Representation Benefits of Deep Feedforward Networks

Matus Telgarsky

Abstract

This note provides a family of classification problems, indexed by a positive integer k, where all shallow networks with fewer than exponentially (in k) many nodes exhibit error at least $1/6$, whereas a deep network with 2 nodes in each of $2k$ layers achieves zero error, as does a recurrent network with 3 distinct nodes iterated k times. The proof is elementary, and the networks are standard feedforward networks with ReLU (Rectified Linear Unit) nonlinearities.

1 Overview

A *neural network* is a function whose evaluation is defined by a 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 typically a nonlinear function, for instance the ReLU (Rectified Linear Unit) $\sigma_{\mathrm{R}}(z) = \max\{0, z\}$. Internal nodes perform a similar computation, but now their input vector is the collective output of their parents. The choices of w_0 and w may vary from node to node, and the possible set of functions obtained by varying these parameters gives the function class $\mathfrak{N}(\sigma; m, l)$, which has l layers each with at most m nodes.

The representation power of $\mathfrak{N}(\sigma; m, l)$ 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 $\mathcal{R}_{\mathrm{z}}(f) := n^{-1} \sum_i \mathbb{1}[\tilde{f}(x_i) \neq y_i]$.

Theorem 1.1. *Let positive integer k, number of layers l, and number of nodes per layer m be given with $m \leq 2^{(k-3)/l-1}$. Then there exists a collection of $n := 2^k$ points $((x_i, y_i))_{i=1}^n$ with $x_i \in [0, 1]$ and $y \in \{0, 1\}$ such that*

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

For example, approaching the error of the $2k$-layer network (which has $\mathcal{O}(k)$ nodes and weights) with 2 layers requires at least $2^{(k-3)/2-1}$ nodes, and with $\sqrt{k-3}$ layers needs at least $2^{\sqrt{k-3}-1}$ nodes.

The purpose of this note is to provide an elementary proof of Theorem 1.1 and its refinement Theorem 1.2, which amongst other improvements will use a *recurrent neural network* in the upper bound. Section 2 will present the proof, and Section 3 will tie these results to the literature on neural network expressive power and circuit complexity, which by contrast makes use of product nodes rather than standard feedforward networks when showing the benefits of depth.

1.1 Refined bounds

There are three refinements to make: the classification problem will be specified, the perfect network will be an even simpler *recurrent* network, and σ need not be σ_{R}.

Let n-ap (the *n-alternating-point* problem) denote the set of n uniformly spaced points within $[0, 1 - 2^{-n}]$ with alternating labels, as depicted in Figure 1; that is, the points $((x_i, y_i))_{i=1}^n$ with $x_i = i2^{-n}$, and $y_i = 0$ when i is even, and

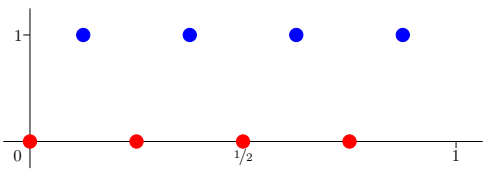

Figure 1: The 3-ap.

otherwise $y_i = 1$. As the x values pass from left to right, the labels change as often as possible; the key is that adding a constant number of nodes in a flat network only corrects predictions on a constant number of points, whereas adding a constant number of nodes in a deep network can correct predictions on a constant *fraction* of the points.

Let $\mathfrak{R}(\sigma; m, l; k)$ denote k iterations of a *recurrent network* with l layers of at most m nodes each, defined as follows. Every $f \in \mathfrak{R}(\sigma; m, l; k)$ consists of some fixed network $g \in \mathfrak{N}(\sigma; m, l)$ applied k times:

$$f(x) = g^k(x) = \big(\underbrace{g \circ \cdots \circ g}_{k \text{ times}}\big)(x).$$

Consequently, $\mathfrak{R}(\sigma; m, l; k) \subseteq \mathfrak{N}(\sigma; m, lk)$, but the former has $\mathcal{O}(ml)$ parameters whereas the latter has $\mathcal{O}(mlk)$ parameters.

Lastly, say that $\sigma : \mathbb{R} \to \mathbb{R}$ is *t-sawtooth* if it is piecewise affine with t pieces, meaning $\mathbb{R}$ is partitioned into t consecutive intervals, and σ is affine within each interval. Consequently, σ_{R} is 2-sawtooth, but this class also includes many other functions, for instance the decision stumps used in boosting are 2-sawtooth, and decision trees with $t - 1$ nodes correspond to t-sawtooths.

Theorem 1.2. *Let positive integer k, number of layers l, and number of nodes per layer m be given. Given a t-sawtooth $\sigma : \mathbb{R} \to \mathbb{R}$ and $n := 2^k$ points as specified by the n-ap, then*

$$\min_{f \in \mathfrak{R}(\sigma_{\mathrm{R}}; 2, 2; k)} \mathcal{R}_{\mathrm{z}}(f) = 0 \qquad \text{and} \qquad \min_{g \in \mathfrak{N}(\sigma; m, l)} \mathcal{R}_{\mathrm{z}}(g) \geq \frac{n - 4(tm)^l}{3n}.$$

This more refined result can thus say, for example, that on the 2^k-ap one needs exponentially (in k) many parameters when boosting decision stumps, linearly many parameters with a deep network, and constantly many parameters with a recurrent network.

2 Analysis

This section will first prove the lower bound via a counting argument, simply tracking the number of times a function within $\mathfrak{N}(\sigma; m, l)$ can cross $1/2$. The upper bound will exhibit a network in $\mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2)$ which can be composed with itself k times to exactly fit the n-ap. These bounds together prove Theorem 1.2, which in turn implies Theorem 1.1.

2.1 Lower bound

The lower bound is proved in two stages. First, composing and summing sawtooth functions must also yield a sawtooth function, thus elements of $\mathfrak{N}(\sigma; m, l)$ are sawtooth whenever σ is. Secondly, a sawtooth function can not cross $1/2$ very often, meaning it can't hope to match the quickly changing labels of the n-ap.

To start, $\mathfrak{N}(\sigma; m, l)$ is sawtooth as follows.

Lemma 2.1. *If σ is t-sawtooth, then every $f \in \mathfrak{N}(\sigma; m, l)$ with $f : \mathbb{R} \to \mathbb{R}$ is $(tm)^l$-sawtooth.*

The proof is straightforward and deferred momentarily. The key observation is that adding together sawtooth functions grows the number of regions very slowly, whereas composition grows the number very quickly, an early sign of the benefits of depth.

Given a sawtooth function, its classification error on the n-ap may be lower bounded as follows.

Lemma 2.2. *Let $((x_i, y_i))_{i=1}^n$ be given according to the n-ap. Then every t-sawtooth function $f : \mathbb{R} \to \mathbb{R}$ satisfies $\mathcal{R}_{\mathrm{z}}(f) \geq (n - 4t)/(3n)$.*

Proof. Recall the notation $\tilde{f}(x) := \mathbb{1}[f(x) \geq 1/2]$, whereby $\mathcal{R}_{\mathrm{z}}(f) := n^{-1} \sum_i \mathbb{1}[y_i \neq \tilde{f}(x_i)]$. 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 $\tilde{f}$. □

To close, the proof of Lemma 2.1 proceeds as follows. First note how adding and composing sawtooths grows their complexity.

Lemma 2.3. *Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be respectively k- and l-sawtooth. Then $f + g$ is $(k + l)$-sawtooth, and $f \circ g$ is kl-sawtooth.*

Proof of Lemma 2.3. 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 moreover any intervals $U_f \in \mathcal{I}_f$ and $U_g \in \mathcal{I}_g$. Necessarily, $f + g$ has a single slope along $U_f \cap U_g$. Consequently, $f + g$ is $|\mathcal{I}|$-sawtooth, where $\mathcal{I}$ is the set of all intersections of intervals from $\mathcal{I}_f$ and $\mathcal{I}_g$, meaning $\mathcal{I} := \{U_f \cap U_g : U_f \in \mathcal{I}_f, U_g \in \mathcal{I}_g\}$. By sorting the left endpoints of elements of $\mathcal{I}_f$ and $\mathcal{I}_g$, it follows that $|\mathcal{I}| \leq k + l$ (the other intersections are empty).

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 U_g was arbitrary, it holds that $f \circ g$ is $(|\mathcal{I}_f| \cdot |\mathcal{I}_g|)$-sawtooth. □

The proof of Lemma 2.1 follows by induction over layers of $\mathfrak{N}(\sigma; m, l)$.

Proof of Lemma 2.1. 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 2.3. 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 2.3, whereby applying σ yields a $(tm)^i$-sawtooth function (once again by Lemma 2.3). □

2.2 Upper bound

Consider the *mirror map* $f_{\mathrm{m}} : \mathbb{R} \to \mathbb{R}$, depicted in Figure 2, and 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 $f_{\mathrm{m}} \in \mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2)$; 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 $f_{\mathrm{m}}^k \in \mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2; k) \subseteq \mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2k)$.

To assess the effect of the *post-composition* $f_{\mathrm{m}} \circ g$ for any $g : \mathbb{R} \to \mathbb{R}$, note that $f_{\mathrm{m}} \circ g$ 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 (2007), that Fourier transforms are efficiently represented with deep networks.

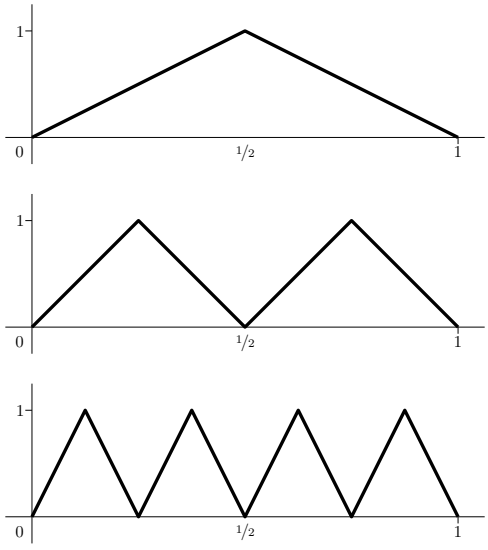

Figure 2: f_{m}, f_{m}^2, and f_{m}^3.

These compositions may be written as follows.

Lemma 2.4. *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 2.4. 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. □

Before closing this subsection, it is interesting to view f_{m}^k in one more way, namely its effect on $((x_i, y_i))_{i=1}^n$ provided by the n-ap with $n := 2^k$. Observe that $((f_{\mathrm{m}}(x_i), y_i))_{i=1}^n$ is an $(n/2)$-ap with all points duplicated except $x_1 = 0$, and an additional point with x-coordinate 1.

2.3 Proof of Theorems 1.1 and 1.2

It suffices to prove Theorem 1.2, which yields Theorem 1.1 since σ_{R} is 2-sawtooth, whereby the condition $m \le 2^{(k-3)/l-1}$ implies

$$\frac{n - 4(2m)^l}{3n} = \frac{1}{3} - (2m)^l 2^{-k} \left(\frac{4}{3}\right) \ge \frac{1}{3} - 2^{k-3} 2^{-k} \left(\frac{4}{3}\right) = \frac{1}{3} - \frac{1}{6},$$

and the upper bound transfers since $\mathfrak{R}(\sigma_{\mathrm{R}}; 2, 2; k) \subseteq \mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2k)$.

Continuing with Theorem 1.2, any $f \in \mathfrak{N}(\sigma; m, l)$ is $(tm)^l$-sawtooth by Lemma 2.1, whereby Lemma 2.2 gives the lower bound. For the upper bound, note that $f_{\mathrm{m}}^k \in \mathfrak{R}(\sigma_{\mathrm{R}}; 2, 2; k) \subseteq \mathfrak{N}(\sigma_{\mathrm{R}}; 2, 2k)$ by construction, and moreover $f_{\mathrm{m}}^k(x_i) = \tilde{f}_{\mathrm{m}}^k(x_i) = y_i$ on every (x_i, y_i) in the n-ap by Lemma 2.4.

3 Related work

The standard classical result on the representation power of neural networks is due to Cybenko (1989), who proved that neural networks can approximate continuous functions over $[0,1]^d$ arbitrarily well. This result, however, is for flat networks.

An early result showing the benefits of depth is due to Håstad (1986), who established, via an incredible proof, that boolean circuits consisting only of and gates and or gates require exponential size in order to approximate the parity function well. These gates correspond to multiplication and addition

over the boolean domain, and moreover the parity function is the Fourier basis over the boolean domain; as mentioned above, f_{m}^k as used here is a piecewise affine approximation of a Fourier basis, and it was suggested previously by Bengio and LeCun (2007) that Fourier transforms admit efficient representations with deep networks. Lastly, note that Håstad (1986)'s work has one of the same weaknesses as the present result, namely of only controlling a countable family of functions which is in no sense dense.

More generally, networks consisting of sum and product nodes, but now over the reals, have been studied in the machine learning literature, where it was showed by Bengio and Delalleau (2011) that again there is an exponential benefit to depth. While this result was again for a countable class of functions, more recent work by Cohen et al. (2015) aims to give a broader characterization.

Still on the topic of representation results, there is a far more classical result which deserves mention. Namely, the surreal result of Kolmogorov (1957) states that a continuous function $f : [0, 1]^d \to \mathbb{R}$ can be *exactly* represented by a network with $\mathcal{O}(d^2)$ nodes in 3 layers; this network needs multiple distinct nonlinearities and therefore is not an element of $\mathfrak{N}\left(\sigma; \mathcal{O}(d^2), 3\right)$ for a fixed σ, however one can treat these specialized nonlinearities as goalposts for other representation results. Indeed, similarly to the f_{m}^k used here, Kolmogorov's nonlinearities have fractal structure.

Lastly, while this note was only concerned with finite sets of points, it is worthwhile to mention the relevance of representation power to statistical questions. Namely, by the seminal result of Anthony and Bartlett (1999, Theorem 8.14), the VC dimension of $\mathfrak{N}(\sigma_{\mathrm{R}}; m, l)$ is at most $\mathcal{O}(m^8 l^2)$, indicating that these exponential representation benefits directly translate into statistical savings. Interestingly, note that f_{m}^k has an exponentially large Lipschitz constant (exactly 2^k), and thus an elementary statistical analysis via Lipschitz constants and Rademacher complexity (Bartlett and Mendelson, 2002) can inadvertently erase the benefits of depth as presented here.

References

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, Nov 2002.

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

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.

Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. 2015. `arXiv:1509.05009 [cs.NE]`.

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

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

Andrey Nikolaevich Kolmogorov. On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition. 114:953–956, 1957.