Summary for Meeting 2

February 24, 2019

Abstract

We are broadly interested in studying the computation power of neural networks. Towards this goal, we could investigate the expressive power of NNs. It is well-known that a function under proper regularity condition could be approximated by a 2NN; this result is known as "universality" of NNs. It might be reasonable to ask whether NNs preserve simililar "universality" in approximating functionals.

- "Information processing": Algorithms/Functions
- Universality of Neural Networks: Function approximation v.s. Functional approximation

1 "Information Processing": Algorithms/Functions

An algorithm consists of (1) input, (2) output, and (3) the procedure that operates on the given input and generates the output. A function (broadly defined) also consists of (1) input, (2) output, and (3) a mapping that connects the input and output. In particular, let $\mathcal{X}$ be the domain and $\mathcal{Y}$. Let $h : \mathcal{X} \to \mathcal{Y}$ be a mapping such that

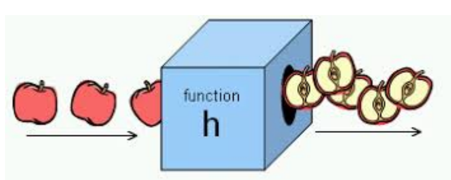
$$h(x) = y, \quad \text{where } x \in \mathcal{X}, y \in \mathcal{Y}.$$

The mapping h could even be random. With some abuse of terminology, we could write

$$h(x, \xi) = y$$

to emphasize the randomness in h. Notably, y is also random. We give a couple of examples next.

- $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$, and $h : \mathbb{R}^d \to \mathbb{R}$.
- $\mathcal{X} = \{\text{all } p : \mathbb{R}_+ \to \{0, 1\}\}$, $\mathcal{Y} = \{\text{all } q : \mathbb{R}_+ \to \{0, 1\}\}$, and $h : \mathcal{X} \to \mathcal{Y}$.



2 Universality of Neural Networks

2.1 Function approximation

It is well-known that a function $h : \mathbb{R}^d \to \mathbb{R}$ (under mild regularity conditions) admits a neural network approximation.

Theorem 1. *[Mon18, Cyb89] Let $f : \mathbb{R}^d \to \mathbb{R}$. Assume that $\mathbb{E}\left[f(\boldsymbol{x})^2\right] < \infty$, where the expectation is taken over the randomness of $\boldsymbol{x}$. Assume $\sigma : \mathbb{R} \to \mathbb{R}$ to be continuous with*

$$\lim_{x\to\infty} \sigma(x) = 1, \quad \textit{and} \quad \lim_{x\to-\infty} \sigma(x) = 0. \tag{1}$$

Then, for any $\epsilon > 0$, there exists $M = M(\epsilon)$ such that

$$\inf_{\{(a_i,b_i,\boldsymbol{w}_i)\}} \mathbb{E}\left[\left(f(\boldsymbol{x}) - \frac{1}{M}\sum_{i=1}^{M} a_i \cdot \sigma\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i\right)\right)^2\right] \le \epsilon. \tag{2}$$

Similar universality result on ReLU activation can be found in [KB18].

Remark 2. Here $\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i$ can be viewed as the local input of neuron i, and $\sigma\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i\right)$ is the local output. From an algorithm's prospective, the above theorem says that the given function f can be well-approximated as follows:

- Given $\boldsymbol{x} \in \mathbb{R}^d$, obtain a collection of one-dimensional local inputs $\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i$;
- Each neuron computes $\sigma\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i\right)$;
- These locally computed quantities are aggregated $\frac{1}{M}\sum_{i=1}^{M} a_i \cdot \sigma\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle + b_i\right)$ to obtain a final approximation of $f(\boldsymbol{x})$.

2.2 Functional approximation

In computational neuroscience, we might need to deal with the scenario where the inputs are functions rather than simple vectors in $\mathbb{R}^d$. It might worth asking whether there exists similar universality results of NNs for functional approximation or not. One type of universality result, by Maass, will be discussed next week.

Note that Euclidean space has nice properties; for instance, some relevant operations can be defined. To work with functionals, we might want to put some constraints on the collection of input functions. In addition, we might also need to constraint the family of functionals under consideration.

References

[Cyb89] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 2

[KB18] Jason M Klusowski and Andrew R Barron. Approximation by combinations of relu and squared relu ridge functions with l1 and l0 controls. 2018. 2

[Mon18] Andrea Montanari. One lecture on two-layer neural networks. 2018. 2