Depth Separation in Learning via Representation Costs

Suzanna Parkinson, Ph.D. Candidate
University of Chicago
Committee on Computational and Applied Mathematics

https://arxiv.org/abs/2402.08808

Depth Separation: Gaps in behavior between neural networks at different depths

- Approximation Width: $\exists f$ you can approximate with many **fewer** units using deeper networks

  Pinkus 1999, Telgarsky (2016), Eldan & Shamir (2016), Daniely (2017), Safran et al. (2021)

- Representation Cost: $\exists f$ you can represent with much **smaller** parameters using deeper networks

  Ongie et al. (2019)

How does this translate to gaps in **generalization** & **learning**?

What do we mean by **learning**?

- True underlying distribution $\mathbf{x} \sim \mathscr{D}$, $y = f(\mathbf{x})$
- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
- Use a **learning rule** $\mathscr{A}(S)$ to choose a model from a **model class** based on training samples

Ex: Try to minimize ***sample loss:*** $\mathscr{A}(S) \in \arg\min_{g \in \mathscr{G}} \mathscr{L}_S(g) := \frac{1}{m}\sum_{i=1}^{m}\left(g(\mathbf{x}_i) - y_i\right)^2$

- Want small **generalization error/expected loss**

$$\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S)) := \mathbb{E}_{\mathbf{x}\sim\mathscr{D}}\left[\left(\mathscr{A}(S)(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \|\mathscr{A}(S) - f\|_{L_2(\mathscr{D})}$$

  - Only get **finitely many training samples**
  - Using a **limited model class**

$\Longrightarrow$Best we can hope for is to be **Probably Approximately Correct (PAC)**.

Probably Approximately Correct (PAC) Learning

Definition: *The output of a learning rule $\mathscr{A}$ trained with m samples is (ε, δ)-**Probably Approximately Correct** if with probability $1 - \delta$ over the training samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, the **generalization error** is less than ε:*

$$\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S)) < \varepsilon\,.$$

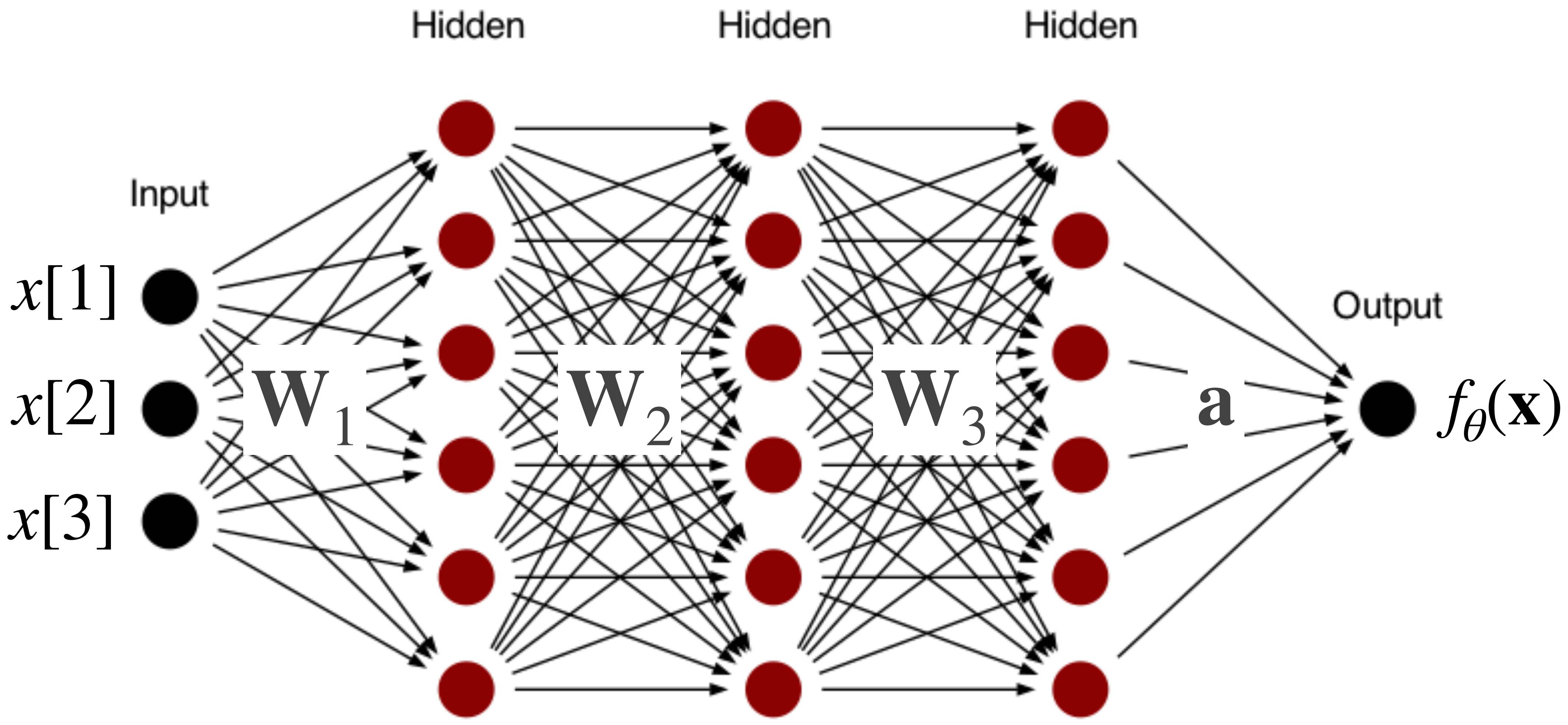
If our learning rule $\mathscr{A}$ gives a model that is (ε, δ)-**Probably Approximately Correct** using $m(\varepsilon, \delta)$ samples, then we say that we can **learn** with **sample complexity** $m(\varepsilon, \delta)$.

Generalization vs. **Approximation** vs. **Estimation** Error

$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\substack{\text{Generalization}\\ \text{Error}\\ \text{(expected loss)}}} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)}_{\substack{\text{Approximation}\\ \text{Error}}} + \underbrace{2 \sup_{g \in \mathscr{G}} | \mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g) |}_{\substack{\text{Estimation}\\ \text{Error}}}$$

- **Approximation Error:** Need existence of **one** good approximator g in model class. *Hornik (1991), Shen et al. (2022)*
- **Estimation Error:** Control via the **size** of model class, as measured by VC-dimension, **Rademacher complexity**, metric entropy, etc. *Vapnik & Chervonenkis (1971), Bartlett & Mendelson (2001), Neyshabur et al. (2015).*

Neural Networks



$$\theta = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{L-1}, \mathbf{a}\right)$$

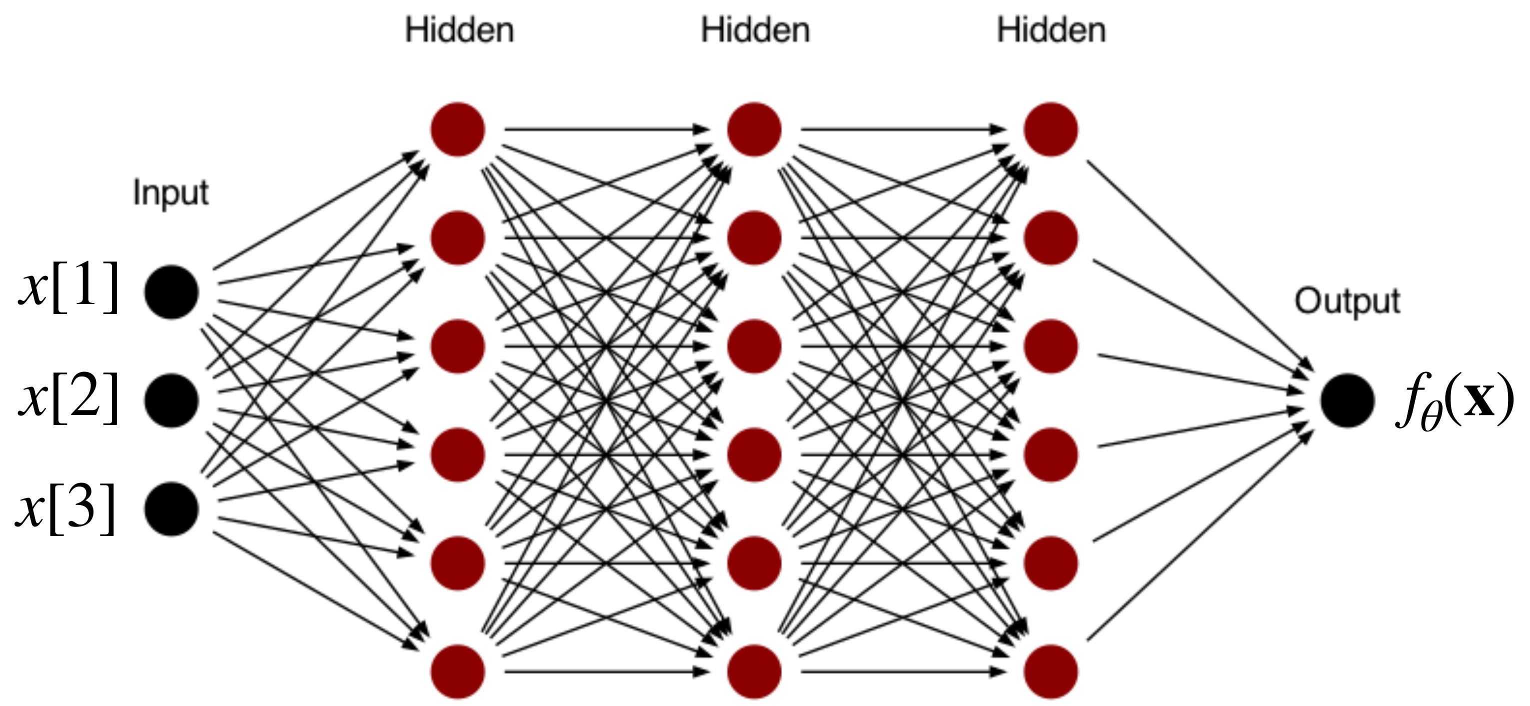
$$f_\theta(\mathbf{x}) = \mathbf{a}^\top \sigma\left(\mathbf{W}_{L-1} \cdot \sigma\left(\cdots \sigma\left(\mathbf{W}_2 \sigma\left(\mathbf{W}_1 \mathbf{x}\right)\right)\right)\right)$$

$$\sigma(x) = \mathrm{ReLU}(x)$$

Function Space Perspective

Parameter Space Cost

$$\hat{\theta}_S \in \arg\min_{\theta} \ \mathscr{L}_S(f_\theta) + \lambda C_L(\theta) \text{ where } C_L(\theta) = \frac{1}{L}\left(\sum_{\ell=1}^{L-1} \|\mathbf{W}_\ell\|_F^2 + \|\mathbf{a}\|_2^2\right)$$



$$\hat{f}_S \in \arg\min_{g \in \mathcal{N}_L} \mathscr{L}_S(g) + \lambda R_L(g) \text{ where } \underbrace{R_L(g) = \inf_{\theta} C_L(\theta) \text{ s.t. } f_\theta = g}_{\textbf{Representation Cost}}$$

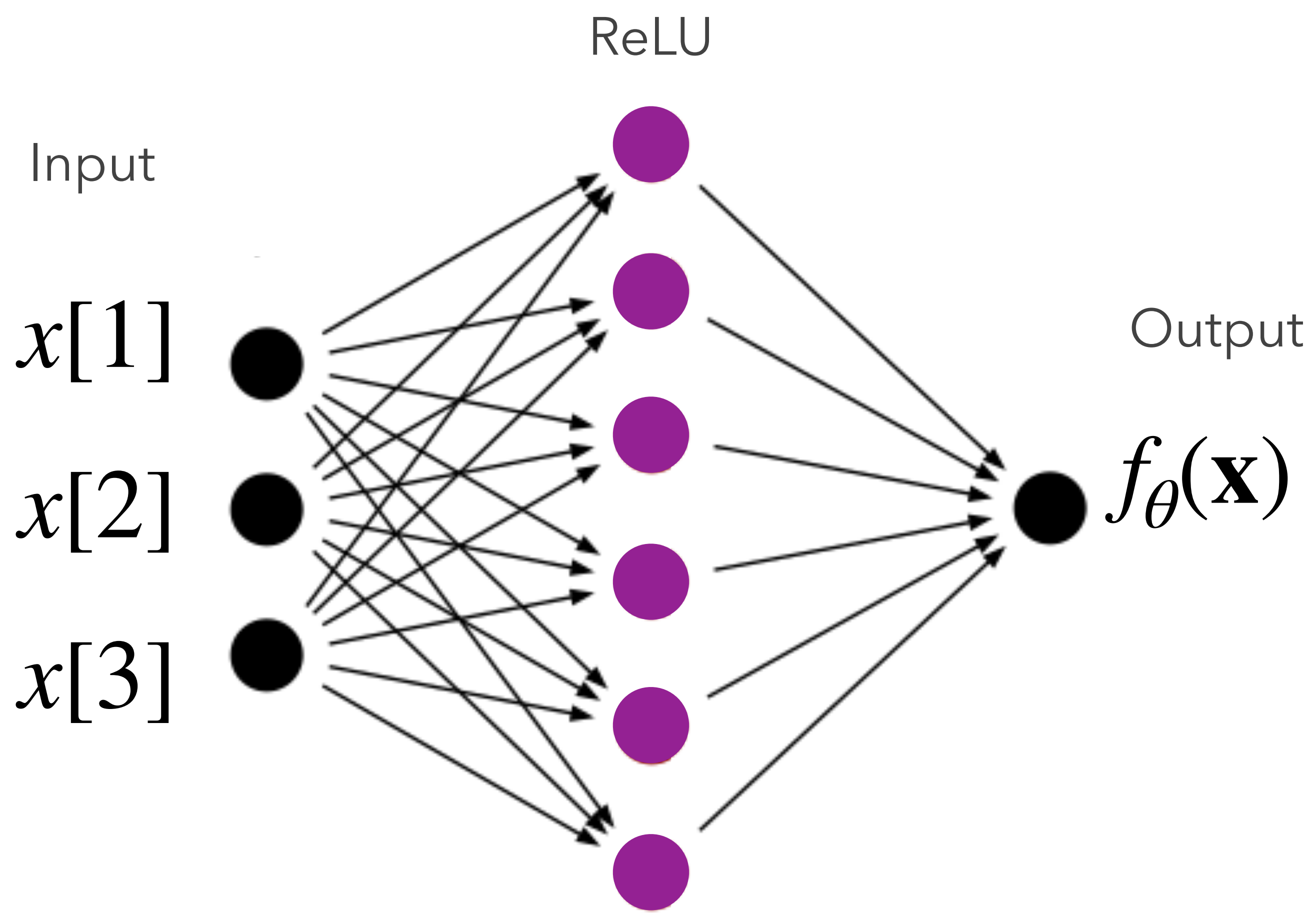
Can understanding representation costs across different depths help us understand gaps in **learning/generalization** capabilities?

Are **deeper** neural networks
better at **learning**?

Are **depth-2** or **depth-3** neural networks better at **learning**?

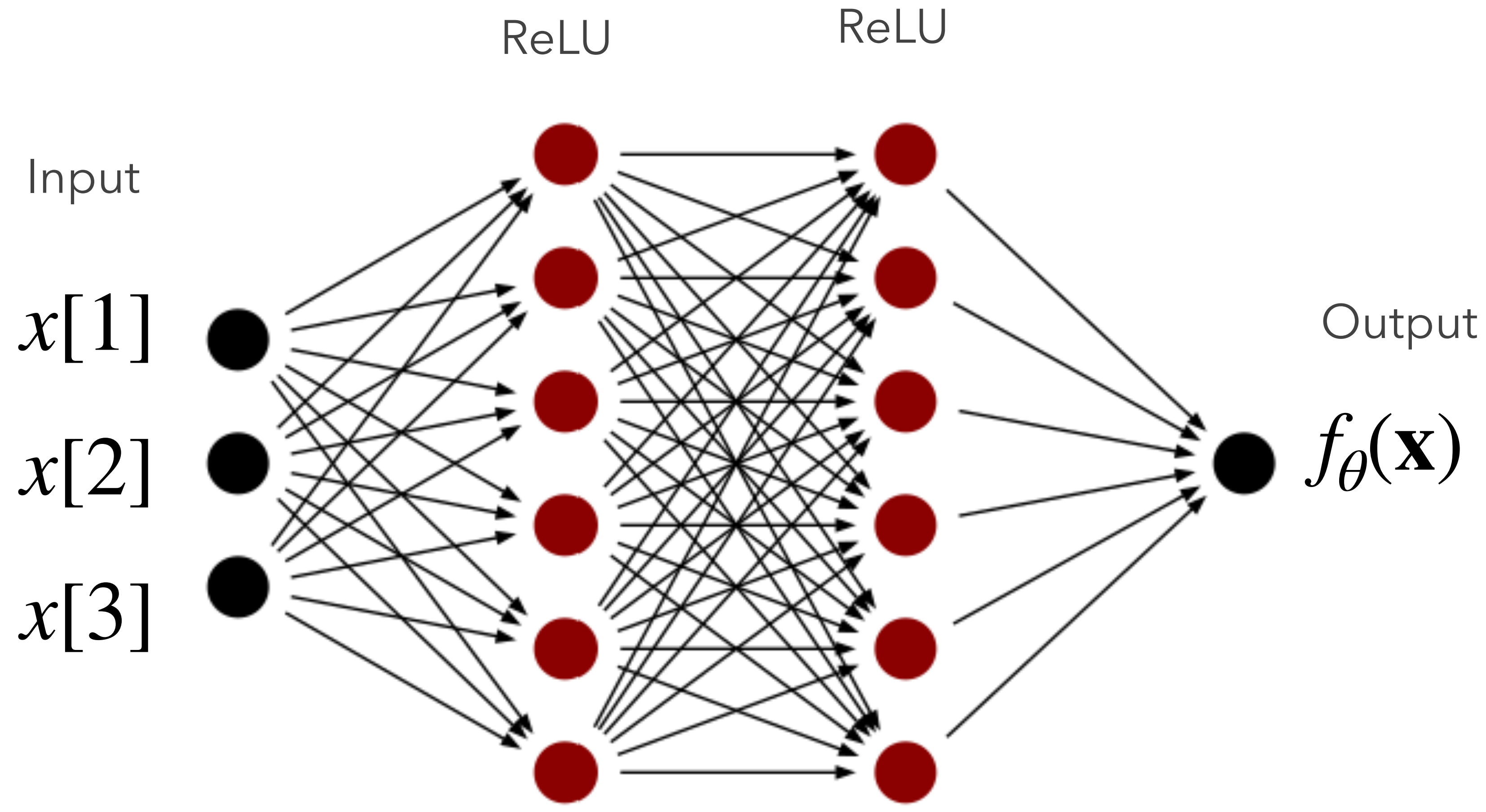
First Pass Intuition

Depth-2 ReLU Network



- **Universal approximator** of continuous functions with **arbitrary width**. *Hornik (1991)*
- **Fewer parameters** = smaller model class

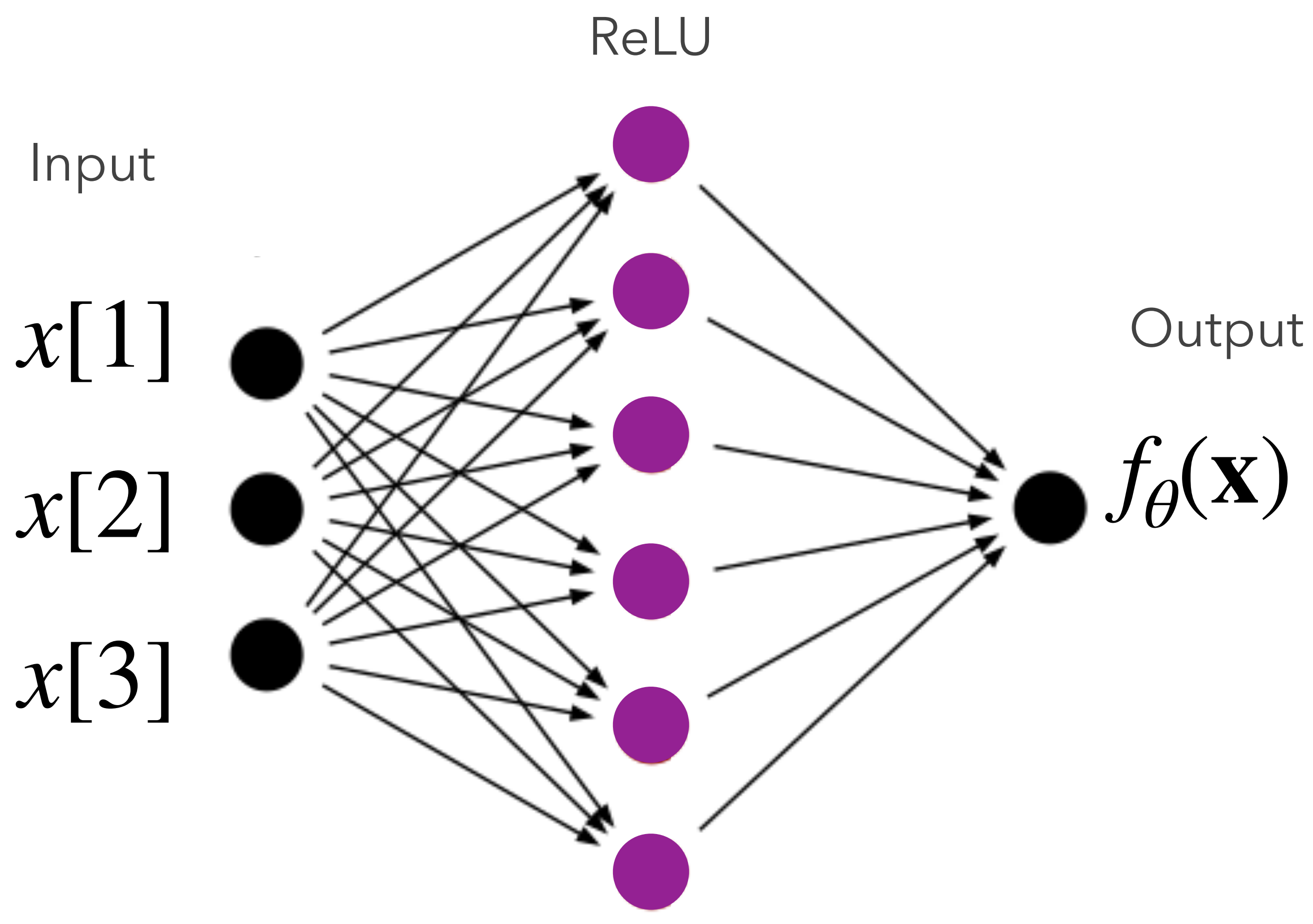
Depth-3 ReLU Network



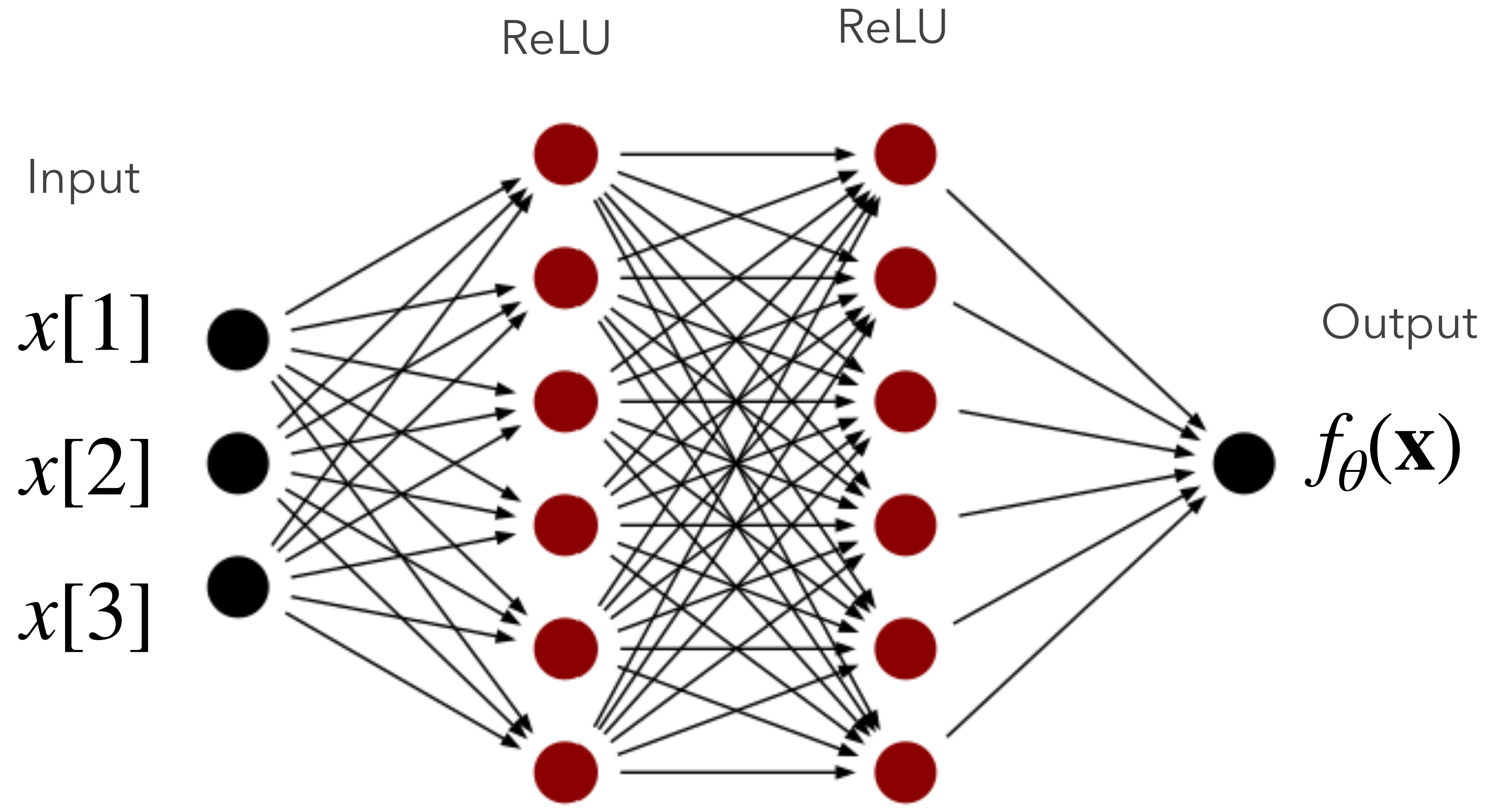
- **Universal approximator** of continuous functions with **arbitrary width**. *Hornik (1991)*
- **More parameters** = bigger model class

Depth Separation in Width to Approximate

Depth-2 ReLU Network



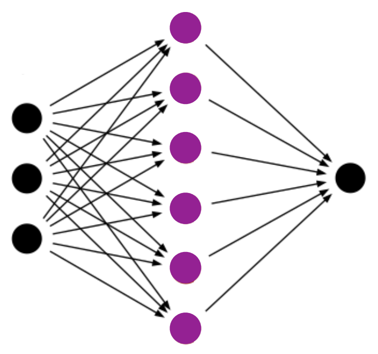
Depth-3 ReLU Network

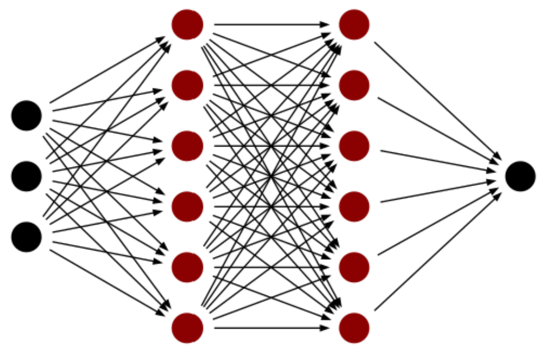


$\exists f_d : \mathbb{R}^d \to \mathbb{R}$ that…

- Requires $\geq 2^d$ **width** to **approximate** to within a fixed ε with **depth 2**
- **Approximable** with **poly**(d, ε^{-1}) **width** with **depth 3**

Pinkus 1999, Telgarsky (2016), Eldan & Shamir (2016), Daniely (2017), Safran et al. (2021)





What if we measure model **size** in terms of **norm** of parameters instead of **number** of parameters?

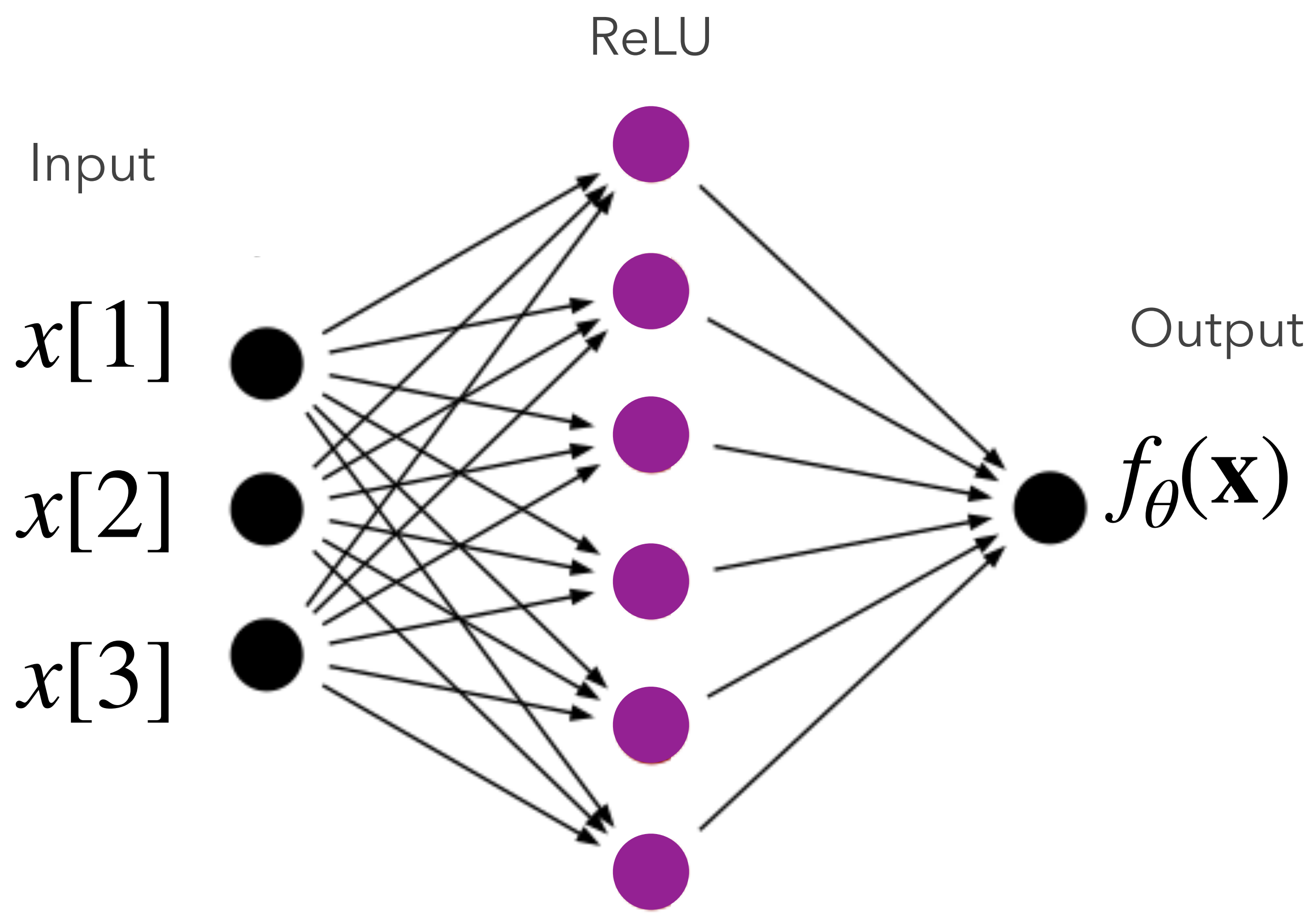
Bartlett 1996, Neyshabur, Tomioka & Srebro 2015

For valid generalization, the size of the weights is more important than the size of the network

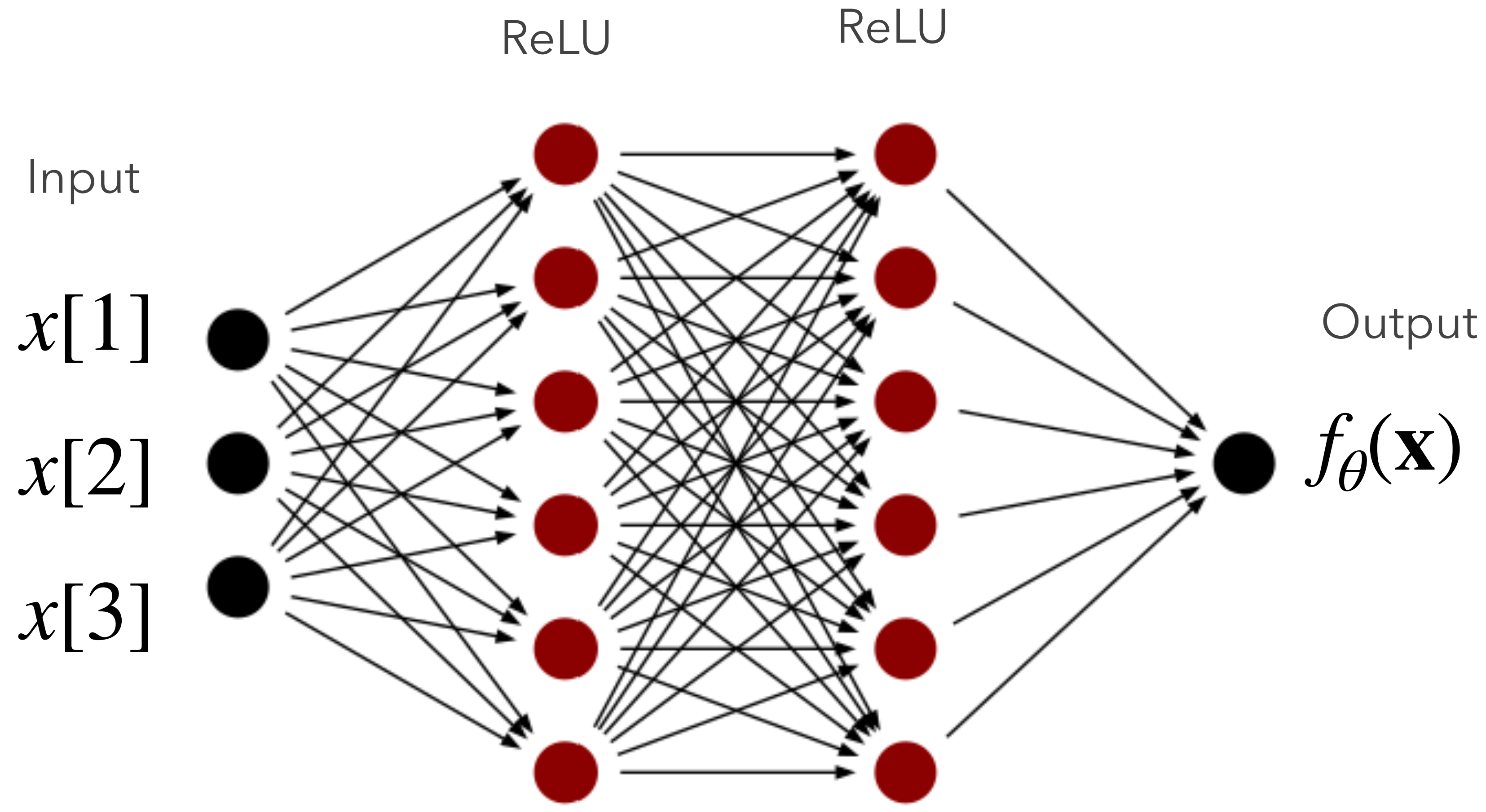
Peter L. Bartlett
Department of Systems Engineering
Research School of Information Sciences and Engineering
Australian National University
Canberra, 0200 Australia
Peter.Bartlett@anu.edu.au

Depth Separation in Representation Cost

Depth-2 ReLU Network



Depth-3 ReLU Network

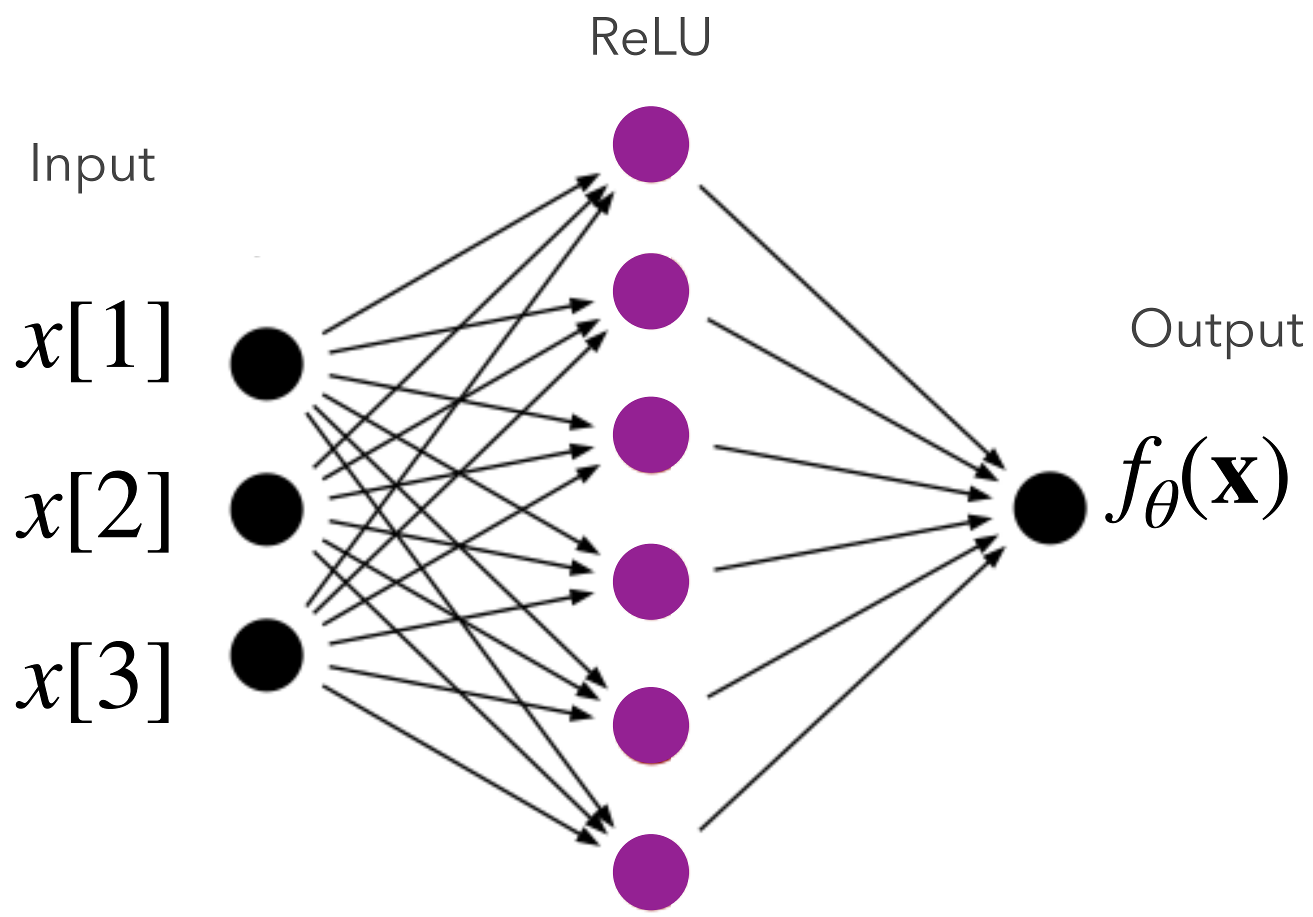


$\exists f : \mathbb{R}^d \to \mathbb{R}$ for which $R_2(f) \gg R_3(f)$

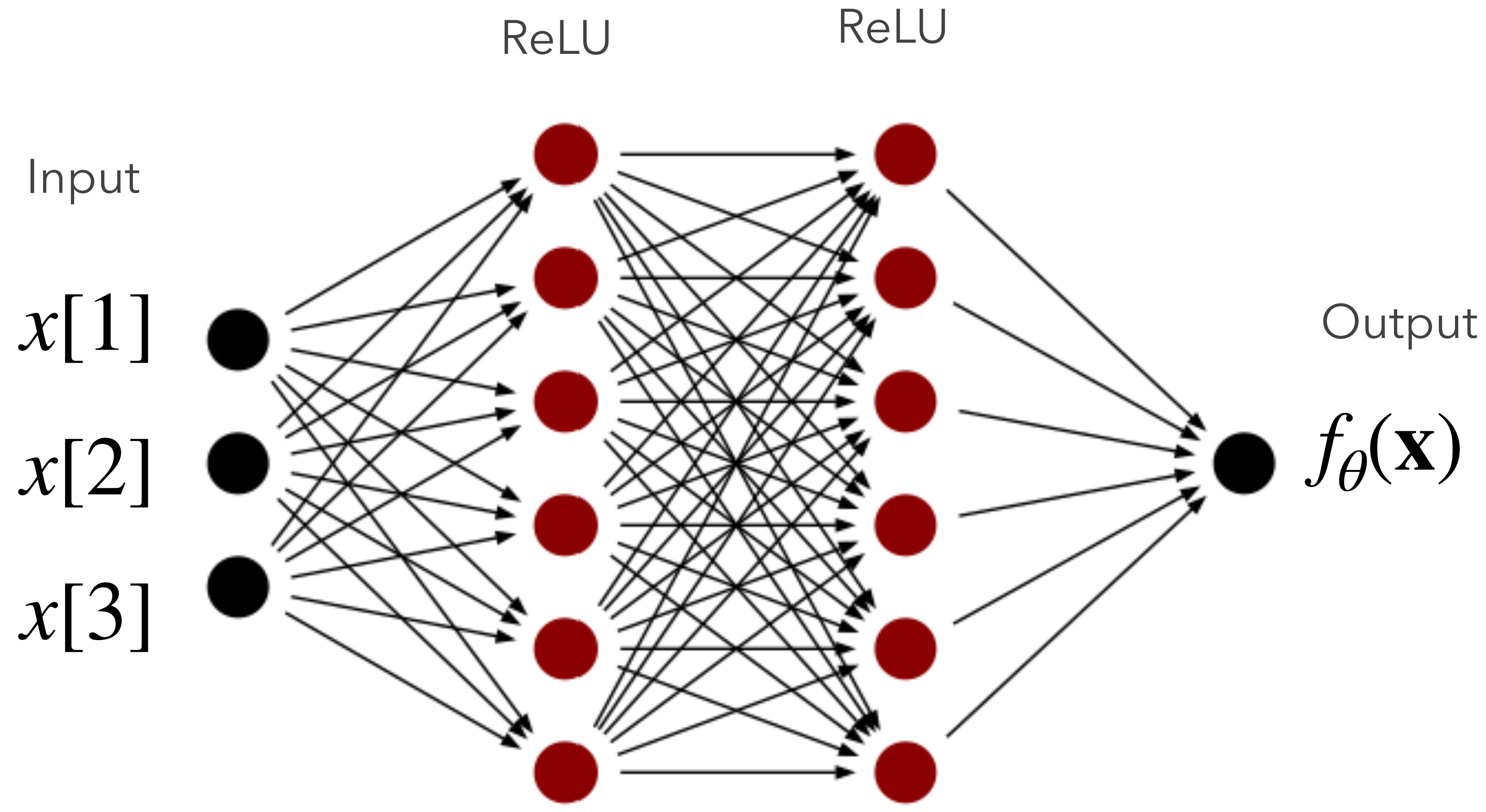
Ongie et al. 2019

Depth Separation in Learning?

Depth-2 ReLU Network



Depth-3 ReLU Network



$\exists f_d : \mathbb{R}^d \to \mathbb{R}$ and and distributions $\mathbf{x} \sim \mathscr{D}_d$ on $\mathbb{R}^d$ that…

- Require $2^{\omega(d)}$ **samples** to **learn** to within a fixed ε and δ with **depth 2**

$$\mathscr{A}_2^\lambda(S) = \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_2(g)$$

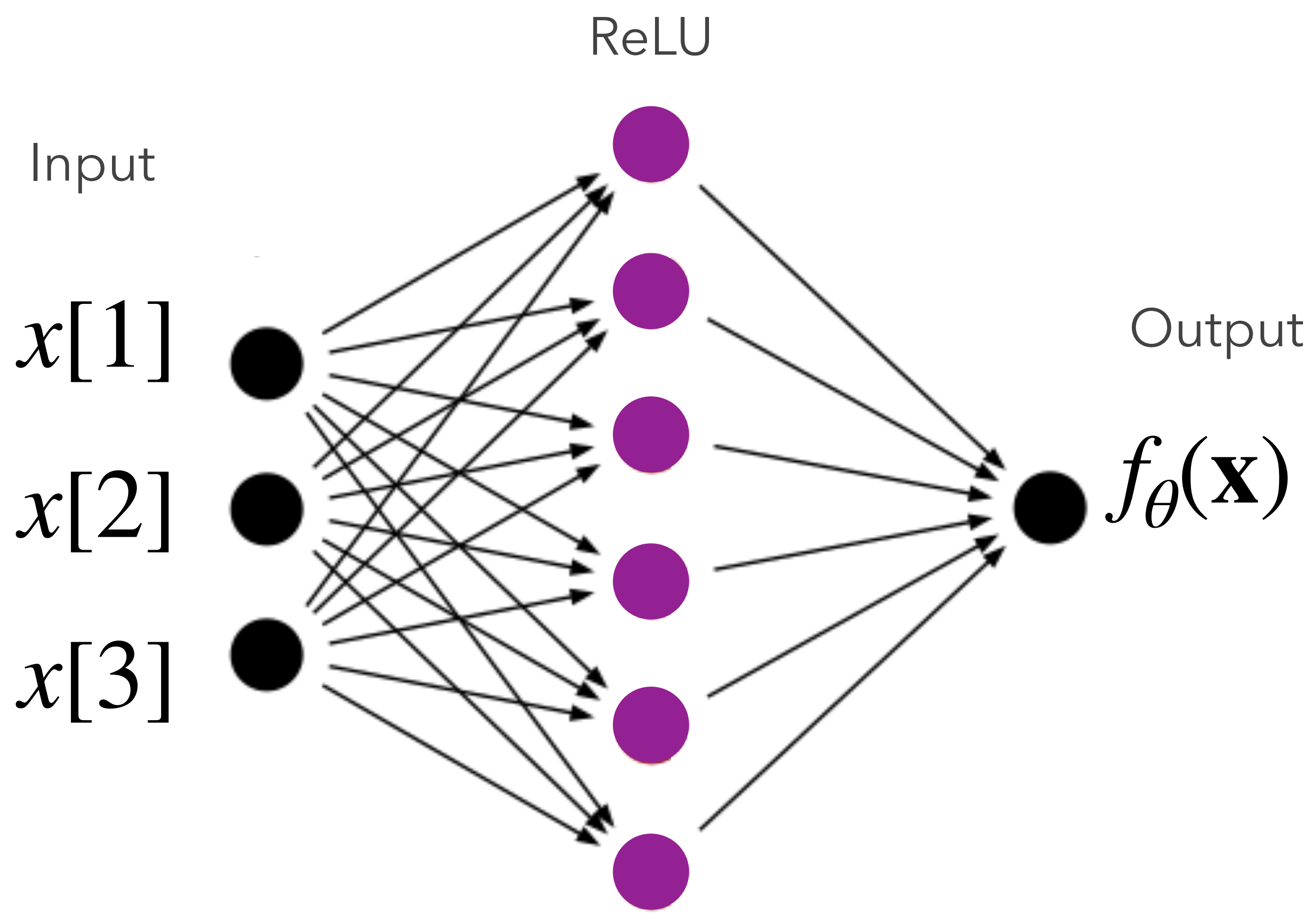
- Only need **poly**$(d, \varepsilon^{-1}, \delta^{-1})$ **samples** to **learn** with **depth 3**

$$\mathscr{A}_3^\lambda(S) = \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_3(g)$$

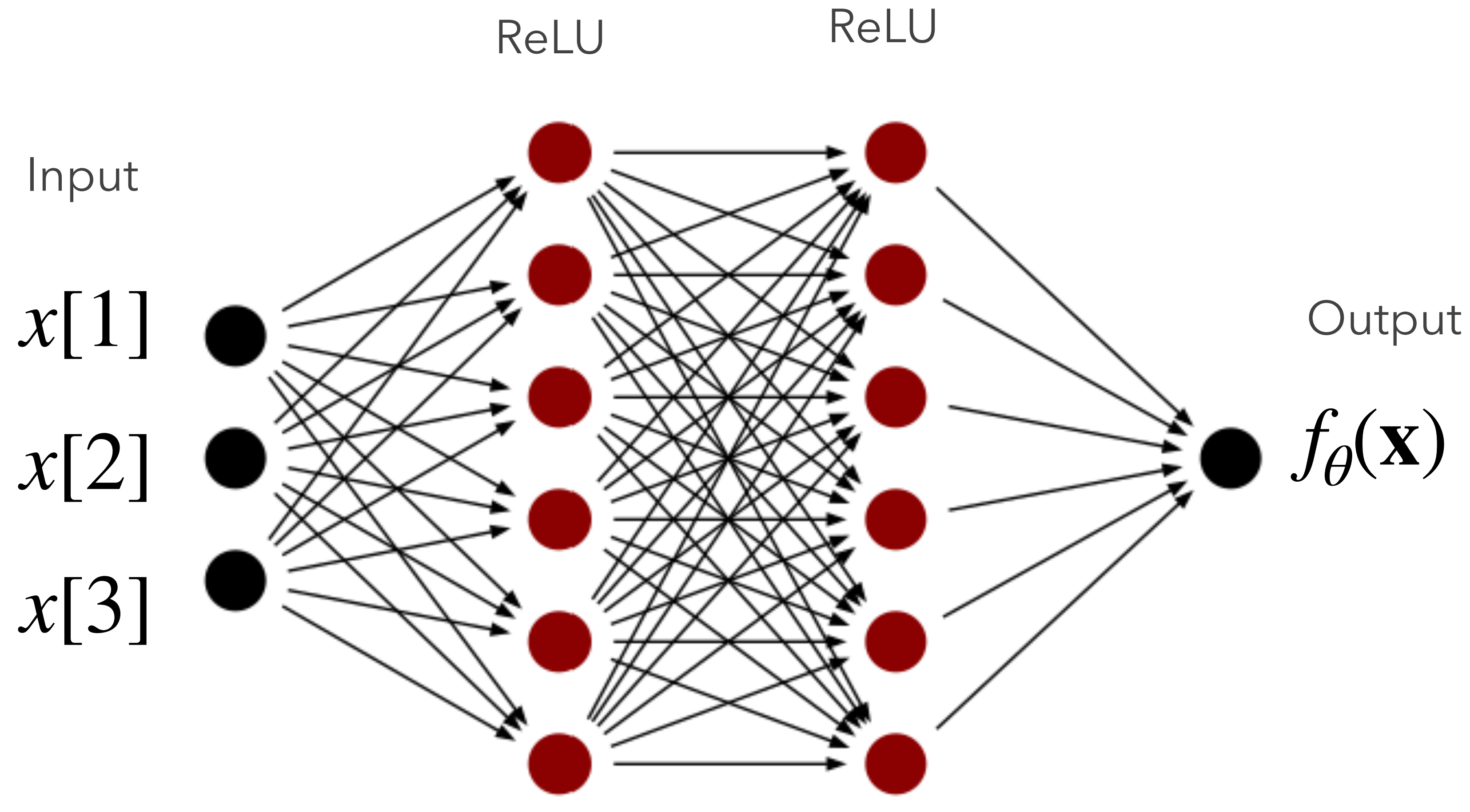
Pinkus 1999, Telgarsky (2016), Eldan & Shamir (2016), Daniely (2017), Safran et al. (2021)

Reverse Depth Separation in Learning?

Depth-2 ReLU Network



Depth-3 ReLU Network



$\exists f_d : \mathbb{R}^d \to \mathbb{R}$ and and distributions $\mathbf{x} \sim \mathscr{D}_d$ on $\mathbb{R}^d$ that…

- Only need **poly**$(d, \varepsilon^{-1}, \delta^{-1})$ **samples** to **learn** with **depth 2**

$$\mathscr{A}_2^\lambda(S) = \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_2(g)$$

- Require $2^{\omega(d)}$ **samples** to **learn** to within a fixed ε with **depth 3**

$$\mathscr{A}_3^\lambda(S) = \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_3(g)$$

Pinkus 1999, Telgarsky (2016), Eldan & Shamir (2016), Daniely (2017), Safran et al. (2021)

Understanding **representation costs** can help us answer these questions about the role of **depth**

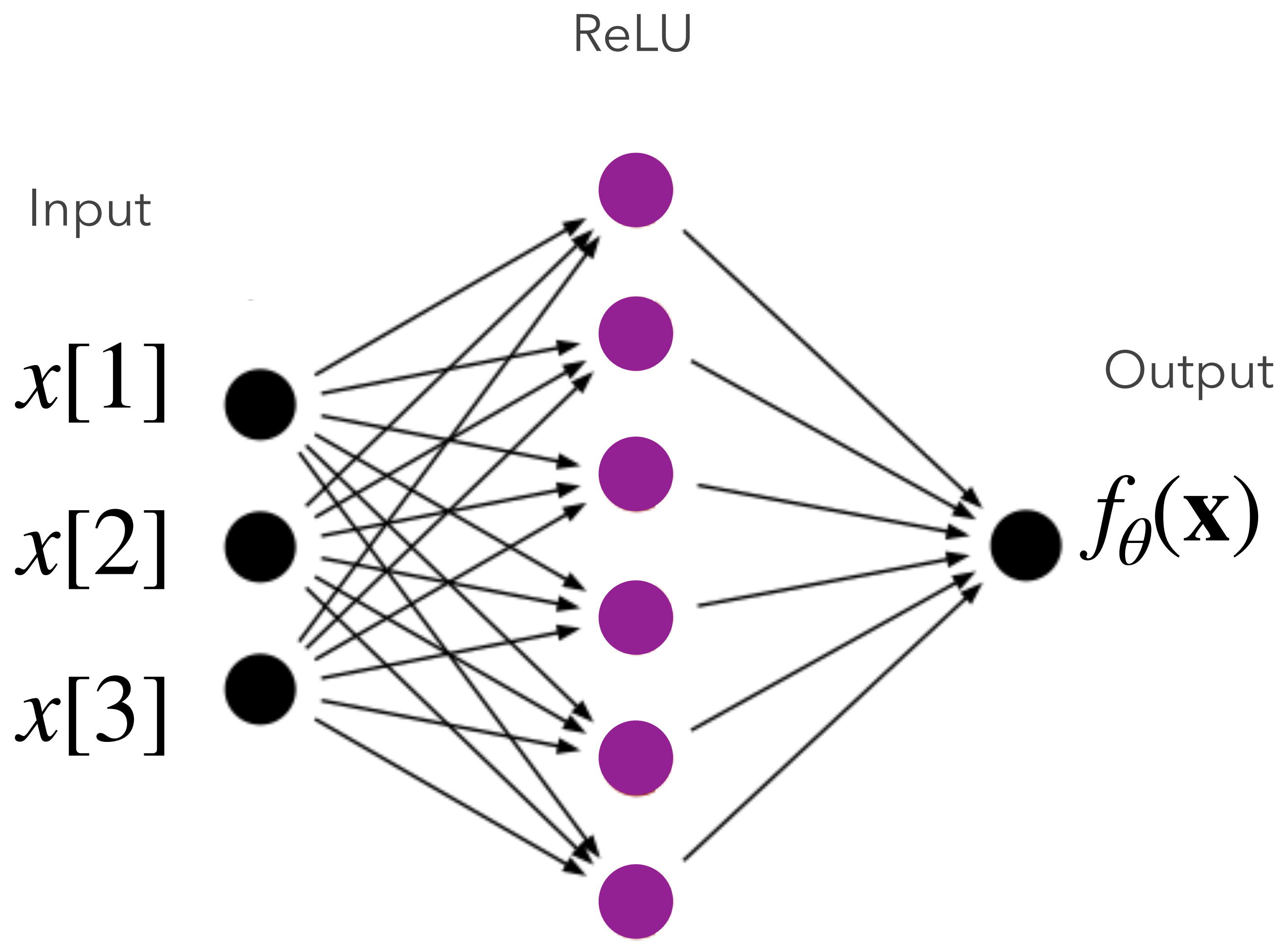
For valid generalization, the size of the weights is more important than the size of the network

Peter L. Bartlett
Department of Systems Engineering
Research School of Information Sciences and Engineering
Australian National University
Canberra, 0200 Australia
Peter.Bartlett@anu.edu.au

Depth Separation: $\exists f_d$ that is **"hard"** with **depth 2** but **"easy"** with **depth 3**

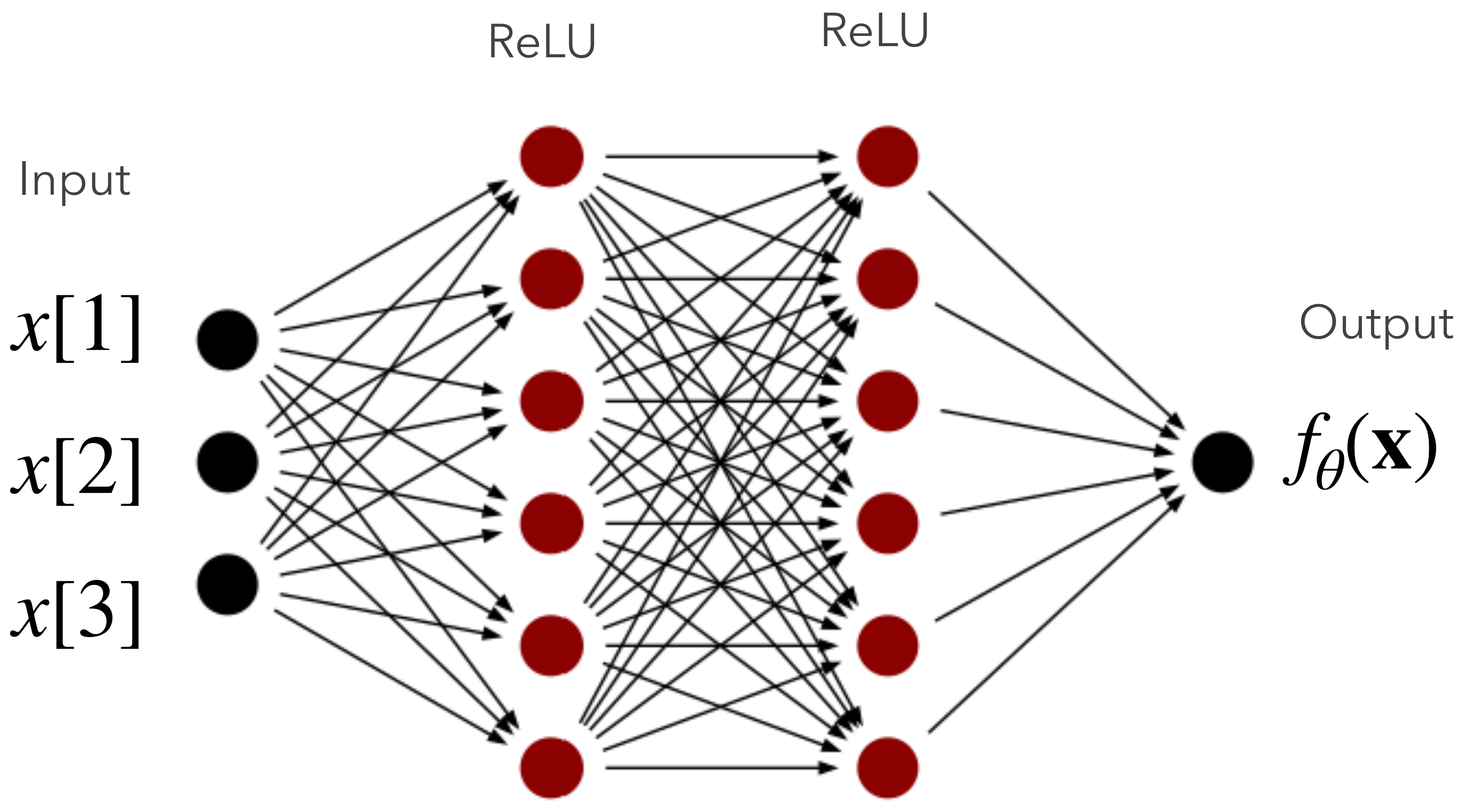
Key: Choose f_d so that…

Large **representation cost** with **depth 2**



Expensive

Small **representation cost** with **depth 3**



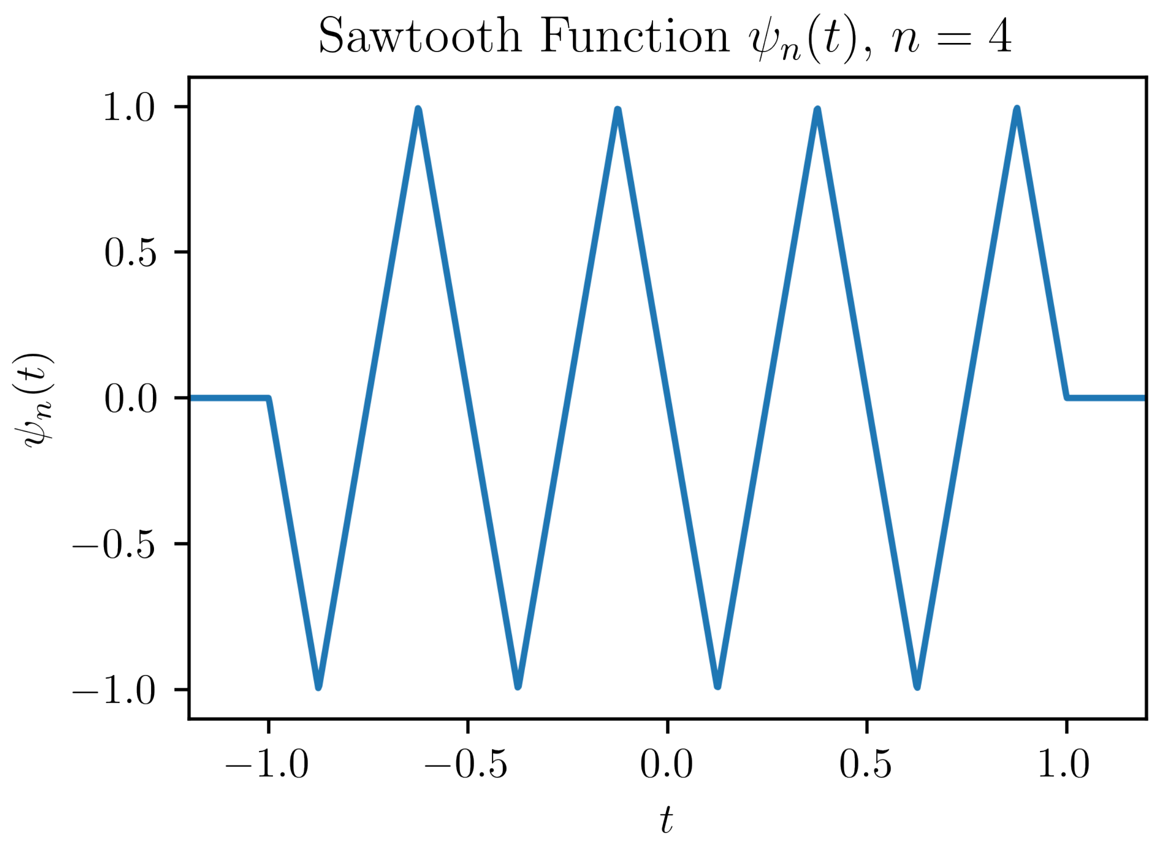
Cheap

Depth Separation: $\exists f_d$ that is **"hard"** with **depth 2** but **"easy"** with **depth 3**

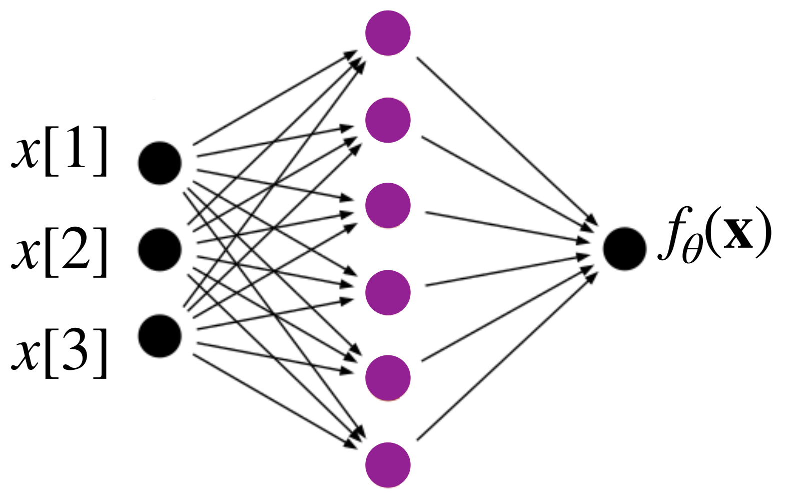
Proof Sketch:

- $\mathbf{x} \sim \mathrm{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$, $f(\mathbf{x}) = \psi_{3d}\left(\sqrt{d}\langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \rangle\right)$

  Slight modification of Daniely (2017) construction for separation in width to approximate

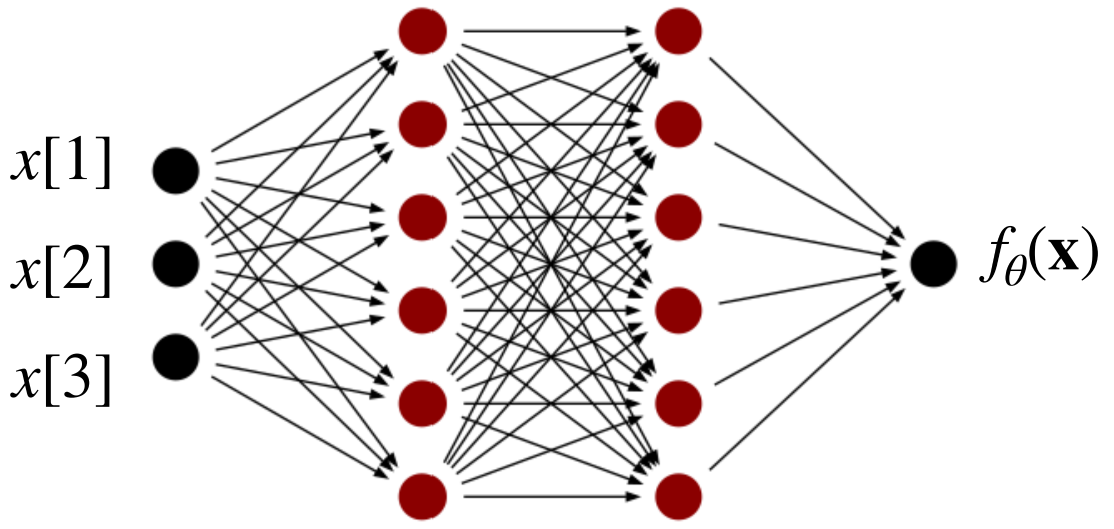


- Daniely showed that **depth 2** networks need to be very wide to approximate functions that are compositions of a function that is **very non-polynomial** with an **inner-product**



Expensive

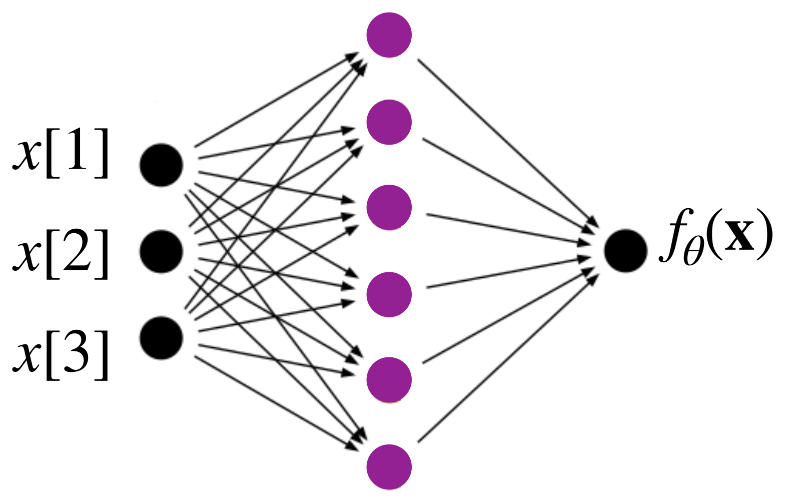
- Naturally approximated by a **depth 3** network…
  - The inner product can be approximated with first hidden layer
  - Sawtooth function can be expressed exactly with second hidden layer



Cheap

Depth Separation: $\exists f_d$ that is **"hard"** with **depth 2** but **"easy"** with **depth 3**

Proof Sketch: ***"Hard"*** *with* $\mathcal{A}_2^\lambda(S) \in \arg\min_{g \in \mathcal{N}_2} \mathcal{L}_S(g) + \lambda R_2(g)$



Expensive

- **Lemma:** f can be ε-approximated with **depth 2** with $R_2 \leq C$

  $\implies f$ can be ε-approximated with **depth 2** with width $\lesssim \dfrac{C}{\varepsilon^2}$

- **Converse:** f_d requires width $> 2^{\Omega(d)}$ to ε-approximate with **depth 2**

  $\implies f_d$ requires $R_2 > 2^{\Omega(d)}$ to ε-approximate with **depth 2**

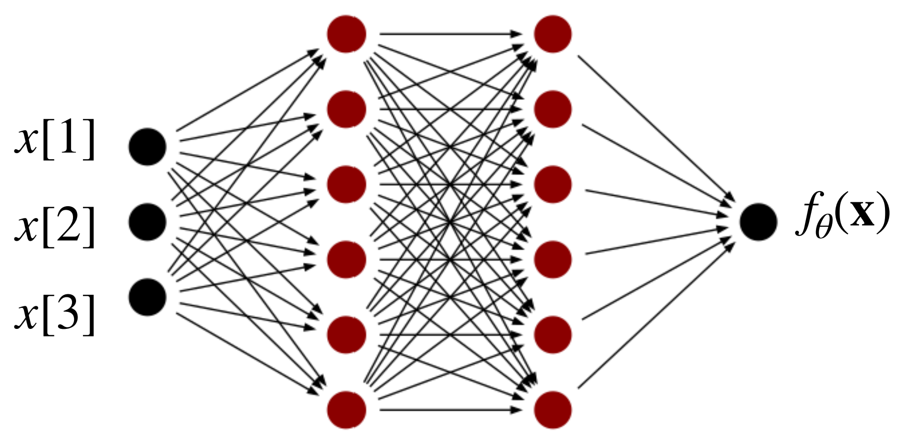
- With probability $1 - \delta$, a **depth 2** interpolant of the samples $\hat{f}$ exists with $R_2(\hat{f}) \leq O(|S|^2)$

- $R_2(\mathcal{A}_2^\lambda(S)) \leq R_2(\hat{f}) = O(|S|^2)$

- So $\mathcal{A}_2^\lambda(S)$ is a bad approximation of f_d unless $|S| \geq 2^{\Omega(d)}$

Depth Separation: $\exists f_d$ that is **"hard"** with **depth 2** but **"easy"** with **depth 3**

Proof Sketch: ***"Easy"*** *with* $\mathscr{A}_3^{\lambda}(S) \in \arg\min_{g \in \mathscr{N}_3} \mathscr{L}_S(g) + \lambda R_3(g)$



Cheap

- $\exists f_\varepsilon$ of depth **3** with $\mathscr{L}_{\mathscr{D}}(f_\varepsilon) \le \varepsilon/2$ and $R_3(f_\varepsilon) \le \mathrm{poly}(d)$

- If you choose λ in a reasonable way, you get $R_3(\mathscr{A}_3^{\lambda}(S)) \le R_3(f_\varepsilon) \le \mathrm{poly}(d)$

$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}_3(S))}_{\text{Generalization Error (expected loss)}} \le \underbrace{\inf_{R_3(g) \le \mathrm{poly}(d)} \mathscr{L}_{\mathscr{D}}(g)}_{\text{Approximation Error}} + \underbrace{2 \sup_{R_3(g) \le \mathrm{poly}(d)} |\mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)|}_{\text{Estimation Error}}$$

- **Rademacher complexity analysis:** If $R_3(g) \le \mathrm{poly}(d)$, then with probability $1 - \delta$,

$$|\mathscr{L}_{\mathscr{D}}(g) - \mathscr{L}_S(g)| \le \mathrm{poly}(d)\sqrt{\frac{\log 1/\delta}{|S|}}$$

Neyshabur et al. 2015

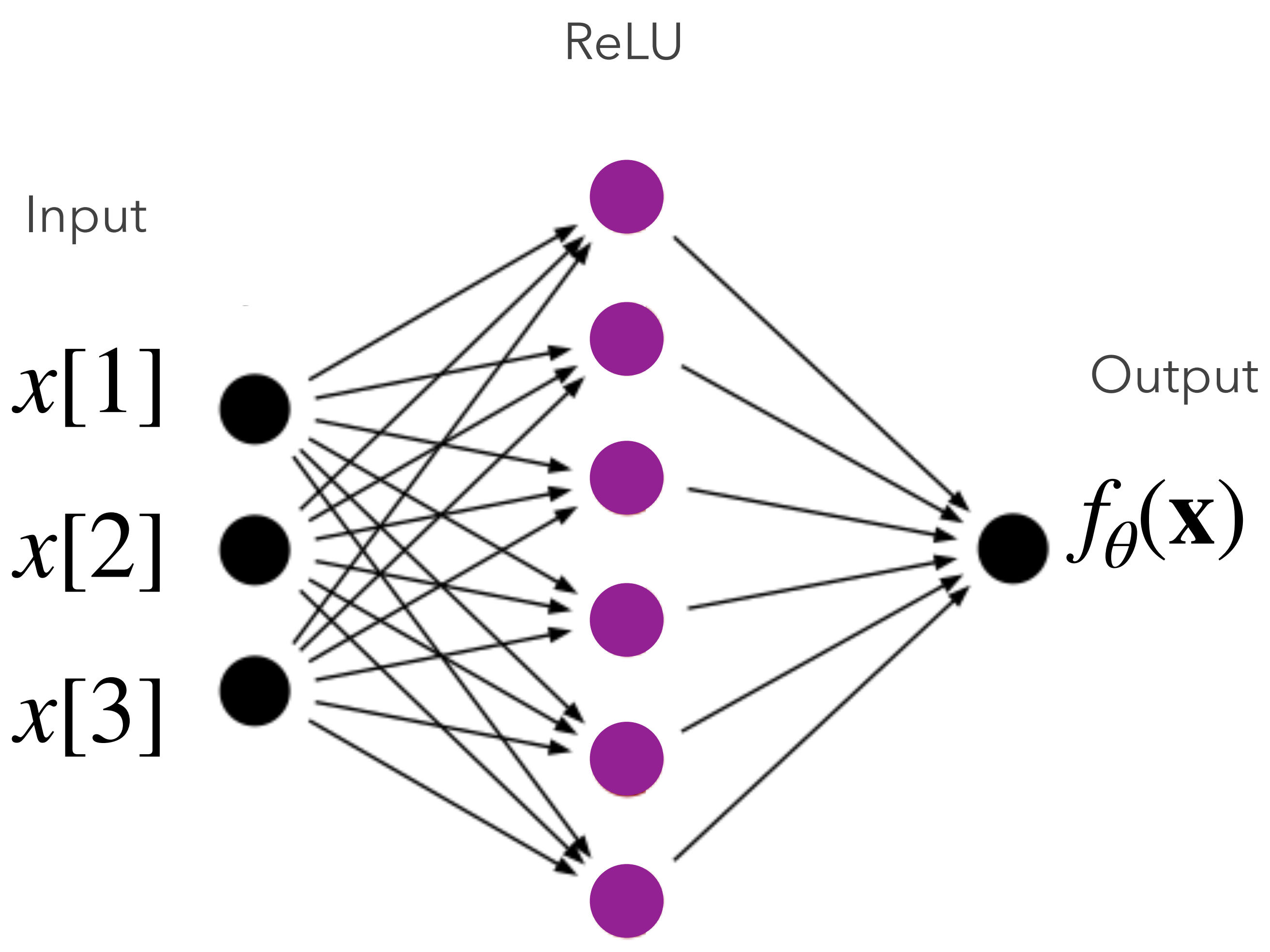
- Therefore, $\mathscr{L}_{\mathscr{D}}(\mathscr{A}_3^{\lambda}(S)) \le \varepsilon$ with high probability as long as

$$|S| = \mathrm{poly}(d)\varepsilon^{-2}\log(1/\delta)$$

No Reverse Depth Separation: f_d "easy" with **depth 2** $\Longrightarrow$ "easy" with **depth 3**

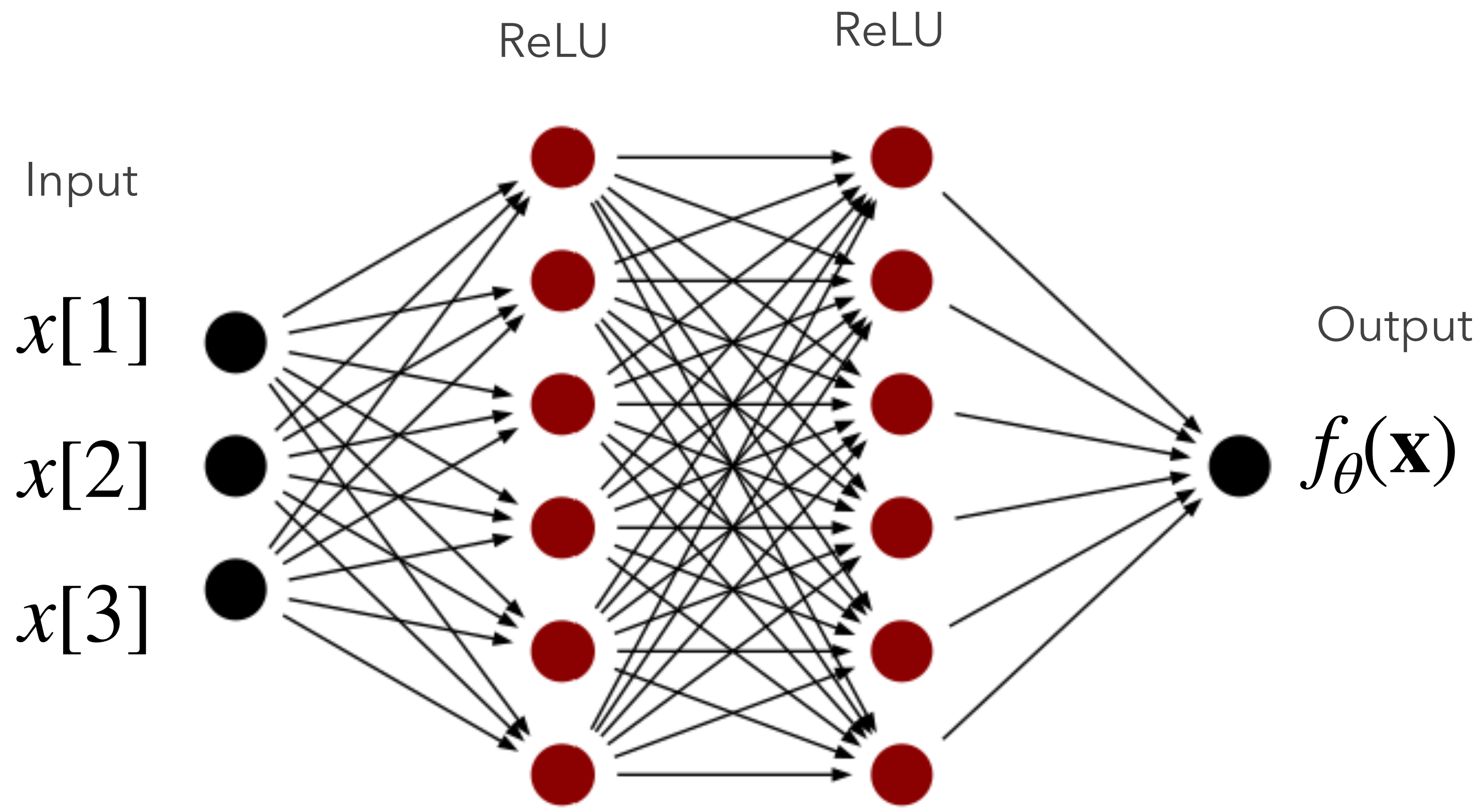
Key:

Small **representation cost** with **depth 2**



Cheap

$\Longrightarrow$ Small **representation cost** with depth **3**

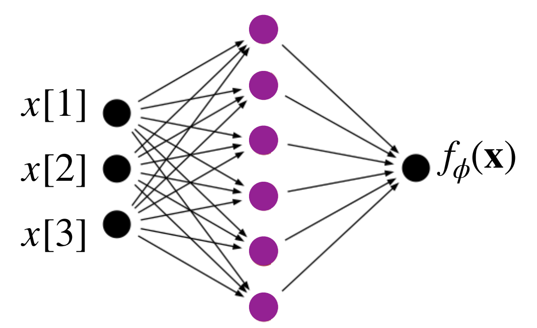


Cheap

No Reverse Depth Separation: f_d "easy" with depth 2 $\Longrightarrow$ "easy" with depth 3

Proof Sketch:

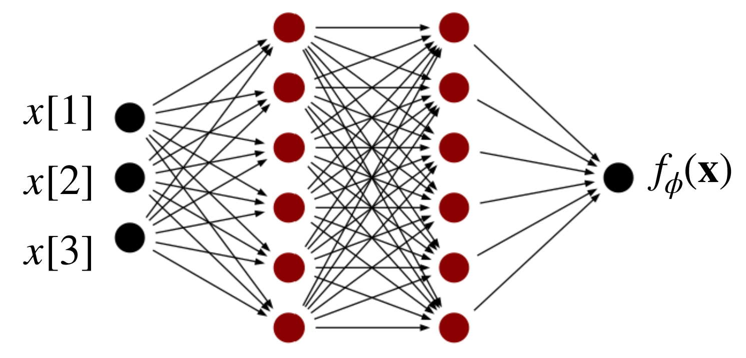
- If $\mathscr{A}_2^\lambda(S)$ learns with polynomial sample complexity, $\exists f_\varepsilon$ of **depth 2** such that $\mathscr{L}_{\mathscr{D}}(f_\varepsilon) \leq \varepsilon/2$ and $R_2(f_\varepsilon) \leq \mathrm{poly}(d, \varepsilon^{-1})$.



Cheap

$\Downarrow$

- $R_3(f_\varepsilon) = O\left(d + R_2(f_\varepsilon)\right) \leq \mathrm{poly}(d, \varepsilon^{-1})$



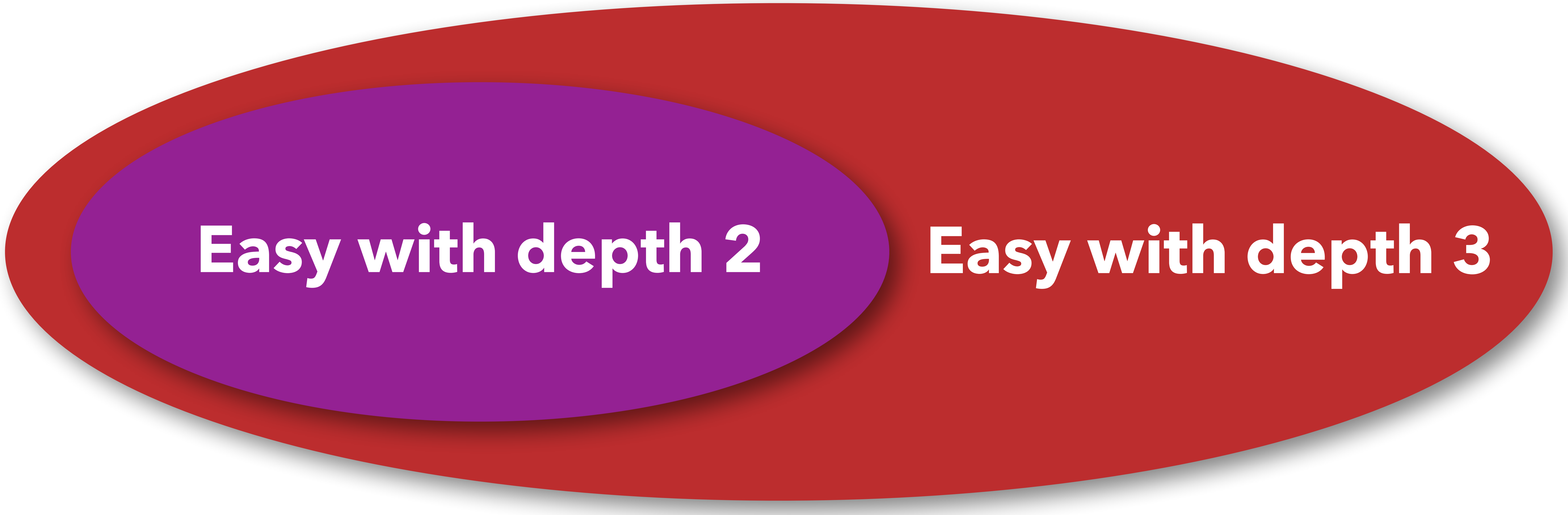
Cheap

- If you choose λ in a reasonable way, you get $R_3(\mathscr{A}_3^\lambda(S)) \leq R_3(f_\varepsilon) \leq \mathrm{poly}(d, \varepsilon^{-1})$

$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}_3(S))}_{\substack{\textbf{Generalization Error} \\ \textbf{(expected loss)}}} \leq \underbrace{\inf_{R_3(g) \leq \mathrm{poly}(d, \varepsilon^{-1})} \mathscr{L}_{\mathscr{D}}(g)}_{\textbf{Approximation Error}} + \underbrace{2 \sup_{R_3(g) \leq \mathrm{poly}(d, \varepsilon^{-1})} |\mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)|}_{\textbf{Estimation Error}}$$

- Therefore, using similar **Rademacher complexity analysis**, $\mathscr{L}_{\mathscr{D}}(\mathscr{A}_3^\lambda(S)) \leq \varepsilon$ with high probability as long as $|S| = \mathrm{poly}(d, \varepsilon^{-1})\log(1/\delta)$.

Functions that are **"easy" to learn** with **depth 2** networks form a **strict subset** of functions that are **"easy" to learn** with **depth 3** networks.



We've assumed that we're **(nearly) minimizing** our objective. How does the **loss-landscape** affect learning at different depths?

Thank you!

Greg Ongie
Marquette University

Rebecca Willett
University of Chicago

Ohad Shamir
Weizmann Institute of Science

Nati Srebro
Toyota Technical Institute at Chicago

https://arxiv.org/abs/2402.08808

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2140001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.