Theoretical Deep Learning

Lecture 8: Deep neural nets and depth separation

August 3, 2021

Lecturer: Lei Wu *Scribe: Lei Wu*

A deep fully-connected neural net is given by

$$\begin{aligned} z^0 &= x \\ z^{l+1} &= \sigma(W^{l+1}z^l + b^l), \quad l = 0, 1, \ldots, L-1 \\ f_L(x;\theta) &= z^L, \end{aligned}$$

where $W^l \in \mathbb{R}^{m_l \times m_{l-1}}, b^l \in \mathbb{R}^{m_l}$. Here, L denotes the depth and m_j denotes the width of j-th layer.

A deep residual network (ResNet) is given by

$$\begin{aligned} z^0 &= Vx + b \\ z^{l+1} &= z^l + F(z^l;\theta^l) \\ f_L(x;\theta) &= a^T z^L, \end{aligned}$$

where $V \in \mathbb{R}^{D\times d}, b \in \mathbb{R}^D, a \in \mathbb{R}^D$ and $F(\cdot;\theta^l) : \mathbb{R}^D \mapsto \mathbb{R}^D$ is a small neural net. θ denotes all the learnable parameters.

For these two deep nets, we ask the following questions?

- What is the continuum limit?
- What is the appropriate complexity measures? In other words, can we define associate function norm $\|\cdot\|$ induced by these networks such that the approximation can avoid the CoD?
- Can we show the benefits of depths, i.e., deep nets perform better than shallow nets in some sense.

1 Continuum limits and function norms

Following the Barron space theory, we first need to identify the appropriate continuum limits, which should be in certain "expectation" form such that the approximation ("discretization") is free of CoD. Accordingly, then function norms are defined by the certain "second-order moment" of the corresponding "representations" (certain prob. distributions).

1.1 Fully-connected neural networks

For this case, we are interested in the limit: L fixed and $m \to \infty$. Here, we only consider the case of $L = 3$. For general cases, we refer to [Pham and Nguyen, 2021, Nguyen and Pham, 2020, Araújo et al., 2019, Sirignano and Spiliopoulos, 2021, Wojtowytsch et al., 2020]. It is worth mentioning that all these works have their limitations, and how to define the appropriate function spaces for multilayer fully-connected networks is still an open problem.

Consider the scaled three-layer network:

$$f_m(x;\theta) = \frac{1}{m}\sum_{i=1}^{m} a_i \sigma\left(\frac{1}{m}\sum_{j=1}^{m} b_{i,j}\sigma(w_j^T x)\right). \tag{1.1}$$

What is the limit of $f_m(\cdot;\theta)$ as $m \to \infty$? Let us take a look at the two-layer case,

$$f_m(x;\theta) = \frac{1}{m}\sum_{j=1}^{m} a_j \sigma(w_j^T x) \to \int a\sigma(w^T x)\,\mathrm{d}\rho(a,w).$$

Here, we can think of $(a_1, w_1), \cdots, (a_m, w_m)$ as m independent input-output paths. Hence, a natural choice is to consider parameterization with the distribution over the path space (a, w).

For the three-layer network (1.1), one may want to parameterize the network with a probability distribution over the path space: $(a, b, w) \in \mathbb{R}^{1\times 1\times d}$. Unfortunately, this is impossible since the paths $\{(a_i, b_{i,j}, w_j)\}_{i,j}$ are not independent.

Path-based representations One can think the intermediate layer weight b as a function of a and w, i.e., $b_{i,j} = b(a_i, w_j)$. Notice that $\{a_i, w_j\}$ are independent drawn from $\pi_a \otimes \pi_w \in \mathcal{P}(\mathbb{R}\otimes\mathbb{R}^d)$. Then, the limit becomes

$$f(x;\pi_a,\pi_w,b) = \mathbb{E}_{a\sim\pi_a}\left[a\sigma\left(\mathbb{E}_{w\sim\pi_w}[b(a,w)\sigma(w^T x)]\right)\right]. \tag{1.2}$$

The function norm can be defined as

$$\|f\|_{\mathcal{F}} := \inf_{f=f(\cdot;\pi_a,\pi_w,b)} \mathbb{E}_{a\sim\pi_a, w\sim\pi_w}[a|b(a,w)|\|w\|].$$

Notice that this formulation excludes the case that $b(a, w)$ can be different for the same a and w.

Index particle-based representations Let us take a further look at the path $(a_i, b_{i,j}, w_j)$. Essentially, only the index i and j are independent. We can think of that (i, j) are drawn from certain index space $I \otimes J$. let $\pi_I \otimes \pi_J \in \mathcal{P}(I \otimes J)$. Then, the limit becomes

$$f(x;a,b,w) = \mathbb{E}_{s\sim\pi_I}[a_s\sigma\left(\mathbb{E}_{t\sim\pi_J}[b_{s,t}\sigma(w_t^T x)]\right)], \tag{1.3}$$

where $a \in L^2(\pi_I), b \in L^2(\pi_I\otimes\pi_J), w \in L^2(\pi_J)$. In this formulation, the index space I and J can be arbitrary. In particular, (1.2) can be viewed as a special index particle-based representation, where $I = \mathbb{R}$ and $J = \mathbb{R}^d$. This general index particle-based representations allow us to define the continuum limits of multilayer fully-connected networks [Wojtowytsch et al., 2020].

1.2 Deep ResNets

Let the residual block $F(\cdot;\theta^l)$ be parameterized by a scaled two-layer neural net:

$$F(z;\theta^l) = \frac{1}{m}\sum_{j=1}^{m} a_j^l \sigma(b_j^l \cdot z),$$

where $a_j \in \mathbb{R}^D, b_j \in \mathbb{R}^D$. Then, the update of hidden state becomes

$$z^{l+1} = z^l + \frac{1}{m}\sum_{j=1}^{m} a_j^l \sigma(b_j^l \cdot z^l).$$

Taking $m \to \infty$, we obtain

$$z^{l+1} = z^l + \mathbb{E}_{(a,b)\sim\rho_l}[a\sigma(b \cdot z^l)].$$

To take $L \to \infty$, we need to introduce another explicit scaling factor:

$$z^{l+1} = z^l + \frac{1}{L}\mathbb{E}_{(a,b)\sim\rho_l}[a\sigma(b \cdot z^l)] \tag{1.4}$$

$$\Rightarrow \frac{z^{l+1} - z^l}{L^{-1}} = \mathbb{E}_{(a,b)\sim\rho_l}[a\sigma(b \cdot z^l). \tag{1.5}$$

which can be viewed as a Forward Euler discretization of the following ODE:

$$\frac{\mathrm{d}z(x;t)}{\mathrm{d}t} = \mathbb{E}_{(a,b)\sim\rho_t}[a\sigma(b^T z(x;t))].$$

In a summary, the continuum net is

$$\begin{aligned} z(x;0) &= Vx + b \\ \frac{\mathrm{d}z(x;t)}{\mathrm{d}t} &= \mathbb{E}_{(a,b)\sim\rho_t}[a\sigma(b^T z(x;t))] \\ f(x;\{\rho_t\}) &= a^T z(x;t), \end{aligned} \tag{1.6}$$

where we omit the dependence of a, b, V for simplicity. Now, the primary parameters becomes a sequence of probability measure $\{\rho_t\}_{t\in[0,1]}$.

For more details about how to define the function norms, please refer to [E et al., 2019].

2 Depth separation

In "deep" learning, it is observed that deep nets often perform much better than shallow nets. There exist many reasons behind this benefit of depth. However, we will focus on understanding this issue from the perspective of approximation power. In addition, in this section the activation function is ReLU.

In the following analysis, we once again will heavily use the (shift) triangular function. Let $g(x) = t(2x-1)$ be the shift triangular function and

$$g_l(x) = \underbrace{g \circ g \circ \cdots \circ g}_{l}(x).$$

An illustration of g (i.e, g_1) and g_l is provided in Figure 1. Obviously, g_l has 2^{l+1} linear pieces.

Remark 2.1. Note that one challenge in examining the function composition is the change of input domains. However, the shift triangular function satisfies that $g : [0,1] \mapsto [0,1]$, which dramatically simplifies the analysis of function composition.

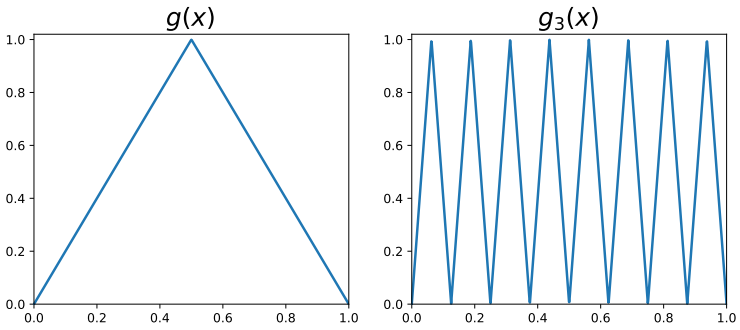

Figure 1: Illustration of the shift triangular function and the triangular wave.

2.1 One-dimensional results

2.1.1 Approximating oscillated functions

Theorem 2.2 ([Telgarsky, 2016])**.** *Consider the target function $f^* = g_l$. Then, g_l can be implemented as a $c_1 l$-layer neural nets with the width less than c_2. For any 2-layer ReLU net $f_m(\cdot;\theta)$ with the width $m = \mathrm{poly}(l)$, we have*

$$\int_0^1 |f_m(x;\theta) - g_l(x)|\,\mathrm{d}x \geq c_3.$$

Here c_1, c_2, c_3 are absolute constants.

Proof. First, $g(x) = \mathrm{ReLU}(2x) + \mathrm{ReLU}(2x-2) - 2\,\mathrm{ReLU}(2x-1)$, i.e., g can be exactly represented as three neurons. Hence, g_l can be represented as $2l$-layer neural net with the width less than or equal to 3.

For a two-layer neural network of width m, let M denote its number of linear pieces. Obviously, $M \sim m$. The proof of the lower bound proceeds by counting triangles as illustrated in Figure 2. Draw the horizontal line $y = 1/2$. Then, there are 2^{l+1} (half) triangles.

$$\begin{aligned}\int_0^1 |f_m(x;\theta) - g_l(x)|\,\mathrm{d}x &\geq [\text{number of surviving triangles}] \cdot [\text{area of the triangle}] \\ &\geq (2^{l+1} - 2M) \cdot (\frac{1}{2} \cdot \frac{1}{2^{l+1}} \cdot \frac{1}{2}) \\ &\geq \frac{1}{4} - \frac{M}{2^{l+2}} \geq \frac{1}{8}. \end{aligned} \tag{2.1}$$

□

2.1.2 Approximating smooth functions

We first study the approximation of $f(x) = x^2$, which will be used later for approximating general smooth functions. We first need the following lemma.

Lemma 2.3. *Let $S_M := (x_k)_{k=0}^M$ be the set of uniform grid points in $[0,1]$ with grid size $h = 1/M$. For any function f, let $P_M f$ be the piecewise linear interpolation of f with the uniform grid points S_M:*

$$P_M f(x) = \sum_{k=1}^{M} f(x_k) t\left(\frac{x - x_k}{h}\right), \tag{2.2}$$

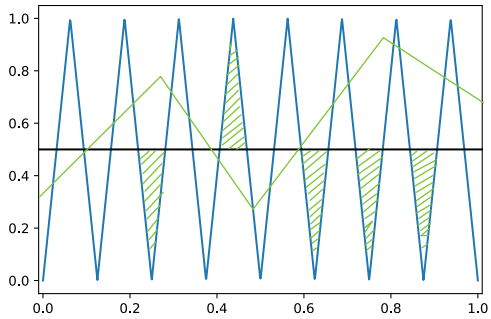

Figure 2: Illustration for the proof of Theorem 2.2

where, $t(\cdot)$ is the triangular function. Then,

$$\sup_{x\in[0,1]} |P_M f(x) - f^*(x)| \lesssim \frac{\sup_{x\in[0,1]} |f''(x)|}{m^2}$$

Proposition 2.4. *For any $\varepsilon > 0$, there exits a neural net $\tilde{f}$, whose depth and width is $O(\log(1/\varepsilon))$ and $O(1)$, respectively, such that*

$$\sup_{x\in[0,1]} |\tilde{f}(x) - x^2| \le \varepsilon.$$

Proof. We can show that $l = 2, 3, \ldots,$

$$P_{2^{l-1}} f^*(x) - P_{2^l} f^*(x) = \frac{g_l(x)}{2^{2l}}, \qquad \forall x \in [0,1]. \tag{2.3}$$

Construct a neural net as follows

$$\begin{aligned} y_0 &= x \\ y_l &= g(y_{l-1}) \\ \tilde{f}(x) &= \sum_{l=1}^{L} \frac{y_l}{2^l}. \end{aligned}$$

Note that g can be implemented using 3 neurons. The last step introduces skip connections from each layer to the output layer. So, the depth and width of this net is $O(L)$ and $O(1)$, respectively. By Lemma 2.3,

$$\sup_{x\in[0,1]} |\tilde{f} - f(x)| = \sup_{x\in[0,1]} |P_{2^L} f(x) - f(x)| \lesssim \frac{1}{4^L}.$$

Taking $1/(4^L) = \varepsilon$, we complete the proof.

□

Lemma 2.5. *Let $\mathcal{G}_m$ denote the set of piecewise linear functions with the number of linear pieces less than or equal to m. Then, for any $g \in \mathcal{G}_m$, we have*

$$\sup_{x\in[0,1]} |g(x) - x^2| \gtrsim \frac{1}{m^2}.$$

Proof. First any interval $[a,b] \subset [0,1]$, we have

$$\begin{aligned} I_{a,b} := \min_{c,d\in\mathbb{R}} \max_{t\in[a,b]} |cx + d - x^2| &= \min_{c,d\in\mathbb{R}} \max_{t\in[0,b-a]} |(x+c)^2 + d| \\ &= \min_{c,d} \max\{(b-a+c)^2 + d, c^2 + d, d\}. \end{aligned}$$

If $c^2 + d \gtrsim (b-a)^2$ or $d \gtrsim (b-a)^2$. Then, $I_{a,b} \gtrsim (b-a)^2$. Otherwise, we must have $c = o(|b-a|), d = o(|b-a|^2)$. This results in the first term satisfies $(b-a+c)^2 + d \gtrsim (|b-a|^2)$. Combining them, we have $I_{b,a} \gtrsim (b-a)^2$.

Since $g \in \mathcal{G}_m$, the number of piecewise linear parts of g is at most m. Hence, for any $g \in \mathcal{G}$, there must exist a domain $[a,b]$ such that (1) there exist c, d such that $g(x) = cx + d$ for $x \in [a,b]$; (2) $|b-a| \gtrsim 1/m$. Then,

$$\sup_{x\in[0,1]} |g(x) - x^2| \geq \sup_{x\in[a,b]} |cx + d - x^2| \gtrsim |b-a|^2 \gtrsim \frac{1}{m^2}.$$

□

Remark 2.6. For any fixed depth $L \geq 1$ and width $m \geq 1$, L-layer ReLU networks can only represent still piecewise linear functions. Moreover, the number of linear pieces is not greater than $(m/L)^L$ (The case $L = 1, 2$ is trivial. The proof of the general $L > 0$ can be found in Section 6 of [Telgarsky's note]). Hence, for a target accuracy ε, the width needs to satisfies $\frac{1}{(m/L)^{2L}} \leq \varepsilon$, which yields $m \geq L\varepsilon^{-1/(2L)}$. Hence,

$$\text{Total parameters} \gtrsim m^2 L \gtrsim L^3 \varepsilon^{-1/L} = \text{poly}(1/\varepsilon).$$

Comparison of Proposition 2.4 and Lemma 2.5 provides a interesting separation result for this specific target function. Specifically, for a given target accuracy ε, deep ReLU network only need $O(\log(1/\varepsilon))$ parameters, while shallow ReLU network needs at least $O(\text{poly}(1/\varepsilon))$ parameters.

Why is approximating x^2 interesting? From the approximation of $f(x) = x^2$, we can get many other results.

- Fast approximation of the *multiplication* $(x,y) \mapsto xy$ using
$$xy = \frac{(x+y)^2 - x^2 - y^2}{2}.$$
- Fast approximation of any monomials: x^k.
- Fast approximation of polynomials: $a_0 + a_1 x + \cdots + a_k x^k$.
- Fast approximation of functions that can be efficiently approximated by polynomials, e.g., Sobolev space.

Theorem 2.7 ([Yarotsky, 2017])**.** *Assume that $f \in C^r([0,1]^d)$ and $\max_{|\alpha|\leq r} \operatorname{ess\,sup}_{x\in[0,1]} |D^\alpha f(x)| \leq 1$. Then, there exists a neural net $\tilde{f}$ of depth at most $C(\log(1/\varepsilon) + 1)$ and width at most $C\varepsilon^{-d/r}(\log(1/\varepsilon) + 1)$ such that*

$$\sup_{x\in[0,1]^d} |\tilde{f}(x) - f(x)| \leq \varepsilon.$$

Here, the constant C depends on d, r.

Remark 2.8. The preceding result only separate deep and shallow nets for the *non-smooth* ReLU activation function. If considering smooth activation function, such as, Tanh, we may do not have this separation. In fact, [Maiorov and Meir, 2000, Mhaskar, 1996] shows that for the Tanh activation function, depth-3 nets can achieve the same approximation rate as Theorem 2.7 (up to logarithmic terms). Can we achieve the same rate for two-layer nets with some smooth activation functions?

2.2 High-dimensional results

The preceding separation results only apply to the one-dimensional case, since the extension to the high one-dimensional case always suffer from CoD for both shallow and deep nets. [Eldan and Shamir, 2016] provides some examples working for the high-dimensional setting.

Theorem 2.9. *Suppose* $|\sigma(x)| \lesssim (1+|x|^\alpha)$ *for all* $x \in \mathbb{R}$ *and some constants* $\alpha > 0$. *Then, for* $d \gtrsim 1$, *there exists* $\mu \in \mathcal{P}(\mathbb{R}^d)$ *and a radial function* $g(x) = g_0(\|x\|)$ *such that*

- $g_0(\|x\|)$ *can be approximated with* 3*-layer neural network with* $\mathrm{poly}(1/\varepsilon, d)$ *parameters.*
- *For any two-layer net of width at most* $\exp(d)$*,*

$$\int |g_0(\|x\|) - f(x)|^2 \, \mathrm{d}\mu(x) \gtrsim 1.$$

Remark.

- The proof is rather intricate, which heavily utilizes the property of Fourier transform of two-layer neural nets. Moreover, this result is also unsatisfying in the sense that μ is not explicit.
- [Daniely, 2017] provides a more explicit construction: $f : \mathbb{S}^{d-1} \otimes \mathbb{S}^{d-1} \mapsto \mathbb{R}$,

  $$f(x, x') = \sin(d^3 \langle x, x' \rangle),$$

  and the error is measure with respect to $\mu = \mathrm{Unif}(\mathbb{S}^{d-1} \otimes \mathbb{S}^{d-1})$. However, this result needs to restrict the parameter magnitudes of two-layer net is not larger than 2^d.

References

[Araújo et al., 2019] Araújo, D., Oliveira, R. I., and Yukimura, D. (2019). A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193*.

[Daniely, 2017] Daniely, A. (2017). Depth separation for neural networks. In *Conference on Learning Theory*, pages 690–696. PMLR.

[E et al., 2019] E, W., Ma, C., and Wu, L. (2019). Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*.

[Eldan and Shamir, 2016] Eldan, R. and Shamir, O. (2016). The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940.

[Maiorov and Meir, 2000] Maiorov, V. and Meir, R. (2000). On the near optimality of the stochastic approximation of smooth functions by neural networks. *Advances in Computational Mathematics*, 13(1):79–103.

[Mhaskar, 1996] Mhaskar, H. N. (1996). Neural networks for optimal approximation of smooth and analytic functions. *Neural computation*, 8(1):164–177.

[Nguyen and Pham, 2020] Nguyen, P.-M. and Pham, H. T. (2020). A rigorous framework for the mean field limit of multilayer neural networks. *arXiv preprint arXiv:2001.11443*.

[Pham and Nguyen, 2021] Pham, H. T. and Nguyen, P.-M. (2021). Global convergence of three-layer neural networks in the mean field regime. *arXiv preprint arXiv:2105.05228*.

[Sirignano and Spiliopoulos, 2021] Sirignano, J. and Spiliopoulos, K. (2021). Mean field analysis of deep neural networks. *Mathematics of Operations Research*.

[Telgarsky, 2016] Telgarsky, M. (2016). Benefits of depth in neural networks. In *Conference on learning theory*, pages 1517–1539. PMLR.

[Wojtowytsch et al., 2020] Wojtowytsch, S. et al. (2020). On the banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *arXiv preprint arXiv:2007.15623*.

[Yarotsky, 2017] Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.