On Space Folds of ReLU Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Recent findings suggest that the consecutive layers of ReLU neural networks can be understood geometrically as space folding transformations of the input space, revealing patterns of self-similarity. In this paper, we present the first quantitative analysis of this space folding phenomenon in ReLU neural networks. Our approach focuses on examining how straight paths in the Euclidean input space are mapped to their counterparts in the Hamming activation space. In this process, the convexity of straight lines is generally lost, giving rise to non-convex folding behavior. To quantify this effect, we introduce a novel measure based on range metrics, similar to those used in the study of random walks, and provide the proof for the equivalence of convexity notions between the input and activation spaces. Furthermore, we provide empirical analysis on a geometrical analysis benchmark (CantorNet) as well as an image classification benchmark (MNIST). Our work advances the understanding of the activation space in ReLU neural networks by leveraging the phenomena of geometric folding, providing valuable insights on how these models process input information.

1 Introduction

Neural networks are inspired by the biological structure of the brain (Rosenblatt, 1958). They achieve outstanding performance across various domains, including computer vision (Krizhevsky et al., 2012) and speech recognition (Maas et al., 2013). However, despite these impressive results, their underlying mechanisms remain poorly understood from a mathematical perspective, and current advances lack a solid foundation in rigorous mathematical analysis (Zhang et al., 2017; Neyshabur et al., 2017; Marcus, 2018; Sejnowski, 2020).

As we will discuss in this work, geometric folding can provide valuable insights and serve as a useful tool to expand our understanding of neural networks. The phenomena of geometric folding can be described as the process by which a structure undergoes a transformation from a linear or planar form into a more compact, layered configuration, where space is efficiently organized through recursive bending or folding patterns. For example, in biological systems, DNA molecules fold into complex yet highly organized shapes to fit within the confines of a cell nucleus (Dekker et al., 2013). Further, also proteins fold into precise, three-dimensional shapes, transforming from linear amino acid chains into complex structures essential for their specific functions (Crescenzi et al., 1998; Dill et al., 2008; Jumper et al., 2021). Folding has been argued to appear in neural networks, as the layering of data representations across the network depth allows for increasingly abstract, compact, and hierarchical information encoding, capturing patterns at multiple scales. It was proposed more than a decade ago, that successive layers of ReLU neural networks can be interpreted as folding operators (Montúfar et al., 2014; Raghu et al., 2017). These folds result in the replication of shapes formed by the network and contribute to understanding how the space is folded, which can help reveal symmetries in the decision boundaries that the network learns. Keup and Helias (2022) likened this process to the physical process of paper folding, where the input space is "folded" during learning. However, folding occurred in neural networks is elusive in the continuous input space. In case of protein folding, this process has been quantified using discrete mathematics on sequences of amino acids (Crescenzi et al., 1998). For neural ReLU networks, the activation space offers a possibility for further analyses of this phenomenon. In this paper, we focus on the activation space to investigate symmetries and self-similarity in the learned regions (Balestriero et al., 2024). The activation space and its related linear regions have also been used also as a measure of the network's expressivity (e.g., Montúfar et al. (2014); Raghu et al. (2017); Hanin and Rolnick (2019a)).

Insofar, the concept of space folding by neural networks remains largely qualitative, with no prior work attempting to quantify these effects. In this paper, we introduce the first method for measuring these transformations using range measures (Weyl, 1916; Moser, 2012) and discrete mathematics. Our analysis is based on a topological investigation of the activation patterns along a straight path in the activation space. In the Euclidean space the shortest path between two points is a straight line. Walking along such a path without turns monotonically increases the distance to the starting point. However, this observation no longer applies in the activation space. During the folding operation, the convexity of the created linear regions (defined in Sec. 3), may not be preserved and the Hamming distance on a straight path between two (non-adjacent) patterns can decrease. This lack of preservation inspires the introduction of our space folding measure, which measures the deviations from convexity on a straight path in both the input (Euclidean) and the activation (Hamming) spaces. In summary, our contributions in this work are as follows:

- We prove the equivalence of convexity notions between the input and activation spaces.
- We introduce a *space folding* measure to quantify local deviations from convexity in the activation space of ReLU networks. We provide both local and global versions of our measure.
- We experimentally investigate the behaviour of our measure on (*i*) CantorNet, a specially constructed synthetic example with an arbitrarily ragged decision surface, and (*ii*) ReLU networks with varying depth and width with constant number of hidden neurons trained on the MNIST benchmark.

The remainder of the paper is organized as follows: Sec. 2 details the related work; Sec. 3 recalls some basic facts and fixes notation for the rest of the paper; Sec. 4 establishes convexity results, Sec. 5 introduces the space folding measure; Sec. 6 describes the experimental results; Sec. 7 discusses the results; Sec. 8 provides concluding remarks and possible future directions for our work.

2 Related work

Activation Space. The pioneering study by Makhoul et al. has investigated partitioning the input space with neural networks. Theis work has examined 2-hidden-layer networks, thresholding neurons with the ReLU activation function (without naming it). With two hidden layers, the first layer creates hyperplanes that divide the input space into regions, adjacent if they have a Hamming distance of 1 (Makhoul et al., 1991). Connected regions are defined by a path through adjacent regions. The interest in the number of these regions was revived in 2014 by Montúfar et al., with several follow-up works, e.g., (Raghu et al., 2017; Serra et al., 2018; Xiong et al., 2020; Hanin and Rolnick, 2019a;b). The authors provided ever tighter bounds on the number of activation regions, and used them as a proxy for its expressiveness, among others.

Space Folds. The idea of folding the space has been investigated in the computational geometry (Demaine et al., 2000). Demaine and Demaine surveyed the phenomenon, focusing on the type of object being folded, e.g., paper, or polyhedra. Bern and Hayes explored whether a given crease pattern can be folded into a flat origami (non-crossing polygons in 2D with layers). Later, Bern and Hayes showed that any compact, orientable, piecewise-linear 2-manifold with a Euclidean metric can achieve this structure. In Montúfar et al. (2014) in Section 2.4, the authors briefly mentioned the folding phenomena, although through the lens of linear regions. They argue that each hidden layer in a neural network acts as a folding operator, recursively collapsing input-space regions. This folding depends on the network's weights, biases, and activation functions, resulting in input regions that vary in size and orientation, highlighting the network's flexible partitioning. In Phuong and Lampert (2020), in the Appendix A.2 the authors explored the folding operation by ReLU neural networks, but leave the exploration quite early on. In Keup and Helias (2022), the authors argued that it is through the folding operation that the neural networks arrive at their approximation power.

Self-Similarity and Symmetry. Self-similarity and symmetry are related but distinct concepts, often found in nature, mathematics, and physics. Self-similarity means that a structure or pattern looks similar to itself at different scales, and is also present in numerical data, e.g., images (Wang et al., 2020), audio tracks (Foote, 1999) or videos (Alemán-Flores and Álvarez León, 2004). Symmetry implies that an object or pattern is invariant under certain transformations, e.g., reflection, rotation, or translation. In the context of neural networks, in Grigsby et al. (2023), the authors describe a number of mechanisms through which hidden symmetries can arise. Their experiments indicate that the probability that a network has no hidden symmetries decreases towards 0 as depth increases, while increasing towards 1 as width and input dimension increase. Many fractal shapes, such as the Mandelbrot set (Mandelbrot, 1983) or CantorNet (Lewandowski et al., 2024), exhibit both self-similarity and certain symmetries. Moreover, both concepts relate to the folding operation: invariance under reflection (symmetry) can be equivalently understood as a space fold, and self-similarity can be interpreted as recursive folding or scaling operations that replicate the pattern across different levels. In neural network architectures, these principles can manifest through hierarchical structures, where each layer effectively "folds" information from previous layers, producing patterns that may repeat or reflect across layers or nodes (Raghu et al., 2017).

Distance Alteration. Lipschitz constant, a well established concept in mathematical analysis, bounds how much function's output can change in proportion to a change in its input. In context of neural networks, it has been linked to adversarial robustness, e.g., (Tsuzuku et al., 2018; Virmaux and Scaman, 2018), or generalization properties, e.g., (Bonicelli et al., 2022). Cisse et al. showed that the Lipschitz constant of a neural network can grow exponentially with its depth. Anil et al. observe that enforcing the Lipschitz property leads to some limitations, and show that norm-constrained ReLU networks are less expressive than unconstrained ones. The exact computation of the Lipschitz constant, even for shallow neural networks (two layers), is NP-hard (Virmaux and Scaman, 2018). Finally, Hanin et al. prove that the expected length distortion slightly shrinks for ReLU networks with standard random initialization, building on the results of Price and Tanner. While important, none of the aforementioned work touch on the activation space of neural networks, nor do they investigate monotonicity of a mapped straight line. Our analysis goes beyond the concept of the Lipschitz constant by investigating the input space convolution under a neural network.

3 Preliminaries

We define a *ReLU neural network* $\mathcal{N} : \mathcal{X} \to \mathcal{Y}$ with the total number of N neurons as an alternating composition of the ReLU function $\sigma(x) := \max(x, 0)$ applied element-wise on the input x, and affine functions with weights W_k and biases b_k at layer k. An input $x \in \mathcal{X}$ propagated through $\mathcal{N}$ generates non-negative activation values on each neuron. A *binarization* is a mapping $\pi : \mathbb{R}^N \to \{0,1\}^N$ applied to a vector $v = (v_1, \ldots, v_N) \in \mathbb{R}^N$, resulting in a binary vector by clipping strictly positive entries of v to 1, and non-positive entries to 0, that is $\pi(v_i) = 1$ if $v_i > 0$, and $\pi(v_i) = 0$ otherwise. In our case, the vector v is the concatenation of all neurons of all hidden layers, called an *activation pattern*, and it represents an element in a binary hypercube $\mathcal{H}_N := \{0,1\}^N$ where the dimensionality is equal to the number N of hidden neurons in network $\mathcal{N}$. A *linear region* is an element of a partition covering the input domain where the network behaves as an affine function (Montúfar et al., 2014) (see Fig. 1, left). The Hamming distance, $d_H(u,v) := |\{u_i \neq v_i \text{ for } i = 1, \ldots, N\}|$, measures the difference between $u, v \in \mathcal{H}_N$.

4 Convexity

Convexity is a key concept in computational geometry and plays a critical role in various computer engineering applications, such as robotics, computer graphics, and optimization (Boissonnat and Yvinec, 1998). In Euclidean space, convexity can be defined as a property of sets that are closed under convex combinations, where the set contains all line segments between any two points within it (Roy and Stell, 2003). We extend this notion of convexity to the Hamming space as follows.

Definition 1 (Adapted from Moser et al. (2022))**.** *A subset S of the Hamming cube $\mathcal{H}^n$ is convex if, for every pair of points $x, y \in S$, all points on every shortest path between x, y are also in S.*

Example 1. *Consider points $\pi_1 = (000)$ and $\pi_4 = (111)$ (Fig. 1, right). Then the Hamming distance $d_H(\pi_1, \pi_4) = 3$. Every shortest path consists of three edges, flipping one "bit" at a time, and thus a convex set is the whole Hamming cube $\mathcal{H}^3$.*

Example 2. *Consider activation patterns $\pi_4 = (0111), \pi_5 = (0001), \pi_6 = (1011)$ as shown in Fig. 2 (see Appendix A for more details). For a walk through any two of the activation patterns, (1) $\pi_4 \to \pi_5$, (2) $\pi_5 \to \pi_6$,*

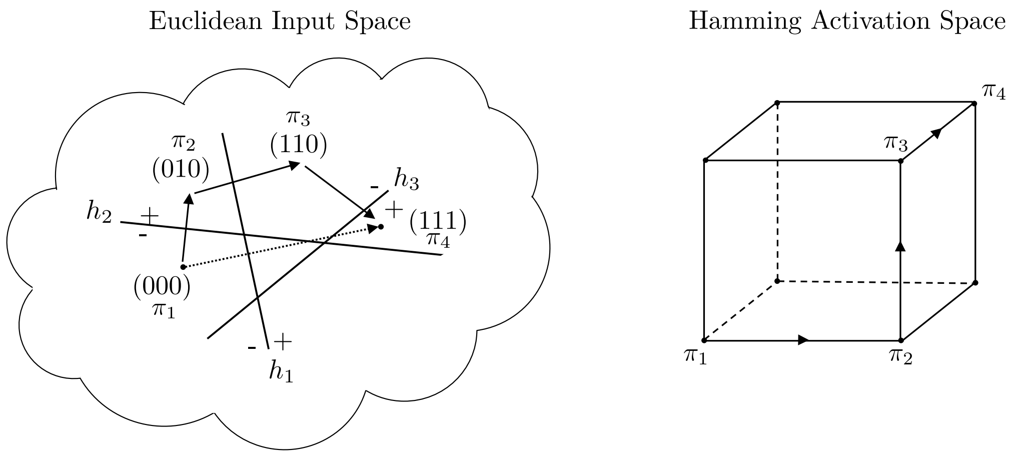


Figure 1: Illustration of a walk on a straight path in the Euclidean input space and the Hamming activation space. Left: the dotted line represent the shortest path in the Euclidean space. The arrows represent *a* shortest path in the Hamming distance between activation patterns π_1 and π_4 (note that in the Hamming space the notion of the shortest path becomes ambiguous). Right: The illustration of a shortest path connecting π_1 and π_4 in the Hamming activation space.

(3) $\pi_4 \to \pi_6$, there are intermediate, non-observable activation patterns that we traverse. They are, respectively: (1) $\pi_{non-obs} = \{(0011), (0101)\}$, (2) $\pi_{non-obs} = \{(0011), (1111)\}$, (3) $\pi_{non-obs} = \{(0011), (1001)\}$. Observe that none of them is contained in $\{\pi_4, \pi_5, \pi_6\}^C$ (the complement is taken on $[0,1] \times [0,1]$ with observable activation patterns $\{\pi_1, \dots, \pi_6\}$). Hence, the activation patterns $\{\pi_4, \pi_5, \pi_6\}$ form a convex set in the Hamming cube sense.

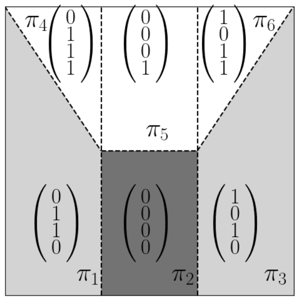

π_4 $\begin{pmatrix}0\\1\\1\\1\end{pmatrix}$
$\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$
$\begin{pmatrix}1\\0\\1\\1\end{pmatrix}$ π_6
π_5
$\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$ π_1
$\begin{pmatrix}0\\0\\0\\0\end{pmatrix}$ π_2
$\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$ π_3

Figure 2: Activation patterns π_i of recursion-based representation of CantorNet (see Appendix A). We skip neurons with unchanged values. (Adapted from Lewandowski et al. (2024) with the authors' approval.)

Before introducing the space folding measure, which relies on the notion of convexity, we prove the equivalance of convexity notions between the input and activation spaces for hyperplanes that intersect the entire input space. This further justifies the need of deeper layers to observe any space folding effects.

Lemma 1. *Consider a tessellation of activation regions formed by N hyperplanes $h_1, \dots, h_N$ with activation regions $R_{\pi_1}, \dots, R_{\pi_r} \subset \mathbb{R}^n$ and corresponding activation patterns $\mathcal{A} = \{\pi_1, \dots, \pi_r\}$. A union $R = \bigcup_{\pi \in \mathcal{A}} R_\pi$ of activation regions is convex in $\mathbb{R}^n$ if and only if the set $\mathcal{A}$ of corresponding activation patterns is convex in the Hamming space $\mathcal{H}^m$.*

Proof. Convexity in $\mathbb{R}^n \Rightarrow$ Convexity in Hamming space:

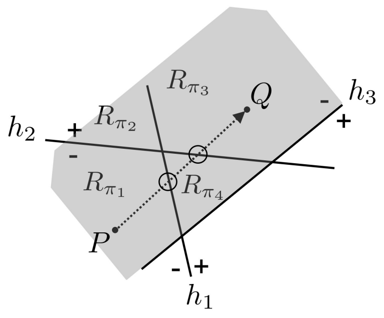


Figure 3: The shaded gray area illustrates a convex set in the Euclidean space. The hyperplanes h_1, h_2, h_3 intersect the entire input space (it holds for the hyperplanes described by neurons from the first hidden layer of a ReLU neural network). A straight line $[P, Q]$ connecting points P and Q crosses hyperplanes h_1 and h_3, resulting in a "bit" flip at a time.

Assume that the set $R = \bigcup_{\pi \in \mathcal{A}} R_\pi$ is convex in $\mathbb{R}^n$. We want to show that the set $\mathcal{A}$ is convex in the Hamming space. We start with showing the connectivity of $\mathcal{A}$. Let $\pi_i, \pi_j \in \mathcal{A}$ be any two activation regions, and choose any points $P \in R_{\pi_i}$ and $Q \in R_{\pi_j}$ in respective activation regions. Since R is convex, the line segment $[P, Q]$ lies entirely within R. As we move along $[P, Q]$, we may cross hyperplanes h_k where the activation state changes. Each such crossing corresponds to flipping exactly one bit in the activation pattern (see Fig. 3). This sequence of bit flips forms a path in the Hamming space from π_i to π_j, showing that $\mathcal{A}$ is connected. Let us now show the convexity of $\mathcal{A}$. Assume, for contradiction, that $\mathcal{A}$ is not convex in the Hamming space. Then, there exists a shortest path γ in the Hamming space connecting π_i and π_j that leaves $\mathcal{A}$; that is, some activation patterns along γ are not in $\mathcal{A}$. However, from connectivity, the path corresponding to the line segment $[P, Q]$ stays entirely within $\mathcal{A}$, as it corresponds to activation patterns of points within R. Since $[P, Q]$ is a straight line, it corresponds to a minimal sequence of bit flips (i.e., a shortest path in the Hamming space). Therefore, there exists a shortest path within $\mathcal{A}$, contradicting the assumption. Hence, $\mathcal{A}$ is convex in the Hamming space.

Convexity in Hamming space $\Rightarrow$ Convexity in $\mathbb{R}^n$:

Now assume that the set $\mathcal{A}$ of activation patterns is convex in the Hamming space. We want to show that the union $R = \bigcup_{\pi \in \mathcal{A}} R_\pi$ is convex in $\mathbb{R}^n$. Let $P, Q \in R$ be any two points, and denote by $R_{\pi_1}, R_{\pi_3} \in \mathcal{A}$ their activation patterns. Consider the line segment $[P, Q]$ in $\mathbb{R}^n$. As we move from P to Q, we may cross hyperplanes h_k, changing activation patterns. Each crossing of a hyperplane h_k corresponds to flipping a bit in the activation pattern, forming a path in the Hamming space from π_P to π_Q, as previously (again, see Fig. 3). Since $\mathcal{A}$ is convex in the Hamming space, all shortest paths between π_P and π_Q remain within $\mathcal{A}$, and in particular, the sequence of activation patterns along $[P, Q]$ is such a shortest path. Therefore, all activation patterns along $[P, Q]$ are in $\mathcal{A}$. Since every point along $[P, Q]$ has an activation pattern in $\mathcal{A}$, it lies within R. Thus, $[P, Q] \subset R$, showing that R is convex in $\mathbb{R}^n$. $\square$

Lemma 1 is important for two reasons. First, its direct consequence is that the space folding effects are observable only in networks with at least two hidden layers. Moreover, it proposes another angle to see the classic XOR problem (Minsky and Papert, 1969), and why one layer is insufficient for class separation. Second, observe limitations of Lemma 1: We assume that along a walk on a shortest path (in the Hamming sense) between any activation patterns corresponding to linear regions in a convex arrangement $\mathcal{A}$ in the Euclidean input space, the Hamming distance of adjacent linear regions differs by one, which only holds in case of hyperplanes that intersect the entire input space, and such hyperplanes are described by the 1^{st} hidden layer of a ReLU neural network (e.g., Raghu et al. (2017)). For deeper layers, there may appear activation regions, which, although neighboring, may have the Hamming distance exceeding one (see Example 2).

5 Analysis in the Activation Space

Range Measures. We proceed to introduce a space folding measure, which is inspired by the construction of range measures of random walks. Consider a walk along a line given by the sequence $s = (s_k)_{k=0}^N$ of N steps of length s_i at i^{th} step. At each step i the walk can go either up or down. The maximum absolute route amplitude of the walk is given by $r_A(s) := \max_{n \leq N} |\sum_{j=0}^n s_j|$, dependent on the choice of the coordinate system. However, we can remove this dependency by considering $r_D(s) := \max_{n \leq N}\{\sum_{k=0}^n s_k, 0\} - \min_{n \leq N}\{\sum_{k=0}^n s_k, 0\} = \max_{k \leq n \leq N} |\sum_{j=k}^n s_j|$, which represents the diameter of the walk invariant under translational coordinate transformations (Moser, 2014). Both $r_A(s)$ and $r_D(s)$ are examples of range metrics encountered in the field of random walks in terms of an asymptotic distribution resulting from a diffusion process (Jain and Orey, 1968). It turns out that $r_A(s)$ and $r_D(s)$ are norms, $\|s\|_A$ and $\|s\|_D$, respectively (Alexiewicz, 1948). Note that $\|s\|_A \leq \|s\|_D \leq 2\|s\|_A$, stating the norm-equivalence of $\|.\|_A$ and $\|.\|_D$.

Example 3. *A simple example of a range measure is a variance estimator of a sample with n observations,* $\mathbf{x} = (x_1, \dots, x_n)$*, defined as* $\mathrm{Var}(\mathbf{x}) = \frac{1}{n(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2$*, where* $\bar{x}$ *is the sample's arithmetic average.*

The Space Folding Measure. Consider a straight line connecting two samples $\mathbf{x}_1, \mathbf{x}_2$ in the Euclidean input space realized as a convex combination $\lambda_i \mathbf{x}_1 + (1 - \lambda_i)\mathbf{x}_2$, where λ_i are equally spaced on $[0, 1]$ (the equal spacing is due to practicality, but is not necessary). Then, consider the mapping of the straight line $[\mathbf{x}_1, \mathbf{x}_2]$ to a *path* Γ in the Hamming activation space, with intermediate points $(\pi_1, \dots, \pi_n)$, $\pi_i \in \mathcal{H}^N$ under a neural network $\mathcal{N}$ (see Fig. 4, left). We consider a change in the Hamming distance at each step i

$$\Delta_i := d_H(\pi_{i+1}, \pi_1) - d_H(\pi_i, \pi_1). \tag{1}$$

We then look at the maximum of the cumulative change $\max_k \sum_{i=1}^k |\Delta_i|$ along the path Γ_n,

$$r_1(\Gamma_n) = \max_{i \in \{1, \dots, n-1\}} \sum_{j=1}^{i} (d_H(\pi_{j+1}, \pi_1) - d_H(\pi_j, \pi_1)) = \max_{i \in \{1, \dots, n-1\}} d_H(\pi_i, \pi_1). \tag{2}$$

The above expression equals to the maximum Hamming distance to the starting point reached along the path. Next, we keep track of the total distance traveled on the hypercube when following the path,

$$r_2(\Gamma_n) = \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}). \tag{3}$$

For the measure of space flatness, we consider the ratio $\phi(\Gamma_n) := r_1(\Gamma_n)/r_2(\Gamma_n)$. Equivalently, the space folding measure equals

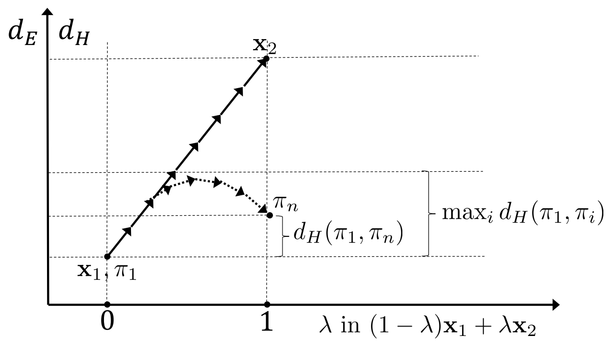
$$\chi(\Gamma_n) := 1 - \phi(\Gamma_n) = 1 - \max_{i \in \{1, \dots, n\}} d_H(\pi_i, \pi_1) \Big/ \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}). \tag{4}$$

Lemma 1 guarantees that a straight line in both the input and the activation space is convex, and χ measures the deviation from convexity along this path, effectively measuring deviation from flatness, hence its name. The higher χ is, the more folded the space is along the path Γ_n. We say that the space is flat if it is not folded, and in that sense "folding" is opposite to "flatness".

Lemma 2. *For every path* Γ_n *the space folding measure satisfies* $0 \leq \chi(\Gamma_n) \leq 1$ *(provided that* $\sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}) > 0$*, i.e., the path* Γ_n *traverses more than one region).*

Proof. We only show the upper bound as the lower is obtained in the similar way. From the triangle inequality for any activation patterns $\pi_1, \pi_i, \pi_{i+1} \in \mathcal{H}^N$ it holds that $d_H(\pi_i, \pi_{i+1}) \leq d_H(\pi_i, \pi_1) + d_H(\pi_1, \pi_{i+1})$. Writing this for every index $i \in \{1, \dots, n-1\}$, $n > 2$, and summing by sides we obtain

$$\sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}) \leq \sum_{i=1}^{n-1} d_H(\pi_i, \pi_1) + \sum_{i=1}^{n-1} d_H(\pi_1, \pi_{i+1}) \leq 2(n-1) \max_i d_H(\pi_i, \pi_1).$$


d_E d_H
$\mathbf{x}_2$
π_n
$\max_i d_H(\pi_1, \pi_i)$
$d_H(\pi_1, \pi_n)$
$\mathbf{x}_1, \pi_1$
0
1
λ in $(1-\lambda)\mathbf{x}_1 + \lambda\mathbf{x}_2$

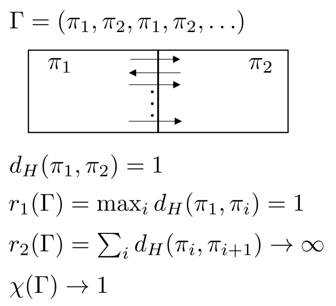

$\Gamma = (\pi_1, \pi_2, \pi_1, \pi_2, \ldots)$
π_1
π_2
$d_H(\pi_1, \pi_2) = 1$
$r_1(\Gamma) = \max_i d_H(\pi_1, \pi_i) = 1$
$r_2(\Gamma) = \sum_i d_H(\pi_i, \pi_{i+1}) \to \infty$
$\chi(\Gamma) \to 1$

Figure 4: Left: straight line between $\mathbf{x}_1$ and $\mathbf{x}_2$ in the Euclidean space. Observe that, when mapped to the Hamming activation space (dotted arrows), the path may not be straight anymore, i.e., it might happen that $d_H(\pi_1, \pi_n) < \max_i d_H(\pi_1, \pi_i)$, motivating the notion of the *devation* from convexity. Right: an extreme case when space folding $\chi(\Gamma) = 1$. Note that it is sufficient that $r_1(\Gamma) = c$ for some $c \in \mathbb{R}_+$, and that the path Γ is looped between the same regions, resulting in $r_2(\Gamma) \to \infty$. This construction, although theoretically possible, might not be realizable in practice.

Recall that $\sum_i d_H(\pi_i, \pi_{i+1}) > 0$ and divide each side by this sum. It follows that

$$\chi(\Gamma) \leq 1 - \frac{1}{2(n-1)} \leq 1.$$

□

To understand the motivation behind the construction of the measure, consider a straight path Γ in the Euclidean input space that gets mapped to a straight path in the Hamming space. In this case, the range measures r_1, r_2 increase and their ratio $r_1(\Gamma)/r_2(\Gamma) = 1$, thus there is no space folding, i.e., $\chi(\Gamma) = 0$. If a straight path in the Euclidean path gets mapped to a curved path in the Hamming activation space (Fig. 4, left), we observe non-zero values of the space folding measure χ. The space folding can equal $\chi(\Gamma) = 1$ in the case presented in Fig. 4, right. Consider a path $\Gamma = (\pi_1, \pi_2, \pi_1, \pi_2, \ldots)$. Then, the range measure $r_1(\Gamma) = 1$ and $r_2(\Gamma) \to \infty$, hence $\chi(\Gamma) \to 1$. Our measure can be made global by considering the supremum over all possible paths Γ in the Hamming activation space, i.e.,

$$\Phi_{\mathcal{N}} := \sup_{\Gamma \in \mathcal{X}} \chi(\Gamma). \tag{5}$$

6 Experiments

6.1 Experimental Setup

CantorNet. We start the experimental evaluation of our measure on CantorNet, a hand-designed example inspired by the fractal construction of the Cantor set (see Appendix A for the summary). Choose recursion depth $k = 2$ and edge points $\mathbf{x}_1 = (0, \frac{3}{4}), \mathbf{x}_2 = (1, \frac{3}{4})$ of a path Γ_n. We present range measures r_1, r_2 and the space folding measure in Fig. 6. Observe that we do not need to take all the layers into account, however we might not detect space folding. Indeed, if we evaluate $\chi(\Gamma)$ for CantorNet of the recursion level $k = 1$, and consider only take the activation pattern in the first layer, we obtain $\chi(\Gamma) = 0$. However, if we include all the layers into account (removing constant neurons), then $\chi(\Gamma) = \frac{1}{2}$ (see Fig. 5).

MNIST. Further, we study the behavior of the space folding measure on ReLU neural nets trained on MNIST (LeCun et al., 1998). We keep the number of hidden neurons constant (equal 60), and we experiment with the depth and the width of the network, trying the following architectures: $2 \times 30, 3 \times 20, 4 \times 15, 5 \times$

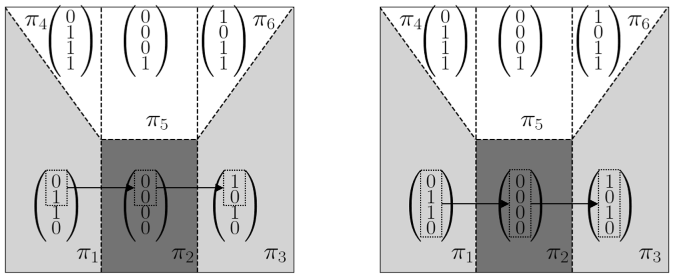


Figure 5: Activation patterns π_i of recursion-based representation of CantorNet; straight path with highlighted activations in the first layer, resp. all layers (left/right). We skip neurons with unchanged values.

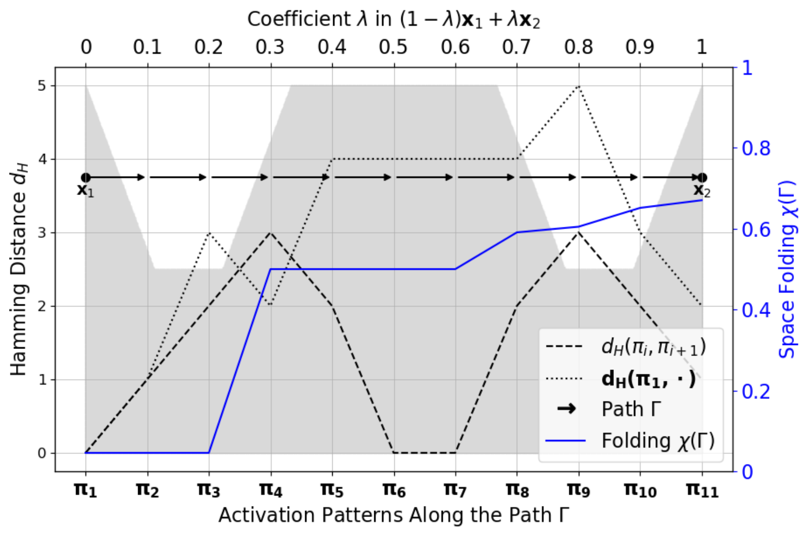


Figure 6: Behaviour of $d_H(\pi_1, \cdot)$ for respective π_i (dotted line) and distance between neighboring patterns $d_H(\pi_i, \pi_{i+1})$ (dashed line) on path Γ constructed between points $\mathbf{x}_1 = (0, \frac{3}{4})$ and $\mathbf{x}_2 = (1, \frac{3}{4})$ (indicated by arrows). Background represents CantorNet recursion-based representation at recursion level $k = 2$. The cumulative maximum (equation 2) is $d_H(\pi_1, \pi_9) = 5$. Note that the Hamming distance d_H between the initial activation pattern π_1 *can* decrease (dotted line), indicating deviations from convexity, as discussed previously. The blue line represents the space folding measure $\chi(\Gamma)$. (Best viewed in colors.)

$12, 6 \times 10$, with the notation (nb layers)$\times$(nb neurons).[1] We then train those networks for 30 epochs to high validation accuracy (≥ 0.9), and store their parameters. We present the results in Fig. 7.

6.2 Results

In Figure 6, we have shown an interesting phenomenon. We first mapped a straight-line walk from the Euclidean input space to the Hamming activation space. During this mapping, while walking along the mapped path, we have sometimes observed a *decrease* in the Hamming distance with respect to the initial input point, while in the input space, the Euclidean distance is increasing. This indicates that there is a replication of the activation pattern along the path in the activation space, which we call *folding*. This is important, as it allows us to understand how neural networks transform and compress input data, revealing

[1]As underpinned by Lemma 1, we do not expect to see folding effects in a network with 1 hidden layer and 60 neurons and hence we omit it. We remark that it might be interesting to investigate the folding effects for significantly deeper networks (in our context and keeping the number of neurons constant, that would be architectures $10 \times 6, 12 \times 5, 15 \times 4, 20 \times 3, 30 \times 2$).

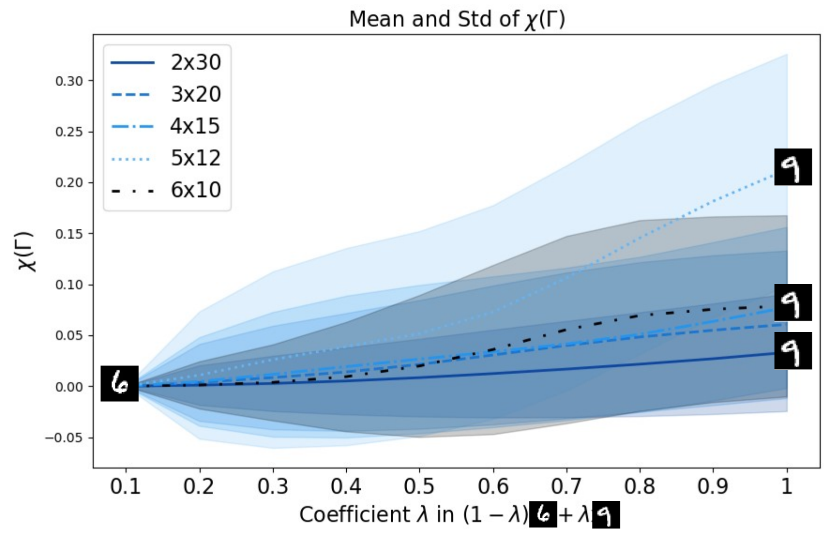


Figure 7: The mean and std of space folding measures $\chi(\Gamma)$ for five different ReLU networks trained to a low generalization error on MNIST. The path Γ is constructed between every pair of images of digits "6" and "9" from the test set. The displayed results are aggregated over all paths ($\sim 1M$ paths). Note the increasing behaviour of the measure: for shallow architectures (two and three layers) its values are significantly lower than for the deeper architectures (four to six layers). For the chosen classes, the network with five hidden layers features the highest value of space folding. (Best viewed in colors.)

the intrinsic geometric properties of the network's activation space. In the next section, we provide more discussions and hypothesis for these results.

7 Discussion

In our experiments, we have tried various pairs of digits (intra- and inter-class) and observed qualitatively similar behavior across tested architectures: the max value of χ increases with the depth of the network, reaching max for the network with five hidden layers, and then decreases with a lower max value for the network with six hidden layers. We hypothesize that the maximal value of χ is associated with the network's generalization capacity, as depth has been shown to be necessary (but insufficient) for generalization in neural networks by enabling deeper layers to learn hierarchical features (Telgarsky, 2016). So far, we have proposed a theoretical measure inspired by geometrical folding occurring in biological structures (e.g., DNA, proteins), and we have proposed to use the Hamming activation space as a counterpart to discrete spaces used for the quantitative analysis of folding occuring in those biological structures. We summarize our findings as follows.

The question:

Given a space folding value $\chi(\Gamma) = \tau > 0$ for some path Γ, what can we learn?

The answer:

We propose to see the space folding measure χ as a *feature* of a neural network. It tells us how twisted the Euclidean input space becomes under a neural network. It is upper and lower bounded, thereby it may serve as a reference point for various neural networks. As we have seen, on neural networks with a high degree of self-similarity (CantorNet), it can reach values close to 0.8, whereas for a simple neural network trained on MNIST to high validation accuracy its values are significantly lower.

The next step of our study is to empirically investigate various classes of (pre-trained) models $\mathcal{N}$, and record their global space folding measure, Φ, defined in equation 5.

We note that while computing the space folding measure for large architectures may seem infeasible due to the potentially high number of linear regions, this should not be a limitation, as actual counts of linear regions are often much lower than theoretical bounds suggest (Hanin and Rolnick, 2019b). Additionally, since the measure is computed along a path, the number of intermediate steps can be adjusted to keep computation manageable.

8 Conclusions

We have proposed a novel method to measure deviations from convexity between a walk on a straight line in the Euclidean input space and its mapping to the Hamming activation space. Further, we have used the introduced measure to analyze deviations from convexity under a synthetic example, CantorNet, and ReLU neural networks trained on MNIST. Our work highlights the convexity transformation of the input space by a ReLU neural network.

References

Alemán-Flores, M. and Álvarez León, L. (2004). Video segmentation through multiscale texture analysis. In Campilho, A. and Kamel, M., editors, *Image Analysis and Recognition, ICIAR 2004, Lecture Notes in Computer Science*, volume 3212, pages 339–346. Springer, Berlin, Heidelberg.

Alexiewicz, A. (1948). Linear functionals on denjoy-integrable functions. *Colloq. Math.*, 1:289–293.

Anil, C., Lucas, J., and Grosse, R. (2019). Sorting out Lipschitz function approximation. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 291–301. PMLR.

Balestriero, R., Humayun, A. I., and Baraniuk, R. (2024). On the geometry of deep learning. *arXiv preprint arXiv:2408.04809*.

Bern, M. and Hayes, B. (1996). The complexity of flat origami. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–183, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Bern, M. and Hayes, B. (2008). Origami embedding of piecewise-linear two-manifolds. In Laber, E. S., Bornstein, C., Nogueira, L. T., and Faria, L., editors, *LATIN 2008: Theoretical Informatics*, pages 617–629, Berlin, Heidelberg. Springer Berlin Heidelberg.

Boissonnat, J.-D. and Yvinec, M. (1998). *Algorithmic Geometry.* Cambridge University Press.

Bonicelli, L., Boschini, M., Porrello, A., Spampinato, C., and Calderara, S. (2022). On the effectiveness of lipschitz-driven rehearsal in continual learning. *Advances in Neural Information Processing Systems*, 35:31886–31901.

Cantor, G. (1883). Über unendliche, lineare punktmannigfaltigkeiten. *Math. Annalen*, 21(4):545–591.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. (2017). Parseval networks: Improving robustness to adversarial examples. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 854–863. PMLR.

Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., and Yannakakis, M. (1998). On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–465.

Dekker, J., Marti-Renom, M. A., and Mirny, L. A. (2013). Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. *Nature Reviews Genetics*, 14(6):390–403.

Demaine, E. D. and Demaine, M. L. (2005). A survey of folding and unfolding in computational geometry. In Goodman, J. E., Pach, J., and Welzl, E., editors, *Combinatorial and Computational Geometry*, volume 52 of *Mathematical Sciences Research Institute Publications*, pages 167–211. Cambridge University Press, Cambridge, UK.

Demaine, E. D., Demaine, M. L., and Lubiw, A. (2000). Folding and cutting paper. In Akiyama, J., Kano, M., and Urabe, M., editors, *Discrete and Computational Geometry*, pages 104–118, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dill, K. A., Ozkan, S. B., Shell, M. S., and Weikl, T. R. (2008). The protein folding problem. *Annual Review of Biophysics*.

Foote, J. (1999). Visualizing music and audio using self-similarity. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, pages 77–80.

Grigsby, E., Lindsey, K., and Rolnick, D. (2023). Hidden symmetries of ReLU networks. In *International Conference on Machine Learning*. PMLR.

Hanin, B., Jeong, R., and Rolnick, D. (2021). Deep relu networks preserve expected length. In *International Conference on Learning Representations*.

Hanin, B. and Rolnick, D. (2019a). Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pages 2596–2604. PMLR.

Hanin, B. and Rolnick, D. (2019b). Deep relu networks have surprisingly few activation patterns. In *NeurIPS*.

Jain, N. and Orey, S. (1968). On the range of random walk. *Israel J. Math.*, 6:373–380.

Jumper, J. et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*.

Keup, C. and Helias, M. (2022). Origami in n dimensions: How feed-forward networks manufacture linear separability. *arXiv preprint arXiv:2203.11355*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lewandowski, M., Eghbal-zadeh, H., and A.Moser, B. (2024). Cantornet: A sandbox for testing topological and geometrical measures. In *NeurIPS Workshop On Symmetry and Geometry in Neural Networks*, Proceedings of Machine Learning Research. PMLR.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Makhoul, J., El-Jaroudi, A., and Schwartz, R. (1991). Partitioning capabilities of two-layer neural networks. *IEEE Transactions on Signal Processing*, 39(6):1435–1440.

Makhoul, J., Schwartz, R., and El-Jaroudi, A. (1989). Classification capabilities of two-layer neural nets. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 635–638 vol.1.

Mandelbrot, B. B. (1983). *The Fractal Geometry of Nature*. Macmillan.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv*.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *NeurIPS*, volume 27.

Moser, B. A. (2012). Geometric characterization of weyl's discrepancy norm in terms of its n-dimensional unit balls. *Discrete Comput. Geom.*, 48(4):793–806.

Moser, B. A. (2014). The range of a simple random walk on z: An elementary combinatorial approach. *Electron. J. Comb.*, 21.

Moser, B. A., Lewandowski, M., Kargaran, S., Zellinger, W., Biggio, B., and Koutschan, C. (2022). Tessellation-filtering relu neural networks. *IJCAI*.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017). Exploring generalization in deep learning. *Advances in neural information processing systems*, 30.

Phuong, M. and Lampert, C. H. (2020). Functional vs. parametric equivalence of relu networks. In *International Conference on Learning Representations*.

Price, I. and Tanner, J. (2021). Trajectory growth lower bounds for random sparse deep relu networks. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1004–1009.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. *ICML*.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Roy, A. J. and Stell, J. G. (2003). Convexity in discrete space. In Kuhn, W., Worboys, M. F., and Timpf, S., editors, *Spatial Information Theory. Foundations of Geographic Information Science*, pages 253–269, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sejnowski, T. J. (2020). The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pages 4558–4566.

Telgarsky, M. (2016). benefits of depth in neural networks. In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539, Columbia University, New York, New York, USA. PMLR.

Tsuzuku, Y., Sato, I., and Sugiyama, M. (2018). Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in Neural Information Processing Systems*, 31.

Virmaux, A. and Scaman, K. (2018). Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, volume 31.

Wang, X., Wang, K., and Lian, S. (2020). A survey on face data augmentation for the training of deep neural networks. *Neural Computing and Applications*, 32(19):15503–15531.

Weyl, H. (1916). Uber die gleichverteilung von zahlen. *Eins.Mathematische Annalen*, 77:331–352.

Xiong, H., Huang, L., Yu, M., Liu, L., Zhu, F., and Shao, L. (2020). On the number of linear regions of convolutional neural networks. In *International Conference on Machine Learning*. PMLR.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *ICLR*.

A CantorNet

CantorNet (Lewandowski et al., 2024) is a synthetic example inspired by the triadic construction of the Cantor set (Cantor, 1883). It features two representations opposite in terms of their Kolmogorov complexities, one linear in the recursion depth k, and one exponential. It is defined through the function $A : [0,1] \to [0,1] : x \mapsto \max\{-3x+1, 0, 3x-2\}$, as the *generating function* which is then nested as $A^{(k+1)}(x) := A(A^{(k)}(x))$, $A^{(1)}(x) := A(x)$. Based on the generating function, the decision manifold R_k is defined as:

$$R_k := \{(x,y) \in [0,1]^2 : y \leq (A^{(k)}(x)+1)/2\}. \tag{6}$$

The decision surface of R_k (equation 6) equals to the 0-preimage of a ReLU net $\mathcal{N}_A^{(k)} : [0,1]^2 \to \mathbb{R}$ with weights and biases defined as

$$W_1 = \begin{pmatrix} -3 & 0 \\ 3 & 0 \\ 0 & 1 \end{pmatrix}, b_1 = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}, W_2 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

and the final layer $W_L = \begin{pmatrix} -\frac{1}{2} & 1 \end{pmatrix}, b_L = \begin{pmatrix} -\frac{1}{2} \end{pmatrix}$. For recursion depth k, we define $\mathcal{N}_A^{(k)}$ as

$$\mathcal{N}_A^{(k)}(\mathbf{x}) := W_L \circ \sigma \circ g^{(k)}(\mathbf{x}) + b_L, \tag{8}$$

where $g^{(k+1)}(\mathbf{x}) := g^{(1)}(g^{(k)}(\mathbf{x})), \sigma$ is the ReLU function, and

$$g^{(1)}(\mathbf{x}) := \sigma \circ W_2 \circ \sigma \circ (W_1 \mathbf{x}^T + b_1). \tag{9}$$

We leave the equivalent representation aside as it is not of a direct interest.

B Range Measures for Networks Trained on MNIST

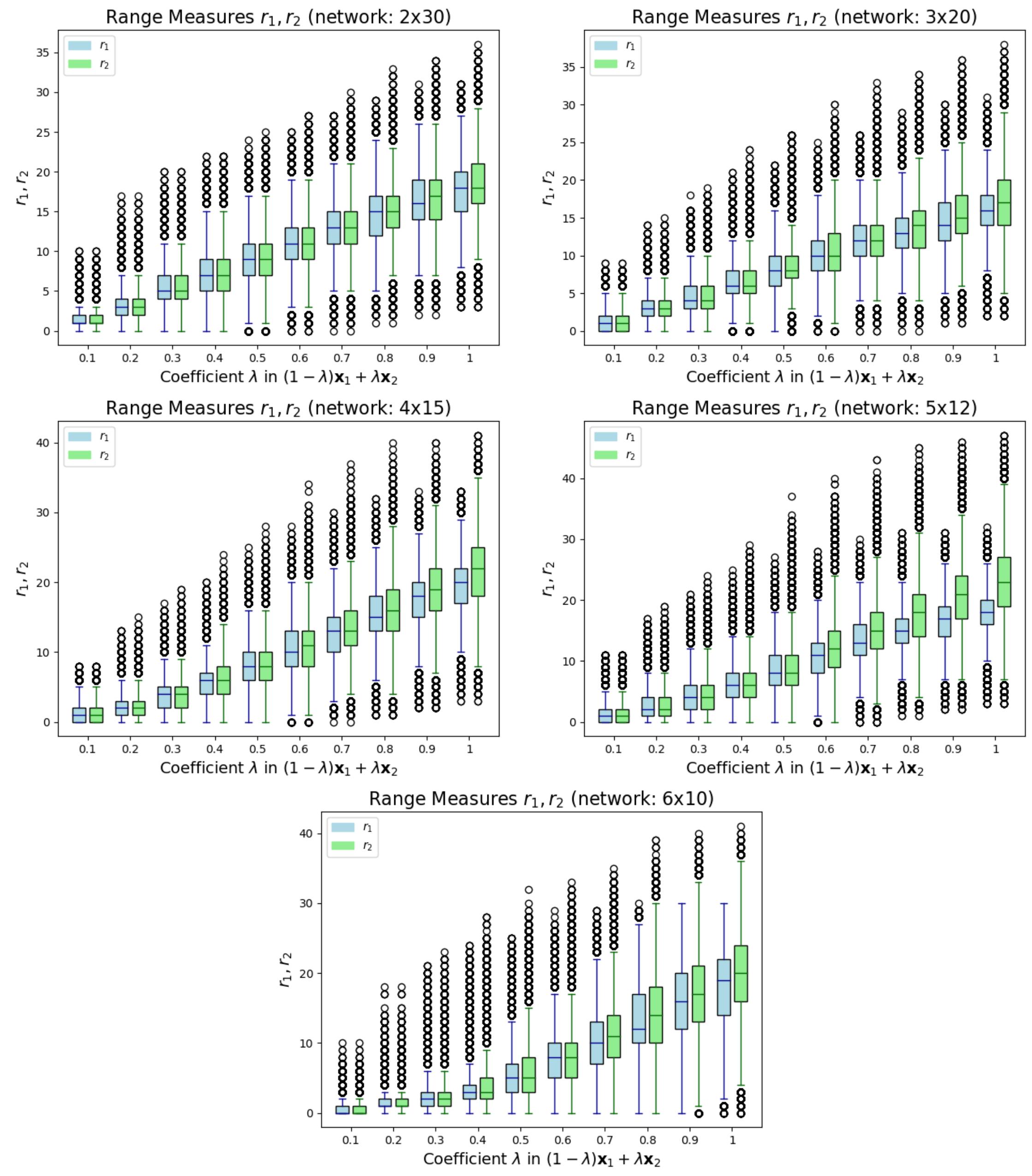


Figure 8: Range measures r_1, r_2 for ReLU networks trained on MNIST to a low generalization error for 30 epochs. Path Γ is put between images of the digit “6” and “9”, and we consider every path, thus making the results aggregated. The boxplots are constructed for paths for every pair of image of the digit “6” and “9”.