Approximation Theory

Specific Setups

- "Average" approximation: given a distribution μ

$$\|f - g\|_\mu = \int_x |f(x) - g(x)| \, d\mu(x)$$

- "Everywhere" approximation

$$\|f - g\|_\infty = \sup_x |f(x) - g(x)| \geq \|f - g\|_\mu$$

Multivariate Approximation

Theorem: Let g be a continuous function that satisfies $\|x - x'\|_\infty \leq \delta \Rightarrow |g(x) - g(x')| \leq \epsilon$ (Lipschitzness).
Then there exists a 3-layer ReLU neural network with $O(\frac{1}{\delta^d})$ nodes that satisfy

$$\int_{[0,1]^d} |f(x) - g(x)| dx = \|f - g\|_1 \leq \epsilon$$

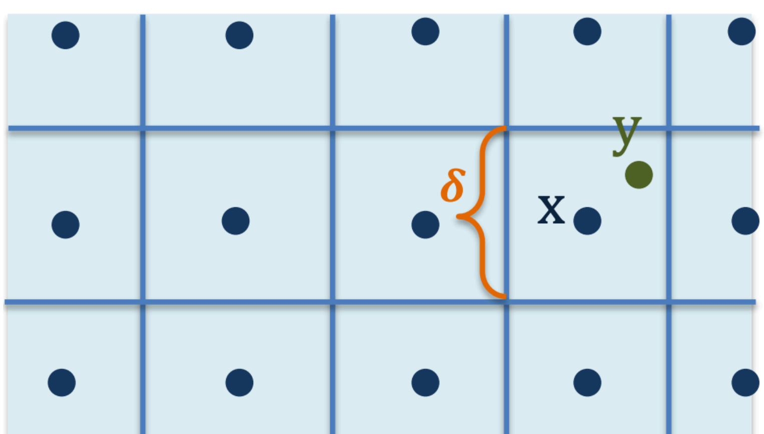
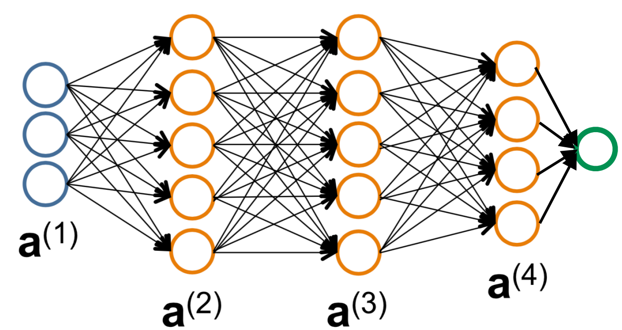


Figure credit to Andrej Risteski



Universal Approximation

Definition: A class of functions $\mathcal{F}$ is universal approximator over a compact set S (e.g., $[0,1]^d$), if for every continuous function g and a target accuracy $\epsilon > 0$, there exists $f \in \mathcal{F}$ such that

$$\sup_{x \in S} |f(x) - g(x)| \leq \epsilon$$

Stone-Weierstrass Theorem

Theorem: If $\mathscr{F}$ satisfies

1. Each $f \in \mathscr{F}$ is continuous.

2. $\forall x, \exists f \in \mathscr{F}, f(x) \neq 0$

3. $\forall x \neq x', \exists f \in \mathscr{F}, f(x) \neq f(x')$

4. $\mathscr{F}$ is closed under multiplication and vector space operations,

Then $\mathscr{F}$ is a universal approximator:

$$\forall g : S \to R, \epsilon > 0, \exists f \in \mathscr{F}, \|f - g\|_\infty \leq \epsilon.$$

Example: cos activation

Example: cos activation

Other Examples

Exponential activation

ReLU activation

Curse of Dimensionality

- Unavoidable in the worse case

Barron's Theory

- Can we avoid the curse of dimensionality for "nice" functions?
- What are nice functions?
  - Fast decay of the Fourier coefficients

- Fourier basis functions:
  $$\{e_w(x) = e^{i\langle w,x\rangle} = \cos(\langle w, x\rangle) + i\sin(\langle w, x\rangle) \mid w \in \mathbb{R}^d\}$$

- Fourier coefficient: $\hat{f}(w) = \int_{\mathbb{R}^d} f(x)e^{-i\langle w,x\rangle}dx$
- Fourier integral / representation: $f(x) = \int_{\mathbb{R}^d} \hat{f}(w)e^{i\langle w,x\rangle}dw$

Barron's Theorem

Definition: The Barron constant of a function f is:

$$C \triangleq \int_{\mathbb{R}^d} \|w\|_2 |\hat{f}(w)| dw.$$

Theorem (Barron '93): For any $g : \mathbb{B}_1 \to \mathbb{R}$ where $\mathbb{B}_1 = \{x \in \mathbb{R} : \|x\|_2 \leq 1\}$ is the unit ball, there exists a 3-layer neural network f with $O(\frac{C^2}{\epsilon})$ neurons and sigmoid activation function such that

$$\int_{\mathbb{B}_1} (f(x) - g(x))^2 dx \leq \epsilon.$$

Examples

- Gaussian function: $f(x) = (2\pi\sigma^2)^{d/2}\exp\left(-\frac{\|x\|_2^2}{2\sigma^2}\right)$

- Other functions:
  - Polynomials
  - Function with bounded derivatives

Proof Ideas for Barron's Theorem

Step 1: show any continuous function can be written as an infinite neural network with cosine-like activation functions.

(Tool: Fourier representation.)

Step 2: Show that a function with small Barron constant can be approximated by a convex combination of a small number of cosine-like activation functions.

(Tool: subsampling / probabilistic method.)

Step 3: Show that the cosine function can be approximated by sigmoid functions.

(Tool: classical approximation theory.)

Simple Infinite Neural Nets

Definition: An infinite-wide neural network is defined by a signed measure ν over neuron weights (w, b)

$$f(x) = \int_{w \in \mathbb{R}^d, b \in \mathbb{R}} \sigma(w^\top x + b) d\nu(w, b).$$

Theorem: Suppose $g : \mathbb{R} \to \mathbb{R}$ is differentiable, if $x \in [0,1]$, then $g(x) = \int_0^1 \mathbf{1}\{x \geq b\} \cdot g'(b) db + g(0)$

Step 1: Infinite Neural Nets

The function can be written as

$$f(x) = f(0) + \int_{\mathbb{R}^d} |\hat{f}(w)| (\cos(b_w + \langle w, x \rangle) - \cos(b_w)) dw.$$

Step 1: Infinite Neural Nets Proof

The function can be written as

$$f(x) = f(0) + \int_{\mathbb{R}^d} |\hat{f}(w)| (\cos(b_w + \langle w, x \rangle) - \cos(b_w)) dw.$$

Step 2: Subsampling

Writing the function as the expectation of a random variable:

$$f(x) = f(0) + \int_{\mathbb{R}^d} \frac{|\hat{f}(w)|\|w\|_2}{C} \left(\frac{C}{\|w\|_2}(\cos(b_w + \langle w, x \rangle) - \cos(b_w)) \right) dw.$$

Step 2: Subsampling

Writing the function as the expectation of a random variable:

$$f(x) = f(0) + \int_{\mathbb{R}^d} \frac{|\hat{f}(w)|\|w\|_2}{C} \left(\frac{C}{\|w\|_2} (\cos(b_w + \langle w, x \rangle) - \cos(b_w)) \right) dw.$$

Sample one $w \in \mathbb{R}^d$ with probability $\dfrac{|\hat{f}(w)|\|w\|_2}{C}$ for r times.

Step 3: Approximating the Cosines

Lemma: Given $g_w(x) = \frac{C}{\|w\|_2}(\cos(b_w + \langle w, x \rangle) - \cos(b_w))$, there exists a 2-layer neural network f_0 of size $O(1/\epsilon)$ with sigmoid activations, such that $\sup_{x \in [-1,1]} |f_0(y) - h_w(y)| \leq \epsilon$.

Depth Separation

So far we only talk about 2-layer or 3-layer neural networks.

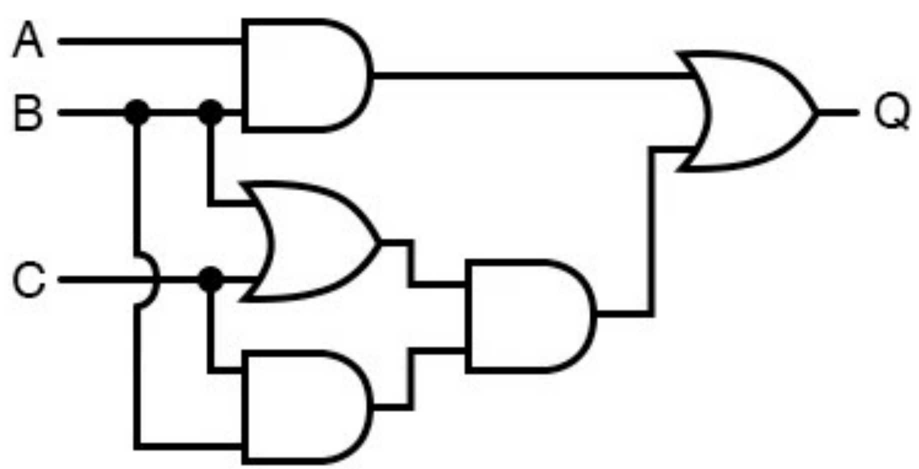
Why we need **Deep** learning?

Can we show deep neural networks are **strictly** better than shallow neural networks?

A brief history of depth separation

Early results from theoretical computer science

Boolean circuits: a directed acyclic graph model for computation over binary inputs; each node (“gate”) performs an operation (e.g. OR, AND, NOT) on the inputs from its predecessors.



A brief history of depth separation

Early results from theoretical computer science

Boolean circuits: a directed acyclic graph model for computation over binary inputs; each node ("gate") performs an operation (e.g. OR, AND, NOT) on the inputs from its predecessors.

Depth separation: the difference of the computation power: shallow vs deep Boolean circuits.

Håstad ('86): parity function cannot be approximated by a small constant-depth circuit with OR and AND gates.

Modern depth-separation in neural networks

- **Related architectures / models of computation**
  - Sum-product networks [Bengio, Delalleau ’11]
- **Heuristic measures of complexity**
  - Bound of number of linear regions for ReLU networks [Montufar, Pascanu, Cho, Bengio ‘14]
- **Approximation error**
  - A small deep network cannot be approximated by a small shallow network [Telgarsky ’15]

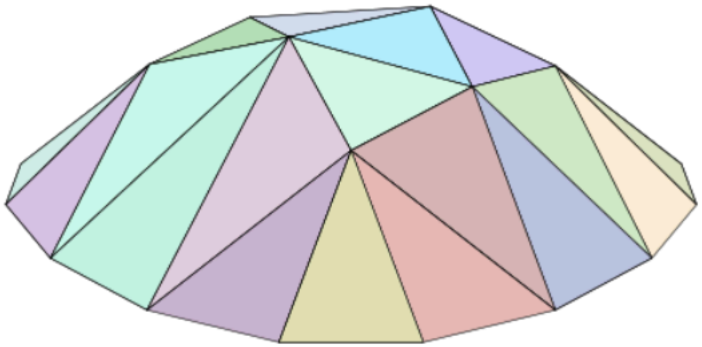
Shallow Nets Cannot Approximate Deep Nets

Theorem (Telgarsky '15): For every $L \in \mathbb{N}$, there exists a function $f : [0,1] \to [0,1]$ representable as a network of depth $O(L^2)$, with $O(L^2)$ nodes, and ReLU activation such that, for every network $g : [0,1] \to \mathbb{R}$ of depth L and $\leq 2^L$ nodes, and ReLU activation, we have

$$\int_{[0,1]} |f(x) - g(x)| \, dx \geq \frac{1}{32}.$$

Intuition

A ReLU network f is piecewise linear, we can subdivide domain into a finite number of polyhedral pieces $(P_1, P_2, \ldots, P_N)$ such that in each piece, f is linear: $\forall x \in P_i, f(x) = A_i x + b_i$.



Deeper neural networks can make exponentially more regions than shallow neural networks.

Make each region has different values, so shallow neural networks cannot approximate.

Benefits of depth for smooth functions

Theorem (Yarotsky '15): Suppose $f : [0,1]^d \to \mathbb{R}$ has all partial derivatives of order r with coordinate-wise bound in $[-1,1]$, and let $\epsilon > 0$ be given. Then there exists a $O(\ln \frac{1}{\epsilon})$ - depth and $\left(\frac{1}{\epsilon}\right)^{O(\frac{d}{r})}$ -size network so that $\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \epsilon$.

Remarks

- All results discussed are existential: they prove that a good approximator exists. Finding one efficiently (e.g., using gradient descent) is the next topic (optimization).
- The choices of non-linearity are usually very flexible: most results we saw can be re-proven using different non-linearities.
- There are other approximation error results: e.g., deep and narrow networks are universal approximators.
- Depth separation for optimization and generalization is widely open.