Tessellation-Filtering ReLU Neural Networks

Bernhard A. Moser[1*] , **Michal Lewandowski**[1] , **Somayeh Kargaran**[1] ,
Werner Zellinger[1] , **Battista Biggio**[2] , **Christoph Koutschan**[3]
[1]Software Competence Center Hagenberg GmbH (SCCH), Austria
[2] Pattern Recognition and Application Lab (PRA Lab) at University of Cagliari, Italy
[3]Radon Inst. for Computational and Applied Mathematics (RICAM) at Austrian Academy of Sciences
{bernhard.moser,michal.lewandowski,somayeh.kargaran,werner.zellinger}@scch.at,
battista.biggio@unica.it, christoph.koutschan@ricam.oeaw.ac.at

Abstract

We identify tessellation-filtering ReLU neural networks that, when composed with another ReLU network, keep its non-redundant tessellation unchanged or reduce it. The additional network complexity modifies the shape of the decision surface without increasing the number of linear regions. We provide a mathematical understanding of the related additional expressiveness by means of a novel measure of shape complexity by counting deviations from convexity which results in a Boolean algebraic characterization of this special class. A local representation theorem gives rise to novel approaches for pruning and decision surface analysis.

1 Introduction

We present a novel approach to analyze and quantify the shape complexity of the decision surface of a deep model based on Rectified Linear Units. To this end, we introduce a special class of functions that reflects the non-convexity characteristics of the decision surface in an extended neighborhood of a point in a Boolean-algebraic way, thus providing a rich mathematical structure for analysis and computation. As main result we provide a Representation Theorem that decomposes the network into a shape complexity part, represented by our special class, and an underlying hyperplane tessellation. For sake of comprehensibility we use an instructive example that resembles the construction of a Cantor set in fractal geometry.

The Rectified Linear Unit (ReLU), $\sigma(x) := \max(x, 0)$, $x \in \mathbb{R}$, is currently the most commonly used non-linear activation function in deep learning models [Glorot *et al.*, 2011; Szandała, 2021]. It is motivated by the neocortex [Hahnloser *et al.*, 2000] and many of the most widely used neural network models such as VGG [Simonyan and Zisserman, 2015], GoogLeNet [Szegedy *et al.*, 2015] or ResNet [He *et al.*, 2016] are based on the ReLU activation function. ReLU networks do also play a prominent role in theoretical analysis, particularly in the context of understanding complexity and expressiveness of deep neural networks [Montúfar *et al.*, 2014; Raghu *et al.*, 2017; Arora *et al.*, 2018; Montúfar *et al.*, 2021].

*Contact Author

Since a deep ReLU neural network (ReLU DNN) represents a piecewise linear function it can be partitioned into affine functions over polyhedral regions, also called *linear regions* in the literature. The transition from one linear region to another one changes the slopes in the input-output function as well as it changes the activation pattern of firing neurons. A larger number of linear regions therefore indicates greater expressiveness of the underlying DNN architecture. Previous research on understanding ReLU DNNs has focused on analyzing the number of linear regions by searching for lower and upper bounds for a given network architecture in terms of number of network layers and number of neurons [Montúfar *et al.*, 2014; Serra *et al.*, 2018]. It is argued that, in general, the number of linear regions grows polynomially with the number of neurons but can grow exponentially with the number of layers, underpinning the intuition that deep neural networks have greater expressiveness than shallow architectures. However there remains a gap in explaining the theoretical bounds found by specifically constructed ReLU DNNs and the number of linear regions observed in practice [Hanin and Rolnick, 2019]. From a topological perspective the concept of Betti numbers has been investigated [Bianchini and Scarselli, 2014] to characterize the connectivity properties of the induced decision function. For a subset $S \subset \mathbb{R}^n$ there are n Betti numbers [Peterson, 1969], $b_k(S) \in \mathbb{N}_0$, $0 \leq k \leq n-1$, where $b_0(S)$ denotes the number of connected components and $b_k(S)$ the number of k-dimensional holes in S. Since geometric bodies with different Betti numbers cannot transformed into each other by means of bicontinuous mappings, Betti numbers provide a consistent notion of complexity of geometric regions, respectively decision surfaces in high-dimensional spaces. For instance, a disc in $\mathbb{R}^2$ has no holes, therefore, we have $b_1(\text{disc}) = 0$ while $b_1(\text{circle}) = 1$, and b_1 equals 1 for a $2D$-sphere in $\mathbb{R}^3$, while $b_1 = 2$ for a torus. It can be shown that the sum of Betti numbers can grow exponentially in the number of neurons of a deep network while it is limited by polynomial growth for shallow networks [Bianchini and Scarselli, 2014].

To start the discussion we introduce two illustrative examples with decision surfaces without any holes, i.e., with vanishing Betti numbers, but showing different effects when being modeled by means of ReLU DNNs.

Example 1. *For our first example we start with the function* $A(x) := \max\{-3x + 1, 0, 3x - 2\}$ *on* $[0, 1]$ *and its*

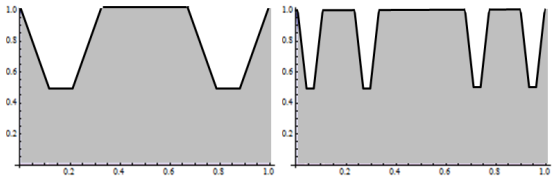

Figure 1: Ragged decision surfaces ($k = 2, 3$) from Example 1.

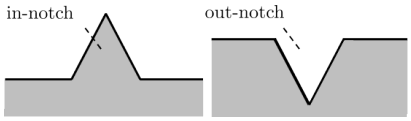

Figure 2: Illustration of in- and out-notches as two types of deviations from convexity.

recursively nested composition $A^{(k+1)}(x) := A(A^{(k)}(x))$, $A^{(1)}(x) := A(x)$, *defining decision surfaces (here a* 1*-dimensional decision curve) as upper border of the regions*

$$R_k := \{(x, y) \in [0, 1]^2 \mid y \leq (A^{(k)}(x) + 1)/2\}. \quad (1)$$

See Figure 1 for an illustration.

Example 2. *Our second example is the function that checks whether all elements of a finite list of non-negative values* $(x_1, \ldots, x_n) \in [0, \infty)^n$ *are strictly positive or not.*

Despite vanishing Betti numbers, this example shows a complex decision surface approximating a fractal geometry for growing recursion depth, which can be realized in various ways: for example, as a deep network that is constructed from the recursion, but also as a shallow network by modeling each of the zigzag components separately in a first layer and then combining them in a second layer. This also shows that ReLU DNN representations are not unique. Though the second example can be fully characterized by n half-space decisions, its realization by means of a ReLU DNN, e.g., $\min\{x_1, x_2\} = (x_1 + x_2 - |x_1 - x_2|)/2$ induces a tessellation with an exponential number of linear regions. The examples demonstrate (1) that Betti numbers are a too coarse concept of complexity to represent the expressive power of neural networks, and (2) that there is a gap between expressiveness in terms number of linear regions on the one hand and expressiveness in terms of which decision surfaces can be represented. In this context, shape complexity beyond Betti numbers has received too little attention in the literature until now. To bridge this gap, after fixing notation and providing preliminaries from discrete mathematics in Section 2, we will introduce in Section 3 the central notion of *notches* to characterize deviations from convexity. As illustrated in Figure 2, we distinguish between *in-notches* and *out-notches*. We start in Section 3 with introducing a measure of shape complexity for quantifying non-convexity structures of a decision surface by exploiting invariant geometric properties between a hyperplane tessellation and its representation in the hypercube of activation patterns. In this context a special class of functions come into play which distinguishes by its shape complexity properties, see Section 4. This special class leads to a Local Representation Theorem, which is discussed in terms of relevance and future research, see Section 5.

2 Preliminaries

The ReLU activation function σ is defined on vectors $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ through entry-wise operation, i.e., $\sigma(x) := (\max\{x_1, 0\}, \ldots, \max\{x_n, 0\})$. The architecture $A = (W_i, b_i)_{i=1,\ldots,L}$ of a deep neural network (DNN) is specified by a sequence of matrix-vector pairs (W_i, b_i) with $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and $b_i \in \mathbb{R}^{n_i}$. The number n_0 denotes the dimension of the input space, while n_i is called the width of the i-th layer. L is the depth of the network and $N = \sum_{i=1}^{L} n_i$ the total number of neural units. Together with the ReLU activation function σ the architecture $A = (W_i, b_i)_{i=1,\ldots,L}$ defines an input-output function $\mathcal{N}_L \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ resulting from the composition of layers

$$\mathcal{N}_L(x_0) = \sigma \circ f_L \circ \ldots \circ \sigma \circ f_1(x_0) \quad (2)$$

with $f_i(x) := W_i\, x + b_i$. For sake of simplicity we focus on binary classification problems. One way to define such a classifier is by means of $n_L = 2$. The maximal value in the final output $(\alpha_{L,1}(x_0), \alpha_{L,2}(x_0))$ indicates the class. If $\alpha_{L,1}(x_0) \geq \alpha_{L,2}(x_0)$ the point x_0 is classified to belong to the class $\mathcal{C}$, otherwise not. Note that this classification rule can be equivalently represented by an $(L + 1)$-layered network with $n_{L+1} = 1$ by setting $\alpha_{L+1}(x_0) := \max\{\alpha_{L,2}(x_0) - \alpha_{L,1}(x_0), 0\}$. Because of $a \geq b$ if and only if $\max\{b - a, 0\} = 0$, the points classified to belong to the class $\mathcal{C}$ can be represented as 0-preimage, $\mathcal{N}^{-1}(0)$, of the function $\mathcal{N} \colon \mathbb{R}^{n_0} \to \mathbb{R}_0$. We are interested in the geometry of 0-preimages of ReLU DNNs and say that two *ReLU neural networks are equivalent* if their 0-preimages coincide.

2.1 Activation Space and Tessellations

Given some data point x_0 we obtain the neural *activation pattern* $\alpha(x_0) = ((\mathcal{N}_1(x_0))^T, \ldots, (\mathcal{N}_L(x_0))^T) \in \mathbb{R}_0^N$. An entry in the activation pattern is either 0 or positive, referred to as non-active (non-firing) or active (firing) status, respectively. The binary activation states, active or not, are represented by a vector $\pi_{\mathcal{N}}(x_0)$ in the hypercube $\mathcal{H}_N = \{0, 1\}^N$. We call $\pi_{\mathcal{N}} : x \mapsto \pi_{\mathcal{N}}(x) \in \mathcal{H}_N$ the *activation pattern mapping* w.r.t $\mathcal{N}$. The activation region $R(\pi_{\mathcal{N}}(x_0)) = \{x \in \mathbb{R}^{n_0} \mid \pi_{\mathcal{N}}(x) = \pi_{\mathcal{N}}(x_0)\}$ of points x with the same activation pattern $\pi_{\mathcal{N}}(x_0) \in \mathcal{H}_N$ results as solution from a finite number of affine inequalities, therefore yielding a (convex) polyhedron [Hein *et al.*, 2018; Shepeleva *et al.*, 2020]. Let denote by $\mathrm{Tess}[\mathcal{N}]$ the resulting tessellation induced by the ReLU DNN $\mathcal{N}$.

The collection of all half-spaces that describe such polyhedral cells as intersections induces another tessellation. We call it its h-tessellation $\mathrm{Tess}_h[\mathcal{N}]$ and its activation pattern mapping $\pi^h_{\mathcal{N}}(x_0)$, defined accordingly. Note that a cell in $\mathrm{Tess}[\mathcal{N}]$ is a finite union of cells in $\mathrm{Tess}_h[\mathcal{N}]$.

According to our setup of the classification problem, we distinguish between '0'- and '> 0'-activations patterns, respectively, depending on whether an activation pattern results from a point in $\mathcal{N}^{-1}(0)$ or its complement. We denote by $\mathcal{A}_0 := \{\pi_{\mathcal{N}}(x) \mid x \in \mathcal{N}^{-1}(0)\}$ the set of 0-activation patterns, and we define $\mathcal{A}_{>0} := \{\pi_{\mathcal{N}}(x) \mid x \in \mathbb{R}^{n_0} \setminus \mathcal{N}^{-1}(0)\}$;

we write $\mathcal{A}_0^h := \{\pi_{\mathcal{N}}^h(x) \mid x \in \mathcal{N}^{-1}(0)\}$, resp., $\mathcal{A}_{>0}^h$, if the activation patterns refer to the h-tessellation.

We call an activation region *in-cell* if it is contained in the 0-preimage of $\mathcal{N}$, i.e., its associated activation pattern π is an element of $\mathcal{A}_0$. Otherwise we call an activation region *out-cell*.

A tessellation $\mathrm{Tess}_h[\mathcal{N}]$ can be redundant in the sense that the removal of some of its half-spaces does not change the decision surface. We write $\mathrm{effTess}_h[\mathcal{N}]$ if all redundant half-spaces have been removed, and call it the *effective h-tessellation* of $\mathcal{N}$.

If only a single entry differs when comparing two activation patterns π_A and π_B, the corresponding polyhedral activation regions R_{π_A} and R_{π_B} touch each other by sharing a common polyhedral face. This way we can introduce notions of connectedness and other topological concepts.

We say that two activation patterns π_A and π_B are *adjacent* if their Hamming distance equals 1, i.e., $d_H(\pi_A, \pi_B) = 1$. A path between π_A and π_B is a sequence of adjacent activation patterns connecting π_A with π_B. The shortest path length is given by the Hamming distance $d = d_H(\pi_A, \pi_B)$. For $M \subseteq \mathcal{H}_N$, we write $\pi_A \sim_M \pi_B$ if there is a connecting path in M between $\pi_A \in M$ and $\pi_B \in M$. Accordingly a set C of activation patterns is *connected* if there is a path connecting any pair of points in the set C such that all activation patterns along the path are also in C. A *connected component* is a maximally connected set. In contrast to Euclidean spaces the shortest path in a Hamming space is not unique.

2.2 Convexity in Hamming Space

The convexity of a set C in Euclidean geometry is characterized by the property that any line segment connecting two points in the set is also contained in the set C. As a line segment is the shortest path in Euclidean geometry we generalize this concept to the Hamming space as follows.

Definition 1 (Convex Sets (Hulls) of Activation Patterns). *A set $C \subseteq \{0,1\}^m$ of activation patterns is convex if and only if it contains all shortest Hamming paths connecting any two points in C. The convex hull $\mathrm{conv}_H[C]$ of activation patterns $C \subseteq \{0,1\}^m$ is the smallest convex set containing C.*

The following lemma allows us to reformulate geometric problems in h-tessellations in terms of combinatorial problems in the hypercube $\mathcal{H}_m = \{0,1\}^m$. A k-dimensional face of $\mathcal{H}_m$ consists of all points that agree on a collection of $m-k$ coordinates, thereby forming a hypercube of dimension k.

Lemma 1 (Equivalence of Convexity Notions). *A convex arrangement of activation regions of an h-tessellation with m half-spaces in Euclidean space corresponds to a convex set of activation patterns in Hamming space, i.e., a face of the hypercube $\mathcal{H}_m = \{0,1\}^m$, and vice versa.*

3 Notches and Shape Complexity

Instead of counting holes by means of Betti numbers [Bianchini and Scarselli, 2014], we are counting notches, i.e., deviations from convexity, in terms of minimal convex arrangements of activation regions in an h-tessellations that cover the regions of non-convexity. Due to Lemma 1 we can equivalently formulate this notion of a notch in the Hamming space of activation patterns. An *out-notch* can be characterized as convex region ν_o of out-cells (resp. activation patterns of out-cells) for which there is a shortest path between activation patterns of in-cells visiting ν_o. See Figure 3 for an illustration. Analogously, we define an *in-notch* to be a convex region ν_ι of in-cells for which there is a shortest path between activation patterns of out-cells visiting ν_ι.

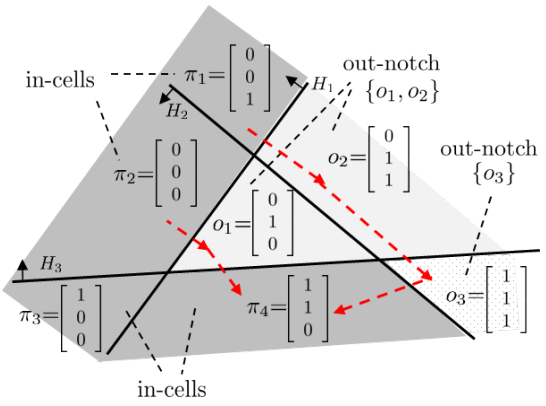

Figure 3: In this arrangement of activation regions of an h-tessellation there are two possible configurations of out-notches: first, $\{o_1, o_2\}$ and $\{o_2, o_3\}$ (as shown), and, second, $\{o_1\}$ and $\{o_2, o_3\}$. The red paths indicate corresponding shortest paths between activation patterns of in-cells visiting the out-notches.

Definition 2 (Notches in Hamming Space). *Consider an h-tessellation of non-trivial cells generated by m many half-spaces with cells being either in- or out-cells. A notch to the set $M \subset \{0,1\}^m$ within the hypercube $\mathcal{H}_M := \mathrm{conv}_H[M]$ is a face $\nu \subseteq \mathcal{H}_M \setminus M$ that is adjacent to some $\pi_1, \pi_2 \in M$ with non-empty overlap with their convex hull $\mathrm{conv}_H[\{\pi_1, \pi_2\}]$, i.e., $d_H(\nu, \pi_1) = 1$, $d_H(\nu, \pi_2) = 1$ and $\mathrm{conv}_H[\{\pi_1, \pi_2\}] \cap \nu \neq \emptyset$. In case of $M \subseteq \mathcal{A}_0^h$ we call the notch an out-notch, and if $M \subseteq \mathcal{H}_M \setminus \mathcal{A}_0^h$ we call it an in-notch.*

The cardinality of a minimal configuration of notches such that no further notch can be constructed gives the Out-Notch-Number (resp. In-Notch-Number). For convenience we denote $\mathcal{H}_M := \mathrm{conv}_H[M]$, $\langle \pi^* \rangle := \{\pi \in M \mid \pi \sim_M \pi^*\}$, and $\mathcal{H}_{\pi^*} := \mathrm{conv}_H[\langle \pi^* \rangle]$.

Definition 3 (Notch Number). *Given an h-tessellation ReLU DNN $\mathcal{N} : \mathbb{R}^n \to \mathbb{R}_0$, we define the Notch Number w.r.t. the set M, $\Theta_{\mathcal{N}}[M]$, to be the minimal exhaustive number of notches to M within $\mathcal{H}_M$. We write $\Theta_{\mathcal{N}}^o$ in case of $M = \mathcal{A}_0^h$ and $\Theta_{\mathcal{N}}^\iota$ if $M = \mathcal{H}_{\mathcal{A}_0}^h \setminus \mathcal{A}_0^h$. Further, we define the local versions, Local Out-Notch Number, resp. Local In-Notch Number, at the data point x_0, resp. its associated activation pattern $\pi^* := \pi(x_0)$ by setting*

$$\begin{aligned} \Theta_{\mathcal{N}}^o(x_0) &:= \Theta_{\mathcal{N}}[\langle \pi^* \rangle], \\ \Theta_{\mathcal{N}}^\iota(x_0) &:= \Theta_{\mathcal{N}}[\mathcal{H}_{\pi^*} \setminus \langle \pi^* \rangle]. \end{aligned} \tag{3}$$

3.1 ReLU Representation w.r.t. In-Notches

We can construct a ReLU DNN $\mathcal{N}^\iota$ by starting with an exhaustive list of in-notches $\{\nu_1^\iota, \ldots, \nu_{\Theta_f^\iota}^\iota\}$ to $\mathbb{R}^n \setminus f^{-1}(0)$ and checking the condition

$$\exists k : x_0 \in \nu_k^\iota \iff \mathcal{N}^\iota(x_0) = 0.$$

This can be achieved by setting

$$\mathcal{N}^{\iota}(x_0) := \min_{k=1}^{\Theta_f^{\iota}} \left\{ \sum_{j=1}^{N_k^{\iota}} \max\{\widetilde{a}_k^j(x_0), 0\} \right\}, \tag{4}$$

where N_k^{ι} denotes the number of faces of notch ν_k^{ι}, and $\widetilde{a}_k^j(x_0) := \sigma_k^j \, d(x_0, F_k^i)$ denotes the distance between x_0 and the hyperplane containing the k-th face F_k^i of in-notch ν_k^{ι} with the sign $\sigma_k^j \in \{-1, 1\}$ defined such that $\widetilde{a}_k^j(x_0) \leq 0$ if $x_0 \in \nu_k^{\iota}$.

3.2 Reconstruction of h-Tessellation

Due to tropical algebra [Zhang *et al.*, 2018; Alfarra *et al.*, 2020; Trimmel *et al.*, 2021], a ReLU DNN $\mathcal{N}$ can always be represented in the form of

$$\mathcal{N}(x) = \max_{i=1}^{r}\{a_i(x)\} - \max_{j=1}^{s}\{b_j(x)\} \tag{5}$$

with affine functions a_j and b_j. Now, let us transform (5) to obtain the form of (4) by means of $\max_{i=1}^{r}\{a_i(x)\} + \min_{j=1}^{s}\{-b_j(x)\} = \min_{j=1}^{s}\{\max_{i=1}^{r}\{a_i(x) - b_j(x)\}\}$. By setting

$$h_{i,j}(x) := \max\{a_i(x) - b_j(x), 0\} \tag{6}$$

we obtain an h-tessellation that contains the effective h-tessellation.

3.3 Global Shape Complexity

Note that the sum of Out- and In-Notch Numbers,

$$\Theta_{\mathcal{N}} = \Theta_{\mathcal{N}}^{o} + \Theta_{\mathcal{N}}^{\iota}, \tag{7}$$

reflects the minimal cardinality of a partition of the input space into (convex) polytopes that allows to correctly represent the decision problem whether a point belongs to the 0-preimage of f or not. As a consequence, $\Theta_{\mathcal{N}}$ proves to be a property of the decision surface itself which is invariant to the complexity of architectures of ReLU DNNs $\mathcal{N}$ realizing the same decision problem. In particular, $\Theta_{\mathcal{N}}$ remains unchanged if the network architecture is differently encoded say by changing the orientations of the half-spaces, or even by relabeling the classes, i.e., interchanging 0 with > 0 (encoded by 1). In the latter, we obtain a new function $\widetilde{f}$ with $\widetilde{f}^{-1}(0) = f^{-1}(> 0)$ and, accordingly, $\widetilde{f}^{-1}(> 0) = f^{-1}(0)$, which leads to a change of in- to out-notches and vice versa, so that in the end $\Theta_{\mathcal{N}}$ remains unchanged.

Therefore, it is justified to write synonymously $\Theta_f = \Theta_{\mathcal{N}}$. This motivates us to consider $\Theta_{\mathcal{N}}$ as a measure for the shape complexity of the decision surface of a ReLU DNN $\mathcal{N}$.

For $x_0 \in \mathcal{N}^{-1}(0)$ and $\pi^* = \pi(x_0) = (\alpha_1^*, \ldots, \alpha_m^*) \in \mathcal{H}_m$, we consider the mapping $T_{\pi^*}\colon \mathcal{H}_m \to \mathcal{H}_m$, $(\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_m) = T_{\pi^*}(\alpha_1, \ldots, \alpha_m)$ given by

$$\widetilde{\alpha}_i := \begin{cases} 1 - \alpha_i, & \text{if } \alpha_i^* = 1, \\ \alpha_i, & \text{if } \alpha_i^* = 0. \end{cases} \tag{8}$$

The mapping T_{π^*} preserves Hamming distances, $d_H(T_{\pi^*}(\pi_1), T_{\pi^*}(\pi_2)) = d_H(\pi_1, \pi_2)$, and the convexity property, i.e., M is convex if and only if $T_{\pi^*}(M)$ is convex. Therefore, Θ_f also remains invariant w.r.t T_{π^*}-mappings. This means that, without loss of generality, we may assume that the origin $\pi_0 = (0, \ldots, 0) \in \mathcal{A}_0^h$.

Summarizing these findings give the following theorem.

Theorem 1 (Invariance of Shape Complexity)**.** *Given a piecewise linear function $f\colon \mathbb{R}^n \to \mathbb{R}_0$ and let Λ_f be the class of ReLU DNNs $\mathcal{N}$ realizing the decision problem represented by f, i.e., to be able to distinguish between the sets $f^{-1}(0)$ and $f^{-1}(> 0)$. Then, $\Theta_{\mathcal{N}}$ for some $\mathcal{N} \in \Lambda_f$ satisfies:*

(1) $\Theta_{\mathcal{N}}$ is invariant w.r.t. architectures $\mathcal{N} \in \Lambda_f$;

(2) $\Theta_{\mathcal{N}}$ is invariant w.r.t. T_{π^*}-mappings in the activation space.

4 Tessellation-Filtering ReLU Networks

Fix $n, m \in \mathbb{N}$ and consider an r-class ReLU neural network $\mathcal{V}\colon \mathbb{R}^n \to \mathbb{R}_0^m$, given by

$$\mathcal{V}(x_1, \ldots, x_n) := (\mathcal{V}_1, \ldots, \mathcal{V}_m)^T, \tag{9}$$

where $\mathcal{V}_1, \ldots, \mathcal{V}_m$ are ReLU DNNs acting on the same domain, i.e., $\mathcal{V}_i\colon \mathbb{R}^n \to \mathbb{R}_0$. We are interested in the construction of a ReLU DNN $\mathcal{U}\colon [0, \infty)^m \to [0, \infty)$ such that for any choice of body-ReLU DNN (9) the effective tessellation resulting from its concatenation $\mathcal{U} \circ \mathcal{V}$ does not have other cells (resp., hyperplanes) than already contained in the effective tessellation $\mathrm{effTess}_h[\mathcal{V}] = \bigcup_i \mathrm{effTess}_h[\mathcal{V}_i]$, i.e.,

$$\mathrm{effTess}_h[\mathcal{U} \circ \mathcal{V}] \subseteq \mathrm{effTess}_h[\mathcal{V}]. \tag{10}$$

Condition (10) enforces that the 0-preimage of $\mathcal{U}$ does not introduce an additional intersection with any of the half-spaces of $\mathrm{Tess}_h[\mathcal{V}]$. This can only be the case if

$$\begin{aligned} y_i > 0 \quad &\text{and} \quad (y_1, \ldots, y_i, \ldots, y_m) \in \mathcal{U}^{(-1)}(0) \\ \implies \quad &\forall \widetilde{y}_i \geq 0 : \quad (y_1, \ldots, \widetilde{y}_i, \ldots, y_m) \in \mathcal{U}^{(-1)}(0). \end{aligned}$$

This means that $\mathcal{U}^{(-1)}(0)$ is enforced to be the union of sets of the form

$$F_1 \times \ldots \times F_j \times \ldots \times F_m, \tag{11}$$

where F_j is either $\{0\}$ or $[0, \infty)$, $j \in \{1, \ldots, m\}$. All together we obtain the form

$$\mathcal{U}^{(-1)}(0) = \bigcup_{\psi \in \Psi \subseteq \{0,1\}^m} F_{\psi}, \tag{12}$$

where $F_{\psi} = F_1^{(\psi)} \times \ldots \times F_m^{(\psi)}$ and $\psi \in \Psi \subseteq \{0, 1\}^m$ and

$$F_i^{(\psi)} = \begin{cases} \{0\} & \ldots \quad \psi_i = 0, \\ [0, \infty) & \ldots \quad \psi_i = 1. \end{cases} \tag{13}$$

Note that F_{ψ} in (12) characterizes the in-notches of a ReLU DNN $\mathcal{U}$ satisfying (10). By specifying the collection Ψ of in-notches by means of (13) and utilizing (4), this allows us to construct

$$\mathcal{U}_{\Psi}(y_1, \ldots, y_m) := \min_{\psi \in \Psi} \left\{ \sum_{i=1}^{m} (1 - \psi_i)\, y_i \right\}. \tag{14}$$

Let us denote by $\mathrm{tessINV}_m$ the set of tessellation-filtering ReLU DNNs (14) on $[0, \infty)^m$. Since a coordinate-wise

lower-relation $\psi^* \leq \psi$ implies $\sum_{i=1}^m (1-\psi_i)\, y_i \leq \sum_{i=1}^m (1-\psi_i*)\, y_i$, it suffices to consider only the maximal elements $\psi \in \Psi$ to evaluate (14). Further note that if $\psi_{i_0} = 1$ for all $\psi \in \Psi\}$ the convex hull of $(\mathcal{U}_\Psi)^{(-1)}(0)$ has dimension lower than $m-1$. To avoid redundancy, therefore we require that Ψ to satisfy $\max\{\psi_i | \, \psi \in \Psi\} = 1$ for all $i = 1, \ldots, m$. In graph theory two sets of vertices are called *independent* if no two vertices in the set are adjacent. For details see [Kahn and Park, 2022]. Let $\mathcal{I}_m$ denote the collection of maximally independent sets in the corresponding discrete hypercube. Then, considering only maximal vertices such that the maximum yields the upper vertex $(1, \ldots, 1)$ means that $\Psi \in \mathcal{I}_m$. This establishes a one-to-one correspondence between tessINV$_r$ and $\mathcal{I}_m$. Due to [Kahn and Park, 2022], $|\mathcal{I}_m|$ is asymptotically $2\, m\, 2^{2^m/4}$, which quantifies the diversity, hence expressiveness, of the special subclass of tessellation-filtering ReLU DNNs in m variables.

As Ψ consists of maximal elements, as a byproduct we get a characterization of the In-Notch Number of its induced decision surface in Theorem 2.

Theorem 2 (In-Notch Number of $\mathcal{U} \in$ tessINV$_m$)**.** *Let $\Psi \in \mathcal{I}_r$ and $\mathcal{U}_\Psi \in$ tessINV$_m$, then*

$$\Theta^\iota_{\mathcal{U}_\Psi} = |\Psi|. \tag{15}$$

A tessellation-filtering ReLU DNN has the special property that its global In-Notch Number equals its Local In-Notch Number, which turns out to be characteristic.

Theorem 3 (*tessINV$_m$* Characterization)**.** *For a ReLU DNN $\mathbb{N} : [0,\infty)^m \to [0,\infty)$, $m \in \mathbb{N}$, we have $\mathcal{N}$ is tessellation-filtering if and only if $\Theta^\iota_{\mathcal{N}}(0) = \Theta^\iota_{\mathcal{N}}$.*

Proof. (11) implies that for all in-notches ν of a tessellation-filtering $\mathcal{N}$ we have

$$\mathrm{conv}_H[\nu \cup \{0\}] \subset \mathcal{N}^{-1}(0). \tag{16}$$

But $\Theta^\iota_{\mathcal{N}}$ can only be greater than $\Theta^\iota_{\mathcal{N}}(0)$ if there are in-notches not satisfying (16). □

Decomposing now the ReLU DNN in a top and a body network, i.e., $\mathcal{N} = \mathcal{N}'' \circ \mathcal{N}'$, and applying the tropical Tess_h reconstruction (6) on $\mathcal{N}''$, we obtain a local representation of $\mathcal{N}$ at $x_0 \in \mathcal{N}^{-1}(0)$, which allows us to capture non-convexity characteristics of the decision surface. To simplify notation, we write $\pi'' := \pi^h_{\mathcal{N}''}$, $\pi' := \pi_{\mathcal{N}'}$ and $\mathcal{Z} := \{\pi''(x) \mid \mathcal{N}''(x) = 0\}$.

Let us partition $\mathcal{Z} = E \cup G$ via

$$\begin{aligned} E &:= \{v \in \mathcal{Z} \mid \mathrm{conv}_H[\pi''(x_0), v] \subseteq \mathcal{Z}\}, \\ G &:= \mathcal{Z} \setminus E. \end{aligned} \tag{17}$$

Applying (8), $T_{\pi''(x_0)}$ induces the tessellation-filtering ReLU DNN $\mathcal{U}_\Psi$ given by

$$\mathcal{U}_\Psi(v) = 0 \iff v \in T_{\pi''(x_o)}(E). \tag{18}$$

Note that equation (18) also holds on the Hamming ball $B_{d^*}(\pi''(x_0))$ that does not contain activation patterns from G. (18) can be applied to $\mathcal{N}$ by restricting x to the activation region $R := \{x \mid \pi'(x) = \pi''(x_0)\}$, giving Theorem 4.

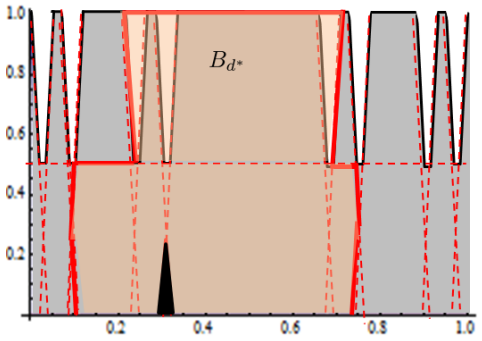

Figure 4: Ball B_{d^*} centered at the marked cell for Example 1 for $k = 3$. The ball covers non-convexity regions according to the structure of (12), which allows the application of lattice-based computations (20) in order to compute the local shape complexity as outlined. The red dotted lines mark the effective h-tessellation. The white cells (out-notches) have $d^* = 0$, while the cells on the bottom (like the black marked) have highest $d^* = 3$, indicating lower shape complexity. See Figure 5 for plotted shape complexities.

Theorem 4 (Local Representation Theorem)**.** *Given the ReLU DNN $\mathcal{N} = \mathcal{N}'' \circ \mathcal{N}'$ and $x_0 \in \mathcal{N}^{-1}(0)$, then there is a $\mathcal{U}_\Psi \in$ tessINV$_m$ (m is the number of half-spaces in $\mathrm{Tess}_h(\mathcal{N}'')$) satisfying*

$$\mathcal{N}(x) = 0 \iff \mathcal{U}_\Psi \circ T_{\pi(x_0)}(\pi(x)) = 0 \tag{19}$$

for π given by the h-tessellation $\{h_i \circ \mathcal{N}' \mid h_i \in \mathrm{Tess}_h(\mathcal{N}'')\}$ and $x \in R \cap B_{d^*}(\pi(x_0))$,

$$d^* = \min\{d_H(v, \pi(x_0)) | \ \mathrm{conv}_H[\pi(x_0), v] \not\subseteq \mathcal{Z}\}. \tag{20}$$

Note that Theorem 4 generalizes the setting of fixing all activation values for a point x_0, giving a polyhedron [Hein *et al.*, 2018]. A single cell does not have any (topological) information about the decision surface geometry. Therefore, we enlarge the view by fixing only the activation values for the body part $\mathcal{N}'$, resulting in a respectively complex region that cannot be handled easily anymore, see, e.g., [Montúfar *et al.*, 2021]. Figure 4 shows such an extended ball. Our theorem opens up a way to make this enlarged environment manageable for shape analysis. Note that already in a second layer we, in general, loose convexity as invariant property between polyhedral arrangements in the input space and the activation space. So, the h-tessellation is necessary if we want to recover convexity properties, which in our view is crucial to be able to characterize shape complexity adequately. Theorem 4 decomposes the network in to purely shape complexity part, represented by $\mathcal{U}_\Psi$, and an underlying h-tessellation. This way, replacing $\mathcal{U}_\Psi$ by $\mathcal{U}_{\Psi'}$ with less shape complexity smooths the surface, thus acting as pruning operator.

But, from a practical view, computing the h-tessellation for the whole network is infeasible and the restriction to a reasonable number of top layers is unavoidable. The choice of the top 2 or 3 layers already provides insights regarding the shape complexity. But future research is needed to analyze these aspects in more detail.

Algorithm 1 Local $\mathcal{U}_\Psi$ Approximation

1: Initialization. Set $\hat{E} = \emptyset, G = \emptyset$, compute $\pi_0 = \pi(x_0)$ and T_{π_0} due to (8); i = 1; Compute the h-tessellation by means of (6) utilizing tropical algebra [Trimmel *et al.*, 2021].
2: Choose $x_i \in D_0$ and compute $T_i = T_{\pi_0}(\pi(x_i))$.
3: Check whether $\mathrm{conv}_H[0, T_i] \subseteq \mathcal{N}^{-1}(0)$ by means of checking whether there is a $y_j \in D_1$ such that $(g_j; 1) = T_{\pi_0}(\pi(y_j))$ satisfies $(g_j; 0) \leq T_i$; if yes, then add $(g_j; 0)$ to G; choose the next candidate from D_0 and repeat step **2**;
4: If no y_j found then check whether T_i is a maximal element of E by iterating $e \in \hat{E}$ via
4.1: $M_e = \max_{k=1}^{N}(T_i - e)_k$, $m_e = \min_{k=1}^{N}(T_i - e)_k$;
4.2: replace $\{e \in E |\, M_e \geq 1, m_e \geq 0\}$ by T_i in E;
4.3: if $M_e - m_e = 2$ for all $e \in \hat{E}$, then T_i is independent to $\hat{E}$ and, therefore, add T_i to $\hat{E}$;
4.4: goto step **4.1** until all points from D_0 have been checked.
5: Take the elements of $\hat{E}$ to define Ψ (see (14)).
6: Due to (20), compute the maximal Hamming distance $d^* = \min_{g \in G}\{d_H(g, 0)\}$.

5 Algorithm and Examples

Theorem 4 gives rise to the following algorithm to identify a locally approximating $\mathcal{U}_\Psi$ in a neighborhood B at x_0. We show here the part for the top part and write $\mathcal{N}$ instead of $\mathcal{N}''$, to simplify notation. We outline an approximating algorithm that checks condition (17) by randomly selected points in a neighborhood B at x_0 by means of a sample $D_0 = \{x_1, \ldots, x_{K_0}\} \subseteq \mathcal{N}^{-1}(0) \cap B$ for class 0 and sample $D_1 = \{y_1, \ldots, y_{K_1}\} \subseteq \mathcal{N}^{-1}(> 0) \cap B$ for class 1. This way, Algorithm 1 guarantees a lower bound for the local decision surface complexity. After applying the initialization step of computing the h-tessellation, the time complexity of this algorithm is $O((K_0 + K_1)^2)$. Analogously to step **4**, we can proceed for G to extract an independent set $\hat{G} \subseteq G$ of minimal elements in G. By this we obtain $\Theta^\iota_{\mathcal{N}}(x_0) = |\hat{E}|$ and $\Theta^o_{\mathcal{N}}(x_0) = |\hat{G}|$, which are invariant to the architecture of $\mathcal{N}$. Note that also the resulting region in the input space given by $R = \bigcup_{\pi \in B_{\delta^*}(\pi(x_0))} R(\pi)$ remains invariant to the architecture of $\mathcal{N}$. This motivates us to define a shape complexity measure that is invariant of the architecture of $\mathcal{N}$ by setting

$$S(x_0) := \frac{|\hat{E}| + |\hat{G}|}{\mathrm{Vol}[R]}, \tag{21}$$

where $\mathrm{Vol}[R]$ can be approximated by $\prod_{i=1}^{n}(\rho_i^{-1} + \rho_i^{+1})$ with $\rho_i^\sigma = \sup\{\lambda \geq 0 |\, d_H(\pi(x_0 + \sigma\, \lambda\, e_i), \pi(x_0)) \leq \delta^*\}$, $\sigma \in \{-1, +1\}$.

See Figure 5 for plotted S-curves for Example 1. The top curve, $y = 1$, shows the highest S values triggered by the out-notched with $\delta^* = 0$. The curves match with our intuition that more rugged regions show higher S values. In general, for Example 1 we obtain $S((x, y)) = O(2^k)$ with k recursions and $(x, y) \in [0, 1]^2$. Note that Example 2 can be modeled by a min, which is a tessellation-filtering ReLU DNN with $\Psi = \{\sum_{j \neq i} e_j |\, i \in \{1, \ldots, n\}\}$ consisting of n independent maximal elements and there is only one out-notch given by $\sum_j e_j$. Therefore, we get $S(x) = O(n)$ for $x \in [0, 1]^n$.

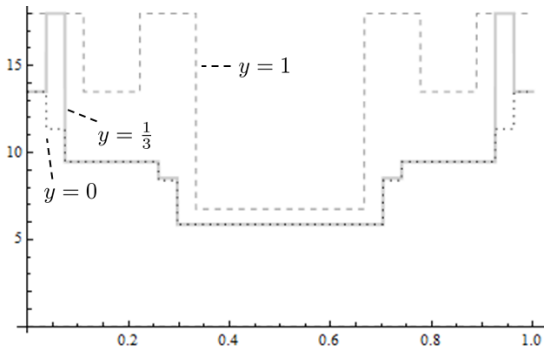

Figure 5: Local Shape Complexity S due to (21) for Example 1 for $k = 2$ and the lines $y = 1$, resp. $y = 1/3$ and $y = 0$.

6 Conclusion

Our main focus was on revealing invariant properties between tessellations induced by ReLU DNNs and their Hamming cube counterpart, and exploiting them for the purpose of geometrical analysis of decision surfaces. Several insights and research questions for the future follow from this approach: (i) the special subclass of tessellation-filtering ReLU DNNs is distinguished by special properties, for example in connection with shape geometry. It is quite rich, yet mathematically well-manageable due to its relationship to Boolean algebra; this connection, as e.g. the role of ideals for shape analysis, will be topic of future research; (ii) so does the Local Representation Theorem as one of our main results of this paper; (iii) as a byproduct we obtain tools to construct illustrative examples showing that the number of linear regions, often seen as the measure for expressiveness, is highly dependent on the architecture and can be misleading; (iv) the derivation of an architecture-independent measure for local shape complexity is a first step towards exploiting this approach for decision surface analysis. Our shape complexity measure captures local topological properties of the decision surface. Putting in oversimplified terms, it counts the number of non-convex dents. That is, our complexity measure is related to local 'smoothness' properties of neural networks, which are well known to relate to generalization. The generality of our topological characterization will pave the way towards obtaining mathematical insight and evidence for practically relevant questions such as adversarial vulnerability and also generalization capabilities.

Acknowledgements

This research was carried out under the Austrian COMET program (project S3AI with FFG no. 872172, www.S3AI.at, at SCCH, www.scch.at, and project InTribology with FFG no. 872176 at AC2T, www.ac2t.at), which is funded by the Austrian ministries BMK, BMDW, and the provinces of Upper Austria, Lower Austria, and Vorarlberg.

References

[Alfarra *et al.*, 2020] Motasem Alfarra, Adel Bibi, Hasan Hammoud, Mohamed Gaafar, and Bernard Ghanem. On

the decision boundaries of neural networks: A tropical geometry perspective. *arXiv*, 2002.08838, 2020.

[Arora *et al.*, 2018] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[Bianchini and Scarselli, 2014] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Trans. Neural Networks Learn. Syst.*, 25(8):1553–1565, 2014.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey G., D. Dunson, and M. Dudik, editors, *Proc. 14th Int. Conf. on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[Hahnloser *et al.*, 2000] Richard Hahnloser, R. Sarpeshkar, M. Mahowald, and Rodney Douglas. Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex. *Nature*, 405:947–951, 01 2000.

[Hanin and Rolnick, 2019] Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604. PMLR, 2019.

[He *et al.*, 2016] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Hein *et al.*, 2018] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *arxiv*, 1812.05720v2, 2018.

[Kahn and Park, 2022] Jeff Kahn and Jinyoung Park. The number of maximal independent sets in the Hamming cube. *Combinatorica*, March 2022.

[Montúfar *et al.*, 2014] Guido F Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[Montúfar *et al.*, 2021] Guido Montúfar, Yue Ren, and Leon Zhang. Sharp bounds for the number of regions of maxout networks and vertices of Minkowski sums. *CoRR*, abs/2104.08135, 2021.

[Peterson, 1969] F. P. Peterson. Review: Edwin H. Spanier, Algebraic topology. *Bulletin of the American Mathematical Society*, 75(5):916 – 917, 1969.

[Raghu *et al.*, 2017] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. PMLR, 06–11 Aug 2017.

[Serra *et al.*, 2018] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4558–4566. PMLR, 10–15 Jul 2018.

[Shepeleva *et al.*, 2020] Natalia Shepeleva, Werner Zellinger, Michal Lewandowski, and Bernhard Moser. ReLU code space: A basis for rating network quality besides accuracy. *ICLR 2020 Workshop on Neural Architecture Search (NAS 2020)*, arXiv preprint arXiv:2005.09903, 2020.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Szandała, 2021] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. In Akash Kumar Bhoi et al., editor, *Bio-inspired Neurocomputing*, pages 203–224. Springer Singapore, 2021.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, and et al. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9. IEEE Computer Society, 2015.

[Trimmel *et al.*, 2021] Martin Trimmel, Henning Petzka, and Cristian Sminchisescu. Tropex: An algorithm for extracting linear terms in deep neural networks. In *International Conference on Learning Representations*, 2021.

[Zhang *et al.*, 2018] Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5824–5832. PMLR, 10–15 Jul 2018.