ResNet with one-neuron hidden layers is a Universal Approximator

Hongzhou Lin   Stefanie Jegelka
presenter: Shen Yuan

中国人民大學
RENMIN UNIVERSITY OF CHINA
高瓴人工智能学院
Gaoling School of Artificial Intelligence

Outline

Outline

Contribution

The main contribution of this paper is to show that **ResNet with one single neuron per hidden layer is enough to provide universal approximation as the depth goes to infinity**.
More precisely, we show that for any Lebesgue-integrable [1] function $f : \mathbb{R}^d \to \mathbb{R}$, for any $\epsilon > 0$, there exists a ResNet R with ReLU activation and one neuron per hidden layer such that

$$\int_{\mathbb{R}^d} |f(x) - R(x)| \mathrm{d}x \leq \epsilon \tag{1}$$

[1] A function f is Lebesgue-integrable if $\int_{\mathbb{R}^d} |f(x)| \mathrm{d}x < \infty$.

A motivating example

We begin by empirically exploring the difference between narrow fully connected networks, with d neurons per hidden layer, and ResNet via a simple example: **classifying the unit ball in the plane**.

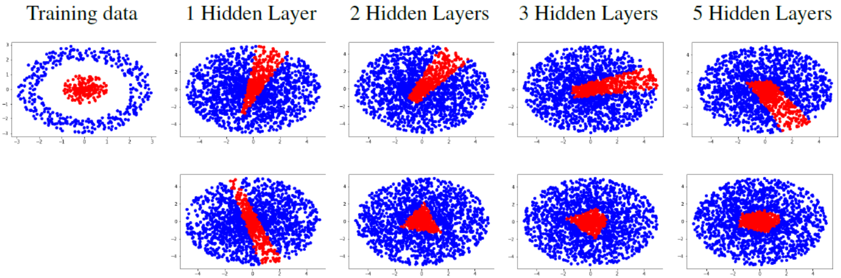


Figure 2: Decision boundaries obtained by training fully connected networks with width $d = 2$ per hidden layer (top row) and ResNet (bottom row) with one neuron in the hidden layers on the unit ball classification problem. The fully connected networks fail to capture the true function, in line with the theory stating that width d is too narrow for universal approximation. ResNet in contrast approximates the function well, empirically supporting our theoretical results.

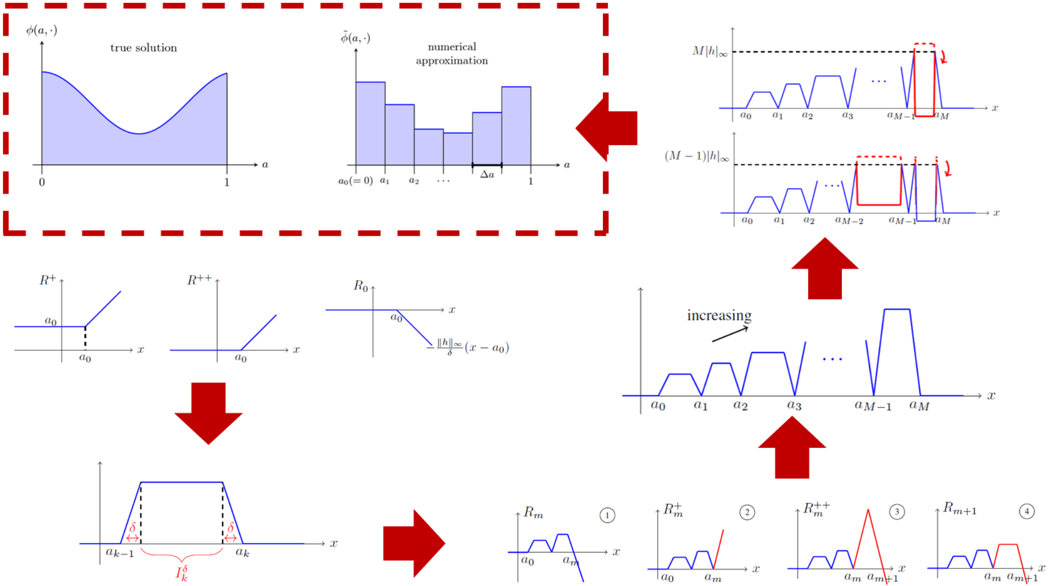
Outline

Universal approximation theorem

Theorem 3.1 (Universal Approximation of ResNet). For any $d \in \mathbb{N}$, the family of ResNet with one-neuron hidden layers and ReLU activation function can universally approximate any $f \in l_1(\mathbb{R}^d)$. In other words, for any $\epsilon > 0$, there is a ResNet R with finitely many layers such that

$$\int_{\mathbb{R}^d} |f(x) - R(x)| \mathrm{d}x \leq \epsilon \tag{2}$$

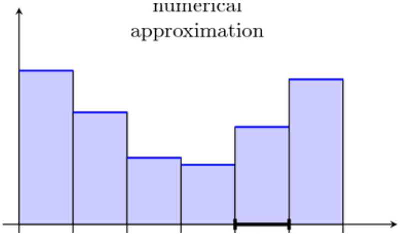
Outline of the proof



Target piecewise constant functions

Given a piecewise constant function h, there is a subdivision $-\infty < a_0 < a_1 < \ldots < a_M < +\infty$ such that

$$h(x) = \sum_{k=1}^{M} h_k \mathbf{1}_{x \in [a_{k-1}, a_k)}, \tag{3}$$



A basic residual block

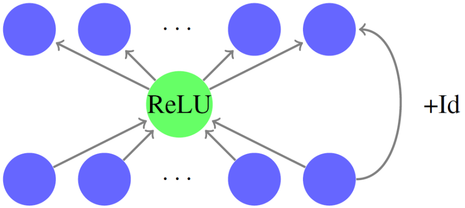


Figure 1: The basic residual block with one neuron per hidden layer.

A basic residual block is a function $\mathcal{T}_{U,V,u}$ from $\mathbb{R}^d$ to $\mathbb{R}^d$ deined by

$$\mathcal{T}_{U,V,u}(x) = V\mathrm{ReLU}(Ux + u) + x = V[Ux + u]_+ + x \tag{4}$$

where $U \in \mathbb{R}^{1\times d}$, $V \in \mathbb{R}^{d\times 1}$, $u \in \mathbb{R}$ and the ReLU actiovation function is defined by

$$\mathrm{ReLU}(x) = \max(x, 0) = [x]_+ \tag{5}$$

A basic residual block

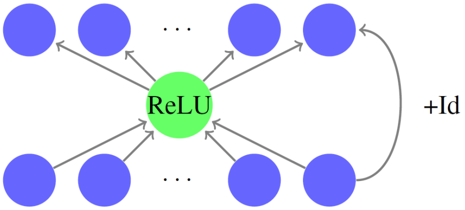


Figure 1: The basic residual block with one neuron per hidden layer.

The resulting ResNet is a combination of several basic residual blocks and a final linear output layer:

$$R(x) = \mathcal{L} \circ \mathcal{T}_N \circ \mathcal{T}_{N-1} \circ \ldots \circ \mathcal{T}_0(x) \tag{6}$$

where $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is a linear operator and $\mathcal{T}_i$ are basic one-neuron residual blocks.

Basic operations

Proposition 3.2 (Basic operations). The following operations are realizable by a single basic residual block of ResNet with one neuron:

- (a) ***Shifting by a constant:*** $R^+ = R + c$ *for any* $c \in \mathbb{R}$;
- (b) ***Min or Max with a constant:*** $R^+ = \min\{R, c\}$ *or* $R^+ = \max\{R, c\}$ *for any* $c \in \mathbb{R}$;
- (c) ***Min or Max with a linear transformation:*** $R^+ = \min\{R, \alpha R + \beta\}$ *(or max) for any* $\alpha, \beta \in \mathbb{R}$;

where R represents the input layer in the basic residual block and R^+ the output layer.

Basic operations

$$\mathcal{T}_{U,V,u}(x) = V[Ux + u]_+ + x \tag{7}$$

- (a) $V = c,\ U = 0,\ u = 1;$
- (b) $R^+ = max\{R, c\} = R + max\{0, c - R\} = [-R + c]_+ + R;$
- (c) $R^+ = max\{R, \alpha R + \beta\} = R + max\{0, (\alpha - 1)R + \beta\} = [(\alpha - 1)R + \beta]_+ + R;$

Initialization of the induction

for $m = 0$, we start with the identity function and sequentially build [2]

$$\begin{aligned} R^{+} &= \max\{x, a_0\} = x + [a_0 - x]_{+} (\text{Cutting off } x \leq a_0); \\ R^{++} &= R^{+} - a_0 (\text{Shifting}); \\ R_0 &= R^{++} - \frac{(\|h\|_\infty + \delta)}{\delta} [R^{++}]_{+} \end{aligned} \tag{8}$$

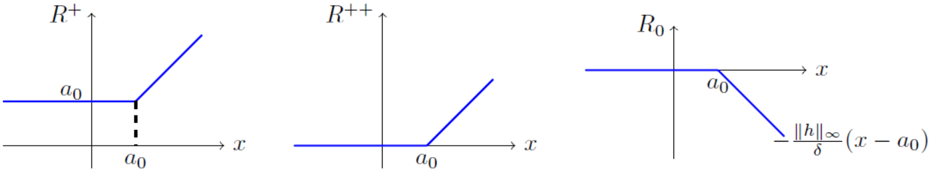


Figure 11: An illustration of constructing the initial function R_0.

[2] $\|h\|_\infty = \max_{k=1,\ldots,M} |h_k|$ is the infinity norm.

A trapezoid function

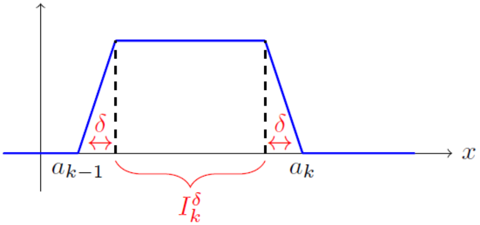


Figure 3: A trapezoid function, which is a continuous approximation of the indicator function. The parameter δ measures the quality of the approximation.

A trapezoid function is constant on the segment $I_k^\delta = [a_{k-1} + \delta, a_k - \delta]$ and linear in the δ-tolerant region $I_k \setminus I_k^\delta$.

Induction from R_m to R_{m+1}

Given R_m, we stack three modules of one-neuron residual blocks on top of it to build R_{m+1}. More precisely, we use R_m as input and sequentially perform

$$
\begin{aligned}
&\text{(a) } R_m^{+} = \max\{R_m, -(1+\frac{1}{m+1})R_m\};\\
&\text{(b) } R_m^{++} = \min\{R_m^{+}, -R_m^{+} + \frac{(m+2)\|h\|_\infty}{\delta}(a_{m+1}-a_m)\};\\
&\text{(c) } R_{m+1} = \min\{R_m^{++}, (m+2)\|h\|_\infty\};
\end{aligned}
\tag{9}
$$

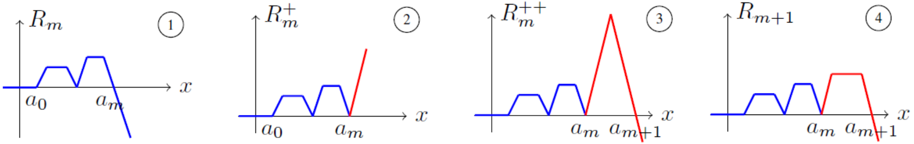


Figure 5: The construction of R_{m+1} based on R_m. We build the next trapezoid function (red) and keep the previous ones (blue) unchanged.

A increasing trapezoid function

With this sequential construction, we can build a increasing trapezoid function as shown in Figure 4.

$$R_{m+1} = \begin{cases} R_m, & \text{if } x < a_m, \\ \frac{(m+2)\|h\|_\infty}{\delta}(x - a_m), & \text{if } x \in [a_m, a_m + \delta), \\ (m+2)\|h\|_\infty, & \text{if } x \in [a_m + \delta, a_{m+1} - \delta) = I^\delta_{m+1}, \\ -\frac{(m+2)\|h\|_\infty}{\delta}(x - a_{m+1}), & \text{if } x \in [a_{m+1} - \delta, +\infty). \end{cases} \tag{10}$$

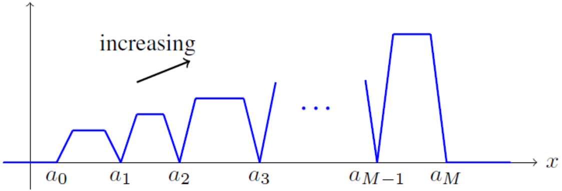


Figure 4: An increasing trapezoid function, which is a special case of grid indicator function when $d = 1$, is trapezoidal on each subdivision with increasing constant value from left to right.

Adjusting function values

Before Adjusting, we remark that $R_M \to -\infty$ as $x \to \infty$.

$$R_M = -\frac{(m+1)\|h\|_\infty}{\delta}(x - a_m), \text{ if } x \in [a_m - \delta, +\infty). \tag{11}$$

This negative tail can be easily removed by a max operator:

$$R_M^* = \max\{R_M, 0\}, \tag{12}$$

Adjusting function values

For any $k = M, \ldots, 1$, we sequentially construct R^*_{k-1} with

$$R^*_{k-1} = R^*_k + \frac{h_k - (k+1)\|h\|_\infty}{\|h\|_\infty}[R^*_k - k\|h\|_\infty]_+ \tag{13}$$

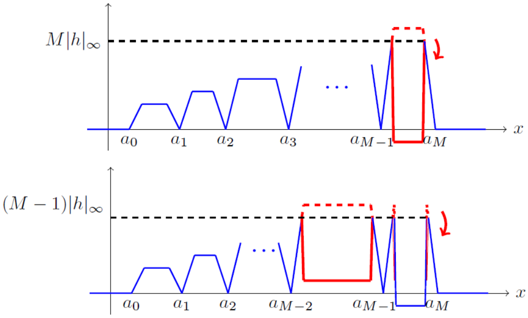


Figure 6: An illustration of the function adjustment procedure applied to the top level sets. At each step, we adjust one I^δ_k to the desired function value h_k.

Adjusting function values

For any $k = M, \ldots, 0$, we show that R_k^* satisfies

- $R_k^* = 0$ on $(-\infty, a_0]$ and $[a_M, +\infty)$;
- $R_k^* = h_j$ on I_j^δ for any $j = M, \ldots, k+1$;
- $R_k^* = (j+1)\|h\|_\infty$ on I_j^δ for any $j = k, \ldots, 1$;
- R_k^* is bounded with $-\|h\|_\infty \leq R_k^* \leq (k+1)\|h\|_\infty$.

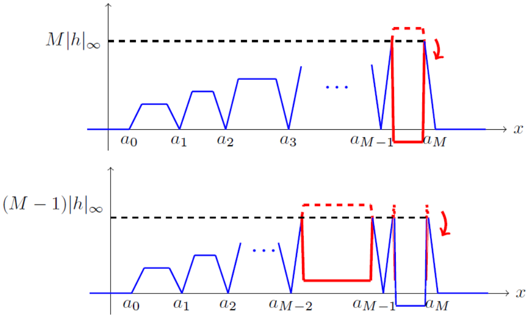


Figure 6: An illustration of the function adjustment procedure applied to the top level sets. At each step, we adjust one I_k^δ to the desired function value h_k.

Adjusting function values

The final R_0^* is the desired approximation of h. More precisely, the R_0^* satisfies

- $R_0^* = 0$ on $(-\infty, a_0]$ and $[a_M, +\infty)$.
- $R_0^* = h_k$ on $I_k^\delta = [a_{k-1} + \delta, a_k - \delta]$ for any $k = 1, \ldots, M$.
- R_0^* is bounded with $-\|h\|_\infty \leq R_0^* \leq \|h\|_\infty$.

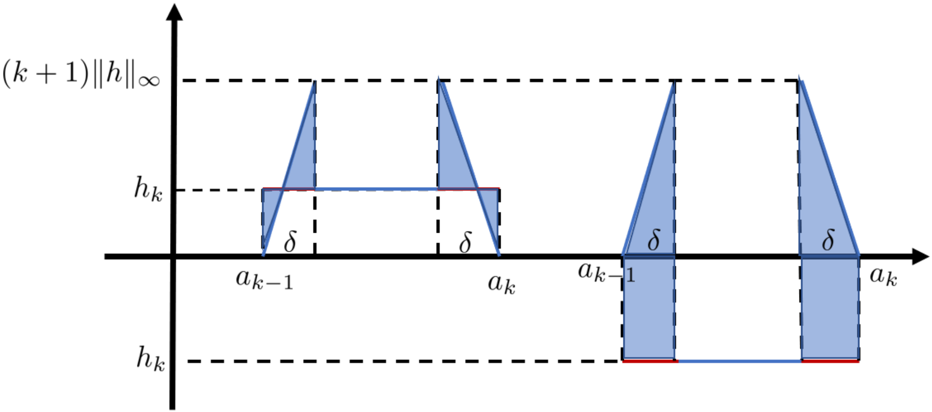
Adjusting function values

As a result, the difference between R_0^* and h can be bounded by

$$\int_{\mathbb{R}} |R_0^*(x) - h(x)| \mathrm{d}x \leq 4M\delta \|h\|_\infty \tag{14}$$

which can be made arbitrarily small by choosing an appropriate δ.

My result



My result

$$
\begin{aligned}
\int_{\mathbb{R}} |R_0^*(x) - h(x)|\mathrm{d}x &\leq \sum_{k=1}^{M}(\frac{1}{2}\delta(k+1)\|h\|_\infty \cdot 2 + \delta h_k \cdot 2) \\
&= \sum_{k=1}^{M}(\delta(k+1)\|h\|_\infty + 2\delta h_k) \\
&\leq \sum_{k=1}^{M}(\delta(k+1)\|h\|_\infty + 2\delta\|h\|_\infty) \\
&= \delta\|h\|_\infty \sum_{k=1}^{M}((k+1)+2) \\
&= \delta\|h\|_\infty \cdot M \cdot \frac{4+(M+3)}{2} \\
&= \frac{M+7}{2} M\delta\|h\|_\infty
\end{aligned} \tag{15}
$$

Outline

Conclusion

- This paper has shown a universal approximation theorem for the ResNet structure with one unit per hidden layer.
- This paper prove the theorem by very interesting constructions and induction step by step.