Lecture 22: November 28

Lecturer: Csaba Szepesvári *Scribes: Kushagra Chandak*

22.1 Outline

- Introduction to neural networks.
- Function approximation.
- Depth vs width in neural networks.

22.2 Neural Networks

A two-layered (one hidden and one output layer) fully connected neural network with m units in the hidden layer is a map $f : \mathbb{R}^d \to \mathbb{R}$ given by

$$f_w(x) = \sum_{i=1}^{m} u_i h(\theta_i^\top x + b_i)\,,$$

where $h : \mathbb{R} \to \mathbb{R}$ is the activation function, $x \in \mathbb{R}^d$ is the input vector, $\theta_i \in \mathbb{R}^d$ is the weight vector, $b_i \in \mathbb{R}$ is the bias/threshold, $u_i \in \mathbb{R}$ is the weight to the output, and $w = (\theta, u, b) \in \mathbb{R}^{m(d+2)}$ are the parameters.

22.2.1 Function Approximation with Neural Networks

Let $\mathcal{F}_m^{(h)} = \{f_w : w \in \mathcal{W}_m\}$, where $\mathcal{W}_m = \mathbb{R}^{m(d+2)}$, be the two-layered neural network function class with m hidden units and activation function h. The next theorem shows that $f \in \mathcal{F}_m$ is a universal approximator.

In this section, we will see how well we can approximate functions of different kinds with neural networks.

Theorem 22.1 (Leshno, 1993)**.** *Let $h : \mathbb{R} \to \mathbb{R}$ be such that $h \notin \mathbb{R}[x]$ (not a polynomial). Let $K \subset \mathbb{R}^d$ be compact. Then $\mathcal{F}_m^{(h)}|_K = \left\{ f|_K : f \in \mathcal{F}_m^{(h)} \right\}$ is dense in $C(K)$.*

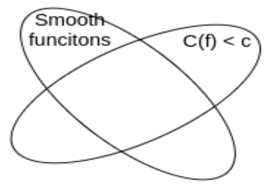
To state the next result, let us introduce a set of functions

$$\Gamma_r = \left\{ f : \mathbb{R}^d \to R : \exists \tilde{f} : \mathbb{R}^d \to C \text{ s.t. } f(x) = \int e^{i\omega^\top x} \tilde{f}(\omega) d\omega,\ \forall x \in B_r \right\}\,,$$

where $B_r = \left\{ x^d : \|x\|_2 \le r \right\}$ is a ball of radius r. The function $\tilde{f}$ is the Fourier transform of f up to constant factors. We have a complexity/smoothness measure/coefficient for $f \in \Gamma_r$ (assuming there exists a unique $\tilde{f}$ for f):

$$C(f) = \int \|\omega\|_2 |\tilde{f}(\omega)| d\omega\,.$$

The quantity $C(f)$ roughly measures the "energy" of f at high frequency. Thus, f is smooth if $C(f)$ is small. With $C(f)$ in hand, we state our next result:



h

Figure 22.1: Barron's theorem (Theorem 22.2) does not hold for all smooth functions but only a "slice".

Theorem 22.2 (Barron, 1993)**.** *Let $h : \mathbb{R} \to \mathbb{R}$ be a measurable bounded function such that $\lim_{z\to-\infty} h(z) = 0$ and $\lim_{z\to\infty} h(z) = 1$. Let $f \in \Gamma_r$ such that $C(f) < \infty$ and $\mu \in \mathcal{M}_1(B_r)$. Then for all $m \geq 1$*

$$\inf_{w\in\mathcal{W}_m} \|f - f(0) - f_w\|_{L_2(\mu)} \leq \frac{(2rC(f))^2}{m} .$$

Remark 22.3. Note that the above result is independent of d. When we approximate a smooth function with polynomial, we get a rate of roughly $(1/m)^{s/d}$, where s is the number of continuous derivative of the target function f. So the above result does not tell us that for any smooth function, the approximation error goes down with $1/m$ rate. But functions with finite $C(f)$ creates a subset of smooth functions for which we get the $1/m$ rate (see Fig. 22.1).

Remark 22.4. Some of the common choices of the activation function are sigmoid ($h(z) = 1/(1+e^{-z})$) and ReLU ($h(z) = \max(0, z)$). Note that while sigmoid satisfies the condition of Theorem 22.2, ReLU does not. However, for ReLU, we can write $s(z) = h(z) - h(z-1)$ such that s satisfies the condition.

Does depth in neurals networks give some advantage? For the next result, let $d = 1$ and the activation function is ReLU. We also index the neural network class with number of layers:

$$\mathcal{F}_{k,m} = \{f : [0,1] \to \mathbb{R} : \ f \text{ can be implemented by a NN with} \leq k \text{ layers and} \leq m \text{ hidden units}\} .$$

Theorem 22.5 (Telgarsky, 2016)**.** *Let $k \geq 3$. Then*

$$\sup_{f\in\mathcal{F}_{2k^2,2}} \inf_{g\in\mathcal{F}_{k,2^{k-2}}} \|f - g\|_\infty \geq \frac{1}{16} .$$

Proof intuition. The proof is done by constructing a function f_k which is difficult to approximate using shallow networks. Let $f_0(x) = \max(0, \min(2x, 2(1-x)))$ on $[0,1]$. Note that $f_0(x)$ can be implemented by a 2 layer neural network with $m = 2$, $\theta_1 = 2$, $\theta_2 = -4$, $b_1 = 0$, and $b_2 = -0.5$ so that

$$f_0(x) = 2\max(0, x) - 4\max(0, x - 0.5) = w_1 h(x) + w_2 h(x - 0.5) .$$

Let $f_k(x) = f_0(f_{k-1}(x))$ with $k \geq 1$. Then $f_k(x)$ can be represented by a $2k$ layer neural network with 2 units in each hidden layer. Fig. 22.2 shows f_k for $k = 0, 1, 2$.

Definition 22.6 (Crossing Number)**.** The crossing number of a function $f : [0,1] \to [0,1]$ is the number of segments in the graph on which f is above the line $y = \frac{1}{2}$.

Combining the below two claims gives us the result.

Claim 22.7. *For every measurable $g : [0,1] \to [0,1]$ such that $C(g) < 2^{k-1}$, $\|f_k - g\|_{L_1} \geq \frac{1}{16}$.*

Claim 22.8. *We have that*

$$\max\{C(g) : g \in \mathcal{F}_{l,m}\} \leq 2(2m)^l .$$

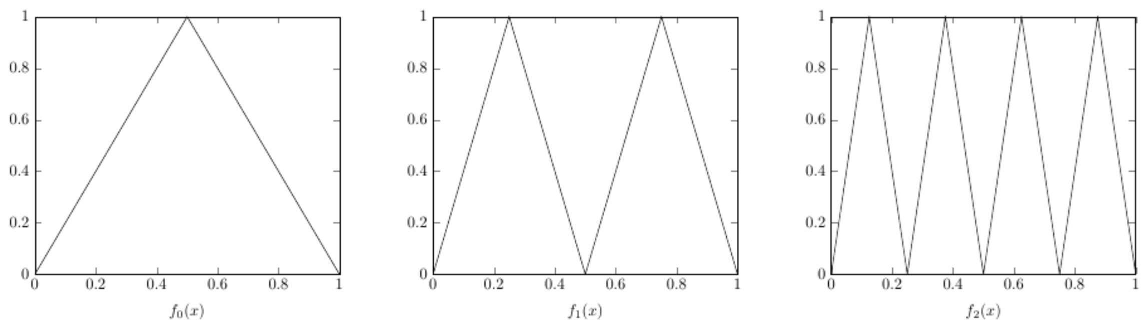


Figure 22.2: $f_k(x)$ for $k = 0, 1, 2$.