Lecture 11: DNN Approximation - Bit Extraction Method

Haizhao Yang
Department of Mathematics
University of Maryland College Park

2022 Summer Mini Course
Tianyuan Mathematical Center in Central China

Topic 2: Results by Bit Extraction

Continuous functions

Ideas: local polynomial approximation $+$ bit extraction

Theorem (Yarotsky, 2018)

For any f Lip. cont. on $[0,1]^d$ and a sufficiently large depth L and width N:

- *approximation rate of shallow NNs with $L = O(1)$: $\epsilon = O(N^{-2/d})$;*
- ***tight** rate (VCdim) of deep NNs with $N = O(d)$: $\epsilon = O(L^{-2/d})$.*

Continuous functions

Ideas: local polynomial approximation $+$ bit extraction

Theorem (Yarotsky, 2018)

For any f Lip. cont. on $[0,1]^d$ and a sufficiently large depth L and width N:

- *approximation rate of shallow NNs with $L = O(1)$: $\epsilon = O(N^{-2/d})$;*
- ***tight** rate (VCdim) of deep NNs with $N = O(d)$: $\epsilon = O(L^{-2/d})$.*

Remark: number of parameters for accuracy ϵ:

- One hidden layer: $O(\frac{1}{\epsilon^d})$;
- $O(1)$ width and deep NN: $O(\frac{1}{\epsilon^{d/2}})$.

Continuous functions

Ideas: local polynomial approximation $+$ bit extraction

Theorem (Yarotsky, 2018)

For any f Lip. cont. on $[0,1]^d$ and a sufficiently large depth L and width N:

- *approximation rate of shallow NNs with $L = O(1)$: $\epsilon = O(N^{-2/d})$;*
- ***tight** rate (VCdim) of deep NNs with $N = O(d)$: $\epsilon = O(L^{-2/d})$.*

Remark: number of parameters for accuracy ϵ:

- One hidden layer: $O(\frac{1}{\epsilon^d})$;
- $O(1)$ width and deep NN: $O(\frac{1}{\epsilon^{d/2}})$.
- Explicit error formulas?
- Very deep is computational efficient?

Too deep might not be necessary considering parallel computing efficiency.

#parameter	layer	width	test error	improvement ratio	time
5038	2	69	1.13×10^{-2}	–	3.84×10^{1}
5041	4	40	1.65×10^{-4}	–	3.80×10^{1}
4993	8	26	1.69×10^{-5}	–	5.07×10^{1}
5029	33	12	4.77×10^{-3}	–	1.28×10^{2}
9997	2	98	4.69×10^{-3}	2.41	4.40×10^{1}
10090	4	57	7.69×10^{-5}	2.14	4.67×10^{1}
9954	8	37	7.43×10^{-6}	2.27	5.92×10^{1}
10021	65	12	2.80×10^{-1}	0.02	2.31×10^{2}
19878	2	139	1.43×10^{-3}	3.28	5.18×10^{1}
20170	4	81	2.30×10^{-5}	3.34	6.26×10^{1}
20194	8	53	2.97×10^{-6}	2.50	7.08×10^{1}
20005	129	12	3.17×10^{-1}	0.88	4.30×10^{2}

Figure: FNN to approximate 1D random smooth functions when $m > N^2$.

Approximation rate in N and L

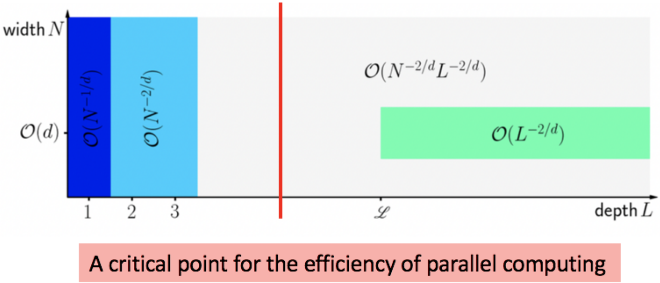

Figure: Color areas: existing work. Grey area: our contribution.

ReLU DNNs, continuous functions $C([0,1]^d)$

ReLU; Fixed width $O(d)$, varying depth L

- Tight error rate $O(L^{-2/d})$ with L^∞-norm
- Yarotsky, 2018

ReLU; Fixed network width $O(N)$ and depth $O(L)$

- Tight error rate $5\omega_f(8\sqrt{d}N^{-2/d}L^{-2/d})$ simultaneously in N and L with L^∞-norm
- ω_f is the modulas of continuity
- Shen, Y., and Zhang (CiCP, 2020)

Curse of dimensionality exists!

ReLU DNNs, smooth functions $C^s([0,1]^d)$

Does smoothness help?

ReLU; Fixed width $O(d)$, varying depth L

- Tight error rate $O(L^{-2s/d})$ with L^∞-norm
- Yarotsky, 2019

ReLU; Fixed network width $O(N)$ and depth $O(L)$

- Tight rate $85(s+1)^d 8^s \|f\|_{C^s([0,1]^d)} N^{-2s/d} L^{-2s/d}$ simultaneously in N and L with L^∞-norm
- Lu, Shen, Y., and Zhang (SIMA, 2021)

The curse of dimensionality exists when s is fixed.

DNNs with advanced activation function

Research methodology

- Previously: fixed one type of NN, vary the target function spaces;
- Now: fixed a generic function space, vary the NN design.

DNNs with advanced activation function

Research methodology

- Previously: fixed one type of NN, vary the target function spaces;
- Now: fixed a generic function space, vary the NN design.

Sine-ReLU; Fixed width $O(d)$, varying depth L

- $\exp(-c_{r,d}\sqrt{L})$ with L^∞-norm for $C^r([0,1]^d)$
- Root exponential convergence achieved
- Curse of dimensionality is not clear
- Yarotsky, 2019

Floor and ReLU activation, width $O(N)$ and depth $O(dL)$, $C([0,1]^d)$

- Error rate $\omega_f(\sqrt{d}N^{-\sqrt{L}}) + 2\omega_f(\sqrt{d})N^{-\sqrt{L}}$ with L^∞-norm
- Merely based on the compositional structure of DNNs
- NO curse of dimensionality for many continuous functions
- Root exponential approximation rate
- Shen, Y., and Zhang (Neural Computation, 2020)

Further interpretation of our result

Explicit error bound

Floor and ReLU activation, width $O(N)$ and depth $O(dL)$, Hölder$([0,1]^d, \alpha, \lambda)$

- Error rate $3\lambda d^{\alpha/2} N^{-\alpha\sqrt{L}}$ with L^∞-norm
- NO curse of dimensionality
- Root exponential approximation rate
- Shen, Y., and Zhang (Neural Computation, 2020)

Further interpretation of our result

Can we get an error bound in terms of the number of parameters $O(W)$?

Floor and ReLU activation, width $O(d)$ and depth $O(dW)$, $C([0,1]^d)$

- Error rate $\omega_f(\sqrt{d}\,2^{-\sqrt{W}}) + 2\omega_f(\sqrt{d})2^{-\sqrt{W}}$ with L^∞-norm
- NO curse of dimensionality for many continuous functions
- Root exponential approximation rate
- Shen, Y., and Zhang (Neural Computation, 2020)

Further interpretation of our result

Does smoothness help? No

Floor and ReLU activation, width $O(N)$ and depth $O(dL)$, $C^s([0,1]^d)$

- Expected error rate $O(\omega_f(\sqrt{d}N^{-s\sqrt{L}}) + 2\omega_f(\sqrt{d})N^{-s\sqrt{L}})$ with a prefactor $O((s+1)^d)$ in the L^∞-norm

Further interpretation of our result

Does the domain $[0,1]^d$ matter? No

Floor and ReLU activation, width $O(N)$ and depth $O(dL)$, $C([-M,M]^d)$

- Error rate $\omega_f^{[-M,M]^d}(2M\sqrt{d}\,N^{-\sqrt{L}}) + 2\omega_f^{[-M,M]^d}(2M\sqrt{d})N^{-\sqrt{L}}$ in the L^∞-norm

Further interpretation of our result

Does ω_f matter? Yes

Floor and ReLU activation, width $O(N)$ and depth $O(dL)$, $C([0,1]^d)$

- Error rate $\omega_f(\sqrt{d}N^{-\sqrt{L}}) + 2\omega_f(\sqrt{d})N^{-\sqrt{L}}$ with L^∞-norm

- $\omega_f(r) = \frac{1}{\ln(1/r)}$

$$3(\sqrt{L}\ln N - \tfrac{1}{2}\ln d)^{-1}$$

- $\omega_f(r) = \frac{1}{\ln^{1/d}(1/r)}$

$$3(\sqrt{L}\ln N - \tfrac{1}{2}\ln d)^{-1/d}$$

- $\omega_f(r) = r^{\alpha/d}$

$$3\lambda d^{\frac{\alpha}{2d}} N^{-\frac{\alpha}{d}\sqrt{L}}$$

DNNs with advanced activation function

Depth is powerful in the previous result. Can width can be as powerful as depth?

Floor, Sign, and 2^x activation, width $O(N)$ and depth 3, $C([0,1]^d)$

- Error rate $\omega_f(\sqrt{d}2^{-N}) + 2\omega_f(\sqrt{d})2^{-N}$ with L^∞-norm
- Merely based on the compositional structure of DNNs
- NO curse of dimensionality for many continuous functions
- Exponential approximation rate
- Shen, Y., and Zhang (Neural Networks, 2021)

Key ideas of our approximation

For $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$:
$$\boldsymbol{x} \to \phi_1(\boldsymbol{x}) = \boldsymbol{\beta} \to \phi_2(\boldsymbol{\beta}) = k_{\boldsymbol{\beta}} \to \phi_3(k_{\boldsymbol{\beta}}) = f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$$

- Piecewise constant approximation: $f(\boldsymbol{x}) \approx f_p(\boldsymbol{x}) \approx \phi_3 \circ \phi_2 \circ \phi_1(\boldsymbol{x})$
- 2^N pieces per dim and 2^{Nd} pieces with accuracy 2^{-N}
- Floor NN $\phi_1(\boldsymbol{x})$ s.t. $\phi_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \mathbb{Z}^d$.
- Linear NN ϕ_2 mapping $\boldsymbol{\beta}$ to an integer $k_{\boldsymbol{\beta}} \in \{1, \ldots, 2^{Nd}\}$
- Key difficulty: NN ϕ_3 of width $O(N)$ and depth $O(1)$ fitting 2^{Nd} samples in 1D with accuracy $O(2^{-N})$
- ReLU NN fails

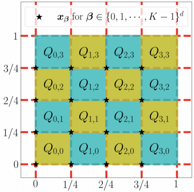

Figure: Uniform domain partitioning.

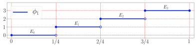

Figure: Floor function.

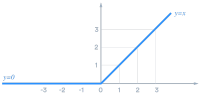

Figure: ReLU function.

Key ideas of our approximation

Binary representation and approximation

$\theta = \sum_{\ell=1}^{\infty} \theta_\ell 2^{-\ell}$ with $\theta_\ell \in \{0,1\}$ is approximated by $\sum_{\ell=1}^{N} \theta_\ell 2^{-\ell}$ with an error 2^{-N}.

Bit extraction via a floor NN of width 2 and depth 1

$$\phi_k(\theta) := \lfloor 2^k \theta \rfloor - 2\lfloor 2^{k-1}\theta \rfloor = \theta_k$$

Bit extraction via a floor NN of width $2N$ and depth 1

Given $\theta = \sum_{\ell=1}^{\infty} \theta_\ell 2^{-\ell}$

$$\phi(\theta) := \begin{pmatrix} \phi_1(\theta) \\ \vdots \\ \phi_N(\theta) \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_N \end{pmatrix} \in \mathbb{Z}^N$$

Key ideas of our approximation

Encoding K numbers to one number

- Extract bits $\{\theta_1^{(k)}, \ldots, \theta_N^{(k)}\}$ from $\theta^{(k)} = \sum_{\ell=1}^{\infty} \theta_\ell^{(k)} 2^{-\ell}$ for $k = 1, \ldots, K$
- sum up to get

$$a = \sum_{\ell=1}^{N} \theta_\ell^{(1)} 2^{-\ell} + \sum_{\ell=N+1}^{2N} \theta_{\ell-N}^{(2)} 2^{-\ell} + \cdots + \sum_{\ell=(K-1)N+1}^{KN} \theta_{\ell-(K-1)N}^{(K)} 2^{-\ell}$$

Decoding one number to get the k-th number

- Extract bits $\{\theta_1^{(k)}, \ldots, \theta_N^{(k)}\}$ from a via

$$\psi(k) := \phi(2^{(k-1)N} a - \lfloor 2^{(k-1)N} a \rfloor).$$

- sum up to get $\theta^{(k)} \approx \sum_{\ell=1}^{N} \theta_\ell^{(k)} 2^{-\ell} = [2^{-1}, \ldots, 2^{-N}]\psi(k) := \gamma(k)$,
- $\gamma(k)$ is an NN of width $O(N)$ and depth $O(1)$.

Key Lemma

There exists an NN γ of width $O(N)$ and depth $O(1)$ that can memorize arbitrary samples $\{(k, \theta^{(k)}\}_{k=1}^{K}$ with a precision 2^{-N}.

Key ideas of our approximation

For $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$:
$$\boldsymbol{x} \to \phi_1(\boldsymbol{x}) = \boldsymbol{\beta} \to \phi_2(\boldsymbol{\beta}) = k_{\boldsymbol{\beta}} \to \phi_3(k_{\boldsymbol{\beta}}) = f(\boldsymbol{x}_{\boldsymbol{\beta}}) \approx f(\boldsymbol{x})$$

- Piecewise constant approximation: $f(\boldsymbol{x}) \approx f_p(\boldsymbol{x}) \approx \phi_3 \circ \phi_2 \circ \phi_1(\boldsymbol{x})$
- 2^N pieces per dim and 2^{Nd} pieces with accuracy 2^{-N}
- Floor NN $\phi_1(\boldsymbol{x})$ s.t. $\phi_1(\boldsymbol{x}) = \boldsymbol{\beta}$ for $\boldsymbol{x} \in Q_{\boldsymbol{\beta}}$ and $\boldsymbol{\beta} \in \mathbb{Z}^d$.
- Linear NN ϕ_2 mapping $\boldsymbol{\beta}$ to an integer $k_{\boldsymbol{\beta}} \in \{1, \ldots, 2^{Nd}\}$
- Key difficulty: NN ϕ_3 of width $O(N)$ and depth $O(1)$ fitting 2^{Nd} samples in 1D with accuracy $O(2^{-N})$
- Key Lemma: There exists an NN γ of width $O(N)$ and depth $O(1)$ that can memorize arbitrary samples $\{(k, \theta^{(k)})\}_{k=1}^{K}$ with a precision 2^{-N}.

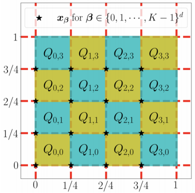

Figure: Uniform domain partitioning.

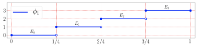

Figure: Floor function.

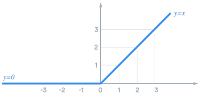

Figure: ReLU function.

Further interpretation of our result

Realistic consideration

- Constructive approximation requires *f* or exponentially many samples given
- Constructed parameters require high precision computation
- Floor and Sign are discontinuous functions leading to gradient vanishing
- The network size has to be increased when $\epsilon \to 0$