Research Statement

Shijun Zhang[*]

November 17, 2021

I am a Research Fellow at National University of Singapore with an interest in contributing to a deeper understanding of deep learning. Deep neural networks make up the backbone of deep learning algorithms and they have made significant impacts on many applications in science, engineering, technology, and industries, especially for large-scale and high-dimensional learning problems. The great advantages of neural-network-based methods have been demonstrated over traditional learning methods in real applications from many empirical points of view. Understanding the approximation power of deep neural networks theoretically has become a key question for revealing the power of deep learning. The majority of my current research focuses on the approximation theory of deep neural networks.

Current work

Most of my papers are collaborated with my Ph.D. supervisors: Professor Zuowei Shen and Haizhao Yang. Our ultimate goal is to explicitly formulate the full error analysis of deep neural networks, including the approximation error (AE), the generalization error (GE), and the optimization error (OE). See Figure 1 for an illustration and this note for a detailed discussion. Our current goal is to estimate an explicit upper bound for the approximation error of deep neural networks in terms of the width and depth. Given a target function space $\mathcal{F}$ and a hypothesis function space $\mathcal{H}$, the best approximation error $\mathscr{E}(\mathcal{F},\mathcal{H})$ is defined by

$$\mathscr{E}(\mathcal{F},\mathcal{H}) := \sup_{f\in\mathcal{F}} \inf_{h\in\mathcal{H}} \|h-f\|_{L^\infty([0,1]^d)}. \tag{1}$$

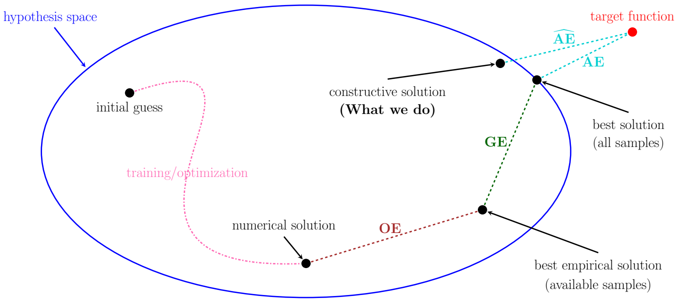

Figure 1: An illustration of AE, GE, and OE. $\widehat{\text{AE}}$ is an upper bound of AE.

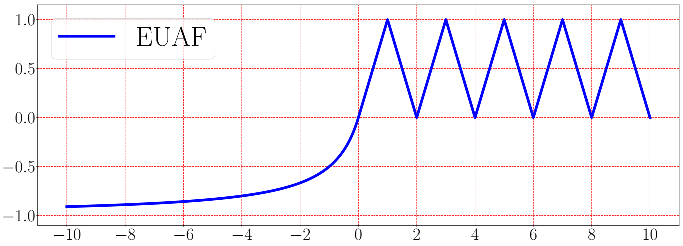

Figure 2: An illustration of our new activation function (EUAF) on $[-10,10]$.

Currently, we only focus on fully-connected neural networks. First, we consider the approximation of ReLU networks. Let $\mathcal{F}$ be the space of 1-Lipschitz functions on $[0,1]^d$;[1] let $\mathcal{H}$ be the space of all functions generated by ReLU networks of width N and depth L. We prove an optimal upper bound of $\mathscr{E}(\mathcal{F},\mathcal{H})$: $\mathscr{E}(\mathcal{F},\mathcal{H}) \le C_d(N^2L^2\ln N)^{-1/d}$, where C_d is a positive constant determined by d and it has a close form in the paper.

The optimality of the upper bound above follows a natural question: Can we further improve the approximation error by adding more conditions? The answer is positive. Based on the definition in (1), there are two natural methods to improve the approximation error: choosing a small $\mathcal{F}$ or designing a bigger $\mathcal{H}$.

[*] Department of Mathematics, National University of Singapore (zhangshijun@u.nus.edu).

[1] In our papers, we consider the generic continuous functions and characterize the approximation error in terms of the modulus of continuity.

For the first method, we consider the space of polynomials and smooth functions since they are two important function spaces in the classical approximation theory. The details can be found below.

⟨1.1⟩ **Polynomials**: Let $\mathcal{F}$ be the space of all polynomials on $[0,1]^d$ of degree $\leq k$ with all coefficients bounded by 1; let $\mathcal{H}$ be the space of all functions generated by ReLU networks of width N and depth L. Then, we prove an exponentially small upper bound of $\mathscr{E}(\mathcal{F},\mathcal{H})$: $\mathscr{E}(\mathcal{F},\mathcal{H}) \leq C_{d,k}N^{-C_k L}$, where $C_{d,k}$ and C_k are two positive constants determined by k, d and k, respectively. They are explicitly formulated in the paper.

⟨1.2⟩ **Smooth functions**: Let $\mathcal{F}$ be the unit ball of $C^s([0,1]^d)$; let $\mathcal{H}$ be the space of all functions generated by ReLU networks of width N and depth L. Then, we prove a nearly optimal upper bound of $\mathscr{E}(\mathcal{F},\mathcal{H})$: $\mathscr{E}(\mathcal{F},\mathcal{H}) \leq C_{s,d}\big(\frac{NL}{\ln N \ln L}\big)^{-2s/d}$, where $C_{s,d}$ is a positive constant determined by s, d and it is explicitly formulated in the paper.

For the second method, we shall design bigger hypothesis spaces. To this end, we introduce and construct new activation functions that are simple, computable, and efficient. They are discussed in detail below.

⟨2.1⟩ ReLU is a piecewise linear function with only two pieces, which limits the approximation power. To break such a limit, we shall consider an activation function with much more pieces. Floor ($\lfloor\cdot\rfloor$) is a simple function with such a property. Thus, we introduce a new type of network built with each hidden neuron activated by either Floor or ReLU. We call such networks **Floor-ReLU networks**. Let $\mathcal{F}$ be the space of 1-Lipschitz functions on $[0,1]^d$; let $\mathcal{H}$ be the space of all functions generated by Floor-ReLU networks of width N and depth L. We prove a root-exponentially small upper bound of $\mathscr{E}(\mathcal{F},\mathcal{H})$: $\mathscr{E}(\mathcal{F},\mathcal{H}) \leq C_{d,1}N^{-C_{d,2}\sqrt{L}}$, where $C_{d,1}$ and $C_{d,2}$ are two positive constants determined by d and they have closed forms in the paper.

⟨2.2⟩ The Floor-ReLU network follows two natural questions: 1) Can we use a continuous activation function to replace Floor? 2) Can we further improve the root-exponentially small approximation error by introducing new activation functions? To solve these two questions, we introduce a new activation function called EUAF, which are constructed via two key points:

- On $[0,\infty)$, let EUAF be a triangle-wave function, which can be regarded as a continuous variant of $x - \lfloor x \rfloor$. This plays a similar role as that of Floor in Floor-ReLU networks.
- On $(-\infty,0)$, let EUAF be the softsign function $\frac{x}{1+|x|}$. Its non-linearity plays a key role in our proof.

See Figure 2 for an illustration of EUAF. The networks activated by EUAF are called **EUAF networks**. Let $\mathcal{F} = C([0,1]^d)$; let $\mathcal{H}$ be the space of all functions generated by EUAF networks of width $36d(2d+1)$ and depth 11. Then, for any $\varepsilon > 0$, we can find $h \in \mathcal{H}$ to approximate any function $f \in \mathcal{F}$ within an error ε; i.e., $\mathscr{E}(\mathcal{F},\mathcal{H}) = 0$. In other words, $\mathcal{H}$ is dense in $\mathcal{F}$. In our construction, as ε goes to 0, the maximum parameter of the network realizing h tends to ∞. In the paper, we also discuss how to construct smooth and sigmoidal variants of EUAF.

All the results discussed above are summarized in Table 1. We have some other results on the approximation theory of deep neural networks. Most of them are adapted or developed from the discussed ones. We will not discuss them due to the space limitation.

Table 1: A summary of all discussed results.

function space	activaiton	width	depth	approximation error	remark
Lipschitz	ReLU	N	L	$C_d(N^2L^2\ln N)^{-1/d}$	optimal
polynomial (degree $\leq k$)	ReLU	N	L	$C_{d,k}N^{-C_k L}$	exponential
$C^s([0,1]^d)$	ReLU	N	L	$C_{s,d}\big(\frac{NL}{\ln N\ln L}\big)^{-2s/d}$	nearly optimal
Lipschitz	Floor/ReLU	N	L	$C_{d,1}N^{-C_{d,2}\sqrt{L}}$	root-exponential
$C([0,1]^d)$	EUAF	$36d(2d+1)$	11	arbitrarily small	

Future work

At this point in my career, my primary interest is in contributing to a deeper understanding of deep learning. My current work focuses on proving explicit approximation upper bounds for various neural networks. To better understand deep learning methodology, I would like to explore several interesting subjects of future work listed below.

- **Application**: I will try to apply our theories to real applications like the classification problem. For example, in some applications on ReLU networks, better results may be attained via replacing ReLU by Floor, EUAF, or their variants.

- **Optimization**: To gain a better optimization error, we shall (partially) solve the following three problems:
  - How do we design an optimization algorithm to numerically compute the best empirical solution (also called the empirical risk minimizer) based on the observed (training) samples?
  - Is the optimization algorithm convergent?
  - If the algorithm is convergent, can we estimate its convergence rate?

- **Generalization**: Why does the trained model work well on similar but unobserved (test) samples? To solve this question, we need to control the generalization error. Thus, it is of great interest and importance to characterize the generalization error explicitly for various neural networks.

In the immediate future, I would like to continue the study of the approximation error and start analyzing the generalization error theoretically for various neural networks. In the long term, I hope to deeply study all the interesting subjects listed above and some other related ones.