UAT: From Shallow to Deep

Ju Sun
Computer Science & Engineering
University of Minnesota, Twin Cities

January 30, 2020

Logistics

- LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

Logistics

- LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

  Mind LaTeX! Mind your math!

  * Ten Signs a Claimed Mathematical Breakthrough is Wrong

> Inspired by Sean Carroll's closely-related Alternative-Science Respectability Checklist, without further ado I now offer the *Ten Signs a Claimed Mathematical Breakthrough is Wrong*.
>
> **1. The authors don't use TeX.** This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. David Deutsch and Lov Grover are among the only known false positives.

Logistics

- LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

  Mind LaTeX! Mind your math!

  * Ten Signs a Claimed Mathematical Breakthrough is Wrong

    Inspired by Sean Carroll's closely-related Alternative-Science Respectability Checklist, without further ado I now offer the *Ten Signs a Claimed Mathematical Breakthrough is Wrong*.

    1. The authors don't use TeX. This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. David Deutsch and Lov Grover are among the only known false positives.

  * Paper Gestalt ($50\%/18\%$, 2009)

Logistics

– LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

Mind LaTeX! Mind your math!

* Ten Signs a Claimed Mathematical Breakthrough is Wrong

  Inspired by Sean Carroll's closely-related Alternative-Science Respectability Checklist, without further ado I now offer the *Ten Signs a Claimed Mathematical Breakthrough is Wrong*.

  1. The authors don't use TeX. This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. David Deutsch and Lov Grover are among the only known false positives.

* Paper Gestalt ($50\%/18\%$, 2009) $\Longrightarrow$ Deep Paper Gestalt ($50\%/0.4\%$, 2018)

Logistics

– LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

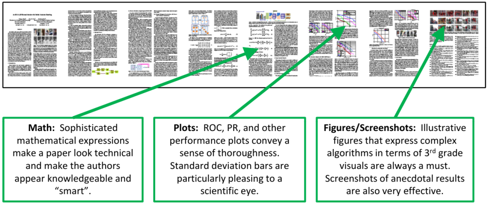
Mind LaTeX! Mind your math!

* Ten Signs a Claimed Mathematical Breakthrough is Wrong

  Inspired by Sean Carroll's closely-related Alternative-Science Respectability Checklist, without further ado I now offer the *Ten Signs a Claimed Mathematical Breakthrough is Wrong*.

  1. The authors don't use TeX. This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. David Deutsch and Lov Grover are among the only known false positives.

* Paper Gestalt ($50\%/18\%$, 2009) $\Longrightarrow$ Deep Paper Gestalt ($50\%/0.4\%$, 2018)



Logistics

– LaTeX source of homework posted in Canvas (Thanks to Logan Stapleton!)

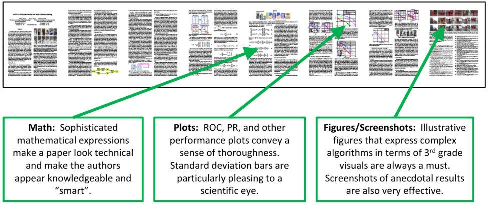
Mind LaTeX! Mind your math!

* Ten Signs a Claimed Mathematical Breakthrough is Wrong

  Inspired by Sean Carroll's closely-related Alternative-Science Respectability Checklist, without further ado I now offer the *Ten Signs a Claimed Mathematical Breakthrough is Wrong*.

  1. The authors don't use TeX. This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. David Deutsch and Lov Grover are among the only known false positives.

* Paper Gestalt ($50\%/18\%$, 2009) $\Longrightarrow$ Deep Paper Gestalt ($50\%/0.4\%$, 2018)



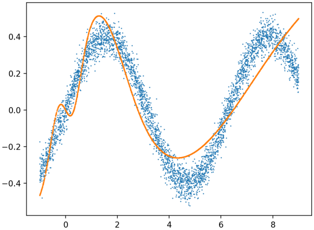
– Matrix Cookbook? Yes and No

Outline

Recap and more thoughts

From shallow to deep NNs

Supervised learning as function approximation



- Underlying true function: f_0
- Training data: $\boldsymbol{y}_i \approx f_0(\boldsymbol{x}_i)$
- Choose a family of functions $\mathcal{H}$, so that $\exists f \in \mathcal{H}$ and

  f and f_0 are close

- **Approximation capacity**: $\mathcal{H}$ matters (e.g., linear? quadratic? sinusoids? etc)
- **Optimization & Generalization**: how to find the best $f \in \mathcal{H}$ matters

We focus on approximation capacity now.

Approximation capacities of NNs

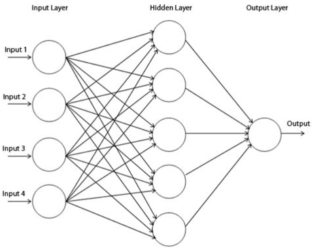
– A single neuron has limited capacity

Approximation capacities of NNs

- A single neuron has limited capacity
- Deep NNs with linear activation is no better

Approximation capacities of NNs

- A single neuron has limited capacity
- Deep NNs with linear activation is no better
- Add in both depth and nonlinearity activation



two-layer network, linear activation at output

universal approximation theorem

> The 2-layer network can approximate **arbitrary** continuous functions **arbitrarily** well, provided that the hidden layer is **sufficiently wide**.

[A] universal approximation theorem (UAT)

Theorem (UAT, [Cybenko, 1989, Hornik, 1991])

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a nonconstant, bounded, and continuous function. Let I_m denote the m-dimensional unit hypercube $[0,1]^m$. The space of real-valued continuous functions on I_m is denoted by $C(I_m)$. Then, given any $\varepsilon > 0$ and any function $f \in C(I_m)$, there exist an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$ for $i = 1, \ldots, N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \sigma \left(w_i^T x + b_i \right)$$

as an approximate realization of the function f; that is,

$$|F(x) - f(x)| < \varepsilon$$

for all $x \in I_m$.

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$?

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$? think of the component functions

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$? think of the component functions
- Map to $[0,1]$, $\{-1,+1\}$, $[0,\infty)$? choose appropriate activation σ at the output

$$F(x) = \sigma\left(\sum_{i=1}^{N} v_i \sigma\left(w_i^T x + b_i\right)\right)$$

... universality holds in modified form

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$? think of the component functions
- Map to $[0,1]$, $\{-1,+1\}$, $[0,\infty)$? choose appropriate activation σ at the output

$$F(x) = \sigma\left(\sum_{i=1}^{N} v_i \sigma\left(w_i^T x + b_i\right)\right)$$

  ... universality holds in modified form
- Get deeper? three-layer NN?

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$? think of the component functions
- Map to $[0,1]$, $\{-1,+1\}$, $[0,\infty)$? choose appropriate activation σ at the output

$$F(x) = \sigma\left(\sum_{i=1}^{N} v_i \sigma\left(w_i^T x + b_i\right)\right)$$

  ... universality holds in modified form
- Get deeper? three-layer NN? change to matrix-vector notation for convenience

$$F(x) = \boldsymbol{w}^\intercal \sigma(W_2 \sigma(W_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2) \quad \text{as} \sum_k w_k g_k(\boldsymbol{x})$$

  use w_k's to linearly combine the same function
- **For geeks**: approximate both f and f'?

Thoughts

- Approximate continuous functions with vector outputs, i.e., $I_m \to \mathbb{R}^n$? think of the component functions
- Map to $[0, 1]$, $\{-1, +1\}$, $[0, \infty)$? choose appropriate activation σ at the output

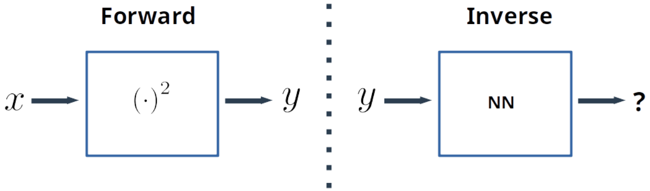
$$F(x) = \sigma\left(\sum_{i=1}^{N} v_i \sigma\left(w_i^T x + b_i\right)\right)$$

  ... universality holds in modified form
- Get deeper? three-layer NN? change to matrix-vector notation for convenience

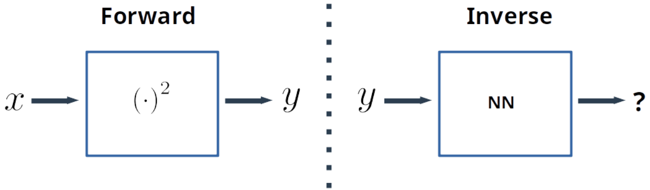
$$F(x) = \boldsymbol{w}^\intercal \sigma(W_2 \sigma(W_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2) \quad \text{as} \sum_k w_k g_k(\boldsymbol{x})$$

  use w_k's to linearly combine the same function
- **For geeks**: approximate both f and f'? check out [Hornik et al., 1990]

Learn to take square-root



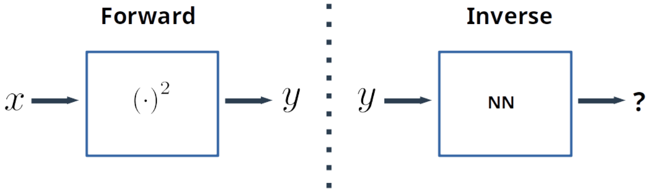
Learn to take square-root



Suppose we lived in a time square-root is not defined ...

– Training data: $\left\{x_i, x_i^2\right\}_i$, where $x_i \in \mathbb{R}$

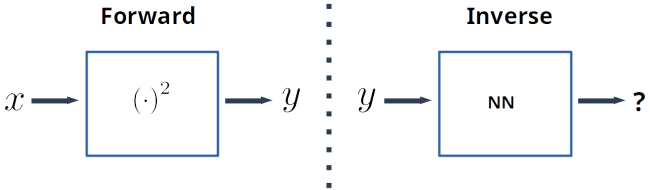
Learn to take square-root



Suppose we lived in a time square-root is not defined ...

- Training data: $\left\{x_i, x_i^2\right\}_i$, where $x_i \in \mathbb{R}$
- Forward: if $x \mapsto y$, $-x \mapsto y$ also

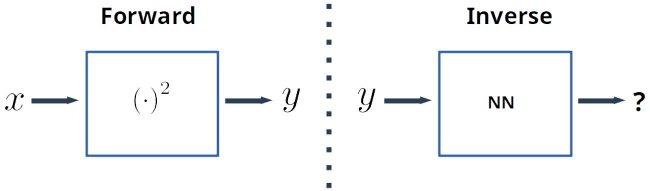
Learn to take square-root



Suppose we lived in a time square-root is not defined ...

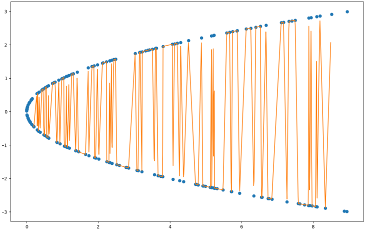
- Training data: $\{x_i, x_i^2\}_i$, where $x_i \in \mathbb{R}$
- Forward: if $x \mapsto y$, $-x \mapsto y$ also
- To invert, what to output? What if just throw in the training data?

Learn to take square-root

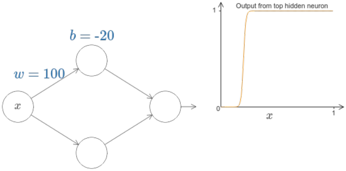


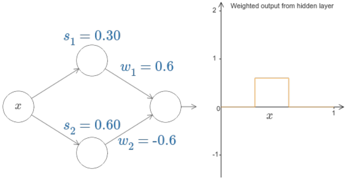
Suppose we lived in a time square-root is not defined ...

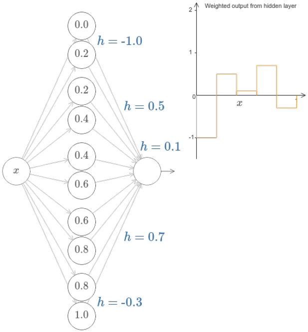
- Training data: $\{x_i, x_i^2\}_i$, where $x_i \in \mathbb{R}$
- Forward: if $x \mapsto y$, $-x \mapsto y$ also
- To invert, what to output? What if just throw in the training data?

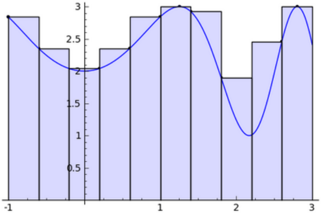


Visual “proof” of UAT

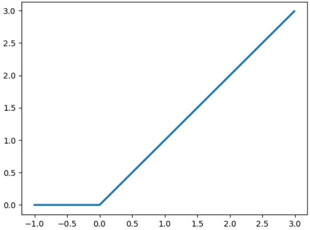




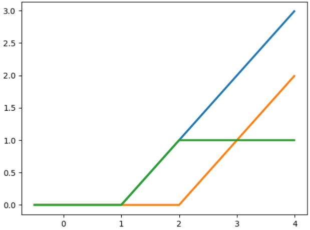




What about ReLU?

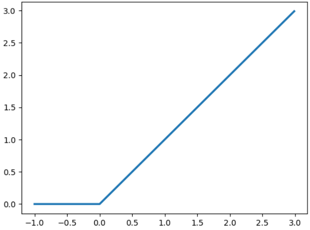


ReLU

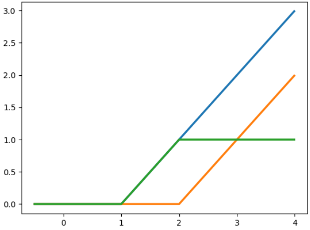


difference of ReLU's

What about ReLU?



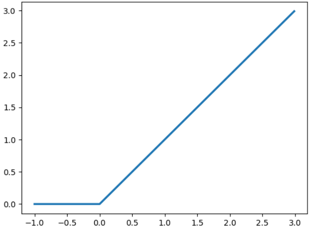
ReLU



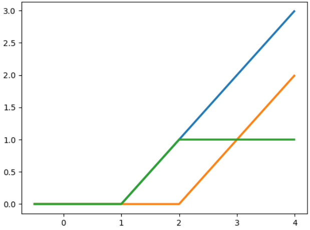
difference of ReLU's

what happens when the slopes of the ReLU's are changed?

What about ReLU?



ReLU

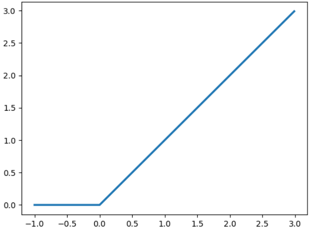


difference of ReLU's

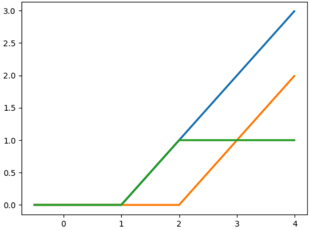
what happens when the slopes of the ReLU's are changed?

How general σ can be?

What about ReLU?



ReLU



difference of ReLU's

what happens when the slopes of the ReLU's are changed?

How general σ can be? ... enough when σ not a polynomial [Leshno et al., 1993]

Outline

Recap and more thoughts

From shallow to deep NNs

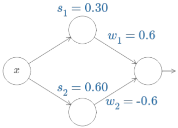
What's bad about shallow NNs?

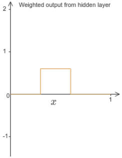
From UAT, "... there exist an interger N, ...", but how large?

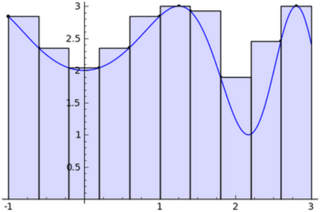
What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $1D$?



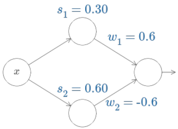


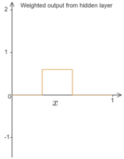


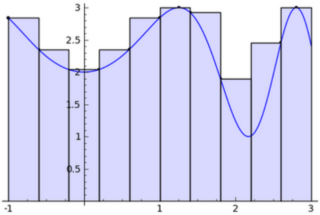
What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $1D$?





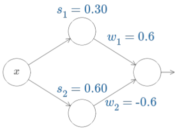


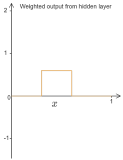
Assume the target f is 1-Lipschitz, i.e., $|f(x) - f(y)| \leq |x - y|, \forall\, x, y \in \mathbb{R}$

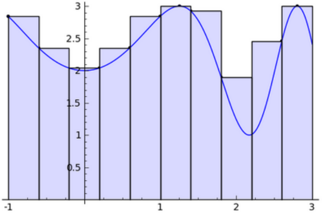
What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $1D$?







Assume the target f is 1-Lipschitz, i.e., $|f(x) - f(y)| \leq |x - y|, \forall\, x, y \in \mathbb{R}$

For ε accuracy, need $\frac{1}{\varepsilon}$ bumps

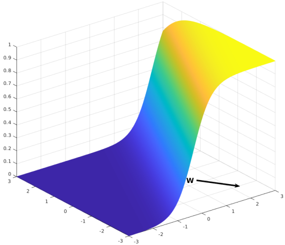
What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$? Visual proof in 2D first

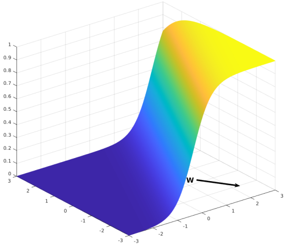


$\sigma(\boldsymbol{w}^\intercal \boldsymbol{x} + \boldsymbol{b})$, σ sigmod

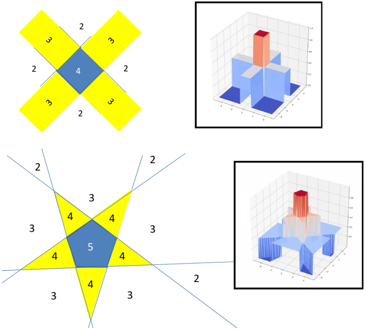
What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$? Visual proof in 2D first



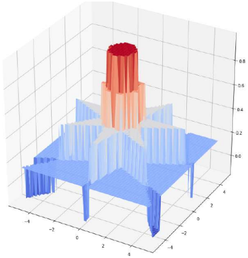
$\sigma(\boldsymbol{w}^\intercal \boldsymbol{x} + \boldsymbol{b})$, σ sigmod approach 2D step function when making $\boldsymbol{w}$ large



Credit: CMU 11-785

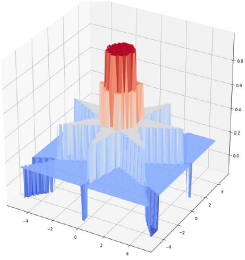
Visual proof for 2D functions

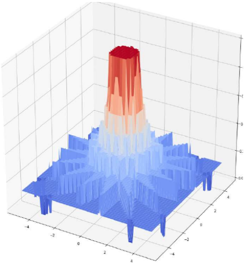
Keep increasing the number of step functions that are distributed evenly ...



Visual proof for 2D functions

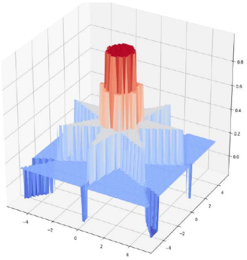
Keep increasing the number of step functions that are distributed evenly ...

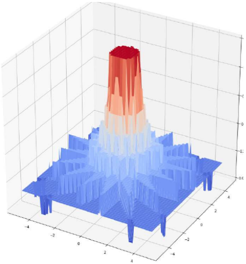


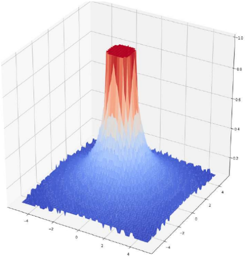


Visual proof for 2D functions

Keep increasing the number of step functions that are distributed evenly ...







Visual proof for 2D functions

Keep increasing the number of step functions that are distributed evenly ...

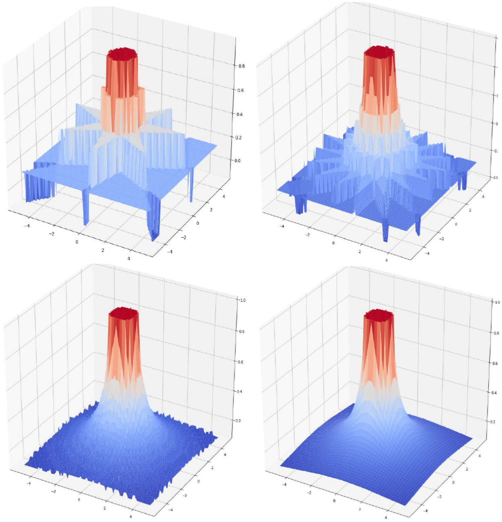


Image Credit: CMU 11-785

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$?

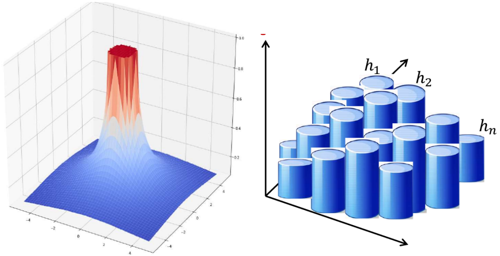


Image Credit: CMU 11-785

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$?

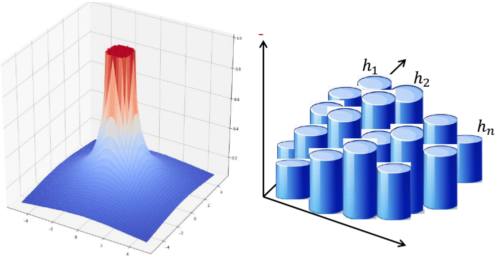


Image Credit: CMU 11-785

Assume the target f is 1-Lipschitz, i.e., $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2, \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$?

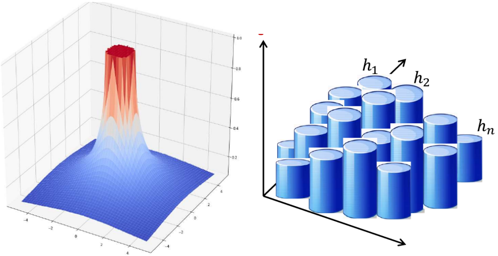


Image Credit: CMU 11-785

Assume the target f is 1-Lipschitz, i.e., $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2, \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$

For ε accuracy, need $O\left(\varepsilon^{-2}\right)$ bumps.

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$?

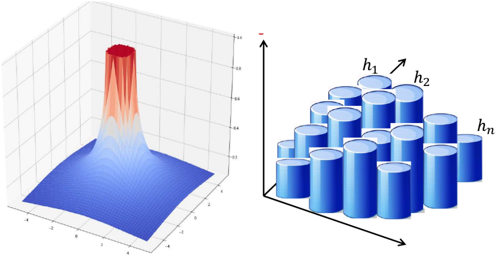


Image Credit: CMU 11-785

Assume the target f is 1-Lipschitz, i.e., $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2, \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$

For ε accuracy, need $O\left(\varepsilon^{-2}\right)$ bumps. What about the n-D case?

What's bad about shallow NNs?

From UAT, "... there exist an interger N, ...", but how large?

What happens in $2D$?

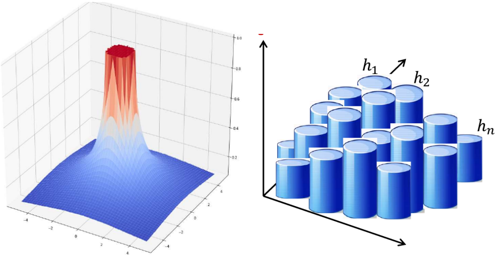


Image Credit: CMU 11-785

Assume the target f is 1-Lipschitz, i.e., $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2, \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$

For ε accuracy, need $O\left(\varepsilon^{-2}\right)$ bumps. What about the n-D case? $O(\varepsilon^{-n})$.

What's good about deep NNs?

Learn Boolean functions ($f : \{+1, -1\}^n \mapsto \{+1, -1\}$): DNNs can have #nodes linear in n, whereas 2-layer NN needs exponential nodes (more in HW1)

What's good about deep NNs?

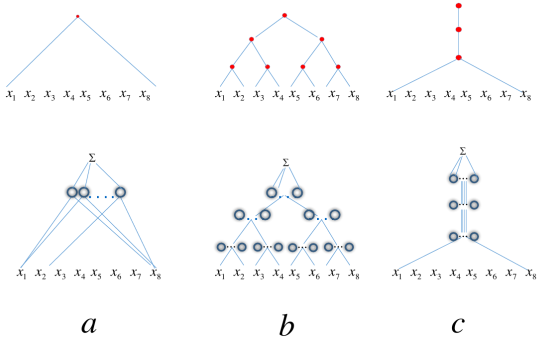
Learn Boolean functions ($f : \{+1, -1\}^n \mapsto \{+1, -1\}$): DNNs can have #nodes linear in n, whereas 2-layer NN needs exponential nodes (more in HW1)

What general functions set deep and shallow NNs apart?

What's good about deep NNs?

Learn Boolean functions ($f : \{+1, -1\}^n \mapsto \{+1, -1\}$): DNNs can have #nodes linear in n, whereas 2-layer NN needs exponential nodes (more in HW1)

What general functions set deep and shallow NNs apart?



A family: compositional function [Poggio et al., 2017]

Compositional functions

$$f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), \\ h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8))) \quad (4)$$

W_m^n: class of n-variable functions with partial derivatives up to m-th order, $W_m^{n,2} \subset W_m^n$ is the compositional subclass following binary tree structures

Theorem 1. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial. For $f \in W_m^n$ the complexity of shallow networks that provide accuracy at least ϵ is*

$$N = \mathcal{O}(\epsilon^{-n/m}) \text{ and is the best possible.} \quad (5)$$

Theorem 2. *For $f \in W_m^{n,2}$ consider a deep network with the same compositonal architecture and with an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ which is infinitely differentiable, and not a polynomial. The complexity of the network to provide approximation with accuracy at least ϵ is*

$$N = \mathcal{O}((n-1)\epsilon^{-2/m}). \quad (6)$$

from [Poggio et al., 2017] ; see Sec 4.2 of [Poggio et al., 2017] for lower bound

Nonsmooth activation

A terse version of UAT

Proposition 2. *Let* $\sigma =: \mathbb{R} \to \mathbb{R}$ *be in* $\mathcal{C}^0$*, and not a polynomial. Then shallow networks are dense in* $\mathcal{C}^0$*.*

Nonsmooth activation

A terse version of UAT

Proposition 2. *Let $\sigma =: \mathbb{R} \to \mathbb{R}$ be in $\mathcal{C}^0$, and not a polynomial. Then shallow networks are dense in $\mathcal{C}^0$.*

Shallow vs. deep

Theorem 4. *Let f be a L-Lipshitz continuous function of n variables. Then, the complexity of a network which is a linear combination of ReLU providing an approximation with accuracy at least ϵ is*

$$N_s = \mathcal{O}\left(\left(\frac{\epsilon}{L}\right)^{-n}\right),$$

wheres that of a deep compositional architecture is

$$N_d = \mathcal{O}\left(\left(n-1)(\frac{\epsilon}{L}\right)^{-2}\right).$$

from [Poggio et al., 2017]

Width-bounded DNNs

Narrower than $n + 4$ is fine

Theorem 1 (Universal Approximation Theorem for Width-Bounded ReLU Networks). *For any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and any $\epsilon > 0$, there exists a fully-connected ReLU network $\mathscr{A}$ with width $d_m \leq n + 4$, such that the function $F_{\mathscr{A}}$ represented by this network satisfies*

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathscr{A}}(x)| \mathrm{d}x < \epsilon. \quad (3)$$

But no narrower than $n - 1$

Theorem 3. *For any continuous function $f : [-1, 1]^n \to \mathbb{R}$ which is not constant along any direction, there exists a universal $\epsilon^* > 0$ such that for any function F_A represented by a fully-connected ReLU network with width $d_m \leq n - 1$, the L^1 distance between f and F_A is at least ϵ^*:*

$$\int_{[-1,1]^n} |f(x) - F_A(x)| \mathrm{d}x \geq \epsilon^*. \quad (5)$$

from [Lu et al., 2017]; see also [Kidger and Lyons, 2019]

Width-bounded DNNs

Narrower than $n + 4$ is fine

Theorem 1 (Universal Approximation Theorem for Width-Bounded ReLU Networks). *For any Lebesgue-integrable function* $f : \mathbb{R}^n \to \mathbb{R}$ *and any* $\epsilon > 0$*, there exists a fully-connected ReLU network* $\mathscr{A}$ *with width* $d_m \leq n + 4$*, such that the function* $F_{\mathscr{A}}$ *represented by this network satisfies*

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathscr{A}}(x)| \mathrm{d}x < \epsilon. \tag{3}$$

But no narrower than $n - 1$

Theorem 3. *For any continuous function* $f : [-1, 1]^n \to \mathbb{R}$ *which is not constant along any direction, there exists a universal* $\epsilon^* > 0$ *such that for any function* F_A *represented by a fully-connected ReLU network with width* $d_m \leq n - 1$*, the* L^1 *distance between* f *and* F_A *is at least* ϵ^**:*

$$\int_{[-1,1]^n} |f(x) - F_A(x)| \mathrm{d}x \geq \epsilon^*. \tag{5}$$

from [Lu et al., 2017]; see also [Kidger and Lyons, 2019]

Deep vs. shallow still active area of research

Number one principle of DL

Fundamental theorem of DNNs

Universal approximation theorems

Number one principle of DL

Fundamental theorem of DNNs

Universal approximation theorems

Fundamental slogan of DL

Where there is a mapping, there is a NN... and fit it!

References i

[Cybenko, 1989] Cybenko, G. (1989). **Approximation by superpositions of a sigmoidal function.** *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

[Hornik, 1991] Hornik, K. (1991). **Approximation capabilities of multilayer feedforward networks.** *Neural Networks*, 4(2):251–257.

[Hornik et al., 1990] Hornik, K., Stinchcombe, M., and White, H. (1990). **Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks.** *Neural Networks*, 3(5):551–560.

[Kidger and Lyons, 2019] Kidger, P. and Lyons, T. (2019). **Universal approximation with deep narrow networks.** *arXiv:1905.08539*.

[Leshno et al., 1993] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). **Multilayer feedforward networks with a nonpolynomial activation function can approximate any function.** *Neural Networks*, 6(6):861–867.

[Lu et al., 2017] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). **The expressive power of neural networks: A view from the width.** In *Advances in neural information processing systems*, pages 6231–6239.

[Poggio et al., 2017] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. (2017). **Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review.** *International Journal of Automation and Computing*, 14(5):503–519.