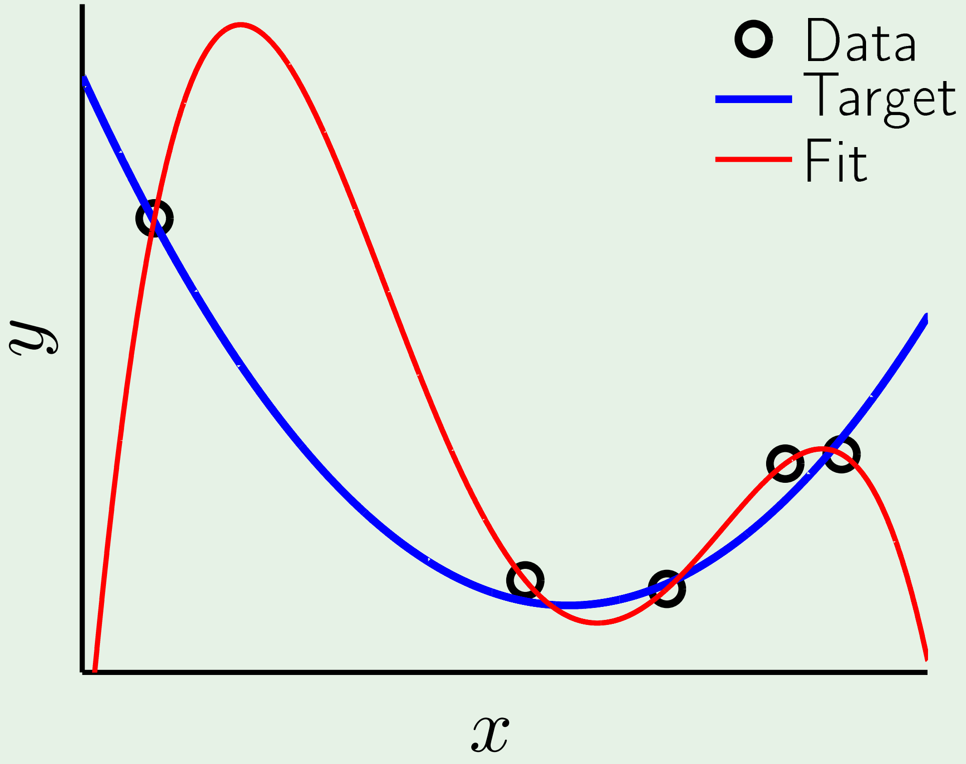
Review of Lecture 11

- **Overfitting**

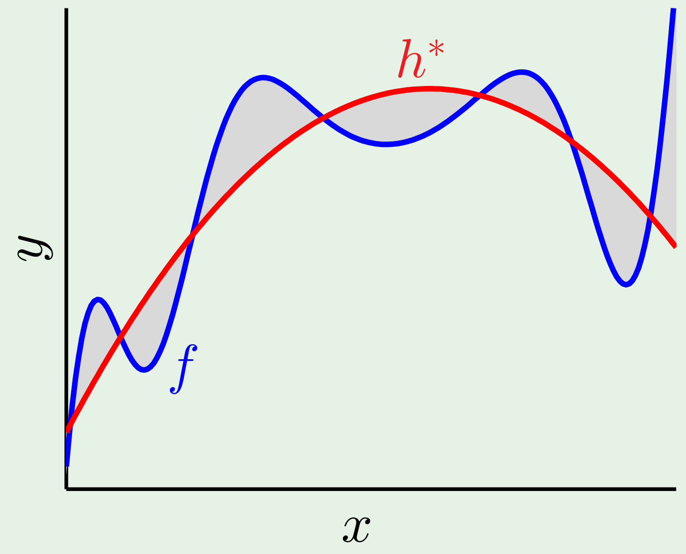
Fitting the data more than is warranted

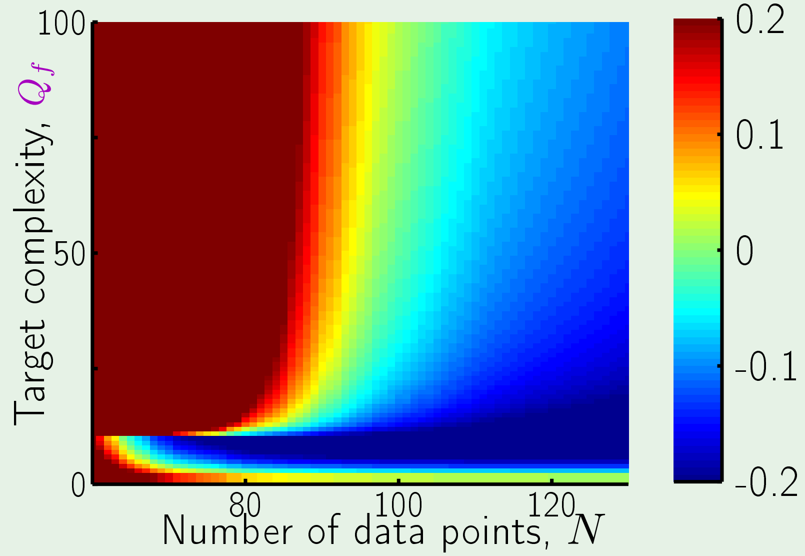


VC allows it; doesn't predict it

Fitting the noise, stochastic/deterministic

- **Deterministic noise**





Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology

Lecture 12: Regularization

Sponsored by Caltech's Provost Office, E&AS Division, and IST • Thursday, May 10, 2012

Outline

- Regularization - informal
- Regularization - formal
- Weight decay
- Choosing a regularizer

Two approaches to regularization

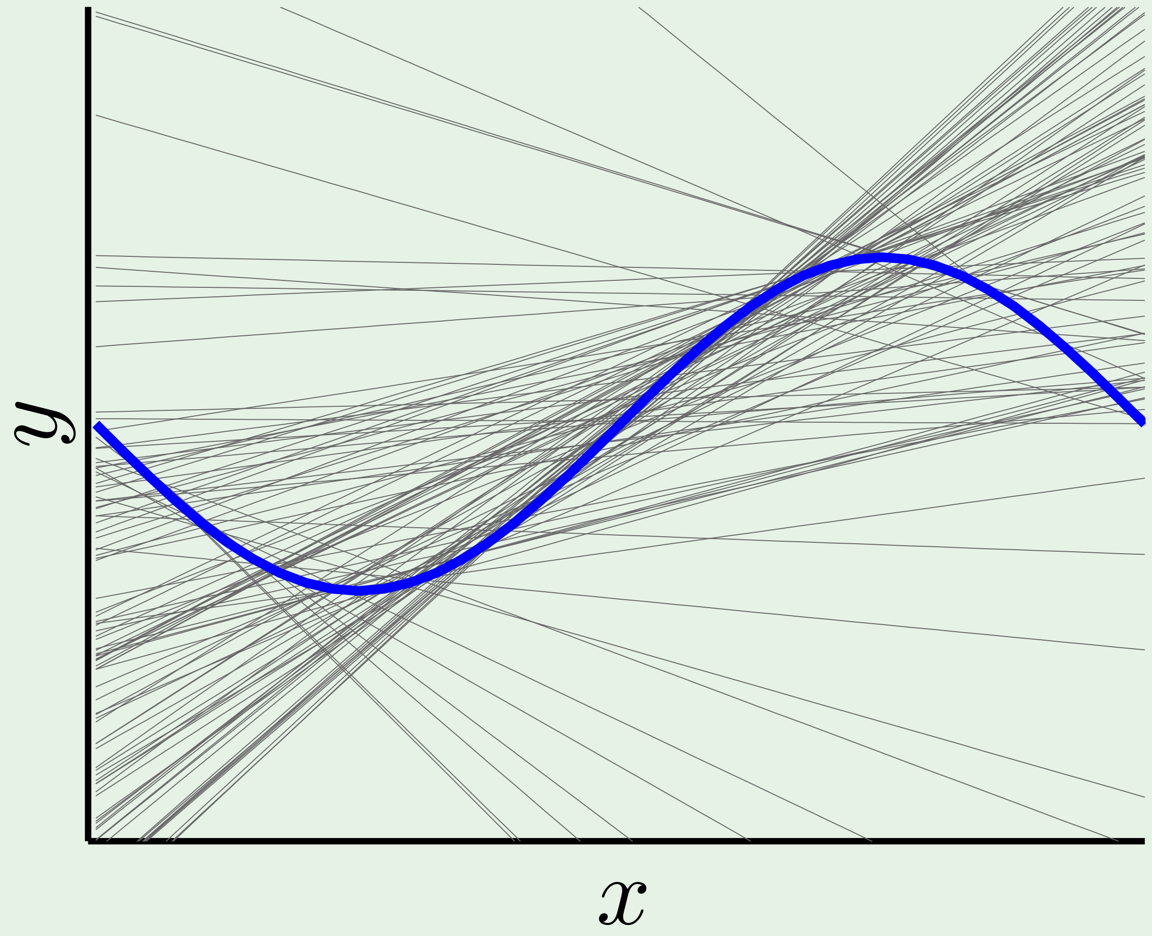
Mathematical:

Ill-posed problems in function approximation

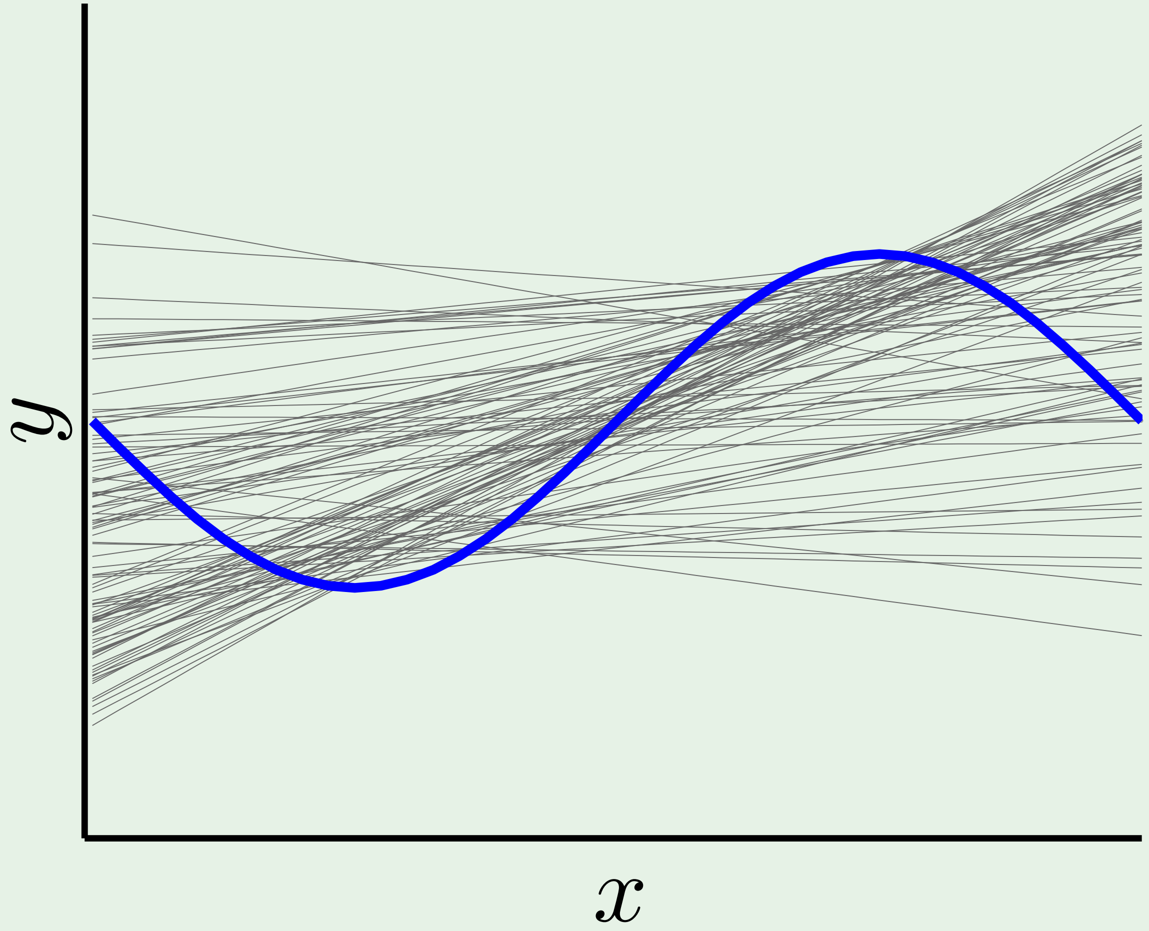
Heuristic:

Handicapping the minimization of E_{in}

A familiar example



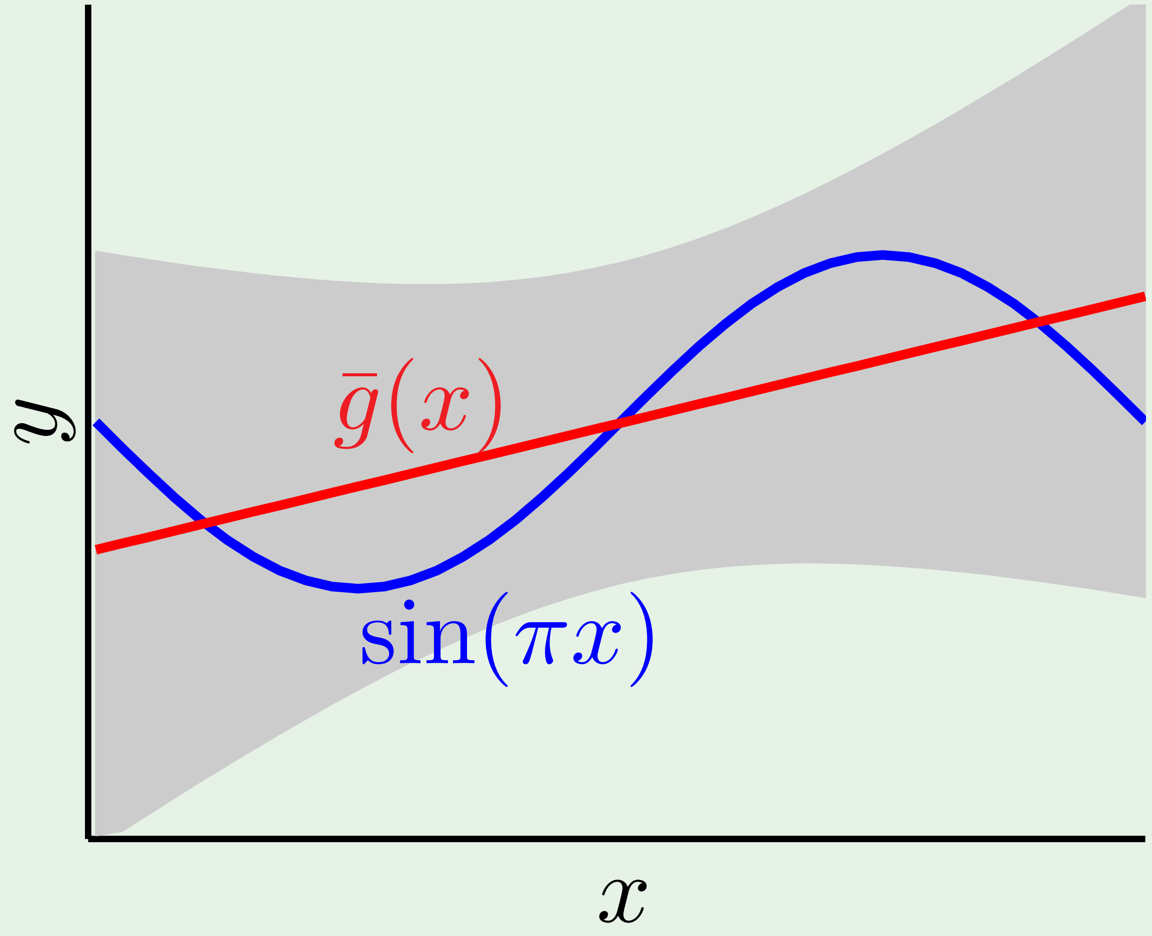
without regularization



with regularization

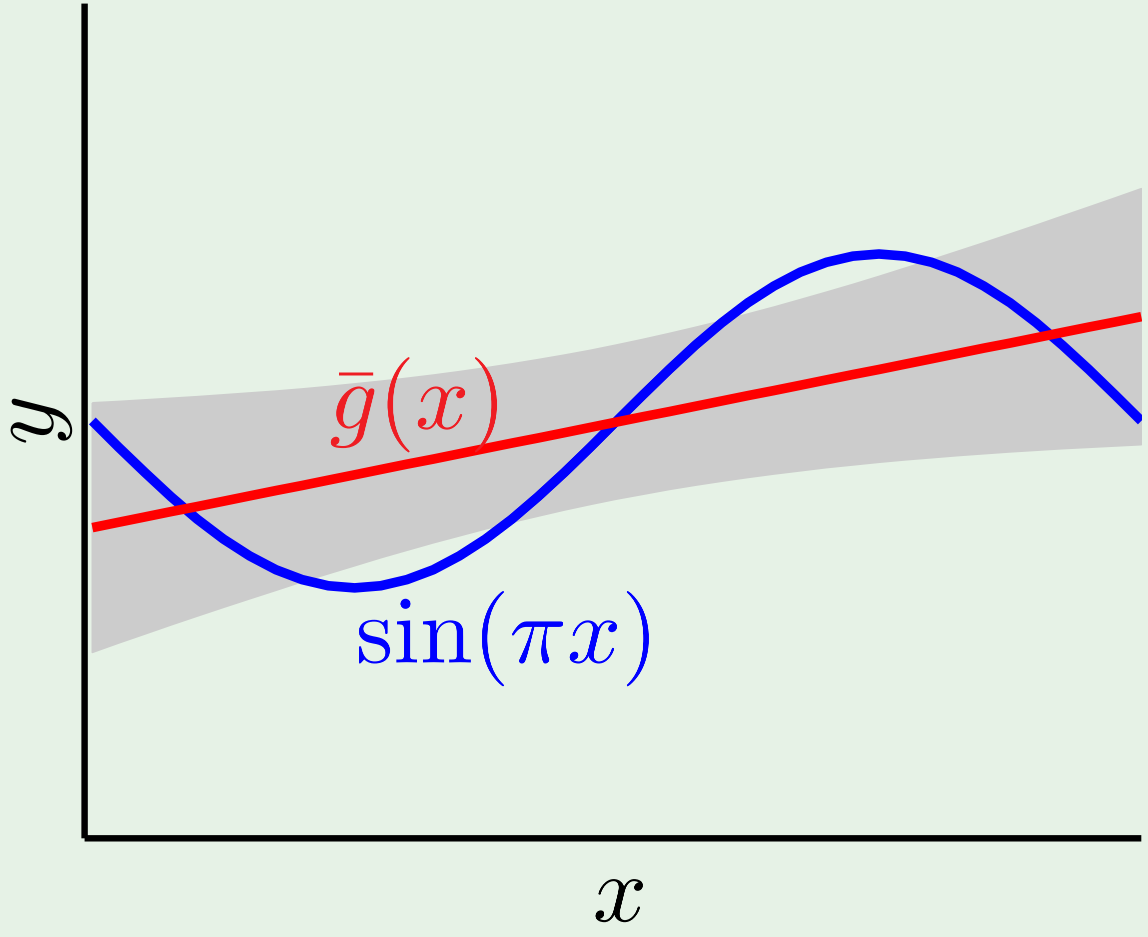
and the winner is ...

without regularization



bias $=$ $\mathbf{0.21}$ var $=$ $\mathbf{1.69}$

with regularization



bias $=$ $\mathbf{0.23}$ var $=$ $\mathbf{0.33}$

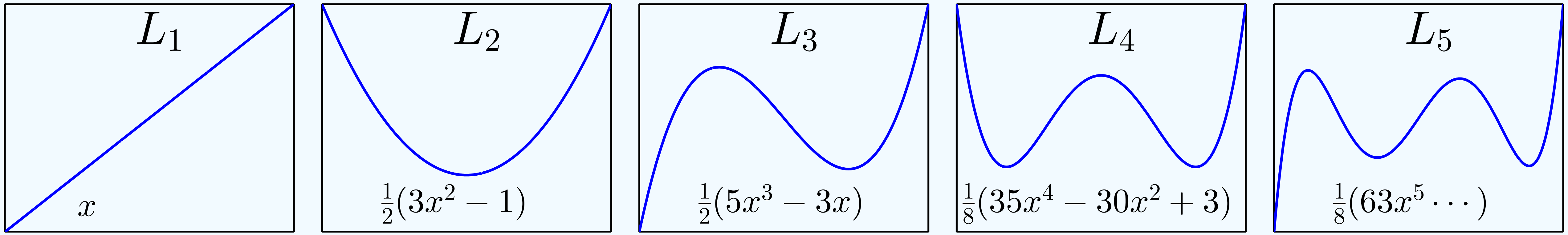
The polynomial model

$\mathcal{H}_{\text{Q}}$: polynomials of order Q

linear regression in $\mathcal{Z}$ space

$$\mathbf{z} = \begin{bmatrix} 1 \\ L_1(x) \\ \vdots \\ L_{\text{Q}}(x) \end{bmatrix} \qquad \mathcal{H}_{\text{Q}} = \left\{ \sum_{q=0}^{Q} w_q \, L_q(x) \right\}$$

Legendre polynomials:



Unconstrained solution

Given $(x_1, y_1), \cdots, (x_N, y_n) \quad \longrightarrow \quad (\mathbf{z}_1, y_1), \cdots, (\mathbf{z}_N, y_n)$

Minimize $E_{\text{in}}(\mathbf{w}) = \dfrac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^{\mathsf{T}} \mathbf{z}_n - y_n)^2$

Minimize $\frac{1}{N} (\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathrm{Z}\mathbf{w} - \mathbf{y})$

$$\mathbf{w}_{\text{lin}} = (\mathrm{Z}^{\mathsf{T}}\mathrm{Z})^{-1}\mathrm{Z}^{\mathsf{T}}\mathbf{y}$$

pseudo inverse

Constraining the weights

Hard constraint: $\mathcal{H}_2$ is constrained version of $\mathcal{H}_{10}$ with $w_q = 0$ for $q > 2$

Softer version: $$\sum_{q=0}^{Q} w_q^2 \le C$$ "**soft-order**" constraint

Minimize $\frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathbf{w} - \mathbf{y})$

subject to: $\mathbf{w}^{\intercal}\mathbf{w} \le C$

Solution: $\mathbf{w}_{\text{reg}}$ instead of $\mathbf{w}_{\text{lin}}$

From UBC Machine Learning Course

Optimization approach

Our aim is to mininimise the quadratic cost between the output labels and the model predictions

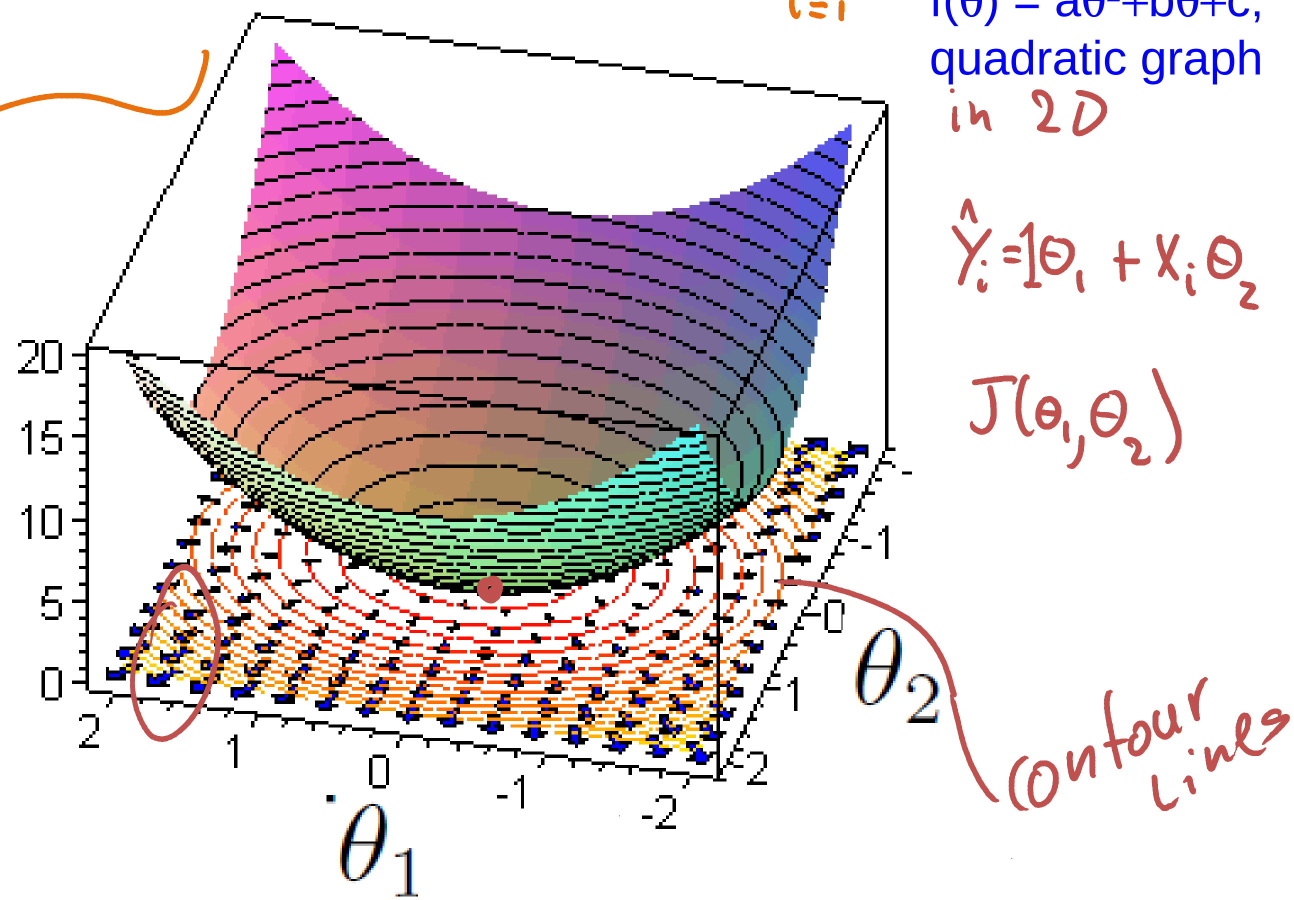
$$\underbrace{J(\boldsymbol{\theta}) = (\underset{n\times 1}{\mathbf{y}} - \overset{n\times d}{\mathbf{X}}\underset{d\times 1}{\boldsymbol{\theta}})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})}_{1\times 1} = \sum_{i=1}^{n}(\underset{1\times 1}{y_i} - \mathbf{x}_i^T\boldsymbol{\theta})^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

f(θ) = aθ²+bθ+c; quadratic graph in 2D

$\hat{y}_i = 1\theta_1 + x_i\theta_2$

$J(\theta_1, \theta_2)$

$\theta^2 <= C$
or $2\theta^2 <= 2C$
or $\theta^2+\theta^2 <= \mathrm{sqrt}(2C)^2$



Solving for $\mathbf{w}_{\text{reg}}$

Minimize $\quad E_{\text{in}}(\mathbf{w}) \; = \; \frac{1}{N}\,(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathsf{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y})$

subject to: $\quad \mathbf{w}^{\mathsf{T}}\mathbf{w} \leq C$

derive: $W^TW \Rightarrow W^2$;
der$(W^2)=2W$

$$\nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) \propto \; -\mathbf{w}_{\text{reg}}$$

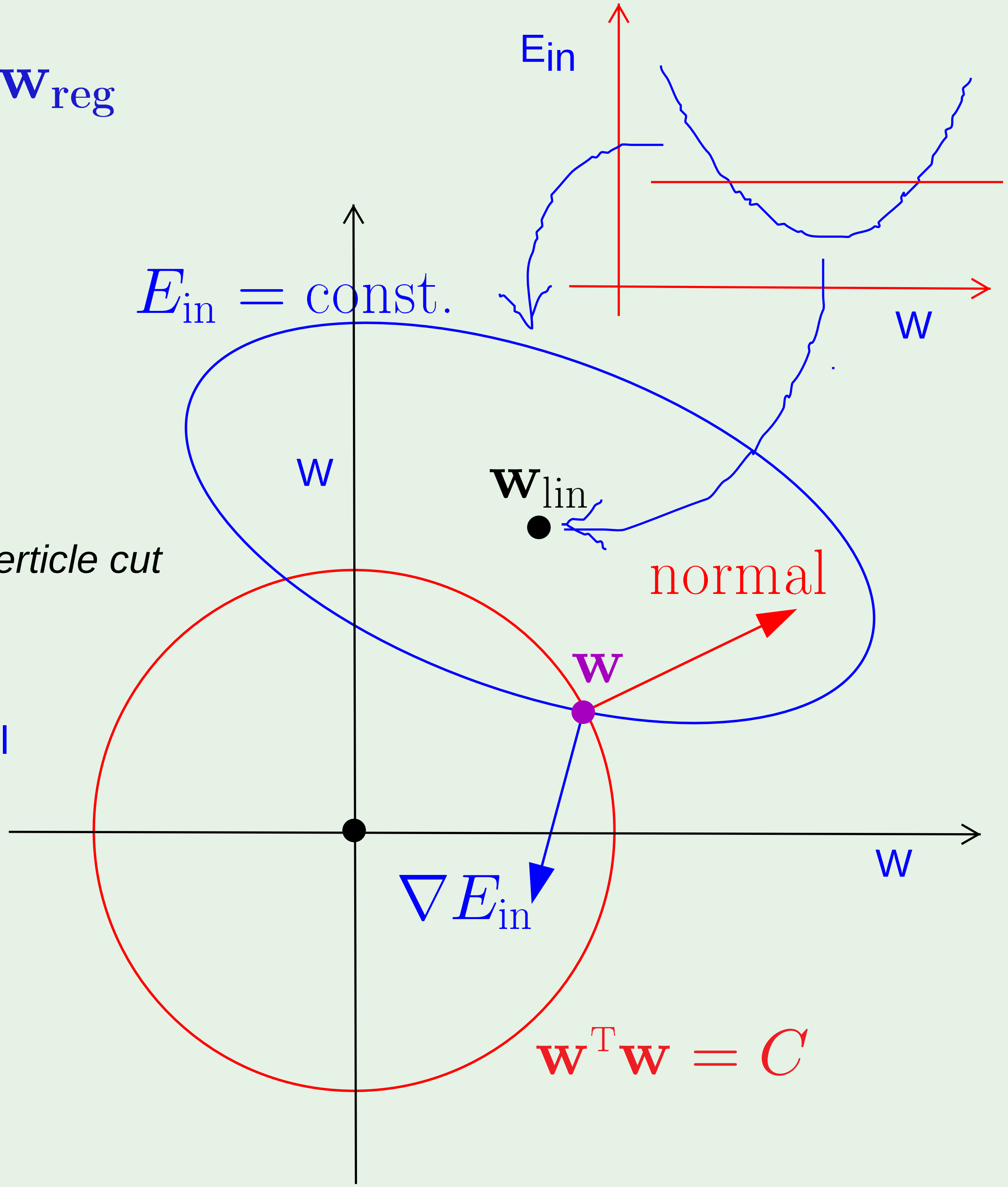
$$= \; -2\frac{\lambda}{N}\mathbf{w}_{\text{reg}}$$

Math Theorem: gradients are always normal to level curves.

$$\nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) + 2\frac{\lambda}{N}\mathbf{w}_{\text{reg}} \; = \; \mathbf{0}$$

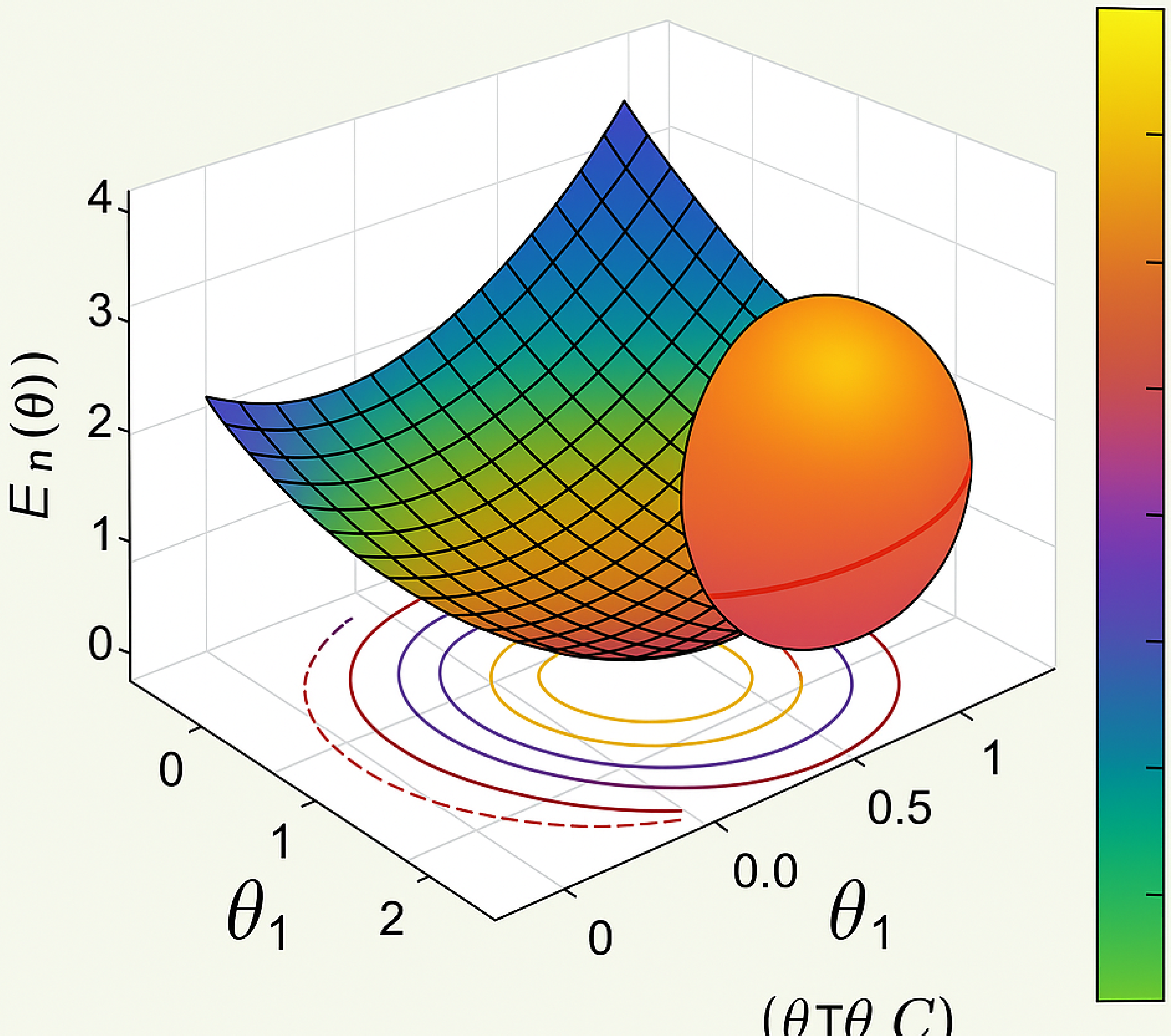
Minimize $\quad E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\mathsf{T}}\mathbf{w} \qquad \boxed{C \uparrow \quad \lambda \downarrow}$

verticle cut



Paraboloid Surface with Constraint Sphere

$(\theta\,\theta \leq C)$



Augmented error

Minimizing $\quad E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\intercal}\mathbf{w}$

$$= \frac{1}{N}\,(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathbf{w} - \mathbf{y}) + \frac{\lambda}{N}\mathbf{w}^{\intercal}\mathbf{w}$$ unconditionally

– solves –

Minimizing $\quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\,(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathbf{w} - \mathbf{y})$

subject to: $\quad \mathbf{w}^{\intercal}\mathbf{w} \le C \qquad \longleftarrow$ VC formulation

The solution

Minimize $$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

$$= \frac{1}{N}\left((\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathsf{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) + \lambda\, \mathbf{w}^{\mathsf{T}}\mathbf{w}\right)$$

$$\nabla E_{\text{aug}}(\mathbf{w}) = \mathbf{0} \implies \mathrm{Z}^{\mathsf{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w} = \mathbf{0}$$

zzw-zy+kw
w(zz+kI) = zy
w = (zz+kI)$^{-1}$.zy

$$\boxed{\mathbf{w}_{\text{reg}} = (\mathrm{Z}^{\mathsf{T}}\mathrm{Z} + \lambda\mathrm{I})^{-1}\,\mathrm{Z}^{\mathsf{T}}\mathbf{y}}$$ (with regularization)

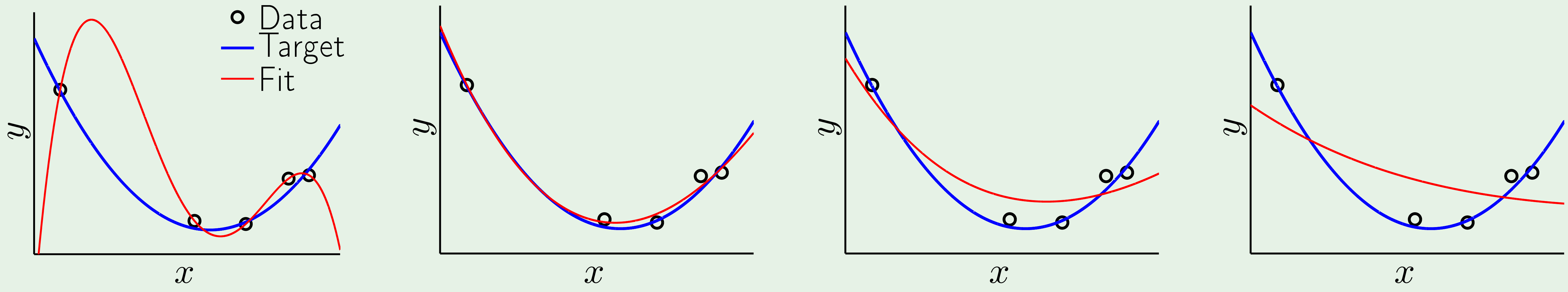
as opposed to $$\mathbf{w}_{\text{lin}} = (\mathrm{Z}^{\mathsf{T}}\mathrm{Z})^{-1}\mathrm{Z}^{\mathsf{T}}\mathbf{y}$$ (without regularization)

pinv

The result

Minimizing $\quad E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\, \mathbf{w}^{\mathsf{T}}\mathbf{w} \quad$ for different λ's:

$\lambda = 0 \qquad \lambda = 0.0001 \qquad \lambda = 0.01 \qquad \lambda = 1$



$W_{reg} = W_{lin}$ $\qquad$ W_{reg} is not reaching W_{lin}

overfitting $\longrightarrow \quad \longrightarrow \quad \longrightarrow \quad \longrightarrow$ **underfitting**

Weight 'decay'

Minimizing $\quad E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\,\mathbf{w}^{\mathsf{T}}\mathbf{w} \quad$ is called weight *decay*. Why?

Gradient descent:

$$\mathbf{w}(t+1) = \mathbf{w}(t) \;-\; \eta\,\nabla E_{\text{in}}\left(\mathbf{w}(t)\right) \;-\; 2\,\eta\,\frac{\lambda}{N}\,\mathbf{w}(t)$$

$$= \mathbf{w}(t)\;(1 - 2\eta\frac{\lambda}{N}) - \eta\,\nabla E_{\text{in}}\left(\mathbf{w}(t)\right)$$

Applies in neural networks:

$$\mathbf{w}^{\mathsf{T}}\mathbf{w} = \sum_{l=1}^{L}\sum_{i=0}^{d^{(l-1)}}\sum_{j=1}^{d^{(l)}}\left(w_{ij}^{(l)}\right)^2$$

Variations of weight decay

Emphasis of certain weights: $\displaystyle\sum_{q=0}^{Q} \gamma_q \, w_q^2$

Examples:

$\gamma_q = 2^q \implies$ **low-order fit**

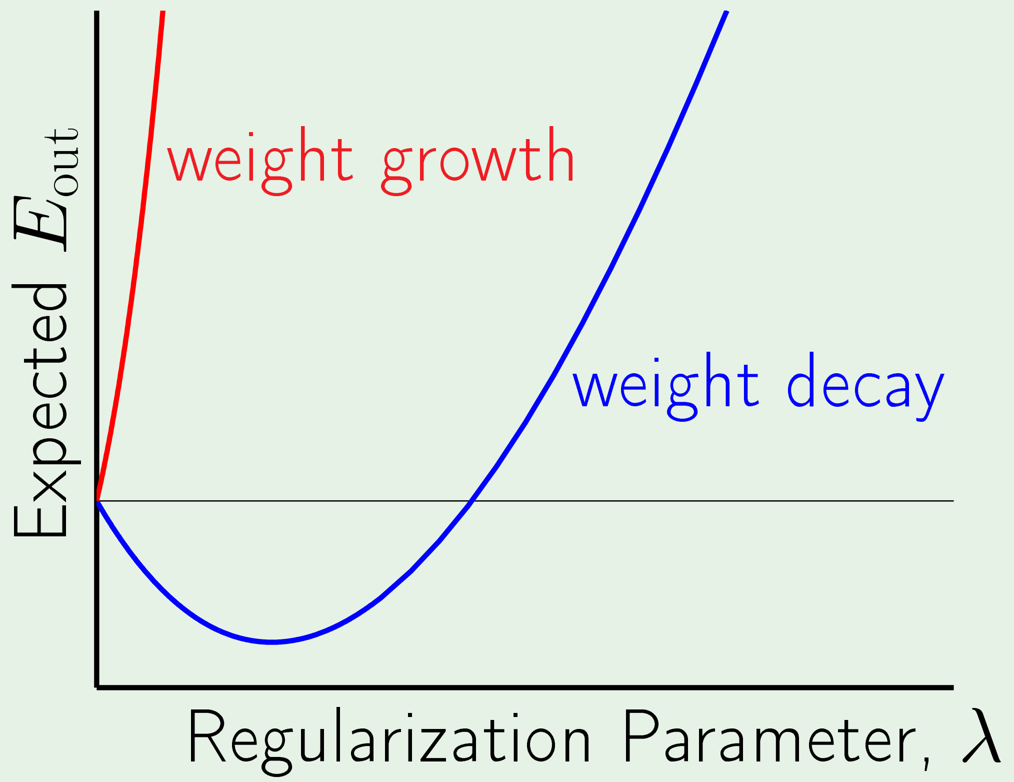
$\gamma_q = 2^{-q} \implies$ **high-order fit**

Neural networks: different layers get different γ's

Tikhonov regularizer: $\mathbf{w}^{\intercal}\Gamma^{\intercal}\Gamma\mathbf{w}$

Even weight growth!

We 'constrain' the weights to be large - bad!



Practical rule:

stochastic noise is 'high-frequency'

deterministic noise is also non-smooth

$\Longrightarrow$ constrain learning towards smoother hypotheses

General form of augmented error

Calling the regularizer $\Omega = \Omega(h)$, we minimize

$$E_{\text{aug}}(h) = E_{\text{in}}(h) + \frac{\lambda}{N}\Omega(h)$$

Rings a bell? $\downarrow \downarrow$

$$E_{\text{out}}(h) \leq E_{\text{in}}(h) + \Omega(\mathcal{H})$$

E_{aug} is better than E_{in} as a proxy for E_{out}

Outline

- Regularization - informal
- Regularization - formal
- Weight decay
- Choosing a regularizer

The perfect regularizer Ω

Constraint in the 'direction' of the target function (going in circles ☺)

Guiding principle:

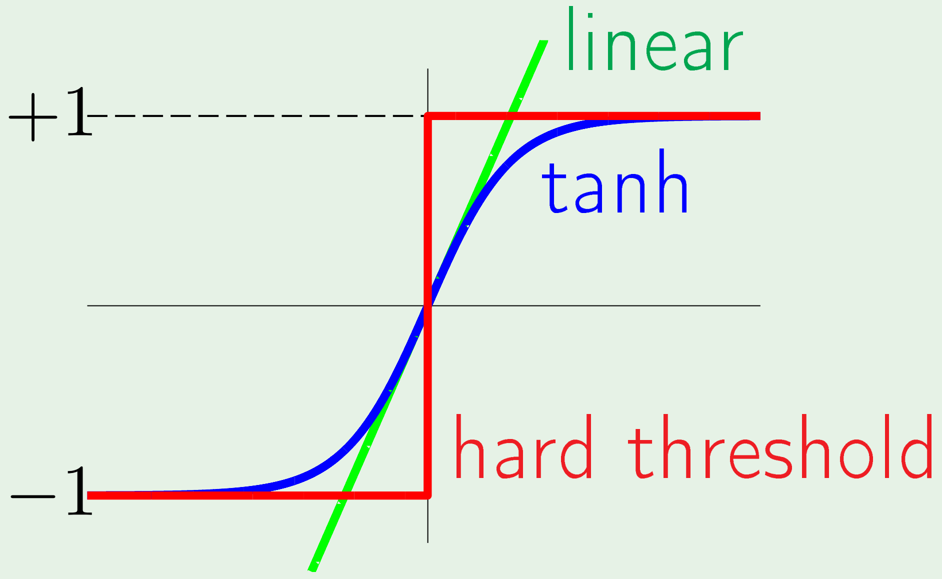
Direction of **smoother** or "simpler"

Chose a bad Ω?

We still have λ!

Neural-network regularizers

Weight decay: From linear to logical



Weight elimination:

Fewer weights $\Longrightarrow$ smaller VC dimension

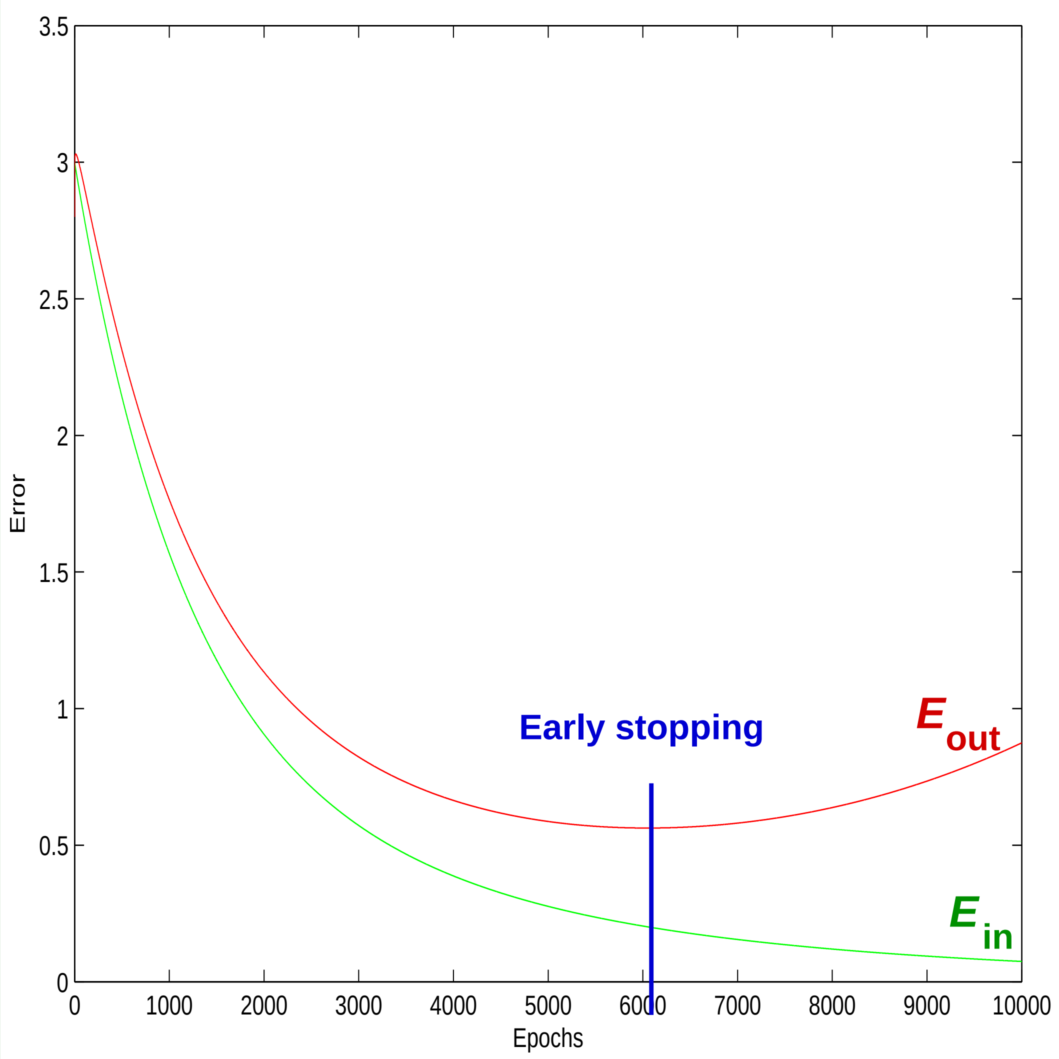
Soft weight elimination:

$$\Omega(\mathbf{w}) = \sum_{i,j,l} \frac{\left(w_{ij}^{(l)}\right)^2}{\beta^2 + \left(w_{ij}^{(l)}\right)^2}$$

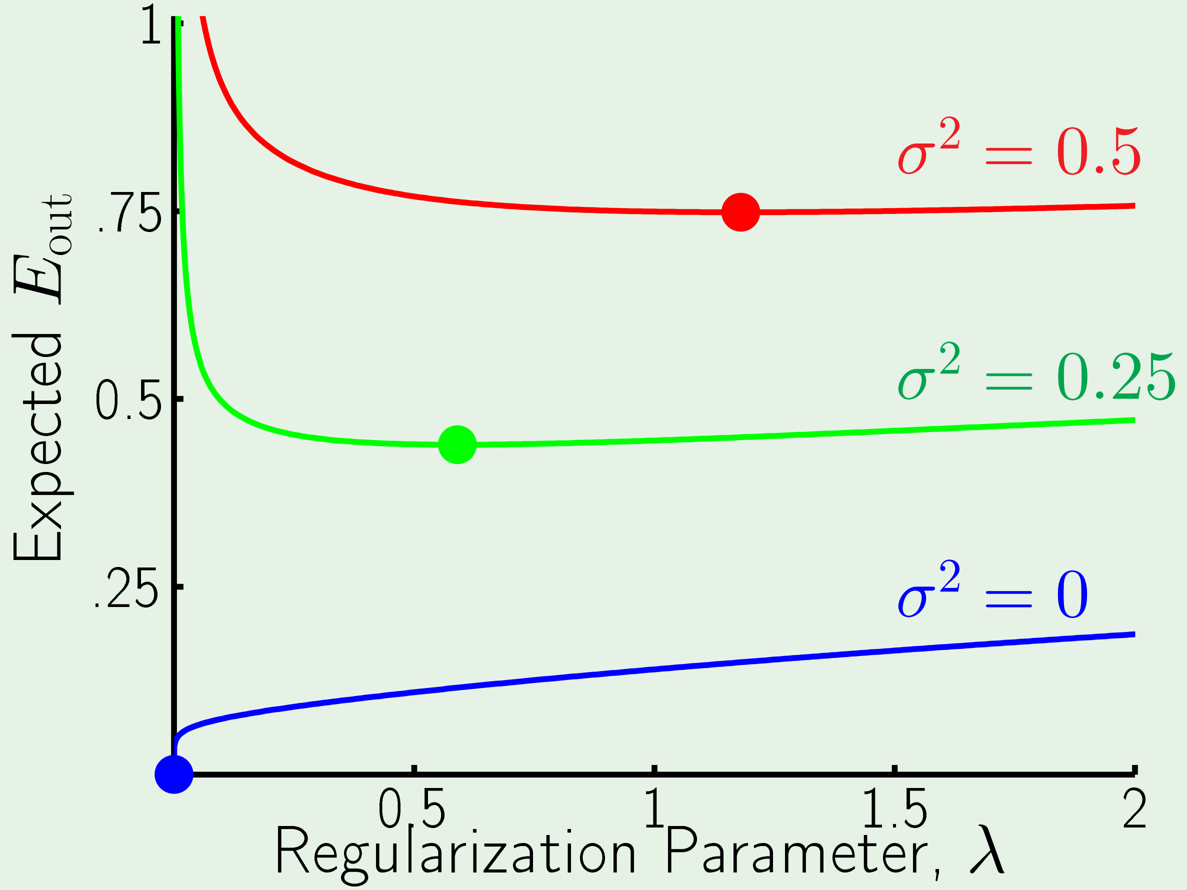
Early stopping as a regularizer

Regularization through the optimizer!

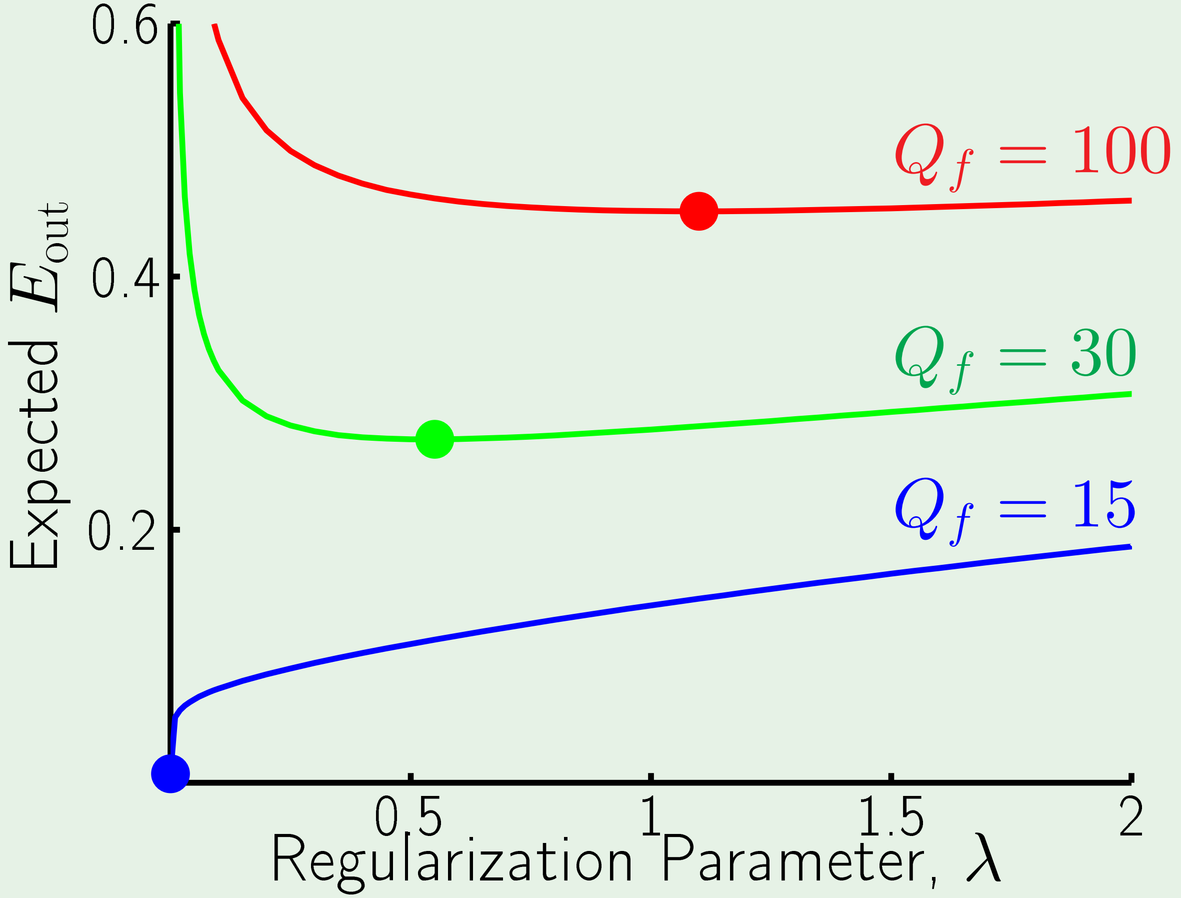
When to stop? **validation**



The optimal λ



Stochastic noise



Deterministic noise