Depth Separation in Learning via Representation Costs

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

Brigham Young University Applied Math Seminar
February 15, 2024

https://arxiv.org/abs/2402.08808

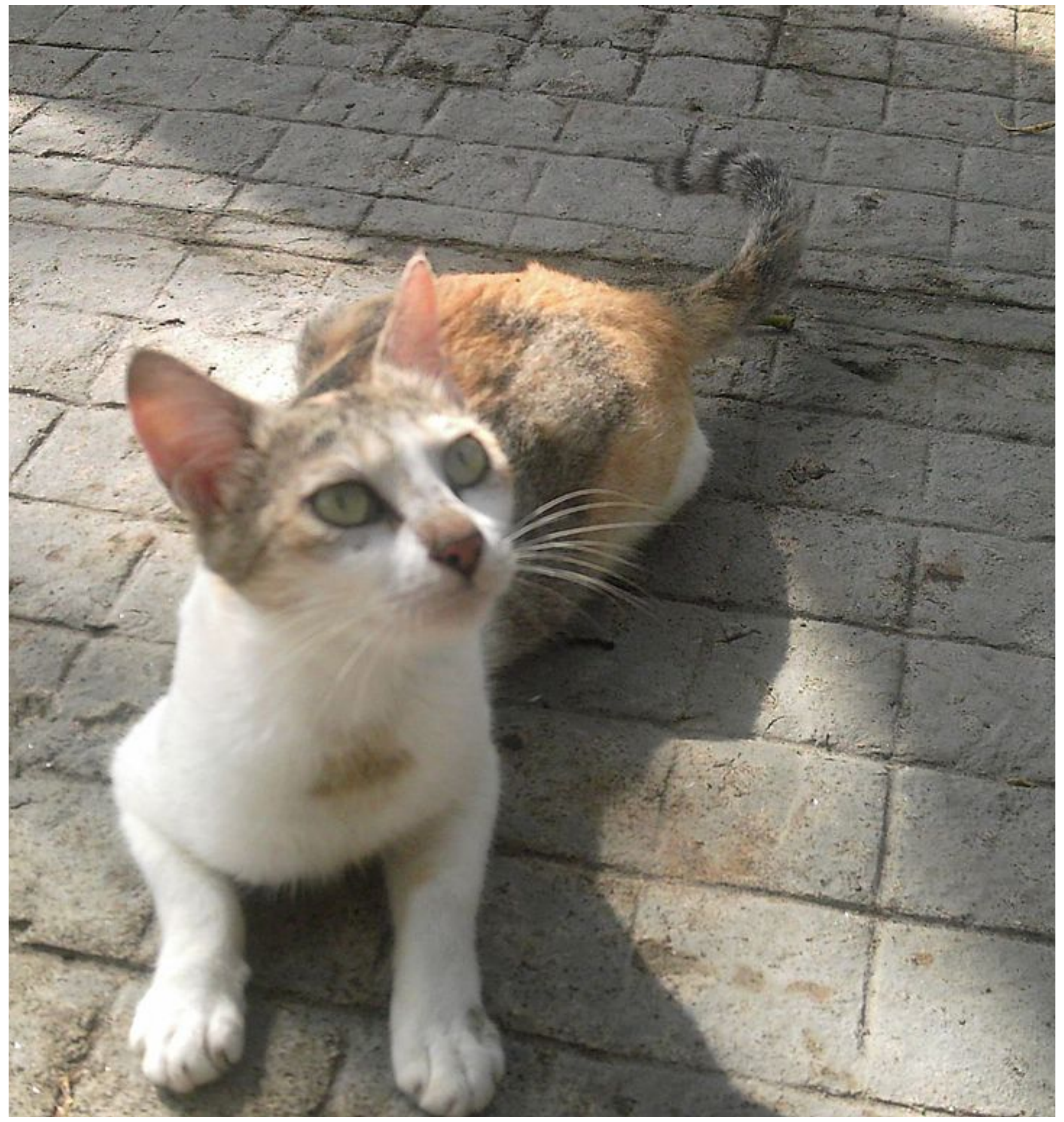
Are **deeper** neural networks better at **learning**?

https://arxiv.org/abs/2402.08808

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples



Cat

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Cat

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Cat

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

New Test Question

???

What is **learning**?

Inductive reasoning: Learning broad **generalizations** from **examples**

Examples

Dog

New Test Question

Cat

What is **learning**?

What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$

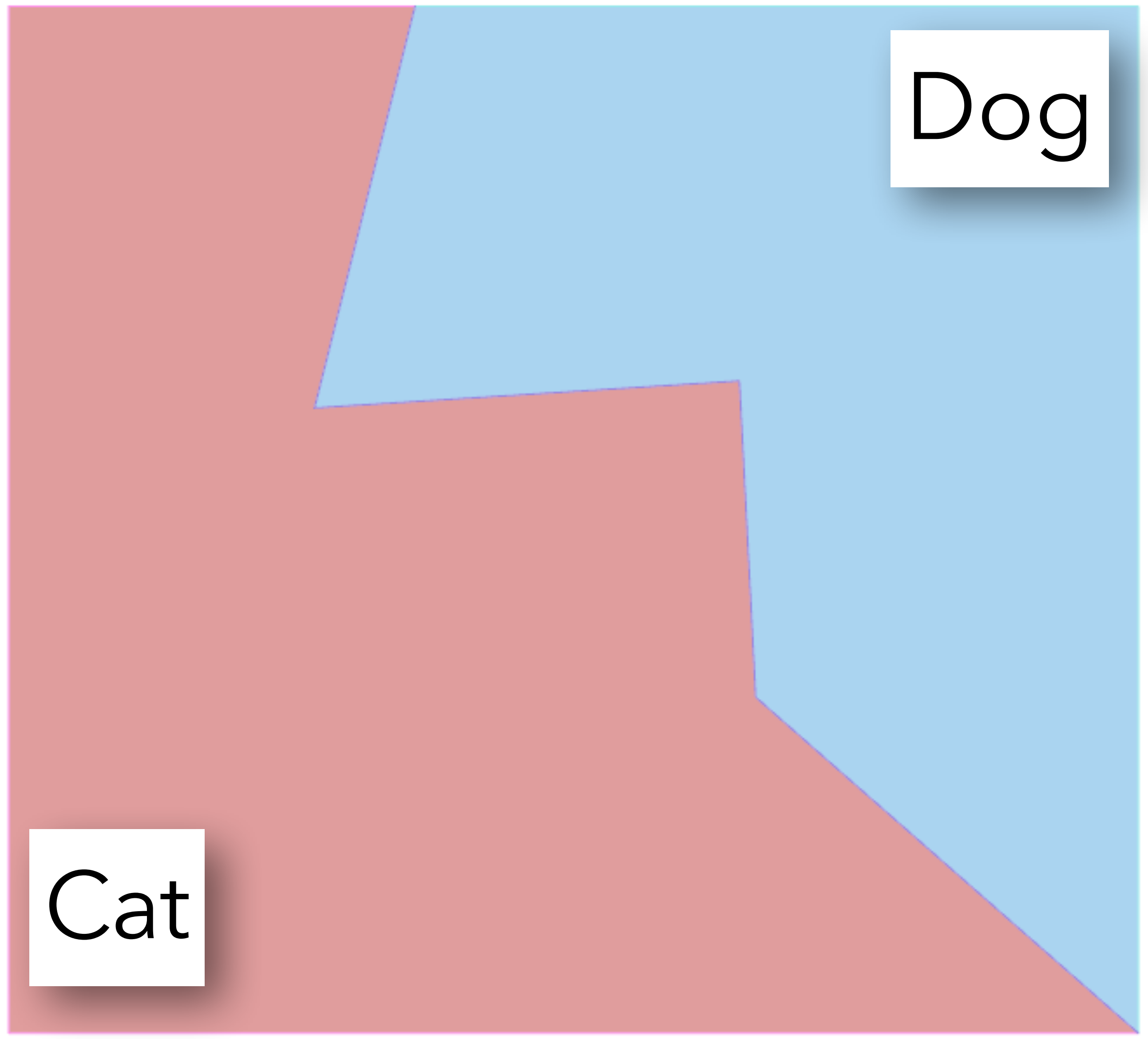
$$y = f(\mathbf{x})$$

What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$

$$y = f(\mathbf{x})$$



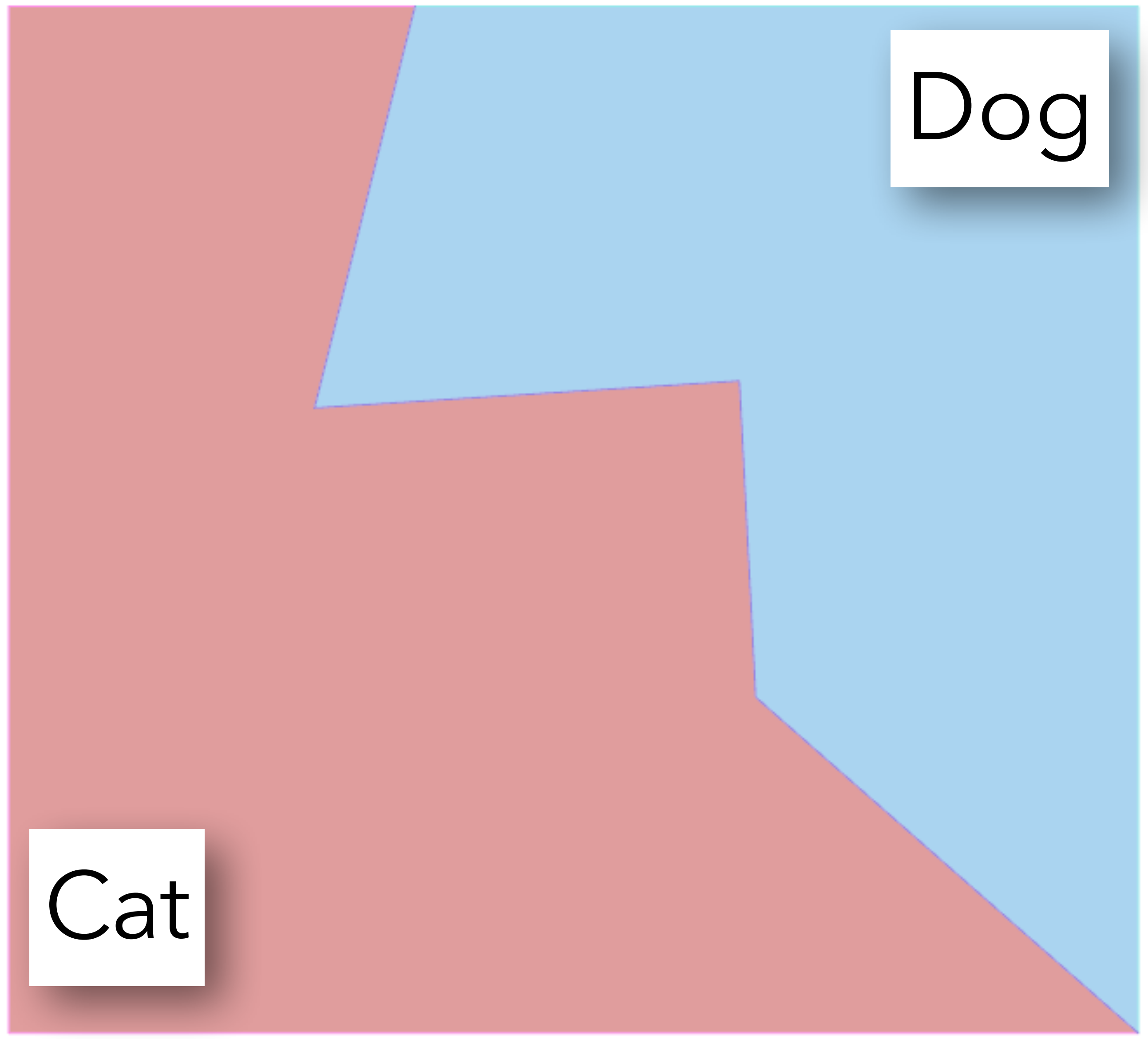
What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

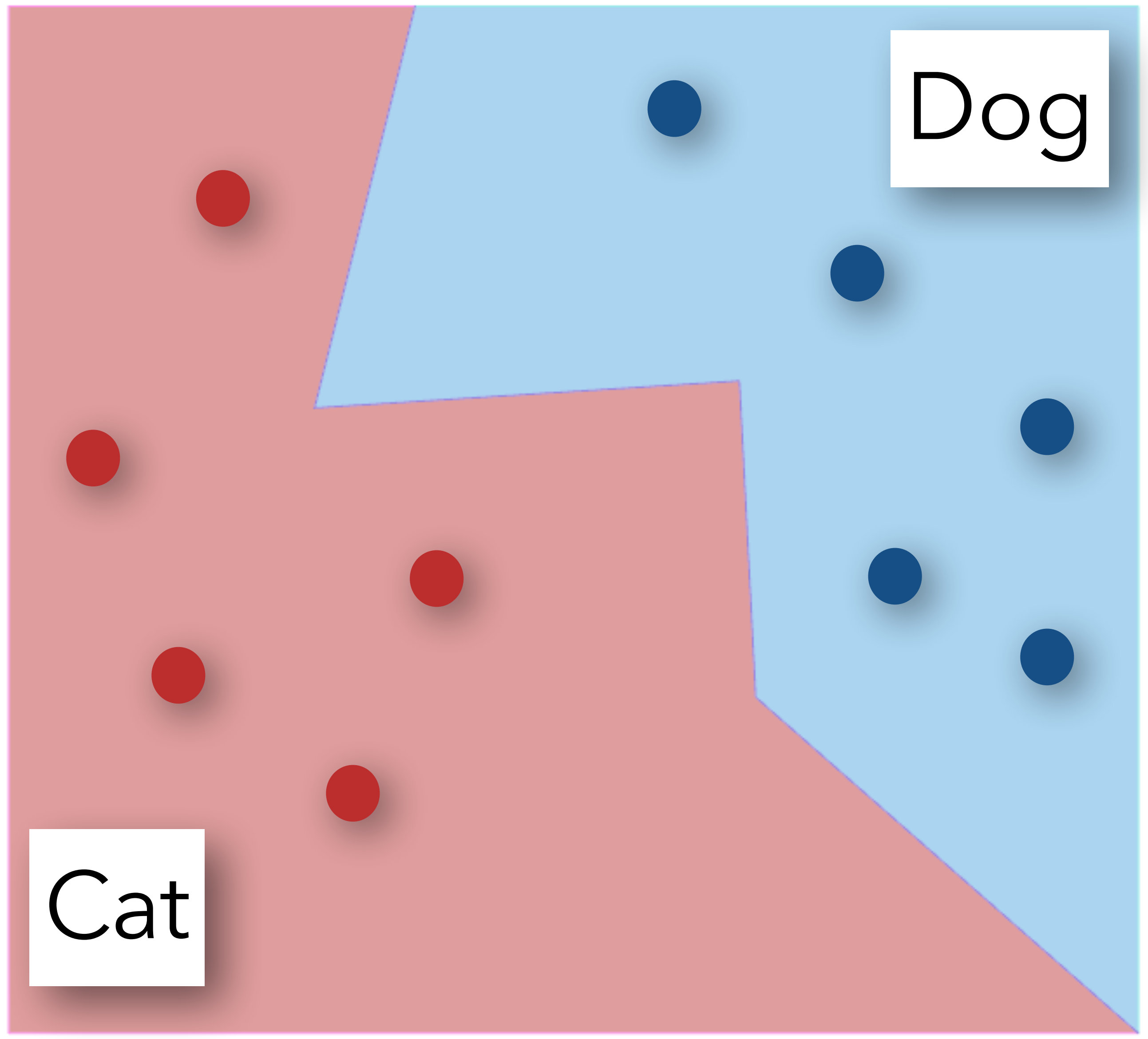
$$\mathbf{x} \sim \mathscr{D}$$

$$y = f(\mathbf{x})$$

- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathscr{X} \times \mathscr{Y})^m$



What is **learning**?



- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$

$$y = f(\mathbf{x})$$

- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m} \in (\mathscr{X} \times \mathscr{Y})^m$

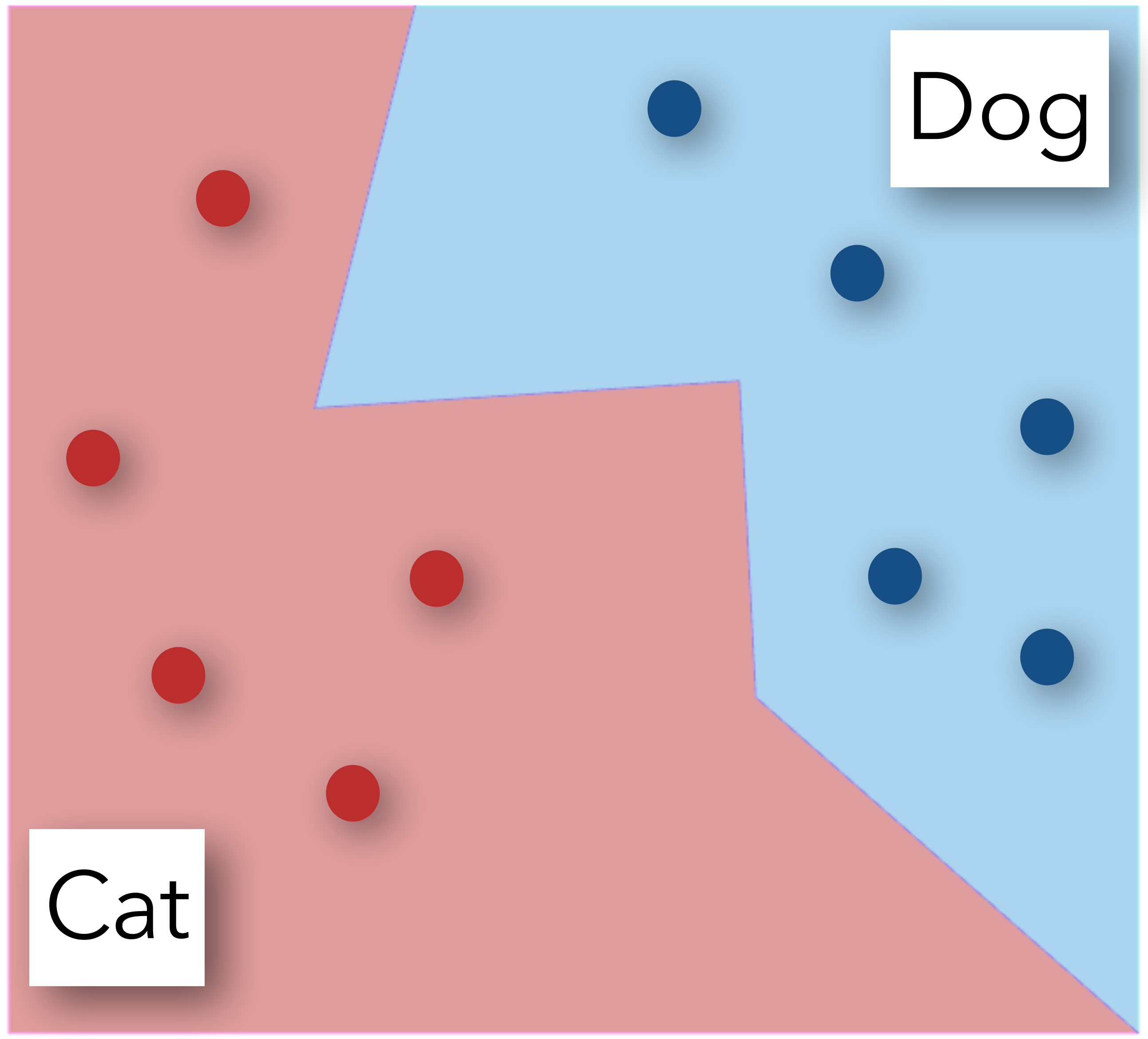
What is **learning**?

- There is some true underlying distribution over $\mathcal{X} \times \mathcal{Y}$

$$\mathbf{x} \sim \mathcal{D}$$
$$y = f(\mathbf{x})$$

- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathcal{X} \times \mathcal{Y})^m$

- Consider a set of possible models $g : \mathcal{X} \rightarrow \mathcal{Y} \in \mathcal{G}$ – **Model class**



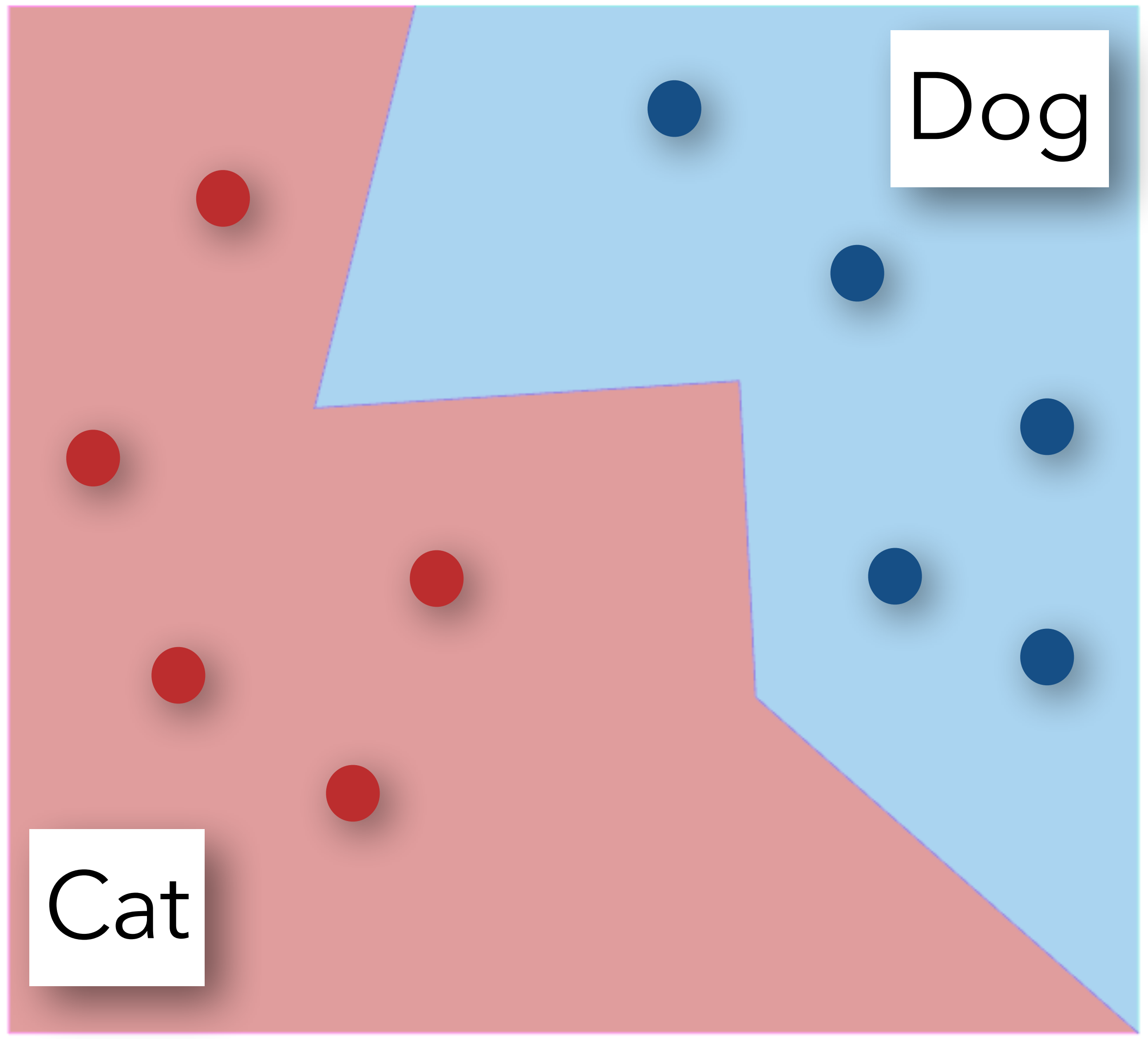
What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$
$$y = f(\mathbf{x})$$

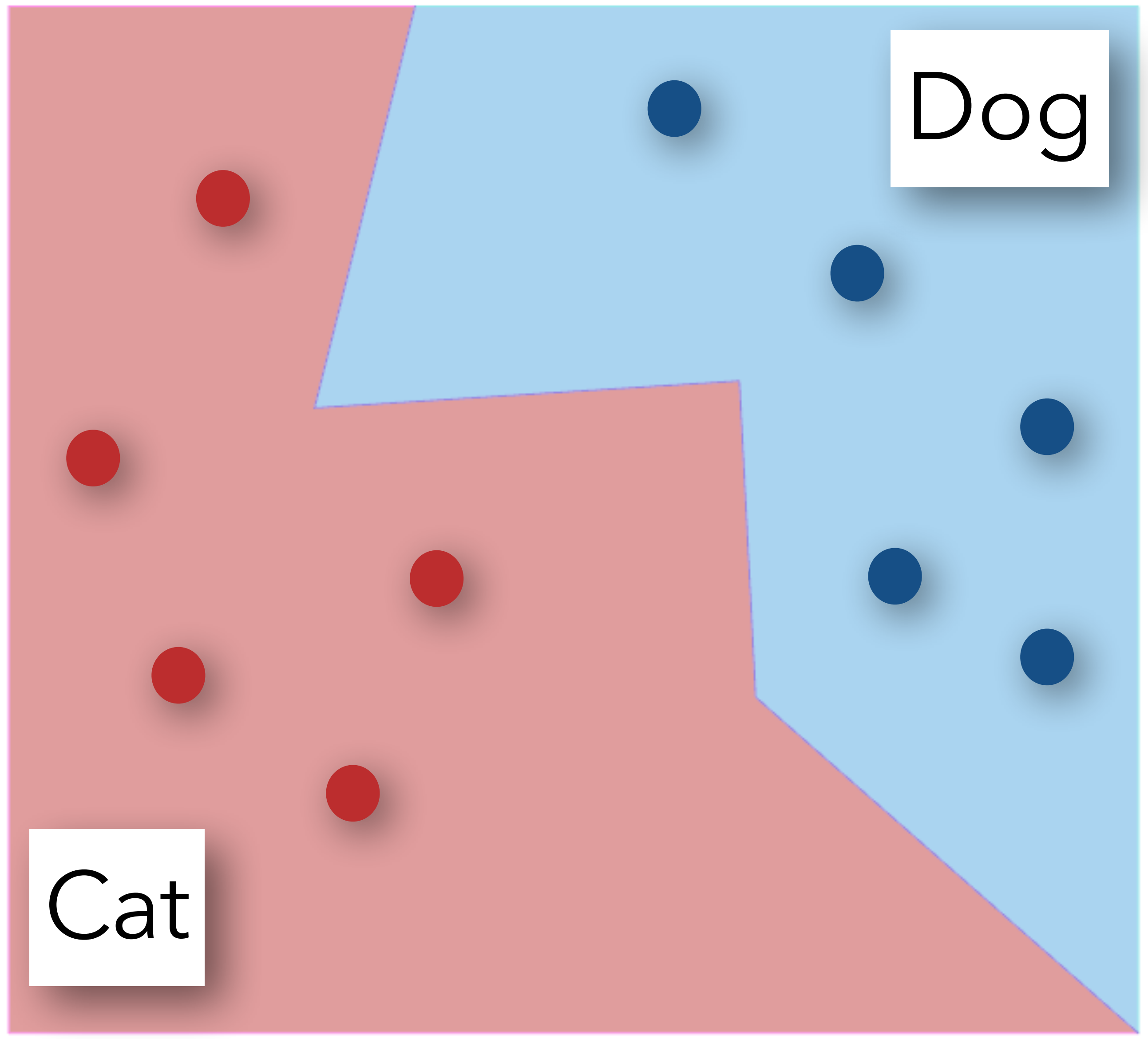
- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathscr{X} \times \mathscr{Y})^m$

- Consider a set of possible models $g : \mathscr{X} \to \mathscr{Y} \in \mathscr{G}$ – **Model class**



$\mathscr{G}$ = {linear separators}

What is **learning**?



- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$
$$y = f(\mathbf{x})$$

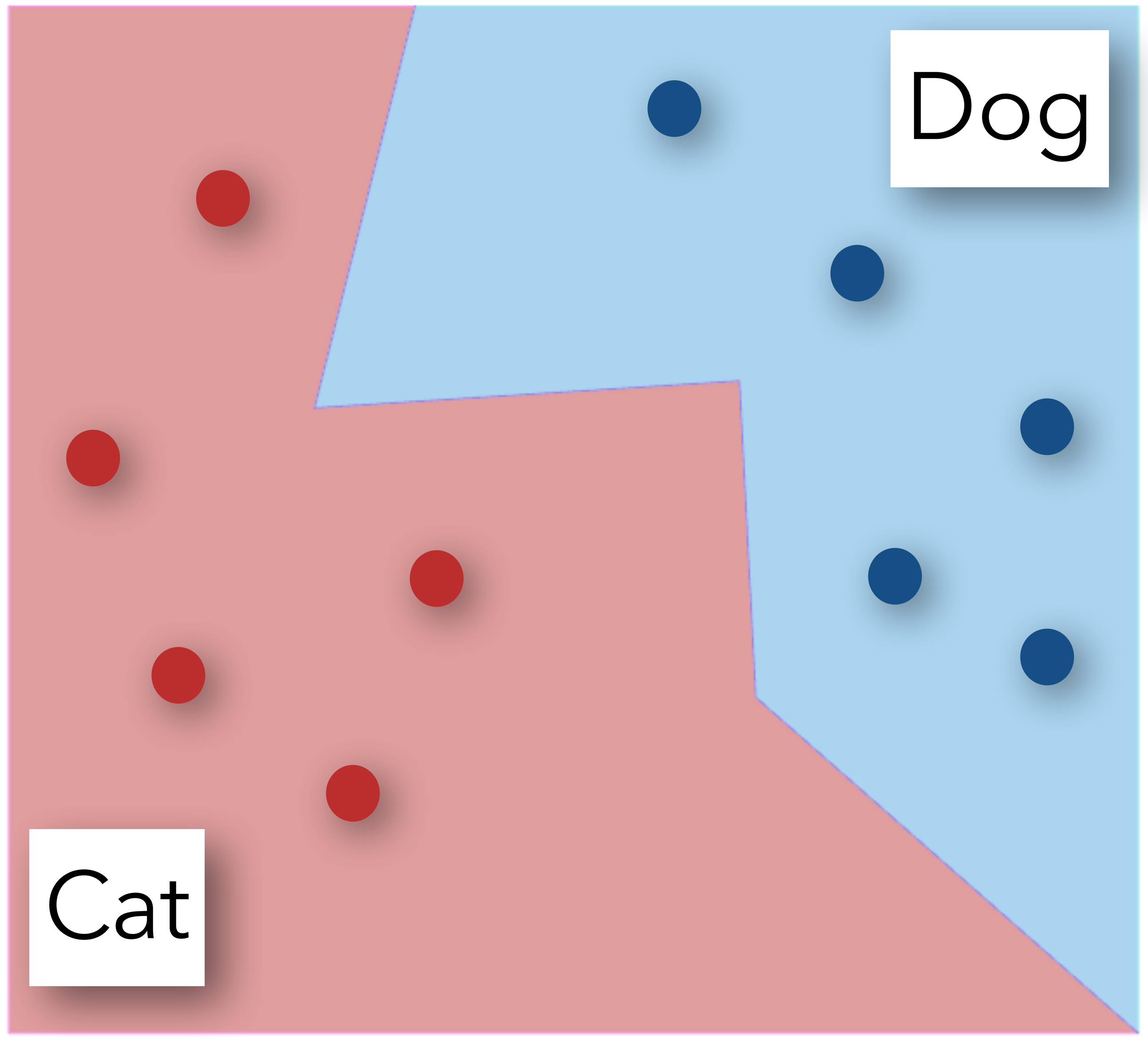
- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathscr{X} \times \mathscr{Y})^m$

- Consider a set of possible models $g : \mathscr{X} \to \mathscr{Y} \in \mathscr{G}$ – **Model class**

$\mathscr{G}$ = {linear separators}

- Use a **learning rule** $\mathscr{A} : (\mathscr{X} \times \mathscr{Y})^m \to \mathscr{G}$ to choose a model based on the training samples

What is **learning**?



- There is some true underlying distribution over $\mathcal{X} \times \mathcal{Y}$

$$\mathbf{x} \sim \mathcal{D}$$
$$y = f(\mathbf{x})$$

- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathcal{X} \times \mathcal{Y})^m$

- Consider a set of possible models $g : \mathcal{X} \to \mathcal{Y} \in \mathcal{G}$ – **Model class**

$\mathcal{G}$ = {linear separators}

- Use a **learning rule** $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{G}$ to choose a model based on the training samples

- Ex: Try to minimize **sample loss:**

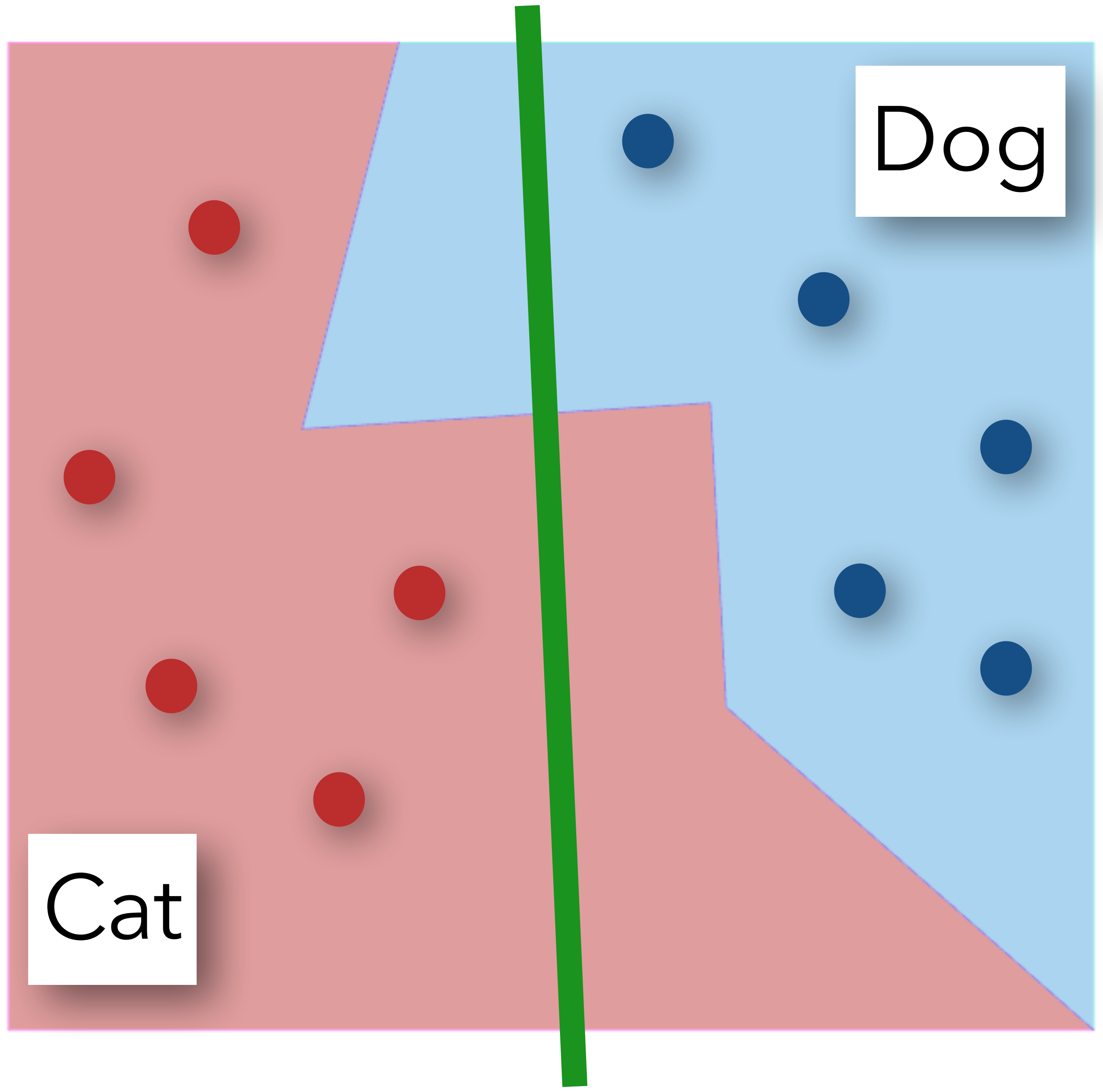
$$\mathcal{A}(S) \in \arg\min_{g \in \mathcal{G}} \mathcal{L}_S(g) := \frac{1}{m} \sum_{i=1}^{m} \left(g(\mathbf{x}_i) - y_i\right)^2$$

What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$
$$y = f(\mathbf{x})$$

- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathscr{X} \times \mathscr{Y})^m$



- Consider a set of possible models $g : \mathscr{X} \to \mathscr{Y} \in \mathscr{G}$ – **Model class**

$\mathscr{G}$ = {linear separators}

- Use a **learning rule** $\mathscr{A} : (\mathscr{X} \times \mathscr{Y})^m \to \mathscr{G}$ to choose a model based on the training samples

$\mathscr{A}(S)$ = green line

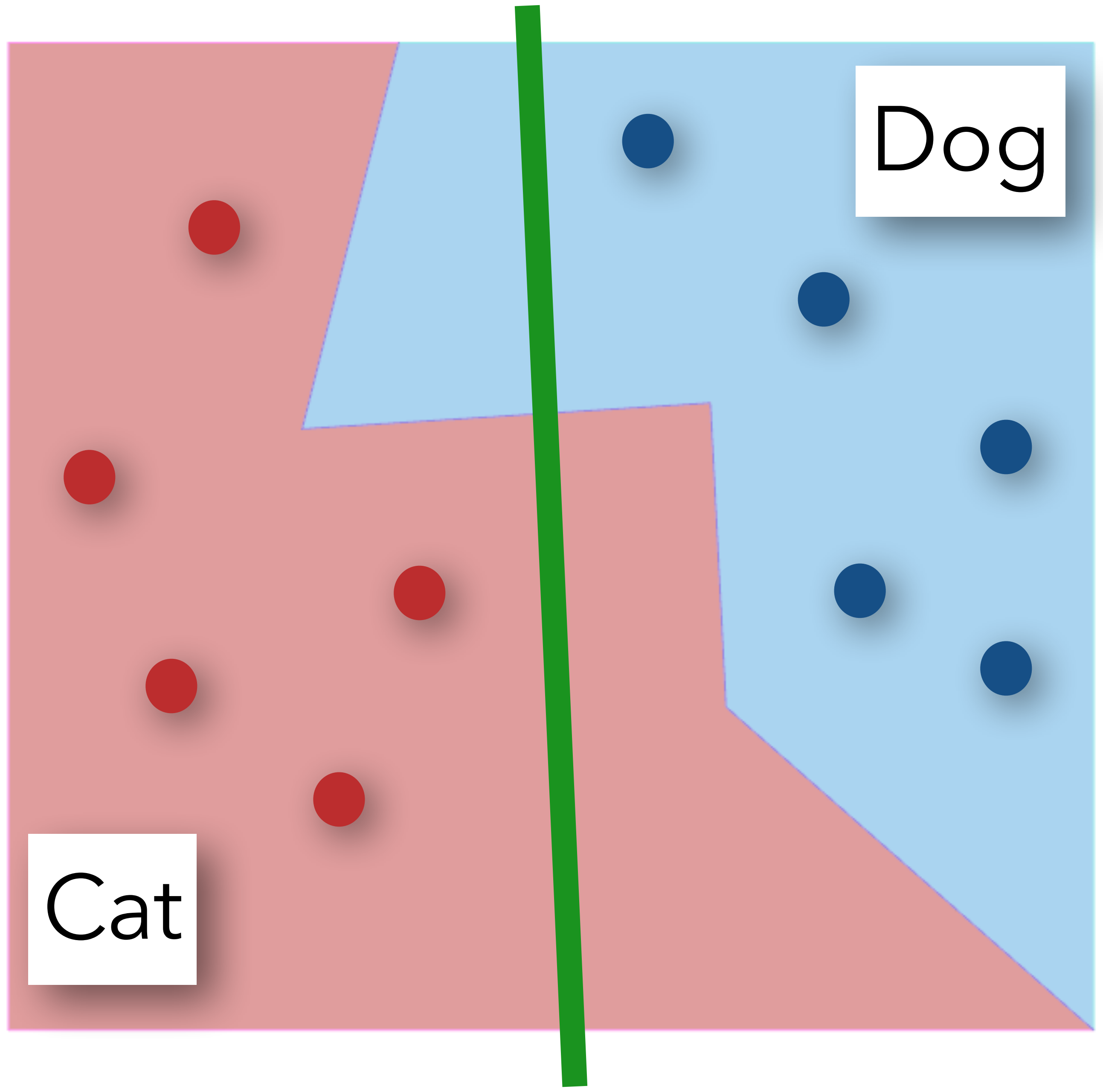
- Ex: Try to minimize **sample loss:**

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

What is **learning**?

- There is some true underlying distribution over $\mathscr{X} \times \mathscr{Y}$

$$\mathbf{x} \sim \mathscr{D}$$
$$y = f(\mathbf{x})$$



- Receive m training examples/samples $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \in (\mathscr{X} \times \mathscr{Y})^m$

- Consider a set of possible models $g : \mathscr{X} \to \mathscr{Y} \in \mathscr{G}$ – **Model class**

$\mathscr{G}$ = {linear separators}

- Use a **learning rule** $\mathscr{A} : (\mathscr{X} \times \mathscr{Y})^m \to \mathscr{G}$ to choose a model based on the training samples

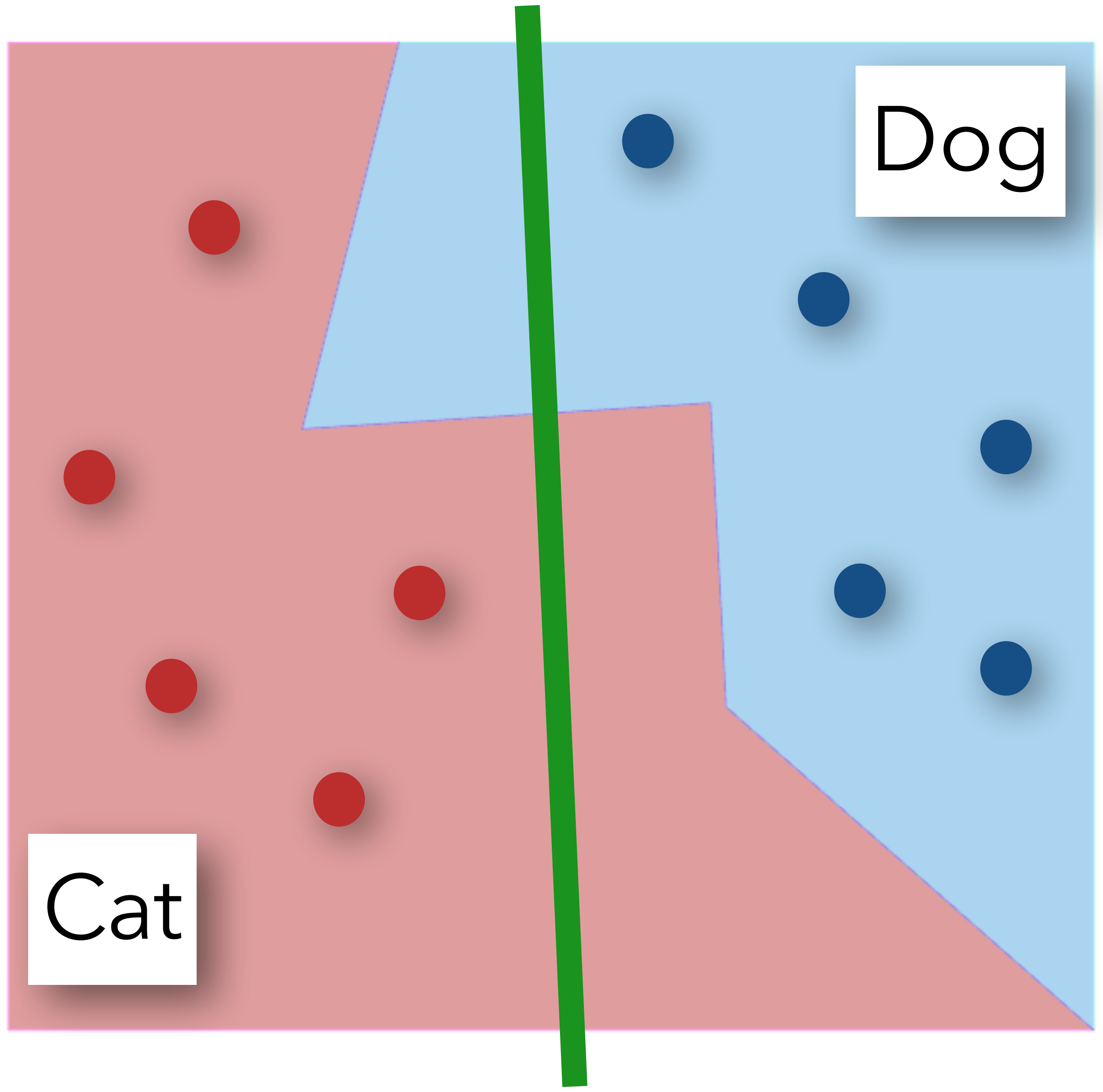
$\mathscr{A}(S)$ = green line

- Ex: Try to minimize **sample loss:**

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

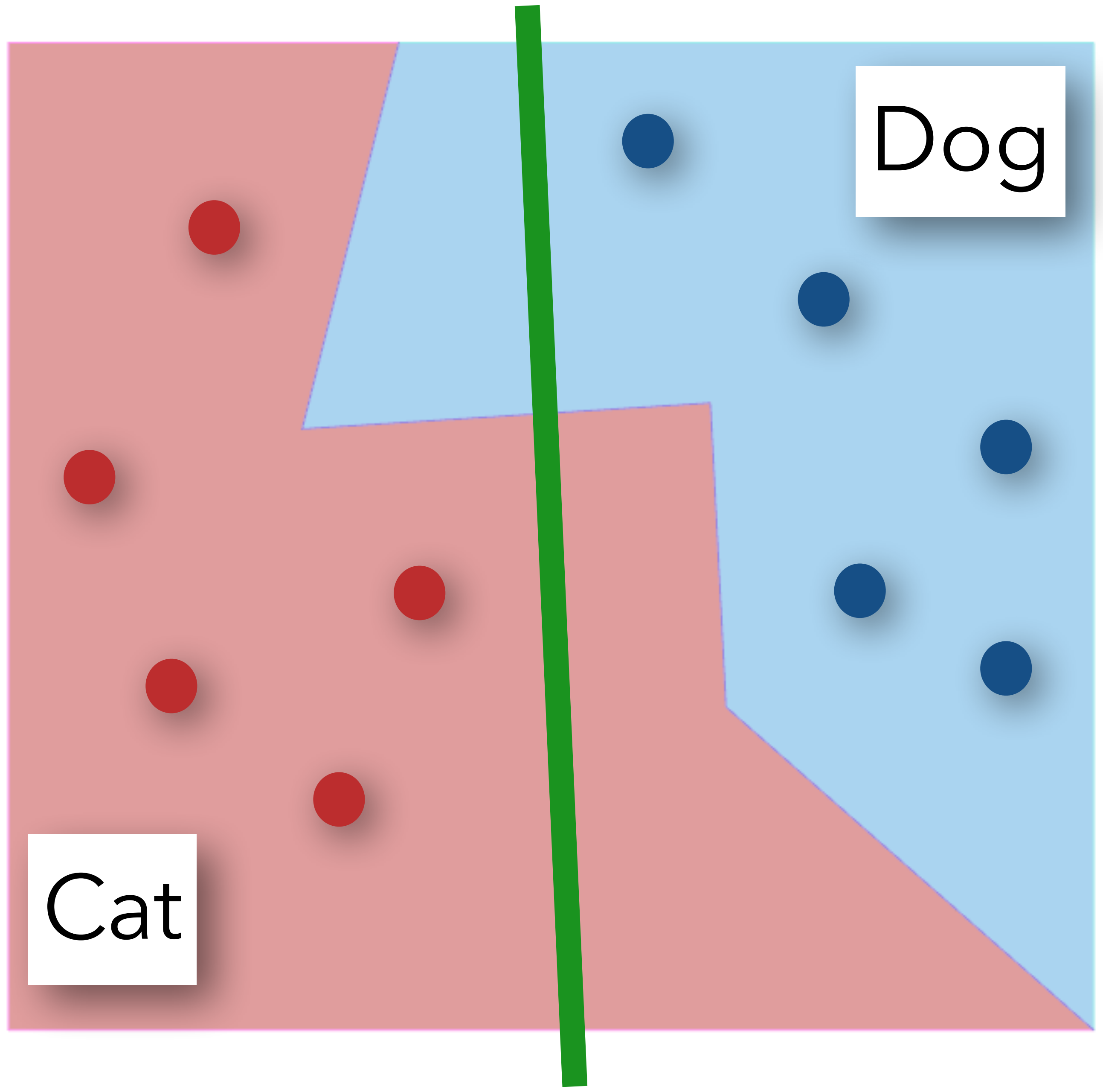
$$\mathscr{L}_S(\mathscr{A}(S)) = 0$$

What is **learning**?

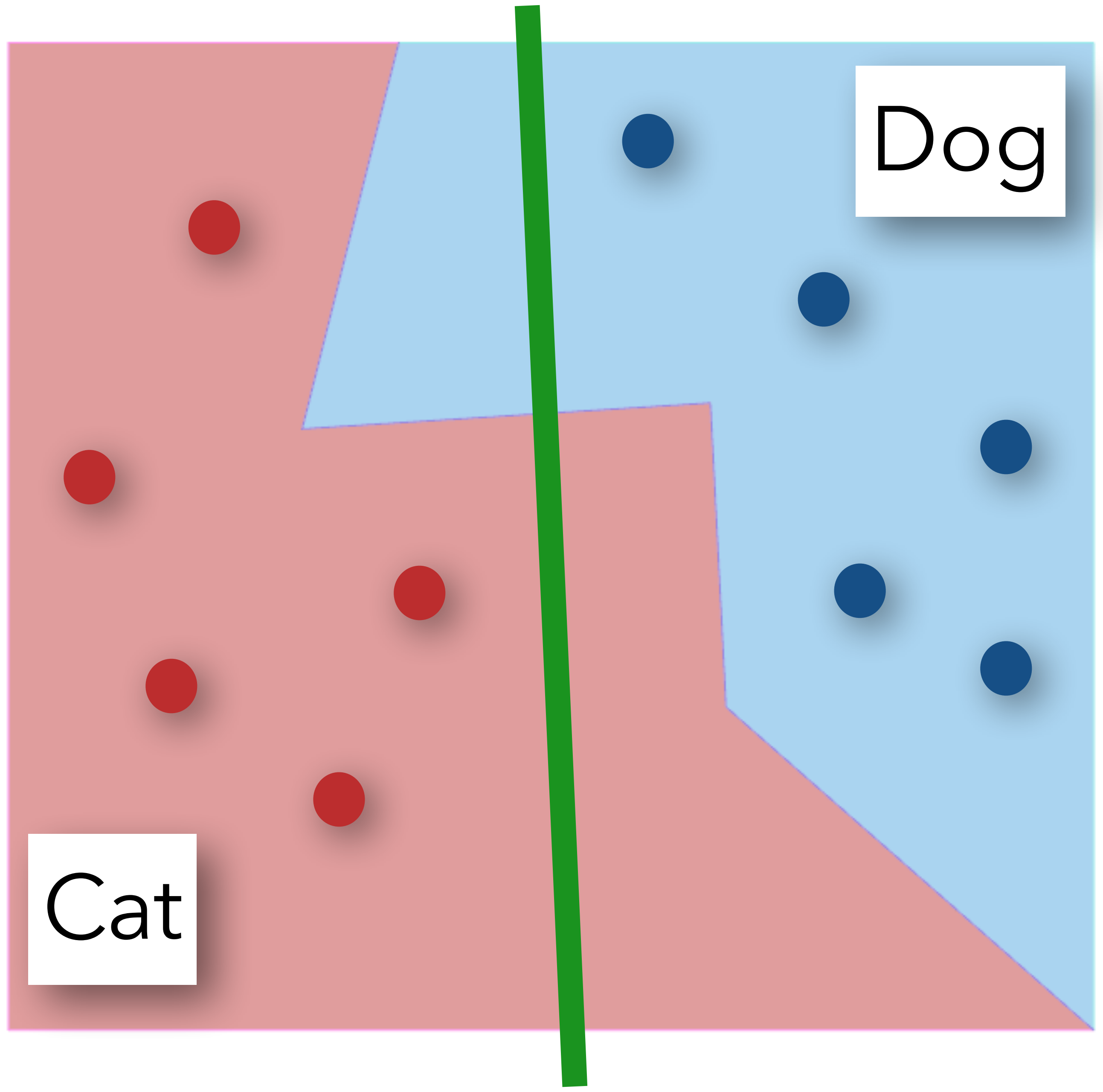


What is **learning**?

- Ideally $\mathscr{A}(S) = f$



What is **learning**?



- Ideally $\mathscr{A}(S) = f$
- Or at least the **generalization error/expected loss**

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

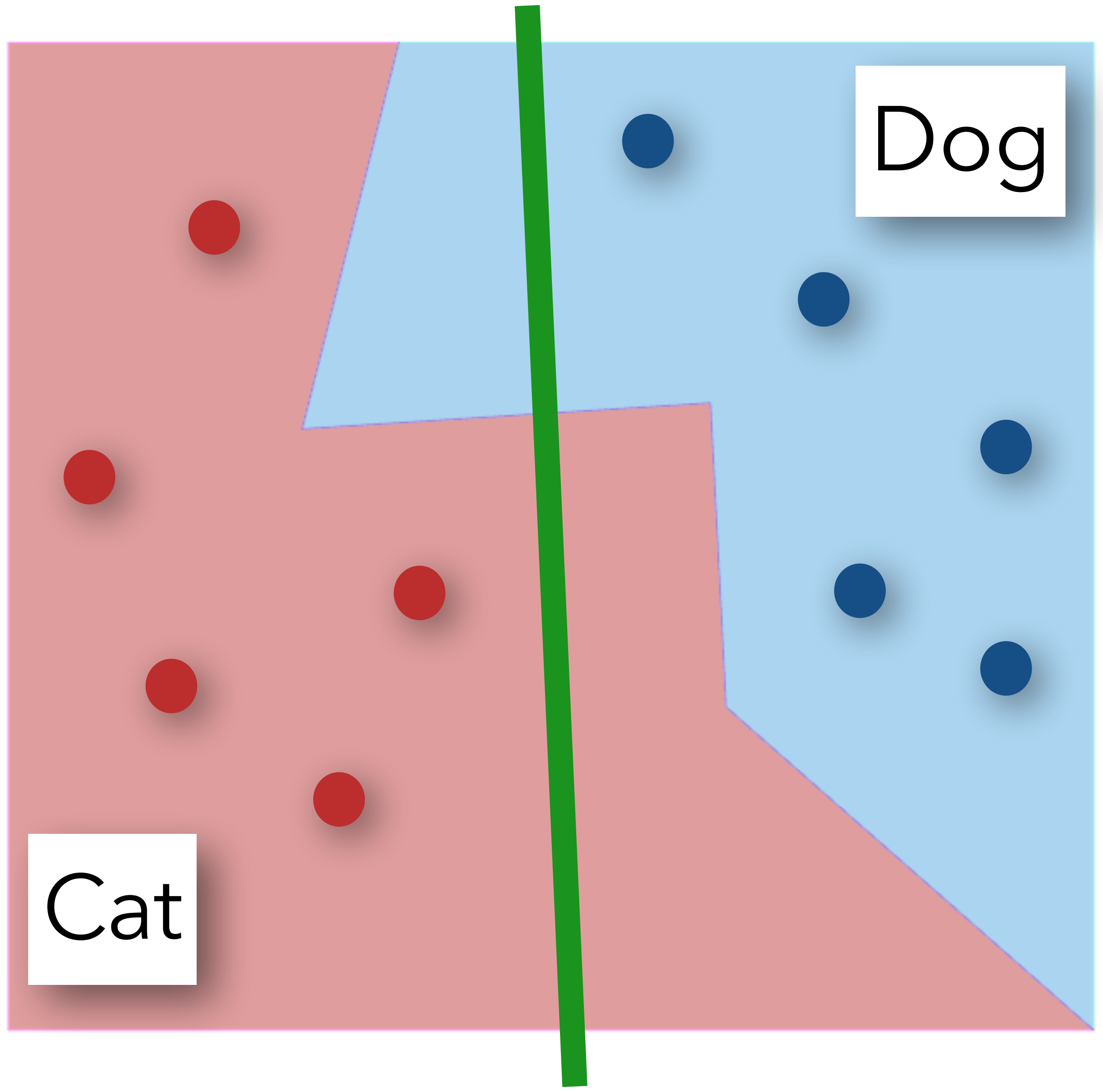
  is small.

What is **learning**?

- Ideally $\mathscr{A}(S) = f$
- Or at least the **generalization error/expected loss**

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

  is small.



$$\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S)) \gg 0$$

What is **learning**?

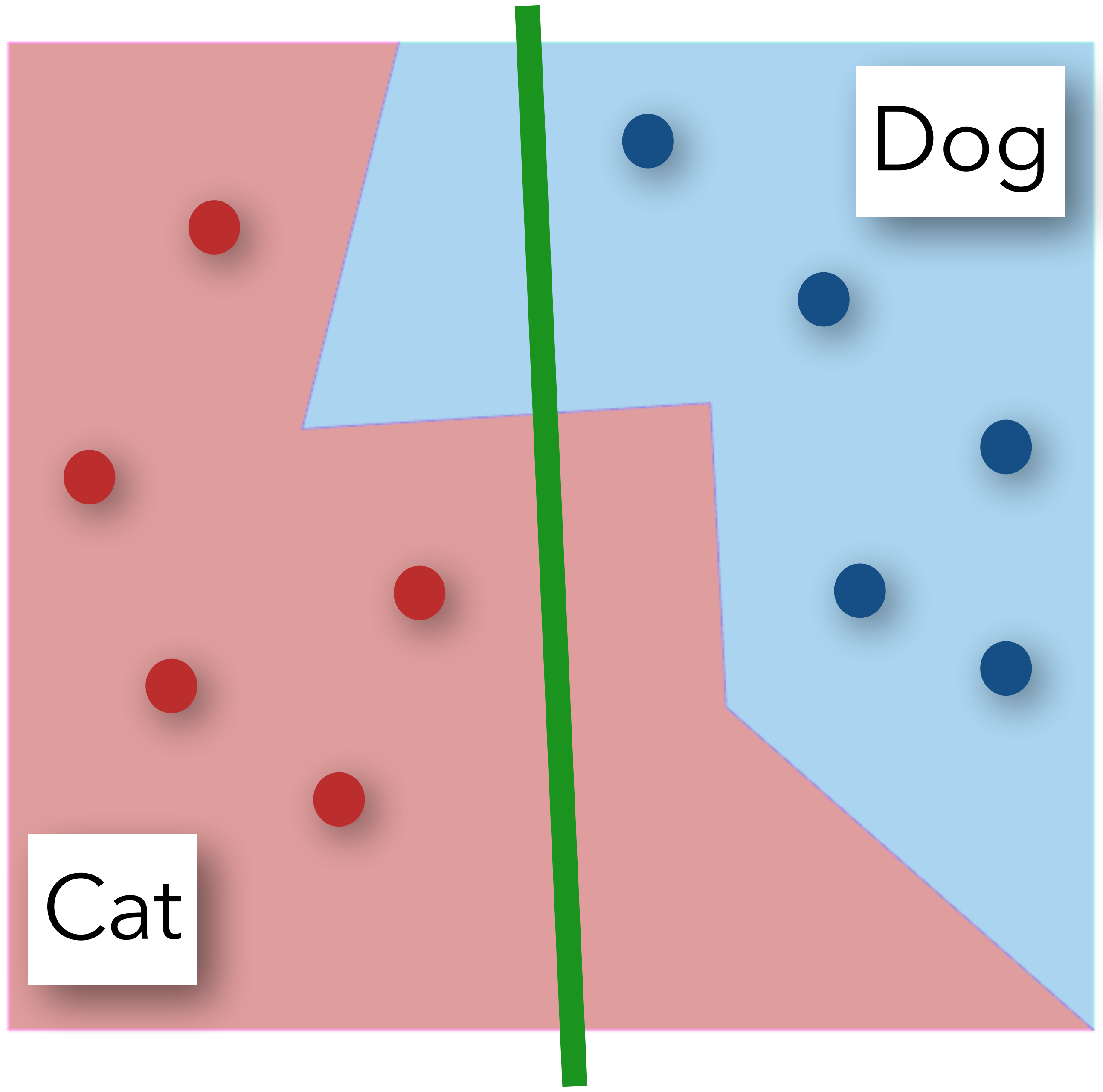
- Ideally $\mathscr{A}(S) = f$

- Or at least the **generalization error/expected loss**

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

  is small.

- Since we only train on **finitely many samples** and we're using a **limited model** class, the best we can hope for is to be **Probably Approximately Correct (PAC)**.



$$\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S)) \gg 0$$

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:*

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

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:*

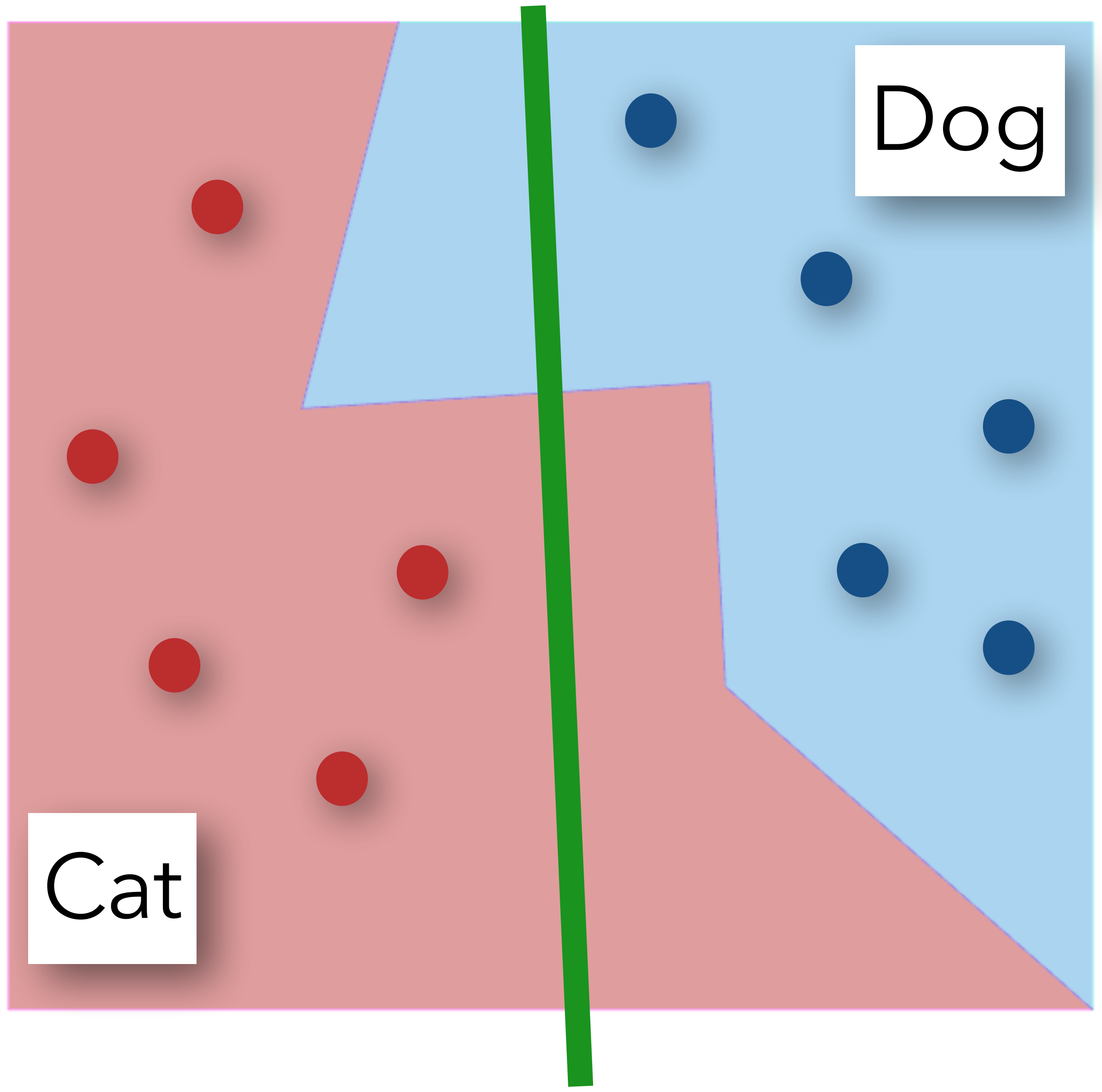
$$\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S)) < \varepsilon \,.$$

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

- We often end up with error bounds like this:

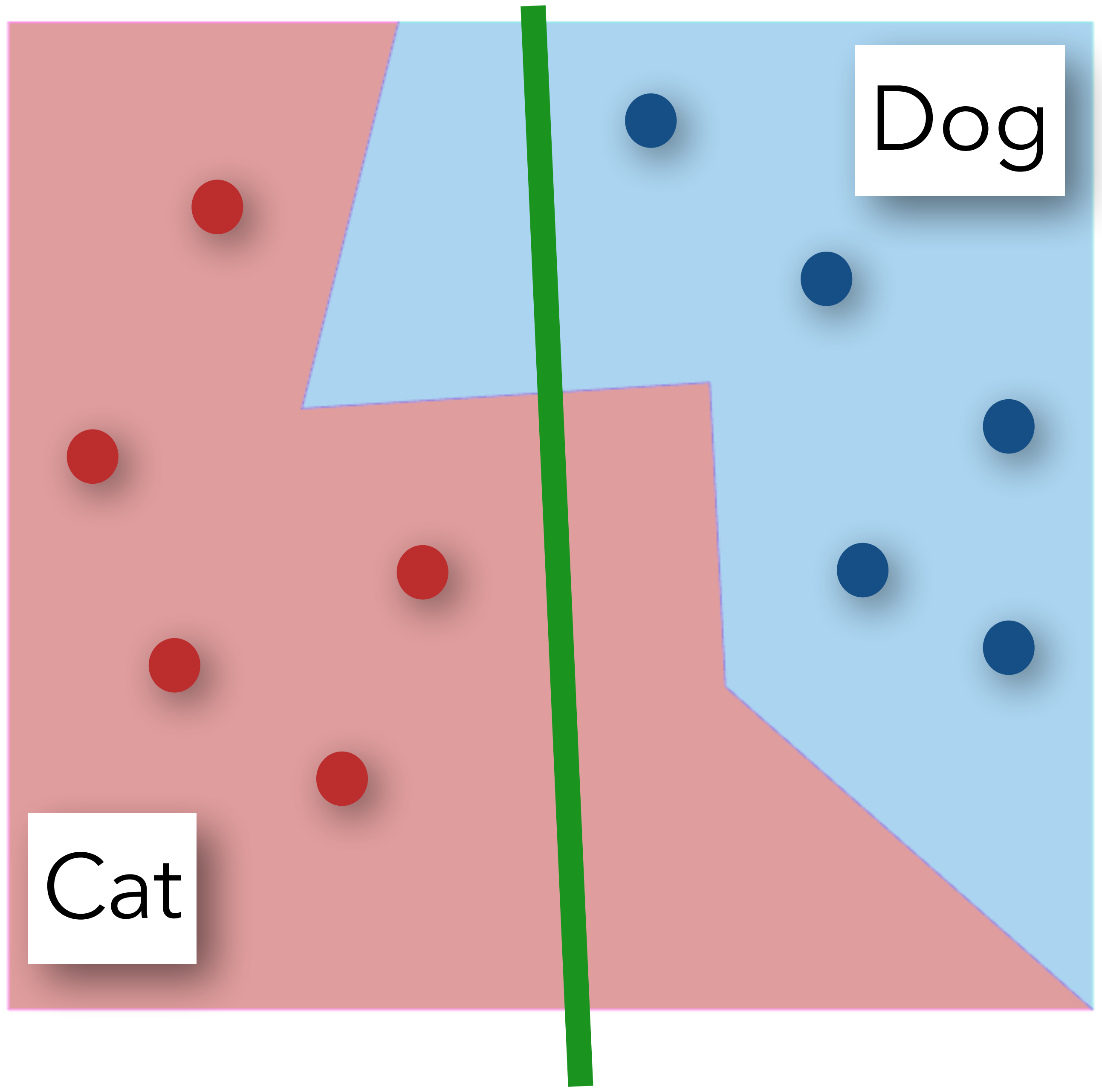
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)} + \underbrace{2 \sup_{g \in \mathscr{G}} |\mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)|}$$



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

- We often end up with error bounds like this:

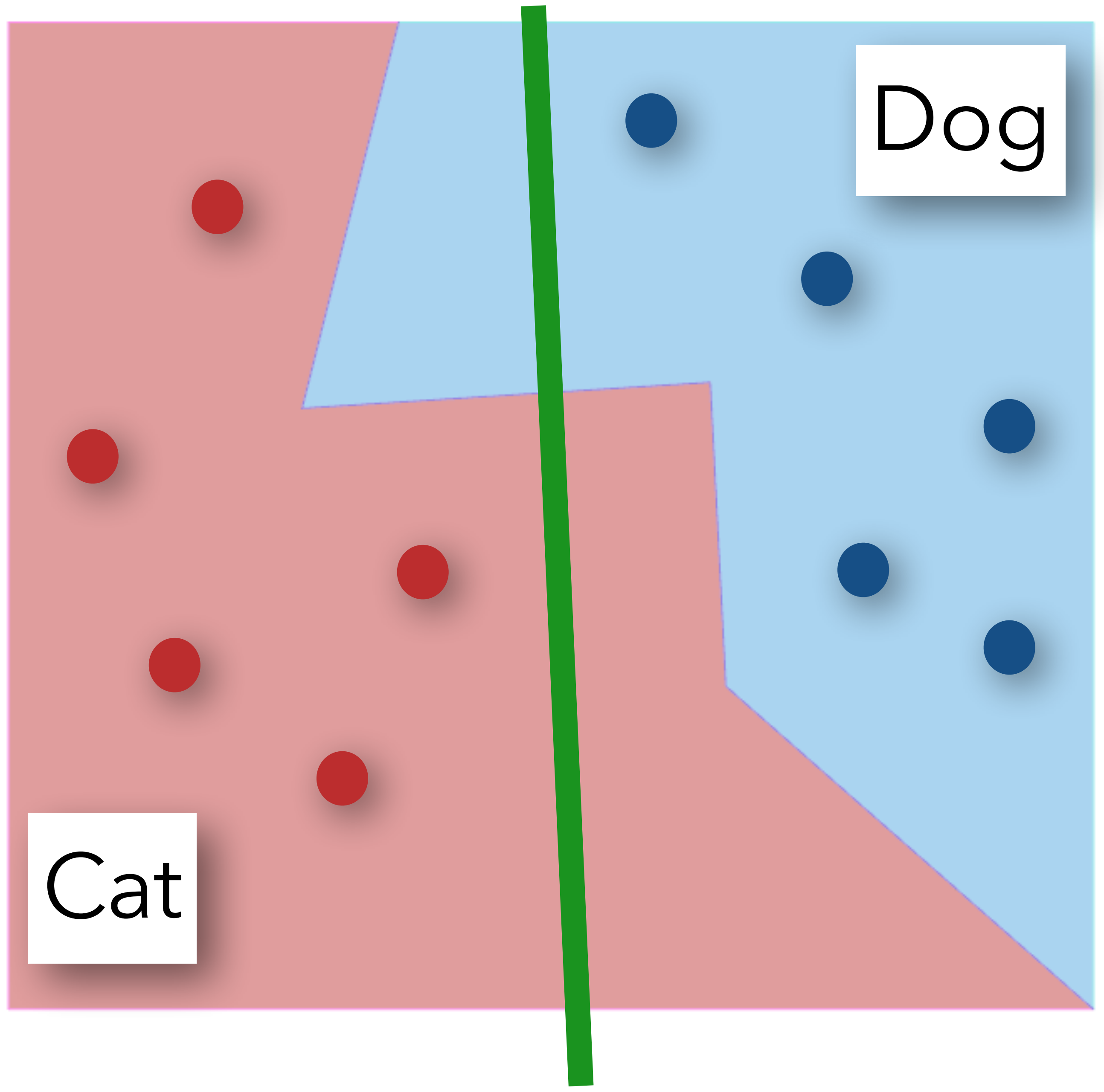
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\substack{\textbf{Generalization} \\ \textbf{Error} \\ \textbf{(expected loss)}}} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)} + \underbrace{2 \sup_{g \in \mathscr{G}} | \mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)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



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

- We often end up with error bounds like this:

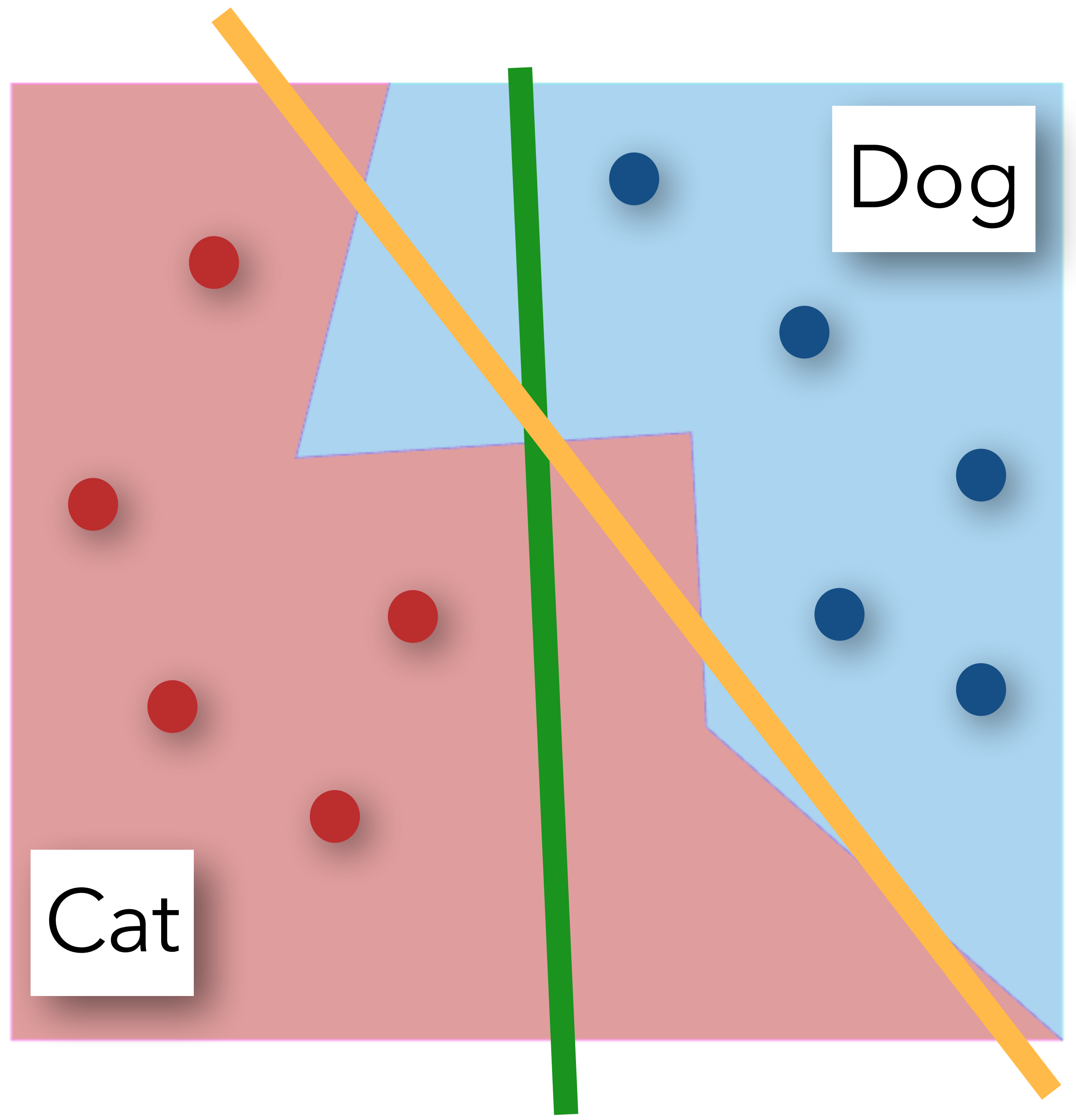
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\text{Generalization Error (expected loss)}} \le \underbrace{\inf_{g\in\mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)}_{\text{Approximation Error}} + \underbrace{2\sup_{g\in\mathscr{G}} |\mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)|}_{}$$



Generalization vs. Approximation vs. Estimation Error

- We often end up with error bounds like this:

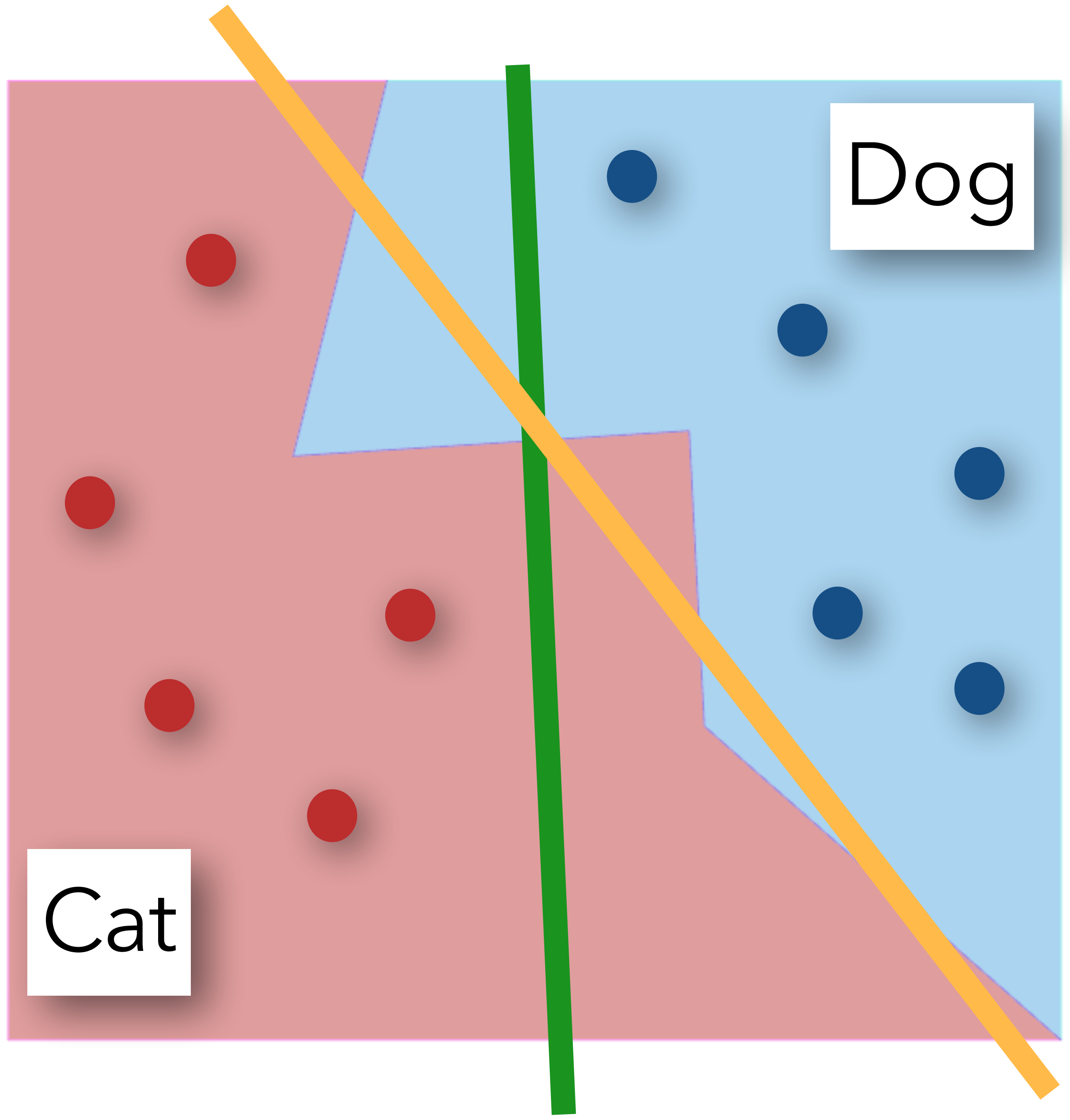
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\text{Generalization Error (expected loss)}} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)}_{\text{Approximation Error}} + \underbrace{2 \sup_{g \in \mathscr{G}} | \mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g) |}$$



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

- We often end up with error bounds like this:

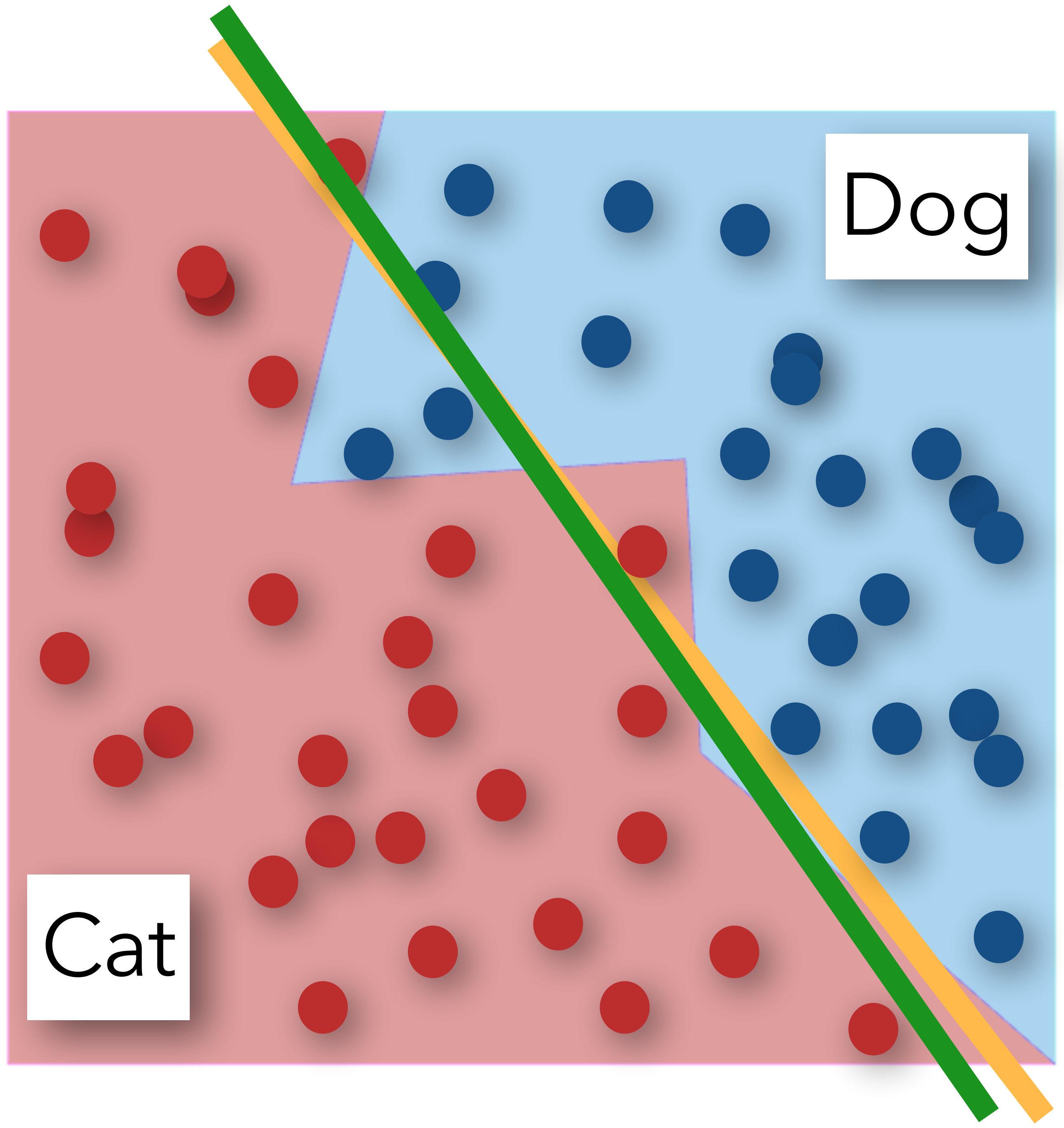
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\substack{\textbf{Generalization}\\\textbf{Error}\\\textbf{(expected loss)}}} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)}_{\substack{\textbf{Approximation}\\\textbf{Error}}} + \underbrace{2 \sup_{g \in \mathscr{G}}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



Generalization vs. Approximation vs. Estimation Error

- We often end up with error bounds like this:

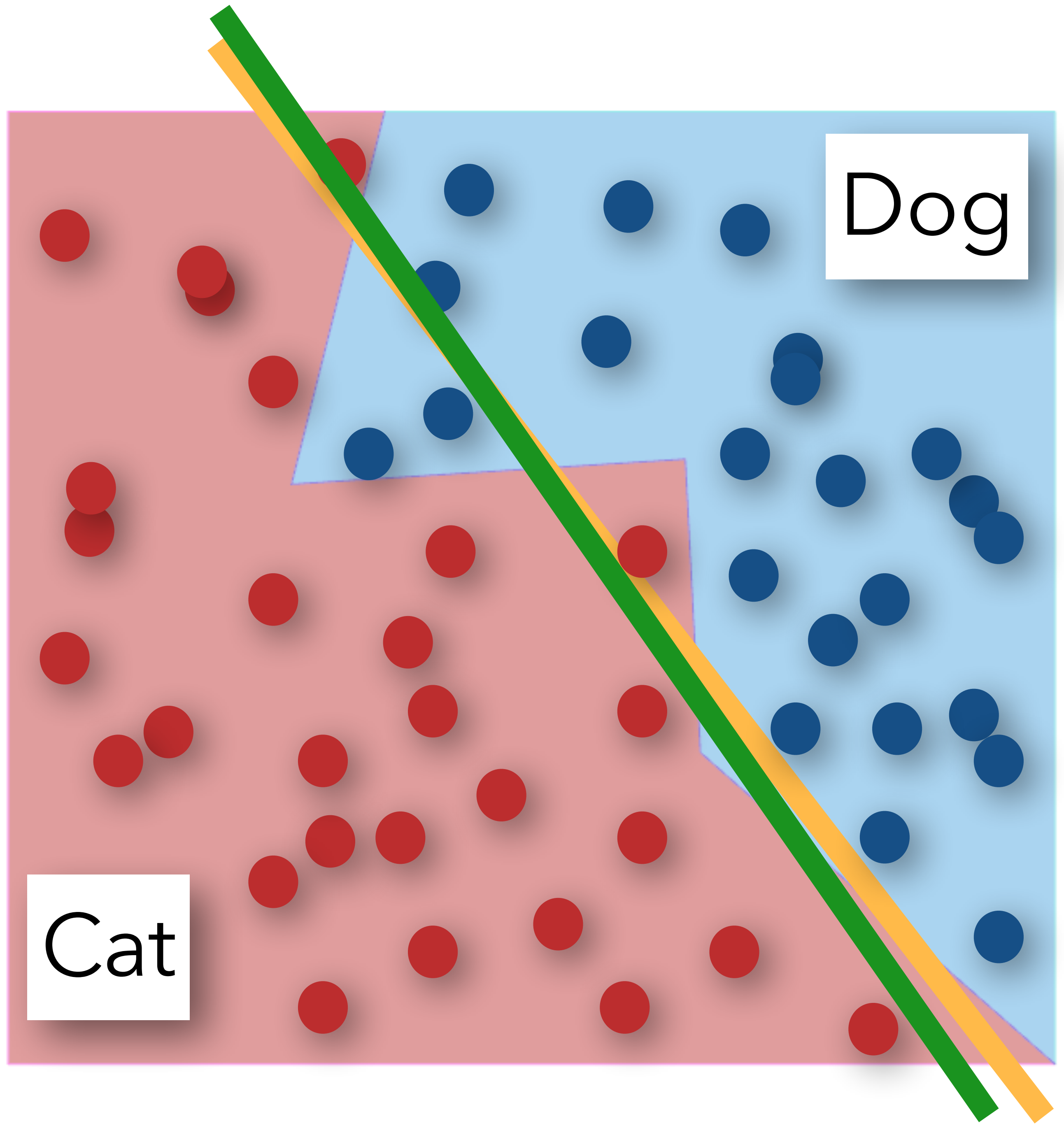
$$\underbrace{\mathscr{L}_{\mathscr{D}}(\mathscr{A}(S))}_{\text{Generalization Error (expected loss)}} \leq \underbrace{\inf_{g \in \mathscr{G}} \mathscr{L}_{\mathscr{D}}(g)}_{\text{Approximation Error}} + \underbrace{2 \sup_{g \in \mathscr{G}} |\mathscr{L}_S(g) - \mathscr{L}_{\mathscr{D}}(g)|}_{\text{Estimation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



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

- We often end up with error bounds like this:

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

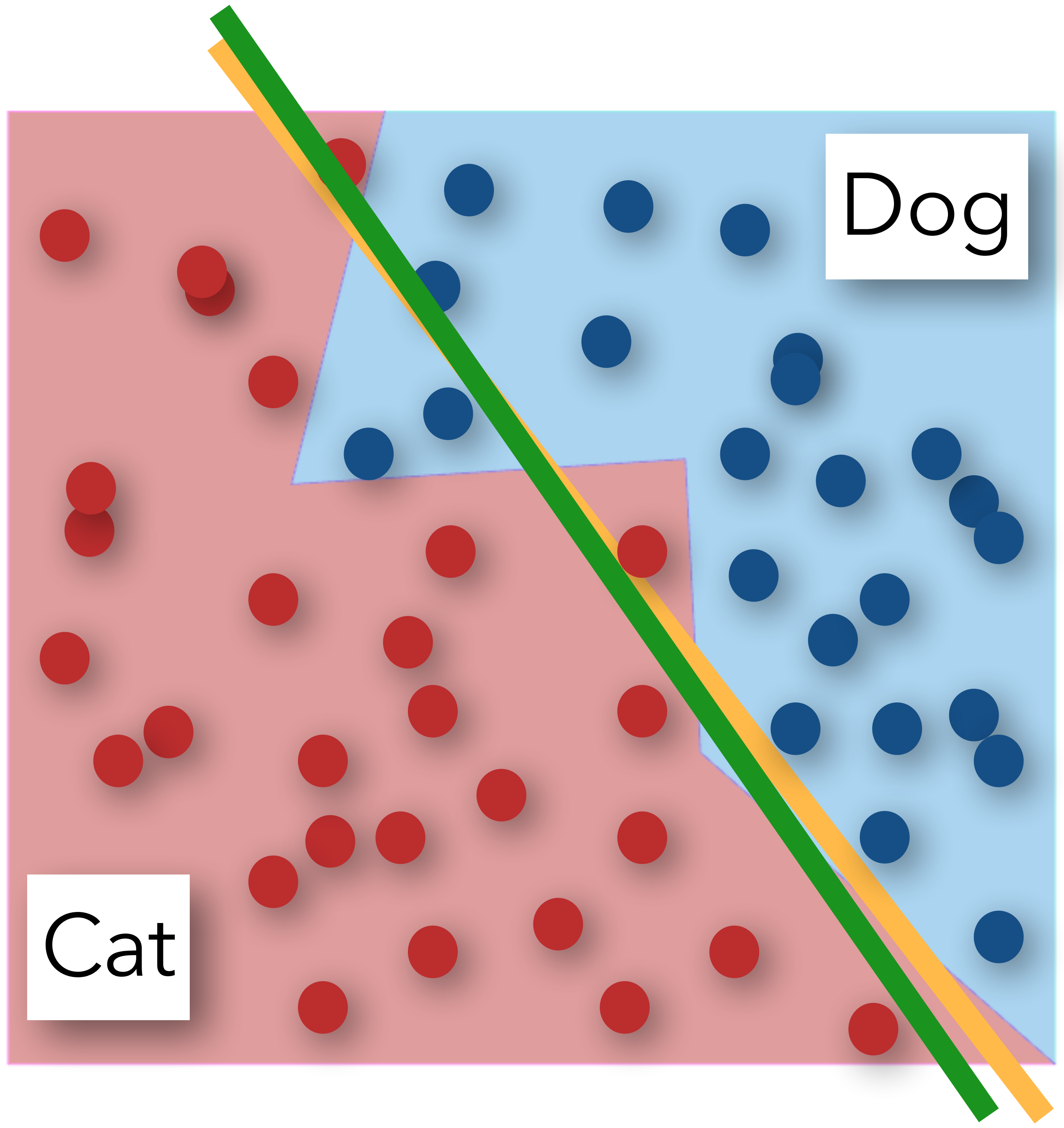


- **Approximation error:** Controlled using Universal Approximation Theorems. Need existence of **one** good approximator $g \in \mathscr{G}$. Hornik (1991), Shen et al. (2022)

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

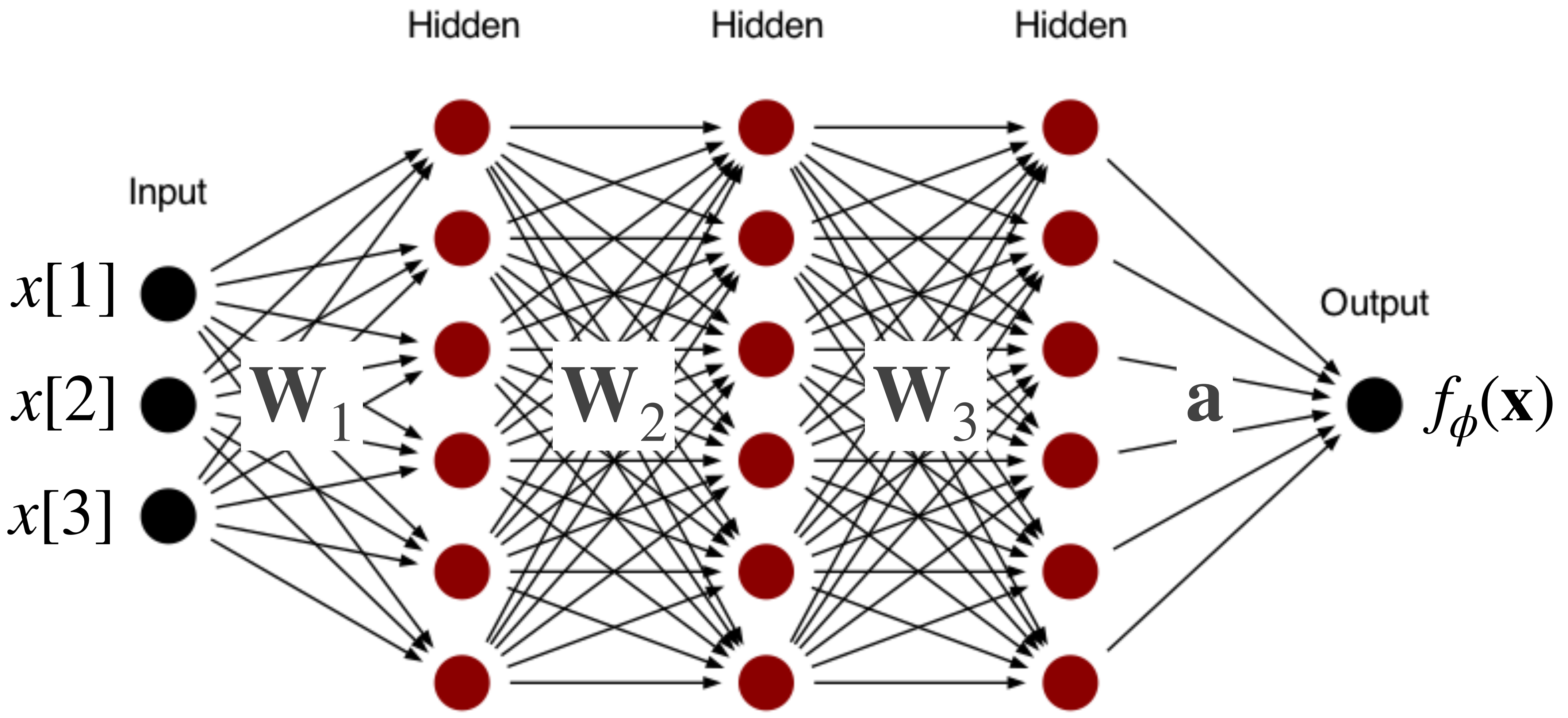
- We often end up with error bounds like this:

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



- **Approximation error:** Controlled using Universal Approximation Theorems. Need existence of **one** good approximator $g \in \mathscr{G}$. Hornik (1991), Shen et al. (2022)
- **Estimation error:** Controlled using **size** of $\mathscr{G}$, as measured by VC-dimension, **Rademacher complexity**, metric entropy, etc. Vapnik & Chervonenkis (1971), Bartlett & Mendelson (2001), Neyshabur et al. (2015),

Neural Networks

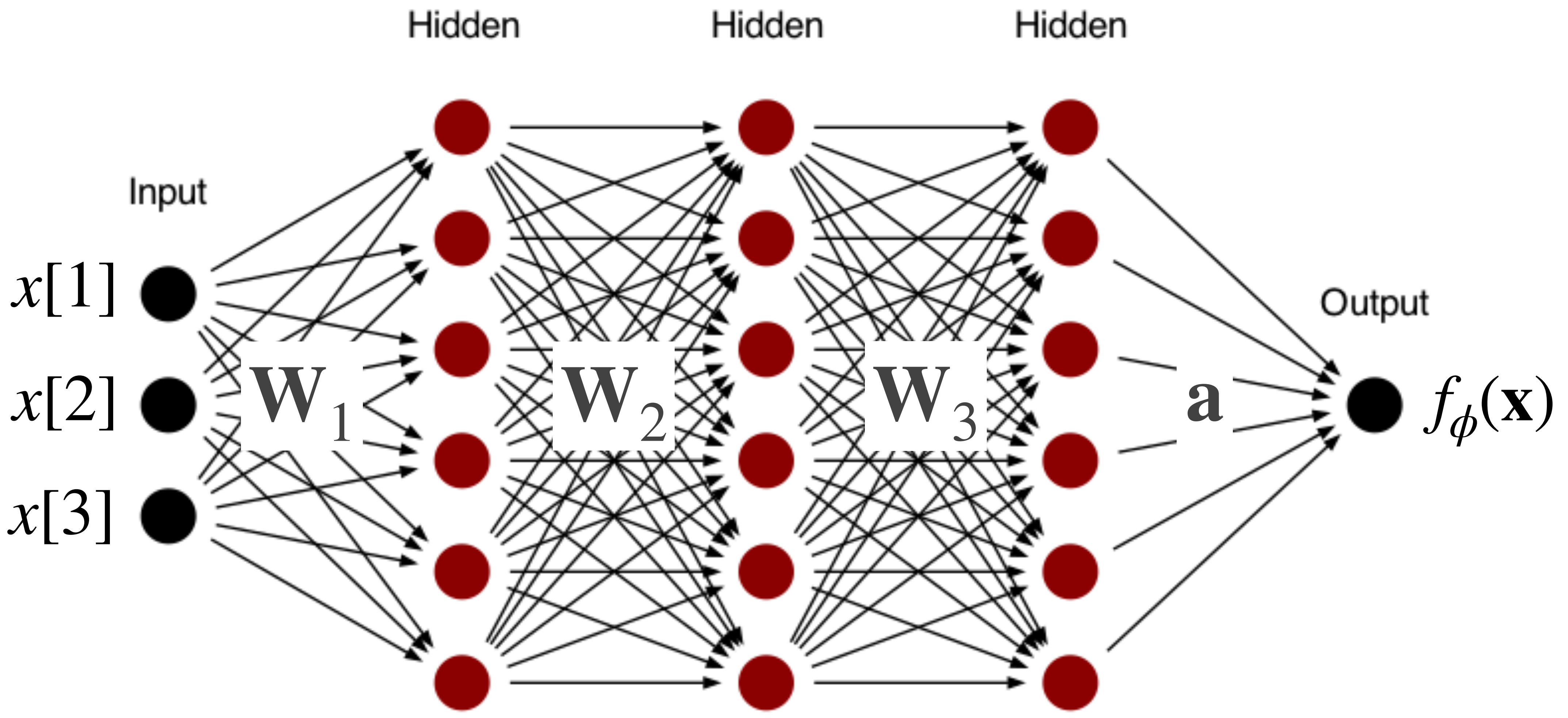


$$\phi = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{L-1}, \mathbf{a}\right)$$

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

Common choice for σ: $\mathrm{ReLU}(x) = \max(0, x)$

Neural Networks



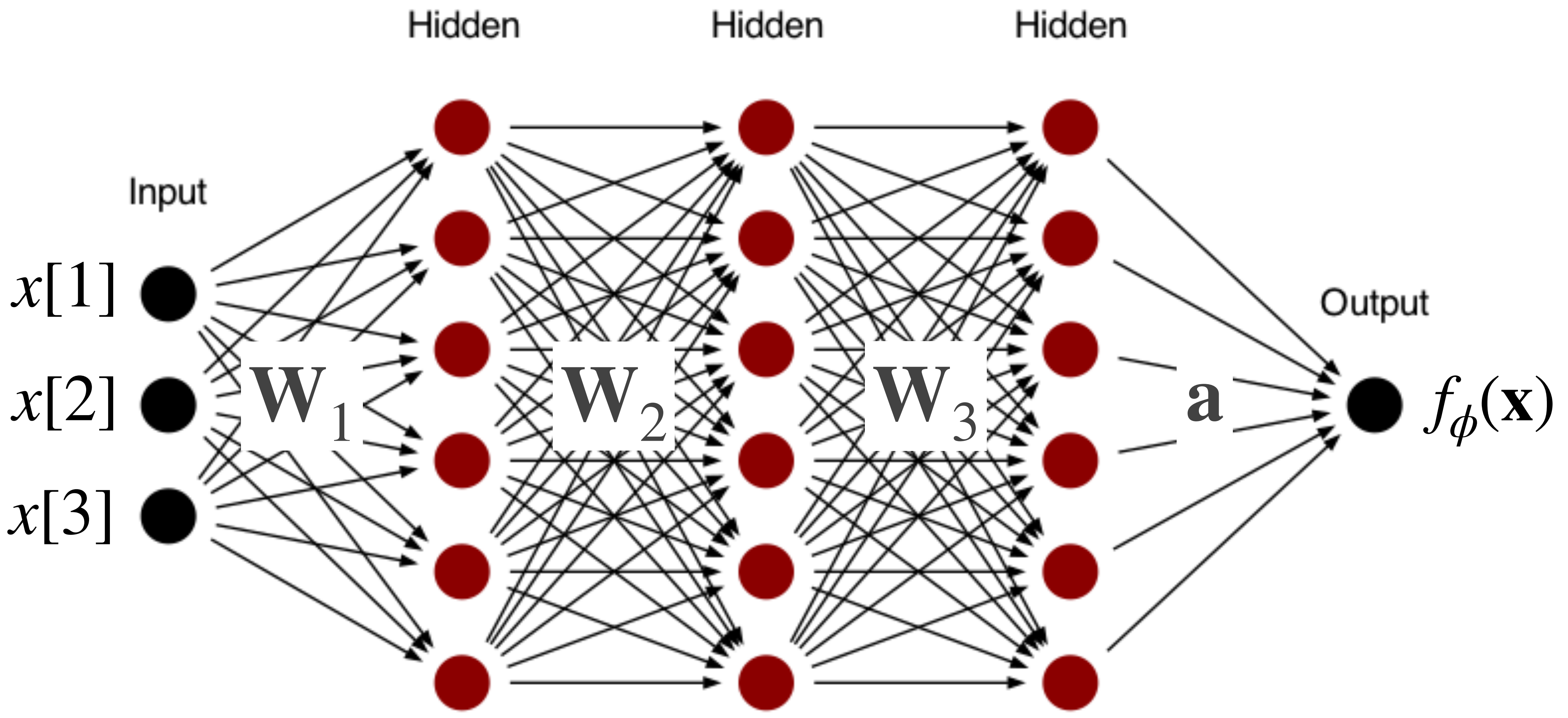
$$\phi = \left(\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{L-1}, \mathbf{a}\right)$$

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

Common choice for σ: $\mathrm{ReLU}(x) = \max(0,x)$

$$\hat{\phi}_S \in \arg\min_{\phi} \ \mathscr{L}_S(f_\phi)$$

Neural Networks



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

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

Common choice for σ: ReLU$(x) = \max(0,x)$

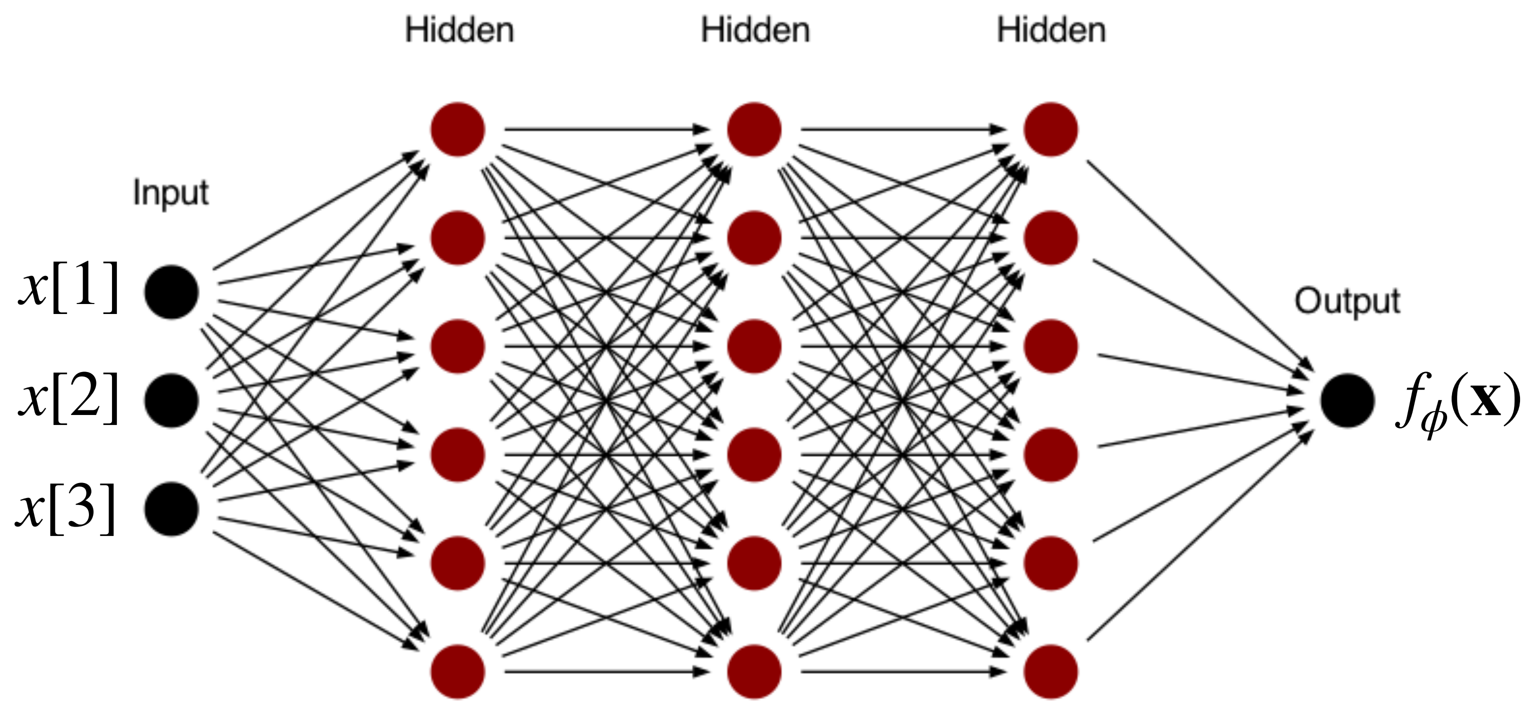
$$\hat{\phi}_S \in \arg\min_\phi \; \mathcal{L}_S(f_\phi) + \lambda C_L(\phi) \text{ where } C_L(\phi) = \frac{1}{L}\left(\sum_{\ell=1}^{L-1} \|\mathbf{W}_\ell\|_F^2 + \|\mathbf{a}\|_2^2\right)$$

"Weight Decay"

Function Space Perspective

"Weight Decay Cost"

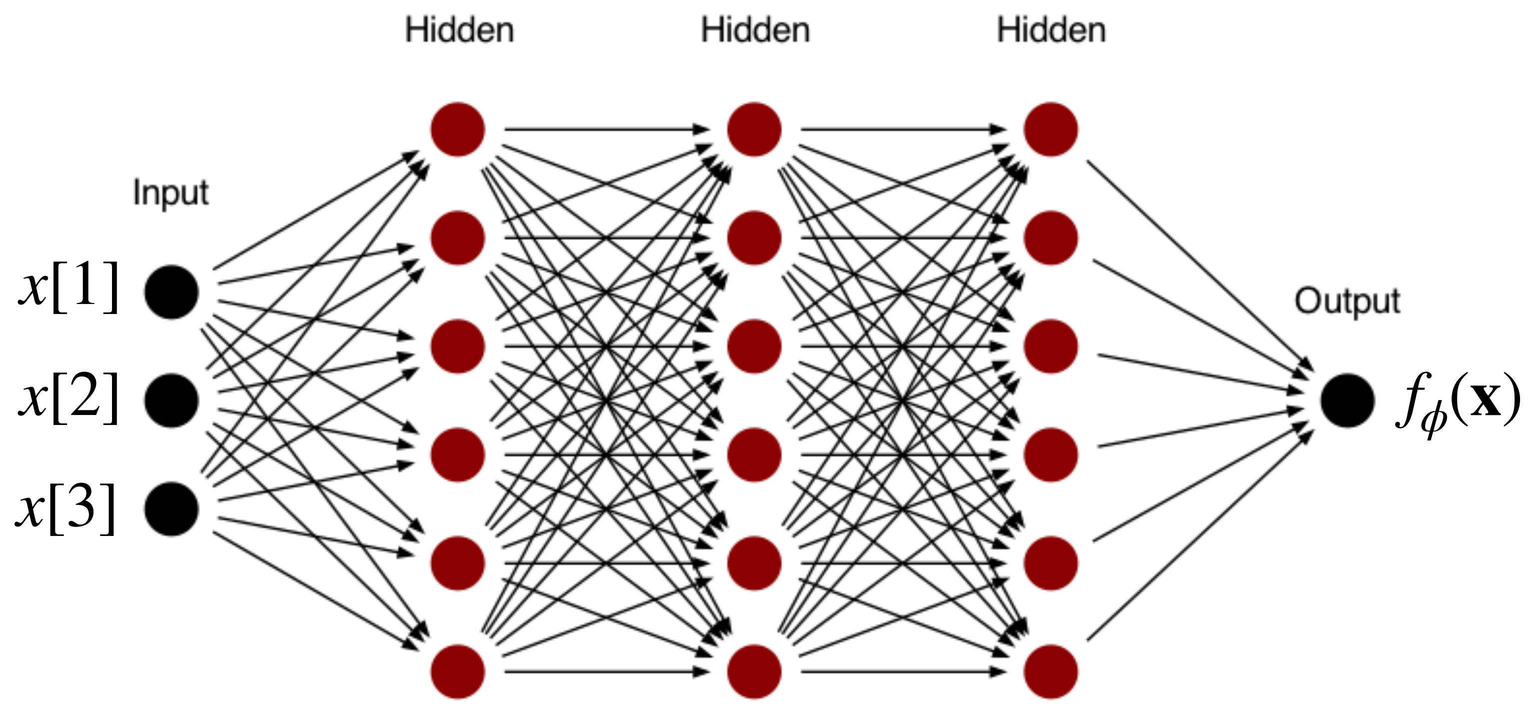
$$\hat{\phi}_S \in \arg\min_{\phi} \; \mathscr{L}_S(f_\phi) + \lambda C_L(\phi) \text{ where } C_L(\phi) = \frac{1}{L}\left(\sum_{\ell=1}^{L-1} \|\mathbf{W}_\ell\|_F^2 + \|\mathbf{a}\|_2^2\right)$$



Function Space Perspective

"Weight Decay Cost"

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

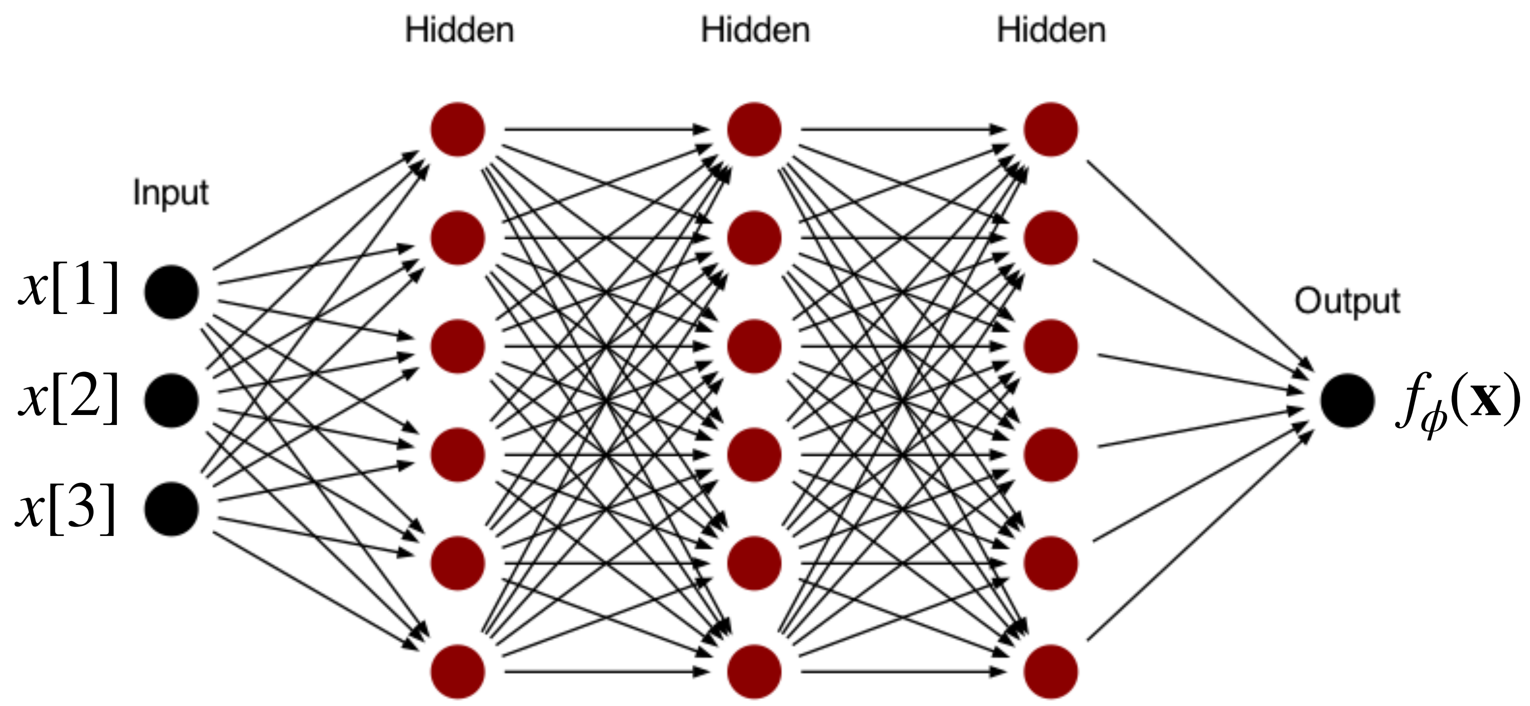


$$\mathscr{A}_L(S) = \hat{f}_S \in \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_L(g) \text{ where } \underbrace{R_L(g) = \inf_{\phi} C_L(\phi) \text{ s.t. } f_\phi = g}_{\text{"Representation Cost"}}$$

Function Space Perspective

"Weight Decay Cost"

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



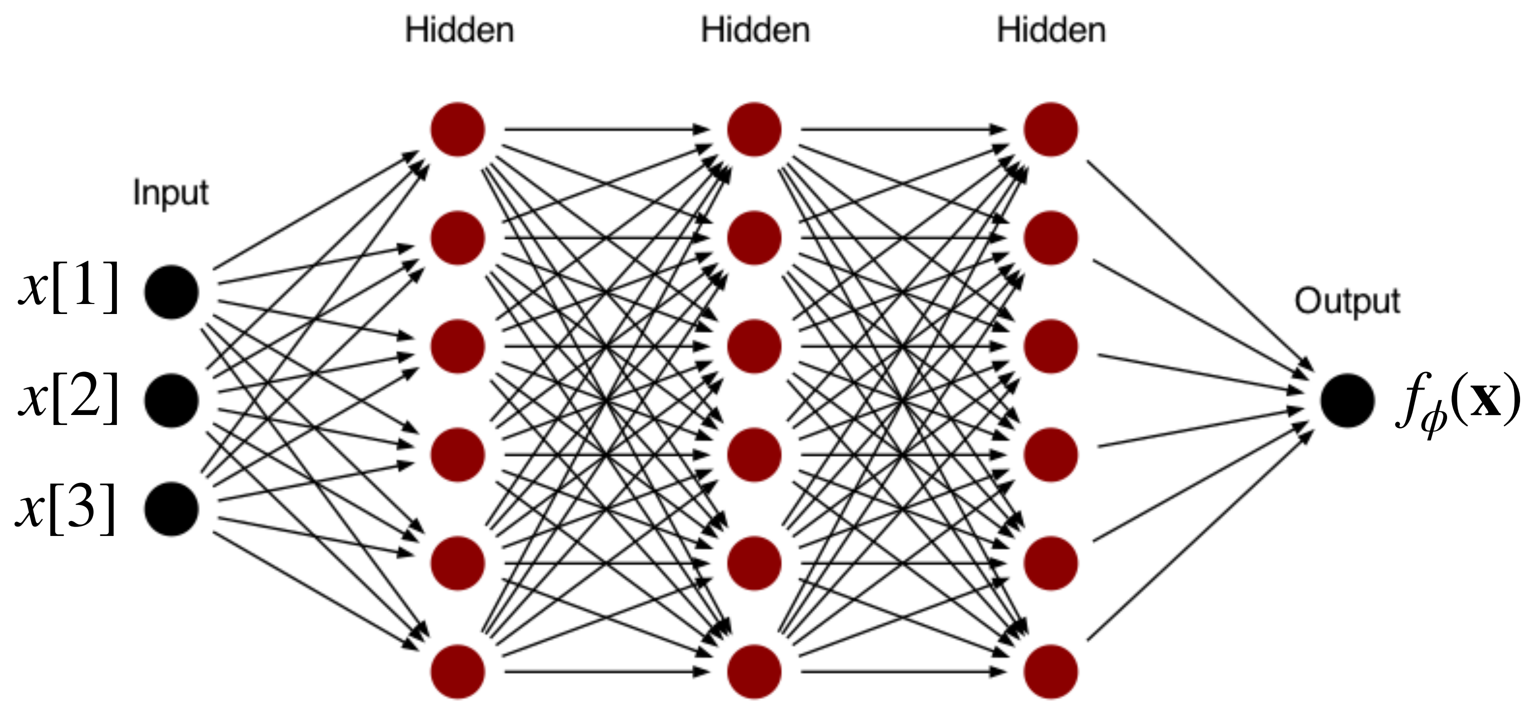
$$\mathscr{A}_L(S) = \hat{f}_S \in \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_L(g) \text{ where } \underbrace{R_L(g) = \inf_{\phi} C_L(\phi) \text{ s.t. } f_\phi = g}_{\text{“Representation Cost”}}$$

What kinds of functions have **small representation cost**?

Function Space Perspective

"Weight Decay Cost"

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



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

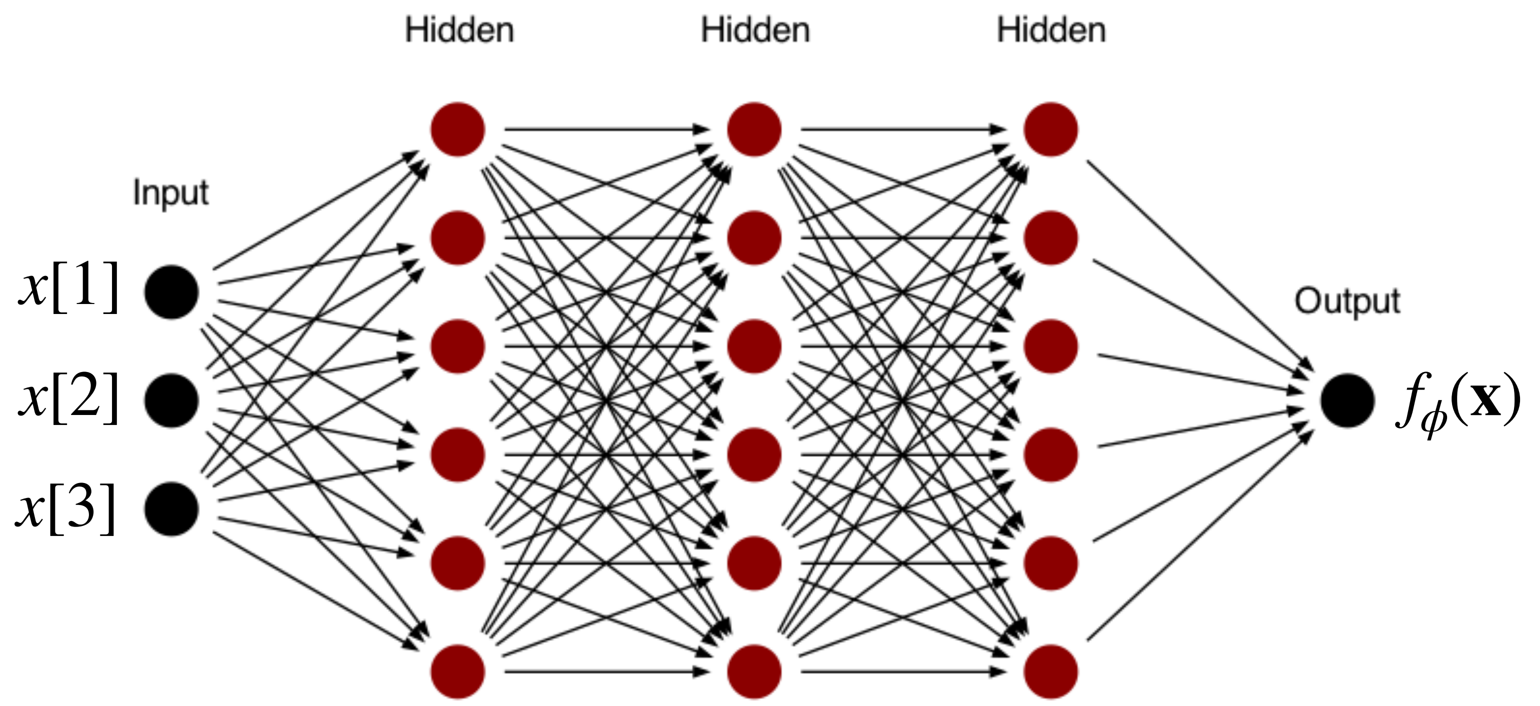
What kinds of functions have **small representation cost**?

How does the representation cost depend on **depth (L)**?

Function Space Perspective

"Weight Decay Cost"

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



$$\mathscr{A}_L(S) = \hat{f}_S \in \arg\min_{g \in \mathscr{N}_L} \mathscr{L}_S(g) + \lambda R_L(g) \text{ where } \underbrace{R_L(g) = \inf_{\phi} C_L(\phi) \text{ s.t. } f_\phi = g}$$

"Representation Cost"

What kinds of functions have **small representation cost**?

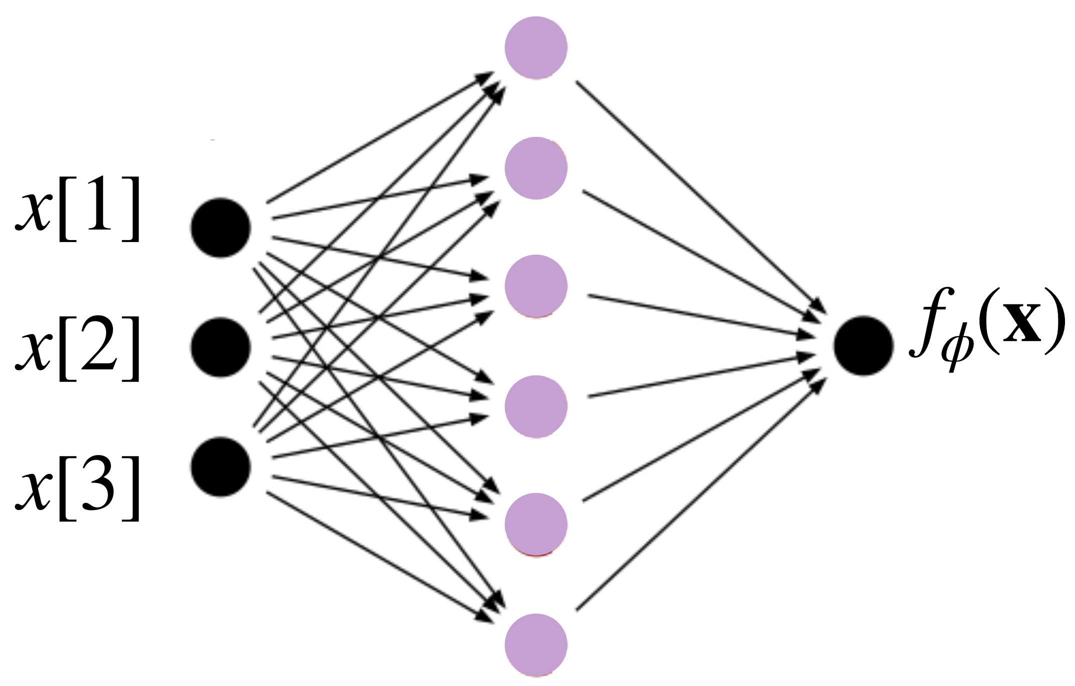
How does the representation cost depend on **depth (L)**?

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

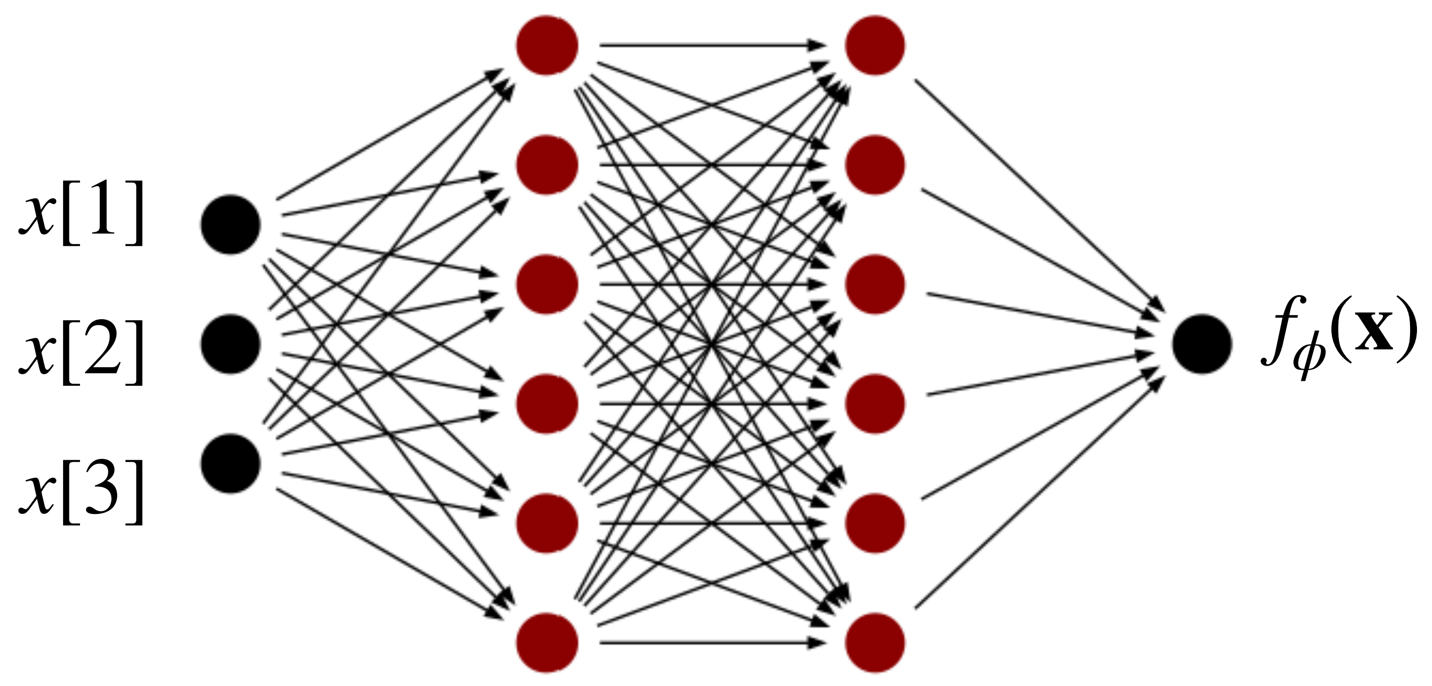
Are **deeper** neural networks better at **learning**?

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

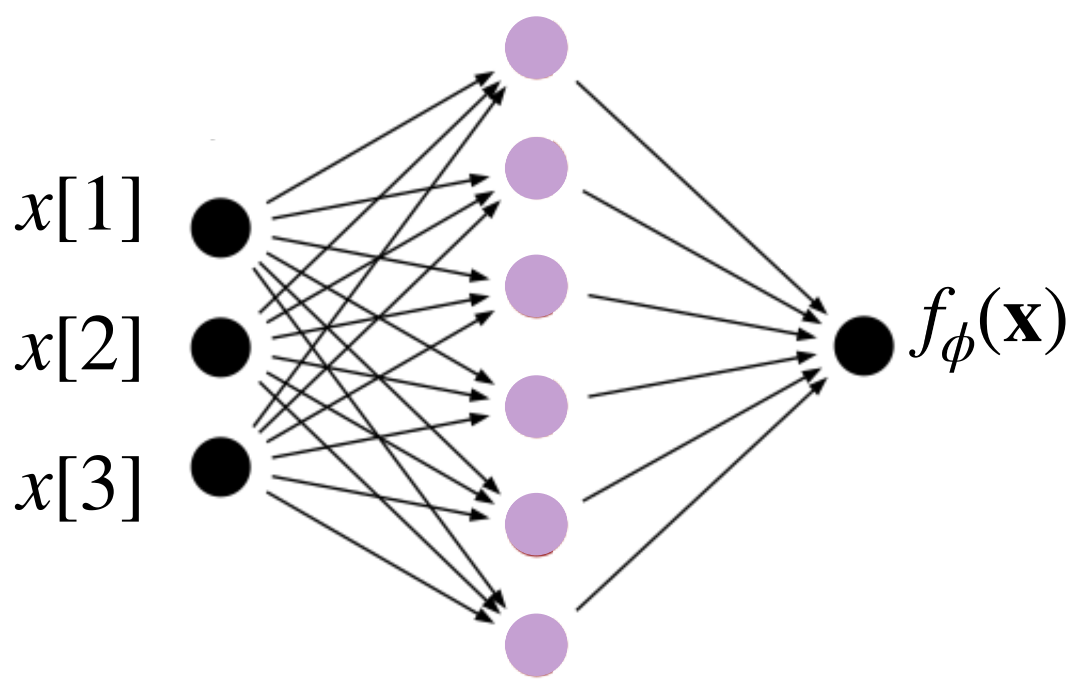
Depth-2 ReLU Network



Depth-3 ReLU Network

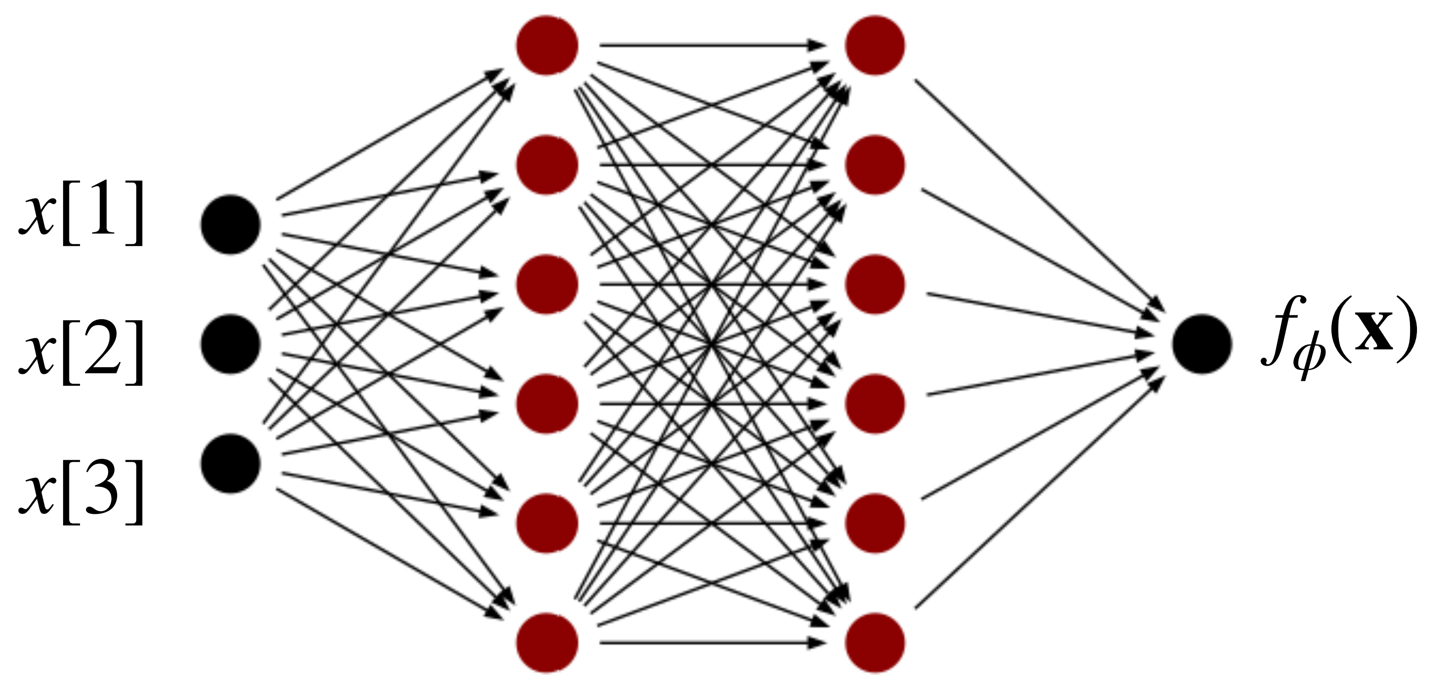


Depth-2 ReLU Network

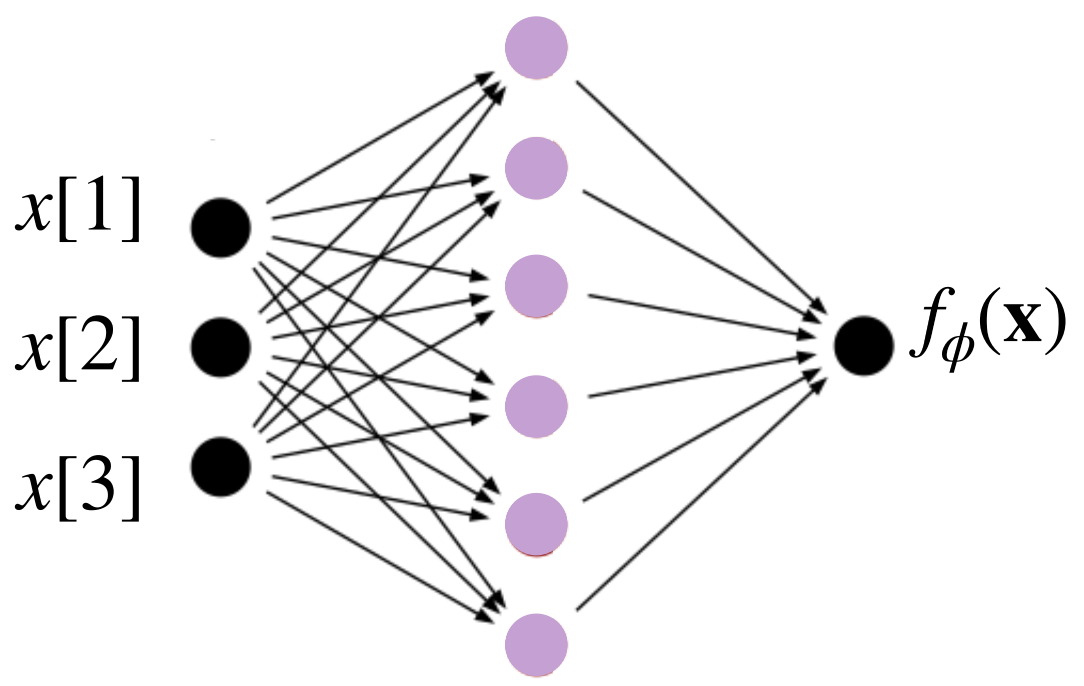


- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)

Depth-3 ReLU Network

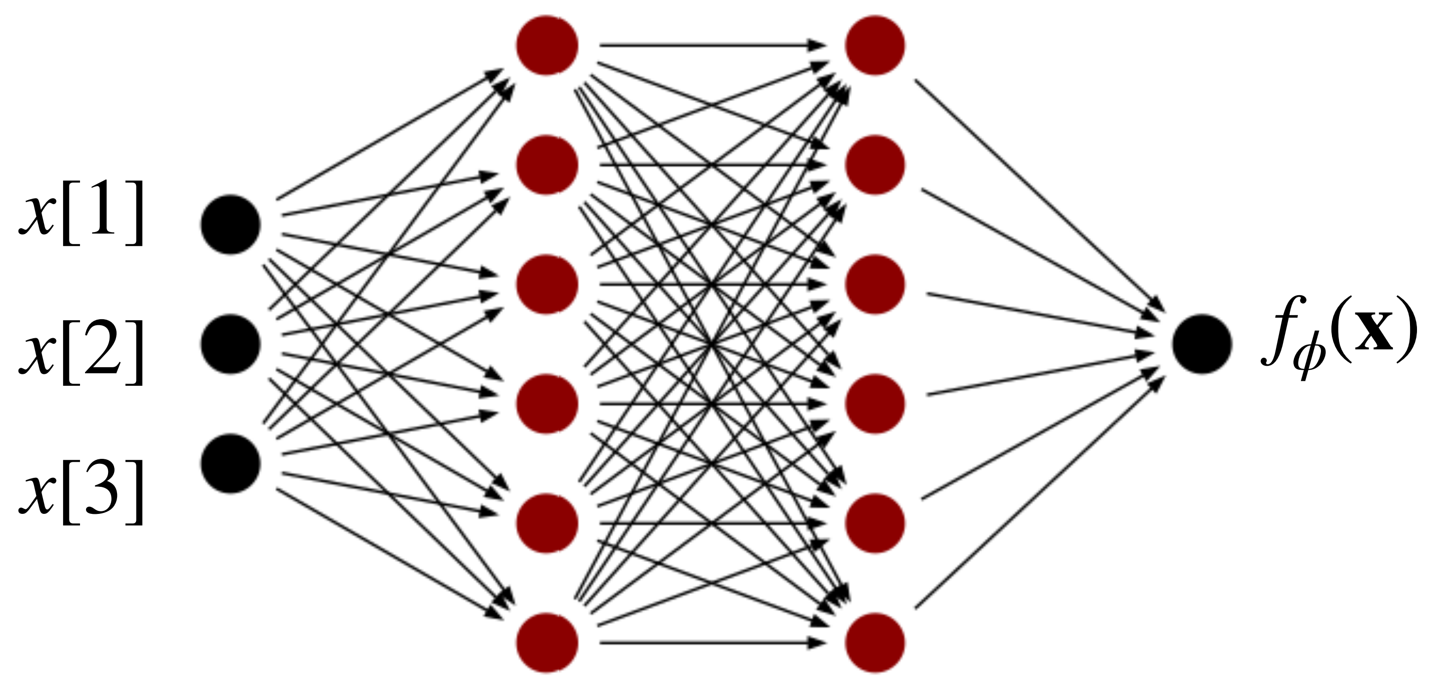


Depth-2 ReLU Network



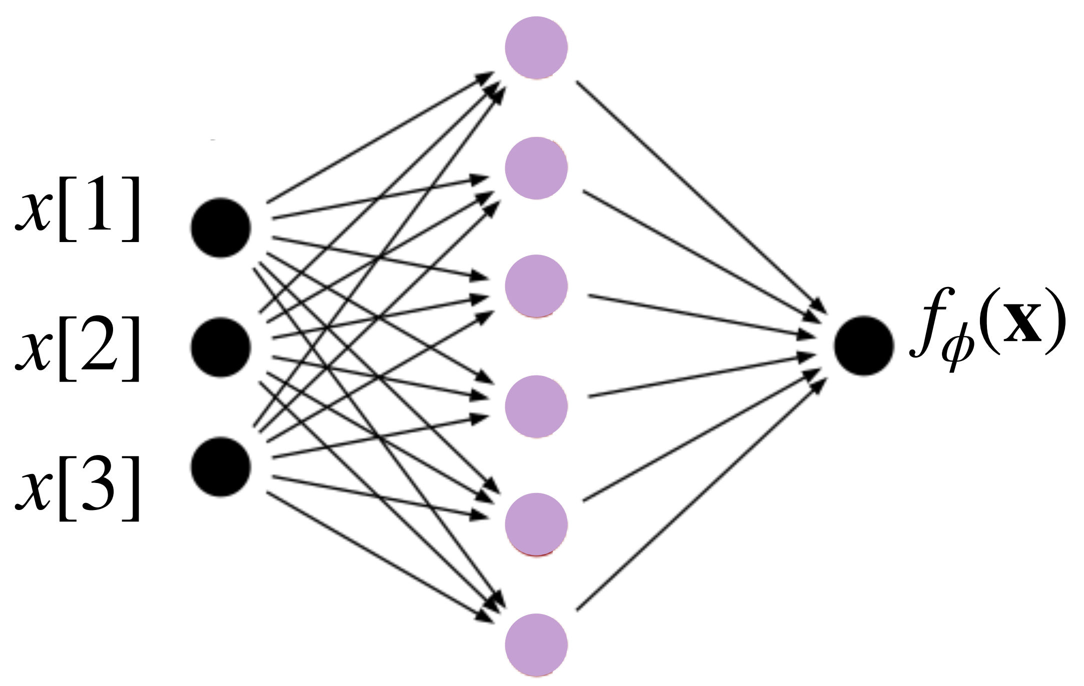
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)

Depth-3 ReLU Network



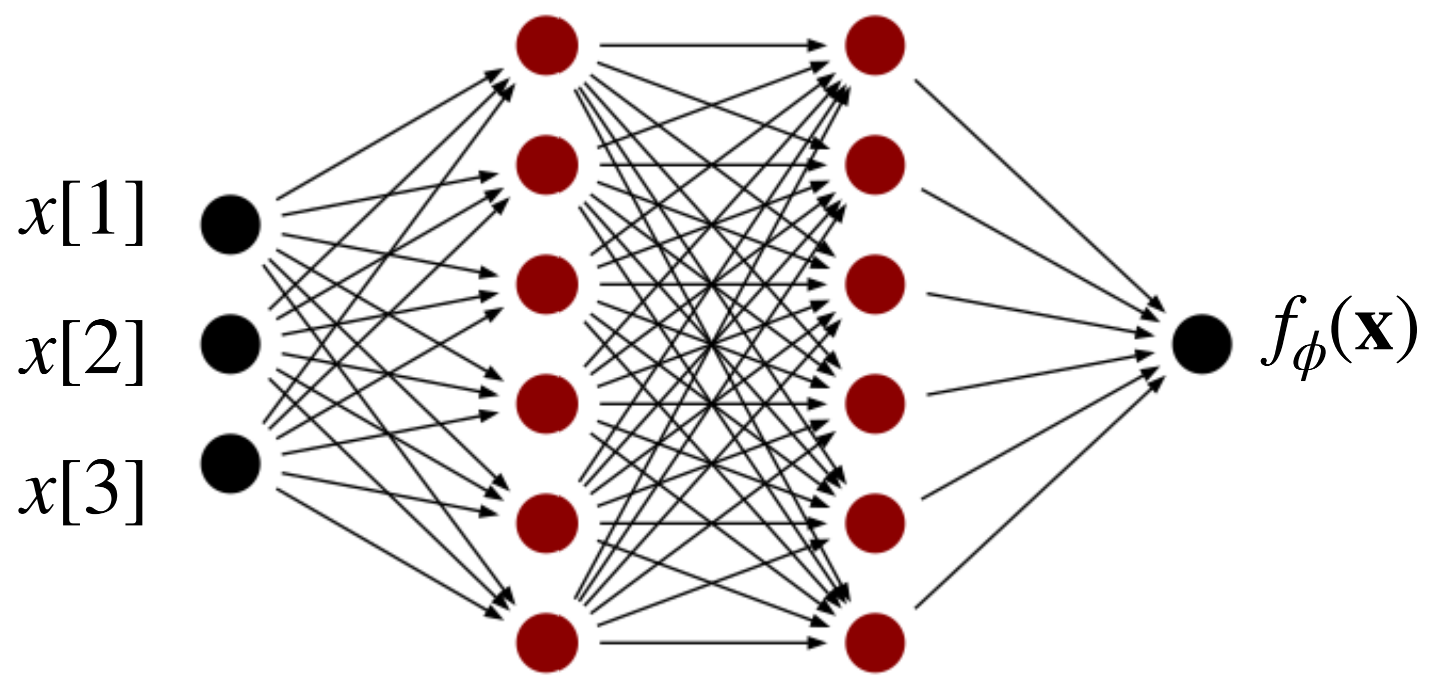
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)

Depth-2 ReLU Network



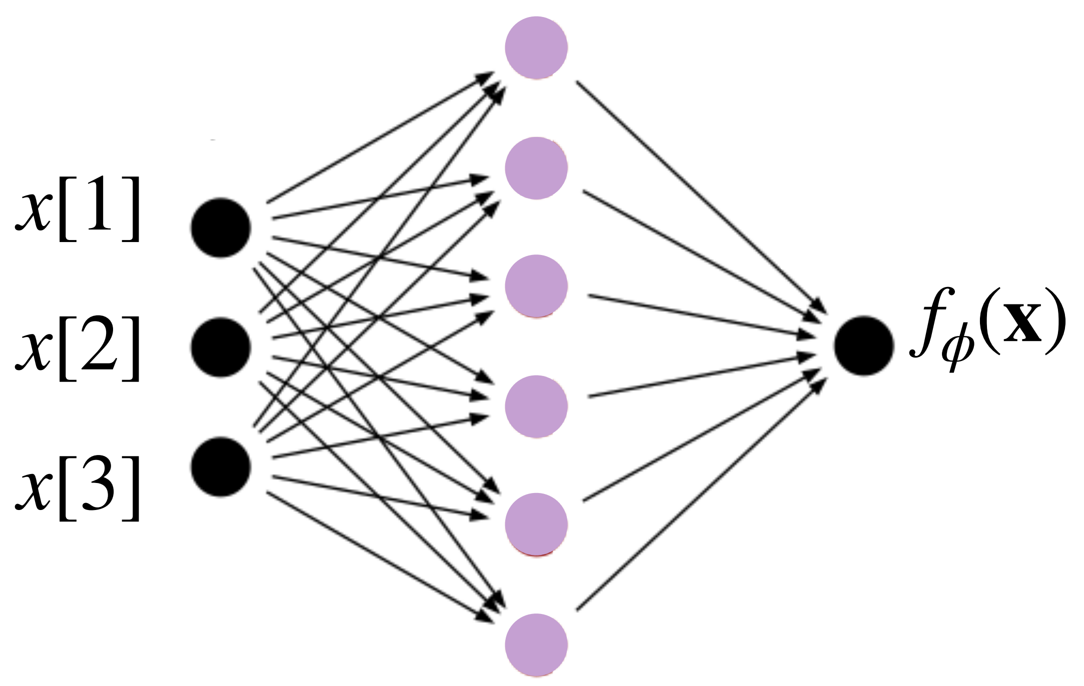
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)
- **Fewer parameters** = smaller model class

Depth-3 ReLU Network



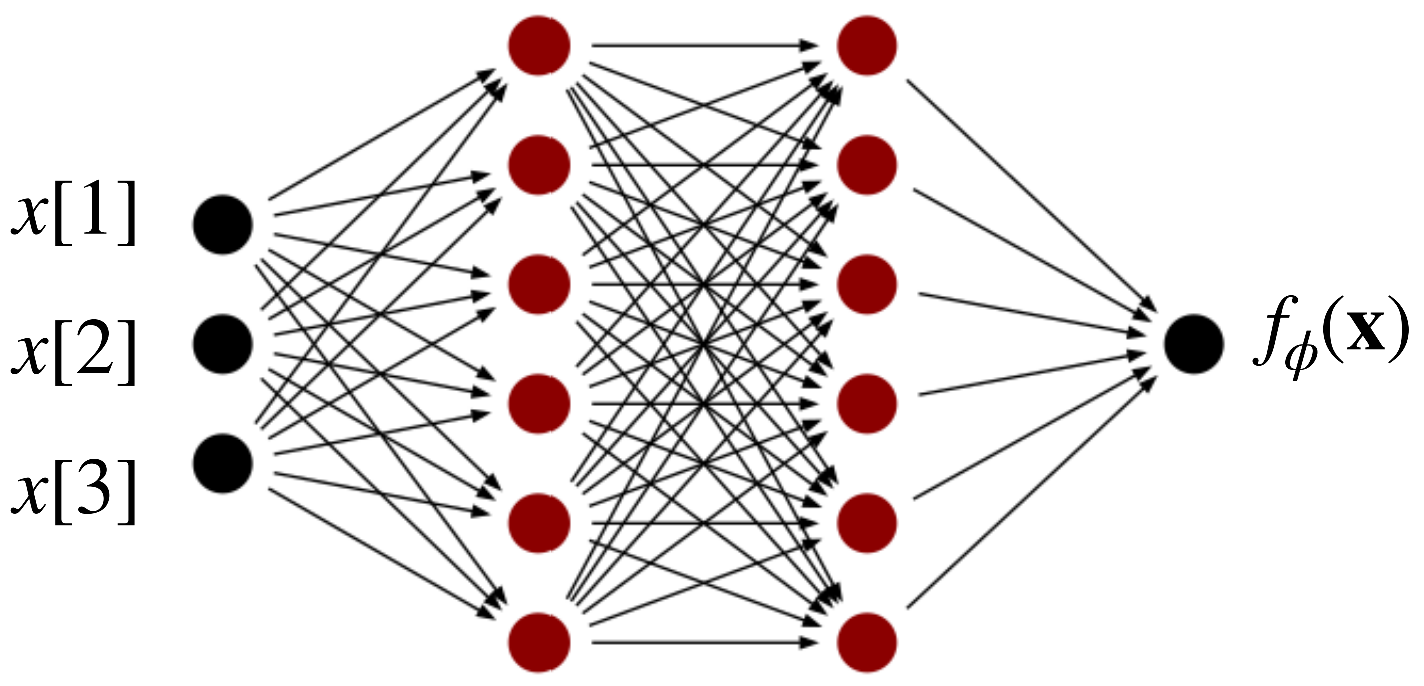
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)

Depth-2 ReLU Network



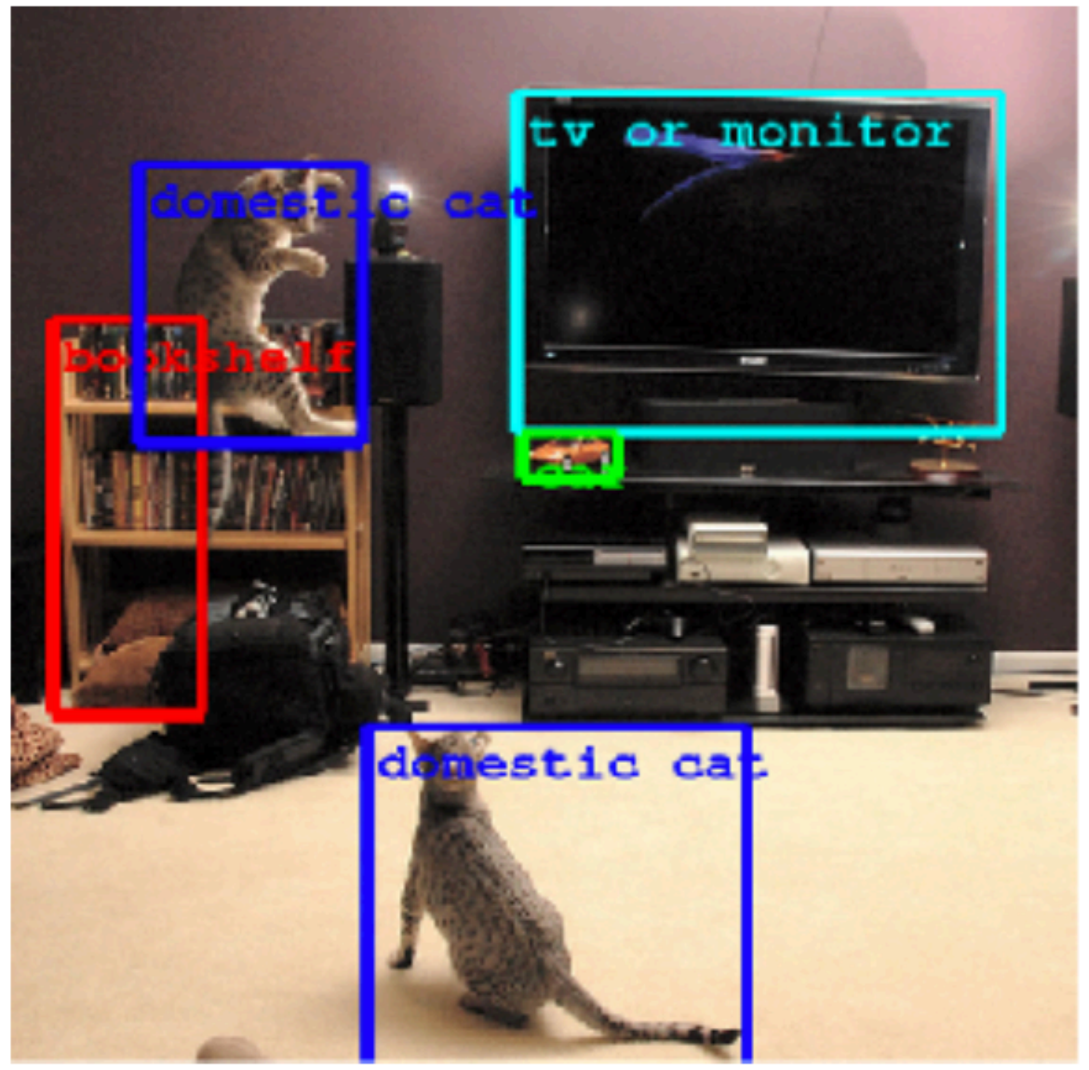
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)
- **Fewer parameters** = smaller model class

Depth-3 ReLU Network



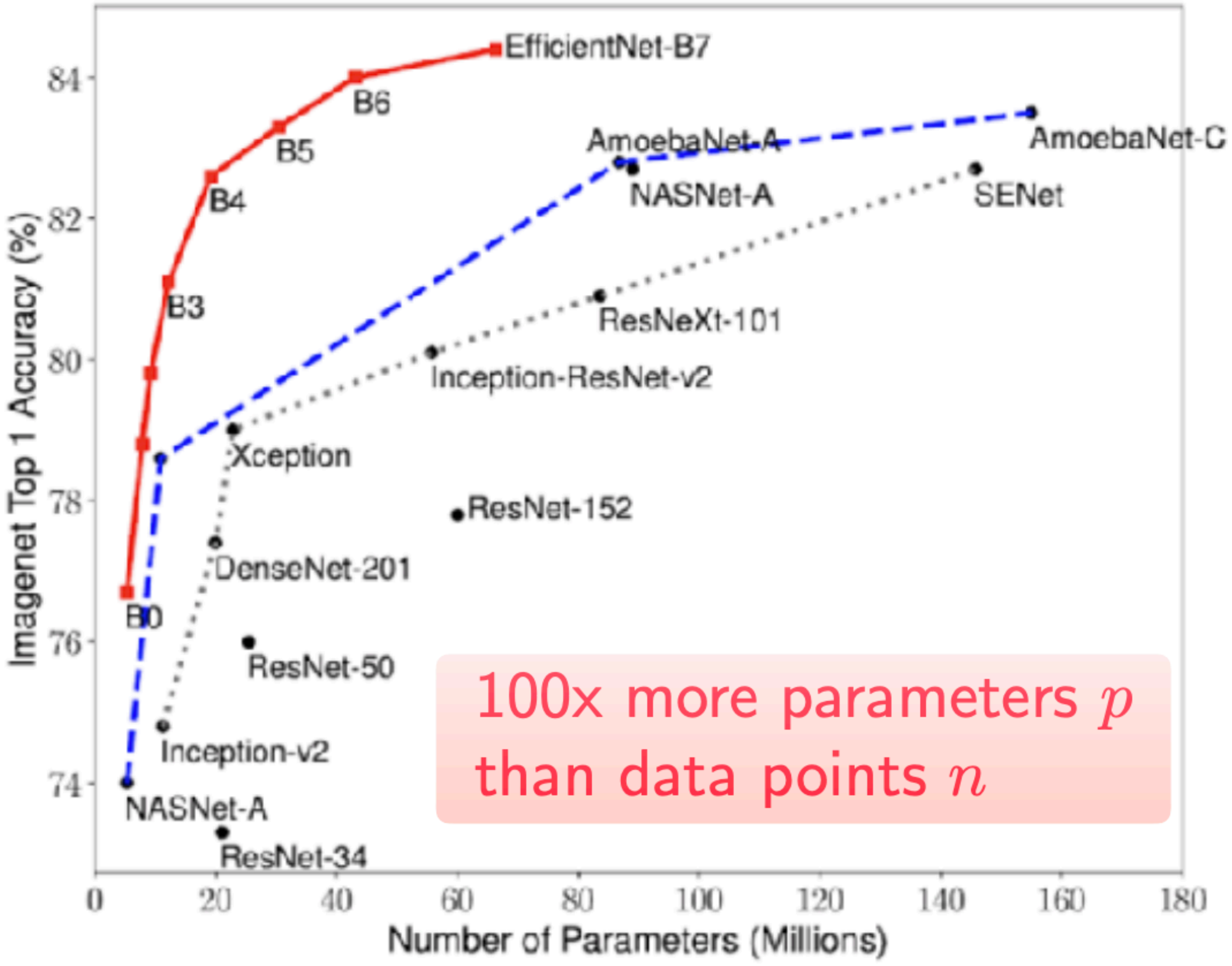
- **Universal approximator** of continuous functions with **arbitrary width**. Hornik (1991)
- **More parameters** = bigger model class

In lots of deep learning problems, bigger seems to be better



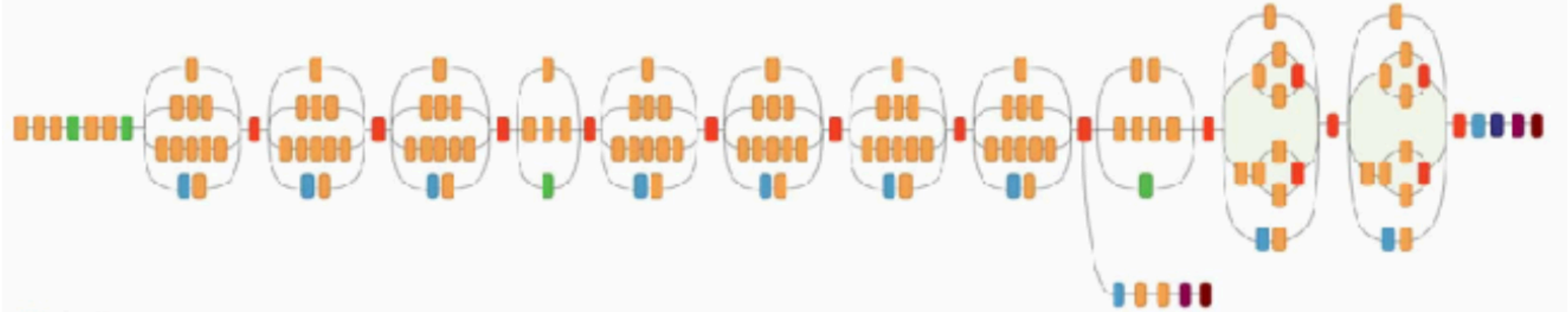
IMAGENET

parameters vs. classification error



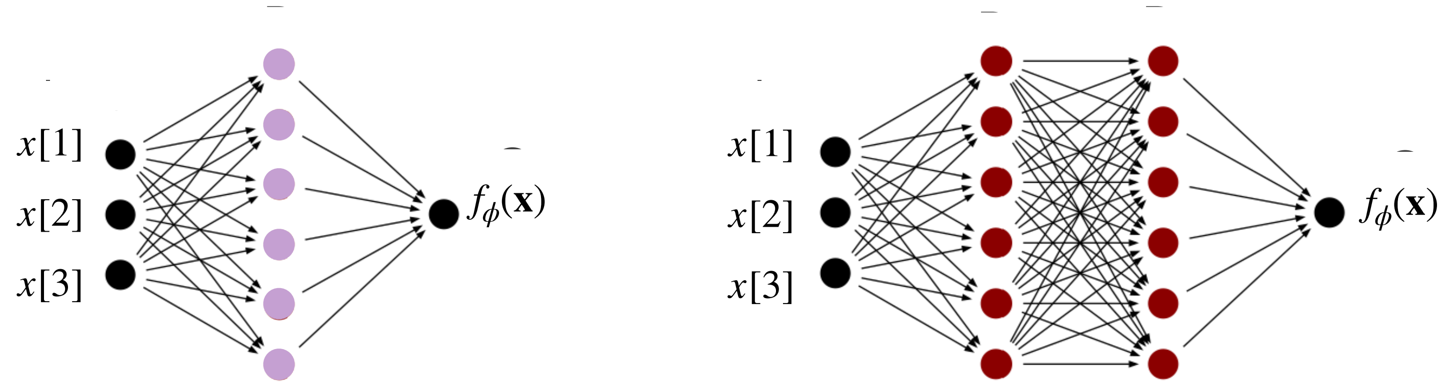
100x more parameters p
than data points n

Inception-ResNet-v2, 50-60 million parameters



Slide Credit: Rob Nowak

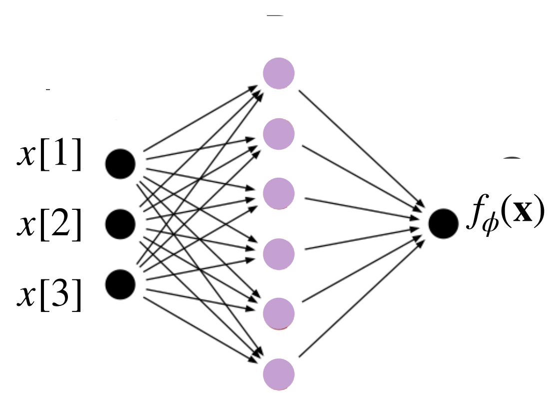
https://ai.googleblog.com/

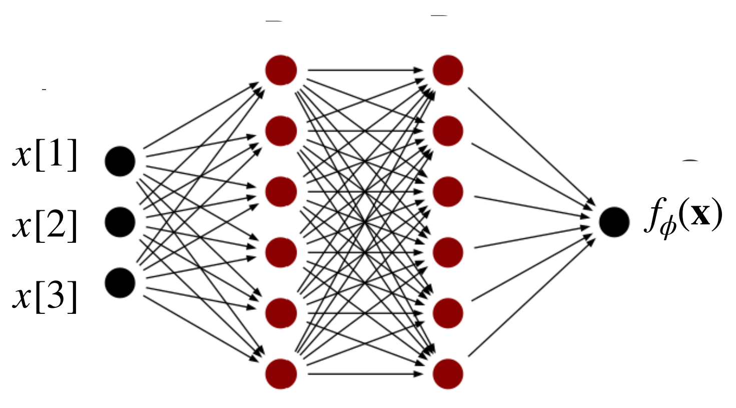


What if we measure model **size** in terms of **norm** of parameters instead of **number** of parameters?

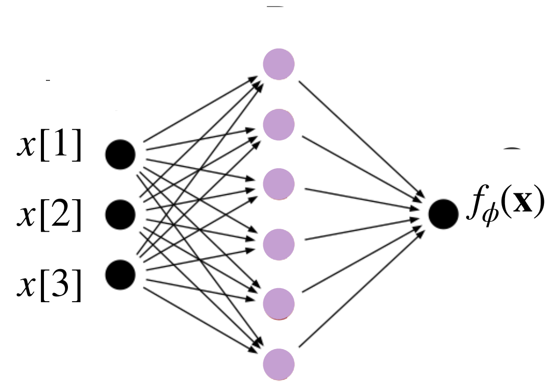
Bartlett 1996, Neyshabur, Tomioka & Srebro 2015

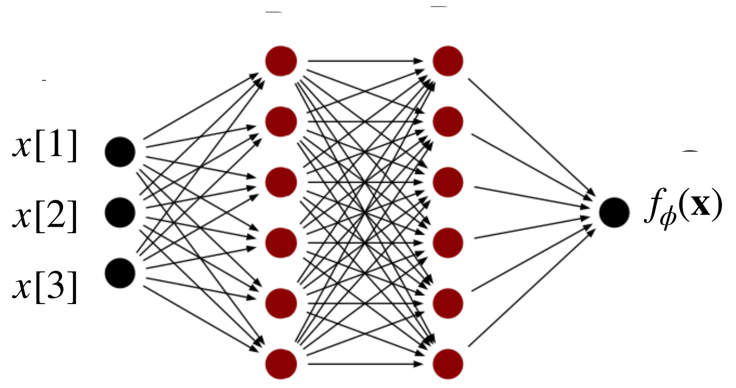
Depth Separation: Is depth 2 or 3 better?





Depth Separation: Is depth 2 or 3 better?

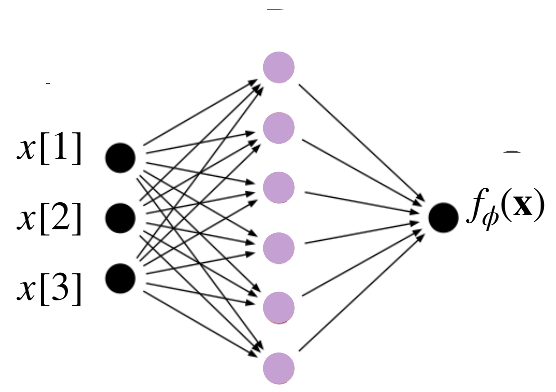


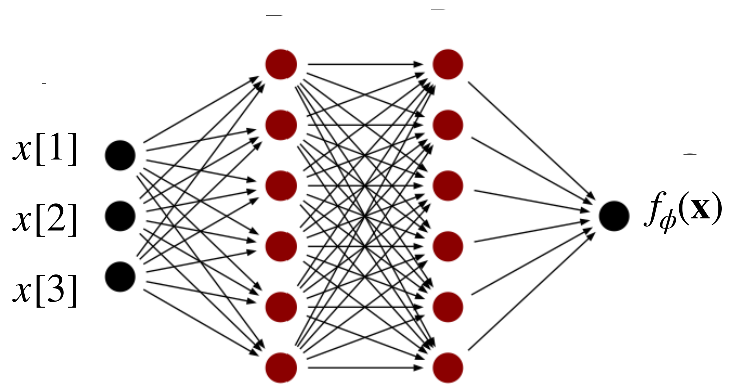


- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

Depth Separation: Is depth 2 or 3 better?



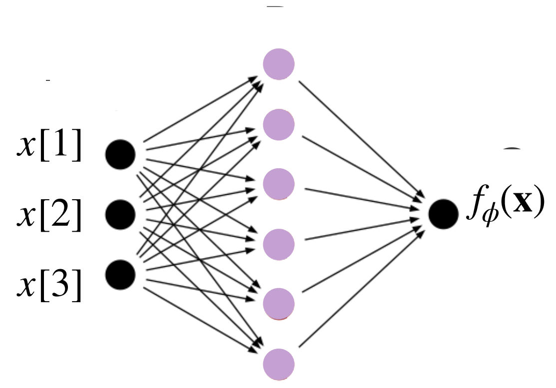


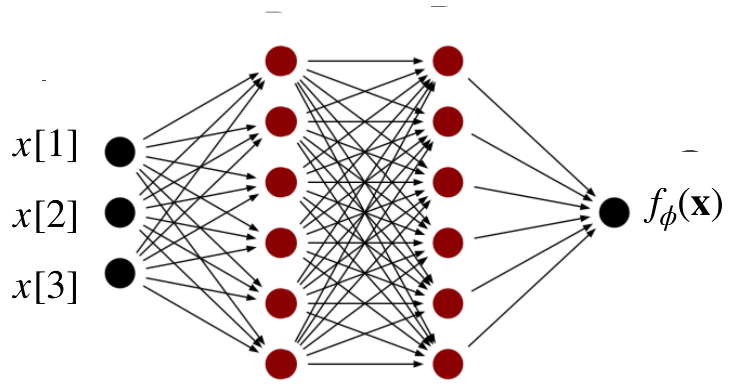
- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

- **In learning/generalization:**

Depth Separation: Is depth 2 or 3 better?





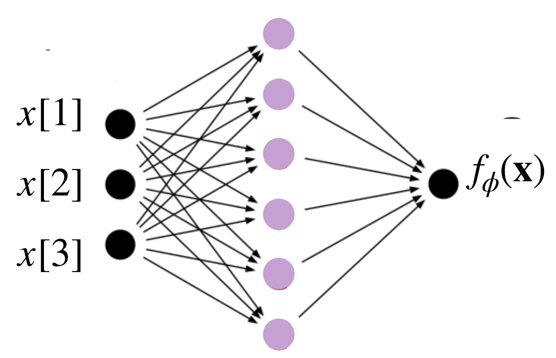
- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

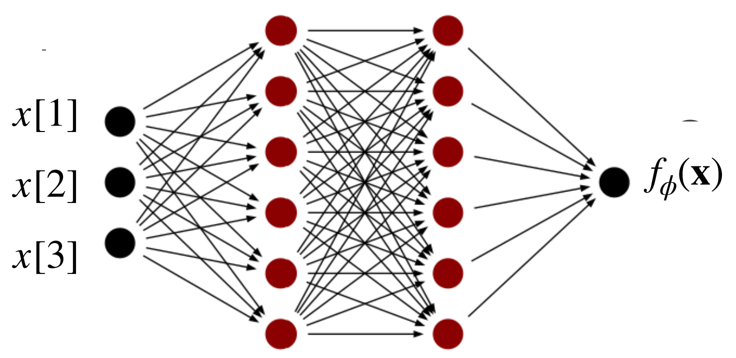
Eldan & Shamir (2016), Daniely (2017), Safran et al. (2021)

- **In learning/generalization:**
  - Depth-2 vs. Depth 3 learning rules:

$$\mathcal{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathcal{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathcal{A}_3(S) \in \arg\min_{g \in \mathcal{N}_3} \mathcal{L}_S(g) + \lambda_3 R_3(g)$$

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth **2** but only **polynomial width** with depth **3** to be **approximated**.

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

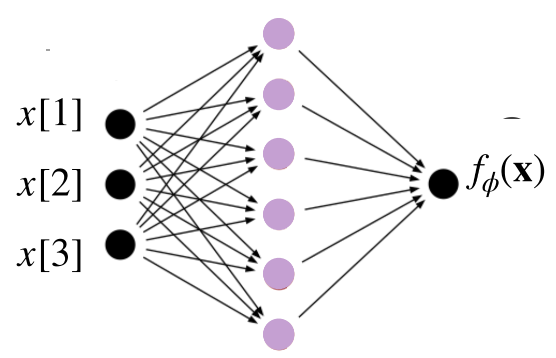
- **In learning/generalization:**
  - Depth-**2** vs. Depth **3** learning rules:

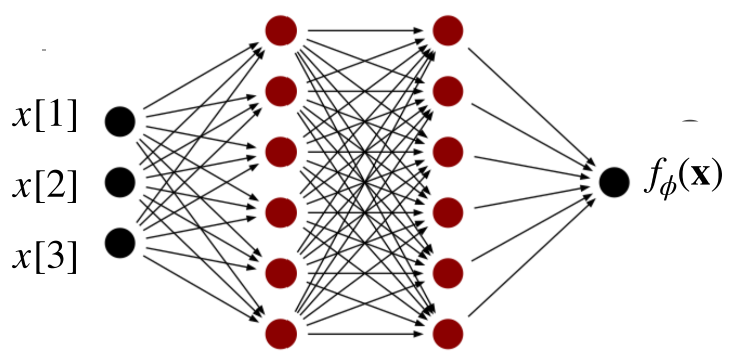
$$\mathscr{A}_2(S) \in \arg\min_{g \in \mathscr{N}_2} \mathscr{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathscr{A}_3(S) \in \arg\min_{g \in \mathscr{N}_3} \mathscr{L}_S(g) + \lambda_3 R_3(g)$$

  - Distributions:

$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth **2** but only **polynomial width** with depth **3** to be **approximated**.

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

- **In learning/generalization:**
  - Depth-**2** vs. Depth **3** learning rules:

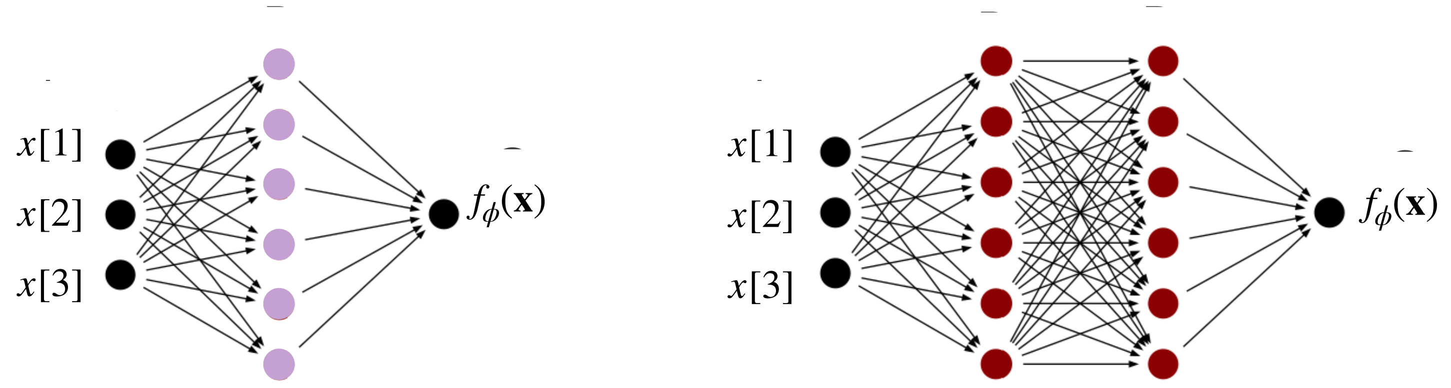
$$\mathscr{A}_2(S) \in \arg\min_{g \in \mathscr{N}_2} \mathscr{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathscr{A}_3(S) \in \arg\min_{g \in \mathscr{N}_3} \mathscr{L}_S(g) + \lambda_3 R_3(g)$$

  - Distributions:

$$\mathbf{x} \sim \mathrm{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

  - Are there functions f that require…

Depth Separation: Is depth 2 or 3 better?



- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

- **In learning/generalization:**
  - Depth-2 vs. Depth 3 learning rules:

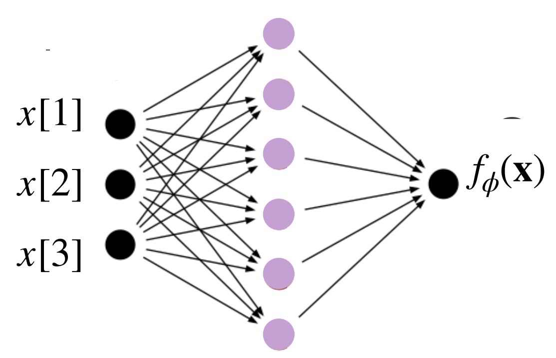
$$\mathscr{A}_2(S) \in \arg\min_{g \in \mathscr{N}_2} \mathscr{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathscr{A}_3(S) \in \arg\min_{g \in \mathscr{N}_3} \mathscr{L}_S(g) + \lambda_3 R_3(g)$$

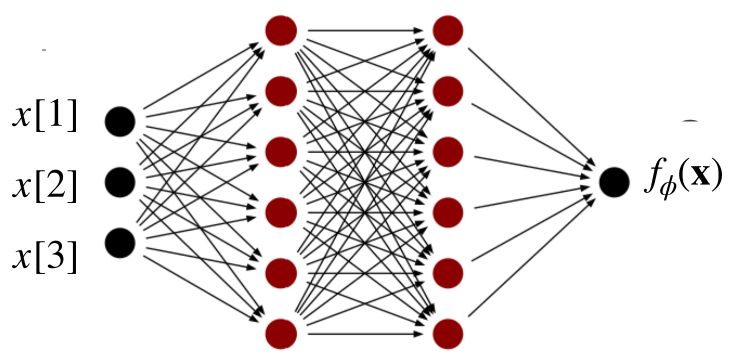
  - Distributions:

$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

  - Are there functions f that require…
    1. **exponential sample complexity** with depth 2 but only **polynomial sample complexity** with depth 3 to be **learned**?

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

- **In learning/generalization:**
  - Depth-2 vs. Depth 3 learning rules:

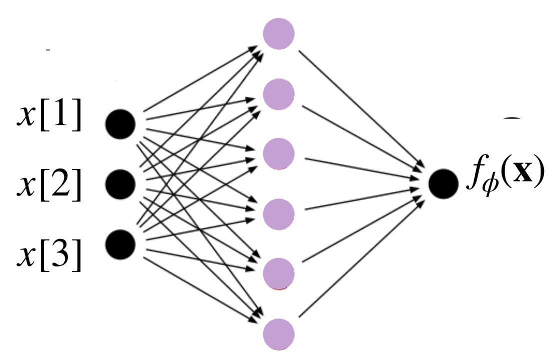
$$\mathscr{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathscr{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathscr{A}_3(S) \in \arg\min_{g \in \mathcal{N}_3} \mathscr{L}_S(g) + \lambda_3 R_3(g)$$

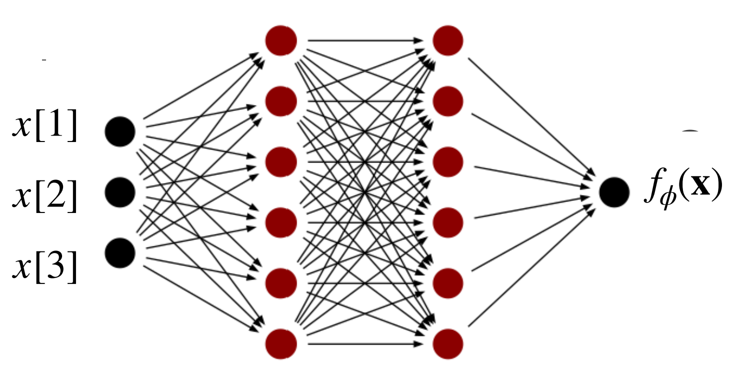
  - Distributions:

$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

  - Are there functions f that require…
    1. **exponential sample complexity** with depth 2 but only **polynomial sample complexity** with depth 3 to be **learned**?
    2. only **polynomial sample complexity** with depth 2 but **exponential sample complexity** with depth 3 to be **learned**?

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

- **In learning/generalization:**
  - Depth-2 vs. Depth 3 learning rules:

$$\mathcal{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathcal{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathcal{A}_3(S) \in \arg\min_{g \in \mathcal{N}_3} \mathcal{L}_S(g) + \lambda_3 R_3(g)$$

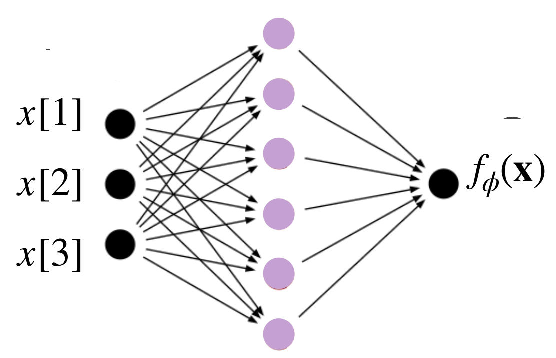
  - Distributions:

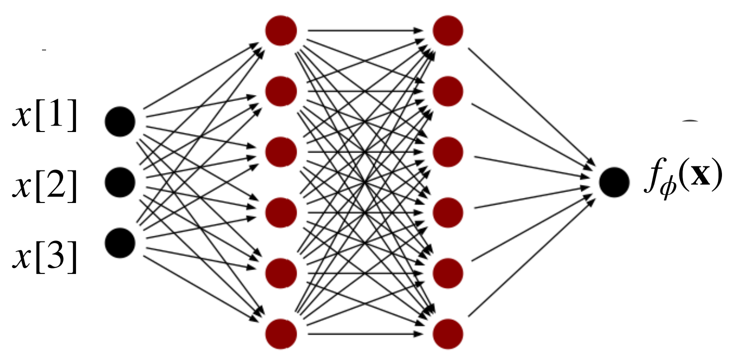
$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

  - Are there functions f that require…
    1. **exponential sample complexity** with depth 2 but only **polynomial sample complexity** with depth 3 to be **learned**?
    2. only **polynomial sample complexity** with depth 2 but **exponential sample complexity** with depth 3 to be **learned**?

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

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth 2 but only **polynomial width** with depth 3 to be **approximated**.

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

- **In learning/generalization:**
  - Depth-2 vs. Depth 3 learning rules:

$$\mathcal{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathcal{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathcal{A}_3(S) \in \arg\min_{g \in \mathcal{N}_3} \mathcal{L}_S(g) + \lambda_3 R_3(g)$$

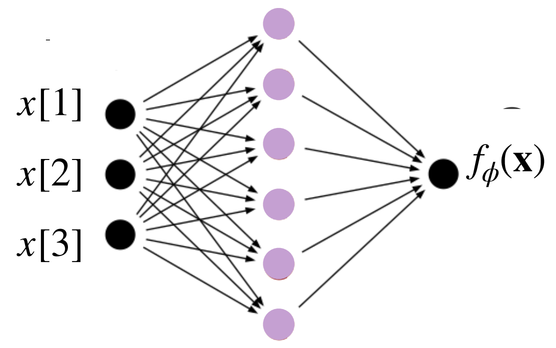
  - Distributions:

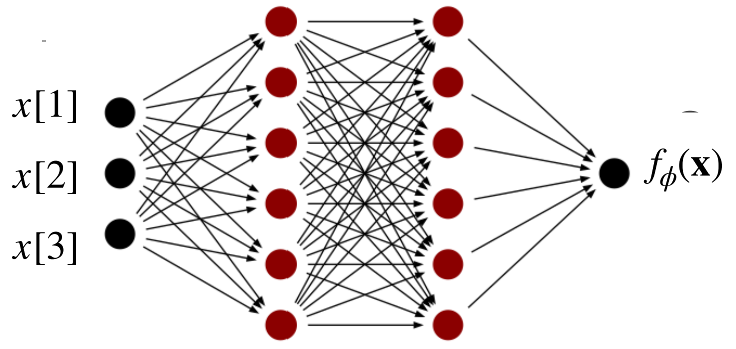
$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

  - Are there functions f that require…
    1. ✔ **exponential sample complexity** with depth 2 but only **polynomial sample complexity** with depth 3 to be **learned**?
    2. only **polynomial sample complexity** with depth 2 but **exponential sample complexity** with depth 3 to be **learned**?

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

Depth Separation: Is depth 2 or 3 better?





- **In approximation:**
  - There are functions f that require…
    - **exponential width** (in dimension) with depth **2** but only **polynomial width** with depth **3** to be **approximated**.

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

- **In learning/generalization:**
  - Depth-**2** vs. Depth **3** learning rules:

$$\mathcal{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathcal{L}_S(g) + \lambda_2 R_2(g) \quad \text{vs.} \quad \mathcal{A}_3(S) \in \arg\min_{g \in \mathcal{N}_3} \mathcal{L}_S(g) + \lambda_3 R_3(g)$$

  - Distributions:

$$\mathbf{x} \sim \text{Unif}(\mathbf{S}^{d-1} \times \mathbf{S}^{d-1})$$
$$y = f(\mathbf{x}) \in [-1,1]$$

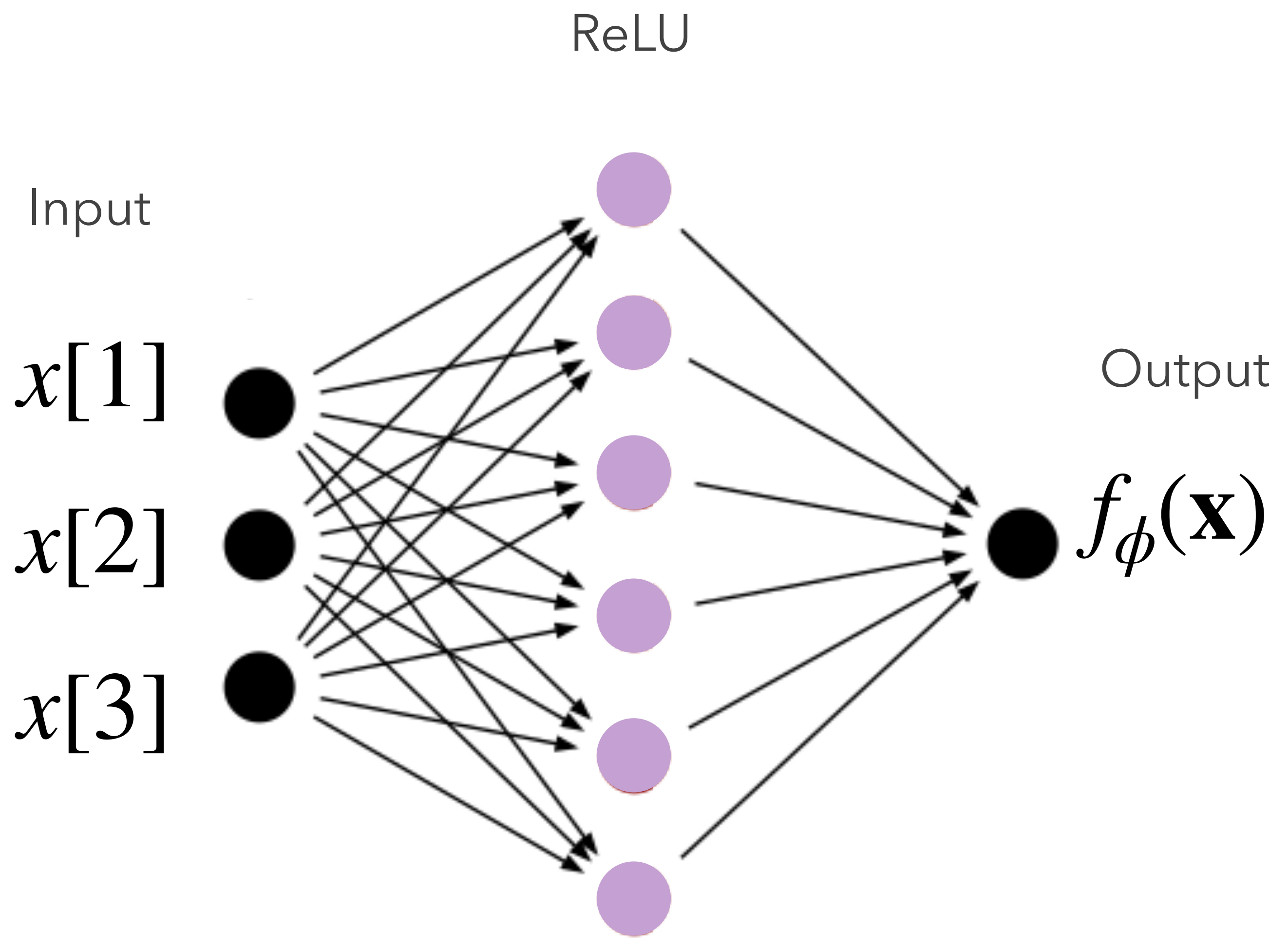
  - Are there functions f that require…
    1. ✔ **exponential sample complexity** with depth **2** but only **polynomial sample complexity** with depth **3** to be **learned**?
    2. ✘ only **polynomial sample complexity** with depth **2** but **exponential sample complexity** with depth **3** to be **learned**?

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

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

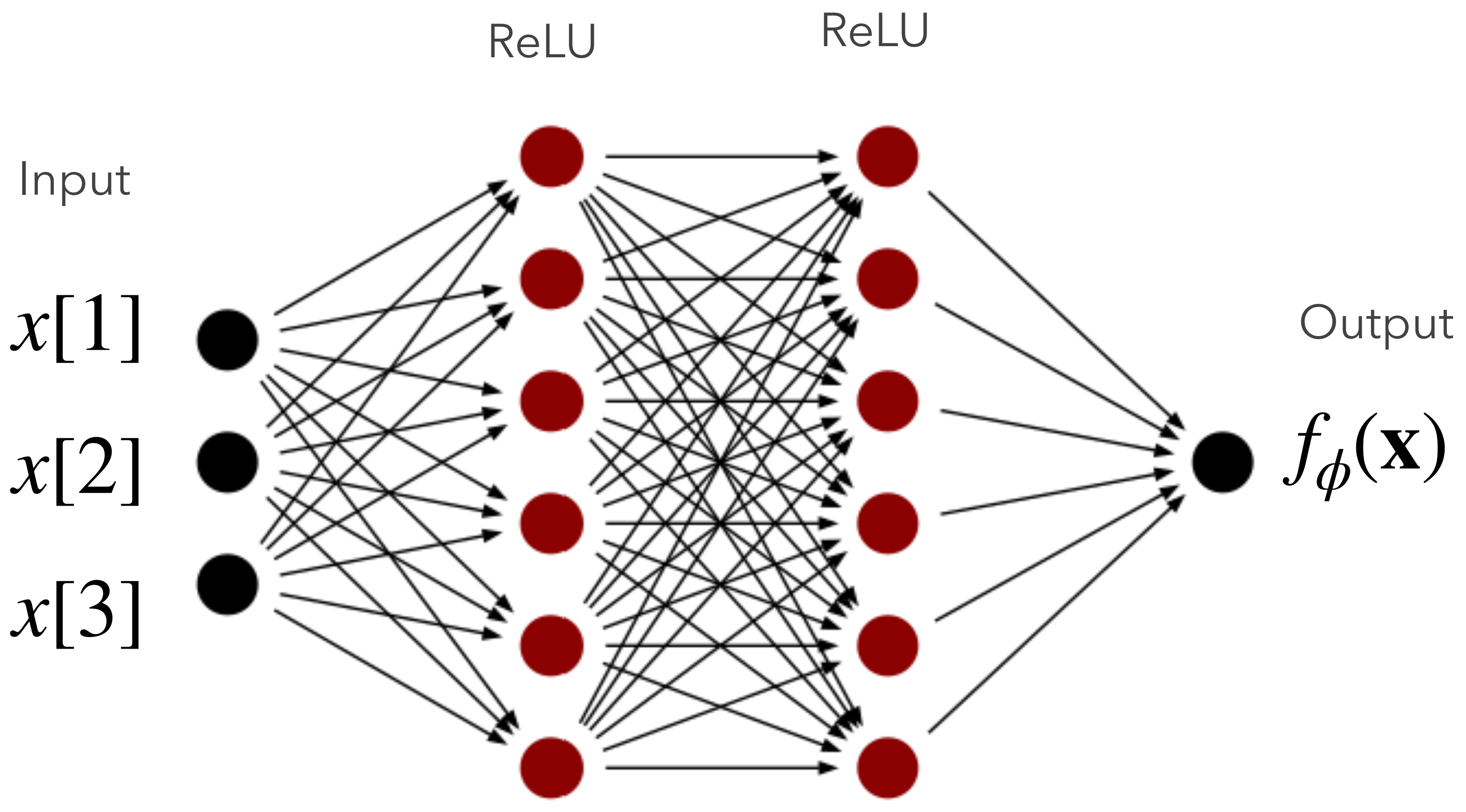
Key: Choose f so that…

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



Expensive

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



Cheap

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

Proof Sketch:

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

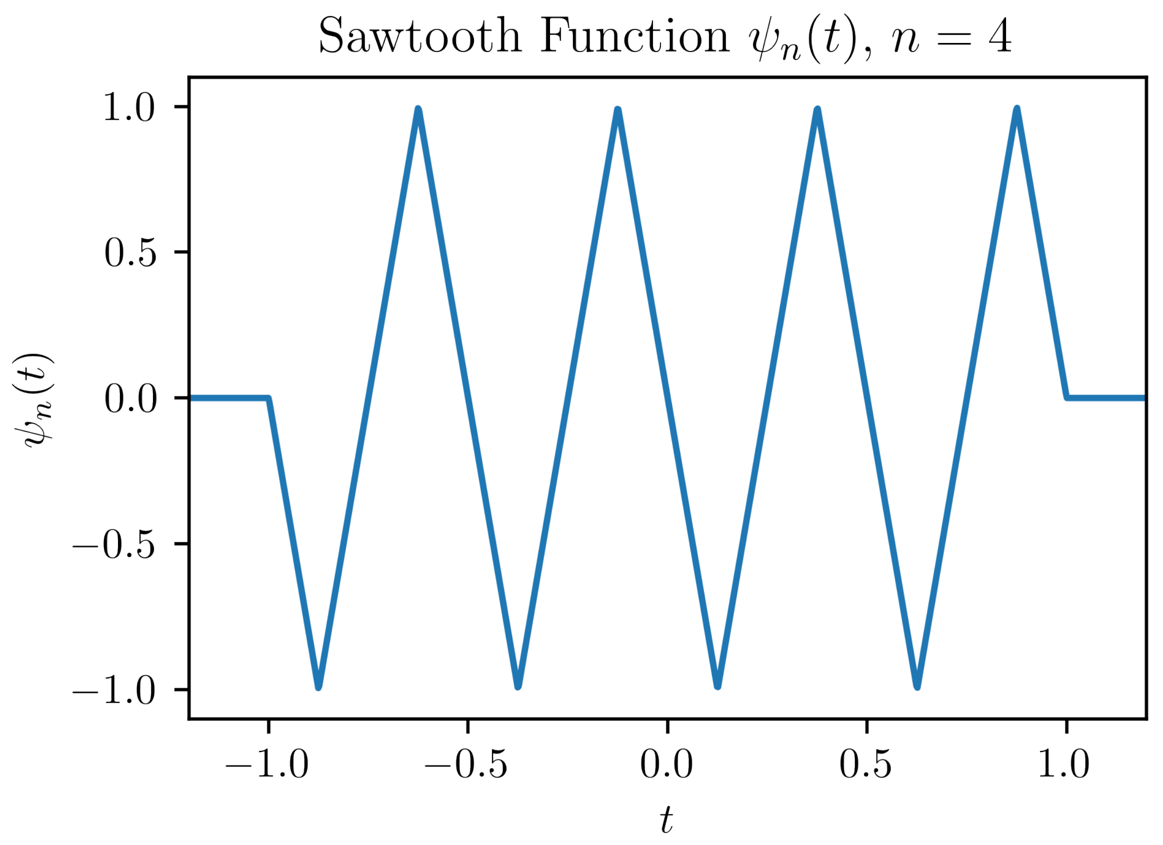
- A slight modification of Daniely (2017) construction for depth separation in width to approximate

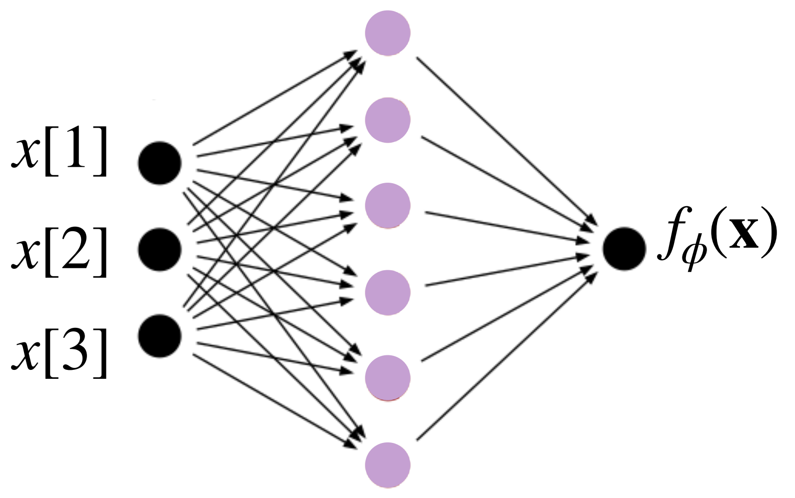
- Daniely showed that **depth 2** networks need to be very wide to approximate functions that are compositions of a function that is **very non-polynomial** with an **inner-product**

- Naturally approximated by a **depth 3** network…

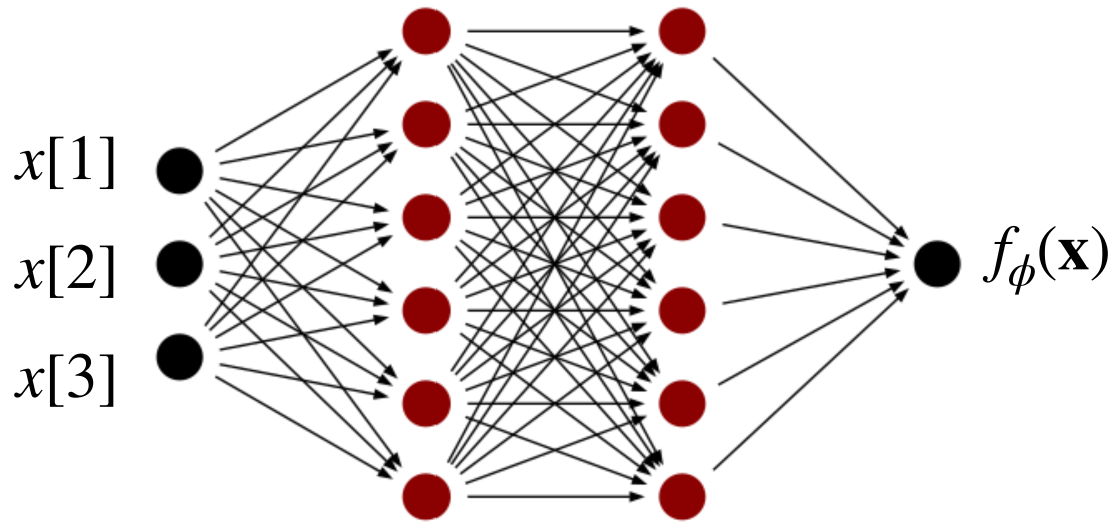
  - The inner product can be approximated with first hidden layer

  - Sawtooth function can be expressed *exactly* with second hidden layer



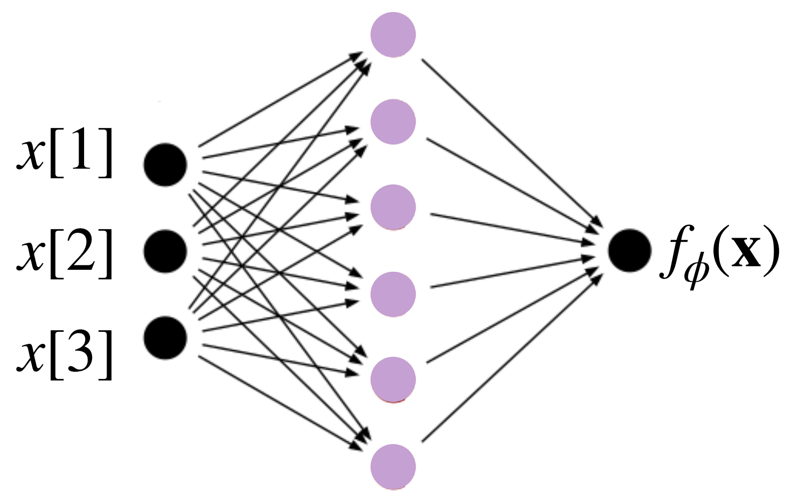


Expensive



Cheap

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



Expensive

Proof Sketch: ***"Hard"*** *with* $\mathscr{A}_2(S) \in \arg\min_{g \in \mathcal{N}_2} \mathscr{L}_S(g) + \lambda_2 R_2(g)$

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

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

- If $R_2(\mathscr{A}_2(S)) < 2^{\Omega(d)}$ then $\mathscr{L}_{\mathscr{D}}(\mathscr{A}_2(S)) \geq 10^{-4}$

- Therefore, $\mathscr{L}_{\mathscr{D}}(\mathscr{A}_2(S)) \geq 10^{-4}$ unless $|S| = 2^{\Omega(d)}$

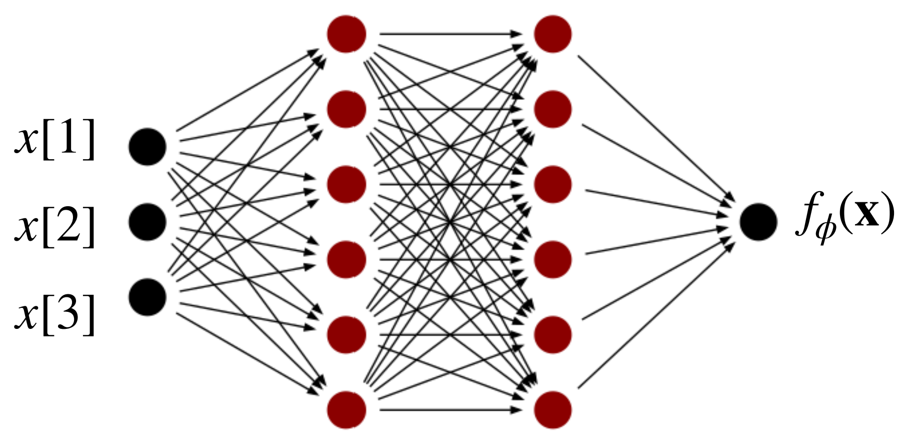
$$\mathscr{L}_S(\mathscr{A}_2(S)) + \lambda_2 R_2(\mathscr{A}_2(S)) \leq \mathscr{L}_S(\hat{f}) + \lambda_2 R_2(\hat{f})$$

$$\mathscr{L}_S(\mathscr{A}_2(S)) + \lambda_2 R_2(\mathscr{A}_2(S)) \leq \lambda_2 R_2(\hat{f})$$

$$\lambda_2 R_2(\mathscr{A}_2(S)) \leq \lambda_2 R_2(\hat{f})$$

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

Proof Sketch: "Easy" with $\mathscr{A}_3(S) \in \arg\min_{g \in \mathscr{N}_3} \mathscr{L}_S(g) + \lambda_3 R_3(g)$



Cheap

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

- Because of how we choose λ_3, we get $R_3(\mathscr{A}_3(S)) \le R_3(f_\varepsilon) \le \mathrm{poly}(d)$

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

- **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

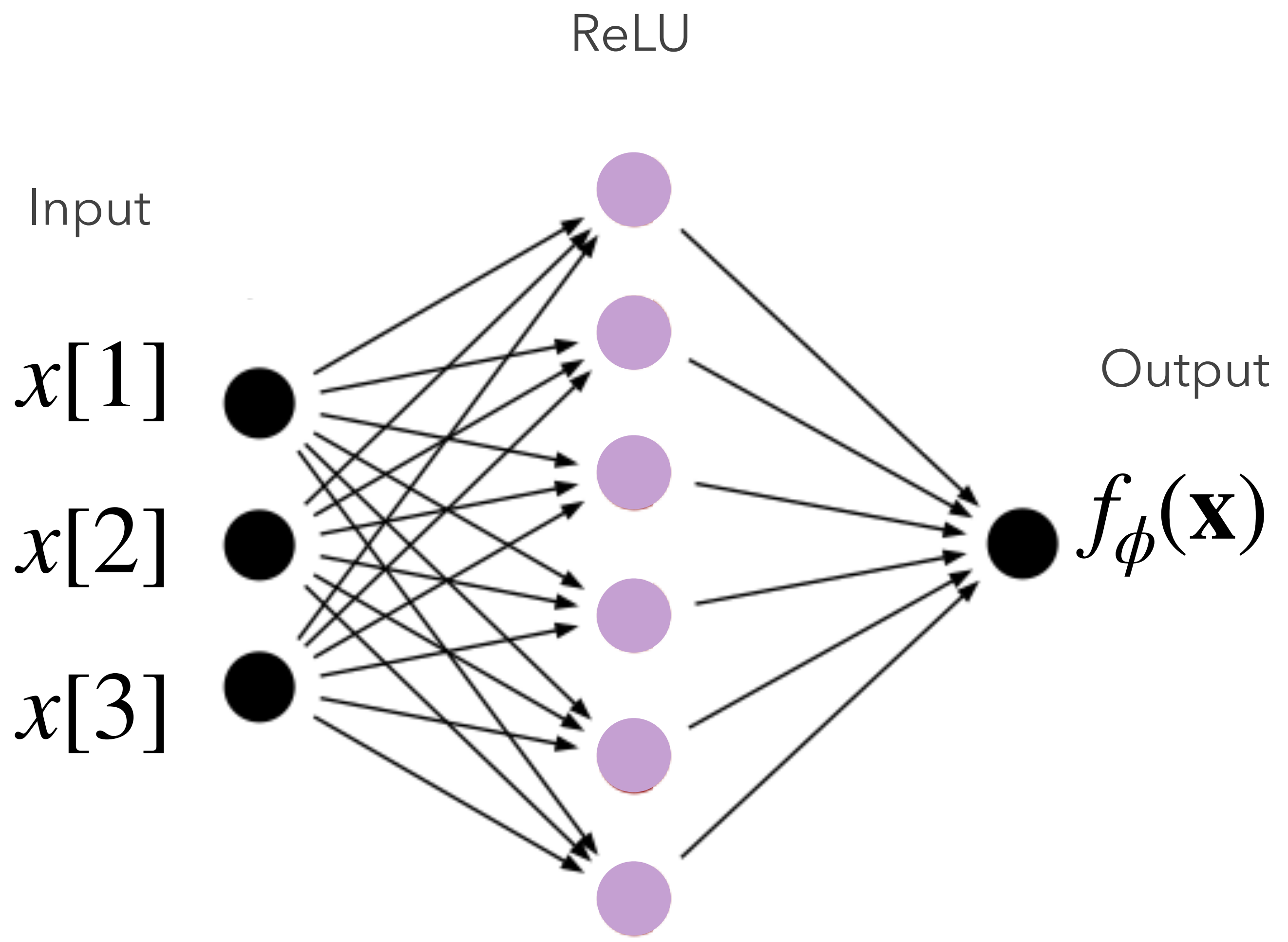
- Therefore, $\mathscr{L}_{\mathscr{D}}(\mathscr{A}_3(S)) \le \varepsilon$ as long as

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

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

Key:

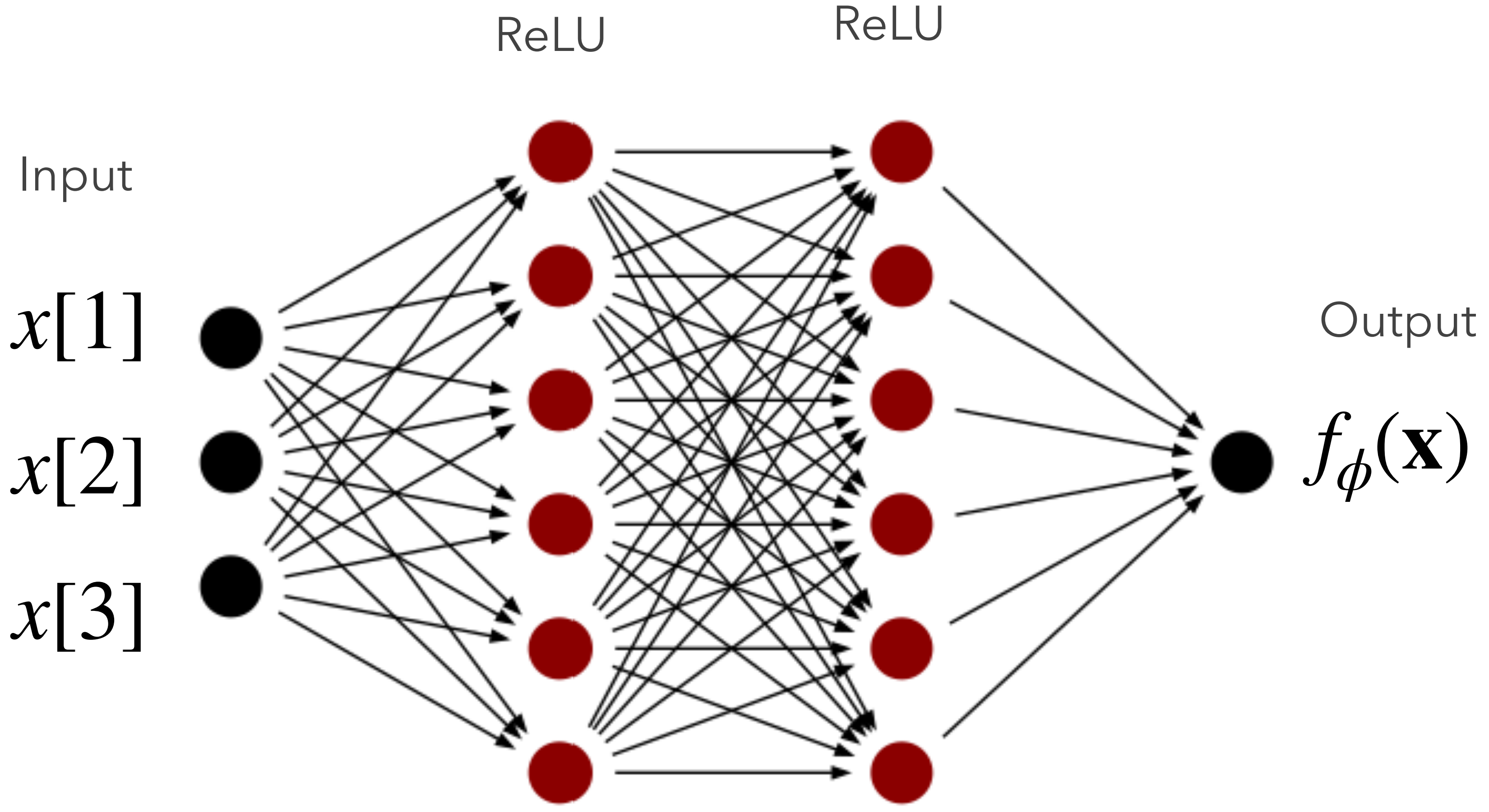
Small **representation cost** with depth 2



Cheap

$\Longrightarrow$

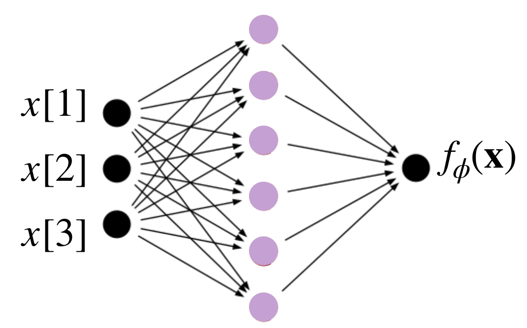
Small **representation cost** with depth 3



Cheap

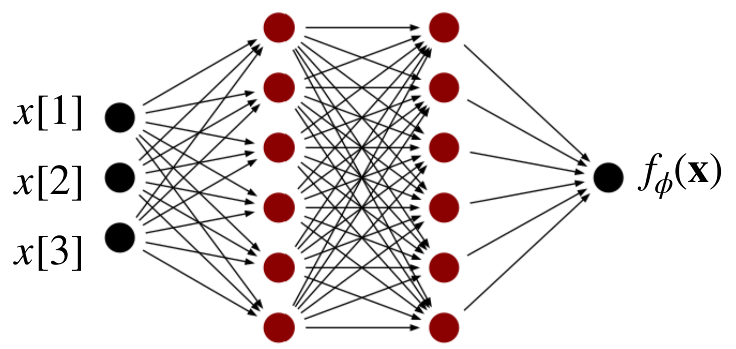
No Reverse Depth Separation: f "easy" with depth 2 $\implies$ "easy" with depth 3

Proof Sketch:



Cheap

$\Downarrow$



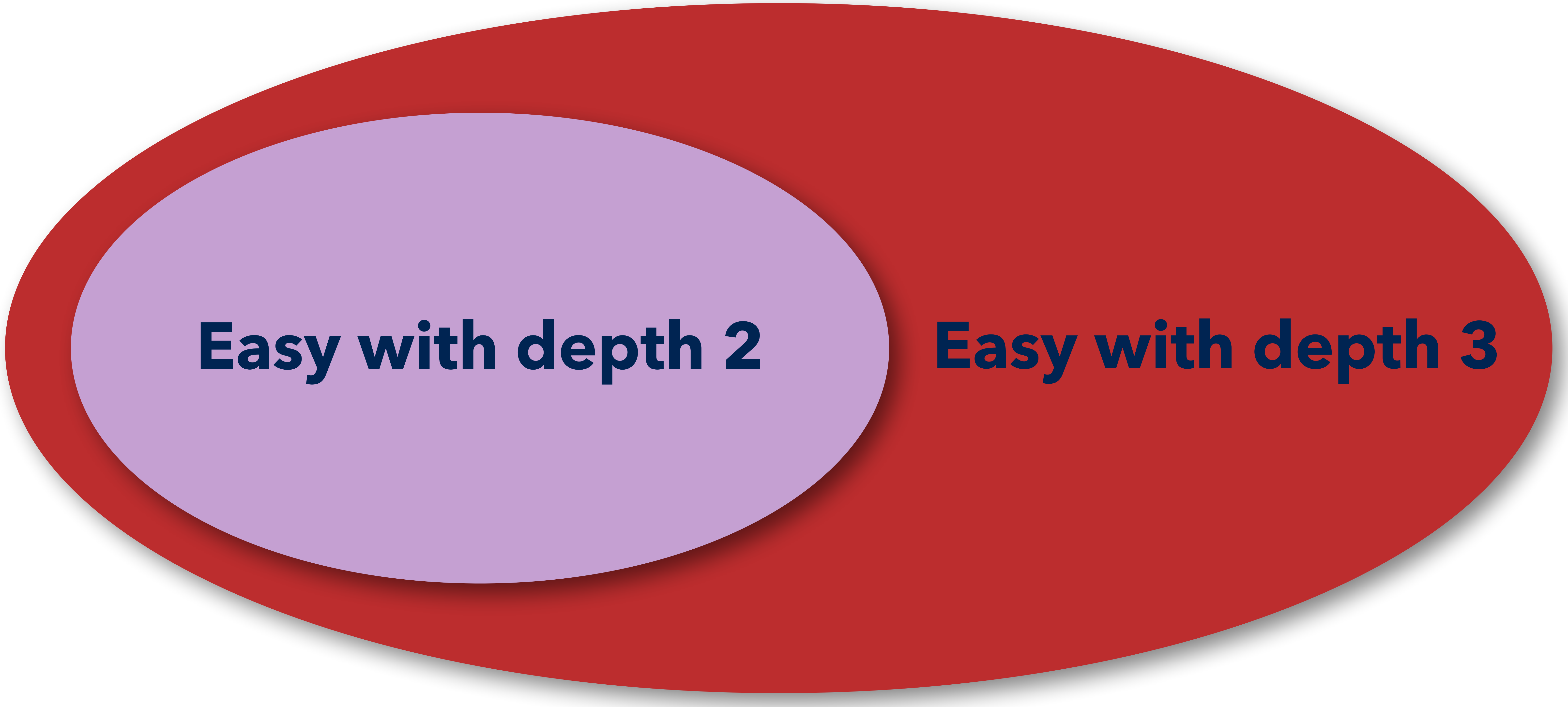
Cheap

- If $\mathscr{A}_2(S)$ learns with polynomial sample complexity, $\exists f_\varepsilon$ of depth 2 such that $\mathscr{L}_\mathscr{D}(f_\varepsilon) \leq \varepsilon/2$ and $R_2(f_\varepsilon) \leq \text{poly}(d, \varepsilon^{-1})$.
- $R_3(f_\varepsilon) = O\left(d + R_2(f_\varepsilon)\right) \leq \text{poly}(d, \varepsilon^{-1})$
- Because of how we choose λ_3, we get $R_3(\mathscr{A}_3(S)) \leq R_3(f_\varepsilon) \leq \text{poly}(d, \varepsilon^{-1})$

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

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

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.



Open Questions & Extensions

- Depth separation between other depths– **3** vs. **4**? Deeper?
- Other architectures beyond MLPs? CNNs, ResNets, etc.?
- We've implicitly assumed that we're **close to global minima** of our objective. How does **optimization** and 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.