottlenecks should be excluded at least for the threshold nonlinearity: for example, any layer with just $n_i = 1$ node makes the number of connections that occur in the further layers irrelevant as free parameters.

In our actual result, we make somewhat stronger assumptions: in counting connections, we exclude not only the first layer but also the second; we rule out all exponentially narrow bottlenecks (not just of size one); we assume that the data points x_k are unit and separated; finally, we allow logarithmic factors.

Theorem 1.1. *Let $n_1, \dots, n_L$ be positive integers, and set $n_0 := \min(n_2, \dots, n_L)$ and $n_\infty := \max(n_2, \dots, n_L)$. Consider unit vectors $x_1, \dots, x_K \in \mathbb{R}^n$ that satisfy*

$$\|x_i - x_j\|_2 \ge C\sqrt{\frac{\log\log n_\infty}{\log n_0}}. \tag{1.4}$$

Consider any labels $y_1, \dots, y_K \in \{0, 1\}$. Assume that the number of deep connections $W := n_3 n_4 + \cdots + n_{L-1} n_L$ satisfies

$$W \ge CK \log^5 K, \tag{1.5}$$

as well as $K \le \exp(cn_0^{1/5})$ and $n_\infty \le \exp(cn_0^{1/5})$. Then the network can memorize the label assignment $x_k \to y_k$ exactly; i.e., there exists a map $F \in \mathcal{F}(n_1, \dots, n_L, 1)$ such that

$$F(x_k) = y_k \quad \text{for all} \quad k = 1, \dots, K. \tag{1.6}$$

Here C and c denote certain positive absolute constants.

In short, Theorem 1.1 states that the memory capacity of a general neural architecture with threshold or ReLU activations (or a mix thereof) is lower bounded by the number of the deep connections. This bound is independent of the depth, bottlenecks (up to exponentially narrow), or any other architectural details.

1.5. Should the data be separated? One can wonder about the necessity of the separation assumption (1.4). Can we just assume that x_k are distinct? While this is true for ReLU and tanh activations [25], it is false for threshold activations. A moment's thought reveals that any pseudolinear map from $\mathbb{R}$ to $\mathbb{R}^m$ transforms any line into a set of cardinality $O(m)$. Thus, by the pigeonhole principle, any map from layer 1 to layer 2 is noninjective on the set of K data points x_k—which makes it impossible to memorize some label assignments—unless $K = O(n_2)$. In other words, if we just assume that the data points x_k are distinct, the network must have at least as many *nodes* in the second layer as the number of data points. Still, the separation assumption (5.2) does not look tight and might be weakened.

1.6. Related notions of capacity. Instead of requiring the network to memorize the training data with 100% accuracy as in Theorem 1.1, one can ask to memorize just $1-\varepsilon$ fraction, or just half of the training data correctly. This corresponds to a relaxed or *fractional memory capacity* of neural architectures that was introduced by Cover in 1965 [8] and studied extensively afterwards.

To estimate fractional capacity of a given architecture, one needs to count all functions F this architecture can realize on a given finite set of points x_k. When this set is the Boolean cube $\{0,1\}^n$, this amounts to counting all Boolean functions $F : \{0,1\}^n \to \{0,1\}$ the architecture can realize. The binary logarithm of the number of all such Boolean functions was called (expressive) capacity by Baldi and the author [3, 4]. For a neural architecture with all threshold activations and L layers, the expressive capacity is equivalent to the cubic polynomial in the sizes of layers n_i:

$$\sum_{i=1}^{L-1} \min(n_1, \dots, n_i) n_i n_{i+1}$$

up to an absolute constant factor [4]. The factor $\min(n_1, \dots, n_i)$ quantifies the effect of any bottlenecks that occur before layer i.

Similar results can be proved for the *restricted expressive capacity* where we count the functions F the architecture can realize on a given finite set of K points x_k [4, section 10.5]. Ideally, one might hope to find that all 2^K functions can be realized on a general K-element set, which would imply that the memory capacity is at least K. However, the current results on restricted expressive capacity are not tight enough to reach such conclusions.

2. The method. Our construction of the function F in Theorem 1.1 is probabilistic. Let us first illustrate our approach for the architecture $[n, n, n, 1]$ with two hidden layers and with threshold activations throughout. We would like to find a composition of pseudolinear functions

$$F : \mathbb{R}^n \xrightarrow{\Phi_1} \mathbb{R}^n \xrightarrow{\Phi_2} \mathbb{R}^n \xrightarrow{\Psi} \{0,1\}$$

that fits the given data (x_k, y_k) as in (1.6).

The first two maps Φ_1 and Φ_2 are *enrichment maps* whose only purpose is to spread the data x_k in the space, transforming it into an almost orthogonal set. Specifically, Φ_1 will transform the separated points x_k into $o(1)$-orthogonal points u_k (Theorem 5.2), Φ_2 will transform the $o(1)$-orthogonal points u_k into $O(1/\sqrt{n})$-orthogonal points v_k (Theorem 5.4), and, finally, the *perception map* $\Psi(x)$ will fit the data: $\Psi(v_k) = y_k$.

2.1. Enrichment. Our construction of the enrichment maps Φ_1 and Φ_2 exploits *sparsity*. Both maps will have the form

$$\Phi(x) = \phi(Gx - \bar{b}) = \big(\mathbf{1}_{\{\langle g_i, x\rangle > b\}}\big)_{i=1}^n,$$

where G is an $n \times n$ Gaussian random matrix with all i.i.d. $N(0,1)$ coordinates, $g_i \sim N(0, I_n)$ are independent standard normal random vectors, and $\bar{b}$ is the vector all of whose coordinates equal some value $b > 0$.

If b is large, $\Phi(x)$ is a sparse random vector with independently and identically distributed (i.i.d.) Bernoulli coordinates. A key heuristic is that independent sparse random vectors are

almost orthogonal. Indeed, if u and u' are independent random vectors in $\mathbb{R}^n$ whose all coordinates are Bernoulli(p), then $\mathbb{E}\langle u, u'\rangle = np^2$ while $\mathbb{E}\,\|u\|_2 = \mathbb{E}\,\|u'\|_2 = np$, so we should expect

$$\frac{\langle u, u'\rangle}{\|u\|_2 \|u'\|_2} \sim p,$$

making u and u' almost orthogonal for small p.

Unfortunately, the sparse random vectors $\Phi(x)$ and $\Phi(x')$ are not independent unless x and x' are exactly orthogonal. Nevertheless, our heuristic that sparsity induces orthogonality still works in this setting. To see this, let us check that the correlation of the coefficients $\Phi(x)$ and $\Phi(x')$ is small even if x and x' are far from being orthogonal. A standard asymptotic analysis of the tails of the normal distribution implies that

$$\mathbb{E}\,\Phi(x)_i \Phi(x')_i = \mathbb{P}\left\{\langle g, x\rangle > b,\ \langle g, x'\rangle > b\right\} \le 2\exp(-b^2\delta^2/8)\ \mathbb{P}\left\{\langle g, x\rangle > b\right\} \tag{2.1}$$

if x and x' are unit and δ-separated (Proposition 3.1), and

$$\mathbb{E}\,\Phi(x)_i \Phi(x')_i = \mathbb{P}\left\{\langle g, x\rangle > b,\ \langle g, x'\rangle > b\right\} \le 2\exp(2b^2\varepsilon)\ \left(\mathbb{P}\left\{\langle g, x\rangle > b\right\}\right)^2 \tag{2.2}$$

if x and x' are unit and ε-orthogonal (Proposition 3.2).

Now we choose b so that the coordinates of $\Phi(x)$ and $\Phi(x')$ are sparse enough, i.e.,

$$\mathbb{E}\,\Phi(x)_i = \mathbb{P}\left\{\langle g, x\rangle > b\right\} = \frac{1}{\sqrt{n}} =: p;$$

thus $b \sim \sqrt{\log n}$. Choose ε sufficiently small to make the factor $\exp(2b^2\varepsilon)$ in (2.2) nearly constant, i.e.,

$$\varepsilon \sim \frac{1}{b^2} \sim \frac{1}{\log n}.$$

Finally, we choose the separation threshold δ large enough to make the factor $2\exp(-b^2\delta^2/8)$ in (2.1) less than ε, i.e.,

$$\delta \sim \sqrt{\frac{\log(1/\varepsilon)}{\log n}} \sim \sqrt{\frac{\log\log n}{\log n}};$$

this explains the form of separation condition in Theorem 1.1.

With these choices, (2.1) gives

$$\mathbb{E}\,\Phi(x)_i \Phi(x')_i \le \varepsilon p,$$

confirming our claim that $\Phi(x)$ and $\Phi(x')$ tend to be ε-orthogonal provided that x and x' are δ-separated. Similarly, (2.2) gives

$$\mathbb{E}\,\Phi(x)_i \Phi(x')_i \lesssim p^2,$$

confirming our claim that $\Phi(x)$ and $\Phi(x')$ tend to be $(p = 1/\sqrt{n})$-orthogonal provided that x and x' are ε-orthogonal.

2.2. Perception. As we just saw, the enrichment process transforms our data points x_k into $O(1/\sqrt{n})$-orthogonal vectors v_k. Let us now find a *perception map* Ψ that can fit the labels to the data: $\Psi(v_k) = y_k$.

Consider the random vector

$$w := \sum_{i=1}^{K} \pm y_i v_i,$$

where the signs are independently chosen with probability $1/2$ each. Then, separating the kth term from the sum defining w and assuming for simplicity that v_k are unit vectors, we get

$$\langle w, v_k \rangle = \pm y_k + \sum_{i:\, i \neq k} \pm y_i \langle v_i, v_k \rangle =: y_k + \text{noise}.$$

Taking the expectation over independent signs, we see that

$$\mathbb{E}(\text{noise})^2 = \sum_{i:\, i \neq k} y_i^2 \langle v_i, v_k \rangle^2,$$

where $y_i^2 \in \{0, 1\}$ and $\langle v_i, v_k \rangle^2 = O(1/n)$ by almost orthogonality. Hence,

$$\mathbb{E}(\text{noise})^2 \lesssim K/n = o(1)$$

if $K \ll n$. This yields $\langle w, v_k \rangle = \pm y_k + o(1)$, or

$$|\langle w, v_k \rangle| = y_k + o(1).$$

Since $y_k \in \{0, 1\}$, the "mirror perceptron"[6]

$$\Psi(v) := \mathbf{1}_{\{\langle w, v \rangle > 1/2\}} + \mathbf{1}_{\{-\langle w, v \rangle > 1/2\}}$$

fits the data exactly: $\Psi(v_k) = y_k$.

2.3. Deeper networks. The same argument can be repeated for networks with variable sizes of layers, i.e., $[n, m, d, 1]$. Interestingly, the enrichment works fine even if $n \ll m \ll d$, making the lower-dimensional data almost orthogonal even in very high dimensions. This explains why (moderate) bottlenecks—narrow layers—do not restrict memory capacity.

The argument we outlined allows the network $[n, m, d, 1]$ to fit around d data points, which is not very surprising, since we expect the memory capacity be proportional to the number of connections and not the number of nodes. However, the power of enrichment allows us to boost the capacity using the standard method of batch learning (or distributed learning).

Let us show, for example, how the network $[n, m, d, r, 1]$ with three hidden layers can fit $K \sim dr$ data points (x_k, y_k). Partition the data into r batches each having $O(d)$ data points. Use our previous result to train each of the r neurons in the fourth layer on its own batch of $O(d)$ points, while zeroing out all labels outside that batch. Then simply sum up the results. (The details are found in Theorem 7.1.)

[6] The mirror perceptron requires not one but two neurons to implement, which is not a problem for us.

This result can be extended to deeper networks using *stacking*, or unrolling a shallow architecture into a deep architecture, thereby trading width for depth. Figure 2 gives an illustration of stacking, and Theorem 7.2 provides the details. A similar stacking construction was employed in [4].

The success of stacking indicates that *depth has no benefit* for memorization purposes: a shallow architecture $[n, m, d, r, 1]$ can memorize roughly as much data as any deep architecture with the same number of connections. It should be noted, however, that training algorithms commonly used by practitioners, i.e., variants of stochastic gradient descent, do not seem to lead to anything similar to stacking; this leaves the question of benefit of depth open.

2.4. Neural networks as preconditioners. As we explained in section 2.1, the first two hidden layers of the network act as preconditioners: they transform the input vectors x_i into vectors v_i that are almost orthogonal. Almost orthogonality facilitates memorization process in the deeper layers, as we saw in section 2.2.

The idea to keep the data well conditioned as it passes through the network is not new. The learning rate of the stochastic gradient descent (which we are not dealing with here) is related to how well conditioned is the so-called gradient Gram matrix H. In the simplest scenario where the activation is ReLU and the network has one hidden layer of infinite size, H is a $K \times K$ matrix with entries

$$H_{ij} = \mathbb{E}\langle x_i, x_j\rangle \mathbf{1}_{\{\langle g,x_i\rangle>0,\, \langle g,x_j\rangle>0\}}, \quad \text{where} \quad g \sim N(0, I_{n_1}).$$

Much effort was made recently to prove that H is well conditioned, i.e., its smallest singular value of H is bounded away from zero, since this can be used to establish a good convergence rate for the stochastic gradient descent; see the papers cited in section 1.3 and especially [1, 10, 9, 19]. However, existing results that prove that H is well conditioned only hold for very overparametrized networks, requiring at least n_1^4 nodes in the hidden layer [19]. This is a much stronger overparametrization requirement than in our Theorem 1.1.

On the other hand, as opposed to many of the results quoted above, Theorem 1.1 does *not* shed any light on the behavior of *stochastic gradient descent*, the most popular method for training deep networks. Instead of training the weights, we explicitly compute them from the data. This allows us to avoid dealing with the gradient Gram matrix: our enrichment method provides an explicit way to make the data well conditioned. This is achieved by *setting the biases high enough* to enforce sparsity. It would be interesting to see if similar preconditioning guarantees can be achieved with small (or even zero) biases and thus without exploiting sparsity.

A different form of enrichment was developed recently in the paper [4] which showed that a neural network can compute a lot of different Boolean functions. Toward this goal, an enrichment map was implemented in the first hidden layer. The objective of this map is to transform the input set (the Boolean cube $\{0,1\}^n$) into a set $S \subset \{0,1\}^m$ on which there are lots of different threshold functions—so that the next layers can automatically compute lots of different Boolean functions. While the general goal of the enrichment map in [4] is the same as in the present paper—to achieve a more robust data representation that is passed to deeper layers—the constructions of these two enrichment maps are quite different.

2.5. Further observations and questions. As we saw in the previous section, we utilized the first two hidden layers of the network to preprocess, or enrich, the data vectors x_k. This made us skip the sizes of the first two layers when we counted the number of connections W. If these vectors are already nice, no enrichment may be necessary, and we have a higher memory capacity.

Suppose, for example, that the data vectors x_k in Theorem 1.1 are $O(1/\sqrt{n_\infty})$-orthogonal. Then, since no enrichment is needed in this case, the conclusion of the theorem holds with

$$W = n_1 n_2 + \cdots + n_{L-1} n_L,$$

which is the sum of *all* connections in the network.

If, on the other hand, the data vectors x_k in Theorem 1.1 are only $O(1/\sqrt{\log n_\infty})$-orthogonal, just the second enrichment is needed, and so the conclusion of the theorem holds with

$$W = n_2 n_3 + \cdots + n_{L-1} n_L,$$

which is the sum of all connections between the noninput layers.

This make us wonder: can enrichment be always achieved in one step instead of two? Can one find a pseudolinear map $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ that transforms a given set of δ-separated vectors x_k (say, for $\delta = 0.01$) into a set of $O(1/\sqrt{n})$-orthogonal vectors? If this is possible, we would not need to exclude the second layer from the parameter count, and Theorem 1.1 would hold for $W := n_2 n_3 + \cdots + n_{L-1} n_L$.

A related question for further study is to find an optimal separation threshold δ in the assumption $\|x_i - x_j\|_2 \geq \delta$ in Theorem 1.1 and to remove the assumption that x_k are unit vectors. Both the logarithmic separation level of δ and the normalization requirement could be artifacts of the enrichment scheme we used.

There are several ways Theorem 1.1 could be extended. It should not be too difficult, for example, to allow the output layer have more than one node; such *multioutput networks* are used in classification problems with multiple classes.

Finally, it should be possible to extend the analysis for *completely general activation functions*. Threshold activations we treated are conceptually the hardest case, since they act as extreme quantizers that restrict the flow of information through the network in the most dramatic way.

2.6. The rest of the paper. In section 3 we prove bounds (2.1) and (2.2) which control $\mathbb{E}\,\Phi(x)_i \Phi(x')_i$, the correlation of the coefficients of the coordinates of $\Phi(x)$ and $\Phi(x')$. This immediately controls the expected inner product $\mathbb{E}\langle \Phi(x), \Phi(x') \rangle = \sum_i \mathbb{E}\,\Phi(x)_i \Phi(x')_i$. In section 4 we develop a deviation inequality to make sure that the inner product $\langle \Phi(x), \Phi(x') \rangle$ is close to its expectation with high probability. In section 5, we take a union bound over all data points x_k and thus control all inner products $\langle \Phi(x_i), \Phi(x_j) \rangle$ simultaneously. This demonstrates how enrichment maps Φ make the data almost orthogonal—the property we outlined in section 2.1. In section 6, we construct a random perception map Ψ as we outlined in section 2.2. We combine enrichment and perception in section 7 as we outlined in section 2.3. We first prove a version of our main result for networks with three hidden layers (Theorem 7.1); then we stack shallow networks into an arbitrarily deep architecture, proving a full version of our main result in Theorem 7.2.

In the rest of the paper, positive absolute constants will be denoted C, c, C_1, c_1, etc. The notation $f(x) \lesssim g(x)$ means that $f(x) \le Cg(x)$ for some absolute constant C and for all values of parameter x. Similarly, $f(x) \asymp g(x)$ means that $cg(x) \le f(x) \le Cg(x)$, where c is another positive absolute constant.

We call a map E *almost pseudolinear* if $E(x) = \lambda\Phi(x)$ for some nonnegative constant λ and some pseudolinear map Φ. For the ReLU nonlinearity, almost pseudolinear maps are automatically pseudolinear, but for the threshold nonlinearity this is not necessarily the case.

3. Correlation decay.

Let $g \sim N(0, I_n)$, and consider the random process

$$Z_x := \phi(\langle g, x\rangle - b) \tag{3.1}$$

which is indexed by points x on the unit Euclidean sphere in $\mathbb{R}^n$. Here ϕ can be either the threshold or ReLU nonlinearity as in (1.1), and $b \in \mathbb{R}$ is a fixed value. Due to rotation invariance, the correlation of Z_x and $Z_{x'}$ only depends on the distance between x and x'. Although it seems to be difficult to compute this dependence exactly, we will demonstrate that the correlation of Z_x and $Z_{x'}$ decays rapidly in b. We will prove this in two extreme regimes—where x and x' are just a little separated, and where x and x' are almost orthogonal.

3.1. Correlation for separated vectors.

Cauchy–Schwarz inequality gives a trivial bound

$$\mathbb{E}\, Z_x Z_{x'} \le \mathbb{E}\, Z_x^2$$

with equality when $x = x'$. Our first result shows that if the vectors x and x' are δ-separated, this bound can be dramatically improved, and we have

$$\mathbb{E}\, Z_x Z_{x'} \le 2\exp(-b^2\delta^2/8)\, \mathbb{E}\, Z_x^2.$$

Proposition 3.1 (correlation for separated vectors). *Consider a pair of unit vectors $x, x' \in \mathbb{R}^n$, and let $b \in \mathbb{R}$ be a number that is larger than a certain absolute constant. Then*

$$\mathbb{E}\,\phi\big(\langle g, x\rangle - b\big)\,\phi\big(\langle g, x'\rangle - b\big) \le 2\exp\left(-\frac{b^2\,\|x - x'\|_2^2}{8}\right)\mathbb{E}\,\phi(\gamma - b)^2,$$

where $g \sim N(0, I_n)$ and $\gamma \sim N(0, 1)$.

Proof. Step 1. *Orthogonal decomposition.* Consider the vectors

$$u := \frac{x + x'}{2}, \quad v := \frac{x - x'}{2}.$$

Then u and v are orthogonal and $x = u + v$, $x' = u - v$. We claim that

$$\phi\big(\langle z, x\rangle - b\big)\,\phi\big(\langle z, x'\rangle - b\big) \le \big(\phi\big(\langle z, u\rangle - b\big)\big)^2 \quad \text{for any } z \in \mathbb{R}^n. \tag{3.2}$$

To check this claim, note that if both $\langle z, x\rangle$ and $\langle z, x'\rangle$ are greater than b, so is $\langle z, u\rangle$. Expressing this implication as

$$\mathbf{1}_{\langle z,x\rangle > b}\,\mathbf{1}_{\langle z,x'\rangle > b} \le \mathbf{1}_{\langle z,u\rangle > b}, \tag{3.3}$$

we conclude that (3.2) holds for the threshold nonlinearity $\phi(t) = \mathbf{1}_{\{t>0\}}$.

To prove (3.2) for the ReLU nonlinearity, note that

$$\big(\langle z,x\rangle - b\big)\big(\langle z,x'\rangle - b\big) = \big(\langle z,u+v\rangle - b\big)\big(\langle z,u-v\rangle - b\big) = \big(\langle z,u\rangle - b\big)^2 - \langle z,v\rangle^2 \le \big(\langle z,u\rangle - b\big)^2.$$

Combine this bound with (3.3) to get

$$\big(\langle z,x\rangle - b\big)\big(\langle z,x'\rangle - b\big)\,\mathbf{1}_{\langle z,x\rangle>b}\,\mathbf{1}_{\langle z,x'\rangle>b} \le \big(\langle z,u\rangle - b\big)^2\,\mathbf{1}_{\langle z,u\rangle>b}.$$

This yields (3.2) for the ReLU nonlinearity $\phi(t) = t\,\mathbf{1}_{\{t>0\}}$.

Step 2. *Taking expectation.* Substitute $z = g \sim N(0, I_n)$ into the bound (3.2), and take expectation on both sides. We get

$$\mathbb{E}\,\phi\big(\langle g,x\rangle - b\big)\,\phi\big(\langle g,x'\rangle - b\big) \le \mathbb{E}\,\phi\big(\langle g,u\rangle - b\big)^2. \tag{3.4}$$

Denote

$$\delta := \|v\|_2 = \frac{\|x - x'\|_2}{2}.$$

Since $x = u + v$ is a unit vector and u, v are orthogonal, we have $1 = \|u\|_2^2 + \|v\|_2^2$, and thus $\|u\|_2 = \sqrt{1-\delta^2}$. Therefore, the random variable $\langle g,u\rangle$ in the right side of (3.4) is distributed identically with $\gamma\sqrt{1-\delta^2}$ where $\gamma \sim N(0,1)$, and we obtain

$$\mathbb{E}\,\phi\big(\langle g,x\rangle - b\big)\,\phi\big(\langle g,x'\rangle - b\big) \le \mathbb{E}\,\phi\big(\gamma\sqrt{1-\delta^2} - b\big)^2. \tag{3.5}$$

Step 3. *Stability.* Now use the stability property of the normal distribution, which we state in Lemma A.2. For $a = b$ larger than a suitable absolute constant, $z = -\delta^2$, and for either the threshold or ReLU nonlinearity ϕ, we see that

$$\frac{\mathbb{E}\,\phi\left(\gamma\sqrt{1-\delta^2} - b\right)^2}{\mathbb{E}\,\phi(\gamma - b)^2} \le 2\exp\left(-\frac{b^2\delta^2}{2(1-\delta^2)}\right)(1-\delta^2)^{3/2} \le 2\exp\left(-\frac{b^2\delta^2}{2}\right).$$

Combine this with (3.5) to complete the proof. ∎

3.2. Correlation for almost orthogonal vectors. We continue to study the covariance structure of the random process Z_x. If x and x' are orthogonal, Z_x and $Z_{x'}$ are independent, and we have

$$\mathbb{E}\,Z_x Z_{x'} = \big[\mathbb{E}\,Z_x\big]^2.$$

In this subsection, we show the stability of this equality. The result below implies that if x and x' are almost orthogonal, namely, $\big|\langle x,x'\rangle\big| \ll b^{-2}$, then

$$\mathbb{E}\,Z_x Z_{x'} \lesssim \big[\mathbb{E}\,Z_x\big]^2.$$

Proposition 3.2 (correlation for ε-orthogonal vectors). *Consider a pair of vectors $u, u' \in \mathbb{R}^m$ satisfying*

$$\big|\|u\|_2^2 - 1\big| \le \varepsilon, \quad \big|\|u'\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u,u'\rangle\big| \le \varepsilon$$

for some $\varepsilon \in (0, 1/8)$. Let $b \in \mathbb{R}$ be a number that is larger than a certain absolute constant. Then

$$\mathbb{E}\,\phi\big(\langle g, u\rangle - b\big)^2 \ge \frac{1}{2}\exp(-b^2\varepsilon)\ \mathbb{E}\,\phi(\gamma - b)^2;$$
$$\mathbb{E}\,\phi\big(\langle g, u\rangle - b\big)\,\phi\big(\langle g, u'\rangle - b\big) \le 2\exp(2b^2\varepsilon)\,\big[\mathbb{E}\,\phi(\gamma - b)\big]^2,$$

where $g \sim N(0, I_m)$ and $\gamma \sim N(0, 1)$.

In order to prove this proposition, we first establish a more general stability property.

Lemma 3.3. *Let $\varepsilon \in (0, 1/2)$, and let u, u', g, and γ be as in Proposition 3.2. Then, for any measurable function $\psi : \mathbb{R} \to [0, \infty)$ we have*

$$\mathbb{E}\,\psi(\langle g, u\rangle)\,\psi(\langle g, u'\rangle) \le \sqrt{\frac{1+2\varepsilon}{1-2\varepsilon}}\,\Big[\mathbb{E}\,\psi\Big(\gamma\sqrt{1+2\varepsilon}\Big)\Big]^2.$$

Proof. Consider the $2 \times m$ matrix A whose rows are u and u', and define the function

$$\Psi : \mathbb{R}^2 \to \mathbb{R}, \quad \Psi(x) := \psi(x_1)\,\psi(x_2).$$

Since the vector Ag has coordinates $\langle g, u\rangle$ and $\langle g, u'\rangle$, we have $\Psi(Ag) = \psi(\langle g, u\rangle)\,\psi(\langle g, u'\rangle)$. Thus

$$\mathbb{E}\,\psi(\langle g, u\rangle)\,\psi(\langle g, u'\rangle) = \mathbb{E}\,\Psi(Ag) = \frac{1}{2\pi\sqrt{\det(\Sigma)}}\int_{\mathbb{R}^2}\Psi(x)\exp\Big(-\frac{x^{\mathsf{T}}\Sigma^{-1}x}{2}\Big)\,dx, \tag{3.6}$$

where

$$\Sigma = \mathrm{Cov}(Ag) = AA^{\mathsf{T}} = \begin{bmatrix} \|u\|_2^2 & \langle u, u'\rangle \\ \langle u, u'\rangle & \|u'\|_2^2 \end{bmatrix}.$$

The assumptions on z, z' then give

$$\det(\Sigma) \ge 1 - 2\varepsilon \quad \text{and} \quad x^{\mathsf{T}}\Sigma^{-1}x \ge \frac{\|x\|_2^2}{1+2\varepsilon} \quad \text{for all } x \in \mathbb{R}^2. \tag{3.7}$$

Indeed, the first bound is straightforward. To verify the second bound, note that each entry of the matrix $\Sigma - I_2$ is bounded in absolute value by ε. Thus, the operator norm of $\Sigma - I_2$ is bounded by 2ε, which we can write as $\Sigma - I_2 \preceq 2\varepsilon I_2$ in the positive-semidefinite order. This implies that $\Sigma^{-1} \succeq (1+2\varepsilon)^{-1} I_2$, and multiplying both sides of this relation by x^{T} and x, we get the second bound in (3.7).

Substitute (3.7) into (3.6) to obtain

$$\begin{aligned}\mathbb{E}\,\psi(\langle g, x\rangle)\,\psi(\langle g, x'\rangle) &\ge \frac{1}{2\pi\sqrt{1-2\varepsilon}}\int_{\mathbb{R}^2}\Psi(x)\exp\Big(-\frac{\|x\|_2^2}{2(1+2\varepsilon)}\Big)\,dx \\ &= \sqrt{\frac{1+2\varepsilon}{1-2\varepsilon}}\,\mathbb{E}\,\Psi\Big(h\sqrt{1+2\varepsilon}\Big),\end{aligned}$$

where $h = (h_1, h_2) \sim N(0, I_2)$.

It remains to recall that $\Psi(x) = \psi(x_1)\,\psi(x_2)$, so

$$\mathbb{E}\,\Psi\left(h\sqrt{1+2\varepsilon}\right) = \mathbb{E}\,\psi\left(h_1\sqrt{1+2\varepsilon}\right)\,\psi\left(h_2\sqrt{1+2\varepsilon}\right) = \left[\,\mathbb{E}\,\psi\left(\gamma\sqrt{1+2\varepsilon}\right)\,\right]^2$$

by independence. Lemma 3.3 is proved. ∎

Proof of Proposition 3.2. By assumption, $\|u\|_2 \geq \sqrt{1-\varepsilon}$, so

$$\mathbb{E}\,\phi\big(\langle g,u\rangle - b\big)^2 = \mathbb{E}\,\phi\big(\gamma\|u\|_2 - b\big)^2 \geq \mathbb{E}\,\phi\left(\gamma\sqrt{1-\varepsilon} - b\right)^2, \tag{3.8}$$

where the last inequality follows by monotonicity; see Lemma A.3 for justification. Now we use the stability property of the normal distribution that we state in Lemma A.2. For $a = b$ larger than a suitable absolute constant and $z = -\varepsilon$, it gives for both threshold and ReLU nonlinearities the following:

$$\frac{\mathbb{E}\,\phi\left(\gamma\sqrt{1-\varepsilon} - b\right)^2}{\mathbb{E}\,\phi(\gamma-b)^2} \geq 0.9\exp\left(-\frac{b^2\varepsilon}{2(1-\varepsilon)}\right)(1-\varepsilon)^{3/2} \geq \frac{1}{2}\exp(-b^2\varepsilon),$$

where the last bound follows since $\varepsilon \leq 1/8$. Combining this with (3.8), we obtain the first conclusion of the lemma.

Next, Lemma 3.3 gives

$$\mathbb{E}\,\phi\big(\langle g,u\rangle - b\big)\,\big(\langle g,u'\rangle - b\big) \leq \sqrt{\frac{1+2\varepsilon}{1-2\varepsilon}}\,\left[\,\mathbb{E}\,\phi\left(\gamma\sqrt{1+2\varepsilon} - b\right)\right]^2. \tag{3.9}$$

Now we again use the stability property of the normal distribution, Lemma A.2, this time for $z = 2\varepsilon$. It gives for both threshold and ReLU nonlinearities the following:

$$\frac{\mathbb{E}\,\phi\left(\gamma\sqrt{1+2\varepsilon} - b\right)}{\mathbb{E}\,\phi(\gamma-b)} \leq 1.01\exp\left(\frac{2b^2\varepsilon}{2(1+2\varepsilon)}\right)(1+2\varepsilon)^{3/2} \leq 1.01(1+2\varepsilon)^{3/2}\exp(b^2\varepsilon).$$

Combining this with (3.9) gives

$$\frac{\mathbb{E}\,\phi\big(\langle g,u\rangle - b\big)\,\phi\big(\langle g,u'\rangle - b\big)}{\big[\,\mathbb{E}\,\phi(\gamma-b)\big]^2} \leq \sqrt{\frac{1+2\varepsilon}{1-2\varepsilon}}\Big(1.01(1+2\varepsilon)^{3/2}\exp(b^2\varepsilon)\Big)^2 \leq 2\exp(2b^2\varepsilon),$$

where the last step follows since $\varepsilon \leq 1/8$. This completes the proof of Proposition 3.2. ∎

4. Deviation. In the previous section, we studied the covariance of the random process

$$Z_x := \phi(\langle g,x\rangle - b), \quad x \in \mathbb{R}^n,$$

where ϕ is either the threshold or ReLU nonlinearity as in (1.1), $g \sim N(0, I_n)$ is a standard normal random variable, and $b \in \mathbb{R}$ is a fixed value. Consider a multivariate version of this

process, a random pseudolinear map $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ whose m components are independent copies of Z_x. In other words, define

$$\Phi(x) := \Big(\phi\big(\langle g_i, x\rangle - b\big)\Big)_{i=1}^m \quad \text{for } x \in \mathbb{R}^n,$$

where $g_i \sim N(0, I_n)$ are independent standard normal random vectors.

We are interested in how the map Φ transforms the distances between different points. Since

$$\mathbb{E}\langle \Phi(x), \Phi(x')\rangle = m\, \mathbb{E}\, Z_x Z_{x'},$$

the bounds on $\mathbb{E}\, Z_x Z_{x'}$ we proved in the previous section describe the behavior of Φ in expectation. In this section, we use standard concentration inequalities to ensure a similar behavior with high probability.

Lemma 4.1 (deviation). *Consider a pair of vectors $x, x' \in \mathbb{R}^n$ such that $\|x\|_2 \le 2$, $\|x'\|_2 \le 2$, and let $b \in \mathbb{R}$ be a number that is larger than a certain absolute constant. Define*

$$p := \mathbb{E}\, \phi\big(\langle g, x\rangle - b\big)\, \phi\big(\langle g, x'\rangle - b\big), \quad \text{where } g \sim N(0, I_n). \tag{4.1}$$

Then for every $N \ge 2$, with probability at least $1 - 2mN^{-5}$ we have

$$\big|\langle \Phi(x), \Phi(x')\rangle - mp\big| \le C_1\Big(\sqrt{mp}\, \log N + \log^2 N\Big).$$

Proof. Step 1. *Decomposition and truncation.* By construction, $\mathbb{E}\langle \Phi(x), \Phi(x')\rangle = mp$. The deviation from the mean is

$$\langle \Phi(x), \Phi(x')\rangle - mp = \sum_{i=1}^m \phi\big(\gamma_i - b\big)\, \phi\big(\gamma_i' - b\big) - mp, \tag{4.2}$$

where $\gamma_i := \langle g_i, x\rangle$ and $\gamma_i' := \langle g_i, x'\rangle$. These two normal random variables are possibly correlated, and each has zero mean and variance bounded by 4.

We will control the sum of i.i.d. random variables in (4.2) using Bernstein's concentration inequality. In order to apply it, we first perform a standard truncation of the terms of the sum. The level of truncation will be

$$M := C_2\sqrt{\log N}, \tag{4.3}$$

where C_2 is a sufficiently large absolute constant. Consider the random variables

$$Z_i := \phi(\gamma_i - b)\, \phi(\gamma_i' - b)\, \mathbf{1}_{\{\gamma_i \le M \text{ and } \gamma_i' \le M\}},$$
$$R_i := \phi(\gamma_i - b)\, \phi(\gamma_i' - b)\, \mathbf{1}_{\{\gamma_i > M \text{ or } \gamma_i' > M\}}.$$

Then we can decompose the sum in (4.2) as follows:

$$\langle \Phi(x), \Phi(x')\rangle - mp = \sum_{i=1}^m (Z_i - \mathbb{E}\, Z_i) + \sum_{i=1}^m (R_i - \mathbb{E}\, R_i). \tag{4.4}$$

Step 2. *The residual is small.* Let us first control the residual, i.e., the second sum on the right side of (4.4). For a fixed i, the probability that R_i is nonzero can be bounded by

$$\mathbb{P}\left\{\gamma_i > M \text{ or } \gamma_i' > M\right\} \le \mathbb{P}\left\{\gamma_i > M\right\} + \mathbb{P}\left\{\gamma_i' > M\right\} \le 2\mathbb{P}\left\{\gamma > M/2\right\} \le N^{-10}, \tag{4.5}$$

where $\gamma \sim N(0,1)$. In the second inequality, we used that γ_i and γ_i' are normal with mean zero and variance at most 4. The third inequality follows from the asymptotic (A.2) on the tail of the normal distribution and our choice (4.3) of the truncation level M with sufficiently large C_0.

Taking the union bound we see that all R_i vanish simultaneously with probability at least $1 - mN^{-10}$. Furthermore, by monotonicity,

$$\begin{aligned} \mathbb{E}\, R_i &\le \mathbb{E}\, \phi(\gamma_i)\, \phi(\gamma_i')\, \mathbf{1}_{\{\gamma_i > M \text{ or } \gamma_i' > M\}} \\ &\le \left(\mathbb{E}\, \phi(\gamma_i)^4\right)^{1/4} \left(\mathbb{E}\, \phi(\gamma_i')^4\right)^{1/4} \left(\mathbb{P}\left\{\gamma_i > M \text{ or } \gamma_i' > M\right\}\right)^{1/2}, \end{aligned} \tag{4.6}$$

where in the last step we used generalized Hölder's inequality. Now, for the threshold nonlinearity $\phi(t) = \mathbf{1}_{\{t>0\}}$, the terms $\mathbb{E}\, \phi(\gamma_i)^4$ and $\mathbb{E}\, \phi(\gamma_i')^4$ obviously equal $1/2$, and for the ReLU nonlinearity $\phi(t) = t_+$ these terms are bounded by the fourth moment of the standard normal distribution, which equals 3. Combining this with (4.5) gives

$$0 \le \mathbb{E}\, R_i \le 2N^{-5}.$$

Summarizing, with probability at least $1 - mN^{-10}$, we have

$$\left|\sum_{i=1}^m (R_i - \mathbb{E}\, R_i)\right| = \left|\sum_{i=1}^m \mathbb{E}\, R_i\right| \le 2mN^{-5} \le 1. \tag{4.7}$$

The last bound holds because otherwise we have $1 - 2mN^{-5} < 0$, and the statement of the proposition holds trivially.

Step 3. *The main sum is concentrated.* To bound the first sum in (4.4), we can use Bernstein's inequality [23], which we can state as follows. If $Z_1, \dots, Z_m$ are independent random variables and $s \ge 0$, then with probability at least $1 - 2e^{-s}$ we have

$$\left|\sum_{i=1}^m (Z_i - \mathbb{E}\, Z_i)\right| \lesssim \sigma\sqrt{s} + Ks, \tag{4.8}$$

where $\sigma^2 = \sum_{i=1}^m \operatorname{Var}(Z_i)$ and $K = \max_i \|Z_i\|_\infty$. In our case, it is easy to check that for both threshold and ReLU nonlinearities ϕ, we have

$$K = \|Z_1\|_\infty \le \phi(M - b)^2 \le M^2,$$

and

$$\sigma^2 = m \operatorname{Var}(Z_1) \le m\, \mathbb{E}\, Z_1^2 \le M^2 m\, \mathbb{E}\, Z_1 \le M^2 m\, \mathbb{E}\, \phi(\gamma_1 - b)\, \phi(\gamma_1' - b) = M^2 mp$$

by definition of p in (4.1). Apply Bernstein's inequality (4.8) for

$$s = C_3 \log N, \tag{4.9}$$

where C_3 is a suitably large absolute constant. We obtain that with probability at least $1 - 2e^{-s} \geq 1 - N^{-10}$,

$$\left| \sum_{i=1}^{m} (Z_i - \mathbb{E}\, Z_i) \right| \lesssim \sqrt{M^2 mps} + M^2 s \lesssim \sqrt{mp}\, \log N + \log^2 N. \tag{4.10}$$

Here we used the choice of M we made in (4.3) and s in (4.9).

Combining the bounds on the residual (4.7) and on the main sum (4.10) and putting them into the decomposition (4.4), we conclude that with probability at least $1 - 2mN^{-10}$,

$$\big|\langle \Phi(x), \Phi(x') \rangle - mp\big| \lesssim \sqrt{mp}\, \log N + \log^2 N + 1 \lesssim \sqrt{mp}\, \log N + \log^2 N.$$

The proof is complete. ∎

5. Enrichment.

In the previous section, we defined a random pseudolinear map

$$\Phi : \mathbb{R}^n \to \mathbb{R}^m, \quad \Phi(x) := \Big(\phi\big(\langle g_i, x \rangle - b \big) \Big)_{i=1}^{m}, \tag{5.1}$$

where ϕ is either the threshold or ReLU nonlinearity as in (1.1), $g_i \sim N(0, I_n)$ are independent standard normal random vectors, and b is a fixed value.

We will now demonstrate the ability of Φ to "enrich" the data, to move different points away from each other. To see why this could be the case, choose the value of b to be moderately large, say, $b = 100\sqrt{\log m}$. Then with high probability, most of the random variables $\langle g_i, x \rangle$ will fall below b, making most of the coordinates $\Phi(x)$ equal zero, thus making $\Phi(x)$ a random sparse vector. Sparsity will tend to make $\Phi(x)$ and $\Phi(x')$ almost orthogonal even when x and x' are just a little separated from each other.

To make this rigorous, we can use the results of section 3 to check that for such b, the coordinates of $\Phi(x)$ and $\Phi(x')$ are almost uncorrelated. This immediately implies that $\Phi(x)$ and $\Phi(x')$ are almost orthogonal in expectation, and the deviation inequality from section 4 then implies that the same holds with high probability. This allows us to take a union bound over all data points x_i and conclude that $\Phi(x_i)$ and $\Phi(x_j)$ are almost orthogonal for all distinct data points.

As in section 3, we will prove this in two regimes, first for the data points that are just a little separated, and then for the data points that are almost orthogonal.

5.1. From separated to ε-orthogonal.

In this part we show that the random pseudolinear map Φ transforms separated data points into almost orthogonal points.

Lemma 5.1 (enrichment I: from separated to ε-orthogonal). *Consider a pair of unit vectors $x, x' \in \mathbb{R}^n$ satisfying*

$$\|x - x'\|_2 \geq C_2 \sqrt{\frac{\log(1/\varepsilon)}{\log m}} \tag{5.2}$$

for some $\varepsilon \in [m^{-1/5}, 1/8]$. *Let* $2 \le N \le \exp(m^{1/5})$, *and let* p *and* b *be numbers such that*

$$p = \frac{C_2 \log^2 N}{\varepsilon^2 m} = \mathbb{E}\,\phi(\gamma - b)^2.$$

Consider the random pseudolinear map $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ *defined in* (5.1). *Then with probability at least* $1 - 4mN^{-5}$, *the vectors*

$$u := \frac{\Phi(x)}{\sqrt{mp}}, \quad u' := \frac{\Phi(x')}{\sqrt{mp}}$$

satisfy

$$\big|\|u\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u, u'\rangle\big| \le \varepsilon.$$

Proof. Step 1. *Bounding the bias* b. We begin with some easy observations. Note that $\|x - x'\|_2$ is bounded above by 2 and below by $C_2/\sqrt{\log m}$. Thus, by setting the value of C_2 sufficiently large, we can assume that m is arbitrarily large, i.e., larger than any given absolute constant. Furthermore, the restrictions on ε and N yield

$$m^{-1} \le p \lesssim m^{-1/10}, \tag{5.3}$$

so p is arbitrarily small, smaller than any given absolute constant. The function $t \mapsto \mathbb{E}\,\phi(\gamma - t)^2$ is continuous, takes an absolute constant value at $t = 0$, and tends to zero as $t \to \infty$. Thus the equation $\mathbb{E}\,\phi(\gamma - t)^2 = p$ has a solution, so b is well defined and $b \ge 1$.

To get a better bound on b, one can use (A.2) for the threshold nonlinearity and Lemma A.1 for ReLU, which give

$$\log \mathbb{E}\,\phi(\gamma - b)^2 \asymp -b^2.$$

Since $p = \mathbb{E}\,\phi(\gamma - b)^2$, this and (5.3) yield

$$b \asymp \sqrt{\log m}. \tag{5.4}$$

Step 2. *Controlling the norm.* Applying Lemma 4.1 for $x = x'$, we obtain with probability at least $1 - 2mN^{-5}$ that

$$\big|\|\Phi(x)\|_2^2 - mp\big| \le C_1\Big(\sqrt{mp}\,\log N + \log^2 N\Big).$$

Divide both sides by mp to get

$$\big|\|u\|_2^2 - 1\big| \le C_1\left(\frac{\log N}{\sqrt{mp}} + \frac{\log^2 N}{mp}\right) \le \varepsilon,$$

where the second inequality follows from our choice of p with large C_2. We proved the first conclusion of the proposition.

Step 3. *Controlling the inner product.* Proposition 3.1 gives

$$q := \mathbb{E}\,\phi\big(\langle g, x\rangle - b\big)\,\phi\big(\langle g, x'\rangle - b\big) \le 2\exp\left(-\frac{b^2\,\|x - x'\|_2^2}{8}\right)p \le \varepsilon^{10} p, \tag{5.5}$$

where in the last step we used the bounds (5.4) on b and the separation assumption (5.2) with a sufficiently large constant C_2. Now, applying Lemma 4.1, we obtain with probability at least $1 - 2mN^{-5}$ that

$$\big|\langle \Phi(x), \Phi(x')\rangle\big| \le mq + C_1\Big(\sqrt{mq}\,\log N + \log^2 N\Big).$$

Divide both sides by mp to obtain

$$\big|\langle u, u'\rangle\big| \le \frac{q}{p} + C_1\Big(\frac{\sqrt{q}\log N}{\sqrt{mp}} + \frac{\log^2 N}{mp}\Big) \le \varepsilon^2,$$

where the last step follows from the bound (5.5) on q and our choice of p with a sufficiently large C_2. This is an even stronger bound than we claimed. ∎

Theorem 5.2 (enrichment I: from separated to ε-orthogonal). *Consider unit vectors $x_1, \dots, x_K \in \mathbb{R}^n$ that satisfy*

$$\|x_i - x_j\|_2 \ge C_2\sqrt{\frac{\log(1/\varepsilon)}{\log m}}$$

for all distinct i, j, where $\varepsilon \in [m^{-1/5}, 1/8]$. Assume that $K \le \exp(c_2 m^{1/5})$. Then there exists an almost[7] pseudolinear map $E : \mathbb{R}^n \to \mathbb{R}^m$ such that the vectors $u_k := E(x_k)$ satisfy

$$\big|\|u_i\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u_i, u_j\rangle\big| \le \varepsilon$$

for all distinct indices $i, j = 1, \dots, K$.

Proof. Apply Lemma 5.1 followed by a union bound over all pairs of distinct vectors x_k. If we chose $N = 2mK$, then the probability of success is at least $1 - K^2 \cdot 4m(2mK)^{-5} > 0$. The proof is complete. ∎

5.2. From ε-orthogonal to $\frac{1}{\sqrt{d}}$-orthogonal. In this part we show that a random pseudolinear map $\Phi : \mathbb{R}^m \to \mathbb{R}^d$ makes almost orthogonal data points even closer to being orthogonal: Φ reduces the inner products from a small constant ε to $O(1/\sqrt{d})$.

The pseudolinear map Φ considered in this part will have the same form as in (5.1), but for different dimensions:

$$\Phi : \mathbb{R}^m \to \mathbb{R}^d, \quad \Phi(u) := \Big(\phi\big(\langle g_i, u\rangle - b\big)\Big)_{i=1}^m, \tag{5.6}$$

where ϕ is either the threshold or ReLU nonlinearity as in (1.1), $g_i \sim N(0, I_m)$ are independent standard normal random vectors, and b is a fixed value.

Lemma 5.3 (enrichment II: from ε-orthogonal to $\frac{1}{\sqrt{d}}$-orthogonal). *Consider a pair of vectors $u, u' \in \mathbb{R}^m$ satisfying*

$$\big|\|u\|_2^2 - 1\big| \le \varepsilon, \quad \big|\|u'\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u, u'\rangle\big| \le \varepsilon \tag{5.7}$$

[7] Recall from section 2.6 that an almost pseudolinear map E is, by definition, a pseudolinear map multiplied by a nonnegative constant. In our case, $E = (mp)^{-1/2}\Phi$.

for some $0 < \varepsilon \le c_3/\log d$. *Let* $2 \le N \le \exp(c_3 d^{1/5})$, *and let* p *and* b *be numbers such that*

$$p := \frac{1}{\sqrt{d}}, \qquad \mathbb{E}\,\phi(\gamma - b)^2 = p.$$

Then with probability at least $1 - 4dN^{-5}$, *the vectors*

$$v := \frac{\Phi(u)}{\sqrt{dp}}, \quad v' := \frac{\Phi(u')}{\sqrt{dp}}$$

satisfy

$$\|v\|_2 \ge \frac{1}{2}, \quad \big|\langle v, v'\rangle\big| \le \frac{C_3(\log d + \log^2 N)}{\sqrt{d}}.$$

Proof. Step 1. *Bounding the bias* b. Following the beginning of the proof of Lemma 5.1, we can check that b exists and

$$b \asymp \sqrt{\log d}. \tag{5.8}$$

Step 2. *Controlling the norm.* Applying Proposition 3.2, we see that

$$p_0 := \mathbb{E}\,\phi\big(\langle g, u\rangle - b\big)^2 \ge \frac{1}{2}\exp(-b^2\varepsilon)p \ge \frac{p}{3},$$

where in the last step we used the bound (5.8) on b and the assumption on ε with a sufficiently small constant c_3. Then, applying Lemma 4.1 for $x = x' = u$, we obtain with probability at least $1 - 2dN^{-5}$ that

$$\|\Phi(u)\|_2^2 \ge dp_0 - C_1\left(\sqrt{dp_0}\,\log N + \log^2 N\right) \ge \frac{3}{4}dp_0 \ge \frac{1}{4}dp,$$

where we used our choice of p and the restriction on N with sufficiently small constant c_3. Divide both sides by dp to get

$$\|v\|_2 \ge \frac{1}{2},$$

which is the first conclusion of the proposition.

Step 3. *Controlling the inner product.* Proposition 3.2 gives

$$q := \mathbb{E}\,\phi\big(\langle g, u\rangle - b\big)\,\phi\big(\langle g, u'\rangle - b\big) \le 2\exp(2b^2\varepsilon)\,\big[\mathbb{E}\,\phi(\gamma - b)\big]^2 \lesssim \big[\mathbb{E}\,\phi(\gamma - b)\big]^2, \tag{5.9}$$

where the last inequality follows as before from bound (5.8) on b and the assumption on ε with sufficiently small c_3.

Next, we will use the following inequality that holds for all sufficiently large $a > 0$:

$$\mathbb{E}\,\phi(\gamma - a) \le a \cdot \mathbb{E}\,\phi(\gamma - a)^2.$$

For the threshold nonlinearity ϕ, this bound is trivial even without the factor a in the right side. For the ReLU nonlinearity, it follows from Lemma A.1 in the appendix. Therefore, we have

$$\mathbb{E}\,\phi(\gamma - b) \le bp \lesssim p\sqrt{\log d},$$

where we used (5.8) in the last step. Substituting this into (5.9), we conclude that

$$q \lesssim p^2 \log d. \tag{5.10}$$

Now, applying Lemma 4.1, we obtain with probability at least $1 - 2mN^{-5}$ that

$$\big|\langle \Phi(u), \Phi(u')\rangle\big| \lesssim dq + \sqrt{dq}\,\log N + \log^2 N.$$

Divide both sides by dp to obtain

$$\big|\langle v, v'\rangle\big| \lesssim \frac{q}{p} + \frac{\sqrt{q}\log N}{\sqrt{d}p} + \frac{\log^2 N}{dp} \lesssim \frac{\log d + \log^2 N}{\sqrt{d}},$$

where in the last step we used (5.10) and our choice of p. ∎

Theorem 5.4 (enrichment II: from ε-orthogonal to $\frac{1}{\sqrt{d}}$-orthogonal). *Consider vectors $u_1, \dots, u_K \in \mathbb{R}^n$ that satisfy*

$$\big|\|u_i\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u_i, u_j\rangle\big| \le \varepsilon$$

for all distinct i, j, where $0 < \varepsilon \le c_3/\log d$. Assume that $K \le \exp(c_3 d^{1/5})$. Then there exists an almost pseudolinear map $R : \mathbb{R}^m \to \mathbb{R}^d$ such that the vectors $v_k := R(u_k)$ satisfy

$$\|v_i\|_2 \ge \frac{1}{2}, \quad \big|\langle v_i, v_j\rangle\big| \le \frac{C_4 \log^2(dK)}{\sqrt{d}}$$

for all distinct indices $i, j = 1, \dots, K$.

Proof. Apply Lemma 5.3 followed by a union bound over all pairs of distinct vectors u_k. If we chose $N = 2dK$, then the probability of success is at least $1 - K^2 \cdot 4d(2dK)^{-5} > 0$. The proof is complete. ∎

6. Perception. The previous sections were concerned with preprocessing, or "enrichment," of the data. We demonstrated how a pseudolinear map can transform the original data points x_k, which can be just a little separated, into ε-orthogonal points u_k with $\varepsilon = o(1)$, and further into η-orthogonal points v_k with $\eta = O(1/\sqrt{d})$. In this section we train a pseudolinear map that can memorize any label assignment y_k for the η-orthogonal points v_k.

We will first try to train a single neuron to perform this task assuming that the number K of the data points v_k is smaller than the dimension d, up to a logarithmic factor. Specifically, we construct a vector $w \in \mathbb{R}^n$ so that the values $|\langle w, v_k\rangle|$ are small whenever $y_k = 0$ and large whenever $y_k = 1$. Our construction is probabilistic: we choose $w = \sum_{k=1}^K \pm y_k v_k$ with random independent signs and show that w succeeds with high probability.

Lemma 6.1 (perception). *Let $\eta \in (0, 1)$, and consider vectors $v_1, \dots, v_K \in \mathbb{R}^d$ satisfying*

$$\|v_i\|_2 \ge \frac{1}{2}, \quad \big|\langle v_i, v_j\rangle\big| \le \eta \tag{6.1}$$

for all distinct i, j. Consider any labels $y_1, \dots, y_K \in \{0, 1\}$, at most K_1 of which equal 1. Assume that $K_1 \log K \le c_4 \eta^{-2}$. Then there exists a vector $w \in \mathbb{R}^d$ that satisfies the following for every $k = 1, \dots, K$:

$$\big|\langle w, v_k\rangle\big| \le \frac{1}{16} \text{ if } y_k = 0; \qquad \big|\langle w, v_k\rangle\big| \ge \frac{3}{16} \text{ if } y_k = 1. \tag{6.2}$$

Proof. Let $\xi_1, \dots, \xi_K$ be independent Rademacher random variables, and define

$$w := \sum_{k=1}^{K} \xi_k y_k v_k.$$

We are going to show that the random vector w satisfies the conclusion of the proposition with positive probability.

Let us first check the conclusion (6.2) for $k = 1$. To this end, we decompose $\langle w, v_1 \rangle$ as follows:

$$\langle w, v_1 \rangle = \xi_1 y_1 \|v_1\|_2^2 + \sum_{k=2}^{K} \xi_k y_k \langle v_k, v_1 \rangle =: \text{signal} + \text{noise}.$$

To bound the noise, we shall use Hoeffding's inequality (see, e.g., [23, Theorem 2.2.2]), which can be stated as follows. If $a_1, \dots, a_N$ are any fixed numbers and $s \geq 0$, then with probability at least $1 - 2e^{-s^2/2}$ we have

$$\left| \sum_{k=1}^{N} \xi_k a_k \right| \leq s \left(\sum_{k=1}^{N} a_k^2 \right)^{1/2}.$$

Using this for $s = 4\sqrt{\log K}$, we conclude that with probability at least $1 - 2K^{-8}$, we have

$$|\text{noise}| \leq 4\sqrt{\log K} \left(\sum_{k=2}^{K} y_k^2 \langle v_k, v_1 \rangle^2 \right)^{1/2} \leq 4\sqrt{\log K}\, \sqrt{K_1}\eta \leq \frac{1}{16},$$

where we used (6.1) and the assumption on K, K_1 with a sufficiently small constant c_4.

If $y_1 = 0$, the signal is zero, and so $|\langle w, v_1 \rangle| = |\text{noise}| \leq 1/16$, as claimed. If $y_1 = 1$, then $|\text{signal}| = \|v_1\|_2^2 \geq 1/4$, and thus

$$\big|\langle w, v_1 \rangle\big| \geq |\text{signal}| - |\text{noise}| \geq \frac{1}{4} - \frac{1}{16} = \frac{3}{16},$$

as claimed.

Repeating this argument for general k, we can obtain the same bounds for $\big|\langle w, v_k \rangle\big|$. Finally, take the union bound over the K choices of k. The random vector satisfies the conclusion with probability at least $1 - 2K^{-7} > 0$. The proof is complete. ∎

Lemma 6.1 essentially says that one neuron can memorize the labels of $O(d)$ data points in $\mathbb{R}^d$. Thus, r neurons should be able to memorize the labels of $O(dr)$ data points in $\mathbb{R}^d$. To make this happen, we can partition the data into r batches of size $O(d)$ each and train each neuron on a different batch. The following lemma makes this formal; to see the connection, apply it for $\eta \asymp 1/\sqrt{d}$.

Theorem 6.2 (one layer). *Consider a number $\eta \in (0, 1)$, vectors $v_1, \dots, v_K \in \mathbb{R}^d$, and labels $y_1, \dots, y_K \in \{0, 1\}$ as in Lemma 6.1. Assume that $(2K_1 + r) \log K \leq c_4 r \eta^{-2}$ where r is a positive integer. Then there exists a pseudolinear map $P : \mathbb{R}^d \to \mathbb{R}^r$ such that for all $k = 1, \dots, K$ we have*

$$P(v_k) = 0 \text{ iff } y_k = 0.$$

Proof. Without loss of generality, assume that the first K_1 of the labels y_k equal 1 and the rest equal zero, i.e., $y_k = \mathbf{1}_{\{k \le K_1\}}$. Partition the indices of the nonzero labels $\{1, \dots, K_1\}$ into $r/2$ subsets I_i ("batches"), each of cardinality at most $2K_1/r + 1$. For each batch i, define a new set of labels

$$y_{ki} = \mathbf{1}_{\{k \in I_i\}}, \quad k = 1, \dots, K.$$

In other words, the labels y_{ki} are obtained from the original labels y_i by zeroing out the labels outside batch i.

For each i, apply Lemma 6.1 for the labels y_{ki}. The number of nonzero labels is $|I_i| \le 2K_1/r + 1$, so we can use this number instead of K_1, noting that the condition $(2K_1/r + 1) \log K \le c_4 \eta^{-2}$ required in Lemma 6.1 does hold by our assumption. We obtain a vector $w_i \in \mathbb{R}^d$ that satisfies the following for every $k = 1, \dots, K$:

$$\big|\langle w_i, v_k \rangle\big| \le \frac{1}{16} \text{ if } k \notin I_i; \qquad \big|\langle w_i, v_k \rangle\big| \ge \frac{3}{16} \text{ if } k \in I_i. \tag{6.3}$$

Define the pseudolinear map $\Phi(v) = \big(\Phi(v)_1, \dots, \Phi(v)_r\big)$ as follows:

$$P(v)_i := \phi\left(\langle w_i, v \rangle - \frac{1}{8}\right), \quad P(v)_{r/2+i} := \phi\left(-\langle w_i, v \rangle - \frac{1}{8}\right), \quad i = 1, \dots, r/2.$$

If $y_k = 0$, then $k > K_1$. Thus k does not belong to any batch I_i, and (6.3) implies that $\big|\langle w_i, v_k \rangle\big| \le 1/16$ for all i. Then both $\langle w_i, v_k \rangle - 1/8$ and $-\langle w_i, v_k \rangle - 1/8$ are negative, and since $\phi(t) = 0$ for negative t, all coordinates of $P(v_k)$ are zero, i.e., $P(v_k) = 0$.

Conversely, if $P(v_k) = 0$, then, by construction, for each i both $\langle w_i, v \rangle - 1/8$ and $\langle w_i, v \rangle - 1/8$ must be nonpositive, which yields $\big|\langle w_i, v_k \rangle\big| \le 1/8 < 3/16$. Thus, by (6.3), k may not belong to any batch I_i, which means that $k > K_1$, and this implies $y_k = 0$. ∎

7. Assembly. In this section we prove a general version of our main result. Let us first show how to train a network with four layers. To this end, choose an enrichment map from layer 1 to layer 2 to transform the data from merely separated to ε-orthogonal, choose a map from layer 2 to layer 3 to further enrich the data by making it $O(1/\sqrt{d})$-orthogonal, and finally make a map from layer 3 to layer 4 memorize the labels. This yields the following result.

Theorem 7.1 (shallow networks). *Consider unit vectors $x_1, \dots, x_K \in \mathbb{R}^n$ that satisfy*

$$\|x_i - x_j\|_2 \ge C_2 \sqrt{\frac{\log \log d}{\log m}}.$$

Consider any labels $y_1, \dots, y_K \in \{0, 1\}$, at most K_1 of which equal 1. Assume that

$$K_1 \log^5(dK) \le c_5 dr$$

as well as $K \le \exp(c_5 m^{1/5})$, $K \le \exp(c_5 d^{1/5})$, and $d \le \exp(c_5 m^{1/5})$. Then there exists a map $F \in \mathcal{F}(n, m, d, r)$ such that for all $k = 1, \dots, K$ we have

$$F(x_k) = 0 \text{ iff } y_k = 0.$$

Proof. *Step* 1. *From separated to ε-orthogonal.* Apply Theorem 5.2 with $\varepsilon = c_5/\log d$. (Note that the required constraints in that result hold by our assumptions with small c_5.) We obtain an almost pseudolinear map $E : \mathbb{R}^n \to \mathbb{R}^m$ such that the vectors $u_k := E(x_k)$ satisfy

$$\big|\|u_i\|_2^2 - 1\big| \le \varepsilon, \quad \big|\langle u_i, u_j\rangle\big| \le \varepsilon$$

for all distinct i, j.

Step 2. *From ε-orthogonal to $\frac{1}{\sqrt{d}}$-orthogonal.* Apply Theorem 5.4. We obtain an almost pseudolinear map $R : \mathbb{R}^m \to \mathbb{R}^d$ such that the vectors $v_k := R(u_k)$ satisfy

$$\|v_i\|_2 \ge \frac{1}{2}, \quad \big|\langle v_i, v_j\rangle\big| \le \frac{C_4 \log^2(dK)}{\sqrt{d}} =: \eta$$

for all distinct indices i, j.

Step 3. *Perception.* Apply Theorem 6.2. (Note that our assumptions with small enough c_5 ensure that the required constraint $(2K_1 + r)\log K \le c_4 r\eta^{-2}$ does hold.) We obtain a pseudolinear map $P : \mathbb{R}^d \to \mathbb{R}^r$ such that the vectors $z_k := P(v_k)$ satisfy

$$z_k = 0 \text{ iff } y_k = 0.$$

Step 4. *Assembly.* Define

$$F := P \circ R \circ E.$$

Since E and R are almost pseudolinear and P is pseudolinear, F can be represented as a composition of three pseudolinear maps (by absorbing the linear factors), i.e., $F \in \mathcal{F}(n, m, d, r)$. Also, $F(x_k) = z_k$ by construction, so the proof is complete. ∎

Finally, we can extend Theorem 7.1 for arbitrarily deep networks by distributing the memorization tasks among all layers evenly. Indeed, consider a network with L layers and with n_i nodes in layer i. As in Theorem 7.1, first we enrich the data, thereby making the input to layer 3 almost orthogonal. Train the map from layer 3 to layer 4 to memorize the labels of the first $O(n_3 n_4)$ data points using Theorem 6.2 (for $d = n_3$, $r = n_4$, and $\eta \asymp 1/\sqrt{d}$). Similarly, train the map from layer 4 to layer 5 to memorize the labels of the next $O(n_4 n_5)$ data points, and so on. This allows us to train the network on the total of $O(n_3 n_4 + n_4 n_5 + \cdots + n_{L-1} n_L) = O(W)$ data points, where W is the number of "deep connections" in the network, i.e., connections that occur from layer 3 and up. This leads us to the main result of this paper.

Theorem 7.2 (deep networks). *Let $n_1, \dots, n_L$ be positive integers, and set $n_0 := \min(n_2, \dots, n_L)$ and $n_\infty := \max(n_2, \dots, n_L)$. Consider unit vectors $x_1, \dots, x_K \in \mathbb{R}^n$ that satisfy*

$$\|x_i - x_j\|_2 \ge C\sqrt{\frac{\log\log n_\infty}{\log n_0}}.$$

Consider any labels $y_1, \dots, y_K \in \{0, 1\}$, at most K_1 of which equal 1. *Assume that the number of deep connections $W := n_3 n_4 + \cdots n_{L-1} n_L$ satisfies*

$$W \ge C K_1 \log^5 K, \tag{7.1}$$

as well as $K \le \exp(cn_0^{1/5})$ *and* $n_\infty \le \exp(cn_0^{1/5})$. *Then there exists a map* $F \in \mathcal{F}(n_1, \dots, n_L)$ *such that for all* $k = 1, \dots, K$ *we have*

$$F(x_k) = 0 \text{ iff } y_k = 0.$$

We stated a simplified version of this result in Theorem 1.1. To see the connection, just take the "OR" of the outputs of all n_L nodes of the last layer.

Proof. Step 1. *Initial reductions.* For networks with $L = 4$ layers, we already proved the result in Theorem 7.1, so we an assume that $L \ge 5$. Moreover, for $K = 1$ the result is trivial, so we can assume that $K \ge 2$. In this case, if we make the constant c in our assumption $2 \le \exp(cn_0^{1/5})$ sufficiently small, we can assume that n_0 (and thus also all n_i and W) are arbitrarily large, i.e., larger than any given absolute constant.

Step 2. *Distributing data to layers.* Without loss of generality, assume that the first K_1 of the labels y_k equal 1 and the rest equal zero, i.e., $y_k = \mathbf{1}_{\{k \le K_1\}}$. Partition the indices of the nonzero labels $\{1, \dots, K_1\}$ into subsets $I_3, \dots, I_{L-1}$ ("batches") so that

$$|I_i| \le \frac{n_i n_{i+1}}{W} K_1 + 1.$$

(This is possible since the numbers $n_i n_{i+1}/W$ sum to one.) For each batch i, define a new set of labels

$$y_{ki} = \mathbf{1}_{\{k \in I_i\}}, \quad k = 1, \dots, K.$$

In other words, y_{ki} are obtained from the original labels y_i by zeroing out the labels outside batch i.

Step 3. *Memorization at each layer.* For each i, apply Theorem 7.1 for the labels y_{ki}, for the number of nonzero labels $|I_i|$ instead of K_1, and for $n = n_1$, $m = n_0/3$, $d = n_i/3$, and $r = n_{i+1}/3$. Thus, if

$$\left(\frac{n_i n_{i+1}}{W} K_1 + 1 \right) \log^5 \left(\frac{n_i K}{3} \right) \le \frac{c_5 n_i n_{i+1}}{9} \tag{7.2}$$

as well as

$$K \le \exp\left(c_5 n_0^{1/5}/3\right), \qquad K \le \exp\left(c_5 n_i^{1/5}/3\right), \qquad n_i \le \exp\left(c_5 n_0^{1/5}/3\right), \tag{7.3}$$

then there exists a map

$$F_i \in \mathcal{F}(n_1, n_0/3, n_i/3, n_{i+1}/3)$$

satisfying the following for all i and k:

$$F_i(x_k) = 0 \text{ iff } y_{ki} = 0. \tag{7.4}$$

Moreover, when we factorize $F_i = P_i \circ R_i \circ E_i$ into three almost pseudolinear maps, then $E_i = E$, the enrichment map from $\mathbb{R}^{n_1}$ into $\mathbb{R}^{n_0/3}$, is trivially independent of i, so

$$F_i = P_i \circ R_i \circ E. \tag{7.5}$$

Our assumptions with sufficiently small c guarantee that the required conditions (7.3) do hold. In order to check (7.2), we will first note a somewhat stronger bound than (7.1) holds; namely, we have

$$3^5 W \geq C K_1 \log^5(n_i K), \quad i = 3, \dots, L-1. \tag{7.6}$$

Indeed, if $W \geq K^2$, then using that $K_1 \leq K \leq W^{1/2}$ and $n_i \leq W$ we get

$$K_1 \log^5(n_i K) \leq W^{1/2} \log^5(W^{3/2}) = \frac{3^5}{2^5} W^{1/2} \log^5 W \leq \frac{3^5}{C} W$$

when W is sufficiently large. If $W \leq K^2$, then using that $n_i \leq W \leq K^2$ we get

$$K_1 \log^5(n_i K) \leq K_1 \log^5(K^3) = 3^5 K_1 \log^5 K \leq \frac{3^5}{C} W,$$

where the last step follows from (7.1). Hence, we verified (7.6) for the entire range of W.

Now, to check (7.2), note that

$$\log^5(n_i K) \leq 2^5 \big(\log^5 n_i + \log^5 K \big) \leq \frac{c_5 n_i}{20} \leq \frac{c_5 n_i n_{i+1}}{20},$$

where we used that n_i is arbitrarily large and the assumption on K with a sufficiently small constant c. Combining this bound with (7.6), we obtain

$$\left(\frac{n_i n_{i+1}}{W} K_1 + 1 \right) \log^5 \left(\frac{n_i K}{3} \right) \leq \left(\frac{3^5}{C} + \frac{c_5}{20} \right) n_i n_{i+1} \leq \frac{c_5 n_i n_{i+1}}{9}$$

if C is sufficiently large. We have checked (7.2).

Step 4. *Stacking.* To complete the proof, it suffices to construct map $F \in \mathcal{F}(n_1, \dots, n_L)$ with the following property:

$$F(x) = 0 \quad \text{iff} \quad F_i(x) = 0 \text{ for all } i = 3, \dots, L-1. \tag{7.7}$$

Indeed, this would imply that $F(x_k) = 0$ happens if and only if $F_i(x_k) = 0$ for all i, which, according to (7.4), is equivalent to $y_{ki} = 0$ for all i. By definition of y_{ki}, this is further equivalent to $k \notin I_i$ for any i, which by construction of I_i is equivalent to $k > K_1$, which is finally equivalent to $y_k = 0$, as claimed.

We construct F by "stacking" the maps $F_i = P_i \circ R_i \circ E$ for $i = 3, \dots, L-1$ as illustrated in Figure 2. To help us stack these maps, we drop some nodes from the original network and first construct

$$F \in \mathcal{F}(n_1', \dots, n_L')$$

with some $n_i' \leq n_i$; we can then extend F trivially to $\mathcal{F}(n_1, \dots, n_L)$. As Figure 2 suggests, we choose $n_1' = n_1$, $n_2' = n_0/3$, $n_3' = n_0/3 + n_3/3$, $n_i' = n_0/3 + 2n_i/3$ for $i = 4, \dots, L-2$ (skip these layers if $L = 5$), $n_{L-1}' = 2n_{L-1}/3$, and $n_L' = n_L/3$. Note that by definition of n_0, we indeed have $n_i' \leq n_i$ for all i.

We made this choice so that the network can realize the maps F_i. As Figure 2 illustrates, the map $F_3 = P_3 \circ R_3 \circ E \in \mathcal{F}(n_1, n_0/3, n_3/3, n_4/3)$ is realized by setting the factor $E : \mathbb{R}^{n_1} \to$

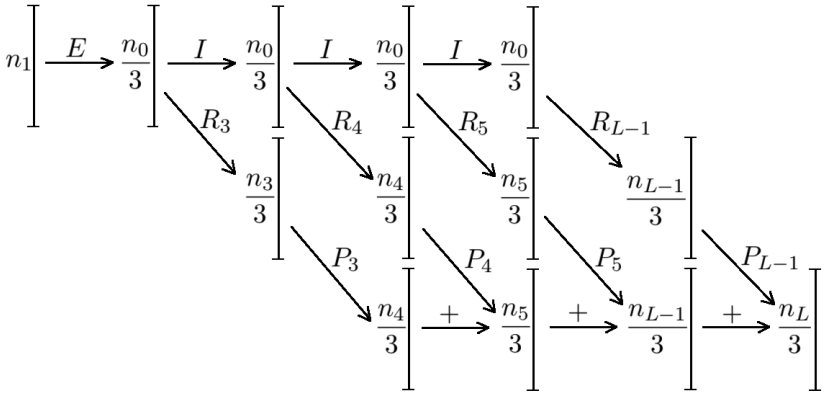

Figure 2. *Trading width for depth: stacking shallow networks into a deep network.*

$\mathbb{R}^{n_0/3}$ to map the first layer to the second, the factor $R_3 : \mathbb{R}^{n_0/3} \to \mathbb{R}^{n_3/3}$ to map the second layer to the last $n_3/3$ nodes of the third layer, and the factor $P_3 : \mathbb{R}^{n_3/3} \to \mathbb{R}^{n_4/3}$ to map further to the last $n_3/3$ nodes of the fourth layer. Moreover, the output of the second layer is transferred to the first $n_0/3$ nodes of the third layer by the identity map[8] I, so we can realize the next map F_4, and so on.

The outputs of all maps F_i are *added together* as the signs "+" in Figure 2 indicate. Namely, the components of the output of F_1, i.e., the last $n_4/3$ nodes of the fourth layer, are summed together and added to any node (say, the last node) of the fifth layer; the components of the output of F_2, i.e., the last $n_5/3$ nodes of the fourth layer, are summed together and added to the last node of the sixth layer, and so on. For ReLU nonlinearity, the + refers to addition of real numbers; for threshold nonlinearity, we replace adding by taking the maximum (i.e., the "OR" operation), which is clearly realizable.

Step 5. *Conclusion.* Due to our construction, the sum of all n'_L components of the function $F(x)$ computed by the network equals the sum (or maximum, for threshold nonlinearity) of all components of all functions $F_i(x)$. Since the components are always nonnegative, $F(x)$ is zero if and only if all components of all functions $F_i(x)$ are zero. In other words, our claim (7.7) holds. ∎

Appendix A. Asymptotical expressions for Gaussian integrals. The asymptotical expansion of the Mills ratio for the normal distribution is well known; see [20]. For our purposes, the first three terms of the expansion will be sufficient:

$$\Psi(a) = \frac{\int_a^\infty e^{-x^2/2}\,dx}{e^{-a^2/2}} = a^{-1} - a^{-3} + 3a^{-5} + O(a^{-7}). \tag{A.1}$$

[8] Note that the identity map restricted to the image of E can be realized as an almost pseudolinear map for both ReLU and threshold nonlinearities. For ReLU this is obvious by setting the bias large enough; for threshold nonlinearity note that the image of the almost pseudolinear map E consists of vectors whose coordinates either are zero or take the same value λ. Thus, the Heaviside function multiplied by λ is the identity on the image of E.

In particular, the tail probability of the standard normal random variable $\gamma \sim N(0,1)$ satisfies

$$\mathbb{P}\left\{\gamma > a\right\} = \frac{1}{\sqrt{2\pi}} e^{-a^2/2}\Big(a^{-1} + O(a^{-3})\Big). \tag{A.2}$$

The following two lemmas give asymptotical expressions for the expected value of the first two moments of the random variable $(\gamma - a)_+ = \max(\gamma - a, 0)$ where, as before, $\gamma \sim N(0,1)$ is standard normal.

Lemma A.1 (ReLU of the normal distribution). *Let $\gamma \sim N(0,1)$. Then, as $a \to \infty$, we have*

$$\mathbb{E}(\gamma - a)_+ = \frac{1}{\sqrt{2\pi}} e^{-a^2/2}\Big(a^{-2} + O(a^{-4})\Big),$$
$$\mathbb{E}((\gamma - a)_+)^2 = \frac{1}{\sqrt{2\pi}} e^{-a^2/2}\Big(2a^{-3} + O(a^{-5})\Big).$$

Proof. Expressing expectation as the integral of the tail (see, e.g., [23, Lemma 1.2.1]), we have

$$\sqrt{2\pi}\ \mathbb{E}(\gamma - a)_+ = \int_0^\infty (x-a)_+ e^{-x^2/2}\, dx = \int_a^\infty (x-a) e^{-x^2/2}\, dx$$
$$= \int_a^\infty x e^{-x^2/2}\, dx - a\int_a^\infty e^{-x^2/2}\, dx.$$

Using substitution $y = x^2/2$, we see that the value of the first integral on the right-hand side is $e^{-a^2/2}$. Using the Mills ratio asymptotics (A.1) for the second integral, we get

$$\sqrt{2\pi}\ \mathbb{E}(\gamma - a)_+ = e^{-a^2/2} - a \cdot e^{-a^2/2}\Big(a^{-1} - a^{-3} + O(a^{-5})\Big) = e^{-a^2/2}\Big(a^{-2} + O(a^{-4})\Big).$$

This finishes the first part of the lemma.

To prove the second part, we start similarly:

$$\sqrt{2\pi}\ \mathbb{E}((\gamma - a)_+)^2 = \int_a^\infty (x-a)^2 e^{-x^2/2}\, dx$$
$$= \int_a^\infty x^2 e^{-x^2/2}\, dx - 2a\int_a^\infty x e^{-x^2/2}\, dx + a^2\int_a^\infty e^{-x^2/2}\, dx.$$

Integrating by parts, we find that the first integral on the right side equals

$$a e^{-a^2/2} + \int_a^\infty e^{-x^2/2}\, dx = a e^{-a^2/2} + \Psi(a) e^{-a^2/2};$$

the second integral equals $e^{-a^2/2}$ as before, and the third integral equals $\Psi(a)e^{-a^2/2}$. Combining these and using the asymptotical expansion (A.1) for $\Psi(a)$, we conclude that

$$\sqrt{2\pi}\ \mathbb{E}((\gamma - a)_+)^2 = a e^{-a^2/2} + \Psi(a) e^{-a^2/2} - 2a e^{-a^2/2} + a^2 \Psi(a) e^{-a^2/2}$$
$$= e^{-a^2/2}\Big((a^2+1)\Psi(a) - a\Big) = e^{-a^2/2}\Big(2a^{-3} + O(a^{-5})\Big).$$

This completes the proof of the second part of the lemma. ∎

Lemma A.2 (stability). *Fix any $z > -1$. Let $\gamma \sim N(0,1)$. Then, as $a \to \infty$, we have*

$$\frac{\mathbb{P}\{\gamma\sqrt{1+z} > a\}}{\mathbb{P}\left\{\gamma > a\right\}} = \exp\left(\frac{a^2 z}{2(1+z)}\right)(1+z)^{1/2}\left(1+O(a^{-2})\right); \tag{A.3}$$

$$\frac{\mathbb{E}(\gamma\sqrt{1+z}-a)_+}{\mathbb{E}(\gamma-a)_+} = \exp\left(\frac{a^2 z}{2(1+z)}\right)(1+z)\left(1+O(a^{-2})\right); \tag{A.4}$$

$$\frac{\mathbb{E}\left((\gamma\sqrt{1+z}-a)_+\right)^2}{\mathbb{E}((\gamma-a)_+)^2} = \exp\left(\frac{a^2 z}{2(1+z)}\right)(1+z)^{3/2}\left(1+O(a^{-2})\right). \tag{A.5}$$

Proof. Use the asymptotics in (A.2) and Lemma A.1 for a and $a/\sqrt{1+z}$ and simplify. ∎

We complete this paper by proving an elementary monotonicity property for Gaussian integrals, which we used in the proof of of Proposition 3.2.

Lemma A.3 (monotonicity). *Let $\psi : \mathbb{R} \to [0,\infty)$ be a nondecreasing function satisfying $\psi(t) = 0$ for all $t < 0$, and let $\gamma \sim N(0,1)$. Then $\sigma \mapsto \mathbb{E}\,\psi(\sigma\gamma)$ is a nondecreasing function on $[0,\infty)$.*

Proof. Denoting by $f(x)$ the probability density function of $N(0,1)$, we have

$$\mathbb{E}\,\psi(\sigma\gamma) = \int_{\infty}^{\infty} \psi(\sigma x) f(x)\,dx = \int_{0}^{\infty} \psi(\sigma x) f(x)\,dx.$$

The last step follows since, by assumption, $\psi(\sigma x) = 0$ for all $x < 0$. To complete the proof, it remains to note that for every fixed $x \geq 0$, the function $\sigma \mapsto \psi(\sigma x)$ is nondecreasing. ∎

REFERENCES

[1] Z. Allen-Zhu, Y. Li, and Z. Song, *A Convergence Theory for Deep Learning via Over-Parameterization*, preprint, https://arxiv.org/abs/1811.03962, 2018.

[2] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang, *Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks*, preprint, https://arxiv.org/abs/1901.08584, 2019.

[3] P. Baldi and R. Vershynin, *On neuronal capacity*, in Proceedings of the Conference on Neural Information Processing Systems, 2018, pp. 7729–7738.

[4] P. Baldi and R. Vershynin, *The capacity of feedforward neural networks*, Neural Networks, 116 (2019), pp. 288–311.

[5] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, *Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks*, J. Mach. Learn. Res., 20 (2019), pp. 1–17.

[6] E. B. Baum, *On the capabilities of multilayer perceptrons*, J. Complexity, 4 (1988), pp. 193–215.

[7] E. B. Baum and D. Haussler, *What size net gives valid generalization?*, in Proceedings of the Conference on Neural Information Processing Systems, 1989, pp. 81–90.

[8] T. M. Cover, *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*, IEEE Trans. Electron. Comput., 3 1965, pp. 326–334.

[9] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, *Gradient Descent Finds Global Minima of Deep Neural Networks*, preprint, https://arxiv.org/abs/1811.03804, 2018.

[10] S. S. Du, X. Zhai, B. Poczos, and A. Singh, *Gradient Descent Provably Optimizes Over-Parameterized Neural Networks*, preprint, https://arxiv.org/abs/1810.02054, 2018.

[11] R. Ge, R. Wang, and H. Zhao, *Mildly Overparametrized Neural Nets Can Memorize Training Data Efficiently*, preprint, https://arxiv.org/abs/1909.11837, 2019.

[12] M. Hardt and T. Ma, *Identity Matters in Deep Learning*, https://arxiv.org/abs/1611.04231, preprint, 2016.
[13] G.-B. Huang, *Learning capability and storage capacity of two-hidden-layer feedforward networks*, IEEE Trans. Neural Netw., 14 (2003), pp. 274–281.
[14] Z. Ji and M. Telgarsky, *Polylogarithmic Width Suffices for Gradient Descent to Achieve Arbitrarily Small Test Error with Shallow ReLu Networks*, preprint, https://arxiv.org/abs/1909.12292, 2019.
[15] A. Kowalczyk, *Estimates of storage capacity of multilayer perceptron with threshold logic hidden units*, Neural Networks, 10 (1997), pp. 1417–1433.
[16] Y. Li and Y. Liang, *Learning overparameterized neural networks via stochastic gradient descent on structured data*, in Proceedings of the Conference on Neural Information Processing Systems, 2018, pp. 8157–8166.
[17] G. J. Mitchison and R. M. Durbin, *Bounds on the learning capacity of some multi-layer networks*, Biol. Cybernet., 60 (1989), pp. 345–365.
[18] S. Oymak and M. Soltanolkotabi, *Towards moderate overparameterization: Global convergence guarantees for training shallow neural networks*, IEEE J. Sel. Areas Inform. Theorey, 1 (2020), pp. 84–105.
[19] A. Panigrahi, A. Shetty, and N. Goyal, *Effect of Activation Functions on the Training of Overparametrized Neural Nets*, preprint, https://arxiv.org/abs/1908.05660, 2019.
[20] I. Pinelis, *Monotonicity properties of the relative error of a Padé approximation for Mills' ratio*, J. Inequal. Pure Appl. Math, 3 (2002), pp. 1–8.
[21] Z. Song and X. Yang, *Quadratic Suffices for Over-Parametrization via Matrix Chernoff Bound*, preprint, https://arxiv.org/abs/1906.03593, 2019.
[22] R. Sun, *Optimization for Deep Learning: Theory and Algorithms*, preprint, https://arxiv.org/abs/1912.08957, 2019.
[23] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, Camb. Ser. Stat. Probab. Math., Cambridge University Press, Cambridge, UK, 2018.
[24] M. Yamasaki, *The lower bound of the capacity for a neural network with multiple hidden layers*, in Proceedings of the International Conference on Artificial Neural Networks, Springer, 1993, pp. 546–549.
[25] C. Yun, S. Sra, and A. Jadbabaie, *Small relu networks are powerful memorizers: A tight analysis of memorization capacity*, in Proceedings of the Conference on Neural Information Processing Systems, 2019, pp. 15532–15543.
[26] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, *Understanding deep learning requires rethinking generalization*, in Proceedings of the International Conference on Learning Representation, Toulon, France, 2017.
[27] D. Zou, Y. Cao, D. Zhou, and Q. Gu, *Gradient descent optimizes over-parameterized deep relu networks*, Mach. Learn., 109 (2020), pp. 467–492.
[28] D. Zou and Q. Gu, *An Improved Analysis of Training Over-Parameterized Deep Neural Networks*, preprint, https://arxiv.org/abs/1906.04688, 2019.