

Supplemental: The Computational Complexity of Counting Linear Regions in ReLU Neural Networks

A Notes on Theory and Literature

A.1 Details on the encoding size

The *encoding size* $\langle n \rangle$ of a nonnegative integer n is $\langle n \rangle := \lceil \log_2(n+1) \rceil$, the encoding size of a fraction $q = a/b$ with $a \in \mathbb{Z}$ and $b \in \mathbb{N}$ is $\langle q \rangle := 1 + \langle |a| \rangle + \langle b \rangle$, and the encoding size of a matrix with rational entries $A = (a_{ij})_{i \in [n], j \in [m]}$ is $\langle A \rangle := nm + \sum_{i \in [n], j \in [m]} \langle a_{ij} \rangle$. Since we are interested in the computational complexity, we restrict ourselves to ReLU networks with rational entries that can be represented with a finite number of bits. Then, a ReLU network N of depth $d+1$ with $A^{(i)} \in \mathbb{Q}^{n_i \times n_{i-1}}$ and a vector $b^{(i)} \in \mathbb{Q}^{n_i}$ has encoding size $\langle N \rangle = \sum_{i=1}^{d+1} \langle A^{(i)} \rangle + \sum_{i=1}^{d+1} \langle b^{(i)} \rangle$. In particular, if A_{\max} denotes the maximum encoding size of an entry in any $A^{(i)}$ and $b^{(i)}$, then $\langle N \rangle \leq (d+1) \cdot (\max\{n, n_1, \dots, n_d\} + 1)^2 \cdot (1 + A_{\max}) = \text{poly}(n, s(N), A_{\max})$.

A.2 Basic properties of linear regions

The following statements hold.

Lemma A.1. *Given a ReLU network N , the set of open connected regions of N is equal to the unique set \mathcal{S} with the minimal number of open connected subsets such that $\bigcup_{S \in \mathcal{S}} \overline{S} = \mathbb{R}^n$ and f_N restricted to any $S \in \mathcal{S}$ is affine.*

Proof. First, we show that the minimal set satisfying the assumptions above is unique. Suppose that there are two distinct sets \mathcal{S} and \mathcal{S}' of open subsets that achieve the minimum. By Zanotti [2025a, Lemma 3.2], each element in $\mathcal{S} \in \mathcal{S}$ is maximal in the sense that there does not exist a nonempty set $U \subseteq \mathbb{R}^n \setminus S$ such that $S \cup U$ is open and connected, and f_N restricted to $S \cup U$ remains affine.

Since \mathcal{S} and \mathcal{S}' are distinct, there is a set $S \in \mathcal{S}$ with $S \notin \mathcal{S}'$, and since $\bigcup_{S' \in \mathcal{S}'} \overline{S'} = \mathbb{R}^n$, there is a set $S' \in \mathcal{S}'$ with $S \cap S' \neq \emptyset$. Since both S and S' are open, the intersection $S \cap S'$ is full dimensional and the affine functions on S and S' are identical. Therefore, $U = S' \setminus S \subset \mathbb{R}^n \setminus S$ is a nonempty set such that $S \cup U = S \cup S'$ is open and f_N restricted to $S \cup U$ remains affine, contradicting the maximality of S .

By Zanotti [2025a, Lemma 3.2], the unique set \mathcal{S} is then exactly the unique set of inclusion-maximal open subsets of \mathbb{R}^n such that f_N restricted to each inclusion-maximal subset is affine, which is by definition the set of open connected regions of N . \square

Lemma A.2. *Given a ReLU network, the closure of every open connected region is the closure of the union of a set of proper activation regions.*

Proof. See Hanin and Rolnick [2019, Lemma 3]. \square

A.3 Additional notes for definitions used in literature

- Raghu et al. [2017] do not use Definition 1 explicitly, but they define activation patterns and derive a bound on the total number of activation patterns, which is equivalent to bounding the number of activation regions.
- The bound of Serra et al. [2018] holds for Definition 2, compare the discussion in [Cai et al., 2023, Section 5]. The MIP counts the number of activation regions (Definition 1).
- Rolnick and Kording [2020] only treat cases where Definitions 2 and 5 are equivalent.
- Tseran and Montúfar [2021] consider activation regions of maxout networks, which is conceptually slightly different from the activation regions defined in this paper.
- Huchette et al. [2023] actually define activation regions using Definition 1, but then later state that they disregard low dimensional linear regions. Therefore, their definition is equivalent to Definition 2.

- Zanotti [2025b] defines linear regions as the closure of open connected regions.

A.4 Inaccuracies in the literature

In this section we list a few cases where misunderstandings about the different definition of a linear region led to small errors or inconsistencies in previous work, alongside with a suggestion how they would be fixable. Usually, this can be achieved by switching to a different, maybe more appropriate definition of a linear region.

1. In Lemma 11 (d), [Chen et al., 2022] claims that $(S_1 \cap S_2) \cap (S_1^\circ \cup S_2^\circ) = \emptyset$ holds for any two closed connected regions S_1, S_2 . While this claim is not true for closed connected regions, it is true that $(\overline{S_1 \cap S_2}) \cap (S_1^\circ \cup S_2^\circ) = \emptyset$ holds for any two *open* connected regions S_1, S_2 . This inaccuracy was first pointed out by Zanotti [2025a].
2. The algorithm of Wang [2022] does not create the set of closed connected regions, since in the algorithm, only closures of activation regions are merged such that (1) the affine function of the regions is identical and (2) the closures of the two activation regions have a non-empty intersection. However, a closed connected region is in general *not* a union of the closure of a set of activation regions, see Appendix C.1 for an example.
However, the algorithm of Wang [2022] can instead be adapted to count the number of *open connected regions* with one minor adjustment. Instead of merging the closure of activation regions with a nonempty intersection, merge only two proper activation regions if their closure has a $(n - 1)$ -dimensional intersection. This computation can still be done in time polynomial in the input size through linear programming, as described in Lemma A.5.
3. Lezeau et al. [2024] define a linear region as follows:

A set of a neural network f is a linear region if it is a maximal connected region (closure of an open set) on which f is linear.

We note that their definition is not equivalent to Definition 4 or 5, since a closed connected region that counts as multiple open connected regions can count as a single linear region in their definition, while a closed connected region can also count as multiple linear regions in their definition; consider, for example, the orange closed connected region in Figure 5.

In Section 6.2, they present an algorithm to estimate the number of linear regions: they compute the number of unique gradients obtained on a set of sample points. They further state that if the gradients of two points are equal but the midpoint of the two points has a different gradient, then the two original points correspond to two different linear regions. This is not always true, since linear regions can be nonconvex according to their definition. Therefore, their algorithm can also overestimate the number of linear regions, but instead always underestimates the number of proper activation regions.

A.5 Technical results

Lemma A.3. *Given a ReLU network N and an activation pattern $a \in \{0, 1\}^{s(N)}$ corresponding to a proper activation region. Let $A_{\max} \in \mathbb{Z}$ be the maximum absolute value of any numerator or denominator in an entry of the matrices $A^{(1)}, \dots, A^{(d+1)}$ and biases $b^{(1)}, \dots, b^{(d+1)}$ and let $n_{\max} = \max\{n_0, \dots, n_{d+1}\}$. Then, the encoding size of every coefficient of the affine function $f_N^a : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_N^a(x) = \sum_{i=1}^n a_i x + b$ is bounded by*

$$36d^2 n_{\max}^2 \langle A_{\max} \rangle.$$

Proof. Given an activation pattern $a \in \{0, 1\}^{s(N)}$ of N , we modify the matrices $A^{(1)}, \dots, A^{(d+1)}$ and biases $b^{(1)}, \dots, b^{(d+1)}$ by replacing the columns of matrices and bias entries corresponding to inactive neurons with 0 entries. Let $A^{(1),a}, \dots, A^{(d+1),a}$ and biases $b^{(1),a}, \dots, b^{(d+1),a}$ denote the modified matrices and biases. Then, a simple calculation shows that for all $x \in \mathbb{R}^n$, we have

$$f_N^a(x) = A^{(d+1),a} \dots A^{(1),a} x + b^{(d+1),a} + \sum_{i=0}^{d-1} A^{(d+1),a} \dots A^{(d+1-i),a} b^{(d-i),a}.$$

To bound the encoding size of all occurring coefficients, we separately give a bound on the absolute value of any occurring denominator and numerator. For this, we turn the $d + 1$ rational matrices and

biases into integral matrices and biases by bringing all fractional entries to a common denominator. The value of the common denominator is bounded by $A_{\max}^{(d+1)(n_{\max}+1)^2}$, since there are fewer than $(d+1)(n_{\max}+1)^2$ entries in the rational matrices and biases.

To bound the maximum absolute numerator value, we now consider the $(d+1)$ integral matrices and biases that arise by multiplying the fractional matrices by the common denominator. The maximum absolute value of an entry in one of the integral matrices or biases is bounded by $A_{\max}^{(d+1)(n_{\max}+1)^2}$. A simple calculation shows that the maximum absolute entry that can be obtained in the product of the $(d+1)$ integral matrices is

$$\left(A_{\max}^{(d+1)(n_{\max}+1)^2}\right)^{d+1} \prod_{i=1}^d n_i \leq n_{\max}^d \cdot A_{\max}^{(d+1)^2(n_{\max}+1)^2}.$$

The constant of f_N^a is equal to $b^{(d+1),a} + \sum_{i=0}^{d-1} A^{(d+1),a} \dots A^{(d+1-i),a} b^{(d-i),a}$ and can thus be bounded by $(d+1)n_{\max}^d A_{\max}^{(d+1)^2(n_{\max}+1)^2}$.

Since the maximum encoding size of an element of a set of integers is obtained by the integer having the maximum absolute value, it follows that the encoding size of a coefficient in f_N^a is at most

$$\begin{aligned} & 1 + \langle (d+1)n_{\max}^d A_{\max}^{(d+1)^2(n_{\max}+1)^2} \rangle + \langle A_{\max}^{(d+1)(n_{\max}+1)^2} \rangle \\ & \leq 1 + \langle d+1 \rangle + d\langle n_{\max} \rangle + (d+1)^2(n_{\max}+1)^2 \langle A_{\max} \rangle + (d+1)(n_{\max}+1)^2 \langle A_{\max} \rangle \\ & \leq 1 + \langle d+1 \rangle + d\langle n_{\max} \rangle + 2(d+1)^2(n_{\max}+1)^2 \langle A_{\max} \rangle \\ & \leq 1 + \langle d+1 \rangle + d\langle n_{\max} \rangle + 32d^2 n_{\max}^2 \langle A_{\max} \rangle \\ & \leq 36d^2 n_{\max}^2 \langle A_{\max} \rangle. \end{aligned}$$

□

Lemma A.4. *Given a ReLU network N , an activation pattern $a \in \{0, 1\}^{s(N)}$ and an index $i \in [s(N)]$ of a neuron, one can compute in time polynomial in $\langle N \rangle$ the (coefficients of the) affine function $f_i^a : \mathbb{R}^n \rightarrow \mathbb{R}$ which is computed at the output of the i -th neuron.*

Proof. The proof is analogous to the proof of Lemma A.3. □

Lemma A.5. *Given a ReLU network N and a vector $a \in \{0, 1\}^{s(N)}$, one can compute the dimension of S_a in time polynomial in $\langle N \rangle$.*

Proof. Let $I \subseteq [s(N)]$ denote the support of the activation pattern $a \in \{0, 1\}^{s(N)}$. By Lemma A.3, the encoding size of the coefficients in every affine function f_i^a are bounded polynomially in $\langle N \rangle$. Thus, we can solve a series of linear programs to compute the dimension of the polyhedron

$$P_a = \{x \in \mathbb{R}^n : f_i^a(x) \geq 0 \text{ for all } i \in I, f_i^a(x) \leq 0 \text{ for all } i \notin I\}.$$

For details, we refer to [Fukuda, 2020-07-10, Section 7.3]. Since P_a is the closure of S_a if S_a is nonempty, it follows that the dimension of P_a is equal to the dimension of S_a unless S_a is empty (in the latter case, P_a cannot be full-dimensional). The latter case is easy to recognize since for all $i \in I$, we can check if $f_i^a(x) = 0$ holds for all $x \in P_a$ by solving the linear program $\max\{f_i^a(x) : x \in P_a\}$. □

B Omitted proofs

B.1 Omitted proofs for the one hidden layer case

Let N be a ReLU network with one hidden layer and let $[n_1]$ denote the set of neurons. For ease of notation, we denote $A^{(2)} = (a_1, \dots, a_{n_1}) \in \mathbb{R}^{1 \times n_1}$ and $A^{(1)} = (w_{ij})_{i \in [n_1], j \in [n]}$. Then, the function computed by the network is

$$f_N(x) = b^{(2)} + \sum_{i=1}^{n_1} a_i \max(0, b_i^{(1)} + \sum_{j=1}^n w_{ij} x_j).$$

For every neuron $i \in [n_1]$, we define the hyperplane $H_i := \{x \in \mathbb{R}^n : b_i^{(1)} + \sum_{j=1}^n w_{ij}x_j = 0\}$ and halfspaces $H_i^+ := \{x \in \mathbb{R}^n : b_i^{(1)} + \sum_{j=1}^n w_{ij}x_j \geq 0\}$, $H_i^- := \{x \in \mathbb{R}^n : b_i^{(1)} + \sum_{j=1}^n w_{ij}x_j \leq 0\}$. The function $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ computed by a neuron $i \in [n_1]$ is $f_i(x) = a_i \max(0, b_i^{(1)} + \sum_{j=1}^n w_{ij}x_j)$.

Proof of Theorem 4.1. We only consider Definitions 3 to 6 here (for Definitions 1 and 2, the answer is trivially yes, as $n_1 > 0$). Let N be a ReLU network as defined above.

The idea is to check for every neuron if the function f_N computed by the ReLU network N has breakpoints along the hyperplane corresponding to the neuron, that is, we check if the gradient of f_N is discontinuous along the hyperplane. Since multiple neurons can correspond to the same hyperplane, it is possible that the functions of these neurons cancel such that no breakpoints along the hyperplane are introduced. First, we group the neurons according to the hyperplanes to which they correspond.

Let $g_1(x) = b_1^{(1)} + \sum_{j=1}^n w_{1j}x_j$ and let I_1 be the subset of $[n_1]$ such that

$$i \in I_1 \iff c_{1i} = w_{i1}/w_{11} = \dots = w_{in}/w_{1n} = b_i^{(1)}/b_1^{(1)}$$

holds for some constant $c_{1i} \in \mathbb{R} \setminus \{0\}$, which is equivalent to the hyperplanes H_1, H_i being identical.

Now, we derive a condition when the function f_N that is computed by the ReLU network has breakpoints along the hyperplane H_1 .

For all $i \in I_1$, we have

$$f_i(x) = a_i \cdot \max(0, c_{1i} \cdot g_1(x)) = a_i |c_{1i}| \cdot \max(0, \text{sign}(c_{1i}) \cdot g_1(x)).$$

We split I_1 into the two sets $I_1^+ = \{i \in I_1 : c_{1i} > 0\}$ and $I_1^- = \{i \in I_1 : c_{1i} < 0\}$. The sum of the functions of the neurons in I_1 is

$$\begin{aligned} & \max(0, g_1(x)) \sum_{i \in I_1^+} a_i c_{1i} - \max(0, -g_1(x)) \sum_{i \in I_1^-} a_i c_{1i} \\ &= g_1(x) \sum_{i \in I_1^-} a_i c_{1i} + \max(0, g_1(x)) \left(\sum_{i \in I_1^+} a_i c_{1i} - \sum_{i \in I_1^-} a_i c_{1i} \right) \end{aligned}$$

Thus, if

$$\sum_{i \in I_1^+} a_i c_{1i} = \sum_{i \in I_1^-} a_i c_{1i}, \quad (1)$$

the sum of all functions having H_1 as corresponding hyperplane is affine and f_N has no breakpoints along the hyperplane H_1 . Otherwise, f_N has breakpoints along the hyperplane H_1 and N has more than one linear region.

Thus, N has only a single linear region if and only if (1) holds for all $j \in [n_1]$ (replace 1 by j in (1) and define the set I_j and constants c_{ji} as before with j instead of 1), which can be verified in polynomial time. \square

Proof of Theorem 4.2. We show #P-hardness by reducing from the problem of counting the number of cells in a hyperplane arrangement, which is #P-complete (see [Linial, 1986]).

Let $\mathcal{H} = (H_i)_{i \in [m]}$ be a hyperplane arrangement in \mathbb{R}^n with $H_i := \{x : w_i^\top x = 0\}$, $w_i \in \mathbb{R}^n \setminus \{0\}$ such that each w_i appears only once. Restricted to such hyperplane arrangements, the problem of counting the number of cells remains #P-complete, see [Linial, 1986]. Given a point $x^* \in \mathbb{R}^n$ in a cell, we first orient the hyperplanes such that $w_i^\top x^* > 0$ for all $i \in [m]$. We will show that the ReLU network $N_{\mathcal{H}}$ computing the convex function

$$f_{N_{\mathcal{H}}}(x) = \sum_{i=1}^m \max(0, w_i^\top x),$$

which can be computed using one hidden layer and one neuron per hyperplane, has exactly as many linear regions as the hyperplane arrangement \mathcal{H} has cells (according to Definitions 1 to 6).

It is easy to see that the number of cells of \mathcal{H} is exactly the number of proper activation regions of $N_{\mathcal{H}}$. We now show that also the number of activation regions is equal to the number of cells of $N_{\mathcal{H}}$. Suppose for the sake of contradiction that there is an activation pattern $a \in \{0, 1\}^m$ with support $I \subseteq [m]$ such that the activation region S_a is neither full dimensional nor empty. If S_a is low dimensional, then by definition the set

$$\{x \in \mathbb{R}^n : w_i^\top x \leq 0 \text{ for all } i \in [m] \setminus I\}$$

is low dimensional and there is an index $j \in [m] \setminus I$ with $S_a \subseteq \{w_j^\top x = 0\}$. Therefore, there is a $\lambda \in \mathbb{R}_{\geq 0}^{[m] \setminus I}$ such that

$$\sum_{i \in [m] \setminus I} \lambda_i w_i = -w_j$$

holds, which leads to the contradiction

$$0 < \sum_{i \in [m] \setminus I} \lambda_i w_i^\top x^* = -w_j^\top x^* \leq 0.$$

Thus, each activation region is a proper activation region.

We now show that the affine functions on two cells cannot be equal, which proves #P-hardness also for Definitions 3 to 6.

Suppose $a, a' \subseteq \{0, 1\}^m$ are two activation patterns corresponding to distinct proper activation regions $S_a, S_{a'}$ with the same affine function. $\overline{S_a}$ cannot have a $(n-1)$ -dimensional intersection with $\overline{S_{a'}}$, since otherwise there would be exactly one hyperplane separating S_a from $S_{a'}$, and by construction, the affine functions $f_{N_{\mathcal{H}}}^a$ and $f_{N_{\mathcal{H}}}^{a'}$ must be distinct.

Therefore, $\text{conv}(S_a \cup S_{a'}) \setminus (S_a \cup S_{a'})$ is full-dimensional, and there exists a proper activation region S_{a^*} with $\dim(\text{conv}(S_a \cup S_{a'}) \cap S_{a^*}) = n$ and $\dim(\overline{S_a} \cap \overline{S_{a^*}}) = n-1$. Since $f_{N_{\mathcal{H}}}$ is convex, $f_{N_{\mathcal{H}}}(x) = f_{N_{\mathcal{H}}}^a(x)$ holds for all $x \in \text{conv}(S_a \cup S_{a'})$. Thus, the function computed on S_{a^*} must be equal to $f_{N_{\mathcal{H}}}^a$, which gives a contradiction as before.

We now show that LINEAR REGION COUNTING is in #P for Definitions 1 to 4.

A certificate for Definitions 1 and 2 is simply an activation pattern $a \in \{0, 1\}^{s(N)}$ of the ReLU network N , which can be checked in polynomial time by computing the dimension of S_a , see Lemma A.5. Thus, LINEAR REGION COUNTING is in #P for Definitions 1 and 2.

We now construct certificates for Definitions 3 and 4. Given a ReLU network N with one hidden layer, the set $\mathcal{H} = (H_i)_{i \in [m]}$ of hyperplanes that correspond to breakpoints of the function f_N can be computed in polynomial time using the procedure described in the proof of Theorem 4.1. By construction, the function f_N is affine on each cell of the hyperplane arrangement \mathcal{H} and the affine functions that are realized on two neighboring cells (cells with an $(n-1)$ -dimensional intersection) cannot be equal. Thus, each cell of the hyperplane arrangement \mathcal{H} is an open connected region.

Since each open connected region is convex, the set of convex regions of the ReLU network N is well defined and its cardinality is equal to the set of open connected regions.

A unique certificate for an open connected region (and a convex region) is now given by the hyperplane arrangement \mathcal{H} as well as a vector $a \in \{-, +\}^m$ specifying a cell $C \subset \mathbb{R}^n$ of the hyperplane arrangement \mathcal{H} , where $C \subseteq H_i^{a_i}$ for every $i \in [m]$. The certificate can be checked in polynomial time: we can verify in polynomial time if the hyperplane arrangement \mathcal{H} is defined as above, and we can verify in polynomial time if the vector a corresponds to a cell of \mathcal{H} .

It follows that LINEAR REGION COUNTING is in #P for Definitions 3 and 4. \square

B.2 Omitted proofs for more than one hidden layers

Intuition for the proof of Lemma 5.3. The proof of Lemma 5.3 improves the reduction of Wang [2022], who use a result of Katz et al. [2017, Appendix I]. Similar to Katz et al. [2017, Appendix I], we rely on the simple fact that given a SAT formula $\phi(x) = \bigwedge_{i=1}^m ((\bigvee_{j \in J_i^+} x_j) \vee (\bigvee_{j \in J_i^-} \neg x_j))$ on

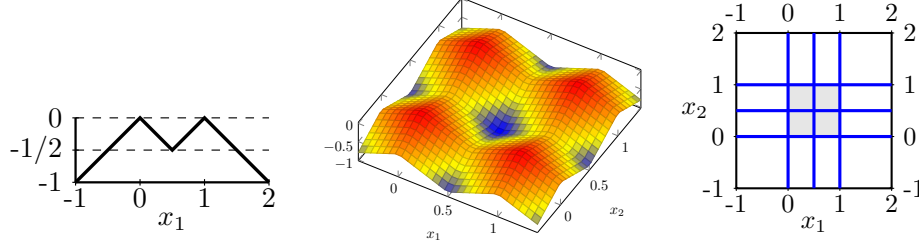


Figure 3: The function T_1 (left), T_2 (center) and its linear regions (right).

the variables x_1, \dots, x_n , the function

$$g_\phi(x) = 1 - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j))$$

takes value 1 on all satisfying (0-1) assignments, and value less than 0 on all non-satisfying assignments, which follows from the fact that $\max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j))$ evaluates to 0 for all (0-1) assignments that satisfy the i -th clause of ϕ , and to 0 for all (0-1) assignments that do not satisfy the i -th clause of ϕ . For every $i \in [m]$, J_i^+ and J_i^- are disjoint subsets of $[n]$ specifying which (negated) variables occur in the i -th clause of ϕ .

Notice that if ϕ is unsatisfiable, there is no (0-1) assignment on which g_ϕ takes value 1. As a result, for any $\varepsilon \in (0, 1)$, ϕ is satisfiable if and only if $\max(1, \varepsilon + g_\phi) - 1 = \max(0, \varepsilon - 1 + g_\phi)$ evaluates to ε on some (0-1) assignment.

This implies that if ϕ is satisfiable, the function $h_{\phi, \varepsilon} = \max(0, \varepsilon - 1 + g_\phi)$ has at least two linear regions according to Definitions 3 to 6, since $h_{\phi, \varepsilon}$ evaluates to ε for a satisfying (0-1) point, and to 0 for all points in an ε -ball around a non-satisfying (0-1) point. Since each clause in a SAT formula is not satisfied by least one (0-1) assignment, we can assume that ϕ has a non-satisfying assignment.

If for any SAT formula ϕ , the function $h_{\phi, \varepsilon}$ had strictly more than one linear region (according to Definitions 3 to 6) only if ϕ is satisfiable, then we would have a complete reduction from SAT to the problem of deciding whether a ReLU network with two hidden layers has strictly more than one linear region (according to Definitions 3 to 6), since $h_{\phi, \varepsilon}$ can be computed using a ReLU network with two hidden layers.

Unfortunately, there exists a SAT formula ψ such that $h_{\psi, \varepsilon}$ has more than one linear region although ψ is unsatisfiable, see Example C.2.

The key idea to resolve this is to add a CWPL function that is negative everywhere but on the elements of the set $\{0, 1\}^n$ (on which it evaluates to zero).

A function with this property is the function $T_n : \mathbb{R}^n \rightarrow \mathbb{R}$ with

$$T_n(x) = \sum_{i=1}^n (-\max(0, -x_i) - \max(0, x_i) + \max(0, 2x_i - 1) - \max(0, 2x_i - 2)),$$

shown in Figure 3. The proof of Lemma 5.3 shows that adding this function recovers the equivalence: A SAT formula is satisfiable if and only if the function $\max(0, T_n + \varepsilon - 1 + g_\phi)$ has more than one linear region according to Definitions 3 to 6.

For an example that visualizes the different steps of the reduction, see Example C.1.

The following lemma is easy to prove based on the plot of T_1 in Figure 3.

Lemma B.1. *For any $n \in \mathbb{N}$, the following implications hold*

$$\begin{aligned} T_n(x) \leq -\varepsilon & \iff \exists i \in [n] : x_i \in (-\infty, -\varepsilon] \cup [\varepsilon, 1 - \varepsilon] \cup [1 + \varepsilon, \infty) \\ T_n(x) = 0 & \iff x \in \{0, 1\}^n. \end{aligned}$$

Proof of Lemma 5.3. We first show that the problem is in NP. If a ReLU network N is a yes-instance of 1-REGION-DECISION, then there are two proper activation regions on which two distinct affine

functions are computed. Two activation patterns corresponding to such proper activation regions serve as a polynomial certificate of a yes-instance. Given two vectors $a, a' \in \{0, 1\}^{s(N)}$, we can verify the certificate in polynomial time. First, we check if a and a' correspond to proper activation regions by computing the dimension of S_a and $S_{a'}$ in polynomial time, see Lemma A.5. If S_a and $S_{a'}$ are proper activation regions and $f_N^a \neq f_N^{a'}$ holds, which can be checked in polynomial time (see Lemma A.4), then a, a' is a valid certificate of a yes-instance. Thus, the problem is in NP.

To show NP-hardness, we reduce the problem of deciding whether a SAT instance is satisfiable to our problem. Let $\phi(x) = \bigwedge_{i=1}^m ((\bigvee_{j \in J_i^+} x_j) \vee (\bigvee_{j \in J_i^-} \neg x_j))$ be a SAT formula on n variables with $|J_i^+| + |J_i^-| \leq n$. Set $\varepsilon = 1/(n+1)$. Consider the network N_ϕ with two hidden layers that computes the function

$$f_{N_\phi}(x) = \max(0, T_n(x) + \varepsilon - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j))),$$

Note that N_ϕ can be constructed from ϕ in polynomial time, since adding T_n increases the encoding size only by an additional $\mathcal{O}(n^2)$ term. The idea is now to show that if ϕ has a satisfying assignment, then N_ϕ has at least two linear regions, and if ϕ has no satisfying assignment, then N_ϕ has only one linear region with the constant zero function, which proves the lemma.

By Lemma B.1, we have $T_n(x) \leq -\varepsilon$ and therefore $f_{N_\phi}(x) = 0$ for all $x \in \mathbb{R}^n$ with some $x_i \in (-\infty, -\varepsilon] \cup [\varepsilon, 1 - \varepsilon] \cup [1 + \varepsilon, \infty)$.

As a result, we have

$$\{x : f_{N_\phi}(x) > 0\} \subseteq ([-\varepsilon, \varepsilon] \cup [1 - \varepsilon, 1 + \varepsilon])^n = \bigcup_{x \in \{0, 1\}^n} B_\varepsilon^\infty(x),$$

where $B_\varepsilon^\infty(x) := \{x' : \|x - x'\|_\infty \leq \varepsilon\}$.

Suppose now that $x^* \in \{0, 1\}^n$ satisfies ϕ . Then,

$$\sum_{j \in J_i^+} x_j^* + \sum_{j \in J_i^-} (1 - x_j^*) \geq 1 \quad \text{for all } i \in [m],$$

which implies $f_{N_\phi}(x^*) = \max(0, T_n(x^*) + \varepsilon) = \varepsilon > 0$. Thus, N_ϕ has at least two linear regions.

Suppose now that $x^* \in \{0, 1\}^n$ does not satisfy ϕ . There is at least one clause i^* with

$$\sum_{j \in J_{i^*}^+} x_j^* + \sum_{j \in J_{i^*}^-} (1 - x_j^*) = 0.$$

In particular, for all $x \in B_\varepsilon^\infty(x^*)$, we have

$$1 - \sum_{j \in J_{i^*}^+} x_j - \sum_{j \in J_{i^*}^-} (1 - x_j) \geq 1 - |J_{i^*}^+|\varepsilon - |J_{i^*}^-|\varepsilon \geq 1 - n\varepsilon = 1 - n/(n+1) = 1/(n+1) = \varepsilon.$$

Therefore, we have $f_{N_\phi}(x) = 0$ for all $x \in B_\varepsilon^\infty(x^*)$. If ϕ has no satisfying assignment, then f_{N_ϕ} is the constant zero function. \square

Proof of Theorem 5.2. Given fixed constants $K, L \in \mathbb{N}_{\geq 1}$, $L \geq 2$ and a SAT formula ϕ , we will create a network with L hidden layers which has strictly more than K linear regions if and only if the network N_ϕ from the proof of Lemma 5.3 has strictly more than one hidden layer.

Let N_ϕ be the network as in the proof of Lemma 5.3. If $K \geq 2$, the network $N_\phi^{(K)}$ computing the function

$$f_{N_\phi}(x) - \max(0, 2(n+m)(x_1 - 2)) - \cdots - \max(0, 2(n+m)(x_1 - K))$$

has K linear regions if N_ϕ has only one linear region and strictly more than K linear regions if N_ϕ has more than one linear region. For Definitions 3 to 5, this follows from the fact that the newly introduced linear regions are outside of the hypercube $[-\varepsilon, 1 + \varepsilon]^n$ that contains all nonzero linear regions of $f_{N_\phi}(x)$. For Definition 6, we additionally have to verify that no newly introduced affine

function was already present in f_{N_ϕ} . To see that no newly affine function was already present in f_{N_ϕ} , observe that the coefficient of x_1 of every newly introduced affine function is at most $-2(n+m)$, while the coefficient of every affine function of f_{N_ϕ} cannot be smaller than $-n-m$, which can be easily seen from the proof of Lemma 5.3.

The additional maximum terms can be created using two neurons in the first hidden layer that correspond to the positive and negative part of x_1 , respectively, and adding $K-1$ neurons in the second hidden layer (using the equation $x_1 = \max(0, x_1) - \max(0, -x_1)$ to build the maximum terms in the second hidden layer).

Thus, the network $N_\phi^{(K)}$ has two hidden layers. To obtain a network $N_\phi^{(K,L)}$ with L hidden layers, we add $L-2$ new hidden layers between the output layer and the last hidden layer of $N_\phi^{(K)}$. Each new hidden layer has two neurons, the first neuron outputs $\max(0, N_\phi^{(K)})$ and the second neuron outputs $\max(0, -N_\phi^{(K)})$. We achieve this by replacing the arcs from the second hidden layer of $N_\phi^{(K)}$ to the output node by connections to the two neurons of the first newly added hidden layer. The theorem now follows by noting that the modified network $N_\phi^{(K,L)}$ has encoding size $\mathcal{O}(K \cdot \langle N_\phi \rangle + L)$ and can be constructed from N_ϕ in polynomial time. \square

Proof of Corollary 5.4. Let $L \in \mathbb{N}$ be a fixed constant. Given two ReLU networks N, N' with L hidden layers, let N^- represent the network with L hidden layers that ‘subtracts’ N' from N by computing the networks N and N' in parallel. N and N' compute the same function if and only if N^- computes the zero function.

To see that L -NETWORK-EQUIVALENCE is in NP, note that a vector in $\{0, 1\}^{s(N^-)}$ that corresponds to a proper activation pattern with a nonzero affine function can be used as a certificate, as in the proof of Lemma 5.3.

If $L = 1$, by Theorem 4.1 we can decide in polynomial time if N^- computes an affine function. If N^- computes an affine function then it computes the zero function if and only if N^- evaluates to zero on $n+1$ affinely independent points, which yields a polynomial time algorithm for 1-NETWORK-EQUIVALENCE. Suppose $L \geq 2$ and let ϕ be a SAT formula, let $N_\phi^{(1)}$ be the ReLU network with L hidden layers from the proof of Theorem 5.2, and let N_0 be a ReLU network with L hidden layers that computes the zero function. By Theorem 5.2, a SAT formula ϕ is satisfiable if and only if $N_\phi^{(1)}$ and N_0 are a no-instance of L -NETWORK-EQUIVALENCE, proving that L -NETWORK-EQUIVALENCE is coNP-hard. \square

Proof of Corollary 5.5. Let $K, L \in \mathbb{N}, L \geq 2$ be fixed constants. Given a 3-SAT formula ϕ on n variables and m clauses, let $N_\phi^{(K,L)}$ be the ReLU network with L hidden layers and input dimension n from the proof of Theorem 5.2. Recall that $N_\phi^{(K,L)}$ has strictly more than $K \in \mathbb{N}$ hidden layers if and only if ϕ is satisfiable.

We now show that $N_\phi^{(K,L)}$ has encoding size $\mathcal{O}(m^2)$. Recall that $N_\phi^{(K,L)}$ has an encoding size of $\mathcal{O}(K \cdot \langle N_\phi \rangle + L)$, and N_ϕ has an encoding size of $\mathcal{O}(n^2 + nm)$. Since $n \leq 3m$ holds for every 3-SAT formula and K and L are constants, the encoding size of $N_\phi^{(K,L)}$ is $\mathcal{O}(m^2)$.

It is well known that, assuming the Exponential Time Hypothesis is true, this implies that there is no $2^{o(n)}$ or $2^{o(\sqrt{\langle N \rangle})}$ time algorithm for K -REGION-DECISION, see [Cygan et al., 2015]. A $2^{o(n)}$ or $2^{o(\sqrt{\langle N \rangle})}$ time algorithm for LINEAR REGION COUNTING problem would directly give a $2^{o(n)}$ or $2^{o(\sqrt{\langle N \rangle})}$ time algorithm for K -REGION-DECISION. \square

Intuition for the proof of Lemma 5.6

Given a SAT formula ϕ , the network N_ϕ from the proof of Lemma 5.3 has some nonzero linear regions near every satisfying assignment of ϕ . Unfortunately, the number of linear regions created per satisfying point depends on the formula ϕ and is not easily computable. Therefore, we modify the network N_ϕ such that the same number of nonzero linear regions is created by every satisfying

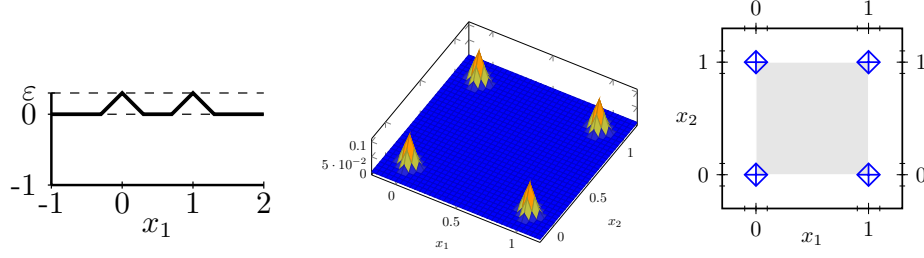


Figure 4: The functions $T_{1,\epsilon}$ (left), $T_{2,\epsilon}$ (center) and its linear regions (right).

assignment of ϕ . For this, we take the minimum of f_{N_ϕ} with the function $T_{n,\epsilon} : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$T_{n,\epsilon}(x) = \max(0, \epsilon + T_n(x)),$$

shown in Figure 4. We further show that $T_{n,\epsilon}$ is strictly smaller than f_{N_ϕ} , but greater than zero near a satisfying point. In this way, the minimum of f_{N_ϕ} and $T_{n,\epsilon}$ is attained by f_{N_ϕ} near each non-satisfying point (where f_{N_ϕ} equals the zero function) and by $T_{n,\epsilon}$ near each satisfying point. We proceed by showing that exactly 2^n nonzero regions are created for every satisfying assignment of ϕ , and if ϕ has exactly k satisfying assignments, the modified network has exactly $1 + k \cdot 2^n$ linear regions according to Definitions 4 and 5. The modified network has three hidden layers. The reduction can then be extended to ReLU networks with $L \geq 3$ hidden layers as before.

The following lemma is required for the proof of Lemma 5.6.

Lemma B.2. *Let $n \geq 2$ and $B_\epsilon^\infty(x) := \{x' \in \mathbb{R}^n : \|x - x'\|_\infty \leq \epsilon\}$. The function $T_{n,\epsilon}$ with $0 < \epsilon < 1/2$ has exactly $1 + 2^{2^n}$ linear regions according to Definitions 4 and 5 and we have*

$$\begin{aligned} T_{n,\epsilon}(x) > 0 &\implies x \in \bigcup_{x^* \in \{0,1\}^n} B_\epsilon^\infty(x^*), \\ T_{n,\epsilon}(x) = \epsilon &\iff x \in \{0,1\}^n, \end{aligned}$$

and for every $x^* \in \{0,1\}^n$, the set $B_\epsilon^\infty(x^*)$ contains exactly 2^n nonzero regions according to Definitions 3 to 6.

Proof of Lemma B.2. By Lemma B.1, if $T_n(x) \geq -\epsilon$, then $x \in B_\epsilon^\infty(x^*)$ for some $x^* \in \{0,1\}^n$. Therefore, we have $T_{n,\epsilon}(x) = 0$ for all $x \in \mathbb{R}^n \setminus (\bigcup_{x^* \in \{0,1\}^n} B_\epsilon^\infty(x^*))$. What is left is to analyze the linear regions of $T_{n,\epsilon}$ in the set $B_\epsilon^\infty(x^*)$ for every $x^* \in \{0,1\}^n$.

Due to the symmetry of $T_{n,\epsilon}$, we only consider the set $B_\epsilon^\infty((1, \dots, 1)^\top) = [1 - \epsilon, 1 + \epsilon]^n$. We will show that $[1 - \epsilon, 1 + \epsilon]^n$ has exactly 2^n nonzero linear regions.

First, observe that for a point $x \in [1 - \epsilon, 1 + \epsilon]^n$, we have

$$T_n(x) = \sum_{i: x_i < 1} (x_i - 1) + \sum_{i: x_i > 1} (1 - x_i).$$

Given a subset I of $[n]$, we define the set

$$C_I := \{x : 1 - \epsilon \leq x_i < 1 \ \forall i \in I, \ 1 \leq x_i \leq 1 + \epsilon \ \forall i \notin I\}.$$

It is easy to see that the disjoint union $\bigcup_{I \subseteq [n]} C_I$ gives exactly the set $[1 - \epsilon, 1 + \epsilon]^n$.

Each set C_I divides into two sets:

$$\begin{aligned} C_I^1 &:= \{x \in C_I : \sum_{i \in I} (x_i - 1) + \sum_{i \notin I} (1 - x_i) \geq -\epsilon\} \\ C_I^0 &:= \{x \in C_I : \sum_{i \in I} (x_i - 1) + \sum_{i \notin I} (1 - x_i) \leq -\epsilon\} \end{aligned}$$

We have $T_{n,\epsilon}(x) = \epsilon + \sum_{i \in I} (x_i - 1) + \sum_{i \notin I} (1 - x_i)$ for all $x \in C_I^1$ and $T_{n,\epsilon}(x) = 0$ for all $x \in C_I^0$.

The set C_I^1 is full dimensional, as $x^* \in \mathbb{R}^n$ with $x_i^* = \begin{cases} 1 - \frac{\varepsilon}{2n}, & i \in I \\ 1 + \frac{\varepsilon}{2n}, & i \notin I \end{cases}$ is an interior point of C_I^1 .

This proves that every C_I contains exactly one nonzero region. Since the function for every C_I is unique, $[1 - \varepsilon, 1 + \varepsilon]^n$ contains exactly 2^n nonzero regions according to Definitions 3 and 6. Since $T_{n,\varepsilon}$ has only a single zero region, it follows that $T_{n,\varepsilon}$ has exactly $1 + 2^{2n}$ linear regions. \square

Proof of Lemma 5.6. Let $\phi(x) = \bigwedge_{i=1}^m ((\bigvee_{j \in J_i^+} x_j) \vee (\bigvee_{j \in J_i^-} \neg x_j))$ be a SAT formula on n variables, where $|J_i^+| + |J_i^-| \leq n$. Set $\varepsilon = 1/(2 + n + nm)$. Consider the network N_ϕ^* that computes the function

$$f_{N_\phi^*}(x) = \min(T_{n,\varepsilon}(x), \max(0, 1 - (n+1)\varepsilon - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j)))),$$

which can be computed with 3 hidden layers. This is due to the fact that the minimum of two terms can be expressed using three neurons $\min(a, b) = -\max(0, b - a) + \max(0, b) - \max(0, -b)$. The reduction is polynomial since the addition of $T_{n,\varepsilon}$ increases the encoding size only by an additional $\mathcal{O}(n^2)$ term. Now, our goal is to show that if ϕ has exactly k satisfying assignments, then N_ϕ^* has exactly $1 + 2^n \cdot k$ linear regions.

By Lemma B.2, if $f_{N_\phi^*}(x) > 0$ then $x \in \bigcup_{x^* \in \{0,1\}^n} B_\varepsilon^\infty(x^*)$, where $B_\varepsilon^\infty(x^*) = \{x' \in \mathbb{R}^n : \|x^* - x'\|_\infty \leq \varepsilon\}$.

We will prove our theorem by showing that the following holds for all $x^* \in \{0,1\}^n$.

1. If $\phi(x^*) = 0$, then $f_{N_\phi^*}$ has no nonzero linear region in $B_\varepsilon^\infty(x^*)$.
2. If $\phi(x^*) = 1$, then $f_{N_\phi^*}$ has exactly 2^n nonzero linear regions in $B_\varepsilon^\infty(x^*)$.

We start with the first implication. Suppose $\phi(x^*) = 0$ holds. Then, there is a clause i^* such that

$$x_j^* = \begin{cases} 0, & j \in J_{i^*}^+ \\ 1, & j \in J_{i^*}^- \end{cases} \quad \text{holds. Thus, we have for all } x \in B_\varepsilon^\infty(x^*):$$

$$\begin{aligned} & - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j)) \\ & \leq - \max(0, 1 - \sum_{j \in J_{i^*}^+} x_j - \sum_{j \in J_{i^*}^-} (1 - x_j)) \\ & \leq - \max(0, 1 - \sum_{j \in J_{i^*}^+} \varepsilon - \sum_{j \in J_{i^*}^-} (1 - (1 - \varepsilon))) \\ & = -1 + (|J_{i^*}^+| + |J_{i^*}^-|)\varepsilon, \end{aligned}$$

implying

$$1 - (n+1)\varepsilon - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j)) \leq (|J_{i^*}^+| + |J_{i^*}^-| - n - 1)\varepsilon \leq 0$$

and thus $f_{N_\phi^*}(x) = 0$ for all $x \in B_\varepsilon^\infty(x^*)$.

To prove the second implication, suppose that $\phi(x^*) = 1$ holds. We will show that the second component f_{N_ϕ} in the minimum of $f_{N_\phi^*}$,

$$f_{N_\phi}(x) = \max(0, 1 - (n+1)\varepsilon - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j)))$$

is greater or equal to $T_{n,\varepsilon}$ for all $x \in B_\varepsilon^\infty(x^*)$. Then $f_{N_\phi^*}(x) = T_{n,\varepsilon}(x)$ holds for all $x \in B_\varepsilon^\infty(x^*)$. By Lemma B.2, this implies that $f_{N_\phi^*}$ has exactly 2^n nonzero linear regions in $B_\varepsilon^\infty(x^*)$, which will prove the second implication.

We now show that $f_{N_\phi^*}(x) = T_{n,\varepsilon}(x)$ holds for all $x \in B_\varepsilon^\infty(x^*)$. W.l.o.g. let $x^* = (1, \dots, 1)^\top$. By assumption, $|J_i^+| \geq 1$ and $|J_i^-| \leq n-1$ holds for all clauses $i \in [m]$. Thus, we have for all $x \in B_\varepsilon^\infty(x^*) = [1-\varepsilon, 1+\varepsilon]^n$ and all $i \in [m]$

$$1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j) \leq 1 - |J_i^+|(1-\varepsilon) + |J_i^-|\varepsilon \leq \varepsilon + (n-1)\varepsilon = n \cdot \varepsilon.$$

As a consequence, for all $x \in [1-\varepsilon, 1+\varepsilon]^n$, we have

$$\begin{aligned} f_{N_\phi}(x) &\geq \max(0, 1 - (n+1)\varepsilon - \sum_{i=1}^m \max(0, 1 - \sum_{j \in J_i^+} x_j - \sum_{j \in J_i^-} (1 - x_j))) \\ &\geq \max(0, 1 - (n+1)\varepsilon - \sum_{i=1}^m n \cdot \varepsilon) \\ &= 1 - (n+1)\varepsilon - m \cdot n \cdot \varepsilon \\ &= 1 - (1 + n + nm)\varepsilon \\ &= \varepsilon \\ &\geq T_{n,\varepsilon}(x), \end{aligned}$$

and thus, $f_{N_\phi^*}(x) = T_{n,\varepsilon}(x)$ for all $x \in [1-\varepsilon, 1+\varepsilon]^n$. By Lemma B.2, $T_{n,\varepsilon}$ has 2^n nonzero linear regions in $[1-\varepsilon, 1+\varepsilon]^n$.

We extend the hardness result to networks with $L \geq 3$ hidden layers as in the proof of Theorem 5.2. \square

The following lemma will be used in the proof of Theorem 5.8.

Lemma B.3. *Let $g : \mathbb{R}^n \rightarrow \mathbb{R}$ be a CWPL function with exactly m affine regions. Then, for every $k \in \mathbb{N}$, the function $g^{(k)} : \mathbb{R}^{nk} \rightarrow \mathbb{R}$,*

$$g^{(k)}(x_{1,1}, \dots, x_{1,n}, \dots, x_{k,1}, \dots, x_{k,n}) = \sum_{i=1}^k g(x_{i,1}, \dots, x_{i,n})$$

has exactly m^k affine regions.

Proof. Let U_1, \dots, U_m be the affine regions of g , let R_1, \dots, R_p be the affine regions of $g^{(k)}$ and let $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be the affine function of the affine region U_i of g for every $i \in [m]$.

For every $i \in [m]^k$ and for all $x \in U_{i_1} \times \dots \times U_{i_k}$, the function $g^{(k)}$ computes the affine function

$$g^{(k)}(x_{1,1}, \dots, x_{1,n}, \dots, x_{k,1}, \dots, x_{k,n}) = \sum_{j=1}^k h_{i_j}(x_{j,1}, \dots, x_{j,n}).$$

Since all affine functions h_1, \dots, h_m are distinct, $U_{i_1} \times \dots \times U_{i_k}$ is contained in a different affine region of $g^{(k)}$ for every $i \in [m]^k$. As $U_{i_1} \times \dots \times U_{i_k}$ is inclusion-maximal with respect to affinity of $g^{(k)}$, it follows that $\{R_1, \dots, R_p\} = \{U_{i_1} \times \dots \times U_{i_k} : i \in [m]^k\}$, which concludes the proof. \square

Proof of Theorem 5.8. Let ϕ be a SAT formula on l variables and let N_ϕ be the ReLU network with two hidden layers constructed in the proof of Lemma 5.3. Recall that for Definitions 3 to 6, the network N_ϕ has at least two linear regions if ϕ is satisfiable and exactly one linear region if ϕ is unsatisfiable.

Let $N_\phi^{(k)}$ be the ReLU network composed of taking k copies of N each with a disjoint set of l variables. The function computed by the ReLU network $N_\phi^{(k)}$ is $f_{N_\phi^{(k)}} : \mathbb{R}^{lk} \rightarrow \mathbb{R}$ with

$$f_{N_\phi^{(k)}}(x_{11}, \dots, x_{1l}, \dots, x_{k1}, \dots, x_{kl}) = \sum_{i=1}^k f_{N_\phi}(x_{i1}, \dots, x_{il}).$$

If ϕ is unsatisfiable then N_ϕ has exactly one linear region which implies that $N_\phi^{(k)}$ also has exactly one linear region (according to Definitions 3 to 6). If ϕ is satisfiable then N_ϕ has at least two affine regions and by Lemma B.3, $N_\phi^{(k)}$ has at least 2^k affine regions. By Theorem 3.1, $N_\phi^{(k)}$ then also has at least 2^k linear regions according to Definitions 3 to 5. It follows that approximating the number of regions within a factor larger than 2^{-k} is NP-hard (according to Definitions 3 to 6). Setting $n = lk$, the theorem now follows by picking $k = l^C$ for a sufficiently large constant C (e.g., C such that $\frac{C}{C+1} > 1 - \varepsilon$) and noting the construction of $N_\phi^{(k)}$ from N_ϕ can be done in polynomial time. \square

B.3 Omitted proofs for polynomial space algorithms

Lemma B.4. *Given a ReLU network N and an affine map $\varphi(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i + b$, we can check in space polynomial in $\langle N \rangle$ and in the encoding size of the coefficients of φ whether φ is the function of an affine region of N .*

Proof. First, note that φ is the function of an affine region of N if and only if there is a proper activation region on which φ is realized.

Now, go over all $2^{s(N)}$ possible activation patterns for neurons of N using space polynomial in $s(N)$. For each vector $a \in \{0, 1\}^{s(N)}$, it is possible to verify in time polynomial in the encoding size $\langle N \rangle$ of the ReLU network N whether S_a is a proper activation region, see Lemma A.5. Further, we can check in polynomial time whether φ is equal to the function f_N^a computed on the proper activation region S_a , see Lemma A.4. \square

Proof of Theorem 6.1. Let N be a ReLU network. As discussed in Section 6, the number of activation regions and proper activation regions can be counted in space which is polynomial in the encoding size $\langle N \rangle$ of the ReLU network N . Now, we describe a polynomial space algorithm for counting the number of affine regions.

By Lemma A.3, the encoding size of any coefficient of an affine function that occurs in one of the affine regions of N is bounded by $M := 36d^2n_{\max}^2\langle A_{\max} \rangle$, which is polynomial in $\langle N \rangle$.

As each coefficient of an affine function of N is a fraction, M is also an upper bound on the encoding size of a numerator and on the encoding size of a denominator. Since each affine function of N is defined by $n + 1$ fractions, we can exhaustively search through all sequences of $n + 1$ fractions, where the numerator and denominator of each fraction can have encoding size of at most M . For each sequence of $n + 1$ fractions, by Lemma B.4 we can compute in space that is polynomial in $\langle N \rangle$ if the corresponding affine function is the function of an affine region of N . If an affine function of an affine region is found, we increase a counter by 1. To avoid counting the same affine function more than once, we only check fraction sequences in which the numerator and denominator of every fraction are relatively prime. \square

C Examples

C.1 A closed connected region which is not a closure of a union of a set of activation regions

Zanotti [2025a, Figure 1] uses the following function as an example:

$$\min(y, \max(-1, -x), \max(3 - 2x, -x)).$$

We turn this function into a ReLU network N with three hidden layers, as illustrated in Figure 5. For the orange closed connected region P in Figure 5, there is no set of activation regions such that P is the closure of a union of activation regions of the ReLU network N .

C.2 Further examples

Example C.1. *Consider the SAT formula $\varphi = (\neg x_1) \wedge (x_1 \vee x_2)$ with the satisfying assignment $(0, 1)$ and the function $g_\varphi(x) = 1 - \max(0, 1 - (1 - x_1)) - \max(0, 1 - x_1 - x_2)$ displayed in Figure 6.*

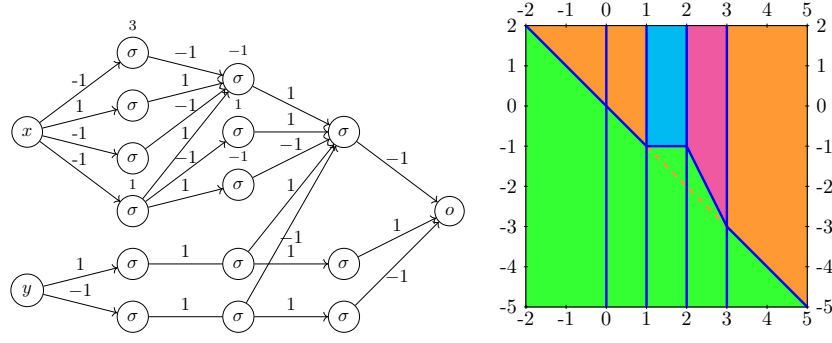


Figure 5: A ReLU network N computing $\min(y, \max(-1, -x), \max(3 - 2x, -x))$. An activation region of N is either a blue line, blue point, or a full dimensional cell as defined by the blue lines. There are four closed connected region as indicated by the colors. The line between the points $(1, -1)$ and $(3, -3)$ belongs to the green as well as the orange region.

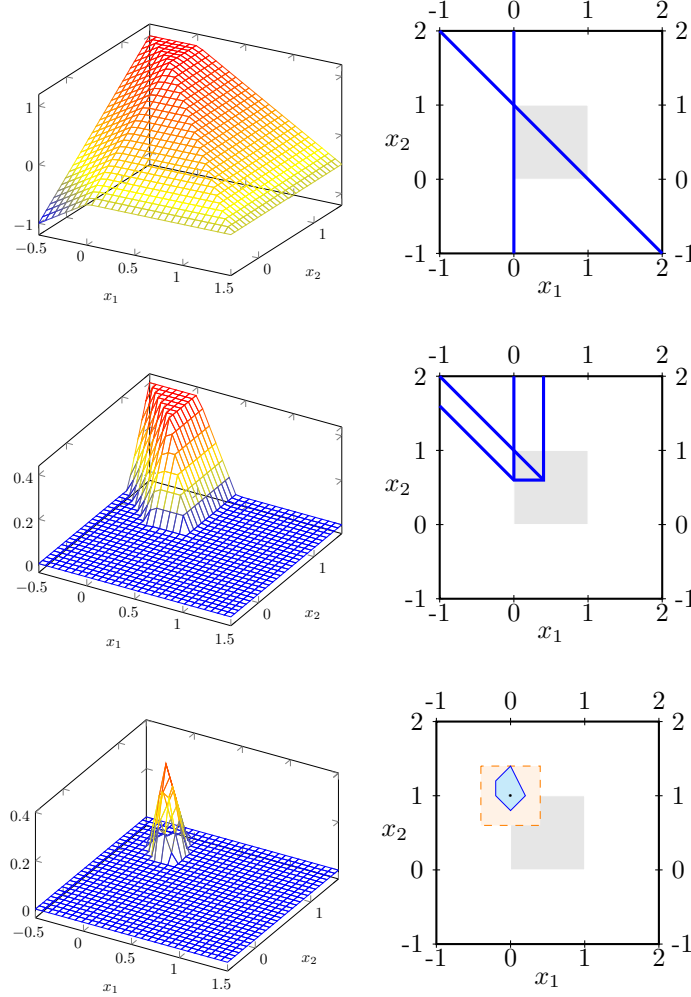


Figure 6: The functions $g_\varphi(x) = 1 - \max(0, x_1) - \max(0, 1 - x_1 - x_2)$ (top), $h_{\varphi, \varepsilon}(x) = \max(0, \varepsilon - 1 + g_\varepsilon(x))$ (center), and $f_{N_\varphi}(x) = \max(0, T_2(x) + \varepsilon - 1 + g_\varphi(x))$ (bottom) for $\varepsilon = 0.4$. The function f_{N_φ} is only nonzero in the blue region, which is contained in the ε -square (orange) around the only satisfying point of φ (black).

As mentioned above, for all $x \in \{0, 1\}^2$, $g_\varphi(x) = 1$ holds if x is a satisfying assignment of φ and $g_\varphi(x) \leq 0$ otherwise. Since φ has an satisfying assignment, the function $h_{\varphi, \varepsilon}$ with $h_{\varphi, \varepsilon}(x) = \max(0, \varepsilon - 1 + g_\varphi(x))$ has strictly more than one linear region, see Figure 6. The final function in the reduction of is $f_{N_\varphi}(x) = \max(0, T_2(x) + \varepsilon - 1 + g_\varphi(x))$, see Figure 6.

Example C.2. Consider the SAT formula and function

$$\begin{aligned} \psi &= (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2), \\ h_{\psi, \varepsilon}(x_1, x_2) &= \max(0, \varepsilon - \max(0, 1 - x_1 - x_2) - \max(0, 1 - (1 - x_1) - x_2) \\ &\quad - \max(0, 1 - x_1 - (1 - x_2)) - \max(0, 1 - (1 - x_1) - (1 - x_2))). \end{aligned}$$

It is clear that ψ is unsatisfiable. However, for every $\varepsilon > 0$ we have $h_{\psi, \varepsilon}(1/2, 1/2) = \varepsilon > 0$.