UNIVERSAL APPROXIMATION WITH CERTIFIED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

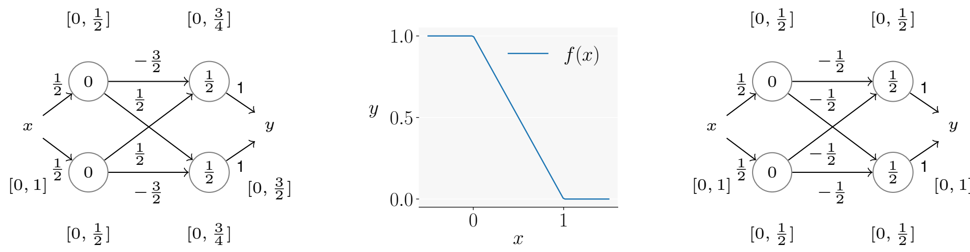
Training neural networks to be certifiably robust is a powerful defense against adversarial attacks. However, while promising, state-of-the-art results with certified training are far from satisfactory. Currently, it is very difficult to train a neural network that is both accurate and certified on realistic datasets and specifications (e.g., robustness). Given this difficulty, a pressing existential question is: given a dataset and a specification, is there a network that is both certified and accurate with respect to these? While the evidence suggests "no", we prove that for realistic datasets and specifications, such a network does exist and its certification can be established by propagating lower and upper bounds of each neuron through the network (interval analysis) – the most relaxed yet computationally efficient convex relaxation. Our result can be seen as a Universal Approximation Theorem for interval-certified ReLU networks. To the best of our knowledge, this is the first work to prove the existence of accurate, interval-certified networks.

1 INTRODUCTION

Much recent work has shown that neural networks can be fooled into misclassifying adversarial examples Szegedy et al. (2013), inputs which are imperceptibly different from images that the neural network classifies correctly. Initial work on defending against adversarial examples revolved around training networks to be empirically more robust, usually by including adversarial examples found via attacks into the training dataset (Gu and Rigazio, 2014; Papernot et al., 2016; Zheng et al., 2016; Athalye and Sutskever, 2017; Evtimov et al., 2017; Moosavi-Dezfooli et al., 2017; Xiao et al., 2018a). However, while experimental robustness can be practically useful, it does not provide safety guarantees. As a result, much recent research has focused on establishing such guarantees by verifying that a network is certifiably robust, typically by employing methods based on mixed integer linear programming, SMT solvers and bound propagation (Gehr et al., 2018; Katz et al., 2017; Singh et al., 2018; Tjeng et al., 2017).

Because the certification rates where far from satisfactory, specific training methods have been recently developed which aim to produce networks that are certifiably robust (Mirman et al., 2018; Wang et al., 2018; Wong and Kolter, 2018; Wong et al., 2018). Fundamentally, these methods work by training the network with standard optimization applied to an over-approximation of the network behavior on a given input region (the region is created around the concrete input point). That is, at a high level, these techniques aim to discover specific weights which facilitate verification. Naturally, the more precise the over-approximation used, the slower the training and certification are.

So far, some of the best results achieved on the popular MNIST (LeCun et al., 1998) and CIFAR10 (Krizhevsky, 2009) datasets have been obtained with the simple Interval approximation (Gowal et al., 2018; Mirman et al., 2019), which scales well at both training and verification time. However, despite this progress, there are still substantial gaps between known standard accuracy, experimental robustness, and certifiably robustness. For example, for CIFAR10, the best reported certifiably robustness is 32.04% with an accuracy of 49.49% when using a fairly modest 8/255 sized l_∞ region (Gowal et al., 2018). This is despite the fact that the state-of-the-art non-robust accuracy for this dataset is $> 95\%$ and the experimental robustness is $> 50\%$. Given the size of this gap, a key question then is: *can certified training ever succeed or is there a fundamental limit*?



(a) Not certifiable network n_1. (b) The function f. (c) Certifiable network n_2.

Figure 2: The ReLU networks n_1 (Figure 2a) and n_2 (Figure 2c) encode the same function f (Figure 2b). Interval analysis fails certify that n_1 does not exceed $[0, 1]$ on $[0, 1]$ while certification succeeds for n_2.

In this paper we take a step in answering this question by proving a result parallel to the Universal Approximation Theorem (Cybenko, 1989; Hornik et al., 1989). We prove that for any continuous function f defined on a compact domain $\Gamma \subseteq \mathbb{R}^m$ and for any desired level of accuracy δ, there exists a ReLU neural network n which can certifiably approximate f up to δ using interval bound propagation. As an interval is a fairly imprecise relaxation, our result directly applies to more precise convex relaxations (e.g., Singh et al. (2019); Weng et al. (2018)).

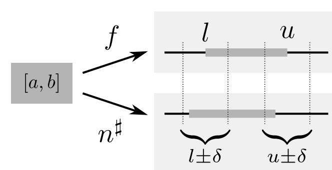


Figure 1: Illustration of Theorem 1.1.

Theorem 1.1 (Universal Interval-Certified Approximation, Figure 1)**.** Let $\Gamma \subset \mathbb{R}^m$ be a compact set and let $f : \Gamma \to \mathbb{R}$ be a continuous function. For all $\delta > 0$, there exists a ReLU network n such that for all boxes $[a, b]$ in Γ defined by points $a, b \in \Gamma$ where $a_k \leq b_k$ for all k, the propagation of the box $[a, b]$ using interval analysis through the network n, denoted $n^\sharp([a, b])$, approximates the set $[l, u] = [\min f(B), \max f(B)] \subseteq \mathbb{R}$ up to δ,

$$[l + \delta, u - \delta] \subseteq n^\sharp([a, b]) \subseteq [l - \delta, u + \delta]. \tag{1}$$

We recover the classical universal approximation theorem ($|f(x) - n(x)| \leq \delta$ for all $x \in \Gamma$) by considering boxes $[a, b]$ describing points ($x = a = b$).

Applicability of Theorem Because interval analysis propagates boxes, the theorem naturally handles l_∞ norm bound perturbations to the input. Other l_p norms can be handled by covering the l_p ball with boxes. The theorem can be extended easily to functions $f : \Gamma \to \mathbb{R}^k$ by applying the theorem component wise. We can apply our theorem to the problem of certified adversarial robustness by instantiating the function f with the ground truth classifier of a dataset such as CIFAR10 (assuming f is continuous). Then, our result guarantees that there exists a ReLU network n such that local stability around the input x can be checked efficiently using interval analysis for all x in the data distribution.

We note that we do not provide a method for training a certified ReLU network – we aim to answer an existential question and thus we focus on proving that a given network exists. Interesting future work items would be to study the requirements on the size of this network and the inherent hardness of finding it with standard optimization methods.

Universal Approximation is insufficient We now discuss why classical universal approximation is insufficient for establishing our result. While classical universal approximation theorems state that neural networks can approximate a large class of functions f, unlike our result, they do not state that a property (e.g., local robustness) of the approximation n of f is actually certified with a scalable proof method (e.g., interval bound propagation). If one uses a (non scalable) complete verifier instead, then the standard Universal approximation theorem is sufficient.

To demonstrate this point, consider the function $f : \mathbb{R} \to \mathbb{R}$ (Figure 2b) mapping all $x \leq 0$ to 1, all $x \geq 1$ to 0 and all $0 < x < 1$ to $1 - x$ and two ReLU networks n_1 (Figure 2a) and n_2 (Figure 2c) perfectly approximating f, that is $n_1(x) = f(x) = n_2(x)$ for all x. For $\delta = \frac{1}{4}$, the interval certification that n_1 maps all $x \in [0,1]$ to $[0,1]$ fails because $[\frac{1}{4}, \frac{3}{4}] \subseteq n_1^\sharp([0,1]) = [0, \frac{3}{2}] \not\subseteq [-\frac{1}{4}, \frac{5}{4}]$. However, interval certification succeeds for n_2, because $n_2^\sharp([0,1]) = [0,1]$. To the best of our knowledge, this is the first work to prove the existence of accurate, interval-certified networks.

2 Related work

After adversarial examples were discovered by Szegedy et al. (2013), many attacks and defenses have been introduced (for a survey, see Akhtar and Mian (2018)). Initial work on verifying neural network robustness used exact methods (Katz et al., 2017) on small networks, while later research introduced methods based on over-approximation (Tjeng et al., 2017; Singh et al., 2018; Gehr et al., 2018; Tjeng et al., 2017) aiming to scale to larger networks. As neural networks that are experimentally robust need not be certifiably robust, there has been significant recent research on training certifiably robust neural networks (Mirman et al., 2018; 2019; Wong and Kolter, 2018; Wong et al., 2018; Wang et al., 2018; Gowal et al., 2018; Dvijotham et al., 2018; Xiao et al., 2018b). As these methods appear to have reached a performance wall, several works have started investigating the fundamental barriers in the datasets and methods that preclude the learning of a robust network (let alone a certifiably robust one) (Khoury and Hadfield-Menell, 2018; Schmidt et al., 2018; Tsipras et al., 2018).

In our work, we focus on the question of whether neural networks are capable of approximating functions in a interval certifiably robust manner. Gehr et al. (2018) presented the first system of using abstract interpretation to analyze neural networks, and Mirman et al. (2018) leveraged the method for training. Gowal et al. (2018) also found that rather imprecise interval analysis could be used to train large, certifiably robust neural networks.

Feasibility Results with Neural Networks Historically, first results to show that a limited function set could be used to approximate another more general function set came from Weierstrass (1885). This work showed that any continuous real valued function defined on an interval could be approximated to arbitrary degree by a polynomial.

Early results on the expressiveness of neural networks, by Cybenko (1989) and Hornik et al. (1989), stated the first versions of what is now known as the Universal Approximation Theorem. Cybenko (1989) showed that networks using sigmoid activations could approximate continuous functions in the unit hypercube, while Hornik et al. (1989) showed that even networks with only one hidden layer are capable of approximating Borel measurable functions (a set larger than the set of continuous functions).

More recent work has investigated the capabilities of ReLU networks. Arora et al. (2018), based on Tarela and Martínez (1999), proved that every continuous piecewise linear function in $\mathbb{R}^m$ can be represented by a ReLU network and He et al. (2018) reduce the number of neurons needed using techniques from finite elements methods. Relevant to our work, Arora et al. (2018) introduced a ReLU network representations of the $\min$ function. Further, we use a construction method that is similar to the one for nodal basis functions given in He et al. (2018).

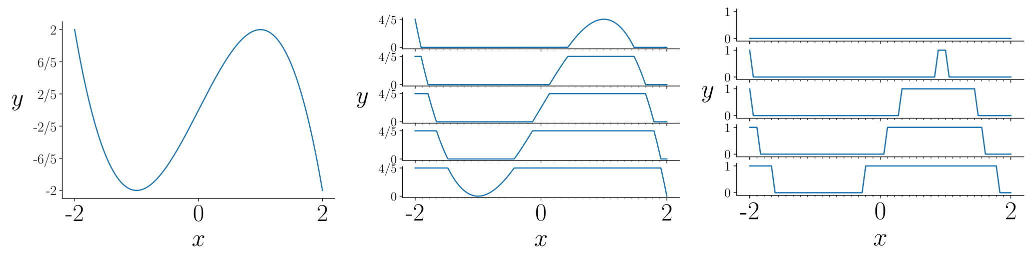
3 Background

In this section we provide the concepts necessary to describe our main result.

Adversarial Examples and Robustness Verification Let $n : \mathbb{R}^m \to \mathbb{R}^k$ be a neural network, which classifies an input x to a label t if $n(x)_t > n(x)_j$ for all $j \neq t$. For a correctly classified input x, an adversarial example is an input y such that x is imperceptible from y to a human, but is classified to a different label by n.

Frequently, two images are assumed to be "imperceptible" if there l_p distance is at most ϵ. The l_p ball around an image is said to be the adversarial ball, and a network is said to be ϵ-robust around x if every point in the adversarial ball around x classifies the same. In this paper, we limit our discussion to l_∞ adversarial balls which can be used to cover to all l_p balls.

The goal of robustness verification is to show that for a neural network n, input point x and label t, every possible input in an l_∞ ball of size ϵ around x (written $\mathbb{B}_\epsilon^\infty(x)$) is also classified to t.



(a) f (b) Slicing of f, $f_0, \dots, f_4$ (c) Networks n_k approximating f_k.

Figure 3: Approximating f (Figure 3a) using a ReLU network $n = \xi_0 + \sum_k n_k$. The ReLU networks n_k (Figure 3c) approximate the N-slicing of f (Figure 3b), as a sum of local bumps (Figure 6).

Verifying neural networks with Interval Analysis The verification technique we investigate in this work is interval analysis. We denote by $\mathcal{B}$ the set of boxes $B = [a, b] \subset \mathbb{R}^m$ for all m, where $a_i \leq b_i$ for all i. Further for $\Gamma \subseteq \mathbb{R}^m$ we define $\mathcal{B}(\Gamma) := \mathcal{B} \cap \Gamma$ describing all the boxes in Γ. The standard interval-transformations for the basic operations we are considering, namely $+, -, \cdot$ and the ReLU function R (Gehr et al. (2018), Gowal et al. (2018)) are

$$[a, b] +^\sharp [c, d] = [a + c, b + d] \qquad -^\sharp [a, b] = [-b, -a]$$
$$R^\sharp([a, b]) = [R(a), R(b)] \qquad \lambda \cdot^\sharp [a, b] = [\lambda a, \lambda b],$$

where $[a, b], [c, d] \in \mathcal{B}(\mathbb{R})$, and $\lambda \in \mathbb{R}_{\geq 0}$. Further, we used $\sharp$ to distinguish the function f from its interval-transformation $f^\sharp$. To illustrate the difference between f and $f^\sharp$, consider $f(x) := x - x$ evaluated on $x = [0, 1]$. We have $f([0, 1]) = 0$, but $f^\sharp([0, 1]) = [0, 1] -^\# [0, 1] = [0, 1] +^\# [-1, 0] = [-1, 1]$ illustrating the loss in precision that interval analysis suffers from.

Interval analysis provides a sound over-approximation in the sense that for all function f, the values that f can obtain on $[a, b]$, namely $f([a, b]) := \{f(x) \mid x \in [a, b]\}$ are a subset of $f^\sharp([a, b])$. If f is a composition of functions, $f = f_1 \circ \dots \circ f_k$, then $f_1^\sharp \circ \dots \circ f_k^\sharp$ is a sound interval-transformer for f.

Further all combinations f of $+, -, \cdot$ and R are monotone, that is for $[a, b], [c, d] \subseteq \mathcal{B}(\mathbb{R}^m)$ such that $[a, b] \subseteq [c, d]$ then $f^\#([a, b]) \subseteq f^\#([c, d])$ (Appendix A). For boxes $[x, x]$ representing points $f^\sharp$ coincides with f, $f^\sharp([x, x]) = f(x)$. This will later be needed.

4 Proving Universal Interval-Provable Approximation

In this section, we provide an explanation of the proof of our main result, Theorem 4.6, and illustrate the main points of the proof. The full proof can be found in Appendix A.

The first step in the construction is to deconstruct the function f into slices $\{f_k : \Gamma \to [0, \frac{\delta}{2}]\}_{0 \leq k < N}$ such that that $f(x) = \xi_0 + \sum_{k=0}^{N-1} f_k(x)$ for all x, where ξ_0 is the minimum of $f(\Gamma)$. We approximate each slice f_k by a ReLU network $\frac{\delta}{2} \cdot n_k$. The network n approximating f up to δ will be $n(x) := \xi_0 + \frac{\delta}{2} \sum_k n_k(x)$. The construction relies on 2 key insights, (i) the output of $\frac{\delta}{2} \cdot n_k^\sharp$ can be confined to the interval $[0, \frac{\delta}{2}]$, thus the loss of analysis precision is at most the height of the slice, and (ii) we can construct the networks n_k using local bump functions, such that only 4 slices can contribute to the loss of analysis precision, two for the lower interval bound, two for the upper one.

The slicing $\{f_k\}_{0 \leq k < 5}$ of the function $f : [-2, 2] \to \mathbb{R}$ (Figure 3a), mapping x to $f(x) = -x^3 + 3x$ is depicted in Figure 3b. The networks n_k are depicted in Figure 3c. In this example, evaluating the interval-transformer of n, namely $n^\sharp$ on the box $B = [-1, 1]$ results into $n^\sharp([-1, 1]) = [-2, 6/5]$ lies is within the $\delta = \frac{8}{5}$ bound of $f([-1, 1]) = [-2, 2]$.

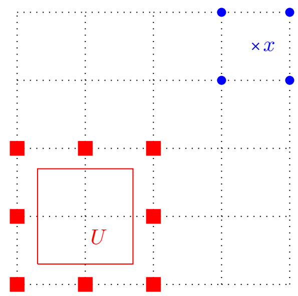


Figure 4: Neighbors $\mathcal{N}(x)$ (blue dots) and $\mathcal{N}(U)$ (red squares).

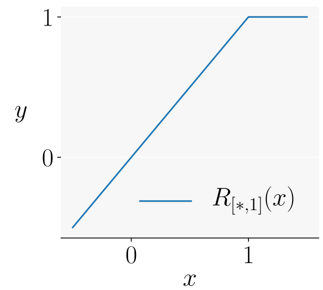


Figure 5: $R_{[*,b]}(x)$

Definition 4.1 (N-slicing (Figure 3b)). Let $\Gamma \subset \mathbb{R}^m$ be a closed m-dimensional box and let $f\colon \Gamma \to \mathbb{R}$ be continuous. The N*-slicing* of f is a set of functions $\{f_k\}_{0\le k<N}$ defined by

$$f_k\colon \Gamma \to \mathbb{R}, \quad x \mapsto \begin{cases} 0 & \text{if } f(x) \le \xi_k, \\ f(x) - \xi_k & \text{if } \xi_k < f(x) < \xi_{k+1}, \\ \xi_{k+1} - \xi_k & \text{if } \xi_{k+1} \le f(x), \end{cases} \qquad \forall k \in \{0, \dots, N-1\},$$

where $\xi_k := \xi_0 + \frac{k}{N}(\xi_N - \xi_0)$, $k \in \{1, \dots, N-1\}$, $\xi_0 := \min f(\Gamma)$ and $\xi_N := \max f(\Gamma)$.

To construct a ReLU network satisfying the desired approximation property (Equation (1)) if evaluated on boxes in $\mathcal{B}(\Gamma)$, we need the ReLU network nmin capturing the behavior of $\min$ as a building block (similar to He et al. (2018)). It is given by

$$\text{nmin}(x, y) := \frac{1}{2}\begin{pmatrix}1 & -1 & -1 & -1\end{pmatrix} R\left(\begin{pmatrix}1 & 1\\ -1 & -1\\ 1 & -1\\ -1 & 1\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}\right).$$

With the ReLU network nmin, we can construct recursively a ReLU network nmin_N mapping N arguments to the smallest one (Definition A.8). Even though the interval-transformation loses precision, we can establish bounds on the precision loss of $\text{nmin}_N^\sharp$ sufficient for our use case (Appendix A).

Now, we use the clipping function $R_{[*,1]} := 1 - R(1-x)$ clipping every value exceeding 1 back to 1 (Figure 5) to construct the local bumps ϕ_c w.r.t. a grid G. G specifies the set of all possible local bumps we can use to construct the networks n_k. Increasing the fines of G will increases the approximation precision.

Definition 4.2 (local bump, Figure 6). Let $M \in \mathbb{N}$, $G := \{(\frac{i_1}{M}), \dots, \frac{i_m}{M} \mid i \in \mathbb{Z}^m\}$ be a grid, $\ell = 2^{\lceil \log_2 2m \rceil + 1}$ and let $c = \{\frac{i_1^l}{M}, \frac{i_1^u}{M}\} \times \dots \times \{\frac{i_m^l}{M}, \frac{i_m^u}{M}\} \subseteq G$ be a set of grid points describing the corner points of a hyperrectangle in G. We define a ReLU neural network $\phi_c\colon \mathbb{R}^m \to [0,1] \subset \mathbb{R}$ w.r.t. G by

$$\phi_c(x) := R\left(\text{nmin}_{2m} \bigcup_{1\le k\le m} \left\{ \begin{matrix} R_{[*,1]}(M\cdot \ell \cdot (x_k - \frac{i_k^l}{M}) + 1), \\ R_{[*,1]}(M \cdot \ell \cdot (\frac{i_k^u}{M} - x_k) + 1) \end{matrix} \right\}\right).$$

We will describe later how M and c get picked. A graphical illustration of a local bump for in two dimensions and $c = \{\frac{i_1^l}{M}, \frac{i_1^u}{M}\} \times \{\frac{i_2^l}{M}, \frac{i_2^u}{M}\} = \{c^{ll}, c^{lu}, c^{ul}, c^{uu}\}$ is shown in Figure 6. The local bump $\phi_c(x)$ evaluates to 1 for all x that lie within the convex hull of c, namely $\text{conv}(c)$, after which $\phi_c(x)$ quickly decreases linearly to 0.

By construction $\phi_c(x)$ decreases to 0 before reaching the next neighboring grid points $\mathcal{N}(\text{conv}(c))$, where $\mathcal{N}(x) := \{g \in G \mid ||x - g||_\infty \le \frac{1}{M}\} \setminus \{x\}$ denotes the neighboring grid points of x and similarly for $\mathcal{N}(U) := \{\mathcal{N}(x) \mid x \in U\} \setminus U$ (Figure 4). The set $\mathcal{N}(\text{conv}(c))$ forms a hyperrectangle in G and is shown in Figure 6 using red squares. Clearly $\text{conv}(c) \subseteq \text{conv}(\mathcal{N}(c))$.

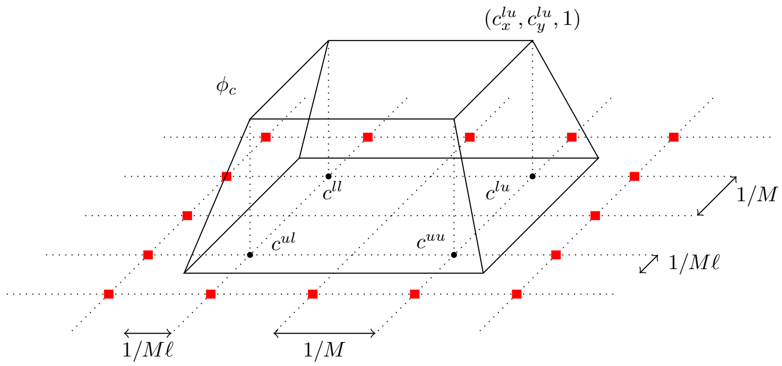


Figure 6: Local bump ϕ_c, where c contains the points $c^{ll}, c^{lu}, c^{ul}, c^{uu}\}$. The points in $\mathcal{N}(\text{conv}(c))$ are depicted by the red squares.

Next, we give bounds on the loss of precision for the interval-transformation $\phi_c^\sharp$. We can show that interval analysis can (i) never produce intervals exceeding $[0, 1]$ and (ii) is precise if B does no intersect $\text{conv}((N)(c)) \setminus \text{conv}(c)$.

Lemma 4.3. For all $B \in \mathcal{B}(\mathbb{R}^m)$, it holds that $\phi_c^\sharp(B) \subseteq [0, 1] \in \mathcal{B}$ and

$$\phi_c^\sharp(B) = \begin{cases} [1, 1] & \text{if } B \subseteq \text{conv}(c) \\ [0, 0] & \text{if } B \subseteq \Gamma \setminus \text{conv}(\mathcal{N}(c)). \end{cases}$$

The formal proof is given in Appendix A. The next lemma shows, how a ReLU network n_k can approximate the slice f_k while simultaneously confining the loss of analysis precision.

Lemma 4.4. Let $\Gamma \subset \mathbb{R}^m$ be a closed box and let $f : \Gamma \to \mathbb{R}$ be continuous. For all $\delta > 0$ there exists a set of ReLU networks $\{n_k\}_{0 \le k < N}$ of size $N \in \mathbb{N}$ approximating the N-slicing of f, $\{f_k\}_{0 \le k < N}$ (ξ_k as in Definition 4.1) such that for all boxes $B \in \mathcal{B}(\Gamma)$

$$n_k^\sharp(B) = \begin{cases} [0, 0] & \text{if } f(B) \le \xi_k - \frac{\delta}{2} \\ [1, 1] & \text{if } f(B) \ge \xi_{k+1} + \frac{\delta}{2}. \end{cases} \tag{2}$$

and $n_k^\sharp(B) \subseteq [0, 1]$.

It is important to note that in Equation (2) we mean f and not $f^\sharp$. The proof for Lemma 4.4 is given in Appendix A. In the following, we discuss a proof sketch.

Because Γ is compact and f is continuous, f is uniformly continuous by the Heine-Cantor Theorem. So we can pick a $M \in \mathbb{N}$ such that for all $x, y \in \Gamma$ satisfying $||y - x||_\infty \le \frac{1}{M}$ holds $|f(y) - f(x)| \le \frac{\delta}{2}$. We then choose the grid $G = (\frac{\mathbb{Z}}{M})^m \subseteq \mathbb{R}^m$.

Next, we construct for every slice k a set Δ_k of hyperrectangles on the grid G: if a box $B \in \mathcal{B}(\Gamma)$ fulfills $f(B) \ge \xi_{k+1} + \frac{\delta}{2}$, then we add a minimal enclosing hyperrectangle $c \subset G$ such that $B \subseteq \text{conv}(c)$ to Δ_k, where $\text{conv}(c)$ denotes the convex hull of c. This implies, using uniform continuity of f and that the grid G is fine enough, that $f(\text{conv}(c)) \ge \xi_{k+1}$. Since there is only a finite number of possible hyperrectangles in G, the set Δ_k is clearly finite. The network fulfilling Equation (2) is

$$n_k(x) := R_{[*,1]}\left(\sum_{c \in \Delta_k} \phi_c(x)\right),$$

where ϕ_c is as in Definition 4.2. The n_k are depicted in Figure 3c.

Now, we see that Equation (2) holds by construction: For all boxes $B \in \mathcal{B}(\Gamma)$ such that $f \ge \xi_{k+1} + \frac{\delta}{2}$ on B exists $c' \in \Delta_k$ such that $B \subseteq \text{conv}(c')$ which implies, using Lemma 4.3, that $\phi_{c'}^\sharp(B) = [1, 1]$, hence

$$\begin{aligned} n_k^\sharp(B) &= R^\sharp_{[*,1]}(\phi^\sharp_{c'}(B) + \sum_{c \in \Delta_k \setminus c'} \phi^\sharp_c(B)) && \forall c \neq c' : \phi^\sharp_c(B) \subseteq [0,1] \text{(Lemma 4.3)} \\ &= R^\sharp_{[*,1]}([1,1] + [p_1, p_2]) && [p_1, p_1] \in \mathcal{B}(\mathbb{R}_{\geq 0}) \\ &= R^\sharp_{[*,1]}([1 + p_1, 1 + p_2]) \\ &= [1,1]. \end{aligned}$$

Similarly, if $f(B) \leq \xi_k - \frac{\delta}{2}$ holds, then it holds for all $c \in \Delta_k$ that B does not intersect $\mathcal{N}(\text{conv}(c))$. Indeed, if a $c \in \Delta_k$ would violate this, then by construction, $f(\text{conv}(c)) \geq \xi_{k+1}$, contradicting $f(B) \leq \xi_k - \frac{\delta}{2}$. Thus $\phi^\sharp_c(B) = [0,0]$, and hence $n^\sharp(B) = [0,0]$.

Theorem 4.5. Let $\Gamma \subset \mathbb{R}^m$ be a closed box and let $f \colon \Gamma \to \mathbb{R}$ be continuous. Then for all $\delta > 0$, exists a ReLU network n such that for all $B \in \mathcal{B}(\Gamma)$

$$[l + \delta, u - \delta] \subseteq n^\sharp(B) \subseteq [l - \delta, u + \delta],$$

where $l := \min f(B)$ and $u := \max f(B)$.

Proof. Pick N such that the height of each slice is exactly $\frac{\delta}{2}$, if this is impossible choose a slightly smaller δ. Let $\{n_k\}_{0 \leq k < N}$ be a series of networks as in Lemma 4.4. Recall that $\xi_0 = \min f(\Gamma)$. We define the ReLU network

$$n(x) := \xi_0 + \frac{\delta}{2} \sum_{k=0}^{N-1} n_k(x). \tag{3}$$

Let $B \in \mathcal{B}(\Gamma)$. Thus we have for all k

$$f(B) \geq \xi_{k+2} \Leftrightarrow f(B) \geq \xi_{k+1} + \frac{\delta}{2} \overset{Lemma\ 4.4}{\Rightarrow} n_k^\sharp(B) = [1,1] \tag{4}$$

$$f(B) \leq \xi_{k-1} \Leftrightarrow f(B) \leq \xi_k - \frac{\delta}{2} \overset{Lemma\ 4.4}{\Rightarrow} n_k^\sharp(B) = [0,0]. \tag{5}$$

Let $p, q \in \{0, \dots, N-1\}$ such that

$$\xi_p \leq l = \min f(B) \leq \xi_{p+1} \tag{6}$$

$$\xi_q \leq u = \max f(B) \leq \xi_{q+1}, \tag{7}$$

as depicted in Figure 7. Thus by Equation (4) for all $k \in \{0, \dots, p-2\}$ it holds that $n_k^\sharp(B) = [1,1]$ and similarly, by Equation (5) for all $k \in \{q+2, \dots, N-1\}$ it holds that $n_k^\sharp(B) = [0,0]$. Plugging this into Equation (3) after splitting the sum into three parts leaves us with

$$\begin{aligned} n^\sharp(B) &= \xi_0 + \frac{\delta}{2} \sum_{k=0}^{p-2} n_k^\sharp(B) + \frac{\delta}{2} \sum_{k=p-1}^{q+1} n_k^\sharp(B) + \frac{\delta}{2} \sum_{k=p+1}^{N-1} n_k^\sharp(B) \\ &= \xi_0 + (p-1)[\tfrac{\delta}{2}, \tfrac{\delta}{2}] + \frac{\delta}{2} \sum_{k=p-1}^{q+1} n_k^\sharp(B) + [0,0]. \end{aligned}$$

Applying the standard rules for interval analysis, leads to

$$n^\sharp(B) = [\xi_{p-1}, \xi_{p-1}] + \frac{\delta}{2} \sum_{k=p-1}^{q+1} n_k^\sharp(B),$$

where we used in the last step, that $\xi_0 + k\frac{\delta}{2} = \xi_k$. For all terms in the sum except the terms corresponding to the 3 highest and lowest k we get

$$n_k^\sharp(B) = [0,1] \qquad \forall k \in \{p+2, \dots, q-2\}. \tag{8}$$

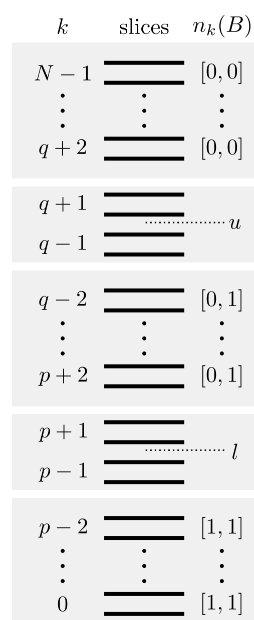


Figure 7: Illustration of the proof for Theorem 4.5.

Indeed, from Equation (6) we know that there is $x \in B$ such that $f(x) \leq \xi_{p+1} = \xi_{p+2} - \frac{\delta}{2}$, thus by Lemma 4.4 $n_k^\sharp([x,x]) = [0,0]$ for all $p+2 \leq k \leq q-2$. Similarly, from Equation (7) we know, that there is $x' \in B$ such that $f(x) \geq \xi_q = \xi_{q-1} + \frac{\delta}{2}$, thus by Lemma 4.4 $n_k^\sharp([x',x']) = [1,1]$ for all $p+2 \leq k \leq q-2$. So $n_k^\sharp(B)$ is at least $[0,1]$, and by Lemma 4.4 also at most $[0,1]$. This leads to

$$\begin{aligned} n^\sharp(B) &= [\xi_{p-1}, \xi_{p-1}] + \frac{\delta}{2} \sum_{k=p-1}^{p+1} n_k^\sharp(B) + \frac{\delta}{2}((q-2)-(p+2)+1)[0,1] + \frac{\delta}{2} \sum_{k=q-1}^{q+1} n_k^\sharp(B) \\ &= [\xi_{p-1}, \xi_{p-1}] + \frac{\delta}{2} \sum_{k=p-1}^{p+1} n_k^\sharp(B) + [0, \xi_{q-1} - \xi_{p+2}] \qquad + \frac{\delta}{2} \sum_{k=q-1}^{q+1} n_k^\sharp(B). \end{aligned}$$

We know further, that if $p+3 \leq q$, than there is an $x \in B$ such that $f(x) \geq \xi_{p+3} = \xi_{p+2} + \frac{\delta}{2}$, hence similar as before $n_{p+1}^\sharp([x,x]) = [1,1]$ and similarly $n_p^\sharp([x,x]) = [1,1]$ and $n^\sharp([x,x]) = [1,1]$. So we know, that $\frac{\delta}{2} \sum_{k=p-1}^{p+1} n_k^\sharp(B)$ includes at least $[3\frac{\delta}{2}, 3\frac{\delta}{2}]$ and at the most $[0, 3\frac{\delta}{2}]$. Similarly, there exists an $x' \in B$ such that $n_{q-1}^\sharp([x',x']) = [0,0]$, $n_q^\sharp([x',x']) = [0,0]$ and $n_{q+1}^\sharp([x',x']) = [0,0]$. This leaves us with

$$[3\tfrac{\delta}{2}, 3\tfrac{\delta}{2}] \subseteq \frac{\delta}{2} \sum_{k=p-1}^{p+1} n_k^\sharp(B) \subseteq [0, 3\tfrac{\delta}{2}]$$

$$[0,0] \subseteq \frac{\delta}{2} \sum_{k=q-1}^{q+1} n_k^\sharp(B) \subseteq [0, 3\tfrac{\delta}{2}],$$

If $p+3 > q$ the lower bound we want to prove becomes vacuous and only the upper one needs to be proven. Thus we have

$$[l+\delta, u-\delta] \subseteq [\xi_{p+2}, \xi_{p-1}] \subseteq n^\sharp(B) \subseteq [\xi_{p-1, \xi_{q+2}}] \subseteq [l-\delta, u+\delta],$$

where $l := \min f(B)$ and $u := \max f(B)$. □

Theorem 4.6 (Universal Interval-Provable Approximation)**.** Let $\Gamma \subset \mathbb{R}^m$ be compact and $f\colon \Gamma \to \mathbb{R}^d$ be continuous. For all $\delta \in \mathbb{R}^m_{\geq 0}$ exists a ReLU network n such that for all $B \in \mathcal{B}(\Gamma)$

$$[l+\delta, u-\delta] \subseteq n^\sharp(B) \subseteq [l-\delta, u+\delta],$$

where $l, u \in \mathbb{R}^m$ such that $l_k := \min f(B)_k$ and $u_k := \max f(B)_k$ for all k.

Proof. This is a direct consequence of using Theorem 4.5 and the Tietze extension theorem to produce a neural network for each dimension d of the codomain of f. □

Note that Theorem 1.1 is a special case of Theorem 4.6 with $d = 1$ to simplify presentation.

5 Conclusion

We proved that for all real valued continuous functions f on compact sets, there exists a ReLU network n approximating f arbitrarily well with the interval abstraction. This means that for arbitrary input sets, analysis using the interval relaxation yields an over-approximation arbitrarily close to the smallest interval containing all possible outputs. Our theorem affirmatively answers the open question, whether the Universal Approximation Theorem generalizes to Interval analysis.

Our results address the question of whether the interval abstraction is expressive enough to analyse networks approximating interesting functions f. This is of practical importance because interval analysis is the most scalable non-trivial analysis.

References

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1J_rgWRW.

Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018.

Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 2017.

Timon Gehr, Matthew Mirman, Petar Tsankov, Dana Drachsler Cohen, Martin Vechev, and Swarat Chaudhuri. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *Symposium on Security and Privacy (SP)*, 2018.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. ReLU Deep Neural Networks and Linear Finite Elements. *arXiv e-prints*, art. arXiv:1807.03973, Jul 2018.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, 2017.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning (ICML)*, 2018.

Matthew Mirman, Gagandeep Singh, and Martin Vechev. A provable defense for deep residual networks. *arXiv preprint arXiv:1903.12519*, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94. Ieee, 2017.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*, pages 10825–10836, 2018.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. An abstract domain for certifying neural networks. *PACMPL*, 3(POPL):41:1–41:30, 2019. doi: 10.1145/3290354. URL https://doi.org/10.1145/3290354.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

J. M. Tarela and M. V. Martínez. Region configurations for realizability of lattice piecewise-linear models. *Math. Comput. Modelling*, 30(11-12):17–27, 1999. ISSN 0895-7177. doi: 10.1016/S0895-7177(99)00195-8. URL https://doi.org/10.1016/S0895-7177(99)00195-8.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *stat*, 1050:11, 2018.

Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of formally robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018.

Karl Weierstrass. Über die analytische darstellbarkeit sogenannter willkürlicher functionen einer reellen veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, 2:633–639, 1885.

Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane S. Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for relu networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018. URL http://proceedings.mlr.press/v80/weng18a.html.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. 2018.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *arXiv preprint arXiv:1805.12514*, 2018.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018a.

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018b.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016.

A PROOFS FOR THE UNIVERSAL INTERVAL-CERTIFIED APPROXIMATION

Lemma A.1 (Monotonicity)**.** The operations $+, -$ are monotone, that is for all $[a_1, b_1], [a_2, b_2], [c_1, d_1], [c_2, d_2] \in \mathcal{B}(R)$ such that $[a_1, b_1] \subseteq [a_2, b_2]$ and $[c_1, d_2] \subseteq [c_2, d_2]$ holds

$$[a_1, b_1] +^\sharp [c_1, d_1] \subseteq [a_2, d_2] +^\sharp [c_2, d_2]$$
$$[a_1, b_1] -^\sharp [c_1, d_1] \subseteq [a_2, d_2] -^\sharp [c_2, d_2]$$
$$[a_1, b_1] \cdot^\sharp [c_1, d_1] \subseteq [a_2, d_2] \cdot^\sharp [c_2, d_2].$$

Further the operation $*$ and R are monotone, that is for all $[a, b], [c, d] \in \mathcal{B}(R)$ and for all $\lambda \in \mathbb{R}_{\geq 0}$ such that $[a, b] \subseteq [c, d]$ holds

$$\lambda \cdot^\sharp [a, b] \subseteq \lambda \cdot^\sharp [c, d]$$
$$R^\sharp([a, b]) \subseteq R^\sharp([c, d]).$$

Proof.

$$[a_1, b_1] +^\sharp [c_1, d_1] = [a_1 + c_1, b_1 + d_1] \subseteq [a_2 + c_2, b_2 + d_2] = [a_2, d_2] +^\sharp [c_2, d_2]$$
$$[a_1, b_1] -^\sharp [c_1, d_1] = [a_1 - d_1, b_1 - c_1] \subseteq [a_2 - d_2, b_2 - c_2] = [a_2, d_2] -^\sharp [c_2, d_2]$$

$$\lambda \cdot^\sharp [a, b] = [\lambda a, \lambda b] \subseteq [\lambda c, \lambda d] = [\lambda c, \lambda d]$$
$$R^\sharp([a, b]) = [R(a), R(b)] \subseteq [R(c), R(d)] = R^\sharp([c, d]).$$

□

Definition A.2 (N-slicing)**.** Let $\Gamma \subset \mathbb{R}^m$ be a compact m-dimensional box and let $f \colon \Gamma \to \mathbb{R}$ be continuous. The N*-slicing* of f is a set of functions $\{f_k\}_{0 \leq k \leq N-1}$ defined by

$$f_k \colon \Gamma \to \mathbb{R}, \quad x \mapsto \begin{cases} 0 & \text{if } f(x) \leq \xi_k, \\ f(x) - \xi_k & \text{if } \xi_k < f(x) < \xi_{k+1}, \\ \xi_{k+1} - \xi_k & \text{otherwise,} \end{cases} \quad \forall k \in \{0, \ldots, N-1\},$$

where $\xi_k := \frac{k}{N}(\xi_{\max} - \xi_{\min})$, $k \in \{0, \ldots, N\}$, $\xi_{\min} := \min f(\Gamma)$ and $\xi_{\max} := \max f(\Gamma)$.

Lemma A.3 (N-slicing)**.** Let $\{f_k\}_{0 \leq k \leq N-1}$ be the N-slicing of f. Then for all $x \in \Gamma$ we have $f(x) := \xi_0 + \sum_{k=0}^{N-1} f_k(x)$.

Proof. Pick $x \in \Gamma$ and let $l \in \{0, \ldots, N-1\}$ such that $\xi_l \leq f(x) \leq \xi_{l+1}$. Then

$$\begin{aligned} \xi_0 + \sum_{k=0}^{N-1} f_k(x) &= \xi_0 + \sum_{k=0}^{l-1} f_k(x) + f_l(x) + \sum_{k=l+1}^{N-1} f_k(x) = \xi_0 + \sum_{k=0}^{l-1} (\xi_{k+1} - \xi_k) + f_l(x) \\ &= \xi_l + f_l(x) = f(x). \end{aligned}$$

□

Definition A.4 (clipping)**.** Let $a, b \in \mathbb{R}$, $a < b$. We define the *clipping* function $R_{[*,b]} \colon \mathbb{R} \to \mathbb{R}$ by

$$R_{[*,b]}(x) := b - R(b - x).$$

Lemma A.5 (clipping)**.** The function $R_{[*,b]}$ sends all $x \leq b$ to x, and all $x > b$ to b. Further, $R^\sharp_{[*,b]}([a', b']) = [R_{[*,b]}(a'), R_{[*,b]}(b')]$.

Proof. We show the proof for $R_{[a,b]}$, the proof for $R_{[*,b]}$ is similar.

$$x < b \Rightarrow R_{[*,b]}(x) = b - R(b - x) = b - b + x = x$$
$$x \geq b \Rightarrow R_{[*,b]}(x) = b - R(b - x) = b - 0 = b$$

Next,

$$\begin{aligned}
R^\sharp_{[*,b]}([a', b']) &= b -^\sharp R^\sharp(b -^\sharp [a', b']) \\
&= b -^\sharp R^\sharp(b +^\sharp [-b', -a']) \\
&= b -^\sharp R^\sharp([b - b', b - a']) \\
&= b -^\sharp [R(b - b'), R(b - a')] \\
&= b +^\sharp [-R(b - a'), -R(b - b')] \\
&= [b - R(b - a'), b - R(b - b')] \\
&= [R_{[*,b]}(a'), R_{[*,b]}(b')].
\end{aligned}$$

□

Definition A.6 (nmin)**.** We define the ReLU network nmin: $\mathbb{R}^2 \to \mathbb{R}$ by

$$\text{nmin}(x, y) := \frac{1}{2} \begin{pmatrix} 1 & -1 & -1 & -1 \end{pmatrix} R \left(\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right).$$

Lemma A.7 (nmin)**.** Let $x, y \in \mathbb{R}$, then $\text{nmin}(x, y) = \min(x, y)$.

Proof. Because nmin is symmetric in its arguments, we assume w.o.l.g. $x \geq y$.

$$\begin{aligned}
\text{nmin}(x, y) &= \frac{1}{2} \begin{pmatrix} 1 & -1 & -1 & -1 \end{pmatrix} R \left(\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right) \\
&= \frac{1}{2} \begin{pmatrix} 1 & -1 & -1 & -1 \end{pmatrix} R \begin{pmatrix} x + y \\ -x - y \\ x - y \\ -x + y \end{pmatrix}
\end{aligned}$$

If $x + y \geq 0$, then

$$\text{nmin}(x, y) = \frac{1}{2}(x + y - x + y) = y.$$

If $x + y < 0$, then

$$\text{nmin}(x, y) = \frac{1}{2}(x + y - x + y) = y.$$

□

Definition A.8 (nmin_N)**.** For all $N \in \mathbb{N}_{\geq 1}$, we define a ReLU network nmin_N defined by

$$\begin{aligned}
\text{nmin}_1(x) &:= x \\
\text{nmin}_N(x_1, \dots, x_N) &:= \text{nmin}(\text{nmin}_{\lceil N/2 \rceil}(x_1, \dots, x_{\lceil N/2 \rceil}), \text{nmin}_{\lceil N/2 \rceil + 1}(x_{\lceil N/2 \rceil + 1}, \dots, x_N)).
\end{aligned}$$

Lemma A.9. Let $[a, b], [c, d] \in \mathcal{B}(\mathbb{R})$. Then $\text{nmin}^\sharp([a, b], [c, d]) = \text{nmin}^\sharp([c, d], [a, b])$ and

$$\text{nmin}^\sharp([a, b], [c, d]) = \begin{cases} [c + \frac{a-b}{2}, d + \frac{b-a}{2}] & \text{if } d \leq a \\ [a + \frac{c-d}{2}, b + \frac{d-c}{2}] & \text{if } a \leq d \text{ and } b < c \\ [a + c - \frac{b+d}{2}, \frac{b+d}{2}] & \text{if } a \leq d \text{ and } b \geq c \end{cases}$$

Proof. The symmetry on abstract elements is immediate. In the following, we omit some of $\sharp$ to improve readability.

$$\begin{aligned}
\text{nmin}^\sharp([a,b],[c,d]) &= \frac{1}{2}\begin{pmatrix}1 & -1 & -1 & -1\end{pmatrix} R^\sharp\left(\begin{pmatrix}1 & 1\\ -1 & -1\\ 1 & -1\\ -1 & 1\end{pmatrix}\begin{pmatrix}[a,b]\\ [c,d]\end{pmatrix}\right)\\
&= \frac{1}{2}\begin{pmatrix}1 & -1 & -1 & -1\end{pmatrix} R^\sharp\left(\begin{pmatrix}[a,b]+[c,d]\\ -[a,b]-[c,d]\\ [a,b]-[c,d]\\ -[a,b]+[c,d]\end{pmatrix}\right)\\
&= \frac{1}{2}\begin{pmatrix}1 & -1 & -1 & -1\end{pmatrix} R^\sharp\left(\begin{pmatrix}[a+c,b+d]\\ [-b-d,-a-c]\\ [a-d,b-c]\\ [c-b,d-a]\end{pmatrix}\right)\\
&= \frac{1}{2}\begin{pmatrix}1 & -1 & -1 & -1\end{pmatrix}\begin{pmatrix}[R(a+c),R(b+d)]\\ [R(-b-d),R(-a-c)]\\ [R(a-d),R(b-c)]\\ [R(c-b),R(d-a)]\end{pmatrix}\\
&= \frac{1}{2}([R(a+c),R(b+d)]-[R(-b-d),R(-a-c)]\\
&\quad -[R(a-d),R(b-c)]-[R(c-b),R(d-a)])\\
&= \frac{1}{2}([R(a+c),R(b+d)]+[-R(-a-c),-R(-b-d)]\\
&\quad +[-R(b-c),-R(a-d)]+[-R(d-a),-R(c-b)])\\
&= \frac{1}{2}([R(a+c)-R(-a-c),R(b+d)-R(-b-d)]\\
&\quad +[-R(b-c)-R(d-a),-R(a-d)-R(c-b)])
\end{aligned}$$

Claim: $R(a+c)-R(-a-c)=a+c$. If $a+c>0$ then $-a-c<0$ thus the claim in this case. Indeed: If $a+c\leq 0$ then $-a-c\geq 0$ thus $R(a+c)-R(-a-c)=-R(-a-c)=-(-a-c)=a+c$. Similarly $R(b+d)-R(-b-d)=b+d$.

So the expression simplifies to

$$\text{nmin}^\sharp([a,b],[c,d])=\frac{1}{2}([a+c,b+d]+[-R(b-c)-R(d-a),-R(a-d)-R(c-b)])$$

We proceed by case distinction:

Case 1: $b-c\leq 0$: Then $a\leq b\leq c\leq d$:

$$\begin{aligned}
\text{nmin}^\sharp([a,b],[c,d]) &= \frac{1}{2}([a+c,b+d]+[a-d,b-c])\\
&= \frac{1}{2}([a+c+a-d,b+d+b-c])\\
&= [a+\tfrac{c-d}{2},b+\tfrac{d-c}{2}]
\end{aligned}$$

Case 2: $a-d\geq 0$: Then $c\leq d\leq a\leq b$. By symmetry of nmin equivalent to Case 1. Hence

$$\text{nmin}^\sharp([a,b],[c,d])=[c+\tfrac{a-b}{2},d+\tfrac{b-a}{2}].$$

Case 3: $a-d<0$ and $b-c>0$:

$$\begin{aligned}
\text{nmin}^\sharp([a,b],[c,d]) &= \frac{1}{2}([a+c,b+d]+[c-b-d+a,0])\\
&= \frac{1}{2}([a+c+c-b-d+a,b+d])\\
&= [a+c-\tfrac{b+d}{2},\tfrac{b+d}{2}]
\end{aligned}$$

Thus we have

$$\text{nmin}^\sharp([a,b],[c,d]) = \begin{cases} [a+\frac{c-d}{2}, b+\frac{d-c}{2}] & \text{if } b \le c \\ [c+\frac{a-b}{2}, d+\frac{b-a}{2}] & \text{if } d \le a \\ [a+c-\frac{b+d}{2}, \frac{b+d}{2}] & \text{if } a < d \text{ and } b > c \end{cases}$$

□

Definition A.10 (neighboring grid points)**.** Let G be as above. We define the set of *neighboring grid points* of $x \in \Gamma$ by

$$\mathcal{N}(x) := \{g \in G \mid g \in ||x-g|| \le \tfrac{1}{M}\} \setminus \{x\}.$$

For $U \subset \mathbb{R}^m$, we define $\mathcal{N}(U) := \{\mathcal{N}(x) \mid x \in U\} \setminus U$.

Definition A.11 (local bump)**.** Let $M \in \mathbb{N}$, $G := (\frac{\mathbb{Z}}{M})^m$, $\ell = 2^{\lceil \log_2 2m \rceil + 1}$ and let $c = \{\frac{i_1^l}{M}, \frac{i_1^u}{M}\} \times \cdots \times \{\frac{i_m^l}{M}, \frac{i_m^u}{M}\} \subseteq G$. We define a ReLU neural network $\phi_c \colon \mathbb{R}^m \to [0,1]$ w.r.t. the grid G by

$$\phi_c(x) := R\left(\text{nmin}_{2m} \bigcup_{1 \le k \le m} \left\{ R_{[*,1]}(M\ell(x_k - \tfrac{i_k^l}{M}) + 1), R_{[*,1]}(M\ell(\tfrac{i_k^u}{M} - x_k) + 1) \right\}\right)$$

Lemma A.12. It holds:

$$\phi_c(x) := \begin{cases} 0 & \text{if } x \notin \text{conv}(\mathcal{N}(c)) \\ 1 & \text{if } x \in \text{conv}(c) \\ \min\left(0, \bigcup_{k=1}^m \{M\ell(x_k - \frac{i_k^l}{M}) + 1\} \cup \{M\ell(\frac{i_k^u}{M} - x_k) + 1\}\right) & \text{otherwise.} \end{cases}$$

Proof. By case distinction:

- Case $x \notin \mathcal{N}(c)$. Then there exists k, such that either $x_k < \frac{i_k^l - 1}{M}$ or $x_k > \frac{i_k^u + 1}{M}$. Then $M\ell(x_k - \frac{i_k^l}{M}) + 1$ or $M\ell(\frac{i_k^u}{M} - x_k) + 1$ is less or equal to 0. Hence

$$\phi_c(x) = 0.$$

- Case $x \in \text{conv}(c)$. Then for all k holds $\frac{i_k^l}{M} \le x_k \le \frac{i_k^u}{M}$. Thus $M\ell(x_k - \frac{i_k^l}{M}) + 1 \ge 1$ and $M\ell(\frac{i_k^u}{M} - x_k) + 1 \ge 1$ for all k Hence

$$\phi_c(x) = 1.$$

  where $\alpha \ge 1$.

- Case otherwise: For all x exists a k such that $M\ell(x_k - \frac{i_k^l}{M}) + 1$ or $M\ell(\frac{i_k^u}{M} - x_k) + 1$ is smaller or equal to all other arguments of the function $\min$ and smaller or equal to 1. If the smallest element is smaller than 0, then $\phi_c(x)$ will evaluate to 0, otherwise it will evaluate to $M\ell(x_k - \frac{i_k^l}{M}) + 1$ or $M\ell(\frac{i_k^u}{M} - x_k) + 1$. Thus we can just drop R and $R_{[*,1]}$ from the equations and take the minimum also over 0:

$$\begin{aligned} \phi_c(x) &= R\left(\min \bigcup_{k=1}^m \left\{ R_{[*,1]}(M\ell(x_k - \tfrac{i_k^l}{M}) + 1), R_{[*,1]}(M\ell(\tfrac{i_k^u}{M} - x_k) + 1) \right\}\right) \\ &= \min\left(0, \bigcup_{k=1}^m \{(M\ell(x_k - \tfrac{i_k^l}{M}) + 1)\} \cup \{(M\ell(\tfrac{i_k^u}{M} - x_k) + 1)\}\right) \\ &= \min \bigcup_{k=0}^m \{M\ell(x_k - \tfrac{i_k^l}{M}) + 1\} \cup \{M\ell(\tfrac{i_k^u}{M} - x_k) + 1\} \end{aligned}$$

□

Lemma A.13. Let $[u_1, 1], \ldots, [u_N, 1]$ be abstract elements of the Interval Domain $\mathcal{B}$. Then

$$\text{nmin}^\sharp_N([u_1, 1], \ldots, [u_N, 1]) = [u_1 + \cdots u_N + 1 - N, 1].$$

Proof. By induction. Base case: Let $N = 1$. Then $\text{nmin}^\sharp_1([u_1, 1]) = [u_1, 1]$. Let $N = 2$. Then $\text{nmin}^\sharp_2([u_1, 1], [u_2, 1]) = [u_1 + u_2 - 1, 1]$.

Induction hypothesis: The property holds for N' s.t. $0 < N' \leq N - 1$.

Induction step: Then it also holds for N:

$$\begin{aligned}
\text{nmin}^\sharp_N([u_1, 1], \ldots, [u_N, 1]) &= \text{nmin}^\sharp(\text{nmin}^\sharp_{\lceil N/2\rceil}([u_1, 1], \ldots, [u_{\lceil N/2\rceil}, 1]), \\
&\quad\ \text{nmin}^\sharp_{N-\lceil N/2\rceil}([u_{\lceil N/2\rceil+1}, 1], \ldots, [u_N, 1])) \\
&= \text{nmin}^\sharp([u_1 + \cdots + u_{\lceil N/2\rceil} + 1 - \lceil N/2\rceil, 1], \\
&\quad\ [u_{\lceil N/2\rceil+1} + \cdots u_N + 1 - N + \lceil N/2\rceil, 1]) \\
&\overset{Lemma\ A.9}{=} [u_1 + \cdots + u_N + 2 - \lceil N/2\rceil - N + \lceil N/2\rceil - 1, 1] \\
&= [u_1 + \cdots + u_N + 1 - N, 1]
\end{aligned}$$

□

Lemma A.14. Let $[a, b], [u, 1] \in \mathcal{B}(\mathbb{R}_{\leq 1})$. Then

$$\text{nmin}^\sharp([a, b], [u, 1]) \subseteq [a + \tfrac{u-1}{2}, \tfrac{b+1}{2}]$$

Proof.

$$\text{nmin}^\sharp([a, b], [u, 1]) = \begin{cases} [a + \frac{u-1}{2}, b + \frac{1-u}{2}] & \text{if } b \leq u \\ [a + u - \frac{b+1}{2}, \frac{b+1}{2}] & \text{if } b \geq u \end{cases}$$

If $b \leq u$ then $b + \frac{1-u}{2} \leq b + \frac{1-b}{2} = \frac{b+1}{2}$. If $u \leq b$ then $a + u - \frac{b+1}{2} \geq a + u - \frac{u+1}{2} = a + \frac{u-1}{2}$. So

$$\text{nmin}^\sharp([a, b], [u, 1]) \subseteq [a + \tfrac{u-1}{2}, \tfrac{b+1}{2}].$$

□

Lemma A.15. Let $N \in \mathbb{N}_{\geq 2}$, let $[u_1, 1], \ldots, [u_{N-1}, 1], [u_N, d] \in \mathcal{B}(\mathbb{R})$ s.t. $b \leq 1$ be abstract elements of the Interval Domain $\mathcal{B}$. Furthermore, let $H(x) := \frac{1+x}{2}$. Then there exists a $u \in \mathbb{R}$ s.t.

$$\text{nmin}^\sharp_N([u_1, 1], \ldots, [u_{N-1}, 1], [u_N, d]) \subseteq [u, H^{\lceil \log_2 N\rceil+1}(d)]$$

Proof. By induction: Let $N = 2$:

$$\text{nmin}^\sharp_2([u_1, 1], [u_2, d]) \overset{Lemma\ A.14}{=} [a + \tfrac{u_1-1}{2}, H(d)]$$

Let $N = 3$:

$$\begin{aligned}
\text{nmin}^\sharp_3([u_1, 1], [u_2, 1], [u_3, d]) &= \text{nmin}^\sharp(\text{nmin}^\sharp([u_1, 1], [u_2, 1]), [u_3, d]) \\
&= \text{nmin}^\sharp([u_1 + u_2 - 1, 1], [u_3, d]) \\
&\subseteq [u_3 + \tfrac{u_1+u_2-2}{2}, H(d)] \\
\text{nmin}^\sharp_3([u_1, 1], [a, b], [u_2, 1]) &= \text{nmin}^\sharp_3([u_3, d], [u_1, 1], [u_2, 1]) \\
&= \text{nmin}^\sharp(\text{nmin}^\sharp([u_3, d], [u_1, 1]), [u_2, 1]) \\
&= \text{nmin}^\sharp([u_3 + \tfrac{u_1-1}{2}, H(d)], [u_2, 1]) \\
&\subseteq [u_3 + \tfrac{u_1+u_2-2}{2}, H^2(d)]
\end{aligned}$$

So $\text{nmin}^\sharp_3([u_3, d], [u_1, 1], [u_2, 1])$ is always included in $[u_3 + \frac{u_1+u_2-2}{2}, H^2(d)]$.

Induction hypothesis: The statement holds for all $2 \leq N' \leq N - 1$.

Induction step: Then the property holds also for N:

$$\begin{aligned}\text{nmin}^{\sharp}_{N}([u_N, d], [u_1, 1], \ldots, [u_{N-1}, 1]) &= \text{nmin}^{\sharp}(\text{nmin}^{\sharp}_{\lceil N/2 \rceil}([u_N, d], [u_1, 1], \ldots, [u_{\lceil N/2 \rceil - 1}, 1]),\\ &\quad \text{nmin}^{\sharp}_{N - \lceil N/2 \rceil}([u_{\lceil N/2 \rceil}, 1], \ldots, [u_{N-1}, 1]))\\ &= \text{nmin}^{\sharp}([u', H^{\lceil \log_2 \lceil N/2 \rceil \rceil + 1}(d)], [u'', 1])\\ &\subseteq \text{nmin}^{\sharp}([u', H^{\lceil \log_2 N/2 \rceil + 1}(d)], [u'', 1])\\ &= \text{nmin}^{\sharp}([u', H^{\lceil \log_2 N - \log_2(2) \rceil + 1}(d)], [u'', 1])\\ &= \text{nmin}^{\sharp}([u', H^{\lceil \log_2 N - 1 \rceil + 1}(d)], [u'', 1])\\ &= \text{nmin}^{\sharp}([u', H^{\lceil \log_2 N \rceil}(d)], [u'', 1])\\ &= [u''', H^{\lceil \log_2 N \rceil + 1}(d)]\end{aligned}$$

and similarly for other orderings of the arguments. □

Lemma A.16. Let $H(x) := \frac{1+x}{2}$. For all $N \in \mathbb{N}_{>0}$, we have that $d \leq 1 - 2^N$ implies $H^N(d) \leq 0$.

Proof. By induction. $N = 1$: Then $H(1-2) = \frac{1+1-2}{2} = 0$

Induction hypothesis. The statement holds for all N' such that $0 < N' \leq N$.

Induction step: $N + 1$: $d \leq 1 - 2^N$:

$$H^{N+1}(d) \leq H^{N+1}(1 - 2^{N+1}) = H^N(H(1 - 2^{N+1})) = H^N(\tfrac{1+1-2^{N+1}}{2}) = H^N(1 - 2^N) \leq 0$$

□

Lemma A.17. For all boxes $B \in \mathcal{B}(\mathbb{R}^m)$, we have

$$\phi^{\sharp}_c(B) = \begin{cases} [1, 1] & \text{if } B \subseteq \text{conv}(c) \\ [0, 0] & \text{if } B \subseteq \Gamma \setminus \text{conv}(\mathcal{N}(c)) \end{cases}$$

Furthermore, $\phi^{\sharp}_c(B) \subseteq [0, 1]$.

Proof. Let ϕ_c be a local bump and let $B = [a, b] \in \mathcal{B}(\mathbb{R}^m)$. Let $[r^1_k, s^1_k], [r^2_k, s^2_k] \in \mathcal{B}(\mathbb{R})$ such that $M\ell([a_k, b_k] - \frac{i^l_k}{M}) + 1 = [r^1_k, s^1_k]$ and $M\ell(\frac{i^u_k}{M} - [a_k, b_k]) + 1 = [r^2_k, s^2_k]$.

- If $[a, b] \subseteq \text{conv}(c)$: Then $1 \leq r^1_k$ and $1 \leq r^2_k$ for all $k \in \{1, \ldots, m\}$. Thus

$$\begin{aligned}\phi^{\sharp}_c([a, b]) &= R^{\sharp}(\text{nmin}^{\sharp}_{2m}\{R^{\sharp}_{[*,1]}([r^p_k, s^p_k])\}_{(p,k) \in \{1,2\} \times \{1,\ldots,m\}})\\ &= R^{\sharp}(\text{nmin}^{\sharp}_{2m}\{[1, 1]\}_{(p,k) \in \{1,2\} \times \{1,\ldots,m\}})\\ &= [1, 1]\end{aligned}$$

- If $[a, b] \subseteq \Gamma \setminus \text{conv}(\mathcal{N}(c))$: Then there exists a $(p', k') \in \{1, 2\} \times \{1, \ldots, m\}$ such that $s^{p'}_{k'} \leq 1 - 2^{\lceil \log_2 N \rceil + 1}$. Using Lemma A.16 and Lemma A.15, we now that there exists a $u \in \mathbb{R}$ s.t.

$$\begin{aligned}\phi^{\sharp}_c([a, b]) &= R^{\sharp}(\text{nmin}^{\sharp}_{2m}\{R^{\sharp}_{[*,1]}([r^p_k, s^p_k])\}_{(p,k) \in \{1,2\} \times \{1,\ldots,m\}})\\ &= R^{\sharp}(\text{nmin}^{\sharp}_{2m}\{[R_{[*,1]}(r^p_k), R_{[*,1]}(s^p_k)]\}_{(p,k) \in \{1,2\} \times \{1,\ldots,m\}})\\ &\subseteq R^{\sharp}(\text{nmin}^{\sharp}_{2m}\{[R_{[*,1]}(r^p_k), 1]\}_{(p,k) \neq (p',k')} \cup \{[r^{p'}_{k'}, s^{p'}_{k'}]\})\\ &\subseteq R^{\sharp}([u, 0])\\ &= [0, 0]\end{aligned}$$

For any $[a, b] \in \mathcal{B}(\Gamma)$ we have $\phi^{\sharp}_c([a, b]) \subseteq [0, 1]$ by construction. □

Lemma A.18. Let $\Gamma \subset \mathbb{R}^m$ be a closed box and let $f\colon \Gamma \to \mathbb{R}$ be continuous. For all $\delta > 0$ exists a set of ReLU networks $\{n_k\}_{0 \le k \le N-1}$ of size $N \in \mathbb{N}$ approximating the N-slicing of f, $\{f_k\}_{0 \le k \le N-1}$ (ξ_k as in Definition A.2) such that for all boxes $B \in \mathcal{B}(\Gamma)$

$$n_k^\sharp(B) = \begin{cases} [0,0] & \text{if } f(B) \le \xi_k - \frac{\delta}{2} \\ [1,1] & \text{if } f(B) \ge \xi_{k+1} + \frac{\delta}{2}. \end{cases}$$

and $n_k^\sharp(B) \subseteq [0,1]$.

Proof. Let $N \in \mathbb{N}$ such that $N \ge 2\frac{\xi_{\max} - \xi_{\min}}{\delta}$ where $\xi_{\min} := \min f(\Gamma)$ and $\xi_{\max} := \max f(\Gamma)$. For simplicity we assume $\Gamma = [0,1]^m$. Using the Heine-Cantor theorem, we get that f is uniformly continuous, thus there exists a $\delta' > 0$ such that $\forall x, y \in \Gamma. ||y - x||_\infty < \delta' \Rightarrow ||f(y) - f(x)|| < \frac{\delta}{2}$. Further, let $M \in \mathbb{N}$ such that $M \ge \frac{1}{\delta'}$ and let G be the grid defined by $G := (\frac{\mathbb{Z}}{M})^m \subseteq \mathbb{R}^m$.

Let $C(B)$ be the set of corner points of the closest hyperrectangle in G confining $B \in \mathcal{B}(\Gamma)$. We construct the set

$$\Delta_k := \{C(B) \mid B \in \mathcal{B}(\Gamma) : f(B) \ge \xi_{k+1} + \tfrac{\delta}{2}\}.$$

We claim that $\{n_k\}_{0 \le k \le N-1}$ defined by

$$n_k(x) := R_{[*,1]}\left(\sum_{c \in \Delta_k} \phi_c(x)\right)$$

satisfies the condition.

Case 1: Let $B \in \mathcal{B}(\Gamma)$ such that $f(B) \ge \xi_{k+1} + \frac{\delta}{2}$. Then for all $g \in \mathcal{N}(B)$ holds $f_k(g) = \delta_2$. By construction exists a $c' \in \Delta_k$ such that $B \subseteq \text{conv}(c')$. Using Lemma 4.3 we get

$$\begin{aligned} n_k^\sharp(B) &= R^\sharp_{[*,1]}\left(\sum_{c \in \Delta_k} \phi_c^\sharp(B)\right) = R^\sharp_{[*,1]}\left(\phi_{c'}^\sharp(B) + \sum_{c \in \Delta_k \setminus c'} \phi_c^\sharp(B)\right) \\ &= R^\sharp_{[*,1]}([1,1] + [p_1, p_2]) = [1,1], \end{aligned}$$

where $[p_1, p_2] \in \mathcal{B}(\mathbb{R}_{\ge 0})$. Indeed, by case distinction:

Case 2: Let $B \in \mathcal{B}(\Gamma)$ such that $f(B) \le \xi_k - \frac{\delta}{2}$. Then for all $g \in \mathcal{N}(B)$ holds $f_k(g) = 0$. Further, $B \cap \text{conv}(\mathcal{N}(c)) = \emptyset$ for all $c \in \Delta_k$ because G is fine enough. Using Lemma 4.3 we obtain

$$n_k^\sharp(B) = R^\sharp_{[*,1]}\left(\sum_{c \in \Delta_k} \phi_c^\sharp(B)\right) = R^\sharp_{[*,1]}([0,0]) = [0,0].$$

By construction we have $n_k^\sharp(B) \subseteq [0,1]$. ☐