International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 10, Number 2, pages 293–307, July 2020

Expressive Numbers of Two or More Hidden Layer ReLU Neural Networks

Kenta Inoue
Chiba University, nekonistyle@gmail.com
1-33 Yayoi-cho, Inage-ku, Chiba-shi, Chiba 263–8522, Japan

Received: February 15, 2020
Revised: May 5, 2020
Accepted: June 1, 2020
Communicated by Kouzou Ohara

Abstract

One of the reasons why neural networks are used in machine learning is their high expressive power, that is, the ability to express functions. Expressive power of neural networks depends on its structures and is measured by some indices. In this paper, we focus on one of these measures named "expressive number", which is based on the number of data that can be expressed. Expressive numbers enable us to see whether the size of a neural network is suitable for the given training data before we conduct machine learning. However, existing works on expressive numbers mainly target single hidden layer neural networks, and little is known about those with two or more hidden layers. In this paper, we give a lower bound of the maximum expressive number of two hidden layer neural networks and an upper bound of that of multilayer neural networks with ReLU activation function. This result shows the expressive number of two hidden layer neural networks is in $O(a_1 a_2)$ where a_1 and a_2 are the numbers of each hidden layer's neurons.

Keywords: Neural Network, Expressive Power, Expressive Number, ReLU Activation Function

1 Introduction

Neural networks are widely used in the field of machine learning, such as image or speech recognition. One of the reasons why neural networks are used is they have high expressive power: how complex functions they can express. Expressive power of neural networks depends on their structure named "hyperparameter", which does not change during learning: the numbers of their hidden layers and each layer's neurons, and activation functions. In machine learning, we usually need to determine the hyperparameter of the neural network before learning. In other words, we should know the expressive power of the neural network. However, we know a little about expressive power, so, we study it theoretically.

To develop a rigorous theory of expressive power, i.e., how complex functions a particular neural network can express, we need to define a measure of the complexity. In this paper, we measure it by "expressive number" [6]. Expressive number of neural networks is defined as the number N such that arbitrary data with cardinality N can be expressed in the neural networks. The more complex a target function in machine learning is, the more training data we need. So, a neural network that has larger expressive number can express more complex functions. In other words, expressive numbers can measure the expressive power of neural networks.

Expressive numbers have two advantages as a measure of the expressive power of neural networks. One advantage is that we can reduce the potential for overfitting before learning. Overfitting is often

caused by too high expressive power to express training data [15]. By expressive number, we can compare the expressive power and the complexity of training data easily. So, we can choose a better neural network by expressive numbers. Another advantage is we can see the differences of expressive powers between activation functions. Some measures of expressive power, such as the number of linear regions [10], are defined on neural networks with only ReLU or piecewise linear activation functions. On the other hand, expressive number is also defined with any activation functions.

Expressive number is a meaningful measure of the expressive power of neural networks. However, we only know a few properties of expressive numbers: the independence of expressive number from the input dimension, and the upper and lower bounds of the maximum expressive numbers of single hidden layer ReLU neural networks [6]. So, the properties of those with two or more hidden layers are not shown. In this paper, we show an upper bound of expressive numbers of multilayer ReLU neural networks and a lower bound of these of two hidden layer ReLU neural networks.

The outline of this paper is as follows: In Section 3, we define expressive number and some notations. Section 4 and Section 5 show the upper and lower bounds of the maximum expressive numbers, respectively. Besides, we show the maximum expressive number of two hidden layer ReLU neural networks is proportional to the product of the numbers of each hidden layer's neurons.

2 Related Work

In the literature, expressive power of neural networks is sometimes described by other measures besides expressive number. One measure is based on the number of linear regions ([10, 9, 14, 4]). A linear region on neural networks with ReLU or piecewise linear activation functions is defined as a maximal connected subset on the input set such that linearity holds. Compared with expressive number, expressive number of a neural network gives us some information about machine learning, that is, whether training data can be expressed in the neural network. However, only a little information can be obtained from the number of linear regions. Other measures based on trajectory length [12] or knots [3] give us little information too. Furthermore, because linear regions are defined only on ReLU or piecewise linear activation neural networks, we cannot compare the expressive powers between those on piecewise linear and those on non piecewise linear activations.

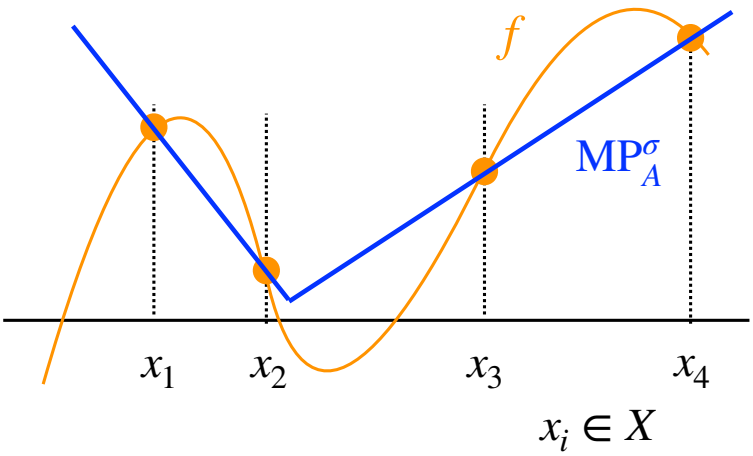
VC-dimension ([8, 7]) is also used as a measure of expressive power, and it enables us to get an upper bound of the generalization error. In contrast, expressive number tells us the greatest lower bound of the empirical error is zero when the number of training data is smaller than the expressive number. Both generalization error and empirical error are usually used as measures of the whole of learning activity, including the learning algorithm and training data as well as the structure of neural networks. However, each error has a different role; generalization error evaluates the difference between the output a neural network returns from unknown data and the true output of the data, and empirical error is used to judge whether the learning has finished.

As these measures, expressive power is often measured for each fixed structure of a neural network. However, sometimes, expressive power is described by the smallest number of neurons, which can express some fixed functions. Rolnick and Tegmark investigated such numbers for several cases of multivariate polynomials [13].

3 Preliminaries

Fix an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. For any $k \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{R}^k$, we simply write $\sigma(\boldsymbol{x}) := \begin{pmatrix} \sigma(x_0) \\ \vdots \\ \sigma(x_{k-1}) \end{pmatrix}$ where $\boldsymbol{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_{k-1} \end{pmatrix}$.

Given $l \in \mathbb{N}$ and $(a_0, \ldots, a_l) \in \mathbb{N}^{l+1}$, "$(a_0, \ldots, a_l)$ neural networks" denote neural networks that have a_0 input neurons, a_l output neurons and $a_1, \ldots, a_{l-1}$ hidden neurons ordered from the input side to the output side. For an $(a_0, \ldots, a_l)$ neural network $A \in \prod_{i=1}^{l} (\mathbb{R}^{a_i \times a_{i-1}} \times \mathbb{R}^{a_i})$, MP_A^σ :



The finite set X is input data and f is an input-output function. We say "the data (X, f) is solvable" on neural networks when there exists some parameter A of the neural network, i.e. weights and biases, such that the function MP^{σ}_{A} the neural network calculates coincides with f on the input data X.

Figure 1: Solvability of data (X, f)

$\mathbb{R}^{a_0} \to \mathbb{R}^{a_l}$ denotes the function the neural network A calculates: $\mathrm{MP}^{\sigma}_{A} = F_l \circ \sigma \circ \cdots \circ \sigma \circ F_2 \circ \sigma \circ F_1$ where $A = ((W_1, \boldsymbol{b}_1), \ldots, (W_l, \boldsymbol{b}_l))$ and $F_i : \mathbb{R}^{a_{i-1}} \to \mathbb{R}^{a_i}$ defined by $F_i(\boldsymbol{x}) := W_i \boldsymbol{x} + \boldsymbol{b}_i$ for any i. We simply write MP_A for MP^{σ}_{A} when σ is clear from the context.

Before introducing expressive number, we define "solvability" as an ability to express the given data.

Definition 1 (Solvability)
*Given $X \subset \mathbb{R}^{a_0}$ and $f : \mathbb{R}^{a_0} \to \mathbb{R}^{a_l}$. We say "***the data (X, f) is solvable on $(a_0, \ldots, a_l)$ neural networks***" if there exists an $(a_0, \ldots, a_l)$ neural network A such that $\forall \boldsymbol{x} \in X, \mathrm{MP}^{\sigma}_{A}(\boldsymbol{x}) = f(\boldsymbol{x})$.*

The above definition says that the data (X, f) is solvable if there exists a solution to express the data with no empirical error on $(a_0, \ldots, a_l)$ neural networks as Figure 1. In other words, the size $(a_0, \ldots, a_l)$ of neural networks is large enough to express the data (X, f).

Next, we define "expressive number" via a relation between the size of neural networks and the number of data.

Definition 2 (Expressive number)
We say "$(a_0, \ldots, a_l)$ neural networks have **expressive number** *N" if the following condition holds:*

> *For any $X \subset \mathbb{R}^{a_0}$ such that $|X| = N$ and any $f : \mathbb{R}^{a_0} \to \mathbb{R}^{a_l}$, the data (X, f) is solvable on $(a_0, \ldots, a_l)$ neural networks.*

Then the **maximum expressive number** *of $(a_0, \ldots, a_l)$ neural networks is defined as the maximum number of expressive numbers the neural networks have.*

The definition says that $(a_0, \ldots, a_l)$ neural networks have expressive number N if any N data is solvable on the neural networks. In other words, for any data, if the number of the data is smaller than the maximum expressive number of $(a_0, \ldots, a_l)$ neural networks, the size of the neural networks is large enough to express the data.

Whereas some papers describe data as an input-output set $\{(x_i, y_i)\}_{i \in I} \subset \mathbb{R}^{a_0} \times \mathbb{R}^{a_l}$, we describe data as a pair (X, f) of an input set X and an input-output function f to define expressive number in a simpler way. If we use the former, the maximum expressive number of any neural network is always 1 unless we explicitly exclude the case of having different outputs for the same input.

The following properties of expressive numbers have already been shown in [6].

I-1 $(a_0, a_1, \ldots, a_l)$ neural networks have expressive number N if and only if $(1, a_1, \ldots, a_l)$ neural networks have expressive number N.

I-2 (n, k, m) ReLU neural networks have expressive number $k+1$.

I-3 The maximum expressive number of (n, k, m) ReLU neural networks is less than or equal to $k+2$.

Property I-1 indicates that expressive numbers are independent of input dimensions. Properties I-2 and I-3 mean that the maximum expressive number of (n, k, m) ReLU neural networks is $k+1$ or $k+2$. (Although Property I-2 is shown only on neural networks with ReLU function in [6], we show it with more general activations in Appendix A.1.) However, the maximum expressive numbers of two or more hidden layer neural networks were not known.

4 Upper Bound

In the remainder of this paper, the activation function σ is fixed as $\mathrm{ReLU}(x) := \max\{0, x\}$.

We show an upper bound of the maximum expressive number of multilayer ReLU neural networks. To show the upper bound, we partially use the proof of Property I-3 giving: the upper bound of the maximum expressive number of single hidden layer neural networks. Then, we use "zigzag function" to show the upper bound.

Definition 3 (Zigzag function)
Let $N \in \mathbb{N}$. $x_1, \ldots, x_N \in \mathbb{R}$ be a finite sequence and $f : \mathbb{R} \to \mathbb{R}^n$. We say "$f$ **is zigzag on** *$x_1, \ldots, x_N$" if $f(x_{i-1}) \prec f(x_i) \succ f(x_{i+1})$ or $f(x_{i-1}) \succ f(x_i) \prec f(x_{i+1})$ holds for any $1 < i < N$ where the binary relation $\prec$ is defined as the lexicographical order on $\mathbb{R}^n$. Then, let X be a finite subset of $\mathbb{R}$. We say "f is zigzag function on X" if f is zigzag on $x_1, \ldots, x_{|X|}$ where $x_1, \ldots, x_{|X|} \in X$ is an increasing sequence of all elements of X.*

The following lemma appears in the proof of Property I-3.

Lemma 1 (Solvability on Zigzag functions)
Let $N \in \mathbb{N}$ and $k, m \in \mathbb{N}$. For any X such that $|X| = N$ and any zigzag function $f : \mathbb{R} \to \mathbb{R}^m$ on X, if the data (X, f) is solvable on $(1, k, m)$ ReLU neural networks then $N \leq k+2$ holds.

Proof.
This lemma's proof appears in the proof of Theorem 6 (2) (Property I-3) in [6] as an example of the upper bound of the maximum expressive number of single hidden layer neural networks. The proof was shown by the following property:

> If there exists a neural network A such that $\mathrm{MP}_A(x) = f(x)$ for any $x \in X$, then the neural network A has at least $N-1$ linear regions [10] and it has at least $N-2$ hidden neurons.

Then, we show the upper bound using this lemma.

Theorem 2 (Upper bound)
The maximum expressive number of $(n, a_1, \ldots, a_l, m)$ ReLU neural networks is less than or equal to $\left(\prod_{i=1}^{l-1}(a_i+1)\right)(a_l+2)$.

Proof.
By Property I-1, it is sufficient to prove the theorem in the case $n = 1$. Let $X \subset \mathbb{R}$ be a finite set and $f : \mathbb{R} \to \mathbb{R}$ such that f is zigzag on X. We assume the data (X, f) is solvable on $(1, a_1, \ldots, a_l, m)$ neural networks. Then, we show $|X| \leq \left(\prod_{i=1}^{l-1}(a_i+1)\right)(a_l+2)$.

To prove this property, we show the following property by induction on l.

If the data (Y, f) is solvable on $(1, a_1, \ldots, a_l, m)$ neural network, then $|Y| \leq \left(\prod_{i=1}^{l-1}(a_i+1)\right)(a_l+2)$ for any $a_1, \ldots, a_l \in \mathbb{N}$ and any $Y \subset \mathbb{R}$ such that f is zigzag on Y.

When $l = 1$, it holds by Lemma 1.

When $l > 1$, by assumption, there exists a $(1, a_1, \ldots, a_l, m)$ neural network A such that $\forall x \in X, \mathrm{MP}_A(x) = f(x)$. we can write $\mathrm{MP}_A(x) = \mathrm{MP}_B(\sigma(W_1 x + \boldsymbol{b}_1))$ where $A = ((W_1, \boldsymbol{b}_1), \ldots, (W_l, \boldsymbol{b}_l))$ and $B := ((W_2, \boldsymbol{b}_2), \ldots, (W_l, \boldsymbol{b}_l))$. Let $K := \{-\frac{b_i}{w_i} \mid w_i \neq 0,\ 0 \leq i < a_1\}$ where $W_1 = \begin{pmatrix} w_0 \\ \vdots \\ w_{a_1-1} \end{pmatrix}$ and $\boldsymbol{b}_1 = \begin{pmatrix} b_0 \\ \vdots \\ b_{a_1-1} \end{pmatrix}$, and $k_0, \ldots, k_{|K|-1}$ be all elements of K such that $k_0 < \cdots < k_{|K|-1}$. Then, $k_0, \ldots, k_{|K|-1}$ are all boundaries of linear regions given by the first hidden neurons. We define a partition $\{X_0, \ldots, X_{|K|}\}$ of X as

$$X_j := \begin{cases} \{x \in X \mid x \leq k_0\} & (j = 0) \\ \{x \in X \mid k_{j-1} < x \leq k_j\} & (0 < j < |K|) \\ \{x \in X \mid k_{|K|-1} < x\} & (j = |K|) \end{cases}$$

(If $|K| = 0$, we define $X_0 := X$.) Let $j_0 \in \underset{0 \leq j \leq |K|}{\arg\max} |X_j|$ and $Y := X_{j_0}$. We have $|Y| \geq \frac{|X|}{|K|+1} \geq \frac{|X|}{a_1+1}$ by the pigeonhole principle. Let $I_{j_0} := \{i \mid w_i k_{j_0-1} + b_i \geq 0 \wedge w_i k_{j_0} + b_i \geq 0\}$ where $k_{-1} := k_0 - 1$ and $k_{|K|} := k_{|K|-1} + 1$. Then, for any $x \in Y$, we have $i \in I_{j_0}$ if and only if $w_i x + b_i \geq 0$. So, we define $w'_i := \begin{cases} w_i & (i \in I_{j_0}) \\ 0 & (o.w.) \end{cases}$ and $b'_i := \begin{cases} b_i & (i \in I_{j_0}) \\ 0 & (o.w.) \end{cases}$ for any $0 \leq i < a_1$, then, we can write $\sigma(W_1 x + \boldsymbol{b}_1) = W' x + \boldsymbol{b}'$ for any $x \in Y$ where $W' := \begin{pmatrix} w'_0 \\ \vdots \\ w'_{a_1-1} \end{pmatrix}$, $\boldsymbol{b}' := \begin{pmatrix} b'_0 \\ \vdots \\ b'_{a_1-1} \end{pmatrix}$. Let $(1, a_2, \ldots, a_l, m)$ neural network $C := ((W_2 W', W_2 \boldsymbol{b}' + \boldsymbol{b}_2), (W_3, \boldsymbol{b}_3), \ldots, (W_l, \boldsymbol{b}_l))$. We can write $f(x) = \mathrm{MP}_A(x) = \mathrm{MP}_B(\sigma(W_1 x + \boldsymbol{b}_1)) = \mathrm{MP}_B(W' x + \boldsymbol{b}') = \mathrm{MP}_C(x)$. Thus, the data (Y, f) is solvable on $(1, a_2, \ldots, a_l, m)$ neural networks. Now, we see f is zigzag on Y. Thus, $|Y| \leq \left(\prod_{i=2}^{l-1}(a_i+1)\right)(a_l+2)$ holds by the induction hypothesis. Therefore, we have $|X| \leq (a_1+1)|Y| \leq \left(\prod_{i=1}^{l-1}(a_i+1)\right)(a_l+2)$.

This theorem shows the maximum expressive number of $(n, a_1, \ldots, a_l, m)$ ReLU neural networks is in $o(\prod_{i=1}^{l} a_i)$.

5 Lower Bound

We show a lower bound of the maximum expressive number of two hidden layer ReLU neural networks.

First, we prove the maximum expressive number is greater than or equal to the product of the numbers of its each hidden neurons when the neural networks have a single output.

Theorem 3 (Lower bound on single output neural networks)
$(n, a_1, a_2, 1)$ *ReLU neural networks have expressive number* a_1a_2.

Proof.
Again, it is sufficient to prove the theorem in the case $n = 1$ by Property I-1.

We divide the proof into 3 steps:

Step 1: Sort given a_1a_2 data in ascending order and define parameters of the first hidden layer as those partition the whole data into a_1 groups of a_2 elements.

Step 2: Find parameters of the second and the last layers. The parameters are given as a solution of some equations and inequalities.

Step 3: Construct a neural network from the parameters defined in Step 1 and Step 2. Then, we show that the neural network expresses the given data.

Step 1 Given $X \subset \mathbb{R}$ such that $|X| = a_1a_2$ and $f : \mathbb{R} \to \mathbb{R}$, we show there exists a $(1, a_1, a_2, 1)$ neural network A such that $\mathrm{MP}_A(x) = f(x)$ for any $x \in X$. We associate a pair of indices to each element of X so that $x_{0,0} < x_{0,1} < \cdots < x_{0,a_2-1} < x_{1,0} < x_{1,1} < \cdots < x_{1,a_2-1} < \cdots < x_{a_1-1,0} < x_{a_1-1,1} < \cdots < x_{a_1-1,a_2-1}$. Let $b_i := \begin{cases} x_{0,0} - 1 & (i = 0) \\ \dfrac{x_{i-1,a_2-1} + x_{i,0}}{2} & (0 < i < a_1) \\ x_{a_1-1,a_2-1} + 1 & (i = a_1) \end{cases}$, so we have $b_0 < x_{0,0} < \cdots < x_{0,a_2-1} < b_1 < x_{1,0} < \cdots < x_{1,a_2-1} < b_2 < \cdots < b_{a_1-1} < x_{a_1-1,0} < \cdots < x_{a_1-1,a_2-1} < b_{a_1}$ and divide input data into a_1 groups, that is, $\{x_{i,0}, \ldots, x_{i,a_2-1}\}_{0 \le i < a_1}$. Define $W_1 := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{a_1 \times 1}$ and $\boldsymbol{b}_1 := -\begin{pmatrix} b_0 \\ \vdots \\ b_{a_1-1} \end{pmatrix} \in \mathbb{R}^{a_1}$ as parameters of the first layer. Then the function $F_1 : \mathbb{R} \to \mathbb{R}^{a_1}$ corresponding to the first layer is written by $F_1(x) = \sigma(W_1x + \boldsymbol{b}_1)$. So, we can write $F_1(x_{i,j}) = \begin{pmatrix} x_{i,j} - b_0 \\ \vdots \\ x_{i,j} - b_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ for any i and j.

Step 2 We show that the neural network can be constructed from the values of $k_{i,j} \in \mathbb{R}$, $\boldsymbol{w}_j \in \mathbb{R}^{a_1}$, $c_j \in \mathbb{R}$ and $C \in \mathbb{R}$ $(0 \le i < a_1,\ 0 \le j < a_2)$ satisfying the following properties:

(1) $(-1)^i x'_{i,j-1} < (-1)^i k_{i,j} < (-1)^i x'_{i,j}$

(2) ${}^t\boldsymbol{w}_j F_1(k_{i,j}) = c_j$

(3) $\displaystyle\sum_{u=0}^{j} ({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) + C = f(x'_{i,j})$

where $x'_{i,j} := \begin{cases} x_{i,j} & (i \bmod 2 = 0) \\ x_{i,a_2-1-j} & (i \bmod 2 = 1) \end{cases}$ and $(x'_{i,-1},\ x'_{i,a_2}) := \begin{cases} \left(\dfrac{b_i + x'_{i,0}}{2},\ \dfrac{x'_{i,a_2-1} + b_{i+1}}{2}\right) & (i \bmod 2 = 0) \\ \left(\dfrac{b_{i+1} + x'_{i,0}}{2},\ \dfrac{x'_{i,a_2-1} + b_i}{2}\right) & (i \bmod 2 = 1) \end{cases}$.

The variables $\boldsymbol{w}_j$ and c_j characterize the wights and bias of the second hidden layer, so the output of second hidden neuron j is ${}^t\boldsymbol{w}_j F_1(x) - c_j$ when input x is given. Then, the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^{a_1} \mid {}^t\boldsymbol{w}_j \boldsymbol{x} = c_j\}$ are the boundary of the linear regions divided by the ReLU activation of the neuron j. The variable $k_{i,j}$ is the intersection point of each hyperplane with the image of the line segment $x'_{i,j-1}\ x'_{i,j}$ by F_1 as Fig. 2. The properties (1), (2) characterize them.

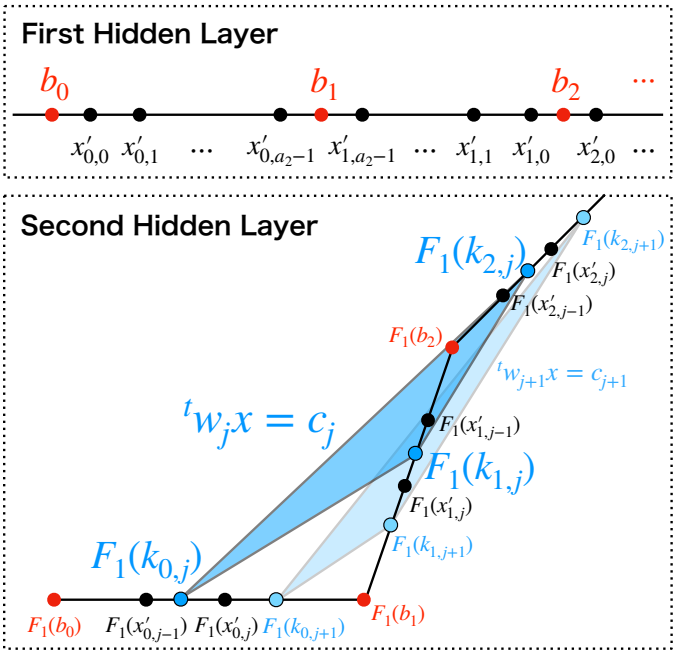


Figure 2: The parameters of the first and the second hidden layers. In the first layer, the parameters $b_0, \ldots, b_{a_1}$ divide input data into a_1 groups of a_2 elements. The parameters b_0, b_{a_1} do not divide the inputs but are needed to define specific hyperplanes in the second layer. In the second layer, the elements of each group are separated individually by the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^{a_1} \mid {}^t\boldsymbol{w}_j\boldsymbol{x} = c_j\}$ defined by $F_1(k_{0,j}), \ldots, F_1(k_{a_1-1,j})$ for any j.

The variable C characterize the bias of the last layer and the property (3) means the neural network constructed by these variables returns the correct outputs. First, we can write $\sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) = \sum_{u=0}^{a_2-1} s_{u,i,j}\sigma(s_{u,i,j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u))$ where $s_{u,i,j} = \operatorname{sgn}(j-u+1/2)\operatorname{sgn}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u)$ where $\operatorname{sgn} : \mathbb{R} \to \mathbb{R}$ is defined as $\operatorname{sgn}(x) = \begin{cases} 1 & (0 < x) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}$. Then, each hyperplane $\{\boldsymbol{x} \in \mathbb{R}^{a_2-1} \mid {}^t\boldsymbol{w}_u\boldsymbol{x} = c_u\}$ separates the points $F_1(x'_{i,0}), \ldots, F_1(x'_{i,a_2-1})$ into $\{F_1(x'_{i,j}) \mid j < u\}$ and $\{F_1(x'_{i,j}) \mid u \le j\}$ for any i. So, we have $s_{u,i,j} = s_{u,i,j'}$ regardless of the signs of $j-u$ and $j'-u$ for any $0 \le j, j' < a_2$. Furthermore, by the definition of $x'_{i,j}$ and the continuity of F_1, we have $s_{u,i,j} = s_{u,i+1,j}$ for any $0 \le i < a_2 - 1$. Therefore, $s_{u,i,j}$ is only depend on the variable u, thus, we can write $\sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) + C = \sum_{u=0}^{a_2-1} s_u\sigma(s_u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u)) + C$.

The right-hand side just means the output of a $(1, a_1, a_2, 1)$ neural network. So, to show the property (3), we can find the parameters of the neural network.

We give values satisfying (1) – (3) as follows:

- $C := \min_{0 \le i < a_1} f(x'_{i,0}) - 1$.
- The values of c_j and $k_{i,j}$ are defined recursively. By Table 1, we see the order to determine these values is c_0, $k_{0,0}$, $\ldots$, $k_{a_1-1,0}$, c_1, $k_{0,1}$, $\ldots$, $k_{a_1-1,1}$, c_2, $\ldots$.

  If c_t and $k_{i,t}$ are defined for any $0 \le t < j$ and any $0 \le i < a_1$, we can define the following

$M_{i,j} \in \mathbb{R}$, $B_{i,j}, D_{i,j} : \mathbb{R} \to \mathbb{R}$ and $E_{i,j}, E'_{i,j} \in \mathbb{R}$.

$$M_{i,j} := \prod_{s=0}^{i-1} \frac{b_{s+1} - k_{s,j}}{k_{s,j} - b_s}$$

$$B_{i,j}(x) := \frac{(-1)^i (x - k_{i,j})}{k_{i,j} - b_i}$$

$$D_{i,j}(x) := f(x) - C - \sum_{t=0}^{j-1} B_{i,t}(x) M_{i,t} c_t$$

$$E_{i,j} := \frac{\max\{x'_{i,j-1}, x'_{i,j}\} - b_i}{(-1)^i (x'_{i,j} - x'_{i,j-1})} |D_{i,j}(x'_{i,j})|$$

$$E'_{i,j} := \frac{\max\{x'_{i,j-1}, x'_{i,j}\} - b_i}{(-1)^i (x'_{i,j+1} - x'_{i,j})} |D_{i,j}(x'_{i,j+1})|$$

Let $K_{i,j} := \max\{|D_{i,j}(x'_{i,j})|, E_{i,j}, E'_{i,j}\} + 1$. We define c_j as

$$c_j := (-1)^j \max_{0 \le i < a_1} \left(\prod_{s=0}^{i-1} \frac{\max\{x'_{s,j-1}, x'_{s,j}\} - b_s}{b_{s+1} - \max\{x'_{s,j-1}, x'_{s,j}\}} \right) K_{i,j}$$

Next, we define $k_{i,j}$ as

$$k_{i,j} := \frac{(-1)^i x'_{i,j} M_{i,j} c_j + b_i D_{i,j}(x'_{i,j})}{(-1)^i M_{i,j} c_j + D_{i,j}(x'_{i,j})}$$

So, we have

$$(k_{i,j} - b_i) D_{i,j}(x'_{i,j}) = (-1)^i (x'_{i,j} - k_{i,j}) M_{i,j} c_j \tag{e1}$$

Table 1: The dependencies of indices of variables c_s, $k_{s,t}$

variable	c_t	$k_{s,t}$
$M_{i,j}$	$-$	$s < i \wedge t = j$
$B_{i,j}(x)$	$-$	$s = i \wedge t = j$
$D_{i,j}(x)$	$t < j$	$s = i \wedge t < j$
$E_{i,j}$	$t < j$	$s = i \wedge t < j$
$E'_{i,j}$	$t < j$	$s = i \wedge t < j$
$K_{i,j}$	$t < j$	$s = i \wedge t < j$
c_j	$t < j$	$s < a_1 \wedge t < j$
$k_{i,j}$	$t \le j$	$(s = i \wedge t < j) \vee (s < i \wedge t = j)$

We see in the table that the variable c_j is defined by the variables c_t satisfying $t < j$ and the variable $k_{s,t}$ satisfying $s < a_1 \wedge t < j$, and the variable $k_{i,j}$ is defined by the variables c_t satisfying $t \le j$ and the variables $k_{s,t}$ satisfying $(s = i \wedge t < j) \vee (s < i \wedge t = j)$. So, we can define c_j and $k_{i,j}$ in the order of c_0, $k_{0,0}$, …, $k_{a_1-1,0}$, c_1, $k_{0,1}$, …, $k_{a_1-1,1}$, c_2, ….

- Define $\boldsymbol{w}_j := {}^t(w_{0,j}, \ldots, w_{a_1-1,j})$, where

$$w_{i,j} := \begin{cases} \dfrac{c_j}{k_{0,j} - b_0} & (i = 0) \\ (-1)^i \dfrac{(k_{i,j} - k_{i-1,j}) M_{i-1,j} c_j}{(k_{i,j} - b_i)(k_{i-1,j} - b_{i-1})} & (i > 0) \end{cases}$$

Then the values satisfy the following properties (see Appendix A.2).

(4) $\sum_{s=0}^{i} w_{s,j} = (-1)^i \frac{M_{i,j}c_j}{k_{i,j} - b_i}$ holds for any i and j.

(5) For any i and j, if $(-1)^s x'_{s,j-1} < (-1)^s k_{s,j} < (-1)^s x'_{s,j}$ whenever $s < i$, then $M_{i,j}(-1)^j c_j \geq K_{i,j}$.

(6) For any j, if $(-1)^j D_{i,j}(x'_{i,j}) > 0$ holds for any i, then $(-1)^i x'_{i,j-1} < (-1)^i k_{i,j} < (-1)^i x'_{i,j}$ holds for any i.

(7) $(-1)^j D_{i,j}(x'_{i,j}) > 0$ for any i and j.

Now, we show (1) – (3).

The property (1) can be shown by (6) and (7) immediately.

We show (2) by induction on i. We can write $F_1(k_{i,j}) = \begin{pmatrix} k_{i,j} - b_0 \\ \vdots \\ k_{i,j} - b_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

When $i = 0$, ${}^t\boldsymbol{w}_j F_1(k_{0,j}) = w_{0,j}(k_{0,j} - b_0) = c_j$.

When $i > 0$, we have ${}^t\boldsymbol{w}_j F_1(k_{i-1,j}) = c_j$ by the induction hypothesis. Thus,

$$\begin{aligned}
{}^t\boldsymbol{w}_j F_1(k_{i,j}) - c_j &= {}^t\boldsymbol{w}_j F_1(k_{i,j}) - {}^t\boldsymbol{w}_j F_1(k_{i-1,j}) \\
&= \sum_{s=0}^{i} w_{s,j}(k_{i,j} - b_s) - \sum_{s=0}^{i-1} w_{s,j}(k_{i-1,j} - b_s) \\
&= w_{i,j}(k_{i,j} - b_i) + \sum_{s=0}^{i-1} w_{s,j}(k_{i,j} - k_{i-1,j}) \\
&= (-1)^i \frac{k_{i,j} - k_{i-1,j}}{k_{i-1,j} - b_{i-1}} M_{i-1,j}c_j + (k_{i,j} - k_{i-1,j}) \sum_{s=0}^{i-1} w_{s,j} \\
&= (k_{i,j} - k_{i-1,j})(\sum_{s=0}^{i-1} w_{s,j} - (-1)^{i-1} \frac{M_{i-1,j}c_j}{k_{i-1,j} - b_{i-1}})
\end{aligned}$$

Because $\sum_{s=0}^{i} w_{s,j} = (-1)^i \frac{M_{i,j}c_j}{k_{i,j} - b_i}$ holds for any i, we have ${}^t\boldsymbol{w}_j F_1(k_{i,j}) - c_j = 0$. Therefore, ${}^t\boldsymbol{w}_j F_1(k_{i,j}) = c_j$ holds for any i.

We show (3).

$$\begin{aligned}
\sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) + C &= \sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - {}^t\boldsymbol{w}_u F_1(k_{i,u})) + C \\
&= \sum_{u=0}^{j}\sum_{s=0}^{i} w_{s,u}(x'_{i,j} - k_{i,u}) + C \\
&= \sum_{u=0}^{j}(x'_{i,j} - k_{i,u})\sum_{s=0}^{i} w_{s,u} + C \\
&= \sum_{u=0}^{j} B_{i,u}(x'_{i,j})M_{i,u}c_u + C \\
&= B_{i,j}(x'_{i,j})M_{i,j}c_j + \sum_{u=0}^{j-1} B_{i,u}(x'_{i,j})M_{i,u}c_u + C \\
&= \frac{(-1)^i(x'_{i,j} - k_{i,j})}{k_{i,j} - b_i}M_{i,j}c_j - D_{i,j}(x'_{i,j}) + f(x'_{i,j})
\end{aligned}$$

Since $(k_{i,j} - b_i)D_{i,j}(x'_{i,j}) = (-1)^i(x'_{i,j} - k_{i,j})M_{i,j}c_j$ (see (e1)), we have $\sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) + C = f(x'_{i,j})$.

Step 3 We construct a neural network A satisfying $\mathrm{MP}_A(x'_{i,j}) = f(x'_{i,j})$ for any $x'_{i,j} \in X$.

Define $W_2 := \begin{pmatrix} {}^t\boldsymbol{w}_0 \\ -{}^t\boldsymbol{w}_1 \\ {}^t\boldsymbol{w}_2 \\ \vdots \\ (-1)^{a_2-1} \cdot {}^t\boldsymbol{w}_{a_2-1} \end{pmatrix}$, $\boldsymbol{c} := -\begin{pmatrix} c_0 \\ -c_1 \\ c_2 \\ \vdots \\ (-1)^{a_2-1}c_{a_2-1} \end{pmatrix}$, $W_3 := (1, -1, 1, \ldots, (-1)^{a_2-1})$, $d = C$ and $A := ((W_1, \boldsymbol{b}), (W_2, \boldsymbol{c}), (W_3, d))$ then we can write

$$\begin{aligned}
\mathrm{MP}_A(x'_{i,j}) &= \sum_{u=0}^{a_2-1}(-1)^u\sigma((-1)^u \cdot {}^t\boldsymbol{w}_u F_1(x'_{i,j}) - (-1)^u c_u) + C \\
&= \sum_{u=0}^{a_2-1}(-1)^u\sigma((-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u)) + C
\end{aligned}$$

Then we show $(-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) \geq 0$ if and only if $u \leq j$.

$$\begin{aligned}
(-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) &= (-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - {}^t\boldsymbol{w}_u F_1(k_{i,u})) \\
&= (-1)^u(\sum_{s=0}^{i} w_{s,u}(x'_{i,j} - k_{i,u})) \\
&= (-1)^u(x'_{i,j} - k_{i,u})\sum_{s=0}^{i} w_{s,u} \\
&= (-1)^i(x'_{i,j} - k_{i,u})\frac{M_{i,u}(-1)^u c_u}{k_{i,u} - b_i}
\end{aligned}$$

Now, we have $M_{i,u} > 0$, $(-1)^u c_u > 0$ and $k_{i,u} - b_i > 0$. By (1), $(-1)^i x'_{i,-1} < \cdots < (-1)^i x'_{i,u-1} < (-1)^i k_{i,u} < (-1)^i x'_{i,u} < \cdots < (-1)^i x'_{i,a_2-1}$ holds, so we have $u \leq j \Leftrightarrow (-1)^i k_{i,u} < (-1)^i x_{i,j}$. Thus, $(-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) \geq 0$ holds if and only if $u \leq j$.

Then, we have

$$\begin{aligned}\mathrm{MP}_A(x'_{i,j}) &= \sum_{u=0}^{a_2-1}(-1)^u\sigma((-1)^u({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u)) + C \\ &= \sum_{u=0}^{j}({}^t\boldsymbol{w}_u F_1(x'_{i,j}) - c_u) + C = f(x'_{i,j}) \quad (\because (3))\end{aligned}$$

Therefore, $(1,\, a_1,\, a_2,\, 1)$ ReLU neural networks have expressive number a_1a_2.

This theorem holds only on neural networks with a single output. Now, we generalize it with any outputs.

Theorem 4 (Lower Bound)
$(n,\, a_1,\, a_2,\, m)$ *ReLU neural networks have expressive number* $\max\{a_1(a_2 \operatorname{div} m) + a_2 \bmod m,\, a_2+1\}$.

Proof.
It is sufficient to show $(1,\, a_1,\, a_2,\, m)$ ReLU neural networks have both expressive numbers $a_1(a_2 \operatorname{div} m) + a_2 \bmod m$ and $a_2 + 1$.

First, we show the latter. Given $X \subset \mathbb{R}$ such that $|X| = a_2 + 1$ and $f : \mathbb{R} \to \mathbb{R}^m$. Let $W_1 := {}^t(1,\, 0,\, \ldots,\, 0) \in \mathbb{R}^{a_1 \times 1}$, $\boldsymbol{b}_1 := {}^t(-\min X,\, 0,\, \ldots,\, 0) \in \mathbb{R}^{a_1}$ and $g(x) := \sigma(W_1 x + \boldsymbol{b}_1)$. Since the restriction of g to X is injective, there exists $h : \mathbb{R}^{a_1} \to \mathbb{R}$ such that $h(g(x)) = x$ for any $x \in X$. By Property I-2, $(a_1,\, a_2,\, m)$ ReLU neural networks have expressive number $a_2 + 1$. Thus, there exists an $(a_1,\, a_2,\, m)$ neural network A such that $\mathrm{MP}_A(y) = (f \circ h)(y)$ holds for any $y \in g(X)$. Let $B := ((W_1, \boldsymbol{b}_1), A)$ be a $(1,\, a_1,\, a_2,\, m)$ neural network. Then

$$\mathrm{MP}_B(x) = \mathrm{MP}_A(g(x)) = (f \circ h)(g(x)) = f(x)$$

holds for any $x \in X$. Therefore, $(1,\, a_1,\, a_2,\, m)$ ReLU neural networks have expressive number a_2+1.

Finally, we show that $(1,\, a_1,\, a_2,\, m)$ ReLU neural networks have expressive number $a_1(a_2 \operatorname{div} m) + a_2 \bmod m$. Let $p := a_1(a_2 \operatorname{div} m)$ and $q := a_2 \bmod m$. Given $X \subset \mathbb{R}$ such that $|X| = p + q$ and $f : \mathbb{R} \to \mathbb{R}^m$. Then let $X' := \{x_1,\, \ldots,\, x_p\}$ and $X'' := \{x_{p+1},\, \ldots,\, x_{p+q}\}$ where $x_1,\, \ldots,\, x_{p+q}$ are all elements of X such that $x_1 < \cdots < x_{p+q}$. Define $f_i : \mathbb{R} \to \mathbb{R}$ as $f_i(x) := y_i(x)$ for any $0 \le i < m$ where $f(x) = {}^t(y_0(x),\, \ldots,\, y_{m-1}(x))$. Then, by Theorem 3, we have $\mathrm{MP}_{A_i}(x) = f_i(x)$ for any $x \in X'$ where $A_i = ((W_{1,i}, \boldsymbol{b}_i), (W_{2,i}, \boldsymbol{c}_i), (W_{3,i}, d_i))$ is a $(1,\, a_1,\, a_2 \operatorname{div} m,\, 1)$ neural network given in the proof of the theorem. By the definition of $W_{1,i}$ and $\boldsymbol{b}_i$ in Theorem 3, we have $W_{1,i} = W_{1,j}$ and $\boldsymbol{b}_i = \boldsymbol{b}_j$ for any $0 \le i, j < m$. Thus, let $W_2 := \begin{pmatrix} W_{2,0} \\ \vdots \\ W_{2,m-1} \end{pmatrix}$, $\boldsymbol{c} := \begin{pmatrix} \boldsymbol{c}_0 \\ \vdots \\ \boldsymbol{c}_{m-1} \end{pmatrix}$, $W_3 := \begin{pmatrix} W_{3,0} & & O \\ & \ddots & \\ O & & W_{3,m-1} \end{pmatrix}$ and $\boldsymbol{d} := \begin{pmatrix} d_0 \\ \vdots \\ d_{m-1} \end{pmatrix}$. $B := ((W_{1,0}, \boldsymbol{b}_0), (W_2, \boldsymbol{c}), (W_3, \boldsymbol{d}))$ is a $(1,\, a_1,\, m(a_2 \operatorname{div} m),\, m)$ neural network and $\mathrm{MP}_B(x) = \begin{pmatrix} y_0(x) \\ \vdots \\ y_{m-1}(x) \end{pmatrix} = f(x)$ holds for any $x \in X'$.

Let $W'_2 := \begin{pmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{q \times a_1}$, $\boldsymbol{c}' := -\begin{pmatrix} x_p - b_{a_1-1} \\ \vdots \\ x_{p+q-1} - b_{a_1-1} \end{pmatrix} \in \mathbb{R}^q$ where b_{a_1-1} is as defined in the proof of Theorem 3, and $W'_3 := (\boldsymbol{w}'_{3,0},\, \ldots,\, \boldsymbol{w}'_{3,q-1})$ where $\boldsymbol{w}'_{3,i} := \dfrac{1}{x_{p+i+1} - x_{p+i}}(f(x_{p+i+1}) - (\mathrm{MP}_B(x_{p+i+1}) + \sum_{j=0}^{i-1}(x_{p+i+1} - x_{p+j})\boldsymbol{w}'_{3,j}))$ for any $0 \le i < q$. Let $W''_2 := \begin{pmatrix} W_2 \\ W'_2 \end{pmatrix}$, $\boldsymbol{c}'' := \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{c}' \end{pmatrix}$,

$W_3'' := (W_3\, W_3')$ and $B' := ((W_{1,0}, \boldsymbol{b}_0), (W_2'', \boldsymbol{c}''), (W_3'', \boldsymbol{d}))$ be a $(1,\, a_1,\, m(a_2 \operatorname{div} m) + a_2 \bmod m,\, m)$ neural network. Then, $\mathrm{MP}_{B'}(x) = f(x)$ holds for any $x \in X' \cup X'' = X$. Therefore, $(1,\, a_1,\, a_2,\, m)$ ReLU neural networks have expressive number $a_1(a_2 \operatorname{div} m) + a_2 \bmod m$.

From this theorem, the maximum expressive number of $(n,\, a_1,\, a_2,\, m)$ ReLU neural networks is greater than or equal to $a_1(a_2 \operatorname{div} m) + a_2 \bmod m$ and $a_2 + 1$. If $a_1 \leq m$ or $a_2 < m$ then the lower bound is equal to $a_2 + 1$, that is also equal to the lower bound of $(n,\, a_2,\, m)$ single hidden layer ReLU neural networks (Property I-2). So, this theorem suggests that when we use two hidden layer neural networks, the numbers of each hidden neurons should be greater than the output dimension.

Besides, we see the maximum expressive number is in $O(a_1 a_2/m)$. In general, the output dimension m is a constant in machine learning. Thus, we can write the number is in $O(a_1 a_2)$. Furthermore by Theorem 2, the maximum expressive number N of $(n,\, a_1,\, a_2,\, m)$ ReLU neural networks satisfies $N \in \Theta(a_1 a_2)$.

6 Conclusion

We have shown an upper bound of the maximum expressive number of multilayer ReLU neural networks and a lower bound of that of two hidden layer ReLU neural networks. Our result suggests the maximum expressive number N of $(n,\, a_1,\, a_2,\, m)$ ReLU neural networks satisfies $a_1(a_2 \operatorname{div} m) + a_2 \bmod m \leq N \leq (a_1 + 1)(a_2 + 2)$, i.e., $N \in \Theta(a_1 a_2)$. In other words, the maximum expressive number of two hidden layer ReLU neural networks is proportional to the product of the numbers of each hidden layer's neurons. In particular, if $m = 1$, that is, the neural networks have a single output, then the maximum expressive number N satisfies $a_1 a_2 \leq N \leq (a_1+1)(a_2+2)$ i.e. $N \sim a_1 a_2$. Namely, $(n,\, a_1,\, a_2,\, 1)$ ReLU neural networks can express any $a_1 a_2$ data and there exists $(a_1 + 1)(a_2 + 2) + 1$ data that the neural networks cannot express.

Furthermore, from our result, the maximum expressive number of $(n,\, a_1,\, \ldots,\, a_l,\, m)$ ReLU neural networks is in $o(\prod_{i=1}^{l} a_i)$. In other words, the upper bound is exponential to the number of layers when the numbers of each hidden layer's neurons are the same. Other measures of the expressive power, such as linear regions, also increase exponentially to the number of layers ([14, 12, 11]). So, we conjecture the maximum expressive number is also in $\Theta(\prod_{i=1}^{l} a_i)$.

References

[1] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[2] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *Neural Networks and Learning Systems, IEEE Transactions on*, 25:1553–1565, 08 2014.

[3] Kevin K. Chen. The upper bound on knots in neural networks, 2016.

[4] Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

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

[6] Kenta Inoue. Expressive power of neural networks by the number of data that can be expressed. *IEICE*, J102-D(6), 2019. In Japanese.

[7] Wolfgang Maass. Neural nets with superlinear VC-dimension. *Neural Computation*, 6(5):877–884, 1994.

[8] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[9] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems Volume 2*, NIPS'14, pages 2924–2932, 2014.

[10] Razvan Pascanu, Guido Montúfar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations, 2013.

[11] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems 29*, pages 3360–3368. Curran Associates, Inc., 2016.

[12] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2847–2854, 2017.

[13] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. In *International Conference on Learning Representations*, 2018.

[14] Thiago Serra and Christian Tjandraatmadja. Bounding and counting linear regions of deep neural networks. *International Conference on Learning Representations*, 2018.

[15] Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. Neural network studies, 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35:826–833, 1995.

A Appendix

A.1 Generalization of Property I-2

We have already shown Property I-2 in [6], that is, (n, k, m) ReLU neural networks have expressive number $k + 1$. Similar result is shown in [5], i.e., if an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ achieves 0 and 1, then, $(r, n, 1)$ neural networks express any n data. In other words, if there exists $M \in \mathbb{R}$ such that $(\forall x \leq M, \sigma(x) = 0)$ and $(\forall x \geq M, \sigma(x) = 1)$, then, (n, k, m) σ neural networks have expressive number k. These two properties are independent, but we can generalize the properties into one theorem.

Theorem 5 (Expressive number of single hidden neural networks)
We assume that the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies the follows: there exist $a, c \in \mathbb{R}$ and $y \in \mathbb{R}$ satisfying the following properties.

- $\forall x \leq a, \sigma(x) = y$.
- $\sigma(c) \neq y$.

Then, (n, k, m) neural networks have expressive number $k + 1$ for any $n, k, m \in \mathbb{N}$.

Proof.
By assumptions, there exist $a, c, y \in \mathbb{R}$ satisfying $(\forall x \leq a, \sigma(x) = y)$ and $\sigma(c) \neq y$. Then, we have $a < c$. It is sufficient to prove the theorem in the case $n = 1$ by Property I-1. Given $X \subset \mathbb{R}$ such that $|X| = k + 1$ and $f : \mathbb{R} \to \mathbb{R}^m$, we show there exists an $(1, k, m)$ neural network A such that $\mathrm{MP}_A(x) = f(x)$ holds for any $x \in X$. Let $x_1, \ldots, x_{k+1} \in X$ be the increasing sequence of all elements of X. We show this theorem by induction on k.

When $k = 0$, it holds trivially.

When $k > 0$, by induction hypothesis, there exists a $(1, k-1, m)$ neural network A such that $f(x_i) = \mathrm{MP}_A(x_i)$ holds for any $i < k+1$. Now we define an (n, k, m) neural network $B := ((V_1, \boldsymbol{c}_1), (V_2, \boldsymbol{c}_2))$ where $A = ((W_1, \boldsymbol{b}_1), (W_2, \boldsymbol{b}_2))$, $V_1 := \begin{pmatrix} W_1 \\ \frac{c-a}{x_{k+1}-x_k} \end{pmatrix}$, $\boldsymbol{c}_1 := \begin{pmatrix} \boldsymbol{b}_1 \\ -\frac{c-a}{x_{k+1}-x_k}x_k + a \end{pmatrix}$, $V_2 := (W_2, \frac{f(x_{k+1}) - \mathrm{MP}_A(x_{k+1}))}{\sigma(c) - y})$, and $\boldsymbol{c}_2 := \boldsymbol{b} - \frac{y(f(x_{k+1}) - \mathrm{MP}_A(x_{k+1}))}{\sigma(c) - y}$. Then, we have $\mathrm{MP}_B(x_i) = f(x_i)$ for any $i \leq k+1$.

Therefore, $(1, k, m)$ neural networks have expressive number $k+1$.

A.2 Proof of properties (4) – (7) in Theorem 3

The property (4) is shown by induction on i.

When $i = 0$, it holds trivially. For any $i > 0$,

$$\begin{aligned}
\sum_{s=0}^{i} w_{s,j} &= w_{i,j} + \sum_{s=0}^{i-1} w_{s,j} \\
&= (-1)^i \left(\frac{(k_{i,j} - k_{i-1,j}) M_{i-1,j} c_j}{(k_{i,j} - b_i)(k_{i-1,j} - b_{i-1})} - \frac{M_{i-1,j} c_j}{k_{i-1,j} - b_{i-1}}\right) \\
&= (-1)^i \left(\frac{k_{i,j} - k_{i-1,j}}{k_{i,j} - b_i} - 1\right) \frac{M_{i-1,j} c_j}{k_{i-1,j} - b_{i-1}} \\
&= (-1)^i \frac{M_{i,j} c_j}{k_{i,j} - b_i}
\end{aligned}$$

holds as required.

Next, we show (5). By hypothesis, we have $\min\{x'_{s,j-1}, x'_{s,j}\} < k_{s,j} < \max\{x'_{s,j-1}, x'_{s,j}\}$ for any $s < i$. Then, $b_s < k_{s,j} < b_{s+1}$ and $M_{i,j} > 0$ hold. Thus,

$$\begin{aligned}
M_{i,j}(-1)^j c_j &\geq M_{i,j} \left(\prod_{s=0}^{i-1} \frac{\max\{x'_{s,j-1},\, x'_{s,j}\} - b_s}{b_{s+1} - \max\{x'_{s,j-1},\, x'_{s,j}\}} \right) K_{i,j} \\
&\geq M_{i,j} \left(\prod_{s=0}^{i-1} \frac{k_{s,j} - b_s}{b_{s+1} - k_{s,j}} \right) K_{i,j} = K_{i,j}
\end{aligned}$$

for $0 \leq i < a_1$.

The property (6) is proved by the complete induction on i. We show the second inequality first:

$$\begin{aligned}
(-1)^i (x'_{i,j} - k_{i,j}) &= (-1)^i \left(x'_{i,j} - \frac{(-1)^i x'_{i,j} M_{i,j} c_j + b_i D_{i,j}(x'_{i,j})}{(-1)^i M_{i,j} c_j + D_{i,j}(x'_{i,j})}\right) \\
&= \frac{(-1)^i (x'_{i,j} - b_i) D_{i,j}(x'_{i,j})}{(-1)^i M_{i,j} c_j + D_{i,j}(x'_{i,j})} \\
&= \frac{(x'_{i,j} - b_i)(-1)^j D_{i,j}(x'_{i,j})}{M_{i,j}(-1)^j c_j + (-1)^i (-1)^j D_{i,j}(x'_{i,j})} \\
&= \frac{(x'_{i,j} - b_i) |D_{i,j}(x'_{i,j})|}{M_{i,j}(-1)^j c_j + (-1)^i |D_{i,j}(x'_{i,j})|} > 0
\end{aligned}$$

The last inequality follows from $x'_{i,j} - b_i > 0$ and $M_{i,j}(-1)^j c_j \geq K_{i,j} > |D_{i,j}(x'_{i,j})|$ ($\because$ induction hypothesis and (5)).

Then we show the first one. From the definition of $k_{i,j}$, we can write

$$(k_{i,j} - b_i) D_{i,j}(x'_{i,j}) = (-1)^i (x'_{i,j} - k_{i,j}) M_{i,j} c_j$$

(see (e1)). Thus,

$$\begin{aligned}
&\frac{k_{i,j}-b_i}{(-1)^i(x'_{i,j}-k_{i,j})} = \frac{M_{i,j}c_j}{D_{i,j}(x'_{i,j})} = \frac{M_{i,j}(-1)^j c_j}{(-1)^j D_{i,j}(x'_{i,j})} \\
&\geq \frac{K_{i,j}}{|D_{i,j}(x'_{i,j})|} > \frac{E_{i,j}}{|D_{i,j}(x'_{i,j})|} = \frac{\max\{x'_{i,j-1}, x'_{i,j}\}-b_i}{(-1)^i(x'_{i,j}-x'_{i,j-1})} \\
&\geq \frac{x'_{i,j-1}-b_i}{(-1)^i(x'_{i,j}-x'_{i,j-1})} \quad (\because (-1)^i(x'_{i,j}-x'_{i,j-1})>0)
\end{aligned}$$

Since the denominators of both the first and the last terms are positive, we have

$$(-1)^i(x'_{i,j}-x'_{i,j-1})(k_{i,j}-b_i) > (-1)^i(x'_{i,j}-k_{i,j})(x'_{i,j-1}-b_i)$$

which is simplified to

$$(-1)^i k_{i,j}(x'_{i,j}-b_i) > (-1)^i x'_{i,j-1}(x'_{i,j}-b_i)$$

Since $x'_{i,j}-b_i>0$, we have $(-1)^i k_{i,j} > (-1)^i x'_{i,j-1}$.

The property (7) is shown by the complete induction on j.

When $j=0$, we have $(-1)^j D_{i,j}(x'_{i,j}) = f(x'_{i,0}) - C > 0$.

When $j>0$,

$$\begin{aligned}
(-1)^j D_{i,j}(x'_{i,j}) &= (-1)^j (f(x'_{i,j}) - C - \sum_{t=0}^{j-1} B_{i,t}(x'_{i,j})M_{i,t}c_t) \\
&= (-1)^j (D_{i,j-1}(x'_{i,j}) - B_{i,j-1}(x'_{i,j})M_{i,j-1}c_{j-1}) \\
&= (-1)^j D_{i,j-1}(x'_{i,j}) + B_{i,j-1}(x'_{i,j})M_{i,j-1}(-1)^{j-1}c_{j-1}
\end{aligned}$$

Then, by the induction hypothesis and (6), we have $(-1)^i x'_{i,j-2} < (-1)^i k_{i,j-1} < (-1)^i x'_{i,j-1} < (-1)^i x'_{i,j}$. Thus,

$$B_{i,j-1}(x'_{i,j}) = \frac{(-1)^i(x'_{i,j}-k_{i,j-1})}{k_{i,j-1}-b_i} > 0$$

Then, because of the induction hypothesis, (6) and (5), we have $M_{i,j-1}(-1)^{j-1}c_{j-1} \geq K_{i,j-1} > E'_{i,j-1}$. Thus,

$$\begin{aligned}
(-1)^j D_{i,j}(x'_{i,j}) &= (-1)^j D_{i,j-1}(x'_{i,j}) + B_{i,j-1}(x'_{i,j})M_{i,j-1}(-1)^{j-1}c_{j-1} \\
&> (-1)^j D_{i,j-1}(x'_{i,j}) + B_{i,j-1}(x'_{i,j})E'_{i,j-1}
\end{aligned}$$

Then

$$\begin{aligned}
B_{i,j-1}(x'_{i,j})E'_{i,j-1} &= \frac{(-1)^i(x'_{i,j}-k_{i,j-1})}{k_{i,j-1}-b_i} \cdot \frac{\max\{x'_{i,j-2}, x'_{i,j-1}\}-b_i}{(-1)^i(x'_{i,j}-x'_{i,j-1})}|D_{i,j-1}(x'_{i,j})| \\
&= \frac{(-1)^i(x'_{i,j}-k_{i,j-1})}{(-1)^i(x'_{i,j}-x'_{i,j-1})} \cdot \frac{\max\{x'_{i,j-2}, x'_{i,j-1}\}-b_i}{k_{i,j-1}-b_i}|D_{i,j-1}(x'_{i,j})| \\
&> |D_{i,j-1}(x'_{i,j})|
\end{aligned}$$

The last inequality follows from $(-1)^i k_{i,j-1} < (-1)^i x'_{i,j-1}$ and $k_{i,j-1} < \max\{x'_{i,j-2}, x'_{i,j-1}\}$. Thus,

$$(-1)^j D_{i,j}(x'_{i,j}) > (-1)^j D_{i,j-1}(x'_{i,j}) + |D_{i,j-1}(x'_{i,j})| \geq 0$$

Therefore $(-1)^j D_{i,j}(x'_{i,j}) > 0$ holds for any i and j.