RETHINK DEPTH SEPARATION WITH INTRA-LAYER LINKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The depth separation theory is nowadays widely accepted as an effective explanation for the power of depth, which consists of two parts: i) there exists a function representable by a deep network; ii) such a function cannot be represented by a shallow network whose width is lower than a threshold. Here, we find that adding intra-layer links can greatly improve a network's representation capability through the bound estimation, dedicated construction, and functional space analysis. Then, we modify the depth separation theory by showing that a shallow network that previously cannot express some hard functions constructed by deep networks can do the job with intra-layer links inserted. Such functions include the renowned "sawtooth" functions. Our results supplement the existing depth separation theory in terms of shedding light on the boundary of the depth separation theory. Also, they suggest that once configured with an appropriate structure, a shallow and wide network may have great potential on a par with a deep network.

1 INTRODUCTION

Due to the widespread applications of deep networks in many important fields (LeCun et al., 2015), mathematically understanding the power of deep networks has been a central problem in deep learning theory. To resolve this problem, the key issue is figuring out how expressive a deep network is or how increasing depth facilitates the expressivity of a neural network, which has been an active research topic in the past years. So far, the studies on the expressivity of deep networks can be roughly divided into two categories. The first category is the complexity characterization that introduces appropriate complexity measures for functions represented by neural networks (Pascanu et al., 2013; Montufar et al., 2014; Telgarsky, 2015; Montúfar, 2017; Serra et al., 2018; Hu & Zhang, 2018; Xiong et al., 2020; Bianchini & Scarselli, 2014; Raghu et al., 2017), and then demonstrates that increasing depth can greatly boost such a complexity measure than increasing width.

The second category is constructing functions which can be expressed by a small network of a given depth, but cannot be approximated by shallower networks, unless its width is sufficiently large (Telgarsky, 2015; 2016; Arora et al., 2016; Eldan & Shamir, 2016; Safran & Shamir, 2017; Venturi et al., 2021). For example, Telgarsky (2015) employed a ReLU network to constructe a one-dimensional "sawtooth" function whose number of pieces scales exponentially over the depth. As such, a deep network can construct a sawtooth function with many pieces, while a shallow network cannot represent a piecewise linear function with the same number of pieces. Arora et al. (2016) derived the upper bound of the maximal number of pieces for a univariate ReLU network, and used this bound to demonstrate the separation between a deep and a shallow network. Eldan & Shamir (2016) constructed a radial function and used Fourier spectrum analysis to show that a two-hidden-layer network can represent it with a polynomial number of a neurons, but a one-hidden-layer network needs an exponential number of neurons to achieve the same level of error. In a broad sense, the elements of establishing a depth separation theorem are twofold: i) there exists a function representable by a deep network; ii) such a function cannot be represented by a shallow network whose width is lower than a threshold. The depth separation theory is nowadays widely accepted as an effective explanation for the power of depth.

However, the existing depth separation results implicitly assume that the structures of both deep and shallow networks are feedforward. As shown in Figure 1, based on the frameworks of ReLU networks (Telgarsky, 2015; Arora et al., 2016), we find that inserting

intra-layer links, which is to let a neuron take the outputs of both its neighboring neuron and neurons of the preceding layer, can greatly increase the maximum number of pieces represented by a shallow network. As such, without the need of going very wide, this shallow network can express as a complicated function as a deep network could. As a result, we can slightly modify the statement of depth separation: there exists a function that is expressed by a small network of a given depth, and cannot be approximated by a shallow network unless its width is sufficiently large, but such a function can be approximated by a shallow and narrow network when its neurons are intra-linked. Our result is valuable in two aspects. On the one hand, it non-trivially supplements the depth separation theory. In reality, a deep neural network often is not feedforward but uses shortcuts to link distant layers to facilitate feature reuse. Exploring the depth separation in the shortcut paradigm can shed light on the boundary of the depth separation results and reveal the genuine power of depth. On the other hand, the superiority of depth over width nowadays seems to become a doctrine in most deep learning practitioners. However, studies such as width-depth equivalence (Fan et al., 2020) and representation ability of wide networks (Lu et al., 2017; Levine et al., 2020) show the essential role of width. In the same vein, our result suggests that the potential of wide networks in expressivity can be unleashed by an appropriate structure, which lays a theoretical foundation for future empirical investigations.

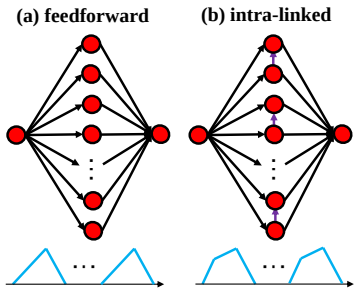

Figure 1: (a) a feedforward and (b) an intra-linked architecture.

Specifically, our roadmap to the modification of depth separation theorems includes two milestones. 1) Through bound analysis (Theorems 4 and 6), meticulous construction (Propositions 1, 2, 5, and 4), and functional space analysis (Theorem 8), we substantiate that a network with intra-layer links can produce much more pieces than without, and the gain is at most exponential, *i.e.*, $(\frac{3}{2})^k$, where k is the depth. 2) Since intra-layer links can give rise to more pieces, we show that intra-layer links can empower a shallow and narrow network such that a function that is representable by a deep network and unrepresentable by a shallow and narrow network now becomes representable by a shallow and narrow network. Such a function includes the sawtooth function constructed in (Telgarsky, 2015). The modification is done in the cases of k vs 2 (Theorem), k vs 3 (Theorem), and the general case k vs k', where $k' < k$. Also, we point out that the depth separation cannot be fully accomplished based on the bound analysis, unless the bound is shown to be tight by a construction.

2 RELATED WORK

In recent years, a good amount of papers have shown the superiority of deep networks over shallow ones from the points of view of complexity analysis and construction analysis.

The complexity analysis is to characterize the complexity of the function represented by a neural network, thereby demonstrating that increasing depth can greatly boost such a complexity measure than increasing width. Currently, one of the most popular complexity measures is the number of linear regions because it conforms to the functional structure of the widely-used ReLU networks. For example, Pascanu et al. (2013); Montufar et al. (2014); Montúfar (2017); Serra et al. (2018); Hu & Zhang (2018); Hanin & Rolnick (2019) estimated the bound of the number of linear regions generated by a fully-connected ReLU network by directly applying Zaslavsky's Theorem (Zaslavsky, 1997). Xiong et al. (2020) offered the first upper and lower bounds of the number of linear regions for convolutional networks. Other complexity measures include classification capabilities (Malach & Shalev-Shwartz, 2019), Betti numbers (Bianchini & Scarselli, 2014), trajectory lengths (Raghu et al., 2017), global curvature (Poole et al., 2016), and topological entropy (Bu et al., 2020). Please note that using complexity measures to justify the power of depth demands a tight bound estimation. Otherwise, it is insufficient to say that shallow networks cannot be as powerful as deep networks, since no networks can reach the upper bound. Unfortunately, the above-mentioned bounds are usually loose for deep networks.

The construction analysis is to find a family of functions that are hard to approximate by a shallow network, but can be efficiently approximated by a deep network. Eldan & Shamir (2016) built a special radial function that is expressible by a 3-layer neural network with a polynomial number

of neurons, but a 2-layer network can only do the same level approximation with an exponential number of neurons. Later, Safran & Shamir (2017) extended this result to a ball function, which is a more natural separation result. Venturi et al. (2021) generalized the construction of this type to a non-radial function. Telgarsky (2015; 2016) used an $\mathcal{O}(k^2)$-layer network to construct a sawtooth function. Thanks to a large number of pieces, such a function cannot be expressed by a $\mathcal{O}(k)$-layer network, unless the width is $\mathcal{O}(\exp(k))$. Arora et al. (2016) estimated the maximal number of pieces a network can produce, and established the size-piece relation to advance the depth separation results from (k, k^2) to (k, k'), where $k' < k$. Other smart constructions include polynomials (Rolnick & Tegmark, 2017), functions of a compositional structure (Poggio et al., 2017), Gaussian mixture models (Jalali et al., 2019), and so on.

3 Notation and Definition

Notations can be refined

Notation 1 (feedforward networks). For a classical ReLU DNN with depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers, we use $f_1^{(i)}$ to denote the output of the i-th neuron in the first layer, where $i = 1, \ldots, w_1$. Then we use $g_j^{(i)}$ and $f_j^{(i)} := \sigma\left(g_j^{(i)}\right)$, $i = 1, \ldots w_j$, $j = 2, \ldots, k$, to denote the pre-activation and output of the i-th neuron of the j-th layer, respectively. We arrange all $f_j^{(i)}$ into a vector $\mathbf{f}_j = [f_j^{(1)}, \cdots, f_j^{(w_i)}]$. Let $\mathbf{f}_0 = \mathbf{x}$, we have $g_{j+1}^{(i)} = \sigma(\langle \mathbf{a}_{j+1}^{(i)}, \mathbf{f}_j \rangle + b_{j+1}^{(i)})$, $j = 0, 1, \cdots, k-1$, where $\mathbf{a}_{j+1}^{(i)} \in \mathbb{R}^{w_j}$ Specially, we have the output of the network is $\mathbf{F}(\mathbf{x}) = \langle \mathbf{a}_{k+1}, \mathbf{f}_k \rangle + b_{k+1}$.

Notation 2 (intra-linked networks). For a ReLU DNN with every two neurons intra-linked in each layer, depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers, we use $\tilde{f}_1^{(i)}$ to denote the neurons of the first layer, where $i = 1, \ldots, w_1$. For the j-th layer, $j = 2, \ldots, w_k$, assume the i-th and $(i+1)$-th neurons are linked and $\tilde{g}_j^{(i)}, \tilde{g}_j^{(i+1)}$ are the corresponding output of the $(j-1)$-th layer. Then we use $\tilde{f}_j^{(i)} = \sigma\left(\tilde{g}_j^{(i)}\right)$ and $\tilde{f}_j^{(i+1)} = \sigma\left(\tilde{g}_j^{(i+1)} - \tilde{f}_j^{(i)}\right)$ to denote the pre-activation and output of the i-th neuron of the j-th layer, respectively. Similarly, we arrange all $\tilde{f}_j^{(i)}$ into a vector $\tilde{\mathbf{f}}_j = [\tilde{f}_j^{(1)}, \cdots, \tilde{f}_j^{(w_i)}]$. Let $\tilde{\mathbf{f}}_0 = \mathbf{x}$, we have $g_{j+1}^{(i)} = \sigma(\langle \mathbf{a}_{j+1}^{(i)}, \tilde{\mathbf{f}}_j \rangle + b_{j+1}^{(i)})$, $j = 0, 1, \cdots, k-1$, where $\mathbf{a}_{j+1}^{(i)} \in \mathbb{R}^{w_j}$ Specially, we have the output of the network is $\tilde{\mathbf{F}}(\mathbf{x}) = \langle \mathbf{a}_{k+1}, \tilde{\mathbf{f}}_k \rangle + b_{k+1}$.

Definition 1 (Width and depth of a feedforward network (Arora et al., 2016))**.** *For any number of hidden layers $k \in \mathbb{N}$, input and output dimensions $w_0, w_{k+1} \in \mathbb{N}$, a $\mathbb{R}^{w_0} \to \mathbb{R}^{w_{k+1}}$ feedforward network is given by specifying a sequence of k natural numbers $w_1, w_2, \ldots, w_k$ representing widths of the hidden layers, a set of k affine transformations $T_i : \mathbb{R}^{w_{i-1}} \to \mathbb{R}^{w_i}$ for $i = 1, \ldots, k$ and a linear transformation $T_{k+1} : \mathbb{R}^{w_k} \to \mathbb{R}^{w_{k+1}}$ corresponding to weights of the hidden layers. The function $f : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ computed or represented by this network is*

$$f = T_{k+1} \circ \sigma \circ T_k \circ \cdots \circ T_2 \circ \sigma \circ T_1, \tag{1}$$

where $\circ$ denotes function composition, and σ is an activation function. The depth of a ReLU DNN is defined as $k+1$. The width of a ReLU DNN is $\max\{w_1, \ldots, w_k\}$.

Definition 2 (Width and depth of a shortcut network (Fan et al., 2020))**.** *Given a shortcut network $\mathbf{\Pi}$, we delete a minimum number of links such that the resultant network $\mathbf{\Pi}'$ is a feedforward network without any isolated neuron. Then, we define the width and depth of $\mathbf{\Pi}$ as the width and depth of $\mathbf{\Pi}'$, respectively.*

Admittedly, defining the width and depth of a network embedded with shortcuts is tricky. Our criterion is that a reasonably good definition should conform to the customary understanding to depth and width. For example, the ResNet shall not be taken as a wide network in the light of the proposed definition, conflicting practioners' common sense. The scheme we adopt satisfies the criterion.

4 RETHINK THE DEPTH SEPARATION WITH INTRA-LAYER LINKS

Our focus is the network using ReLU activation and related estimation of the number of pieces. The seminal depth separation results closest to us are the following:

Theorem 1 (Depth separation k^2 vs k (Telgarsky, 2015; 2016))**.** *For every pair of natural numbers $k \geq 1, w \geq 2$, there exists a sawtooth function representable by an $\mathbb{R} \to \mathbb{R}$ (k^2+1)-layer feedforward ReLU DNN of width 2 such that if it is also representable by a $k+1$-layer feedforward ReLU DNN, this $k+1$-layer feedforward ReLU DNN should at least have the width of 2^k.*

Theorem 2 (Depth separation k vs k' (Arora et al., 2016))**.** *For every pair of natural numbers $k \geq 1, w \geq 2$, there exists functions representable by an $\mathbb{R} \to \mathbb{R}$ $(k+1)$-layer feedforward ReLU DNN of width w such that if it is also representable by a $(k'+1)$-layer feedforward ReLU DNN for any $k' \leq k$, then this $(k'+1)$-layer feedforward ReLU DNN has width at least $\frac{1}{2}w^{\frac{k}{k'}}$.*

Despite the construction is one-dimensional, these results convincingly reveal that increasing depth can produce more pieces than increasing width, which is the heart of depth separation. Here, we shed new light on the depth separation problem with intra-layer links. Our primary argument is that if intra-layer links shown in Figure 1 are inserted, a shallow network that previously cannot express some previously hard functions constructed by deep networks can do the job. Our investigation consists of two parts. First, we show that adding intra-layer links can greatly increase the number of pieces from bound estimation, dedicated construction, and functional space analysis. Then, we show that adding intra-layer links can empower the shallow networks: Previously, a shallow network cannot represent some functions represented by a deep network, *e.g.* sawtooth function, unless it is very wide. But now it can do the job with the intra-layer links.

4.1 INTRA-LAYER LINKS CAN INCREASE THE NUMBER OF PIECES

4.1.1 UPPER BOUND ESTIMATION

Lemma 3. *Let $g : \mathbb{R} \to \mathbb{R}$ be a PWL function with $w+1$ pieces, then the breakpoints of $f := \sigma(g)$ consists of two parts: old breakpoints of g and at most $w+1$ newly generated breakpoints. Furthermore, f can generate $w+1$ new breakpoints if and only if g has $w+1$ zero points, and every two adjacent pieces of g have opposite slopes.*

Proof. A direct calculus. □

Theorem 4 (Upper bound of feedforward networks)**.** *Let $F : \mathbb{R} \to \mathbb{R}$ be a function represented by an $\mathbb{R} \to \mathbb{R}$ ReLU DNN with depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers. Then F is a PWL function whose number of pieces is at most $\prod_{i=1}^{k}(w_i+1)$.*

Proof. This bound is the special case of the bound: $\prod_{i=1}^{k}\sum_{j=0}^{n}\binom{w_i}{j}$ derived in Montúfar (2017) for n-dimensional inputs. □

Remark 1 (Sharpening the bound in Arora et al. (2016)). Previously, Arora et al. (2016) showed that the number of pieces produced by a network of depth $k+1$ and widths $w_1, \ldots, w_k$ is $2^{k+1} \cdot (w_1+1)w_2 \cdots w_k$. The reason why their bound has an extra exponential term is that they repetitively computed the old breakpoints of the output of neurons in the previous layer. Our Lemma 3 implies the ReLU activation in fact cannot generate as many as double pieces.

Lemma 5. *Let $g_1, g_2 : \mathbb{R} \to \mathbb{R}$ be two PWL functions with totally w breakpoints. Set $f_1 := \sigma(g_1)$ and $f_2 := \sigma(g_2 - f_1)$. Then the breakpoints of $\alpha_2 f_2$, $\alpha_1, \alpha_2 \in \mathbb{R}$, consists of three parts: some breakpoints of g_2, some breakpoints of f_1 and at most $2w+2$ newly generated breakpoints. Furthermore, f_2 has $2w+2$ newly generated breakpoints if and only if $g_2 - f_1$ is of opposite slope on every two adjacent pieces and have $2w+2$ distinct zero points.*

Proof. Proof: A direct corollary of Lemma 3. □

Lemma 5 illustrates why the intra-linked architecture can produce more pieces. Given two PWL functions g_1 and g_2 which has totally w breakpoints, in the feedforward architecture, $\sigma(g_1)$ and $\sigma(g_2)$ have totally at most $3w+2$ breakpoints, which consists at most w old breakpoints of g_1, g_2 and at most $2w+2$ newly generated breakpoints. However, in the intra-linked architecture, $\sigma(g_1)$ and $\sigma(g_2 - \sigma(g_1))$ has totally $4w+3$ breakpoints, which consist of at most w old breakpoints of g_1, g_2 , at most $(w+1)+(2w+2) = 3w+3$ newly produced breakpoints.

Theorem 6 (Upper bound of intra-linked networks)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a PWL function represented by a ReLU DNN with depth $k+1$, widths $w_1, \ldots, w_k$, and every two neurons linked in each hidden layer. Assuming that $w_1, \ldots, w_k$ are even, f has at most $\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)$ pieces.*

Proof. We prove by induction on k. For the base case $k=1$, the number of breakpoints of $\tilde{f}_1^{(i)}$, $i = 1, \ldots, w$, is at most $2+(-1)^i$. Note that for every odd i, the breakpoints of $\tilde{f}^{(i)}$ is contained in those of $\tilde{f}^{(i+1)}$. Hence, the first layer gives at most $\frac{3}{2}w_1+1$ pieces. For the induction step, we assume that for some $k \geq 1$, any $\mathbb{R} \to \mathbb{R}$ ReLU DNN with every two neurons linked in each hidden layer, depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers produces at most $\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)$ pieces. Now we consider any $\mathbb{R} \to \mathbb{R}$ ReLU DNN with every two neurons linked in each hidden layer, depth $k+2$ and widths $w_1, \ldots, w_{k+1}$ of $k+1$ hidden layers. By the induction hypothesis, each $\tilde{g}_{k+1}^{(i)}$ has at most $\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)-1$ breakpoints. Then the breakpoints of $\tilde{f}_{k+1}^{(i)}$ consist of some breakpoints of $\tilde{g}_{k+1}^{(i)}$ and at most $\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)$ newly generated breakpoints. We assume the neurons $\tilde{f}_{k+1}^{(i)}$ and $\tilde{f}_{k+1}^{(i+1)}$ are linked. Then $\tilde{g}_{k+1}^{(i)} - \tilde{f}_{k+1}^{(i)}$ has at most $2 \cdot \prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)-1$ breakpoints. Similar with $\tilde{f}_{k+1}^{(i)}$, the breakpoints of $\tilde{f}_{k+1}^{(i+1)}$ consist of some breakpoints of $\tilde{g}_{k+1}^{(i)} - \tilde{f}_{k+1}^{(i)}$ and at most $2 \cdot \prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)$ newly generated breakpoints. Note that $\tilde{g}_{k+1}^{(1)}, \ldots, \tilde{g}_{k+1}^{w_{k+1}}$ have totally at most $\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)-1$ breakpoints. In all, the pieces we can therefore get is at most

$$1+\frac{w_{k+1}}{2}\cdot\left(\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)+2\cdot\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)\right)+\prod_{i=1}^{k}\left(\frac{3}{2}w_i+1\right)-1=\prod_{i=1}^{k+1}\left(\frac{3}{2}w_i+1\right)$$

□

4.1.2 METICULOUS CONSTRUCTION.

Despite that the above bound estimation offers some light, to convincingly illustrate that adding intra-layer links can increase the number of pieces, we need to supply the construction for the intra-linked networks. The number of pieces of the desirable construction should either be bigger than the upper bound of feed-forward networks (Propositions 1 and 2) or the maximal number the feed-forward networks can achieve (Proposition 4).

Proposition 1 (The bound $\prod_{i=1}^{k}(w_i+1)$ is tight for a depth-bounded but width-unbounded network)**.** *Given an $\mathbb{R} \to \mathbb{R}$ two-hidden-layer ReLU network, for any width $w_1 \geq 3, w_2 \geq 2$ in the first and second hidden layers, there exists a piecewise linear function represented by such a network, whose number of pieces is $(w_1+1)(w_2+1)$.*

Proof. Let $f_1^{(1)}(x) = \sigma(3x), f_1^{(2)}(x) = \sigma(-x+3), f_1^{(3)}(x) = \sigma\left(\frac{3}{2}x - \frac{3}{2}\right)$. When $w_1 = 3$, we set

$$\begin{aligned} g_2^{(1)} &= -\left(f_1^{(1)} + f_1^{(2)} - f_1^{(3)} - 3 - \tfrac{1}{w_2+1}\right), \\ g_2^{(i)} &= f_1^{(1)} + f_1^{(2)} - f_1^{(3)} - 3 - \tfrac{i}{w_2+1}, i = 2, \ldots, w_2. \end{aligned}$$

When $w_1 > 3$, we let $f_1^{(j)} = \sigma(-2x - 2(j-3))$ and

$$\begin{aligned} g_2^{(1)} &= -\left(f_1^{(1)} + f_1^{(2)} - f_1^{(3)} + \textstyle\sum_{j=4}^{w_1}(-1)^{j-1} f_1^{(j)} - 3 - \tfrac{1}{w_2+1}\right) \\ g_2^{(i)} &= f_1^{(1)} + f_1^{(2)} - f_1^{(3)} + \textstyle\sum_{j=4}^{w_1}(-1)^{j-1} f_1^{(j)} - 3 - \tfrac{i}{w_2+1}, i = 2, \ldots, w_2. \end{aligned}$$

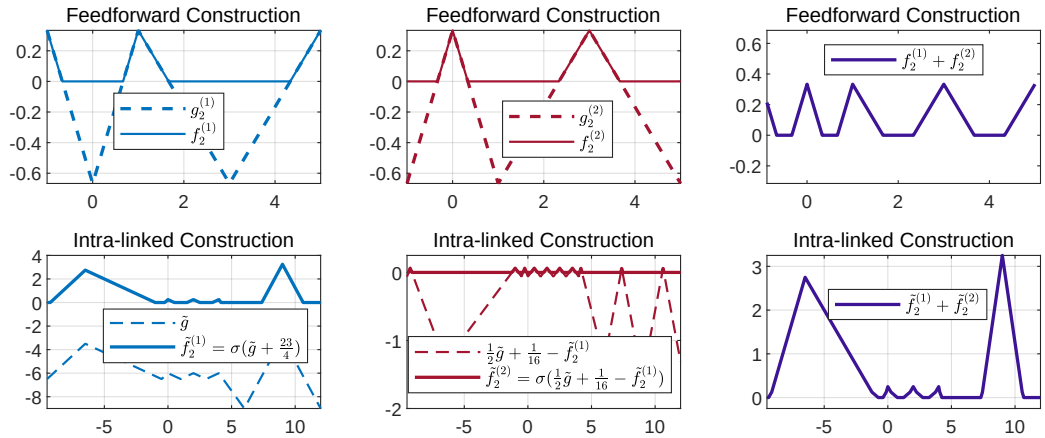

Figure 2: Construction of piecewise linear functions to reach the bound of Proposition 1 when $w_1 = 3$, $w_2 = 2$.

Then $g_2^{(i)}$ has $w_1 + 1$ pieces and every two adjacent pieces have two opposite slopes. Besides, $g_2^{(i)} = 0$ intersects with every piece. Hence for $i = 1, \ldots, w_2$, the breakpoints of $f_2^{(i)} = \sigma\left(g_2^{(i)}\right)$ keeps all breakpoints of $g_2^{(i)}$ and yields $w_1 + 1$ new breakpoints. Note that $f_2^{(i)}$ and $f_2^{(j)}$ do not share new breakpoints, and $f_2^{(1)}$ and $f_2^{(2)}$ covers all the breakpoints of $\left\{g_2^{(i)}\right\}_{i=1}^{w_2}$. Therefore, the total number of pieces via an affine combination of $f_2^{(1)}, \ldots, f_2^{(w_2)}$ is $(w_1 + 1)(w_2 + 1)$ pieces. $\square$

Proposition 2 (The bound $\prod_{i=1}^{k}\left(\frac{3w_i}{2} + 1\right)$ is tight for a depth-bounded but width-unbounded network)**.** *Given an $\mathbb{R} \to \mathbb{R}$ two-hidden-layer ReLU network, with every two neurons linked in each hidden layer, for any even $w_1 \geq 6, w_2 \geq 4$, there exists a piecewise linear function represented by such a network, whose number of pieces is $\left(\frac{3w_1}{2} + 1\right)\left(\frac{3w_2}{2} + 1\right)$.*

Proof. Let

$$\tilde{f}_1^{(1)}(x) = \sigma\left(\tfrac{9}{2}x - 27\right), \ \tilde{f}_1^{(2)}(x) = \sigma\left(\tfrac{3}{2}x - \tilde{f}_1^{(1)}(x)\right)$$
$$\tilde{f}_1^{(3)}(x) = \sigma(-2x + 2), \ \ \tilde{f}_1^{(4)}(x) = \sigma\left(-x + 2 - \tilde{f}_1^{(3)}(x)\right)$$
$$\tilde{f}_1^{(5)}(x) = \sigma\left(-\tfrac{7}{2}x - \tfrac{7}{4}\right), \ \tilde{f}_1^{(6)}(x) = \sigma\left(-2x + 8 - \tilde{f}_1^{(5)}(x)\right).$$

When $w_1 = 6$, we set

$$\tilde{g} = -\frac{2}{9}\tilde{f}_1^{(1)} - \tilde{f}_1^{(2)} + \frac{1}{2}\tilde{f}_1^{(3)} + \tilde{f}_1^{(4)} - \frac{4}{7}\tilde{f}_1^{(5)} - \tilde{f}_1^{(6)}.$$

When $w_1 > 6$, for each odd $i > 6$, we further let

$$f_1^{(i)} = \sigma\left(-5\left(x - a_i + 3\right)\right), f_1^{(i+1)} = \sigma\left(-2\left(x - a_i\right) - f_1^{(i)}\right)$$

where $a_i = -\frac{19}{2} - 9\left(\frac{i-1}{2} - 3\right)$, and we set

$$\tilde{g} = -\frac{2}{9}\tilde{f}_1^{(1)} - \tilde{f}_1^{(2)} + \frac{1}{2}\tilde{f}_1^{(3)} + \tilde{f}_1^{(4)} - \frac{4}{7}\tilde{f}_1^{(5)} - \tilde{f}_1^{(6)} + \sum_{i=7, i \text{ is odd}}^{w_2} (-1)^{\frac{i+1}{2}}\left(\frac{2}{5}f_1^{(i)} + f_1^{(i+1)}\right)$$

Then $\tilde{g}$ has $\frac{3}{2}w_1 + 1$ pieces, and every two adjacent pieces have two opposite slopes. Besides, provided that $b_1 \in (6, 13/2)$, $\tilde{g} + b_1$ has $\frac{3}{2}w_1 + 1$ distinct zero-points. As a result, $\tilde{f}_2^{(1)} := \sigma\left(\tilde{g} + b_1\right)$ have new $\frac{3}{2}w_1 + 1$ breakpoints. Then by choosing some scale parameter $a \in (-1, 0)$, we can make

$\tilde{g}+\tilde{f}_2^{(1)}$ has $\left(\frac{3}{2}w_1+1\right)+\frac{3}{2}w_1 = 3w_1+1$ distinct breakpoints and is of slope with different signs on every two adjacent pieces. By choosing some bias parameter b_2, we can also make $\tilde{g}+b_2+\tilde{f}_2^{(1)}$ has $3w_1+2$ distinct zero-points, which implies $\tilde{f}_2^{(2)} := \sigma\left(a\tilde{g}+b_2-\tilde{f}_2^{(1)}\right)$ has $3w_1+2$ newly produced breakpoints. Therefore, the affine combination of $\tilde{f}_2^{(1)}$ and $\tilde{f}_2^{(2)}$ has $\frac{3}{2}w_1+\left(\frac{3}{2}w_1+1\right)+(3w_1+2)$ breakpoints, which consists of all $\frac{3}{2}w_1$ breakpoints of $\tilde{g}$. Repeating this procedure $\frac{w_2}{2}$ times, we can construct all the $\tilde{f}_2^{(i)}$, $i=1,\dots w_2$, and make the affine combination of $\left\{\tilde{f}_2^{(i)}\right\}_{i=1}^{w_2}$ has pieces of

$$\frac{3}{2}w_1+\frac{w_2}{2}\cdot\left(\frac{3}{2}w_1+1\right)+\frac{w_2}{2}\cdot(3w_1+2)+1=\left(\frac{3w_1}{2}+1\right)\left(\frac{3w_2}{2}+1\right)$$

□

Proposition 3 (Use intra-linked network to achieve $\prod_{i=1}^{k}\left(\frac{3w_i}{2}\right)$ pieces)**.** *There exists a $\mathbb{R}\to\mathbb{R}$ function represented by a $\mathbb{R}\to\mathbb{R}$ intra-linked ReLU DNN with depth $k+1$ and width $w_1,\dots,w_k$ of k hidden layers, whose number of pieces is at least $\frac{3w_1}{2}\cdot\ldots\cdot\frac{3w_k}{2}$.*

Proof. Let $\tilde{g}=x$

Let $\tilde{g}$ be a PWL function and $\tilde{g}=x-a$ on $[a,b]$. Let $w\geq 2$ be some positive even integer and $\delta=\frac{2(b-a)}{3w}$. Set $g^{(1)}=3\tilde{g}-3\delta$, $f^{(1)}=\sigma\left(g^{(1)}\right)$, $g^{(2)}=\tilde{g}$, $f^{(2)}=\sigma\left(g^{(2)}-f^{(1)}+\delta\right)$ and $g^{(2j+1)}=4\tilde{g}-12j\delta$, $f^{(2j+1)}=\sigma\left(g^{(2j+1)}\right)$, $g^{2j+2}=2\tilde{g}-2(3j+1)\delta$, $f^{(2j+2)}=\sigma\left(g^{(2j+2)}-f^{(2j+1)}\right)$, for all $j=1,\dots,w/2-1$. Let

$$g=\frac{1}{3}f^{(1)}+f^{(2)}-\delta+\sum_{j=1}^{\frac{w}{2}-1}(-1)^j\left(\frac{1}{2}f^{(2n+1)}+f^{(2n+2)}\right),$$

which has $3w/2$ pieces on $[a,b]$. Then g is of slope $(-1)^j$ on $[a+j\delta,a+(j+1)\delta]$, $j=0,\dots,3w/2-1$, and ranges from 0 to δ on each piece.

Let $\tilde{g}$ be a PWL function of slope $(-1)^i$ on $[a+ib,a+(i+1)b]$ and ranges from 0 to b on each piece, where $b>0$ is some constant and $i=0,1,\dots,N-1$. Let $w\geq 2$ be some positive even integer and $\delta=\frac{2b}{3w}$. Set $g^{(1)}=3\tilde{g}-3\delta$, $f^{(1)}=\sigma\left(g^{(1)}\right)$, $g^{(2)}=\tilde{g}$, $f^{(2)}=\sigma\left(g^{(2)}-f^{(1)}+\delta\right)$ and $g^{(2j+1)}=4\tilde{g}-12j\delta$, $f^{(2j+1)}=\sigma\left(g^{(2j+1)}\right)$, $g^{2j+2}=2\tilde{g}-2(3j+1)\delta$, $f^{(2j+2)}=\sigma\left(g^{(2j+2)}-f^{(2j+1)}\right)$, for all $j=1,\dots,w/2-1$. Let

$$g=\frac{1}{3}f^{(1)}+f^{(2)}-\delta+\sum_{j=1}^{\frac{w}{2}-1}(-1)^j\left(\frac{1}{2}f^{(2n+1)}+f^{(2n+2)}\right),$$

which has $3Nw/2$ pieces on $[a,b]$. Then g is of slope $(-1)^j$ on $[a+j\delta,a+(j+1)\delta]$, $j=0,\dots,3Nw/2-1$, and ranges from 0 to δ on each piece.

Note that for any $x\in[a+ib,a+(i+1)b]$, $1\leq i\leq N$, $\tilde{g}(x)=\tilde{g}(2a+2(i+1)b-x)$, i.e. every two adjacent pieces are symmetric. The symmetry of $\tilde{g}$.

□

Proposition 4 (Intra-layer links can greatly increase the number of pieces in an $\mathbb{R}\to\mathbb{R}$ ReLU network with width 2 and any depth.)**.** *Let $f:\mathbb{R}\to\mathbb{R}$ be a function represented by a $\mathbb{R}\to\mathbb{R}$ $(k+1)$-layer ReLU DNN with widths 2 of all k hidden layers. Then number of pieces of f is at most*

$$\begin{cases}\sqrt{7}^k, & \text{if } k \text{ is even,}\\ 3\cdot\sqrt{7}^{k-1}, & \text{if } k \text{ is odd.}\end{cases}$$

There exists a $\mathbb{R}\to\mathbb{R}$ $(k+1)$-layer 2-wide ReLU DNN, with neurons linked in each hidden layer, which can produce at least $7\cdot 3^{k-2}+2$ pieces.

Proof. The proof is put in Appendix B. □

4.1.3 FUNCTIONAL SPACE ANALYSIS

The above constructive analyses demonstrate that in the maximal sense, intra-layer links can empower a feedforward network to represent a function with more pieces. Now, we derive an even stronger result: surprisingly, adding intra-layer links can expand the functional space of a network. The reason why this result is surprising is that one tends to think that an intra-linked network produces a different function from a feedforward network. However, such a different can be somehow mitigated by adjusting the weights of a network.

Lemma 7. *Let* $f^{(1)} = \sigma\left(a^{(1)}x + b^{(1)}\right), f^{(2)} = \sigma\left(a^{(2)}x + b^{(2)}\right), f = c^{(1)}f^{(1)} + c^{(2)}f^{(2)} + d$*, where* $a^{(1)}a^{(2)} > 0, b^{(1)}, b^{(2)}, c^{(1)}, c^{(2)}$ *and* $d \in \mathbb{R}$*. Then there exists some* $\tilde{a}^{(1)}, \tilde{a}^{(2)}, \tilde{b}^{(1)}, \tilde{b}^{(2)}, \tilde{c}^{(1)}, \tilde{c}^{(2)}$ *and* $\tilde{d} \in \mathbb{R}$ *such that* $f = \tilde{c}^{(1)}\tilde{f}^{(1)} + \tilde{c}^{(2)}\tilde{f}^{(2)} + \tilde{d}$*, where* $\tilde{f}^{(1)} = \sigma\left(\tilde{a}^{(1)}x + \tilde{b}^{(1)}\right), \tilde{f}^{(2)} = \sigma\left(\tilde{a}^{(2)}x + \tilde{b}^{(2)} - \tilde{f}^{(1)}\right)$.

Proof. Without loss of generality, we assume $a^{(1)}, a^{(2)} > 0$ and $-\frac{b^{(2)}}{a^{(2)}} < -\frac{b^{(1)}}{a^{(1)}}$. Then f is of slope 0, $c^{(2)}a^{(2)}$ and $c^{(1)}a^{(1)} + c^{(2)}a^{(2)}$ on $\left(-\infty, -\frac{b^{(2)}}{a^{(2)}}\right], \left[-\frac{b^{(2)}}{a^{(2)}}, -\frac{b^{(1)}}{a^{(1)}}\right]$ and $\left[-\frac{b^{(1)}}{a^{(1)}}, \infty\right)$, respectively. Now we choose any $\tilde{a}^{(1)}, \tilde{a}^{(2)}$ satisfying $0 < \tilde{a}^{(1)} < \tilde{a}^{(2)}$, and set $\tilde{b}^{(i)} = \frac{b^{(i)}}{a^{(i)}} \cdot \tilde{a}^{(i)}, i = 1, 2$. Then $\tilde{f}^{(1)}$ is of slope 0 and $\tilde{a}^{(1)}$ on $\left(-\infty, -\frac{b^{(1)}}{a^{(1)}}\right]$ and $\left[-\frac{b^{(1)}}{a^{(1)}}, \infty\right)$ respectively while $\tilde{f}^{(2)}$ is of slope 0, $\tilde{a}^{(2)}$ and $\tilde{a}^{(2)} - \tilde{a}^{(1)}$ on $\left(-\infty, -\frac{b^{(2)}}{a^{(2)}}\right), \left[-\frac{b^{(2)}}{a^{(2)}}, -\frac{b^{(1)}}{a^{(1)}}\right]$ and $\left(-\frac{b^{(1)}}{a^{(1)}}, \infty\right)$ respectively. Hence, letting $\tilde{c}^{(2)} = c^{(2)}a^{(2)}/\tilde{a}^{(2)}, \tilde{c}^{(1)} = \left(c^{(1)}a^{(1)} + c^{(2)}a^{(2)} - \tilde{c}^{(2)}\left(\tilde{a}^{(2)} - \tilde{a}^{(1)}\right)\right)/\tilde{a}^{(1)}$ and $\tilde{d} = d$, we have $f = \tilde{c}^{(1)}\tilde{f}^{(1)} + \tilde{c}^{(2)}\tilde{f}^{(2)} + \tilde{d}$.

□

Theorem 8. *Let* f *be any* $\mathbb{R} \to \mathbb{R}$ *PWL representable by a classical* $k+1$*-layer ReLU DNN with widths* $w_1, \ldots, w_k$ *of* k *hidden layers. Then it can also be represented by a* $k+1$*-layer ReLU DNN with widths* $w_1, \ldots, w_k$ *of* k *hidden layers and some of the neurons in the first layer linked. Furthermore, the number of neurons that is not linked can be no more than* 2.

Proof. Assume that the neuron of the first layer in the classical network is $f_1^{(i)}(x) = \sigma\left(a_1^{(i)}x + b_1^{(i)}\right)$, $i = 1, \ldots, w_1$. Since the classical ReLU DNN is invariant under permutation of neurons in a hidden layer, we can link $f_1^{(i)}$ and $f_1^{(j)}$ if $a_1^{(i)}a_1^{(j)} > 0$. Then the result follows directly from Lemma 7. □

Remark 2. Although both going deep and adding intra-layer links involve composition. their mechanisms of producing many pieces are fundamentally different. While the mechanism of going deep is the repetition effect, *i.e.*, the function value of the function being composed is oscillating, and each oscillation can generate corresponding pieces. The mechanism of intra-layer links is the gating effect. The neuron being embedded have two activation states, and each state is leveraged by the neuron being linked to produce a break point. Such a mechanism essentially conforms to the parallelism, which is of width paradigm.

4.2 MODIFY THE DEPTH SEPARATION THEOREM WITH INTRA-LAYER LINKS

In a broad sense, the depth separation theorem consists of two elements: i) there exists a function representable by a deep network; ii) such a function cannot be represented by a shallow network whose width is lower than a threshold. Since the intra-layer links can generally improve the capabilities of networks, if one adds intra-layer links to a shallow network, the function constructed by a deep network can be represented by a shallow network, even if the width of this shallow network is lower than the threshold.

Lemma 9 (A network with width=2 can approximate any univariate piecewise linear function (Fan et al., 2018))**.** *Given a univariate piecewise linear function with* n *pieces* $p(x)$*, there exists a* $(n+1)$*-layer network* $\mathbf{D}(x)$ *with two neurons in each layer such that* $f(x) = \mathbf{D}(x)$.

Theorem 10 (Modify the depth separation k vs 2)**.** *For every $k \geq 1$, there exists a function $p(x)$ that can be represented by a $(k+1)$ -layer ReLU DNN with 2 nodes in each layer, such that it cannot be represented by a classical 2-layer ReLU DNN $\mathbf{W}_1(x)$ with width less than k, but can be represented by a 2-layer, $\frac{2k}{3}$-wide intra-linked ReLU DNN $\tilde{\mathbf{W}}_1(x)$ in the sense that $\tilde{\mathbf{W}}_1(x) = p(x)$.*

Proof. Because the bounds shown in Theorems 1 and 2 are tight for a two-layer network, combining Lemma 9 directly concludes the proof. □

Theorem 11 (Modify the depth separation k^2 vs 3)**.** *For every $k \geq 1$, there exists a function $p(x)$ that can be represented by a (k^2+1) -layer ReLU DNN with 2 nodes in each layer, such that it cannot be represented by a classical 3-layer ReLU DNN $\mathbf{W}_2(x)$ with width less than k, but can be represented by a 2-layer, $\frac{2k}{3}$-wide intra-linked ReLU DNN $\tilde{\mathbf{W}}_2(x)$ in the sense that $\tilde{\mathbf{W}}_2(x) = p(x)$.*

Proof. This theorem can be straightly obtained by combining Propositions 1, 2 and Lemma 9. □

Theorem 12 (Modify the depth separation k vs k')**.** *For every $k \geq 1$,*

Remark 3. Theorems 10, 11, and 12 demonstrate that adding intra-layer links can reduce the bar of the width by 1/3. Although it is not an exponential reduction, these results are still beneficial. On the one hand, our highlight is not how many shallow networks are transformed by intra-layer links but there exist networks can be transformed by intra-layer links to have the representation power on a par with a deep network. On the other hand, suppose every n neuron are intra-linked in each layer, the upper bound of pieces by a network with depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers is $\prod_{i=1}^{k} \left(\frac{(2^n-1)w_i}{n} + 1 \right)$. Therefore, by embedding more intra-layer links, the bar of the width can be further substantially reduced.

5 DISCUSSION AND CONCLUSION

Synergize inner and outer links. We take the intra-layer links as inner links, while the shortcuts of ResNet (He et al., 2016) and so on as outer links. It is widely recognized that the addition of outer links can facilitate networks in the dimension of depth, *e.g.*, the residual connection solves the training issues of deep networks. In this draft, our theoretical results show that inner links can promote the representation capability of networks in the dimension of width. Therefore, the synergy of both inner and outer links in a network will further contribute to the construction of more powerful networks. More favorably, both inner and outer links do not incorporate new parameters; therefore, their synergy is likely to lead to a compact model.

In this draft, via bound estimation, constructions, and functional space analysis, we have shown that a network with intra-layer links is much more expressive than a network without. Then, we have modified the depth separation results to that a shallow network that previously cannot express some previously hard functions constructed by deep networks can do the job with intra-layer links added. Our results supplement the existing depth separation theory, and suggest that the potential of wide networks can be released by an appropriate structure. Future endeavors can be put into training wide networks using intra-layer links to achieve comparable performance with deep networks.

REFERENCES

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems*, 25(8):1553–1565, 2014.

Kaifeng Bu, Yaobo Zhang, and Qingxian Luo. Depth-width trade-offs for neural networks via topological entropy. *arXiv preprint arXiv:2010.07587*, 2020.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pp. 907–940. PMLR, 2016.

Feng-Lei Fan, Rongjie Lai, and Ge Wang. Quasi-equivalence of width and depth of neural networks. *arXiv preprint arXiv:2002.02515*, 2020.

Fenglei Fan, Dayang Wang, Hengtao Guo, Qikui Zhu, Pingkun Yan, Ge Wang, and Hengyong Yu. On a sparse shortcut topology of artificial neural networks, 2018.

Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, pp. 359–368, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Qiang Hu and Hao Zhang. Nearly-tight bounds on linear regions of piecewise linear neural networks. *arXiv preprint arXiv:1810.13192*, 2018.

Shirin Jalali, Carl Nuzman, and Iraj Saniee. Efficient deep learning of gmms. *arXiv preprint arXiv:1902.05707*, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. The depth-to-width interplay in self-attention. *arXiv preprint arXiv:2006.12467*, 2020.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: a view from the width. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6232–6240, 2017.

Eran Malach and Shai Shalev-Shwartz. Is deeper better only when shallow is good? *Advances in Neural Information Processing Systems*, 32, 2019.

Guido Montúfar. Notes on the number of linear regions of deep neural networks. *Sampling Theory Appl., Tallinn, Estonia, Tech. Rep*, 2017.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pp. 2924–2932, 2014.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pp. 2847–2854. PMLR, 2017.

David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*, 2017.

Itay Safran and Ohad Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *International conference on machine learning*, pp. 2979–2987. PMLR, 2017.

Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pp. 4558–4566. PMLR, 2018.

Richard P. Stanley. An introduction to hyperplane arrangements. In *Lecture Notes, IAS/Park City Mathematics Institute*, 2004.

Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on learning theory*, pp. 1517–1539. PMLR, 2016.

Luca Venturi, Samy Jelassi, Tristan Ozuch, and Joan Bruna. Depth separation beyond radial functions. 2021.

Huan Xiong, Lei Huang, Mengyang Yu, Li Liu, Fan Zhu, and Ling Shao. On the number of linear regions of convolutional neural networks. In *International Conference on Machine Learning*, pp. 10514–10523. PMLR, 2020.

T. Zaslavsky. *Facing up to arrangements : face-count formulas for partitions of space by hyperplanes*. Number 154 in Memoirs of the American Mathematical Society. American Mathematical Society, 1975.

T Zaslavsky. Facing up to arrangements: face-count formulas for partitions of space by hyperplanes. *Memoirs of American Mathematical Society*, 154:1–95, 1997.

A Upper Bound in High-Dimensional Cases

Lemma 13 (Zaslavsky's Theorem Zaslavsky (1975); Stanley (2004))**.** *Let $\mathcal{A} = \{H_i \subset V : 1 \leq i \leq m\}$ be an arrangement in $\mathbf{R}^n$. Then, the number of regions for the arrangement $\mathcal{A}$ satisfies*

$$r(\mathcal{A}) \leq \sum_{i=0}^{n} \binom{m}{i}. \tag{2}$$

Theorem 14 (Upper Bound of Feedforward Networks Montúfar (2017).)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a PWL function represented by a $\mathbb{R}^n \to \mathbb{R}$ ReLU DNN with depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers. There we assume each w_i is even. Then f has at most $\prod_{i=1}^{k} \sum_{j=0}^{n} \binom{w_i}{j}$ linear regions.*

Theorem 15 (Upper Bound of Intra-linked Networks)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a PWL function represented by a $\mathbb{R}^n \to \mathbb{R}$ ReLU DNN with every two neurons linked in each hidden layer, depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers. There we assume each w_i is even. Then f has at most $\prod_{i=1}^{k} \sum_{j=0}^{n} \binom{\frac{3w_i}{2}+1}{j}$ linear regions.*

Proof. We prove by induction on k. For the base case $k = 1$, $\tilde{f}_1^{(2i-1)} = \sigma\left(\tilde{g}_1^{(2i-1)}\right)$ produces one hyperplane in the input space $\mathbb{R}^n$. Furthermore, $\tilde{f}_1^{(2i)} = \sigma\left(\tilde{g}_1^{(2i)} - \tilde{f}_1^{(2i-1)}\right) = \sigma\left(\tilde{g}_1^{(2i)} - \sigma\left(\tilde{g}_1^{(2i-1)}\right)\right)$ produces at most two hyperplanes in the input space $\mathbb{R}^n$. Therefore, in total, the w_1 neurons in the first layer produces $(1+2) \cdot \frac{w_1}{2} = \frac{3w_1}{2}$ hyperplanes in the input space $\mathbb{R}^n$. Then by Zaslavsky's Theorem, it will produce at most $\sum_{j=0}^{n} \binom{w_1+1}{j}$ linear regions in the input space $\mathbb{R}^n$. For the induction step, we assume that for some $k \geq 1$, any $\mathbb{R}^n \to \mathbb{R}$ ReLU DNN with every two neurons linked in each hidden layer, depth $k+1$ and widths $w_1, \ldots, w_k$ of k hidden layers produces at most $\prod_{i=1}^{k} \sum_{j=0}^{n} \binom{\frac{3w_i}{2}+1}{j}$ linear regions. Now we consider any $\mathbb{R}^n \to \mathbb{R}$ ReLU DNN with every two neurons linked in each hidden layer, depth $k+2$ and widths $w_1, \ldots, w_{k+1}$ of $k+1$ hidden layers. Then for each linear region S produced by the first $k+1$ layers, again, $\tilde{f}_{k+1}^{(2i-1)} = \sigma\left(\tilde{g}_{k+1}^{(2i-1)}\right)$ produces one hyperplane in S. Furthermore, $\tilde{f}_{k+1}^{(2i)} = \sigma\left(\tilde{g}_{k+1}^{(2i)} - \tilde{f}_{k+1}^{(2i-1)}\right) = \sigma\left(\tilde{g}_{k+1}^{(2i)} - \sigma\left(\tilde{g}_{k+1}^{(2i-1)}\right)\right)$ produces at most two hyperplanes in the S. Therefore, in total, the w_{k+1} neurons in the $k+1$ layer produces $(1+2) \cdot \frac{w_{k+1}}{2} = \frac{3w_{k+1}}{2}$ hyperplanes in S. Then by Zaslavsky's Theorem, it will produce at most $\sum_{j=0}^{n} \binom{w_{k+1}+1}{j}$ linear regions in S. Thus f has at most $\prod_{i=1}^{k} \sum_{j=0}^{n} \binom{\frac{3w_i}{2}+1}{j}$ linear regions. $\square$

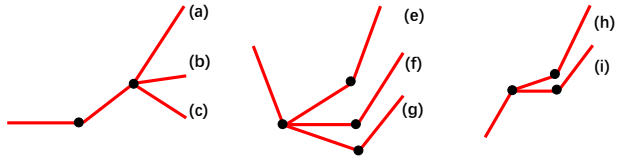

Figure 3: Exhaustion of all possible shapes of $g_2^{(i)}$,$i = 1, 2$ in Lemma 4.

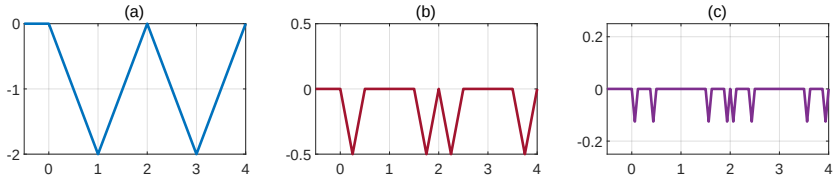

Figure 4: Construction in Proposition : (a) $-f_1^{(2)}$, (b) $-\frac{1}{4}f_2^{(1)} + f_2^{(2)}$, (c) $-\frac{1}{4}f_3^{(1)} + f_3^{(2)}$.

B Proof of Proposition 4

Proof. Obviously, $g_2^{(i)}$, $i = 1, 2$, has at most 3 pieces. If some $g_2^{(i)}$ has 3 pieces, then by exhaustion , we know either it has a 0-slope, or it has two adjacent pieces with slopes of the same sign (see Figure **??**). Hence, $f_2^{(i)}$, $i = 1, 2$, has at most 2 newly produced breakpoints. Then the output of the 2-nd layer has at most $2 + 2 \times 2 = 6$ breakpoints, i.e., 7 pieces. Applying the similar method to each piece, we can get our result via a simple induction step.

For convenience, we say a $\mathbb{R} \to \mathbb{R}$ PWL function f is of "triangle-trapezoid-triangle" shape on $[a, b] \subset \mathbb{R}$, if there exists a partition of $[a, b] : a < x_1 < x_2 < \cdots < x_6 < b$ and a positive constant c, such that

$$f|_{[a,b]}(x) = \begin{cases} c & \text{, if } x = a, x_2, x_5, b \\ -c & \text{, if } x = x_1, x_6 \\ -3c & \text{, if } x \in [x_3, x_4] \\ \text{linear connection} & \text{, otherwise.} \end{cases}$$

Let $f : \mathbb{R} \to \mathbb{R}$ be a PWL function of "triangle-trapezoid-triangle" shape on $[a, b] \subset \mathbb{R}$. Assume the corresponding partition of $[a, b]$ is $a < x_1 < x_2 < \cdots < x_6 < b$, and $f(a) = c > 0$. Set $g^{(1)} = 4f, g^{(2)} = 2f - \frac{3c}{2}, f^{(1)} = \sigma\left(g^{(1)}\right), f^{(2)} = \sigma\left(g^{(2)} - f^{(1)}\right)$ and $g = -\frac{1}{4}f^{(1)} + f^{(2)} + \frac{c}{8}$. Then g is of "triangle-trapezoid-triangle" shape on $[a, x_2], [x_2, x_5]$ *and* $[x_5, b]$ respectively.

Set $f_1^{(1)} = \sigma(2x), f_1^{(2)} = \sigma\left(x - f_1^{(1)} + 1\right)$. Set the output of the first layer to be $g_2^{(1)} = -4f_1^{(2)} + 2$ and $g_2^{(2)} = -2f_1^{(2)} + \frac{3}{2}$, which gives $f_2^{(1)} = \sigma\left(g_2^{(1)}\right)$ and $f_2^{(2)} = \sigma\left(g_2^{(2)} - f_2^{(1)}\right)$ Then through a direct calculus, $-\frac{1}{4}f_2^{(1)} + f_2^{(2)} - \frac{3}{8}$ is of "triangle-trapezoid-triangle" shape on $[-1, 1]$. repeatedly, we can construct a PWL function represented by a $\mathbb{R} \to \mathbb{R}$ $(k + 1)$-layer, 2-wide ReLU DNN, with every two neurons linked in each hidden layer, which is constant on $(-\infty, -1] \cup [1, \infty)$ and has 3^{k-2} "triangle-trapezoid-triangle" shapes on $[-1, 1]$. □

C Supplementary Results for the Tightness of Bounds

Proposition 5 (The bound $\prod_{i=1}^{k}(w_i + 1)$ is tight for a width-bounded but depth-unbounded network)**.** *Given an $\mathbb{R} \to \mathbb{R}$ ReLU network with width w for the first layer and 3 for the other layers, for any depth $k \geq 2$, there exists a piecewise linear function represented by such a network, whose number of pieces is $(w + 1) \cdot 4^{k-1}$.*

Proof. Let $f_1^{(1)}, \dots f_1^{(w)}$ be the same as in Proposition 1. Let

$$\tilde{g}_2 = \begin{cases} f_1^{(1)} + f_1^{(2)} - f_1^{(3)} - 3, & \text{if } w = 3, \\ f_1^{(1)} + f_1^{(2)} - f_1^{(3)} + \sum_{j=4}^{w} (-1)^{j-1} f_1^{(j)} - 3, & \text{if } w > 3. \end{cases}$$

We set

$$\begin{aligned} f_2^{(1)} &= \sigma\left(2\tilde{g}_2 - \tfrac{1}{3}\right), \\ f_2^{(2)} &= \sigma\left(-\tilde{g}_2 + \tfrac{2}{3}\right), \\ f_2^{(3)} &= \sigma\left(\tfrac{3}{2}\tilde{g}_2 - \tfrac{1}{2}\right). \end{aligned}$$

Now we continue our proof by induction. Assume we have constructed $f_i^{(1)}$, $f_i^{(2)}$ and $f_i^{(3)}$, $i \geq 2$. Then we set

$$\tilde{g}_{i+1} = f_i^{(1)} + f_i^{(2)} - f_i^{(3)} - \frac{3}{6^i}$$

and

$$\begin{aligned} f_{i+1}^{(1)} &= \sigma\left(2\tilde{g}_{i+1} - \tfrac{2}{6^i}\right), \\ f_{i+1}^{(2)} &= \sigma\left(-\tilde{g}_{i+1} - \tfrac{4}{6^i}\right), \\ f_{i+1}^{(3)} &= \sigma\left(\tilde{g}_{i+1} + \tfrac{3}{6^i}\right). \end{aligned}$$

Through a direct calculus, we know $\tilde{g}_{i+1}$ has $(w+1) \cdot 4^{i-1}$ pieces with opposite slope in every two adjoint pieces and ranges from 0 to $3/6^i$ in each piece except the leftmost and rightmost piece, which implied we can totally obtain $(w+1) \cdot 4^{k-1}$ pieces. □