Supplementary Material for CantorNet: ReLU Networks as a Sandbox for Studying Topological Properties via the Activation Space

Michał Lewandowski[1,2] and Bernhard A.Moser[1,2]

1- Software Competence Center Hagenberg (SCCH)
Softwarepark 32a, 4232 Hagenberg, Austria

2- Institute of Signal Processing at Johannes Kepler University (JKU)
Altenberger Straße 66b, 4040 Linz, Austria

A ReLU Representation of Minimum

In this section, we present the inductive constructions of how to represent minimum function with a ReLU neural network.

Step 1. For $x_1, x_2 \in \mathbb{R}$ it holds that $\min(x_1, x_2) = x_2 + \min(x_1 - x_2, 0) = x_2 - \max(x_2 - x_1, 0)$, thus we can write

$$\min(x_1, x_2) = \mathrm{relu}(x_2) - \mathrm{relu}(-x_2) - \mathrm{relu}(-x_1 + x_2),$$

what we recover it with the following ReLU neural network architecture

$$\min(x_1, x_2) = \begin{pmatrix} 1 & -1 & -1 \end{pmatrix} \sigma \begin{pmatrix} 0 & 1 \\ 0 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Step 2. For three elements we have that $\min(x_1, x_2, x_3) = \min(\min(x_1, x_2), x_3)$, thus $\min(x_1, x_2, x_3)$ is given by

$$\begin{pmatrix} 1 & -1 & -1 \end{pmatrix} \sigma \begin{pmatrix} 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 \end{pmatrix} \sigma \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Extension to arbitrary number of variables. Denote by $\mathbf{0}_{m \times n}$ a zero matrix with m rows and n columns, and by

$$\mathbf{A} := \begin{pmatrix} 0 & 1 \\ 0 & -1 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{S} := \begin{pmatrix} 1 & -1 & -1 \end{pmatrix}.$$

Assume $\mathbf{x} \in \mathbb{R}^{2k}$ for $k \in \mathbb{N}_+$, then the last hidden layer is described by $\mathbf{S}'\sigma\left(\mathbf{A}'\mathbf{x}\right)$, where $\mathbf{S}' \in \mathbb{R}^{k\times 3k}$, $\mathbf{A}' \in \mathbb{R}^{3k\times 2k}$ are as follows

$$\mathbf{S}' := \begin{pmatrix} \mathbf{S} & \mathbf{0}_{1\times 3} & \dots & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{S} & \mathbf{0}_{1\times 3} & \dots \\ \dots & \dots & \dots & \dots \\ \mathbf{0}_{1\times 3} & \dots & \mathbf{0}_{1\times 3} & \mathbf{S} \end{pmatrix}, \ \mathbf{A}' := \begin{pmatrix} \mathbf{A} & \mathbf{0}_{3\times 2} & \dots & \mathbf{0}_{3\times 2} \\ \mathbf{0}_{3\times 2} & \mathbf{A} & \mathbf{0}_{3\times 2} & \dots \\ \dots & \dots & \dots & \dots \\ \mathbf{0}_{3\times 2} & \dots & \mathbf{0}_{3\times 2} & \mathbf{A} \end{pmatrix}.$$

For $\mathbf{x} \in \mathbb{R}^{2k+1}$ we use matrices $\mathbf{S}', \mathbf{A}'$ modified as follows

$$\mathbf{S}'' := \begin{pmatrix} \mathbf{S}' & \mathbf{0}_{k\times 1} & \mathbf{0}_{k\times 1} \\ \mathbf{0}_{1\times 3k} & 1 & -1 \end{pmatrix}, \ \mathbf{A}'' := \begin{pmatrix} \mathbf{A}' & \mathbf{0}_{3k\times 1} \\ \mathbf{0}_{1\times 2k} & 1 \\ \mathbf{0}_{1\times 2k} & -1 \end{pmatrix}.$$

Recursively applying $\mathbf{S}^*\sigma\mathbf{A}^*\mathbf{x}$ (where * means that the dimensionality must be chosen appropriately) groups the elements in pairs, reducing the problem from n to $\lceil n/2 \rceil$ elements at a time, and eventually returns the minimum element.

B Proof of Lemma 1

In this section, we re-state and proof the Lemma 1 from [Moser et al., 2022].

Lemma 1 (**Equivalence of Convexity Notions).** *An arrangement of activation regions (from the innermost layer) in Euclidean space is convex iff the corresponding activation patterns form a convex set in the activation space.*

Proof. $\Rightarrow$ 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 associated activation patterns $\mathcal{A} = \{\pi_1, \dots, \pi_r\} \subset \{0,1\}^N$. Suppose the union $R = \bigcup_j R_{\pi_j}$ is convex. We'll prove that $\mathcal{A}$ forms a convex set in the activation space.

1. Connectivity: By picking any pair of activation patterns in $\mathcal{A}$, we can show that there exists a path between them by flipping one bit at a time, implying $\mathcal{A}$ is connected.

2. Assuming non-convexity: Assuming $\mathcal{A}$ is not convex in the activation space, there exists a shortest path $\widetilde{\gamma}$ between some $\pi_{i_0}, \pi_{j_0} \in \mathcal{A}$ that is not fully inside $\mathcal{A}$. We've shown $\mathcal{A}$ is connected, so we consider a shortest path in R, γ_R, connecting points of $R_{\pi_{i_0}}$ and $R_{\pi_{j_0}}$. Walking along γ_R, we visit activation patterns and corresponding regions in R. Transitions between adjacent activation patterns flip exactly one bit. Since $\widetilde{\gamma}$ isn't in $\mathcal{A}$, there exists an activation pattern $\widetilde{\pi} \notin \mathcal{A}$.

3. Contradiction: Since the union of activation regions R is convex, we can construct a path between π_{i_0} and π_{j_0} within the convex set R, implying that $\widetilde{\pi}$ must also lie within this convex set. This contradicts our assumption that $\widetilde{\pi} \notin \mathcal{A}$. Hence, the activation patterns $\mathcal{A}$ indeed form a convex set in the activation space.

$\Leftarrow$ Suppose we have a finite collection of regions $R = \bigcup_i R_i$ where the corresponding collection $\mathcal{A}$ of activation patterns $\pi(R_i)$ is convex. Choose any two points P, Q within R and denote by γ^* the path of activation patterns corresponding to a straight line $\overline{PQ}$ connecting P, Q. Since the length $|\gamma^*| \leq |\gamma|$ is smaller or equal to any other shortest path in $\mathcal{A}$ connecting $\pi(P)$ with $\pi(Q)$, and all shortest paths lie in $\mathcal{A}$, it follows that $\gamma^* \subseteq \mathcal{A}$, and thus $\overline{PQ} \subseteq R$. $\square$

C Experiments

MNIST. We explore the relationship between the ratio of active neurons and space foldings measure for neural architectures trained on the MNIST dataset [LeCun and Cortes, 2010]. We train 20 feed forward fully connected ReLU neural networks with varying architectures, ranging from 3 hidden layers with 10 neurons each to 6 layers with 18 neurons each, in increments of 2 neurons per layer, resulting in a total of 20 different neural architectures ($3 \times 10, \ldots, 3 \times 18, 4 \times 10, \ldots, 4 \times 18, \ldots, 6 \times 18$). Each architecture is trained $N = 100$ times until achieving a minimum validation accuracy of 0.9. We use the Adam optimizer [Kingma and Ba, 2015]. Next, we calculate the space folding measure at a straight path between two images representing digits 1 and 4. This process is repeated for N different pairs of images. We then average the results and indicate the standard deviation using shades of the respective colors (Figure 1).

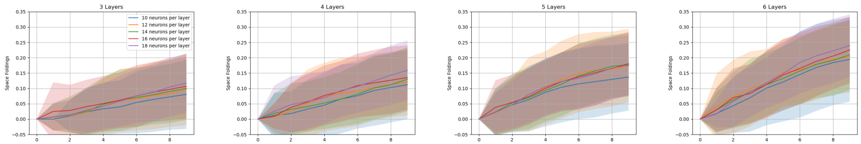

Fig. 1: MNIST dataset space foldings for various neural architectures. Figures show different hidden layer counts, while colors denote different number of neurons per layer. Best viewed in colors.

References

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*.

[LeCun and Cortes, 2010] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

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