Representation Results & Algorithms for Deep Feedforward Networks

Jacob Abernethy **Alex Kulesza** **Matus Telgarsky**
University of Michigan, Ann Arbor
{jabernet,kulesza,mtelgars}@umich.edu

Abstract

Despite the fact that parameter optimization for deep feedforward neural networks is highly non-convex, generic gradient methods remain the dominant approach in practice. In part, this is because our understanding of the functions represented by such networks (as compared with simpler flat networks) is still quite limited. This note presents a new representation result for deep neural networks, establishing a family of classification problems for which any flat network requires exponentially more nodes than an appropriately designed deep network. It then develops a layer-wise training algorithm that is able to efficiently learn these compact deep networks from labeled data. In general, the algorithm recovers a perfect classifier whenever the data has no noise, and it performs well on benchmark datasets.

1 Overview

A neural network is a function whose evaluation is defined by a directed graph, as follows. Root nodes compute $x \mapsto \sigma(w_0 + \langle w, x\rangle)$, where x is the input to the network and $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear function, for instance the ReLU (Rectified Linear Unit) $\sigma_{\mathrm{R}}(z) = \max\{0, z\}$. Internal nodes perform the same computation, but use the collective output of their parents in place of the raw inputs x. The set of all nodes at a given depth in the graph is called a *layer*. The set of functions obtained by varying all of the w_0 and w parameters (which need not be the same from node to node) over networks with l layers, each with at most m nodes, gives the function class $\mathcal{N}(\sigma; m, l)$.

The representation power of $\mathcal{N}(\sigma; m, l)$ will be measured via the *classification error* $\mathcal{R}_{\mathrm{z}}$. Namely, given a function $f : \mathbb{R}^d \to \mathbb{R}$, let $\tilde{f} : \mathbb{R}^d \to \{0, 1\}$ denote the corresponding classifier $\tilde{f}(x) := \mathbb{1}[f(x) \geq 1/2]$, and additionally given a sequence of points $((x_i, y_i))_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, define $\mathcal{R}_{\mathrm{z}}(f) := n^{-1} \sum_i \mathbb{1}[\tilde{f}(x_i) \neq y_i]$.

A function $f : \mathbb{R} \to \mathbb{R}$ is said to be *t-sawtooth* if there exists a partition of $\mathbb{R}$ into at most t intervals such that f is affine within each; e.g., σ_{R} is 2-sawtooth. The core representation result is as follows.

Theorem 1.1 (Informal representation result)**.** *Given any integer k, there exists a set of $n := 2^k + 1$ points $((x_i, y_i))_{i=1}^n$ so that for any t-sawtooth $\sigma : \mathbb{R} \to \mathbb{R}$, any number of layers l, and any number of nodes per layer m with $mt \leq 2^{(k-2)/l}$,*

$$\min_{f \in \mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k)} \mathcal{R}_{\mathrm{z}}(f) = 0 \qquad \text{and} \qquad \min_{g \in \mathcal{N}(\sigma; m, l)} \mathcal{R}_{\mathrm{z}}(g) \geq \frac{1}{6}.$$

Moreover, the perfect classifier in $\mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k)$ can be taken to be k repetitions of a constant size recurrent network*; namely, a network with 3 nodes in 2 layers composed with itself $k - 1$ times.*

In other words, we can construct data sets for which a network of depth $2k$ gives zero error, but any flat network has error at least $1/6$ unless it contains an exponential number of nodes. This result and its proof will be discussed in Section 2. It mirrors circuit complexity results which state that

the parity function on d bits requires exponential size constant-depth circuits [1], and similar results for *sum-product networks*, which are neural networks with summation and product nodes [2]. The result here for standard feedforward networks is to some extent folklore, however neither proof nor precise statement have ever been provided, the only existing results providing that flat networks of arbitrary size may approximate continuous functions [3].

Of course, exotic functions representable by neural networks are irrelevant if they cannot be tractably found from data, and the training loss for neural networks is highly non-convex. Thus, as a companion to the representation result, Section 3 discusses learning algorithm whose guarantees may be summarized as follows.

Theorem 1.2 (Informal algorithmic guarantee)**.** *There exists a layer-wise algorithm which, given the n points from Theorem 1.1 finds $f \in \mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k+1)$ with $\mathcal{R}_{\mathrm{z}}(f) = 0$. More generally, given any set of points $((x_i, y_i))_{i=1}^n$ where $x_i = x_j$ implies $y_i = y_j$, it learns a network g with $\mathcal{R}_{\mathrm{z}}(g) = 0$.*

The first part of the result guarantees learning a compact network over a restrictive class of functions, while the second shows that the algorithm learns more generally given a sufficiently large network. Algorithms of the second type are known when networks are shallow [4], but the result here concerns deep networks where additional layers cannot access the input variables directly. Even so, this result is stylized, disallowing label noise and guaranteeing general learning only for large networks, thus Section 3 also provides an empirical evaluation. All proofs can be found in the full version[1].

2 Representation results

The upper and lower bound follow from a simple intuition: adding piecewise affine functions together grows the number of "bumps" only linearly, whereas composing them increases the number of bumps multiplicatively.

With this in mind, let positive integer k be given, and consider the binary classification problem at right; henceforth, call this problem k-*ap* (k-alternating-points), and note that it consists of $2^k + 1$ evenly spaced points whose labels alternate.

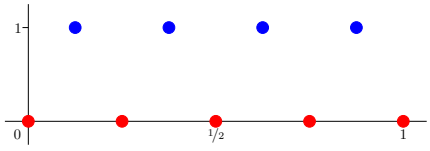

Figure 1: The 3-ap.

First consider the lower bound in Theorem 1.1. As shown in Appendix A.1, given functions $\sigma_1 : \mathbb{R} \to \mathbb{R}$ and $\sigma_2 : \mathbb{R} \to \mathbb{R}$ which are respectively t_1- and t_2-sawtooth, $\sigma_1 + \sigma_2$ is $(t_1 + t_2 - 1)$-sawtooth and $\sigma_1 \circ \sigma_2$ is $t_1 t_2$-sawtooth. By induction over the layers of a neural network, it follows that every element of $\mathcal{N}(\sigma; m, l)$ is $(tm)^l$-sawtooth provided that σ is t-sawtooth.

To complete the lower bound, a counting argument establishes that *any* t'-sawtooth function f must make many errors on the k-ap when $t' < 2^{k-2}$; the counting argument, presented in Appendix A.1, shows that many of the t' intervals defining f receive multiple points, a large fraction of which cannot be classified correctly due to the alternating labels.

Lemma 2.1. *Let $((x_i, y_i))_{i=1}^n$ be given according to the k-ap, with $n := 2^k + 1$. Then every t'-sawtooth function $f : \mathbb{R} \to \mathbb{R}$ satisfies $\mathcal{R}_{\mathrm{z}}(f) \geq (n - 2t')/(3n)$.*

For the upper bound, consider the *mirror map* $f_{\mathrm{m}} : \mathbb{R} \to \mathbb{R}$ depicted in Figure 2, defined as

$$f_{\mathrm{m}}(x) := \begin{cases} 2x & \text{when } 0 \leq x \leq 1/2, \\ 2(1-x) & \text{when } 1/2 < x \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $f_{\mathrm{m}} \in \mathcal{N}(\sigma_{\mathrm{R}}; 2, 2)$; for instance, $f_{\mathrm{m}}(x) = \sigma_{\mathrm{R}}(2\sigma_{\mathrm{R}}(x) - 4\sigma_{\mathrm{R}}(x - 1/2))$. The upper bounds will use $f_{\mathrm{m}}^k \in \mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k)$, where f_{m}^k denotes f_{m} composed with itself $k - 1$ times.

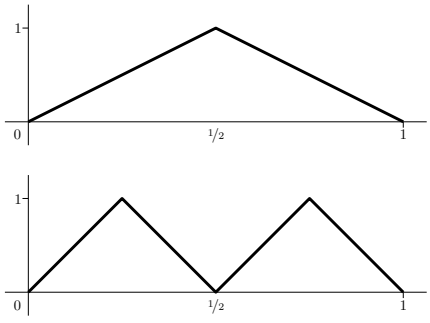

Figure 2: f_{m} and f_{m}^2.

[1] Full version is at `http://cseweb.ucsd.edu/~mtelgars/manuscripts/nc15.pdf`.

As shown in Appendix A.2, each additional composition with f_{m} doubles the frequency and number of peaks, and consequently f_{m}^k precisely weaves through the points of the k-ap, achieving zero classification error.

3 Algorithm

Consider several possible approaches to learning compact deep networks. First, one could try to find succinct representations by minimizing a loss function directly using gradient methods, but there is little hope of this being tractable due to non-convexity. A second option is the popular approach of *unsupervised pre-training*, where a gradient method is initialized using parameters determined from unlabeled data; however, this fails on the k-ap, which, stripped of labels, consists only of uniformly spaced points. A third approach, followed here, is *supervised* pre-training, which initializes parameters in a layer-wise fashion using labeled data.

The supervised pre-training method in Algorithm 1 carries the name GADGETRON since it optimizes small groups of layers at a time, thus searching for *gadgets* (e.g., the 2-layer gadget f_{m} from the previous section). In order to instantiate the GADGETRON, a *gadget class* (connectivity pattern for a collection of layers) must be specified, as well as an objective function Q. It is tempting at first to use classification loss for Q, however this fails even at finding the first mirror map f_{m} given the k-ap: since the optimal zero-one loss is attained by many different translations and scalings of the f_{m}, loss minimization will not generally choose the unaltered f_{m}.

Before describing the choice of Q and its corresponding guarantees, it is useful to sketch desirable properties by studying how f_{m} operates upon the k-ap. As depicted at right, applying f_{m} to the k-ap reflects the points upon themselves, producing a *weighted* $(k-1)$-ap where all new points with $x < 1$ are duplicated (with matching labels). One way to induce this behavior is by trying to reduce the number of *regions* to be labeled 0 and 1. Of course, as stated, this is a nonparametric quantity which is difficult to estimate, but Q_1 and Q_2, described below, give a simple surrogate to this approach.

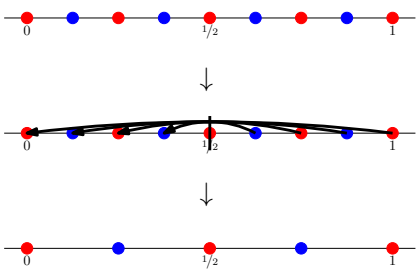

Figure 3: f_{m} applied to the k-ap.

3.1 Guarantees

It is convenient to refine the notation $\mathcal{N}(\sigma; m, l)$ to $\mathcal{N}(\sigma; (m_1, m_2, \ldots, m_l))$, meaning the class of network functions where layer 1 has at most m_1 nodes, layer 2 has at most m_2 nodes, and so on.

To simplify the analysis, the two guarantees use different but related objective functions Q_1 and Q_2. Each is defined over maps of the form $F : S \to [0,1]^{d_2}$, where $S \subseteq \mathbb{R}^{d_1}$ is the data; for instance, Q can be applied to elements of $\{f \in \mathcal{N}(\sigma; (d_1, d_2)) : F(x) \in [0,1]^{d_2} \text{ when } x \in S\}$. Both objective functions will be stated only in terms of separating pairs of examples with differing labels, thus the restriction of the image of the maps to $[0,1]^{d_2}$ is essential to prevent blow-up.

Algorithm 1 Gadgetron.

input Objective function Q, gadget classes $(\mathcal{G}_j)_{j=1}^l$, data S.

1: Initial identity map $F_0(x) = x$.
2: **for** $j = 1, 2, \ldots, l$: **do**
3: Choose gadget $L_j \in \mathcal{G}_j$ by minimization of Q on mapped data $F_{j-1}(S)$:

$$Q(L_j \circ F_{j-1}; S) = \min_{L \in \mathcal{G}_j} Q(L \circ F_{j-1}; S).$$

4: Update mapping: $F_j := L_j \circ F_{j-1}$.
5: **end for**
6: **return** Final mapping F_l.

The first guarantee, showing that GADGETRON can fit the k-ap with $\mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k+1)$, uses

$$Q_1(F;S) := \max_{\substack{(x,y)\in S \\ (x',y')\in S \\ y\neq y'}} \frac{1}{\|F(x)-F(x')\|_1}.$$

Notice that Q_1 tries to place pockets of points with differing labels as far apart as possible, and moreover to collapse points of the same label, since this gives more room to spread out the differently-labeled points. Indeed, the optimum (over all functions) is to place all positive points in one corner, and all negative points in the opposite corner. (Note that this objective is always ∞ if there exist (x,y) and (x',y') with $x=x'$ and $y=y'$; the assumptions rule this case out, but the experiments will circumvent this by adding a tiny positive constant to the denominator.) Using Q_1 gives gives the first part of Theorem 1.2, stated in more detail as follows.

Lemma 3.1. *Let positive integer k be given. Suppose the* GADGETRON *is run with gadget class $\mathcal{N}(\sigma_{\mathrm{R}}; (2,1))$, objective function Q_1, and data S matching the k-ap. Then after k epochs, a function $f \in \mathcal{N}(\sigma_{\mathrm{R}}; 2, 2k)$ is output with either $\mathcal{R}_{\mathrm{z}}(f) = 0$ or $\mathcal{R}_{\mathrm{z}}(1-f) = 0$.*

In order to prove the second part of Theorem 1.2, a slightly more relaxed objective is used:

$$Q_2(F;S) := \frac{1}{|S|} \sum_{(x,y)\in S} \max_{\substack{(x',y')\in S \\ y\neq y'}} \frac{1}{\|F(x)-F(x')\|_1}.$$

The advantage of Q_2 over Q_1 is that moving around any points can improve the cost, not just those attaining the minimum in Q_1. This suffices to prove the second part of Theorem 1.2 by showing that in each epoch, GADGETRON always can choose to construct one of two gadgets which will guarantee a sufficient decrease in cost. These gadgets are complicated geometric objects, which imposes the unpleasant size bound $\mathcal{O}((4d)^d)$.

Lemma 3.2. *Suppose the* GADGETRON *is run with gadget class $\mathcal{N}(\sigma_{\mathrm{R}}; \mathcal{O}((4d)^d), 4)$, objective function Q_2, and any set of points S with $y = y'$ whenever $x = x'$. Then the final mapping f output after $\ln(dQ_2(F_0;S))/\ln(2)$ epochs provides a linearly separable set of points.*

3.2 Experiments

The full experimental setup is described in Appendix C. Networks were trained with a standard gradient method, and three initializations were considered: a standard random initialization [5, Section 4.6, "Initializing the Weights"], a standard unsupervised initialization [6], and GADGETRON. Each method was asked to produce 3-layer and 8-layer networks on each of 8 standard datasets; this process was repeated 5 times, and the approach with the best validation error minus the error of a baseline linear model is reported in Figure 4.

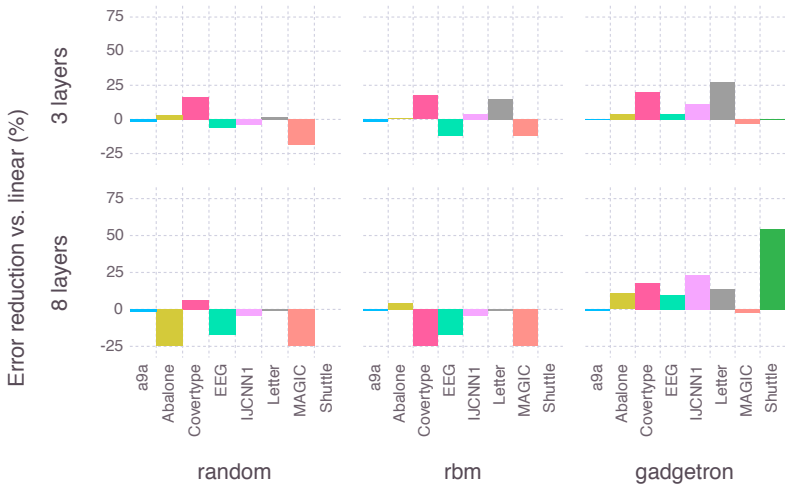

Figure 4: Error reduction versus a linear model for various networks, datasets, and initializations.

References

[1] Johan Håstad. *Computational Limitations of Small Depth Circuits*. PhD thesis, Massachusetts Institute of Technology, 1986.

[2] Yoshua Bengio and Olivier Delalleau. Shallow vs. deep sum-product networks. In *NIPS*, 2011.

[3] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[4] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

[5] Yann Le Cun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag, 1998. URL `http://leon.bottou.org/papers/lecun-98x`.

[6] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[7] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In Léon Bottou, Olivier Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007.

[8] Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. A simple way to initialize recurrent networks of rectified linear units. 2015. `arXiv:1504.00941 [cs.NE]`.

[9] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. In *ICLR, workshop track*, 2015.

[10] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[11] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? 11:625–660, feb 2010.

A Proof of Theorem 1.1

This section will first establish lower and upper bounds in some generality, whereby Theorem 1.1 will follow after some algebra.

A.1 Proof of lower bound

As stated in the body, the lower bound first shows that $\mathcal{N}(\sigma; m, l)$ is $(tm)^l$-sawtooth whenever σ is t-sawtooth, and thereafter completes the proof via a counting argument, reasoning that sawtooth functions can not do well on the k-ap.

In order to prove the sawtooth property of $\mathcal{N}(\sigma; m, l)$, first note how sawtooths grow in complexity when summed or composed.

Lemma A.1. *Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be respectively k- and l-sawtooth. Then $f + g$ is $(k + l - 1)$-sawtooth, and $f \circ g$ is kl-sawtooth.*

Proof. Let $\mathcal{I}_f$ denote the partition of $\mathbb{R}$ corresponding to f, and $\mathcal{I}_g$ denote the partition of $\mathbb{R}$ corresponding to g.

First consider $f + g$, and let $L_f \in \mathcal{I}_f$ and $L_g \in \mathcal{I}_g$ respectively denote the leftmost intervals in the definition of f and g; of course, $f + g$ has a single slope along $L_f \cap L_g$. Thereafter, the slope of $f + g$ can only change when it changes in either f or g. There are $(k - 1) + (l - 1)$ such changes, thus combined with the initial interval $L_f \cap L_g$, $f + g$ is $(k + l - 1)$-sawtooth.

Now consider $f \circ g$, and in particular consider the image $f(g(U_g))$ for some interval $U_g \in \mathcal{I}_g$. g is affine with a single slope along U_g, therefore f is being considered along a single unbroken interval $g(U_g)$. However, nothing prevents $g(U_g)$ from hitting all the elements of $\mathcal{I}_f$; since this argument holds for each $U_g \in \mathcal{I}_g$, thus $f \circ g$ is $(|\mathcal{I}_f| \cdot |\mathcal{I}_g|)$-sawtooth. □

The sawtooth property of $\mathcal{N}(\sigma; m, l)$, now follows by induction.

Lemma A.2. *If σ is t-sawtooth, then every $f \in \mathcal{N}(\sigma; m, l)$ with $f : \mathbb{R} \to \mathbb{R}$ is $(tm)^l$-sawtooth.*

Proof. The proof proceeds by induction over layers, showing the output of each node in layer i is $(tm)^i$-sawtooth as a function of the neural network input. For the first layer, each node starts by computing $x \mapsto w_0 + \langle w, x \rangle$, which is itself affine and thus 1-sawtooth, so the full node computation $x \mapsto \sigma(w_0 + \langle w, x \rangle)$ is t-sawtooth by Lemma A.1. Thereafter, the input to layer i with $i > 1$ is a collection of functions $(g_1, \ldots, g_{m'})$ with $m' \leq m$ and g_j being $(tm)^{i-1}$-sawtooth by the inductive hypothesis; consequently, $x \mapsto w_0 + \sum_j w_j g_j(x)$ is $m(tm)^{i-1}$-sawtooth by Lemma A.1, whereby applying σ yields a $(tm)^i$-sawtooth function (once again by Lemma A.1). □

The lower bound now follows via a counting argument, the statement appearing as Lemma 2.1 in the body.

Proof of Lemma 2.1. Recall the notation $\tilde{f}(x) := \mathbb{1}[f(x) \geq 1/2]$, whereby $\mathcal{R}_z(f) := n^{-1} \sum_i \mathbb{1}[y_i \neq \tilde{f}(x_i)]$. Since f is piecewise monotonic with a corresponding partition $\mathbb{R}$ having at most t pieces, then f has at most $2t - 1$ crossings of 1/2: at most one within each interval of the partition, and at most 1 at the right endpoint of all but the last interval. Consequently, $\tilde{f}$ is piecewise *constant*, where the corresponding partition of $\mathbb{R}$ is into at most $2t$ intervals. This means n points with alternating labels must land in $2t$ buckets, thus the total number of points landing in buckets with at least three points is at least $n - 4t$. Since buckets are intervals and signs must alternate within any such interval, at least a third of the points in any of these buckets are labeled incorrectly by $\tilde{f}$. □

A.2 Proof of upper bound

To assess the effect of the *post-composition* $f_m \circ g$ for any $g : \mathbb{R} \to \mathbb{R}$, note that $f_m \circ g$ is $2g(x)$ whenever $g(x) \in [0, 1/2]$, and $2(1 - g(x))$ whenever $g(x) \in (1/2, 1]$. Visually, this has the effect

of reflecting (or folding) the graph of g around the horizontal line through 1/2 and then rescaling by 2. Applying this reasoning to f_{m}^k leads to f_{m}^2 and f_{m}^3 in Figure 2, whose peaks and troughs match the 2^2-ap and 2^3-ap, and moreover have the form of a piecewise affine approximations to sinusoids; indeed, it was suggested before, by Bengio and LeCun [7], that Fourier transforms are efficiently represented with deep networks.

These compositions may be written as follows.

Lemma A.3. *Let real $x \in [0,1]$ and positive integer k be given, and choose the unique nonnegative integer $i_k \in \{0, \dots, 2^{k-1}\}$ and real $x_k \in [0,1)$ so that $x = (i_k + x_k)2^{1-k}$. Then*

$$f_{\mathrm{m}}^k(x) = \begin{cases} 2x_k & \text{when } 0 \le x_k \le 1/2, \\ 2(1-x_k) & \text{when } 1/2 < x_k < 1. \end{cases}$$

In order to prove this form and develop a better understanding of f_{m}, consider its *pre-composition* behavior $g \circ f_{\mathrm{m}}$ for any $g : \mathbb{R} \to \mathbb{R}$. Now, $(g \circ f_{\mathrm{m}})(x) = g(2x)$ whenever $x \in [0, 1/2]$, but $(g \circ f_{\mathrm{m}})(x) = g(2-2x)$ when $x \in (1/2, 1]$; whereas post-composition reflects around the horizontal line at 1/2 and then scales vertically by 2, pre-composition first scales horizontally by $1/2$ and then reflects around the vertical line at 1/2, providing a condensed mirror image and motivating the name *mirror map*.

Proof of Lemma A.3. The proof proceeds by induction on the number of compositions l. When $l = 1$, there is nothing to show. For the inductive step, the mirroring property of pre-composition with f_{m} combined with the symmetry of f_{m}^l (by the inductive hypothesis) implies that every $x \in [0, 1/2]$ satisfies

$$(f_{\mathrm{m}}^l \circ f)(x) = (f_{\mathrm{m}}^l \circ f)(1-x) = (f_{\mathrm{m}}^l \circ f)(x + 1/2).$$

Consequently, it suffices to consider $x \in [0, 1/2]$, which by the mirroring property means $(f_{\mathrm{m}}^l \circ f_{\mathrm{m}})(x) = f_{\mathrm{m}}^l(2x)$. Since the unique nonnegative integer i_{l+1} and real $x_{l+1} \in [0,1)$ satisfy $2x = 2(i_{l+1} + x_{l+1})2^{-l-1} = (i_{l+1} + x_{l+1})2^{-l}$, the inductive hypothesis applied to $2x$ grants

$$(f_{\mathrm{m}}^l \circ f)(x) = f_{\mathrm{m}}^l(2x) = \begin{cases} 2x_{l+1} & \text{when } 0 \le x_{l+1} \le 1/2, \\ 2(1-x_{l+1}) & \text{when } 1/2 < x_{l+1} < 1, \end{cases}$$

which completes the proof. □

A.3 Proof of Theorem 1.1

With both the lower and upper bounds in place, the proof of Theorem 1.1 is immediate.

Proof of Theorem 1.1. Lemma A.3 gives the desired upper bound; it only remains to massage the lower bound.

Fix any $f \in \mathcal{N}(\sigma; m, l)$. By Lemma A.2, it is $(tm)^l$-sawtooth; thus combining the condition $mt \le 2^{(k-2)/l}$ with Lemma 2.1 gives

$$\frac{(2^k+1) - 2(tm)^l}{3(2^k+1)} \ge \frac{1}{3} - (tm)^l 2^{-k} \left(\frac{2}{3}\right) \ge \frac{1}{3} - 2^{k-2}2^{-k}\left(\frac{2}{3}\right) = \frac{1}{3} - \frac{1}{6}.$$

□

B Proof of Theorem 1.2

Theorem 1.2 is proved by first establishing the guarantee for the k-ap, and then separately for datasets S with at most one label for each distinct input point.

B.1 Fitting the k-ap

Proof sketch of Lemma 3.1. This proof will establish by induction on the number of gadget levels that the mapping F_j produced after composing j gadgets maps S, the k-ap, to a positively reweighted

$(k-j)$-ap, possibly with flipped labels. Consequently, after k gadgets, there are just two reweighted points, so either F_k or its negation give a perfect classification. (The proof will *not* show that f_{m}^k is recovered; indeed, each gadget will be equivalent along $[0,1]$ to either f_{m} or $1-f_{\mathrm{m}}$.)

Proceeding with the proof, there is nothing to show in the base case F_0, thus consider F_j with $j \geq 1$. This inductive step will first establish that the gadget class is a subset of the class of functions which is piecewise monotonic in at most two pieces, and from there show that f_{m} and $1-f_{\mathrm{m}}$ are the only optima.

To establish the structural property on the gadget class $\mathcal{N}(\sigma_{\mathrm{R}};(2,1))$, it will be necessary to reason about sawtooth functions in a more refined way than in Lemma A.1. First note that $z \mapsto \sigma_{\mathrm{R}}(w_0+wz)$ is a translation and scaling of σ_{R}, meaning it is continuous, and either a constant function or 2-sawtooth with slope 0 in one piece. Next note that a linear combination of any two such functions is either monotonic, or it is piecewise monotonic in 2 pieces (meaning there exists some $z_0 \in \mathbb{R}$ such that it is monotonic to the left of z_0, and monotonic to the right of z_0). The case that either function is constant is immediate, thus suppose neither is constant. First consider the case that each function has a sequence of slopes $(0,a)$ and $(0,b)$; thus any linear combination has slope sequence $(0,c,d)$ for some numbers c and d, which is thus piecewise monotone in at most 2 pieces. Without loss of generality, the only remaining case is that the slope sequences are $(0,a)$ and $(b,0)$, the other cases being symmetric. Then given linear combination weights (w_1,w_2), the slope sequence of the resulting 3-sawtooth function is either $(w_2b, 0, w_1a)$ or (w_2b, w_2b+w_1a, w_1a). The first case is immediately piecewise monotonic in at most 2 pieces. In the second case, if w_1b and w_2a have the same sign, thus w_1b+w_2a shares this sign and the function is simply monotonic; otherwise their signs differ, but then w_1a+w_2b matches one of these signs, so the function is monotonic in at most 2 pieces.

Now consider $\mathcal{N}(\sigma_{\mathrm{R}};(2,1))$. Since this is a σ_{R} applied to a function which is piecewise monotonic in at most two pieces, the output is again piecewise monotonic in at most two pieces, since the σ_{R} simply replaces anything below 0 with 0, which leaves monotonicity properties unchanged.

Finally consider optimizing Q_1 over all functions $F : \mathbb{R} \to [0,1]$ which are piecewise monotonic in at most two pieces, where by induction the set of points is the j'-ap with $j' = k-j$, perhaps positively reweighted and with flipped signs; In particular, the distance between any pair of points is $2^{-j'}$. Now first consider the case of a monotonic function. Then if any pair of consecutive points are mapped more than $2^{-j'}$ apart, then some other consecutive points are mapped less than $2^{-j'}$ apart; but the identity mapping is feasible and achieves exactly $2^{-j'}$ error. Now consider the case of a function which is piecewise monotonic in at most two pieces, and first consider the piece occupying more of $[0,1]$, breaking ties arbitrarily. If m is the number of points in this region, then the preceding reasoning grants that the distance between them is at least $1/(m-1)$. Meanwhile, if the shorter segment does anything other than mapping points onto the image of the other segment, then the objective function only worsens. Consequently, the optimal objective is obtained by giving one segment $2^{j'-1}+1$ distinct points and the other $2^{j'-1}$ distinct points, giving objective value $2^{-j'+1}$. As the only 3-sawtooth mappings with this structure are f_{m} and $1-f_{\mathrm{m}}$ and moreover since this objective is strictly smaller than the 1-monotonic case, it follows that either f_{m} or $1-f_{\mathrm{m}}$ are chosen. □

B.2 Fitting data with no label noise

Proof sketch of Lemma 3.2. For convenience, set $n := |S|$, $\tau := (1+1/(n(4d-1)))/d$, and define

$$\phi(F;S;x,y) := \min_{\substack{(x',y')\in S \\ y\neq y'}} \frac{1}{\|F(x)-F(x')\|_1},$$

whereby $Q_2(F;S) = n^{-1}\sum_{(x,y)\in S}\phi(F;S;x,y)$.

First note that $\phi(F;S;x,y) \geq 1/d$ for any mapping F, since points are constrained to fall within $[0,1]^d$, where the largest internal l_1 distance is d, between two corners. As a consequence of this, any mapping $F : \mathbb{R}^d \to [0,1]^d$ with $Q_2(F;S) \leq \tau$ must linearly separate S. To see this, first note it

(combined with $\phi \geq 1/d$) implies $\phi(F; S; x, y) \leq 1/(d-1/4)$ for every $(x, y) \in S$, since otherwise

$$Q_2(F; S) = \frac{1}{n} \sum_{(x,y)\in S} \phi(F; S; x, y) > \frac{1}{nd}\left((n-1) + \frac{d}{d-1/4}\right)$$
$$= \frac{1}{nd}\left((n-1) + \frac{d-1/4+1/4}{d-1/4}\right) = \frac{1}{d}\left(1 + \frac{1}{n(4d-1)}\right).$$

Now fix a pair $(x, y) \in S$, and let $(x', y') \in S$ be any pair with $y \neq y'$, whereby $\|F(x) - F(x')\|_1 \geq d - 1/4$, Consequently, there is some corner $a \in \{0, 1\}^d$ of the hypercube with $\|F(x) - a\|_1 \leq 1/4$, and an opposite corner $b \in \{0, 1\}^d$ (meaning $a + b = (1, 1, \ldots, 1)$) with $\|F(x') - b\|_1 \leq 1/4$. Since $(x', y') \in S$ was arbitrary with $y \neq y'$, it holds that all points with label y' reside within $1/4$ of b, and symmetrically all points with label y reside within $1/4$ of a. In other words, the l_1 balls of radius $d/4$ centered at a and b respectively contain all points with labels y and $y' \neq y$ and are necessarily non-intersecting since at opposite corners but with combined radii less than d (the l_1 distance from corner to corner), thus these balls can be separated by a hyperplane.

Consequently, it suffices to show the error drops below τ. To prove this, it will be shown that $Q(F_{j+1}; S) \leq Q(F_j; S)/2$, whereby it follows that $\ln(Q(F_0; S)/\tau)/\ln(2)$ levels suffice to produce a mapping which linearly separates S.

To this end, first note that no round maps two points of differing labels on top of each other, as this would produce a cost of ∞. Now fix any round $j \geq 1$, meaning there is some mapping $F := F_{j-1}$ from the previous round with the property that $x = x'$ whenever $y = y'$. If $Q_2(F; S) \leq 1/(n(d - 1/4))$, the proof is done; otherwise let $(\bar{x}, \bar{y}) \in S$ be a point with maximal $\phi(F; S; \bar{x}, \bar{y})$, and set $\bar{\phi} := \phi(F_{k-1}; S; \bar{x}, \bar{y})$, whereby $\bar{\phi} \geq Q_2(F; S) \geq 1/(n(d - 1/4))$. There are now two cases to consider.

- Suppose $\bar{\phi} - \min_{(x,y)\in S} \phi(F; S; x, y) \geq Q_2(F; S)/2$, and let $(x', y') \in S$ be a pair attaining the minimum, with $\phi' := \phi(F; S; x', y')$, whereby

  $$\phi' \leq \bar{\phi} - Q_2(F; S)/2.$$

  Since $\bar{\phi}$ is attained both at $(\bar{x}, \bar{y})$ and at some $(x', 1 - \bar{y}) \in S$, without loss of generality $\bar{y} = y'$.

  Now consider the effect of a gadget g which maps $\bar{x}$ to x, and is an identity mapping for all other points. Since points with disagreeing labels are guaranteed to be distinct, such a map can be construct with $\mathcal{O}(d)$ nodes in 3 layers. As such,

  $$Q_2(F_j; S) \leq Q_2(g \circ F; S) \leq Q_2(F; S) - \bar{\phi} + \phi' \leq Q_2(F; S)/2.$$

- Now suppose the preceding case does not hold, which means every $(x, y) \in S$ satisfies

  $$\phi(F; S; x, y) \leq \bar{\phi} < Q_2(F; S)/2 + \min_{(x,y)\in S} \phi(F; S; x, y) \leq 3\bar{\phi}/2.$$

  Expanding the definition of ϕ, this means that every point $(x', y') \in S$ with $y' \neq y$ satisfies $\|F(x) - F(x')\|_1 \geq 2/(3\bar{\phi})$. Since x and x' were arbitrary points of differing labels, this inequality holds in general for any two points with differing labels.

  Now fix $(x'', 1-\bar{y}) \in S$ to denote any example which attains the minimum in the definition of ϕ at point $(\bar{x}, \bar{y})$, whereby $\|F_j(\bar{x}) - F_j(x'')\|_1 = 1/\bar{\phi}$. Set $z := (x + x'')/2$, and consider the l_1 ball B of radius $3d/\bar{\phi}$ centered at z. Since all distances between points of differing labels are at least $3/(2\bar{\phi})$, and due to the doubling dimension of l_1 balls, we can cover this larger ball with $\mathcal{O}((2d)^d)$ balls of radius at most $3/(2\bar{\phi})$ such that each ball contains points with only a single label. Since an indicator for such an l_1 ball can be constructed within $\mathcal{N}(\sigma_{\text{R}}; \mathcal{O}(d), 2)$, the mapping which takes all these purely-labeled balls and maps them to two distinct purely labeled points is therefore within the gadget class $\mathcal{N}(\sigma; \mathcal{O}(d(4d)^d), 4)$.

  There are now two cases to consider for these mapped points. The first case is that there exists a diagonal line (parallel to a line connecting two corners of $[0, 1]^d$) fully contained within $B \cap [0, 1]^d$ and having l_1 length at least $6/\bar{\phi}$; Then this line may be cut into three pieces of length $2/\bar{\phi}$ and the mapped points placed at the ends of the central segment

while still being at least $2/\bar{\phi}$ away from any other points not being mapped, meaning the new error, after applying this mapping g (which is the identity map for points outside B), satisfies

$$Q(F_j; S) \leq Q(g \circ F; S) \leq Q(F; S) - 2\bar{\phi} + 2(\bar{\phi}/2) \leq Q(F; S)/2.$$

Now consider the other case, that the l_1 ball of radius $3d/\bar{\phi}$ centered at z, when intersected with $[0, 1]^d$, does not contain such a diagonal line of length $6/\bar{\phi}$. But since $\|F(\bar{x}) - F(x')\|_1 = 1/\bar{\phi}$ and B has radius $3d/\bar{\phi}$, it follows that there is no diagonal of length $6/\bar{\phi}$ only if B contains the entire cube $[0, 1]^d$, meaning the above mapping g will map all of S to two points in the corners and attain the optimal error $1/d$.

□

B.3 Proof of Theorem 1.2

The results in Theorem 1.2 follow by Lemma 3.1 and Lemma 3.2, adding a single layer with a final linear separator.

C Experimental setup

This section will specify the experimental setup in more detail.

First, the objective function was slightly different from Q_1 and Q_2 from the body; namely, it was

$$Q_3(F) := \sum_{\substack{(x,y)\\(x',y')\\y \neq y'}} \frac{1}{\|F(x) - F(x')\|_2^2 + \epsilon},$$

where $\epsilon = 10^{-6}$ was chosen without any tuning, and a barrier term was added to enforce $Q_3(S) \subseteq [0, 1]^{d_2}$.

Next, the body of the paper did not specify the network layouts. Here again, a simple rule was chosen: given an input of size d, the first hidden layer is of size $d/2$, and all further hidden layers have size $d/4$, the final (output) layer being a single node since all problems had univariate labels.

All datasets were scaled to lie within $[0, 1]^d$. Some more detail on the datasets and their properties may be found in Table 1.

Any implementation of GADGETRON must somehow search over a gadget class. As this is a non-convex problem, the approach here was to try a few random restarts of a gradient descent variant (AdaGrad), with mini-batches of size 64 to speed up training. The random restarts themselves were small perturbations of an identity map, an idea which has been used elsewhere in the neural network literature [8, 9].

For each (data set, algorithm, depth) triple, the algorithm was invoked five times, AdaGrad was applied with a several different step sizes to tune the weights of the network via logistic regression

Dataset	n	s	d
a9a	48841	13.8676	123
Abalone	4176	8	8
Covertype	581011	11.8789	54
EEG	14980	13.9901	14
IJCNN1	24995	13	22
Letter	20000	15.5807	16
MAGIC	19020	9.98728	10
Shuttle	43500	7.04984	9

Table 1: Basic statistics on the evaluation datasets. n is the number of examples, s is the average number of nonzero features, and d is the total input dimension.

(4 passes for the small datasets and 1 for the larger), and the progressive validation error was used to select a best model amongst all these initialization and step size choices. Finally the classification error on the testing set was reported for this selected model.

No regularization was used, partially in order to address concerns that a supervised method may be more prone to overfitting [10], in contrast to unsupervised pre-training which has been argued to provide regularization [11]. Of course, as the datasets here are low-dimensional, a broader investigation of the need for regularization is necessary.