Mathematical Techniques in the Approximation Theory that are Rooted in Neural Networks - Yarotsky's Theorem

Ko, Suh, Huo

Georgia Tech

Summer of 2020

Class of Functions to be approximated

- Paper considers the Sobolev spaces $\mathcal{W}^{n,\infty}([0,1]^d)$ with $n = 1, 2, \ldots$. Recall that $\mathcal{W}^{n,\infty}([0,1]^d)$ is defined as the space of functions on $[0,1]^d$ lying in L^∞ along with their weak derivatives up to order n.
- The norm in $\mathcal{W}^{n,\infty}([0,1]^d)$ can be defined as :

$$\|f\|_{\mathcal{W}^{n,\infty}([0,1]^d)} = \max_{\mathbf{n}:|\mathbf{n}|\leq n} \operatorname*{ess\,sup}_{x\in[0,1]^d} |D^{\mathbf{n}} f(x)|,$$

  where boldface character $\mathbf{n}$ denotes $\mathbf{n} = \{n_1, \ldots, n_d\}$.
- We denote $F_{n,d}$ the unit ball in $\mathcal{W}^{n,\infty}([0,1]^d)$:

$$F_{n,d} = \{f \in \mathcal{W}^{n,\infty}([0,1]^d) : \|f\|_{\mathcal{W}^{n,\infty}([0,1]^d)} \leq 1\}.$$

Statement of Theorem 1.

(Theorem 1.) For any d, n and $\varepsilon \in (0,1)$, there is a ReLU network architecture that

1. is capable of expressing any function from $F_{d,n}$ with error ε;
2. has the depth at most $c(\ln(1/\varepsilon) + 1)$ and at most $c\varepsilon^{-d/n}(\ln(1/\varepsilon) + 1)$ weights and computation units, with some constants $c = c(d, n)$.

Key idea

- Local Taylor Approximation (LTA) : We split the input space into small hyper-cubes and construct a network that approximates a local Taylor expansion on each of these hyper-cubes.
- Approximation of multiplication operator : We need to build networks that for given input (x, y) approximately compute the product xy.

Yarotsky (17) vs Hieber (20)

- Although two key ideas for Theorem 1. are same as those presented in Hieber's paper, there are differences between two papers for some details.
- Following table summarizes those differences :

	Yarotsky (17)	Hieber (20)
Function Class	Sobolev Space	β-hölder
Parameter Bound	Unbounded	Bounded by 1
Partition of Unity	$\prod_{j=1}^{d} \Psi(3Nx_j - 3m_j)$	$\prod_{j=1}^{d}(1 - N\lvert x_j - x_j^{\ell}\rvert)_+$
Product operator	$x^2 \to xy$	$x(1-x) \to xy$

Yarotsky (17) vs Hieber (20)

- Similarly with Hieber's work, in Yarotsky's work, a function $f \in \mathcal{W}^{n,\infty}([0,1]^d)$ is approximated via LTA with partition of unity: $\phi(\mathbf{m}) = \prod_{j=1}^{d} \Psi(3Nx_j - 3m_j)$, for $\mathbf{m} = (m_1, m_2, \ldots, m_d)$.
- Detailed explanations on $\Psi(\cdot)$ is deferred in later Section.
- We denote the approximated function as $f_1(x)$:

$$f_1(x) = \sum_{m \in \{0,1,\ldots,N\}^d} \sum_{\alpha : |\alpha| \leq n-1} \frac{D^\alpha f\left(\frac{\mathbf{m}}{N}\right)}{\alpha!} \underbrace{\phi(\mathbf{m})\left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^\alpha}_{=①} .$$

- The underbraced term ① is the product of at most $d + n - 1$ piece-wise linear univariate factor.
- This term is "directly" approximated through chained applications of product operator $\tilde{X}$ (Prop 3.), eventually leading a different construction of neural network with that of Hieber's.

Yarotsky (17) vs Hieber (20)

- In Hieber's work, he constructs $Hat^d(x_1, x_2, \ldots, x_d)$ for the approximation of partition of unity, $\prod_{j=1}^{d}(1 - N|x_j - x_j^\ell|)_+$ and construct $Mon_{m,\beta}^d(x_1, \ldots, x_d)$ for the approximation of $P_{x_\ell}^\beta(x)/B + 1/2$, respectively. Subsequently, concatenate them into one network.
- Note that the constructions above are due to the assumption that all the parameters are bounded by 1.
- In Yarotsky's work, the unbounded assumption on parameter values allows different construction of product operator, $\tilde{X}$, using squared function x^2. This naturally leads to the direct approximation of the term ①.
- Detailed proof will be provided in following Section.

Proposition 2.

(Proposition 2.) The function $f(x) = x^2$ on the segment $[0, 1]$ can be approximated with any error $\varepsilon > 0$ by a ReLU network having the depth and the number of weights and computation with $\mathcal{O}(\ln(1/\varepsilon))$.

Key idea

1. For any $m \geq 0$, construct a function f_m which is a piece-wise linear interpolation of $f(x) = x^2$ with $2^m + 1$ uniformly distributed breakpoints $\frac{k}{2^m}, k = 0, \ldots, 2^m$.
2. Calculate $f_{m-1} - f_m$ and find out how this can be expressed in terms of "sawtooth" function, which will be detailed later.
3. After obtaining f_m through telescoping sum, calculate the upper-bound for $|f - f_m|_\infty$.
4. Think how f_m can be represented as neural network architecture.

Sawtooth function (Telgarsky, 15)

Consider the “tooth” function (or “mirror”) function $g : [0,1] \to [0,1]$,

$$g(x) = \begin{cases} 2x, & \text{if } x < \frac{1}{2} \\ 2(1-x), & \text{if } x \geq \frac{1}{2}, \end{cases}$$

and iterated functions

$$g_s(x) = \underbrace{g \circ g \circ \cdots \circ g}_{s}(x).$$

Telgarsky has shown that g_s is a “sawtooth” function with 2^{s-1} uniformly distributed “teeth” (each application of g doubles the number of teeth):

$$g_s(x) = \begin{cases} 2^s\left(x - \frac{2k}{2^s}\right), & \text{if } x \in \left[\frac{2k}{2^s}, \frac{2k+1}{2^s}\right], k = 0, 1, \dots, 2^{s-1} - 1, \\ 2^s\left(\frac{2k}{2^s} - x\right), & \text{if } x \in \left[\frac{2k-1}{2^s}, \frac{2k}{2^s}\right], k = 1, 2, \dots, 2^{s-1}. \end{cases}$$

$f_{m-1}(x) - f_m(x)$

For $x \in \left[\frac{2k}{2^m}, \frac{2k+1}{2^m}\right]$, we can construct $f_{m-1}(x)$ and $f_m(x)$ such that

$$f_{m-1}(x) = \frac{4k+2}{2^m}x - \frac{4k^2+4k}{2^{2m}},$$
$$f_m(x) = \frac{4k+1}{2^m}x - \frac{4k^2+2k}{2^{2m}}.$$

Then, we know the difference of two terms is

$$f_{m-1}(x) - f_m(x) = \frac{2^m\left(x - \frac{2k}{2^m}\right)}{2^{2m}}.$$

$f_{m-1}(x) - f_m(x)$ (continue.)

From the previous slide, we know that refining the interpolation from f_{m-1} to f_m amounts to adjusting it by a function proportional to a "saw-tooth" function:

$$f_{m-1}(x) - f_m(x) = \frac{g_m(x)}{2^{2m}}.$$

Through a telescoping sum, for $m \geq 0$,

$$f_m(x) = x - \sum_{s=1}^{m} \frac{g_s(x)}{2^{2s}}.$$

A bound on $|f - f_m|_\infty$

For $\forall x \in [0, 1]$, we can obtain a bound for $|x^2 - f_m(x)|$: First, we can obtain f_m for arbitrary $k \in \{0, 1, \ldots, 2^m - 1\}$,

$$f_m(x) = \frac{2k+1}{2^m}x - \frac{k^2+k}{2^{2m}}.$$

Since $f_m(x) \geq x^2$ for $x \in [0, 1]$,

$$\begin{aligned}
|f(x) - f_m(x)| &= \frac{2k+1}{2^m}x - \frac{k^2+k}{2^{2m}} - x^2 \\
&= -\left(x - \frac{k+\frac{1}{2}}{2^m}\right)^2 + 2^{-2m-2} \\
&\leq 2^{-2m-2}.
\end{aligned}$$

Neural network architecture of $f_m(x)$

Note that g can be implemented via finite ReLU network :

$$g(x) = 2\sigma(x) - 4\sigma(x - \frac{1}{2}) + 2\sigma(x - 1).$$

Then g_m can be constructed through ReLU neural network with

- Computation Units : $3m$,
- Hidden Layers (depth) : m,
- Number of nonzero weights : $9(m-1)+6$.

f_m only involves $\mathcal{O}(m)$ linear operations and compositions of g, we can implement f_m by a ReLU network having depth and the number of weights and computation units all being $\mathcal{O}(m)$. Using an identity $\varepsilon = 2^{-2m-2}$ yields the claim.

Neural network architecture of $f_m(x)$

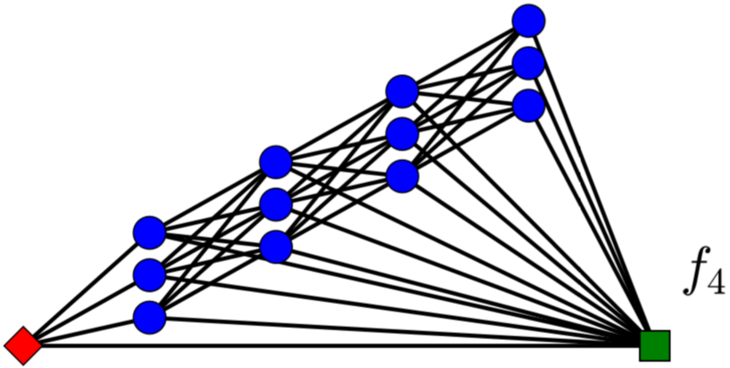

Figure 1: Realization of f_4. A feedforward neural network having 1 input unit (diamond), 1 output unit (square), and $4 \times 3 = 12$ units with ReLU activation (circles).

Proposition 3.

(Proposition 3.) Given $M > 0$ and $\varepsilon \in (0, 1)$, there is a ReLU network η with two input units that implements a function $\tilde{x} : \mathbb{R}^2 \to \mathbb{R}$ so that

- for any inputs x, y, if $|x| \leq M$ and $|y| \leq M$, then $|\tilde{X}(x, y) - xy| \leq \varepsilon$;
- if $x = 0$ or $y = 0$, the $\tilde{X}(x, y) = 0$;
- the depth and the number of weights and computation units in η is not greater than $c_1 \ln(1/\varepsilon) + c_2$ with an absolute constant c_1 and a constant $c_2 = c(M)$.

Key idea

1. Use a polarization identity :

$$xy = \frac{1}{2}\left(\underbrace{(x+y)^2} - \underbrace{x^2} - \underbrace{y^2}\right)$$

2. Approximate three underbraced terms with f_m in Proposition 2.

Proof of Proposition 3.

- Let $\tilde{f}_{\mathrm{sq},\delta}$ be the approximate squaring function from Proposition 2 such that $\tilde{f}_{\mathrm{sq},\delta}(0) = 0$ and $|\tilde{f}_{\mathrm{sq},\delta}(x) - x^2| < \delta$ for $x \in [0, 1]$.
- WLOG set $M \geq 1$, for $|x|, |y| \leq M$, we have $|x + y| \leq 2M$. Using polarization identity, we can construct $\tilde{X}(x, y)$ as follows:

$$\tilde{X}(x, y) = 2M^2 \left(\tilde{f}_{\mathrm{sq},\delta}\left(\frac{|x + y|^2}{2M} \right) - \tilde{f}_{\mathrm{sq},\delta}\left(\frac{|x|^2}{2M} \right) - \tilde{f}_{\mathrm{sq},\delta}\left(\frac{|y|^2}{2M} \right) \right)$$

- By setting $\delta = \frac{\varepsilon}{6M^2}$, we can get the error bound with ε for any $\varepsilon \in [0, 1]$, $|\tilde{X}(x, y) - xy| \leq \varepsilon$.
- Construction of $\tilde{X}$ only involves with three instances of $\tilde{f}_{\mathrm{sq},\delta}$ and finitely many linear and ReLU operations,
- Using Proposition 2, we can implement $\tilde{X}$ by a ReLU network such that its depth and the number of computation units and weights $\mathcal{O}(\ln(1/\delta))$, which is $\mathcal{O}(\ln(1/\varepsilon) + \ln M)$.

Proof of Theorem 1.

First Step

- Neural Network $\tilde{f}$ is not directly used to approximate $f \in \mathcal{W}^{n,\infty}([0,1]^d)$, instead it is used to approximate the approximated f through local Taylor expansion, where the paper denotes it as $f_1(X)$.
- For $X \in [0,1]^d$, the closeness between functions is measured in a L^∞ sense. Approximation error can be decomposed with the help of triangular inequality as follows:

$$\left\|\tilde{f} - f\right\|_{L^\infty[0,1]^d} \leq \underbrace{\|f_1(X) - f(X)\|_{L^\infty[0,1]^d}}_{①} + \underbrace{\left\|\tilde{f}(x) - f_1(X)\right\|_{L^\infty[0,1]^d}}_{②}.$$

- We want to control both terms ① and ② less than or equal to $\frac{\varepsilon}{2}$ respectively. In the first setp, we will focus on controlling ①.

Control on ①

- Recall that a function $f \in \mathcal{W}^{n,\infty}([0,1]^d)$ is approximated via LTA with partition of unity formed by a grid of $(N+1)^d$ functions $\phi(m)$ for positive integer N:

$$\sum_m \underbrace{\prod_{j=1}^d \Psi(3N(x_j - m_j/N))}_{=\phi(m)} = 1, x \in [0,1]^d,$$

  where $m \in \{0, 1, \ldots, N\}^d$.
- $\Psi(x)$ is the univariate trapezoid function : $\Psi(x) = \sigma(x+2) - \sigma(x+1) - \sigma(x-1) + \sigma(x-2)$ supported on $[-2, 2]$, equal to 1 on $[-1, 1]$, and linear on $[-2, -1] \cup [1, 2]$.

Control on ①

- The approximated function via Local Taylor Approximation, we can construct f_1 as follows:

$$f_1(x) = \sum_{\mathbf{m}\in\{0,1,\ldots,N\}^d} \sum_{\alpha:|\alpha|\le n-1} \frac{D^\alpha f\left(\frac{\mathbf{m}}{N}\right)}{\alpha!} \phi(\mathbf{m}) \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^\alpha$$

- We denote the degree-$(n-1)$ Taylor Polynomial for the function f at $x = \frac{m}{N}$ as P_m, and rewrite $f_1(x)$ as follows :

$$\begin{aligned} f_1(x) &= \sum_{\mathbf{m}\in\{0,1,\ldots,N\}^d} \sum_{\alpha:|\alpha|\le n-1} \frac{D^\alpha f\left(\frac{\mathbf{m}}{N}\right)}{\alpha!} \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^\alpha \phi(\mathbf{m}) \\ &= \sum_{\mathbf{m}\in\{0,1,\ldots,N\}^d} P_m(x)\phi(\mathbf{m}). \end{aligned}$$

Control on ①

Observe $f(x) \in \mathcal{W}^{n,\infty}([0,1]^d)$ can be written as follows by Multivariate Taylor's Theorem: for any $\xi \in [0,1]$ and any $a \in [0,1]^d$,

$$f(x) = \sum_{\alpha:|\alpha|\leq n-1} D^\alpha f(a) \frac{(x-a)^\alpha}{\alpha!} + \sum_{\alpha:|\alpha|=n} D^\alpha f(a+\xi(x-a)) \frac{(x-a)^\alpha}{\alpha!}.$$

Let $a = \frac{m}{N}$,

$$|f(x) - f_1(x)| = \left| \sum_m \phi_m(x) \big(f(x) - P_m(x)\big) \right|.$$

In the above identity, we use the fact $\sum_m \phi_m(x) = 1$.

Control on ①

$$\begin{aligned}\left|\sum_m \phi_m(x)\big(f(x)-P_m(x)\big)\right| &\leq \sum_{m:|x_k-\frac{m_k}{N}|<\frac{1}{N}\forall k} |f(x)-P_m(x)| \\ &\leq 2^d \max_{m:|x_k-\frac{m_k}{N}|<\frac{1}{N}\forall k} |f(x)-P_m(x)| \\ &\leq \frac{2^d d^n}{n!}\left(\frac{1}{N}\right)^n \max_{\mathbf{n}:|\mathbf{n}|=n} \operatorname*{ess\,sup}_{x\in[0,1]^d} |D^{\mathbf{n}} f(x)| \\ &\leq \frac{2^d d^n}{n!}\left(\frac{1}{N}\right)^n.\end{aligned}$$

In the first inequality, we use the support condition of $\phi(m)$ and the fact $\|\phi_m\|_\infty = 1$. In the second inequality, the fact that any $x \in [0,1]^d$ belongs to the support of at most 2^d functions ϕ_m is used. In the third, the standard bound for Taylor remainder and in the last inequality, the definition of $\mathcal{W}^{n,\infty}([0,1]^d)$ are used.

Control on ①

It follows that if we choose

$$N = \lceil \left(\frac{n!}{2^d d^n} \frac{\varepsilon}{2} \right)^{-1/n} \rceil,$$

then the error follows

$$\|f - f_1\|_\infty = \left| \sum_m \phi_m(x) \big(f(x) - P_m(x) \big) \right| \leq \frac{2^d d^n}{n!} \left(\frac{1}{N} \right)^n \leq \frac{\varepsilon}{2}.$$