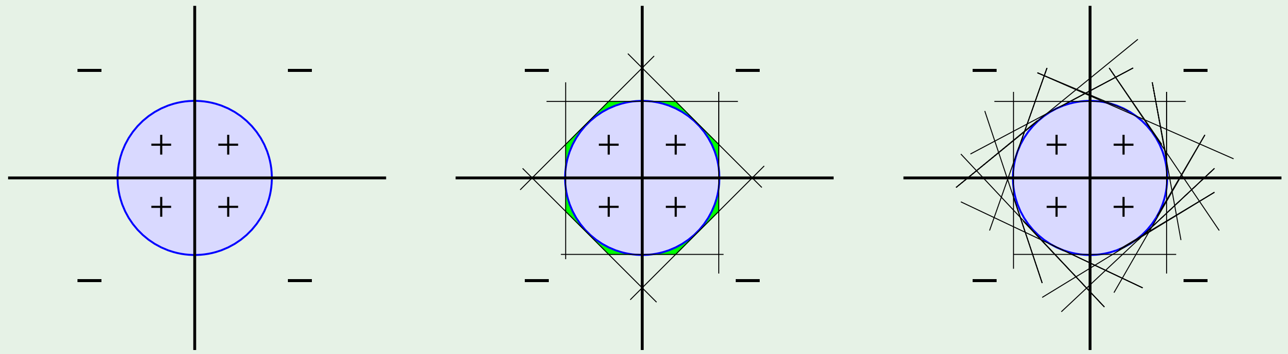
Review of Lecture 10

- Multilayer perceptrons

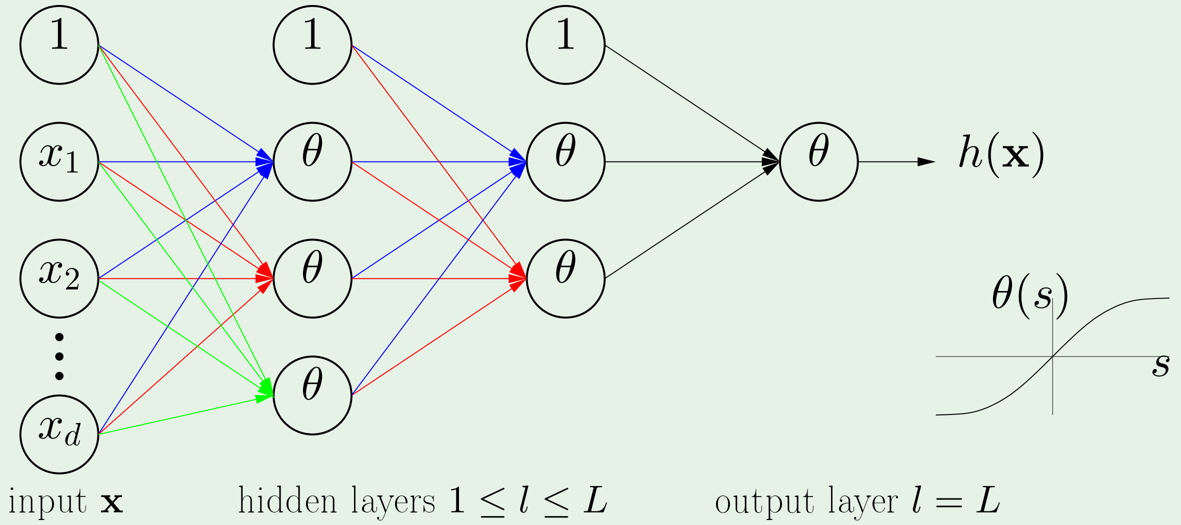


Logical combinations of perceptrons

- Neural networks

$$x_j^{(l)} = \theta\left(\sum_{i=0}^{d^{(l-1)}} w_{ij}^{(l)}\; x_i^{(l-1)}\right)$$

where $\theta(s) = \tanh(s)$



- Backpropagation

$$\Delta w_{ij}^{(l)} \;=\; -\eta\; x_i^{(l-1)} \delta_j^{(l)}$$

where

$$\delta_i^{(l-1)} \;=\; (1 - (x_i^{(l-1)})^2) \sum_{j=1}^{d^{(l)}} w_{ij}^{(l)}\; \delta_j^{(l)}$$

Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology

Lecture 11: Overfitting

Sponsored by Caltech's Provost Office, E&AS Division, and IST • Tuesday, May 8, 2012

Outline

- What is overfitting?
- The role of noise
- Deterministic noise
- Dealing with overfitting

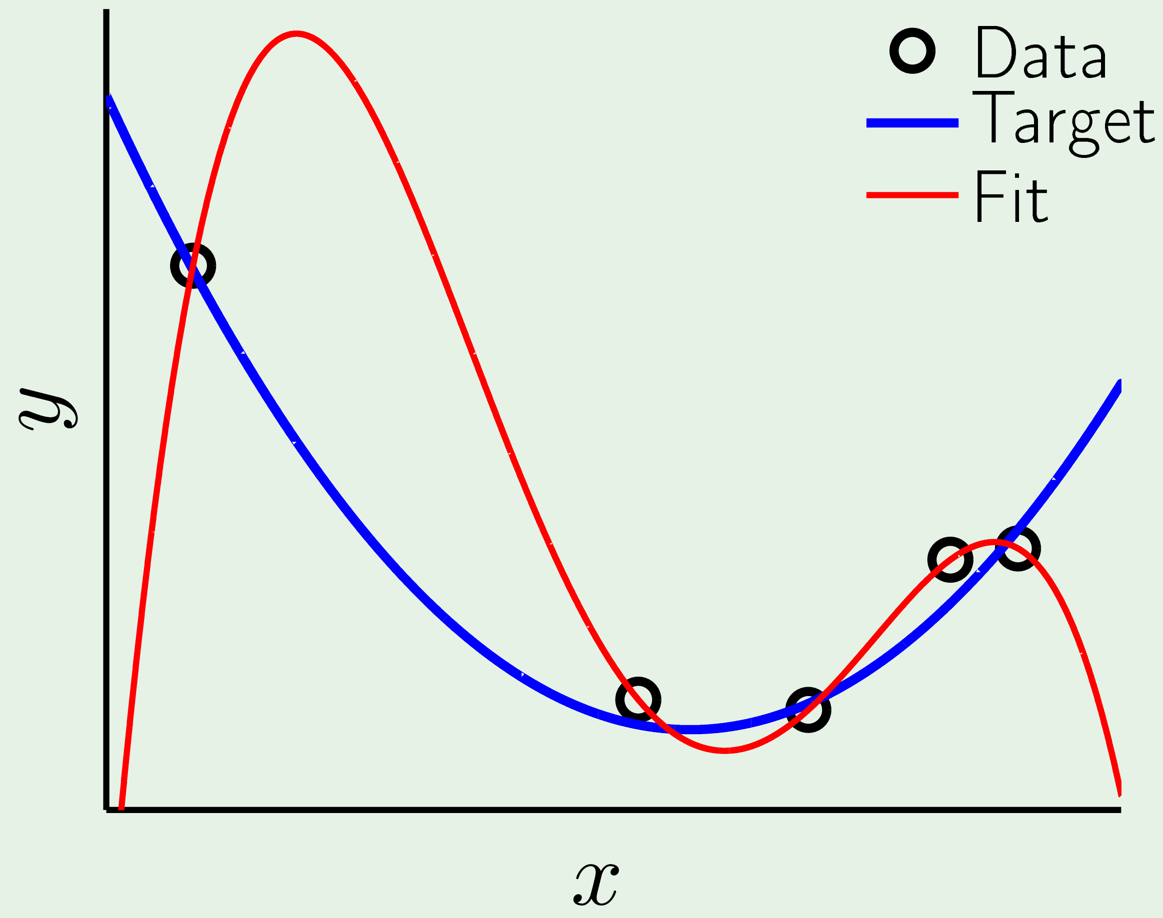
Illustration of overfitting

Simple target function

5 data points- **noisy**

4th-order polynomial fit

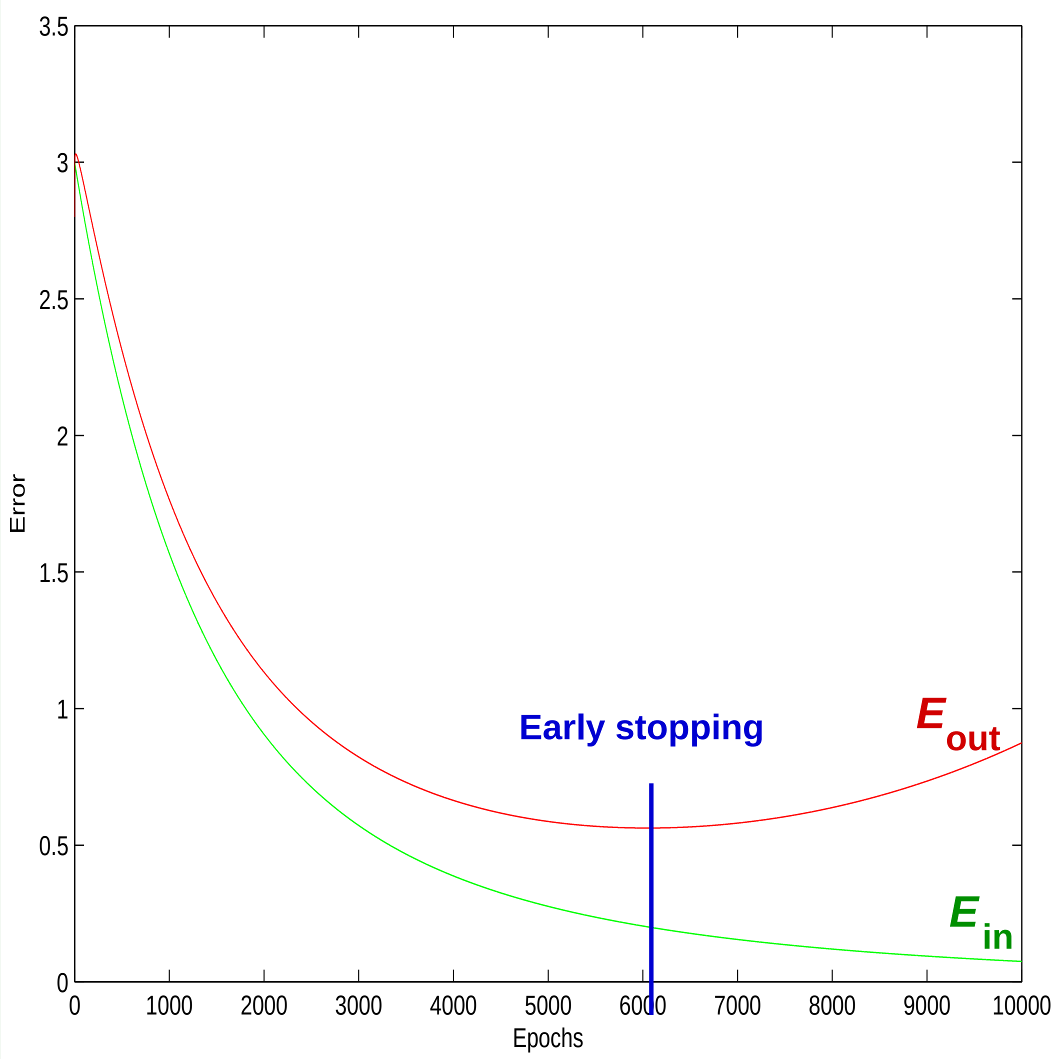
$$E_{\text{in}} = 0, \quad E_{\text{out}} \text{ is huge}$$



Overfitting versus bad generalization

Neural network fitting noisy data

Overfitting: $E_{\text{in}} \downarrow \quad E_{\text{out}} \uparrow$



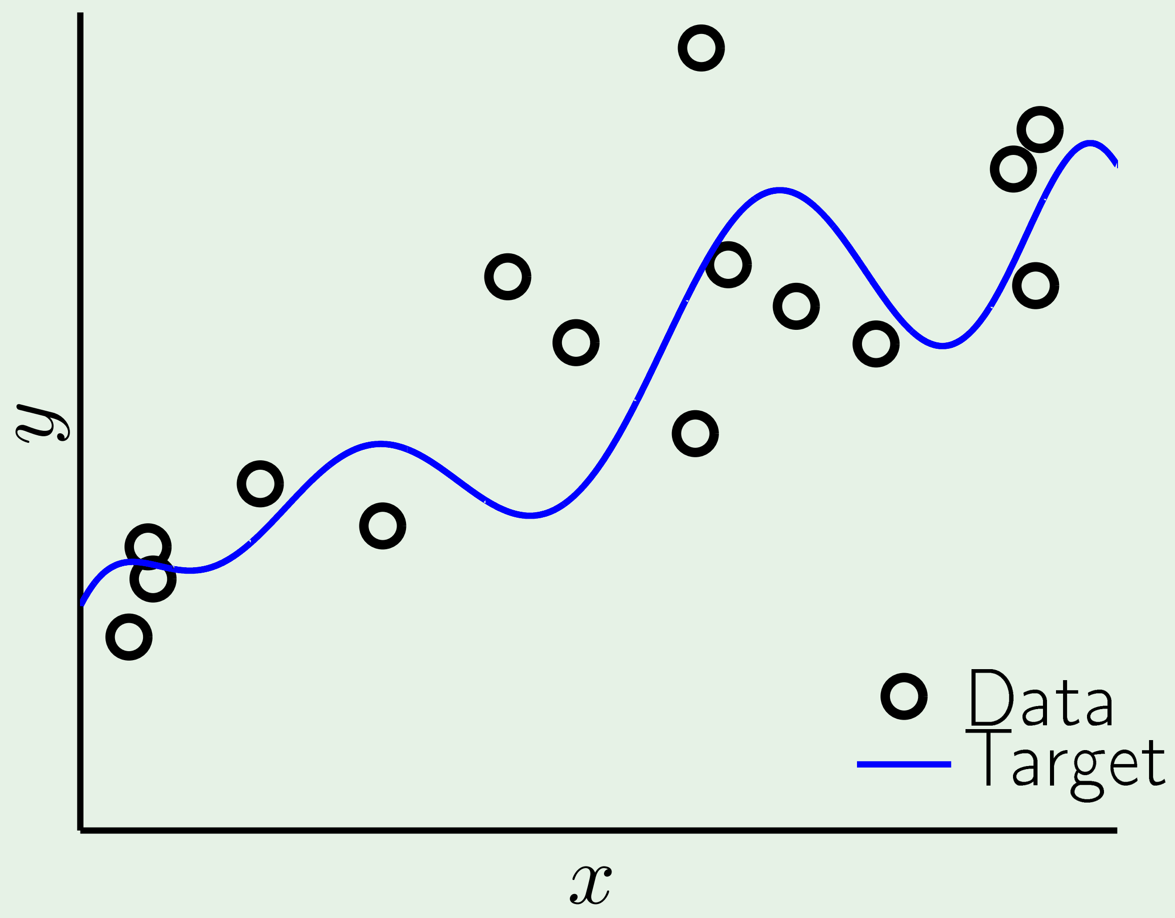
The culprit

Overfitting: "fitting the data more than is warranted"

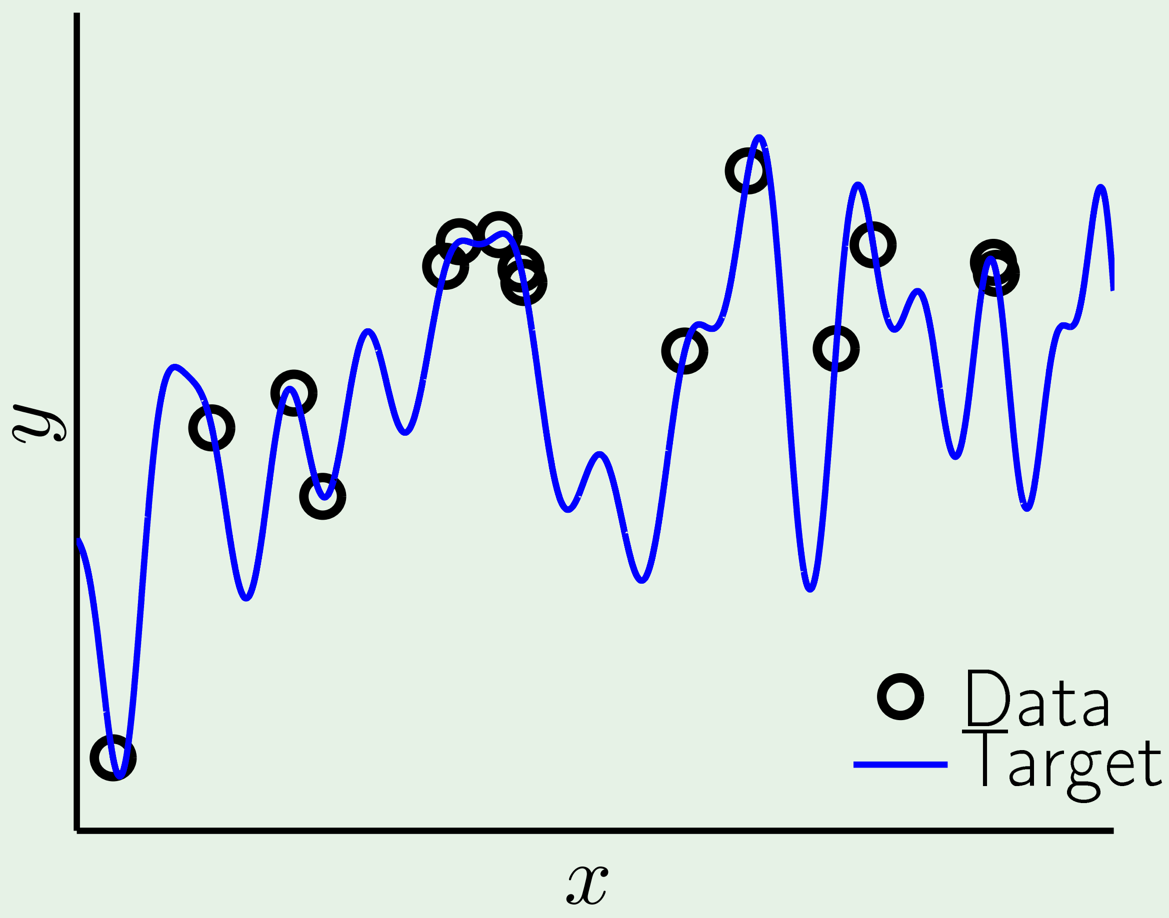
Culprit: fitting the noise - harmful

Case study

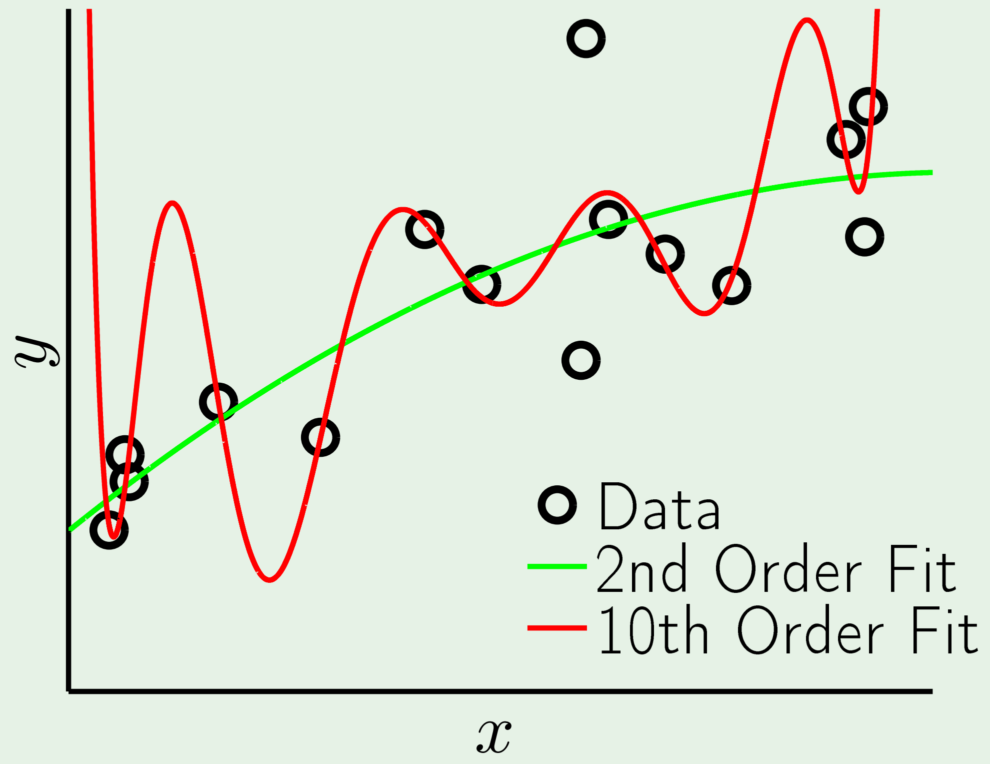
10th-order target + noise

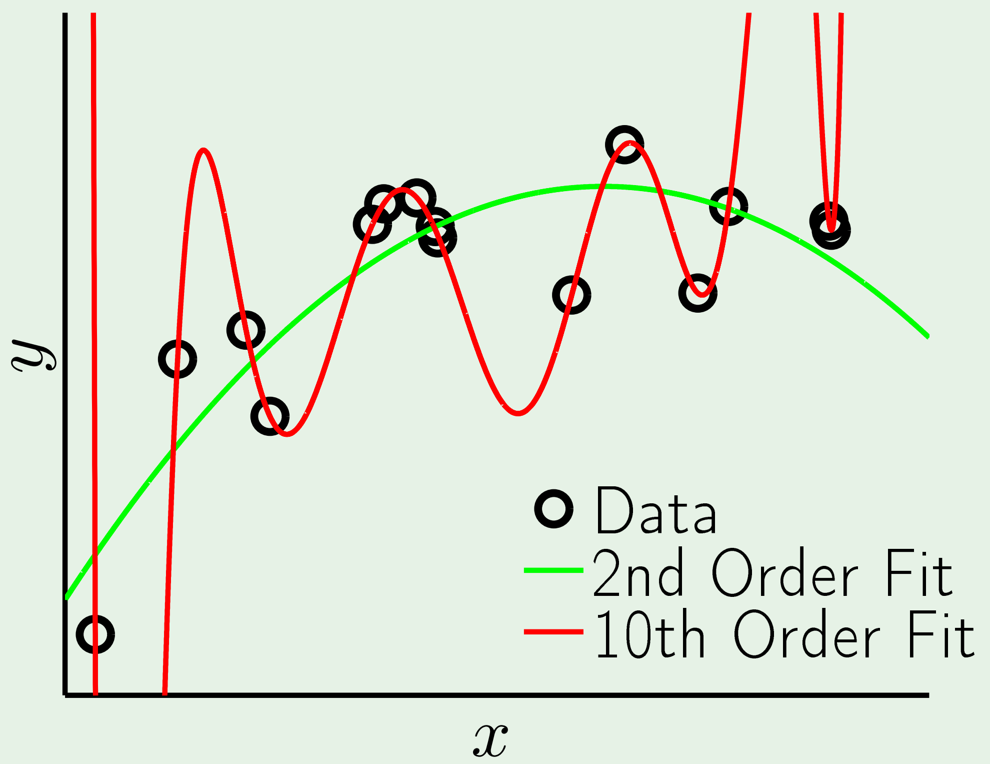


50th-order target



Two fits for each target





Noisy low-order target

	2nd Order	10th Order
E_{in}	0.050	0.034
E_{out}	0.127	**9.00**

Noiseless high-order target

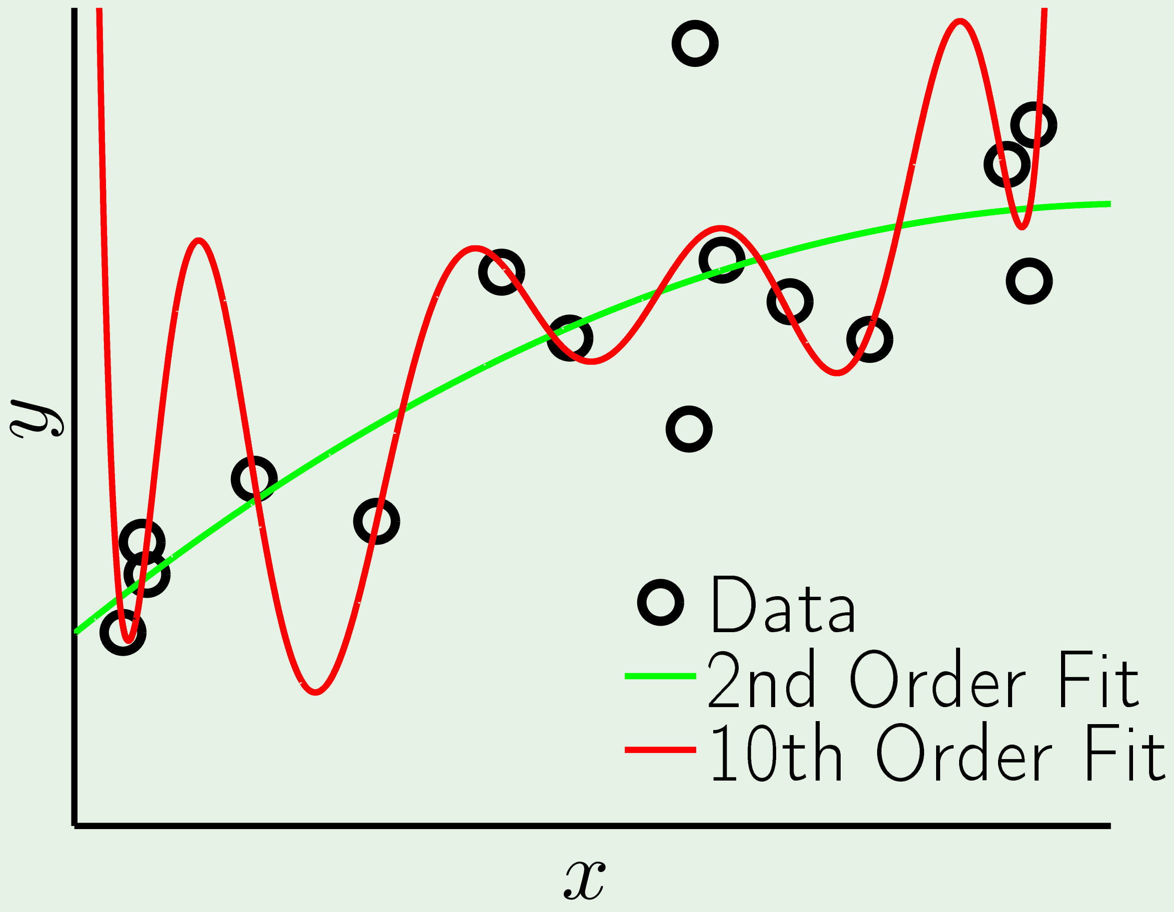
	2nd Order	10th Order
E_{in}	0.029	10^{-5}
E_{out}	0.120	**7680**

An irony of two learners

Two learners O and R

They know the target is 10th order

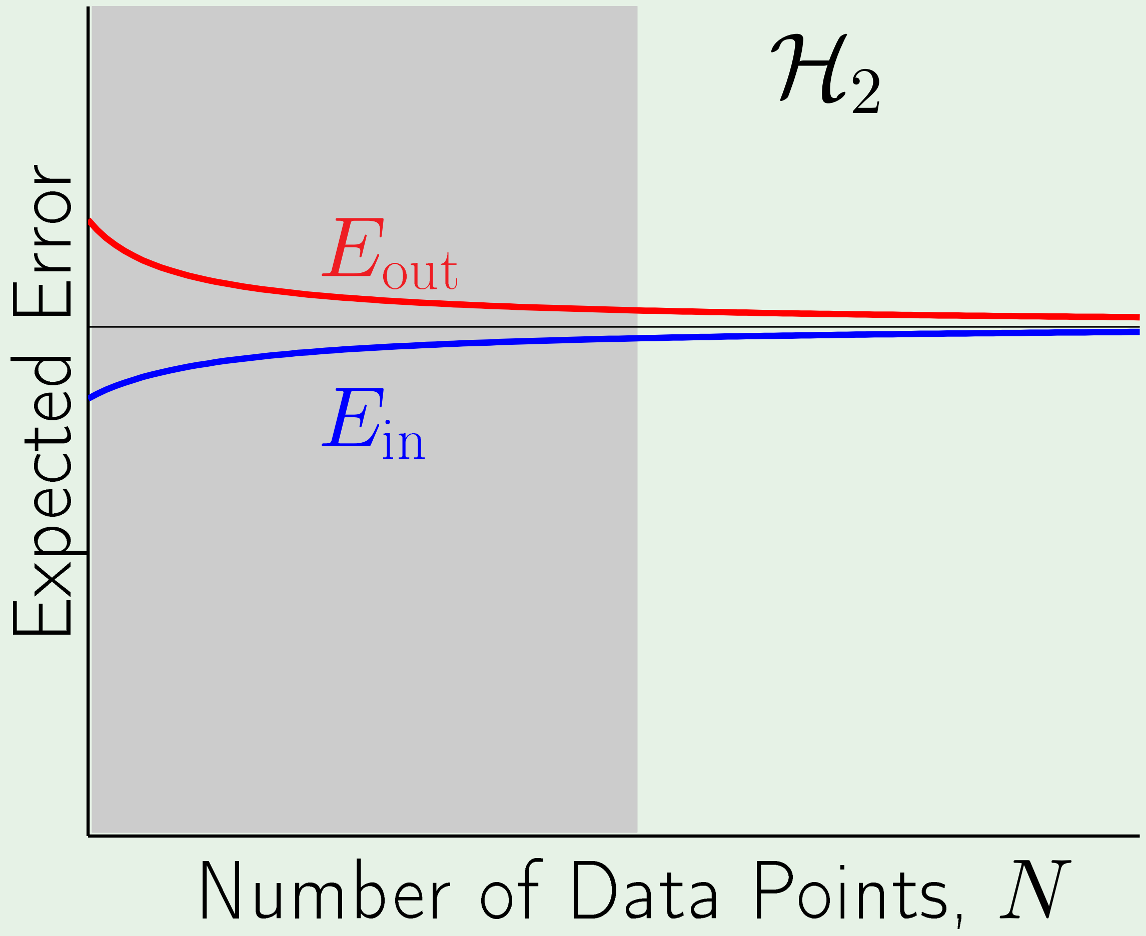
O chooses $\mathcal{H}_{10}$ R chooses $\mathcal{H}_2$

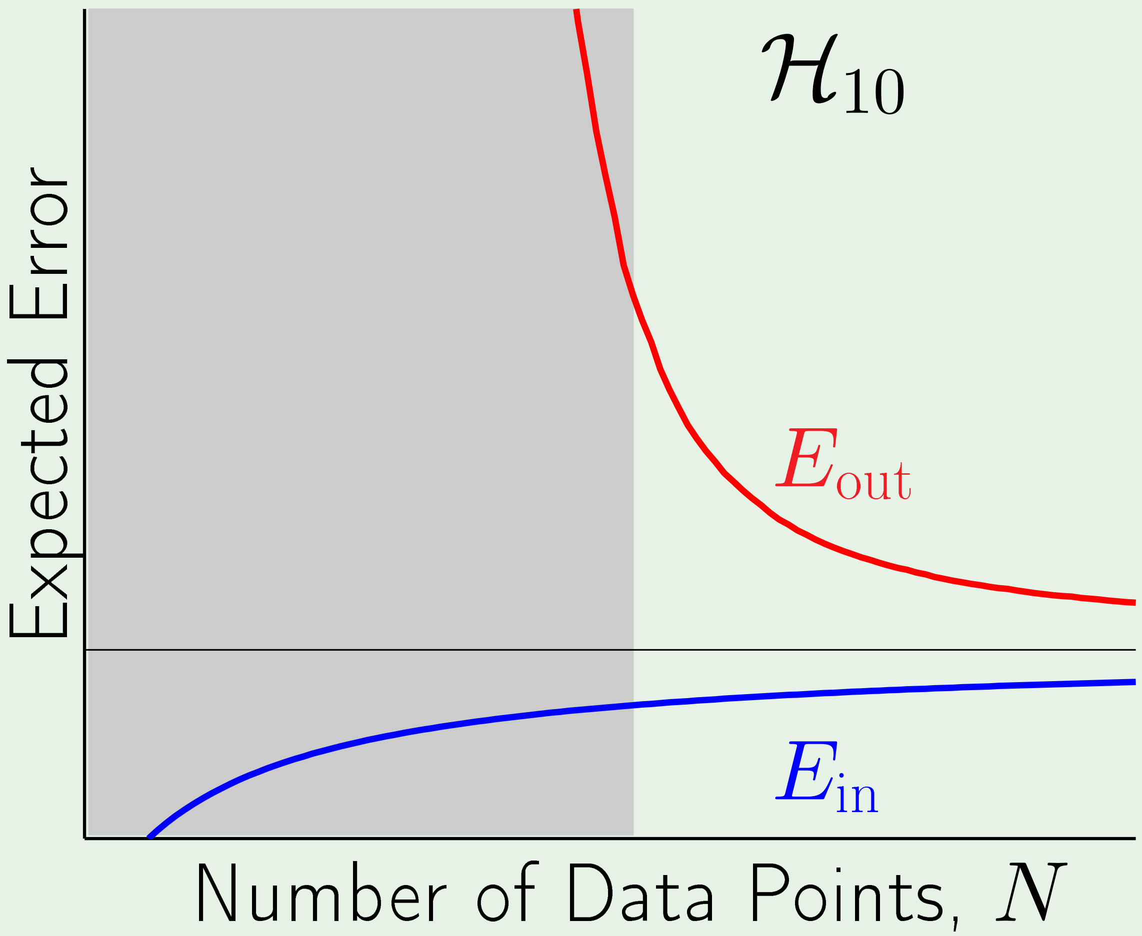


Learning a 10th-order target

We have seen this case

Remember learning curves?



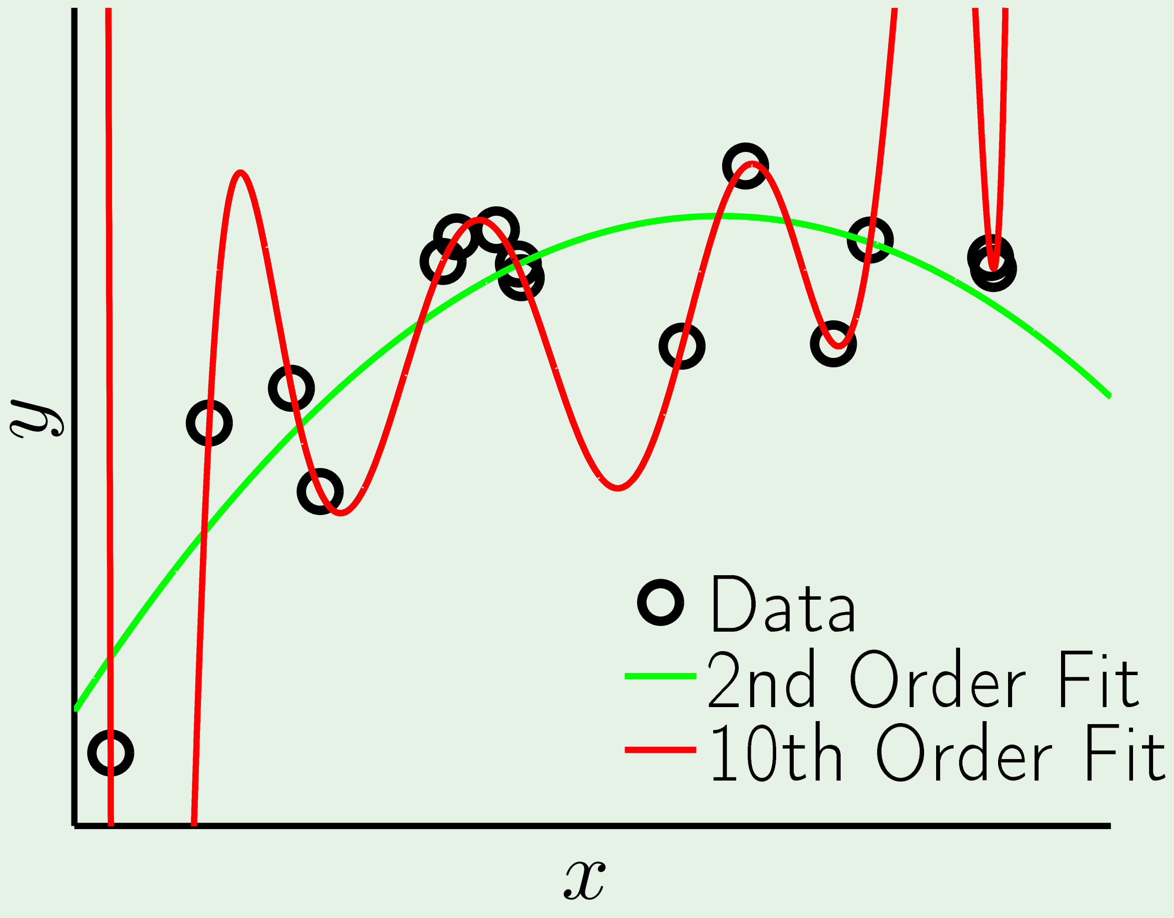


Even without noise

The two learners $\mathcal{H}_{10}$ and $\mathcal{H}_2$

They know there is no noise

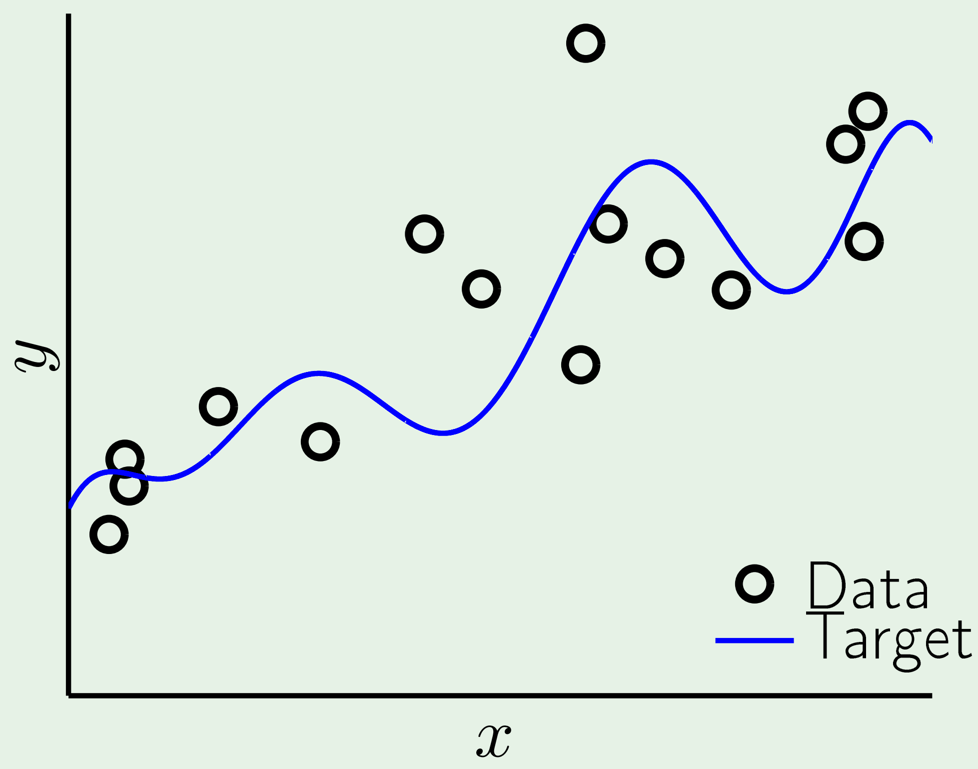
Is there really no noise?



Learning a 50th-order target

A detailed experiment

Impact of **noise level** and **target complexity**



$$y \; = \; f(x) + \underbrace{\epsilon(x)}_{\sigma^2} = \underbrace{\sum_{q=0}^{Q_f} \alpha_q \, x^q}_{\text{normalized}} + \epsilon(x)$$

noise level: σ^2

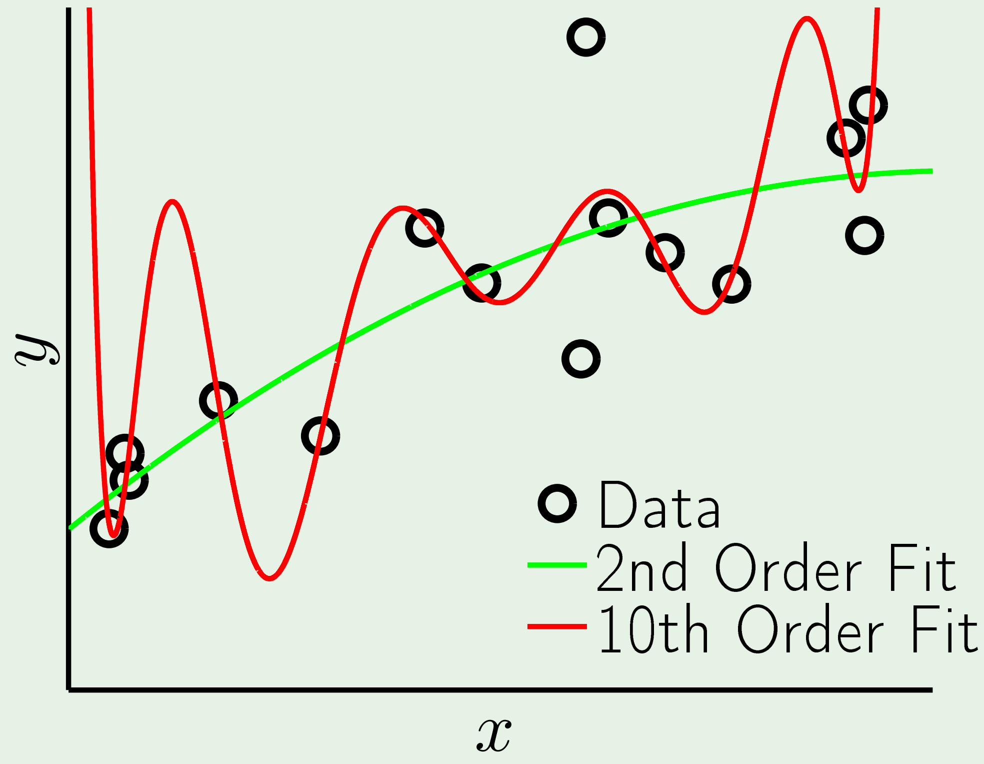
target complexity: Q_f

data set size: N

The overfit measure

We fit the data set $(x_1, y_1), \cdots, (x_N, y_N)$ using our two models:

$\mathcal{H}_2$: 2nd-order polynomials $\mathcal{H}_{10}$: 10th-order polynomials

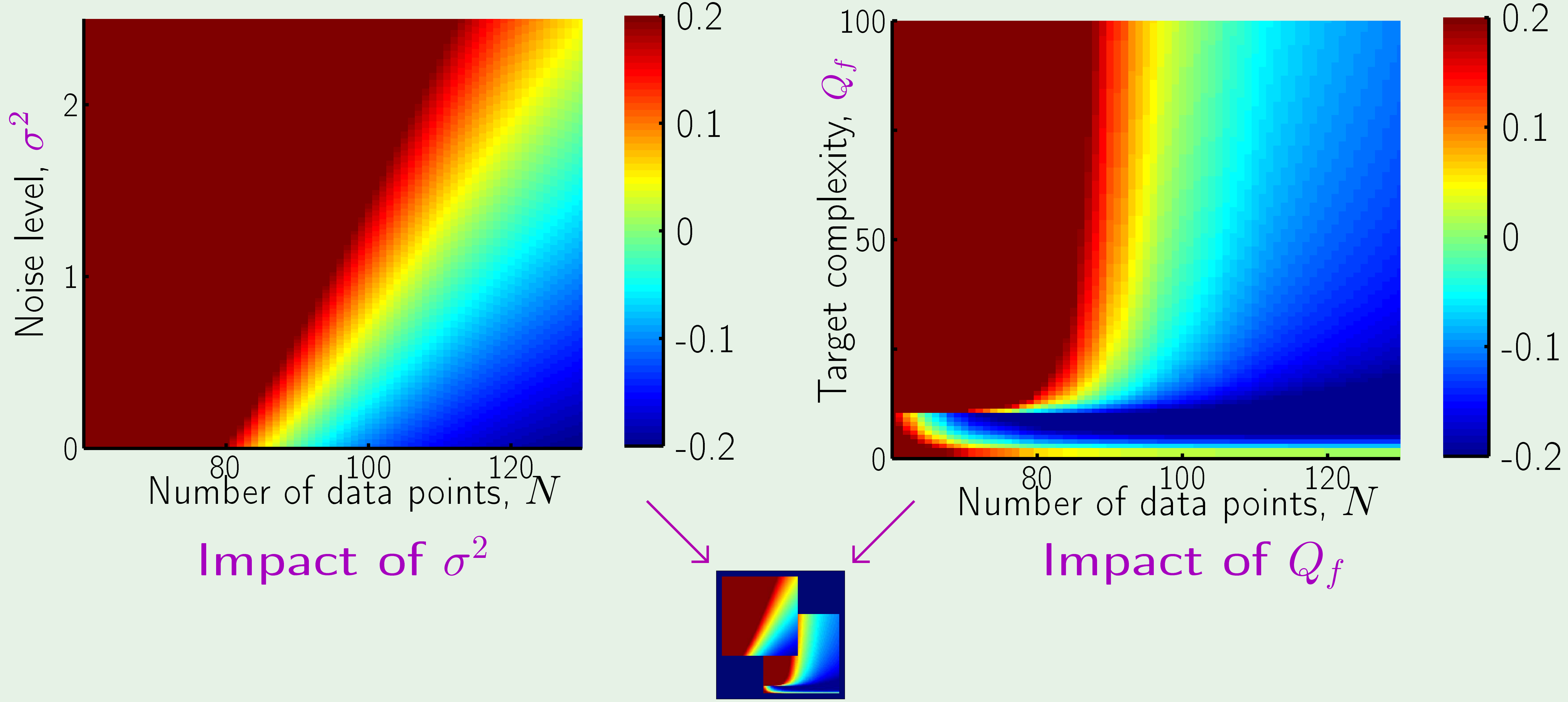


Compare out-of-sample errors of

$$g_2 \in \mathcal{H}_2 \quad \text{and} \quad g_{10} \in \mathcal{H}_{10}$$

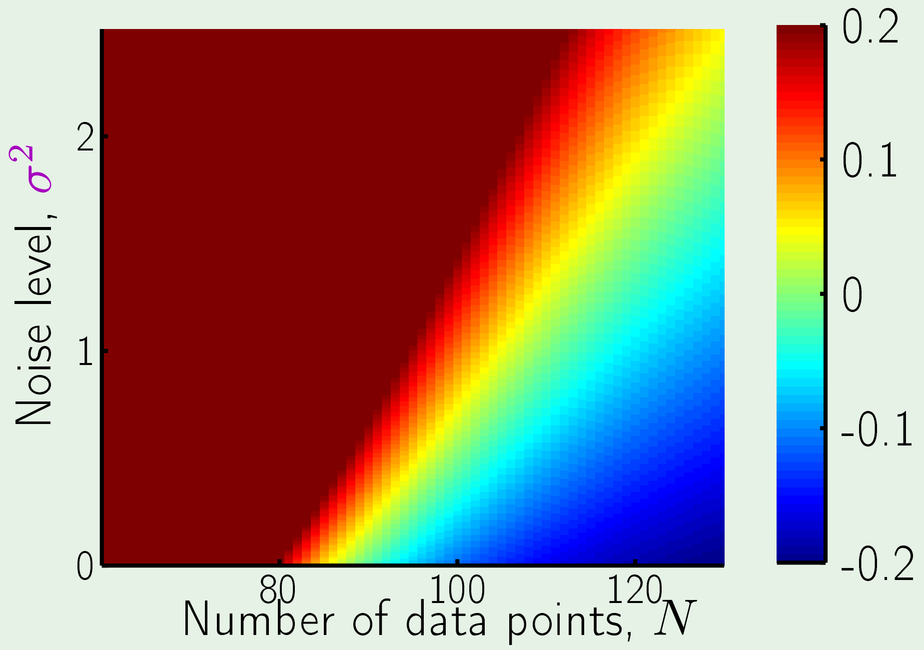
overfit measure: $E_{\text{out}}(g_{10}) - E_{\text{out}}(g_2)$

The results

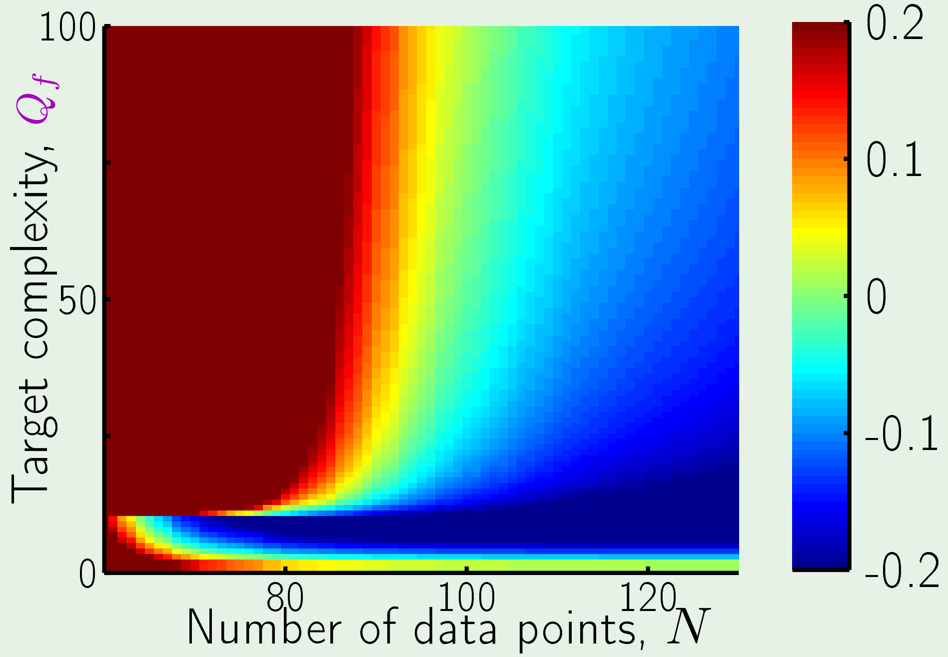


Impact of σ^2 **Impact of Q_f**

Impact of "noise"



Stochastic noise



Deterministic noise

number of data points	$\uparrow$	Overfitting	$\downarrow$
stochastic noise	$\uparrow$	Overfitting	$\uparrow$
deterministic noise	$\uparrow$	Overfitting	$\uparrow$

Outline

- What is overfitting?
- The role of noise
- Deterministic noise
- Dealing with overfitting

Definition of deterministic noise

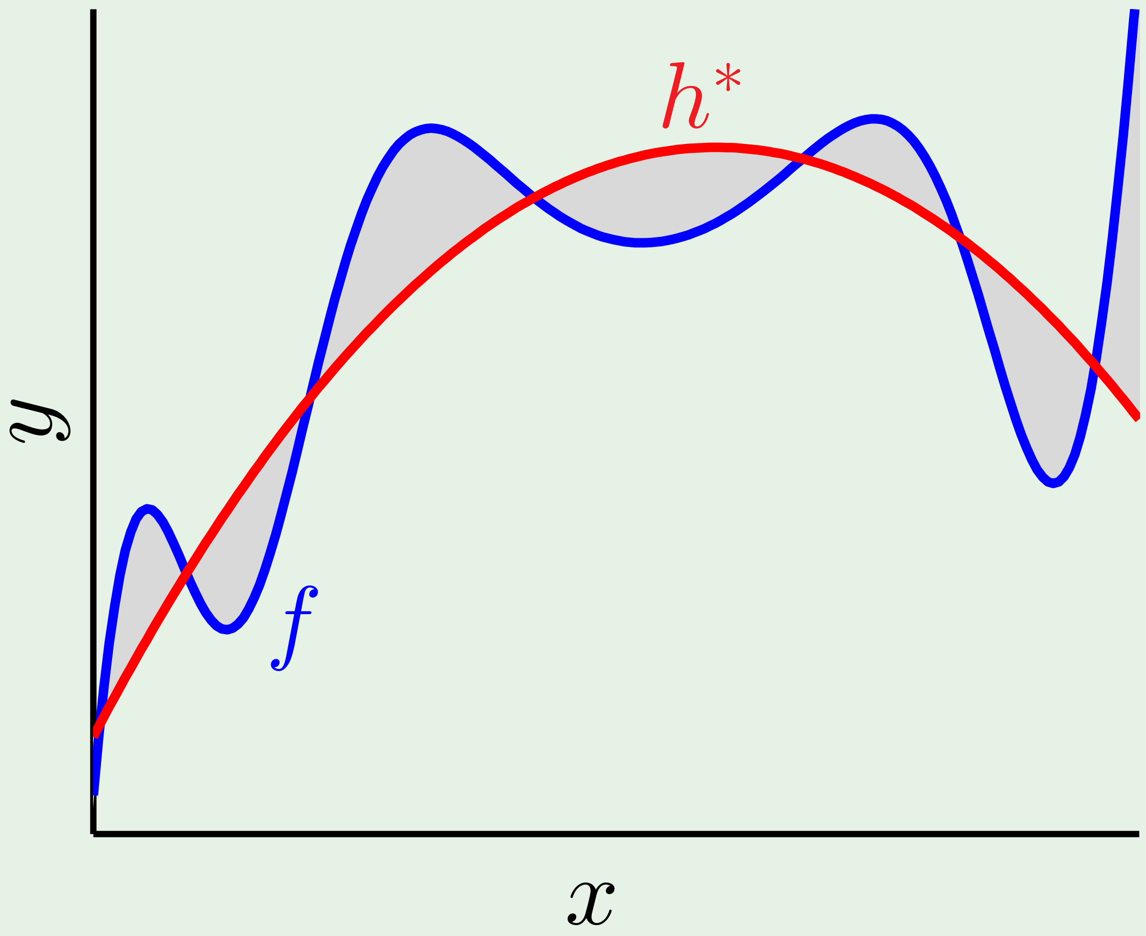
The part of f that $\mathcal{H}$ cannot capture: $\quad f(\mathbf{x}) - h^*(\mathbf{x})$

Why "noise"?

Main differences with stochastic noise:

1. depends on $\mathcal{H}$

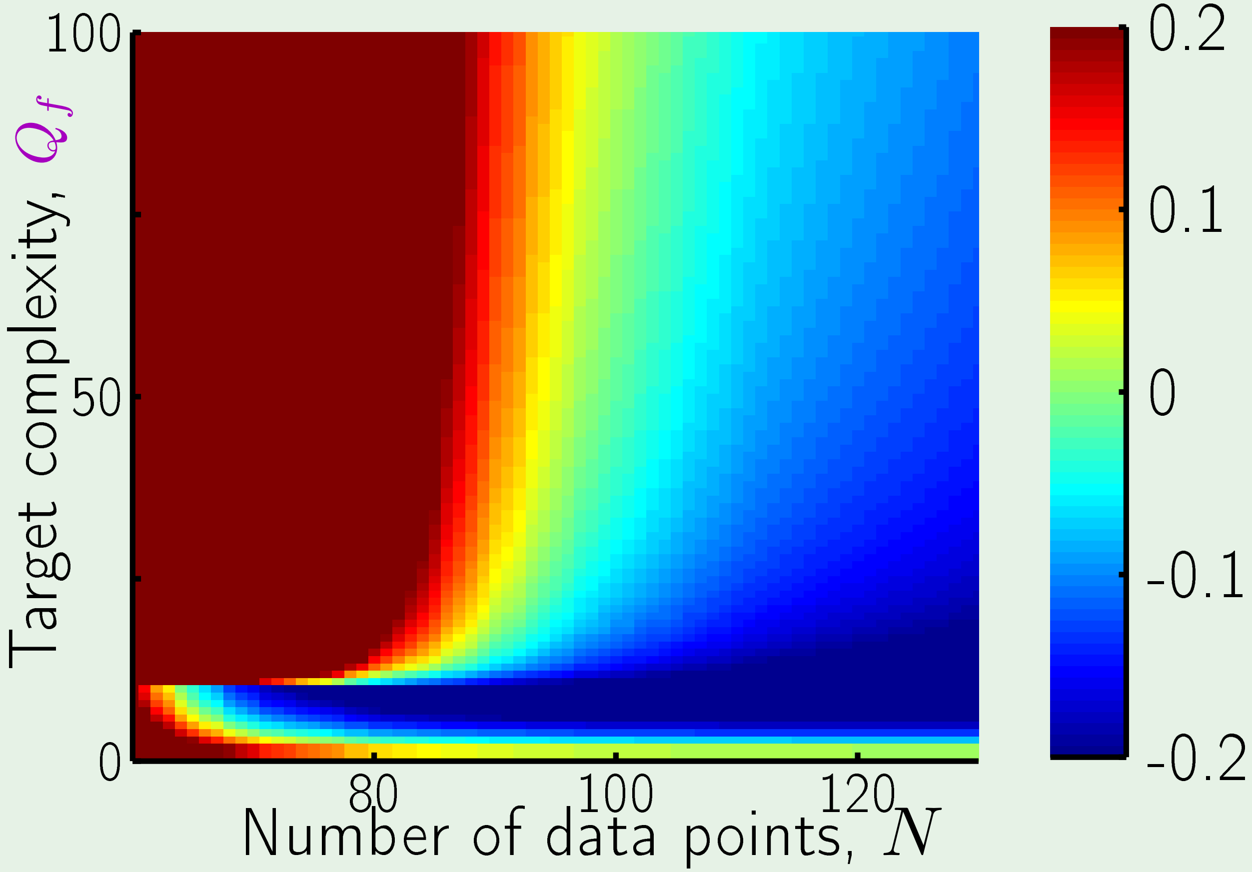
2. fixed for a given $\mathbf{x}$



Impact on overfitting

Deterministic noise and Q_f

Finite N: $\mathcal{H}$ tries to fit the noise



how much overfit

Noise and bias-variance

Recall the decomposition:

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\textbf{var}(\mathbf{x})} + \underbrace{\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]}_{\textbf{bias}(\mathbf{x})}$$

What if f is a noisy target?

$$y = f(\mathbf{x}) + \epsilon(\mathbf{x}) \qquad \mathbb{E}\left[\epsilon(\mathbf{x})\right] = 0$$

A noise term

$$\mathbb{E}_{\mathcal{D},\epsilon}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \quad y \quad\right)^2\right] = \mathbb{E}_{\mathcal{D},\epsilon}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}) - \epsilon(\mathbf{x})\right)^2\right]$$

$$= \mathbb{E}_{\mathcal{D},\epsilon}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}) + \bar{g}(\mathbf{x}) - f(\mathbf{x}) - \epsilon(\mathbf{x})\right)^2\right]$$

$$= \mathbb{E}_{\mathcal{D},\epsilon}\Big[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2 + \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2 + \left(\epsilon(\mathbf{x})\right)^2$$

$$+ \quad \text{cross terms} \Big]$$

Actually, two noise terms

$$\underbrace{\mathbb{E}_{\mathcal{D},\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\textsf{var}} + \underbrace{\mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]}_{\textsf{bias}} + \underbrace{\mathbb{E}_{\epsilon,\mathbf{x}}\left[\left(\epsilon(\mathbf{x})\right)^2\right]}_{\sigma^2}$$

$\uparrow$ deterministic noise

$\uparrow$ stochastic noise

Outline

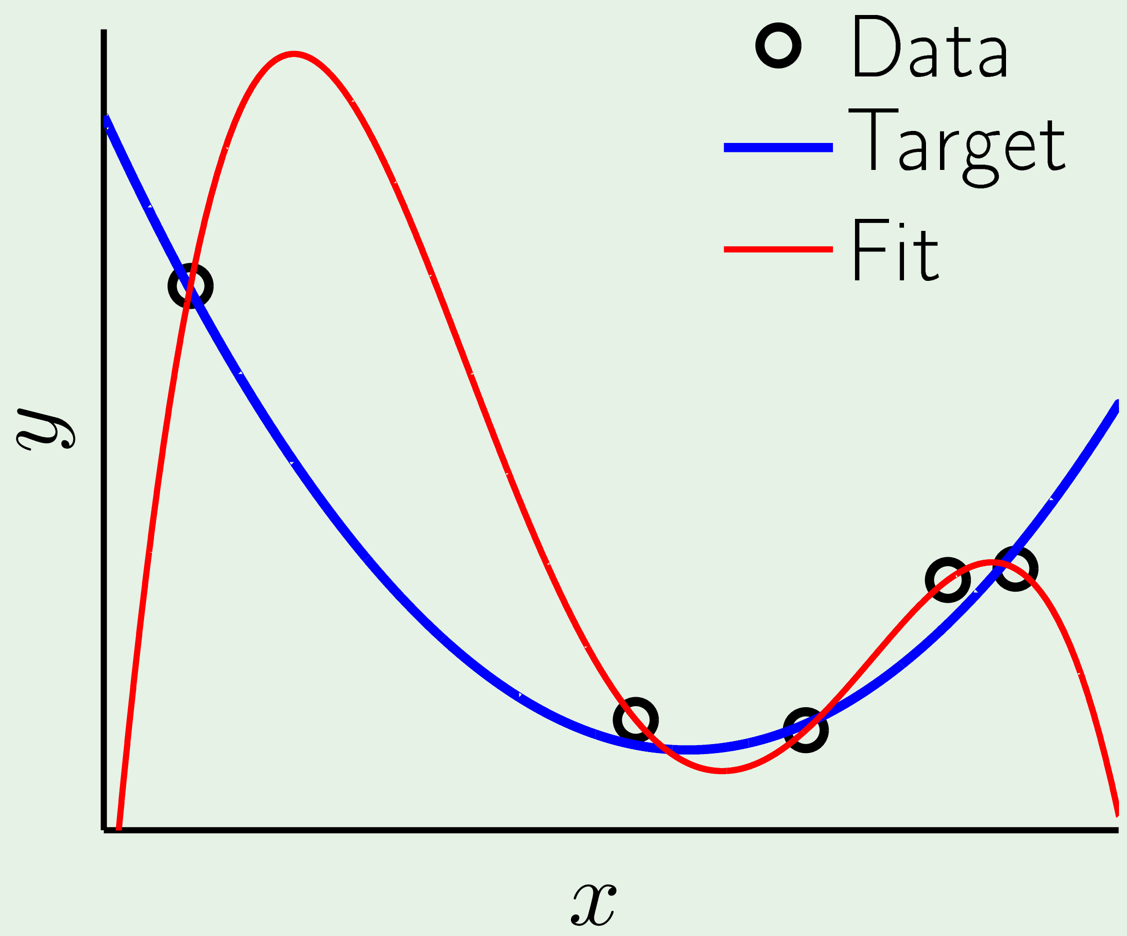
- What is overfitting?
- The role of noise
- Deterministic noise
- Dealing with overfitting

Two cures

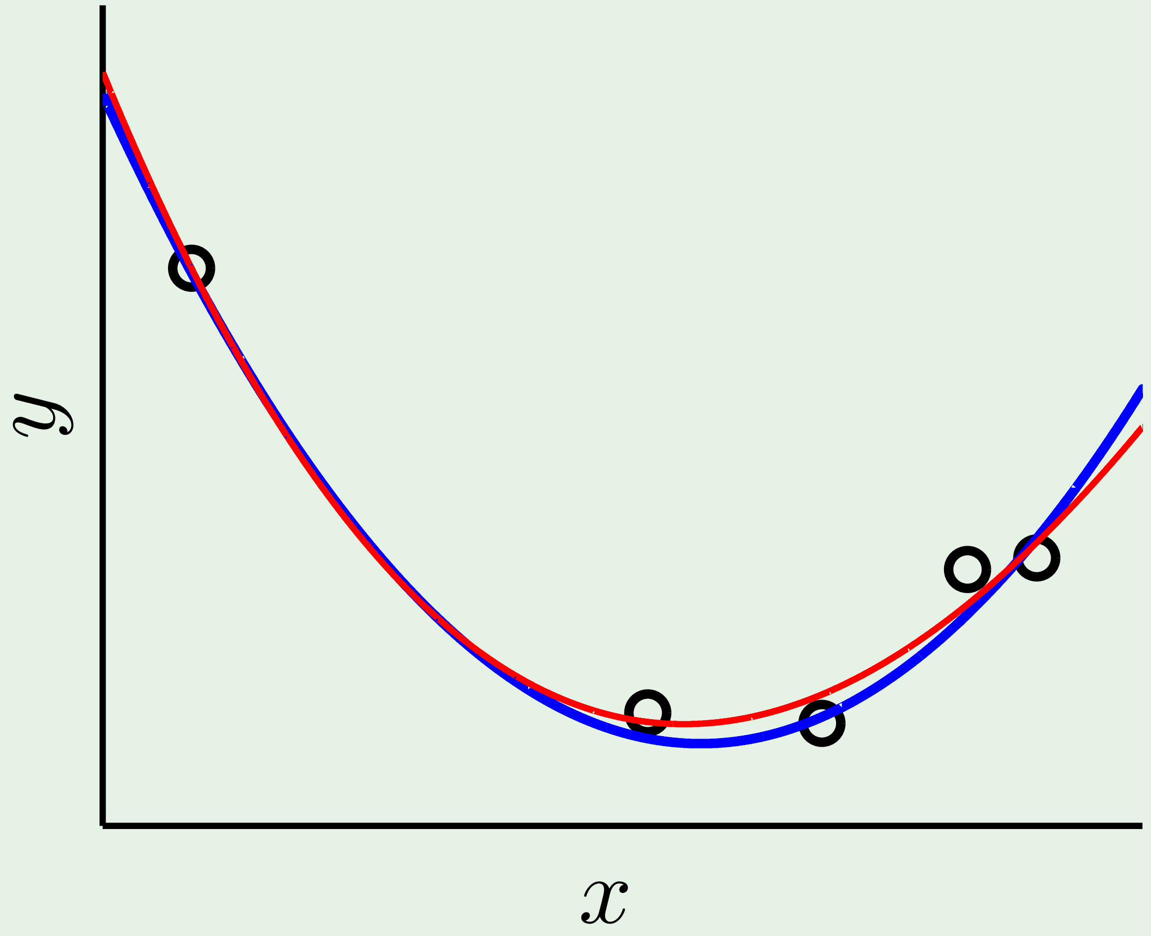
Regularization: Putting the brakes

Validation: Checking the bottom line

Putting the brakes



free fit



restrained fit