Introductions of Approximation Optimization and Generalization Errors

Shijun Zhang*

January 4, 2021

In this note, we introduce approximation, optimization, and generalization errors in order to measure the discrepancy between the target function and the final network attained by a numerical training or optimization method.

Let $\phi(\boldsymbol{x};\boldsymbol{\theta})$ denote a function computed by a (fully-connected) network with $\boldsymbol{\theta}$ as the set of parameters. See Figure 1 for an example of a σ-activated network with width 5 and depth 2.

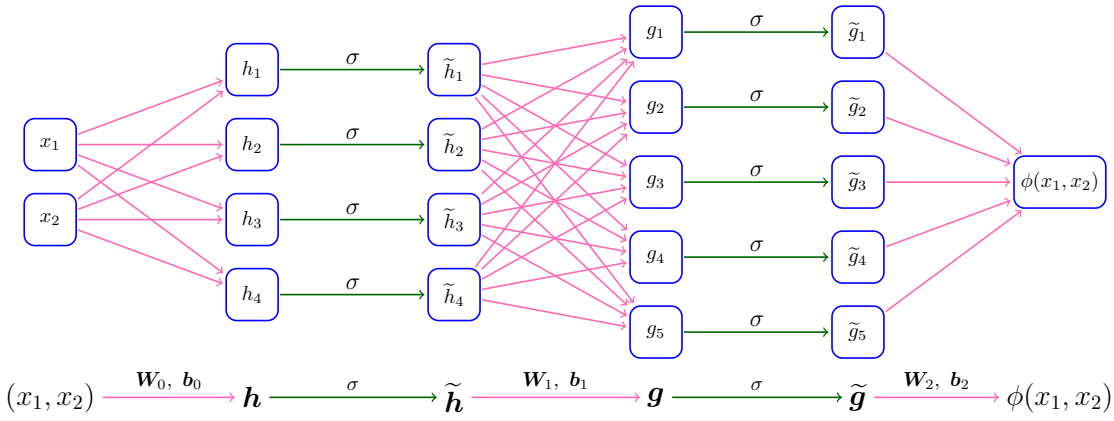

Figure 1: An example of a σ-activated network with width 5 and depth 2. In this example, $\boldsymbol{\theta} = (\boldsymbol{W}_0, \boldsymbol{b}_0, \boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2, \boldsymbol{b}_2)$.

Given a target function f, consider the expected error/risk of $\phi(\boldsymbol{x};\boldsymbol{\theta})$

$$R_{\mathcal{D}}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\boldsymbol{x}\sim U(\mathcal{X})}\left[\ell(\phi(\boldsymbol{x};\boldsymbol{\theta}), f(\boldsymbol{x}))\right]$$

with a loss function typically taken as $\ell(y, y') = \frac{1}{2}|y - y'|^2$, where $U(\mathcal{X})$ is an unknown data distribution over $\mathcal{X}$. For example, when $\ell(y, y') = \frac{1}{2}|y - y'|^2$ and U is a uniform distribution over $\mathcal{X} = [0, 1]^d$,

$$R_{\mathcal{D}}(\boldsymbol{\theta}) = \int_{[0,1]^d} \frac{1}{2}|\phi(\boldsymbol{x};\boldsymbol{\theta}) - f(\boldsymbol{x})|^2 d\boldsymbol{x}.$$

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

The expected risk minimizer $\boldsymbol{\theta}_{\mathcal{D}}$ is defined as

$$\boldsymbol{\theta}_{\mathcal{D}} \coloneqq \arg\min_{\boldsymbol{\theta}} R_{\mathcal{D}}(\boldsymbol{\theta}).$$

It is unachievable in practice since f and $U(\mathcal{X})$ are not available.

In practice, for given samples $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$, we use the empirical risk

$$R_{\mathcal{S}}(\boldsymbol{\theta}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \ell\big(\phi(\boldsymbol{x}_i;\boldsymbol{\theta}), f(\boldsymbol{x}_i)\big).$$

to approximate/model the expected risk $R_{\mathcal{D}}(\boldsymbol{\theta})$. The goal of supervised learning is to identify the empirical risk minimizer

$$\boldsymbol{\theta}_{\mathcal{S}} \coloneqq \arg\min_{\boldsymbol{\theta}} R_{\mathcal{S}}(\boldsymbol{\theta}), \tag{1}$$

obtain $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{S}}) \approx f(\boldsymbol{x})$. When a numerical optimization method is applied to solve (1), it may result in a numerical solution (denoted as $\boldsymbol{\theta}_{\mathcal{N}}$) that is not a global minimizer. Hence, the actually learned function generated by a neural network is $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$. And the discrepancy between the target function f and the actually learned function $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$ is measured by an inference error

$$R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) = \mathbb{E}_{\boldsymbol{x}\sim U(\mathcal{X})}\left[\ell(\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}}), f(\boldsymbol{x}))\right] \overset{e.g.}{=} \int_{[0,1]^d} \tfrac{1}{2}|\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}}) - f(\boldsymbol{x})|^2 d\boldsymbol{x},$$

where the second equality holds when $\ell(y,y') = \frac{1}{2}|y-y'|^2$ and U is a uniform distribution over $\mathcal{X} = [0,1]^d$,

Since $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$ is the expected inference error over all possible data samples, it can quantify how good the learned function $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$ is. Note that

$$\begin{aligned} R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) &= \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})]}_{\text{GE}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{OE}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\leq 0 \text{ by Eq. (1)}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{GE}} + \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{AE}} \\ &\leq \underbrace{R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})}_{\text{Approximation error (AE)}} + \underbrace{[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})]}_{\text{Optimization error (OE)}} + \underbrace{[R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}})] + [R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}}) - R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})]}_{\text{Generalization error (GE)}}. \end{aligned} \tag{2}$$

where the inequality comes from the fact that $[R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{D}})] \leq 0$ since $\boldsymbol{\theta}_{\mathcal{S}}$ is a global minimizer of $R_{\mathcal{S}}(\boldsymbol{\theta})$.

Constructive approximation provides an upper bound of $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$ in terms of the network size, e.g., in terms of the network width and depth, or in terms of the number of parameters. The second term of Equation (2) is bounded by the optimization error of the numerical algorithm applied to solve the empirical loss minimization problem in Equation (1). Note that one only needs to make $R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{N}}) - R_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}})$ small, but not $\boldsymbol{\theta}_{\mathcal{N}} - \boldsymbol{\theta}_{\mathcal{S}}$. The study of the bounds for the third and fourth terms is referred to as the generalization error analysis of neural networks. See Figure 2 for the intuitions of these three errors.

One of the key targets in the area of deep learning is to develop algorithms to reduce $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$. In [1, 2, 3, 4, 5, 6], we provide upper bounds of the approximation error $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$ for several function spaces, which is crucial to estimate an upper bound of $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}})$. Instead of deriving an approximator to attain the approximation error bound,

deep learning algorithms aim to identify a solution $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$ reducing the generalization and optimization errors in Equation (2). Solutions minimizing both generalization and optimization errors will lead to a good solution only if we also have a good upper bound estimate of $R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{D}})$ as shown in Equation (2). Independent of whether our analysis here leads to a good approximator, which is an interesting topic to pursue, the theory here does provide a key ingredient in the error analysis of deep learning algorithms.

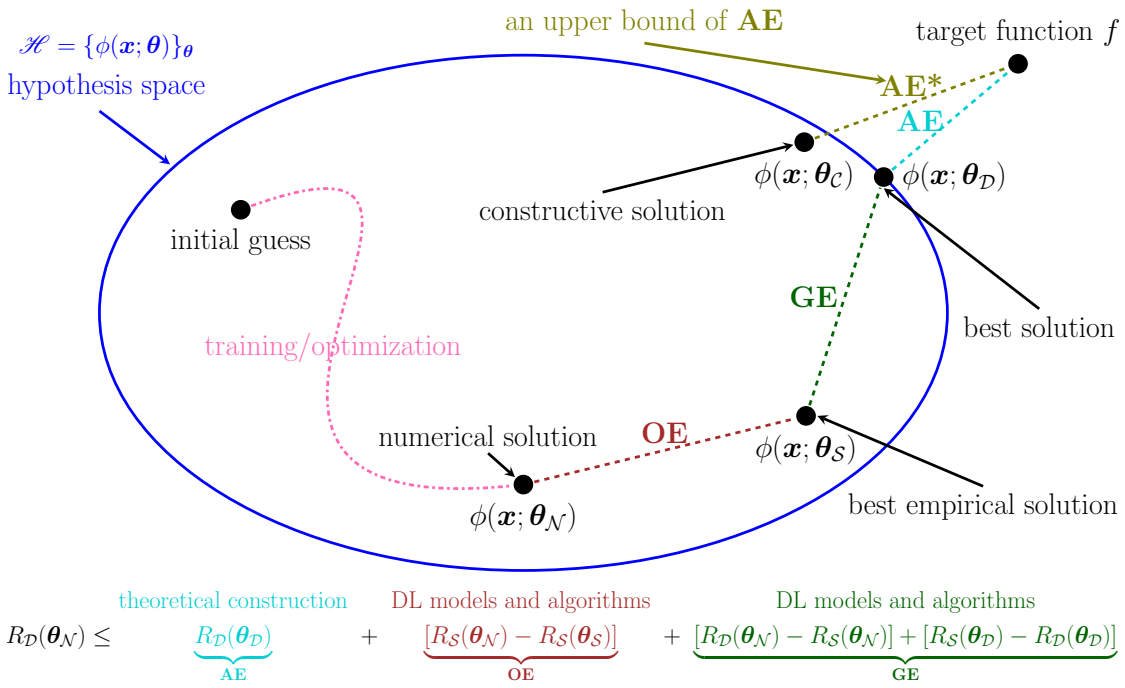

Figure 2: The intuitions of the approximation error (AE), the optimization error (OE), and the generalization error (GE). DL is short of deep learning. One needs to control AE, OE, and GE in order to bound the discrepancy between the target function f and the numerical solution $\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}})$ (what we can get in practice), measured by

$$R_{\mathcal{D}}(\boldsymbol{\theta}_{\mathcal{N}}) = \mathbb{E}_{\boldsymbol{x}\sim U(\mathcal{X})}\left[\ell(\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}}), f(\boldsymbol{x}))\right] \overset{e.g.}{=} \int_{[0,1]^d} \tfrac{1}{2}|\phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{N}}) - f(\boldsymbol{x})|^2 d\boldsymbol{x}.$$

References

[1] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *arXiv e-prints*, page arXiv:2001.03040, Jan 2020.

[2] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019.

[3] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28(5):1768–1811, 2020.

[4] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network with approximation error being reciprocal of width to power of square root of depth. *arXiv e-prints*, page arXiv:2006.12231, Jun 2020.

[5] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Neural network approximation: Three hidden layers are enough. *arXiv e-prints*, page arXiv:2010.14075, Oct 2020.

[6] Shijun Zhang. *Deep Neural Network Approximation via Function Compositions*. PhD thesis, Signapre, 2020.