Modeling the problem and remedy

Overfitting

- Overfitting is the most important and common "error" when we try to fit a model

A "process" is overfitting the data sample when choosing *h* with smaller E_{in} means higher E_{out}

- According to the VC-bound, $E_{out}(g) \leq E_{in}(g) + \Omega(N, \mathcal{H}, \delta)$, but the penalty function increase very fast with the $\mathcal{H}$'s VC-dimension.

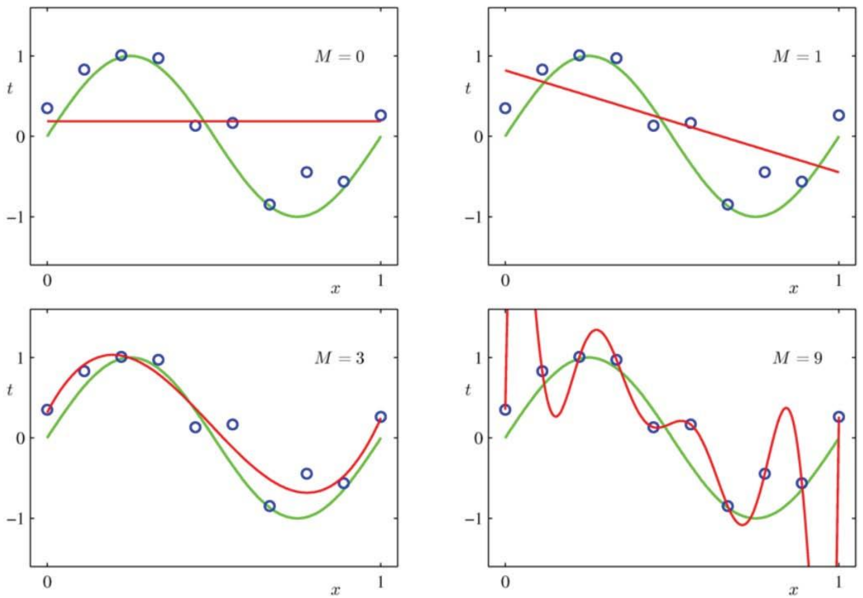
- Why this happen?

1. STOCHASTIC ERROR : Noisy labeling, hence more complex functions are needed to get better in-sample-error

2. DETERMINISTIC NOISE : Noise from model. The complexity of the true function is not well represented by the data sample

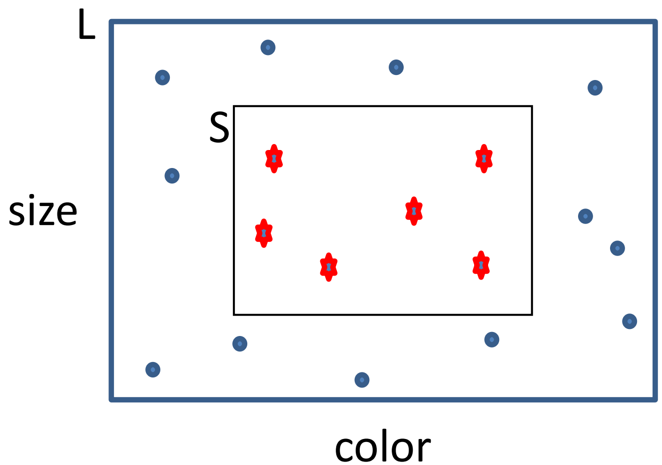
What is overfitting?

- Overfitting means low error in training and high error in test
- Overfitting is the main source of error in M.L. applications
- Usually appears when our model explains the training data too well.
- In general is not easy to detect overfitting since depend of unknow entities (data noise)
- Most of the time overfitting is the consequence of considering a set of function $\mathcal{H}$ more complex than required……but not always !



The ERM rule: appealing but uncertain

- Let's assume that we choose {color , size} as features to identify the tasty mangos.
- Let's assume that the whole mango population is uniformly distributed inside the box L
- Let $S=\{(x_i , y_i)\}$, $i=1,..,\mathcal{N}$ be a random sample of mangos from the whole population
- Let's stars and circles represent tasty and non-tasty mangos respectively



Let's assume that the unknow label function is

$$f(x) = \begin{cases} 0 & if\ x \in L - S \\ 1 & if\ x \in S \end{cases}$$

Where Area(L) = 2xArea(S)

ERM solution

Let's apply the ERM rule ⇨

$$g_S(x) = \begin{cases} y_i & if\ \exists i \in m\ s.t.\ x = x_i \\ 0 & otherwise \end{cases}$$

The error of $g_S(x)$ on the training set is ZERO
The error of $g_S(x)$ on the rest of points is 50%

⇨ OVERFITTING !!

How to protect against overfitting?

- PROBLEM: How to decide the right complexity of the solution?
  - The noise adds independent information to the sample data
  - The ERM/SRM criteria is responsible of the final selection

- SOLUTION-1: A hard-way is to restrict the size of the $\mathcal{H}$ set. (ERM)
  - We restrict the capacity of $\mathcal{H}$ to fit noise.
  - BUT, we also restrict the capacity to find the right solution.
  - The restriction to a particular set of functions $\mathcal{H}$ is called " inductive bias"

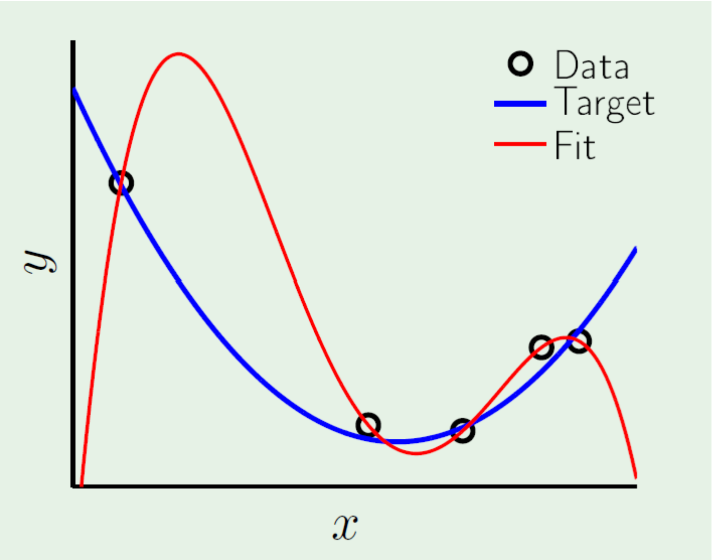
- SOLUTION-2: A softer way is to impose additional conditions on the error function
  - We get a compromise between the best fitting function and its complexity
  - It is soft since the compromise is fixed by a weigthing parameter
  - This technique is called "regularization"

Both approaches INDUCTIVE BIAS / REGULARIZATION can be seen as using some type of prior knowledge

QUESTION: is INDUCTIVE BIAS / REGULARIZATION necessary for the success of learning ?

Overfitting

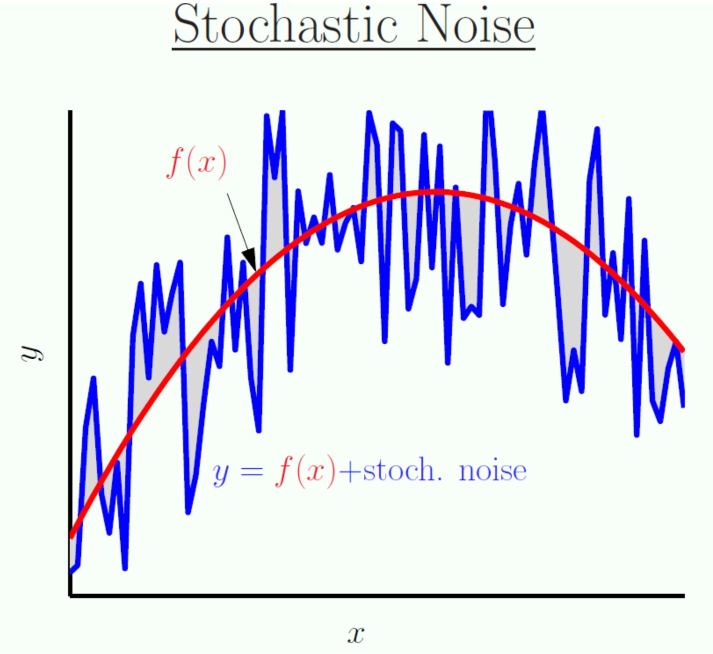
Simple one-dimensional regression example with 5 data plus some noise

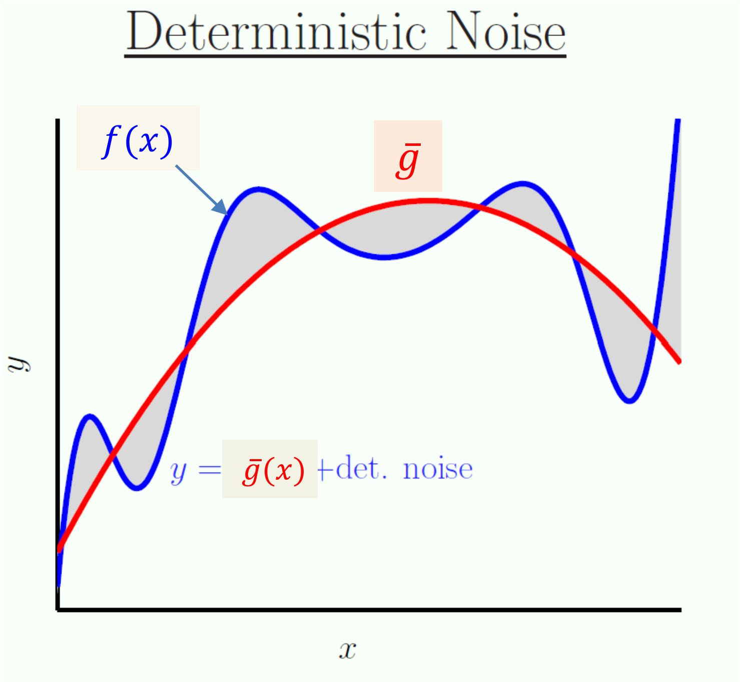


- In blue we show the true function generating the data, 2nd order polynomial
- In red we show the fitted function with zero in-sample-error. A 4th-order polynomial
- The sample have been overfitted !!
- Little noise in the data has mislead the learning

- The fit has zero in-sample-error but huge out-of-sample-error
- In the Bias-Variance treadoff we get BIAS=0 (in sample) but the price is to increase the VARIANCE very much.
  - $\mathbb{E}_D\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \sigma^2 + \textbf{bias} + \textbf{variance}$ (for noisy signals)

Noise: what we cannot model





Stochastic noise: i.i.d random noise added to each data

Deterministic noise: The part of the target function outside of the best fit $\bar{g}$

$$y = g^*_{\mathcal{D}}(x) + \text{noise} \qquad \text{noise} = \text{stoch. noise} + \text{det. noise}(\mathcal{H})$$

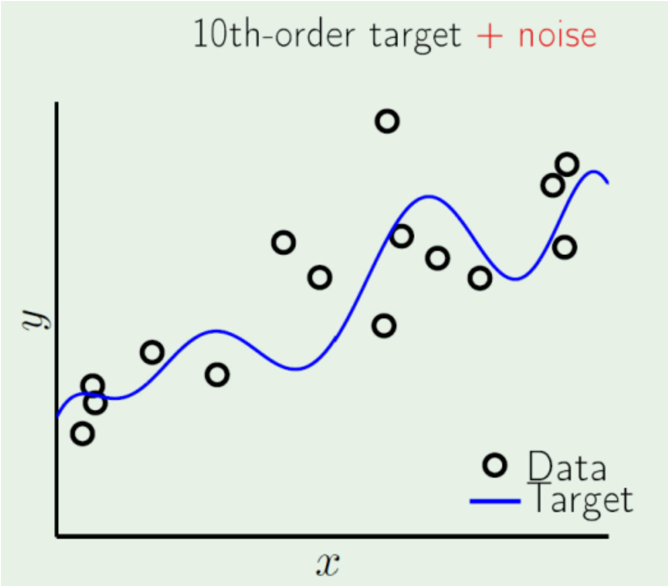
With a given data set $\mathcal{D}$ and $\mathcal{H}$ fixed , we can't differentiate between both types of noise

$$\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \sigma^2 + \texttt{bias} + \texttt{var} = \texttt{stoch.noise + det.noise + var}$$

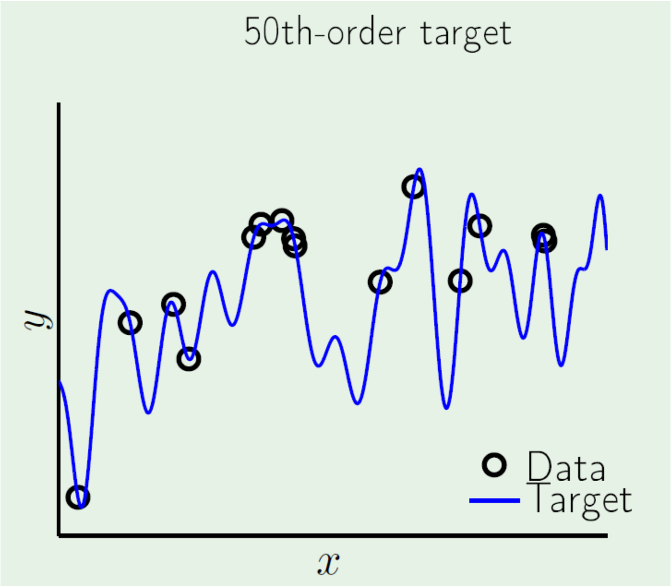
Overfitting: A case study

- Let consider two regression problems.
- In both cases we have 15 polynomial data (10th and 50th order respec.)

With added noise

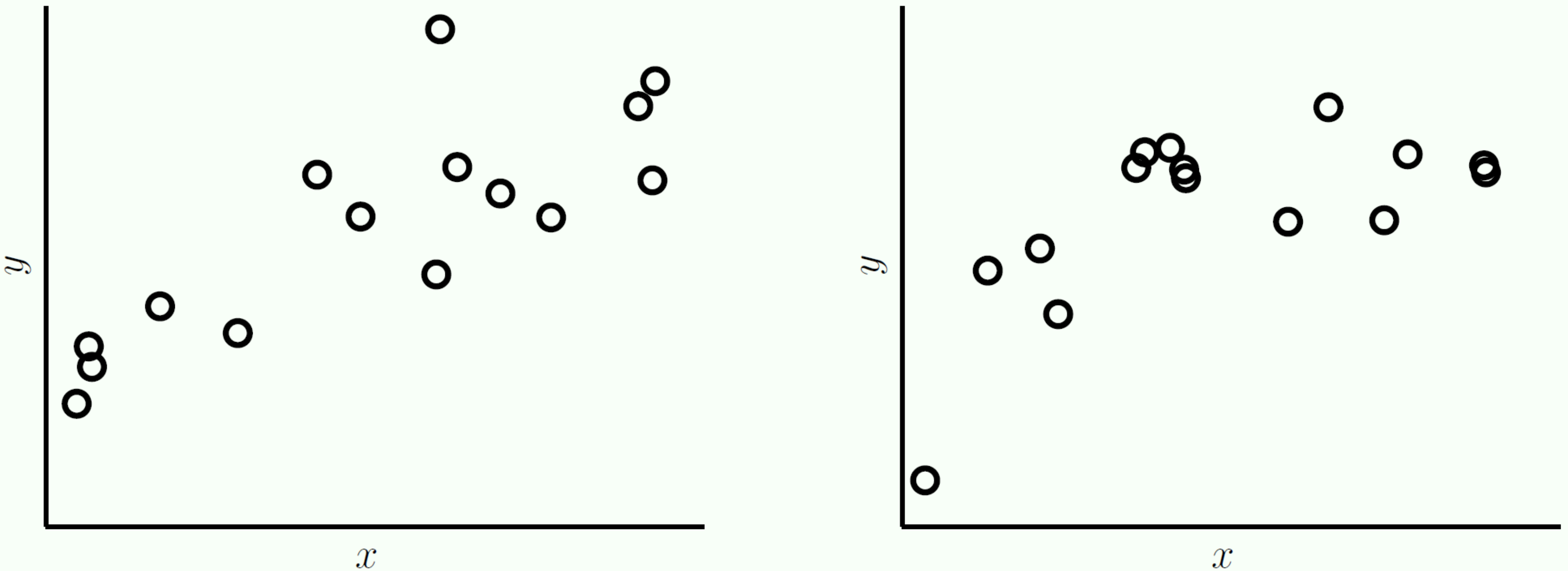


Noiseless



- Let's fit in both cases two polynomial: low and high order (2nd and 10th)
- Let analyze which of both produce lower out-of-sample error

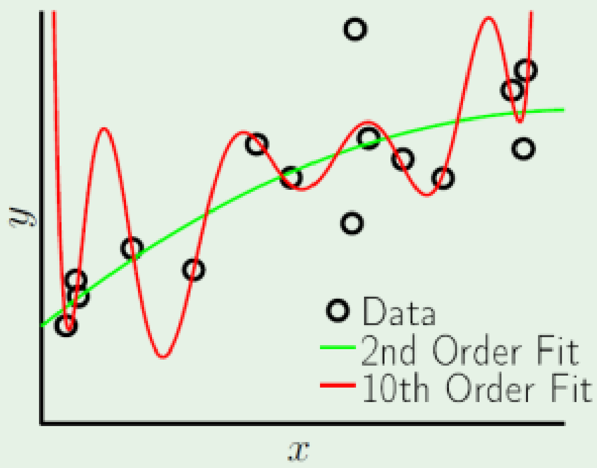
Can the noise be distinguished?



Simple f with noise. Complex f with no noise.

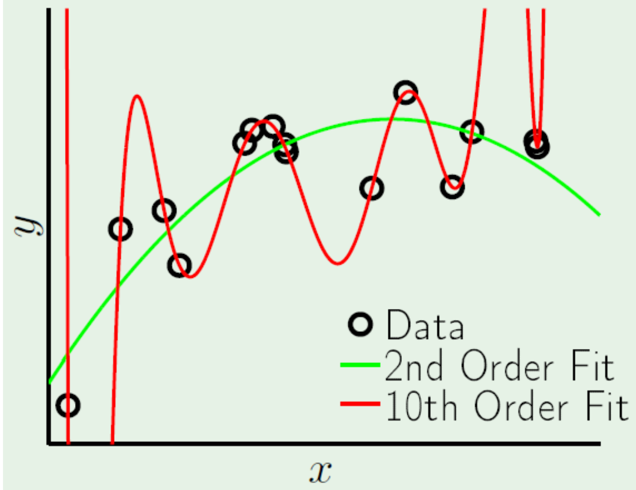
- The learning model should match the quality and quantity of the data NOT f

Overfitting: A case study



Noisy low-order target

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

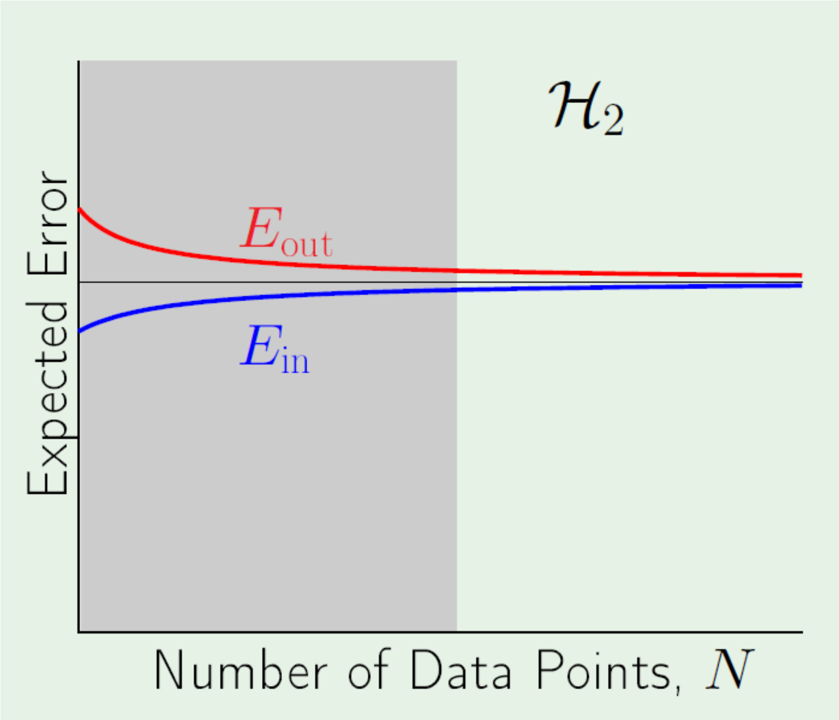


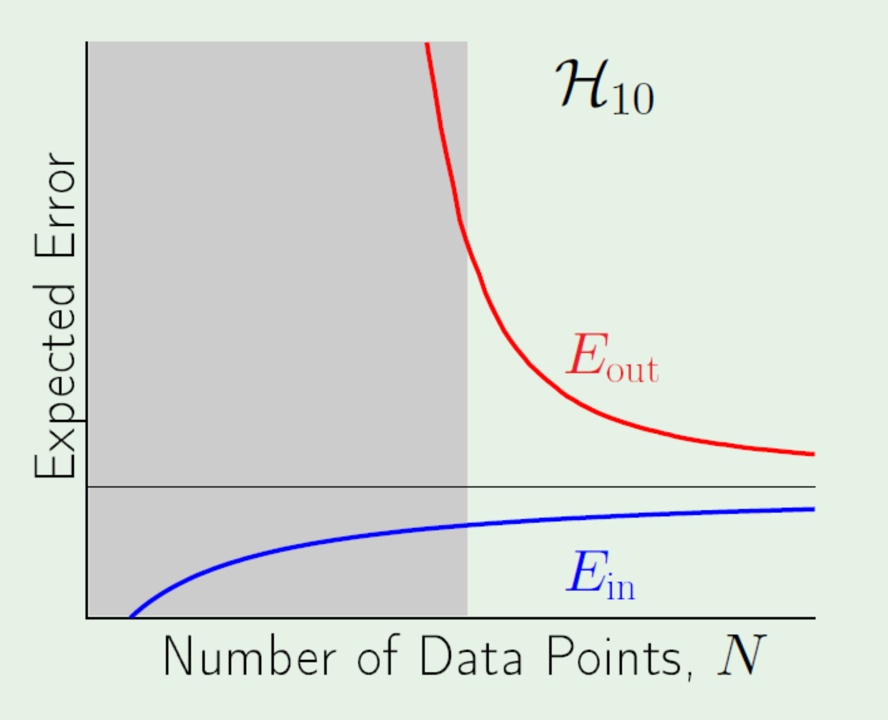
Noiseless high-order target

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

- The figures show the in-sample and out-of-sample errors on each case
- It can be observed the smaller order polynomial presents higher in-sample error but smaller out-of sample error in both cases.
- On the left the reason is the stochastic noise, and on the right the reason is the deterministic noise

Learning curves: overfitting



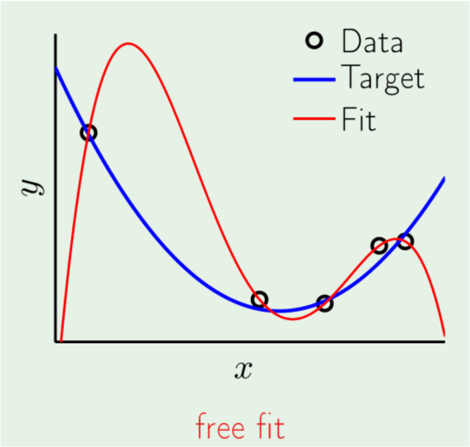


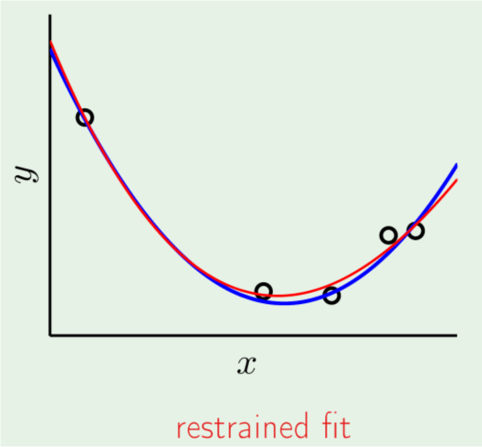
- The gray area shows the range of N values, where $\mathcal{H}_{10}$ has lower E_{in} and higher E_{out}: overfitting is present.
- The learning curves show typical behaviour of a simple and a complex model respectively.
- These pictures show the **importance of the data size in the overfitting**

REGULARIZATION: An smart mechanism to implement SRM

Regularization

- Idea: Constraint the learning model to improve the out-of-sample error





The figures show the dramatic improvement in the fit with a small amount of regularization

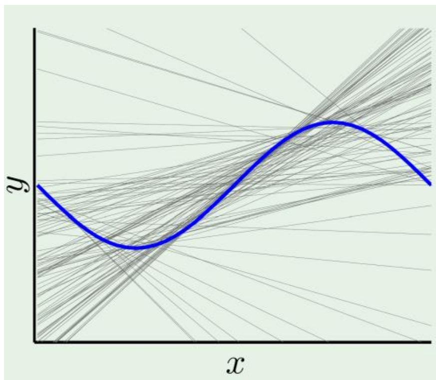
- **Regularization is an heuristic approach** although is in close connection with the optimization techniques

- According to the Approx.-Genera. tradeoff $E_{out}(g) \leq E_{in}(g) + \Omega(\mathcal{H})$, regularization minimizes the right hand of the inequality not only the in-sample error

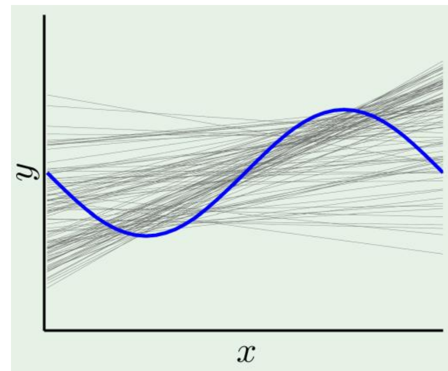
- According to the Bias-Variance tradeoff, regularization increases lightly the Bias to strongly decrease the Variance

Constraining the weights helps: Weight Decay

- The **weight decay** technique measures the complexity of a hypothesis h by the size of the coefficients used to represent h.



without regularization



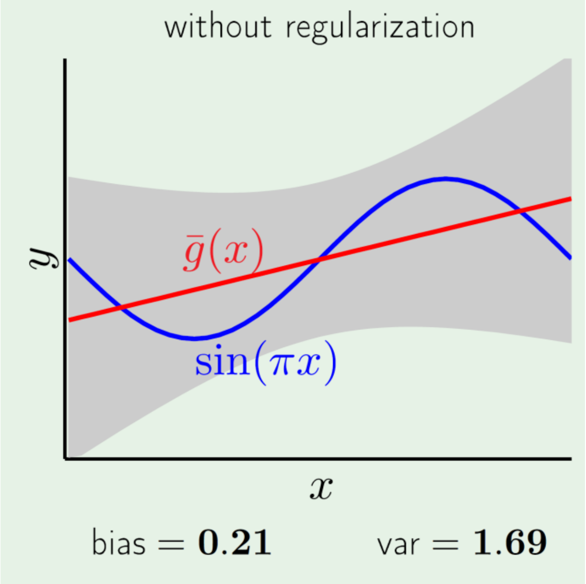
with regularization

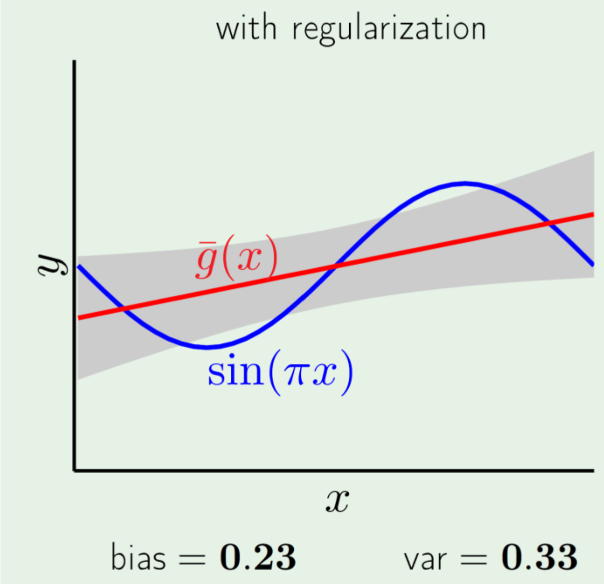
- The figure shows the result of applying weight decay to fit the target $f(x)=\sin(\pi x)$, $x \in [-1,1]$, using samples of N=2 (lines) , x is sampled uniformly in [-1,1]

- Without regularization shows a very high variability in the learning function depending on the sample x

- With regularization (constraining weights to be small) shows how the set of learning functions is much more stable

Constraining the weights helps





- Let analyze the learning using the Bias-Variace tradeoff
- Without regularization we observe a lower bias and higher variance
- With regularization we observe one light increased bias and a large decrease in variance
- In total the regularization provides a learned function with smaller out-of-sample error
- Regularization: we sacrifice a little bias for a significant gain in var

Regularization : a SRM rule

- (Weight Decay) The in-sample optimization problem becomes

$$\min_{\mathbf{w}} E_{in}(\mathbf{w}) \quad \text{subject to } \mathbf{w}^T\mathbf{w} \leq C \quad \text{(constraint problem)}$$

the learning algorithm choose the best solution $\boldsymbol{w}_{reg}$, given the total budget *C.*

- The *C* value defines a constraint on the class of hypothesis:
  - Clearly if $C_1 < C_2$ then $\mathcal{H}(C_1) \subset \mathcal{H}(C_2)$ and so $d_{VC}(\mathcal{H}(C_1)) \leq d_{VC}(\mathcal{H}(C_2))$, we expect better generalization error with $\mathcal{H}(C_1)$

$$\min_{\mathbf{w}} E_{in}(\mathbf{w}) \quad \text{subject to } \mathbf{w}^T\mathbf{w} \leq C \iff \min_{\mathbf{w}} \; E_{in}(\mathbf{w}) + \lambda_C \, \mathbf{w}^T\mathbf{w}, \quad \lambda_C > 0$$

Using Lagrange Multipliers $\min_{\mathbf{w}} \{E_{in}(\mathbf{w}) + \lambda(\mathbf{w}^T\mathbf{w} - C)\} \Longrightarrow E_{aug} = E_{in}(\mathbf{w}) + \lambda \, \mathbf{w}^T\mathbf{w}$ (unscontrained)

- The augmented error for a hypotesis $\boldsymbol{w}$ can be written :

$$E_{aug}(\boldsymbol{w}, \lambda, \Omega) = E_{in}(\boldsymbol{w}) + \frac{\lambda}{N}\Omega(\boldsymbol{w})$$

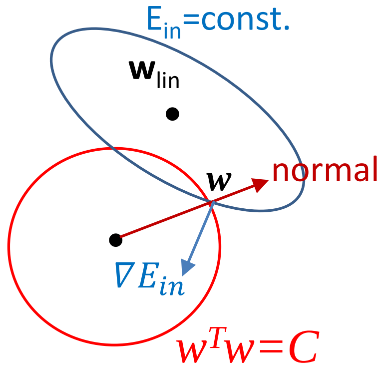
- The λ parameter defines the intensity of the regularization and the "effective VC dimensión"
- For weights decay $\Omega(\boldsymbol{w}) = \mathbf{w}^T\mathbf{w}$ which penalize large weigths

Computing w_{reg}

$w_{reg} = \min_{\mathbf{w}} E_{in}(\mathbf{w})$ subject to $\mathbf{w}^T\mathbf{w} \le C$ (constraint problem)

if $\boldsymbol{w}_{lin}^T \boldsymbol{w}_{lin} \le C$ then $\boldsymbol{w}_{reg} = \boldsymbol{w}_{lin}$, because $\boldsymbol{w}_{lin} \in \mathcal{H}(C)$

if $\boldsymbol{w}_{lin} \notin \mathcal{H}(C)$ *then* $\boldsymbol{w}_{reg}^T \boldsymbol{w}_{reg} = C$



If $\boldsymbol{w}_{reg}$ is to be optimal then $\nabla E_{in}(\boldsymbol{w}_{reg}) = -2\lambda_C \boldsymbol{w}_{reg}$

Rewritten $\nabla\left(E_{in}(\boldsymbol{w}) + \lambda_C \boldsymbol{w}^T\boldsymbol{w}\right)|_{\boldsymbol{w}=\boldsymbol{w}_{reg}} = \mathbf{0}$

Then for some λ_C, $\boldsymbol{w}_{reg}$ locally minimize $E_{in}(\boldsymbol{w}) + \lambda_C \boldsymbol{w}^T\boldsymbol{w}$

λ and $\boldsymbol{w}$ both depend on C, and it is clear that $\lambda_C > 0$

$$\min_{\mathbf{w}} E_{in}(\mathbf{w}) \quad \text{subject to } \mathbf{w}^T\mathbf{w} \le C \quad \Longleftrightarrow \quad \min_{\mathbf{w}} E_{in}(\mathbf{w}) + \lambda_C\, \mathbf{w}^T\mathbf{w}, \quad \lambda_C > 0$$

Augmented Error as a Proxy for E_{out}

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

this was $\mathbf{w}^{\mathsf{T}}\mathbf{w}$

$$\Updownarrow$$

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

this was $\mathcal{O}\left(\sqrt{d_{\text{vc}}\frac{\ln N}{N}}\right)$

E_{aug} **can** (depending on λ) **beat** E_{in} **as a proxy for** E_{out}

Regularization: Penalties

– Soft constraints: imposes that some positive function of the weights be bounded:

Examples: (1) $\sum_{q=0}^{Q} w_q^2 \leq C$, (2) $\sum_{q=0}^{Q} |w_q| \leq C$, (3) $\left(\sum_{q=0}^{Q} w_q\right)^2 \leq C$, (4) $\sum_{q=0}^{Q} \gamma_q w_q^2 \leq C$

– In (1), solutions with low values, but not necessarily zero are encouraged
– In (2), we encourage some values to be zero (LASSO, good for feature selection !)
– In (3), we encourage the same contribution of positive and negative weigths
– In (4), according to the coefficients we encourage the contribution of the weights

- Each restriction encourages a specific solution and defines an optimization problem that must be solved

- **General linear regression problem** : The goal is minimize the in-sample squared error

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

over the hypothesis in $\mathcal{H}_Q$ in order to get $\mathbf{w}_{lin} = \underset{\mathbf{w}}{\operatorname{argmin}}\, E_{in}(\mathbf{w})$

Regularized Regression: Ridge model

- Using matrix notation we have: $E_{aug}(\mathbf{w}) = \|Z\mathbf{w} - \boldsymbol{y}\|^2 + \lambda\|\mathbf{w}\|^2$

- $\mathbf{w}_{reg}$ is the solution of the equation $\nabla_{\mathbf{w}} E_{aug}(\mathbf{w}) = \nabla_{\mathbf{w}} \left(E_{in}(\mathbf{w}) + \lambda \mathbf{w}\mathbf{w}^T\right) = 0$

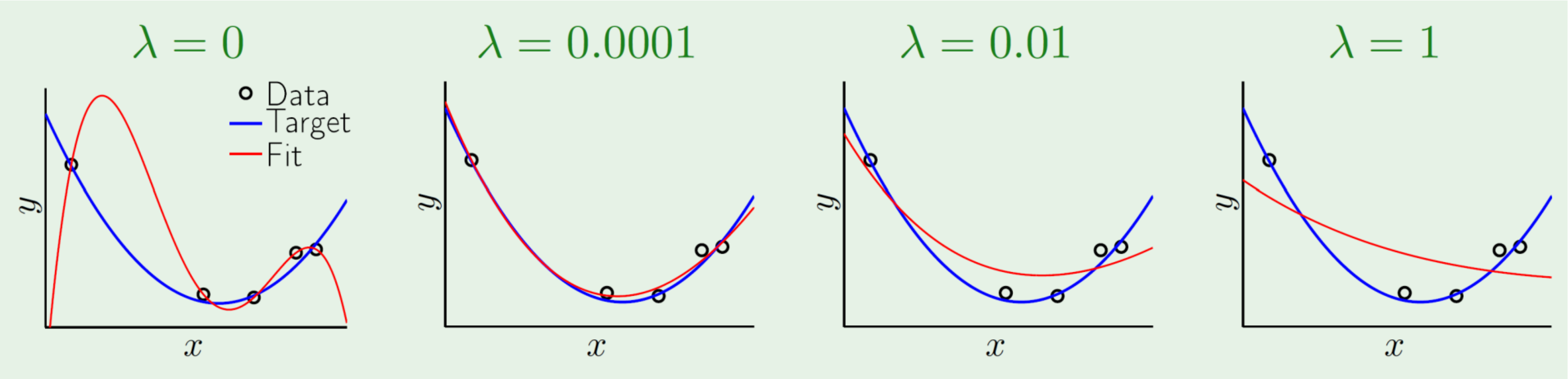
$$\nabla_{\mathbf{w}} E_{aug} = 2\mathrm{Z}^T(\mathrm{Z}\mathbf{w} - \boldsymbol{y}) + \lambda \mathbf{w}^T = 0 \quad \Longrightarrow \quad \mathbf{w}_{reg} = (\mathrm{Z}^{\mathrm{T}}\mathrm{Z} + \lambda \mathrm{I})^{-1}\mathrm{Z}^{\mathrm{T}}\boldsymbol{y}$$

- As expected $\mathbf{w}_{reg} \rightarrow 0$ when $\lambda \rightarrow \infty$
- The predictions on the in-sample data are given by: $\hat{\boldsymbol{y}} = Z\mathbf{w}_{reg} = H(\lambda)\boldsymbol{y}$

$$\mathrm{H}(\lambda) = \mathrm{Z}\left(\mathrm{Z}^{\mathrm{T}}\mathrm{Z} + \lambda \mathrm{I}\right)^{-1}\mathrm{Z}^{\mathrm{T}}$$

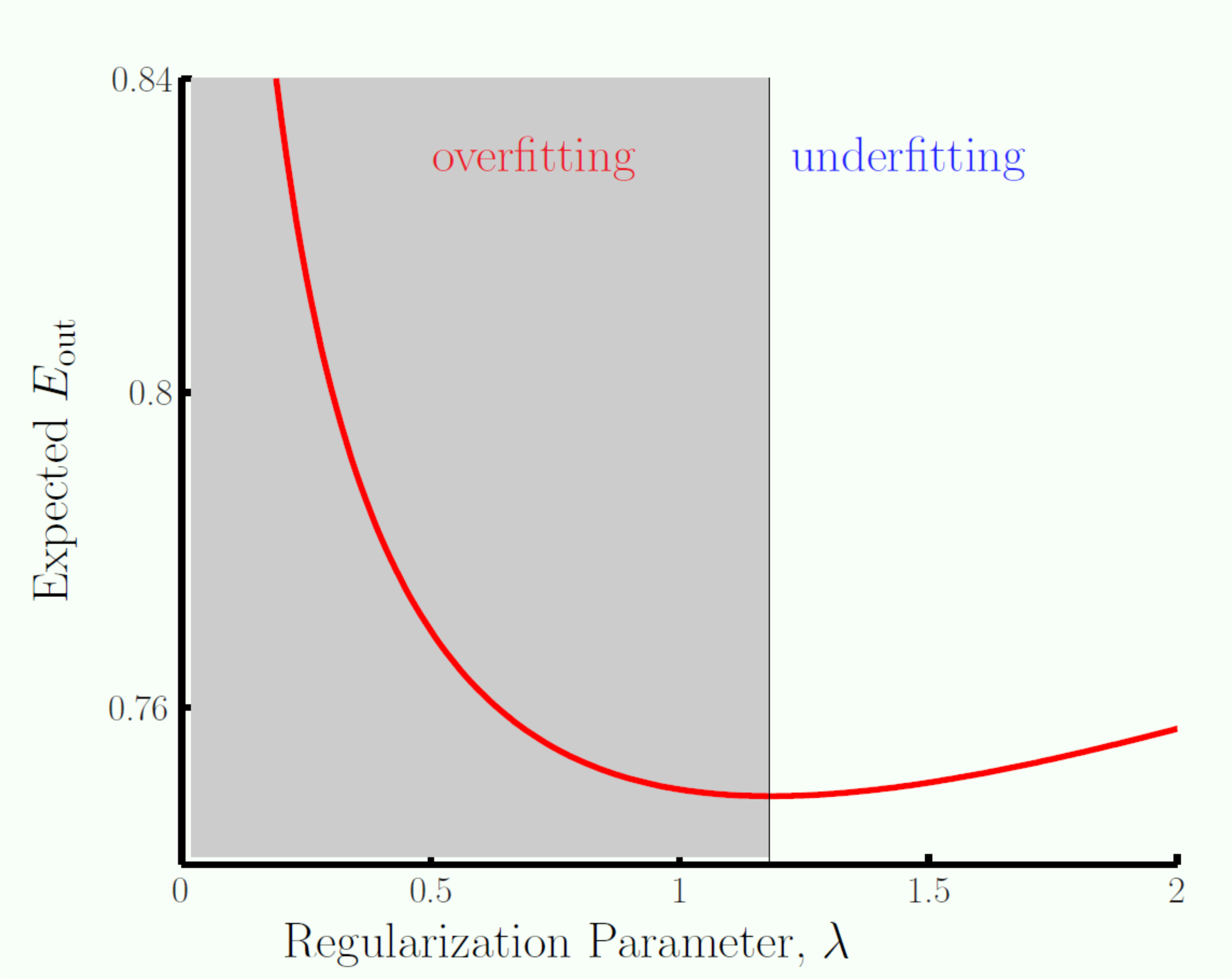
- The matrix hat $\mathrm{H}(\lambda)$ plays a relevant role in defining the efective complexity of the model
  - λ=0, H is the hat-matrix of the linear regression
  - The vector of in-sample errors is : $\boldsymbol{y} - \hat{\boldsymbol{y}} = \left(\mathrm{I} - H(\lambda)\right)\boldsymbol{y}$
  - The in-sample error is : $E_{in}\left(\mathbf{w}_{reg}\right) = \frac{1}{N}\boldsymbol{y}^T(\mathrm{I} - H(\lambda))^2\boldsymbol{y}$

The Influence of Regularization

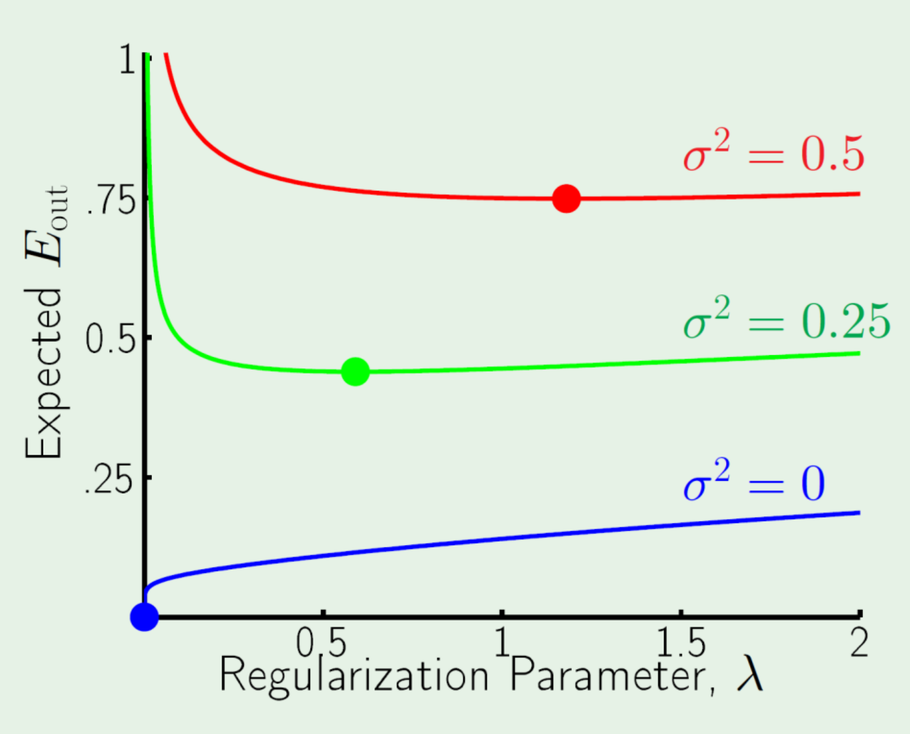


- The figure shows the result of applying different amount of regularization to the same example using weight decay
- It can be seen that non-regularization or too much regularization increases the adjustment error. In the first case due to the variance in the second case due to the bias.

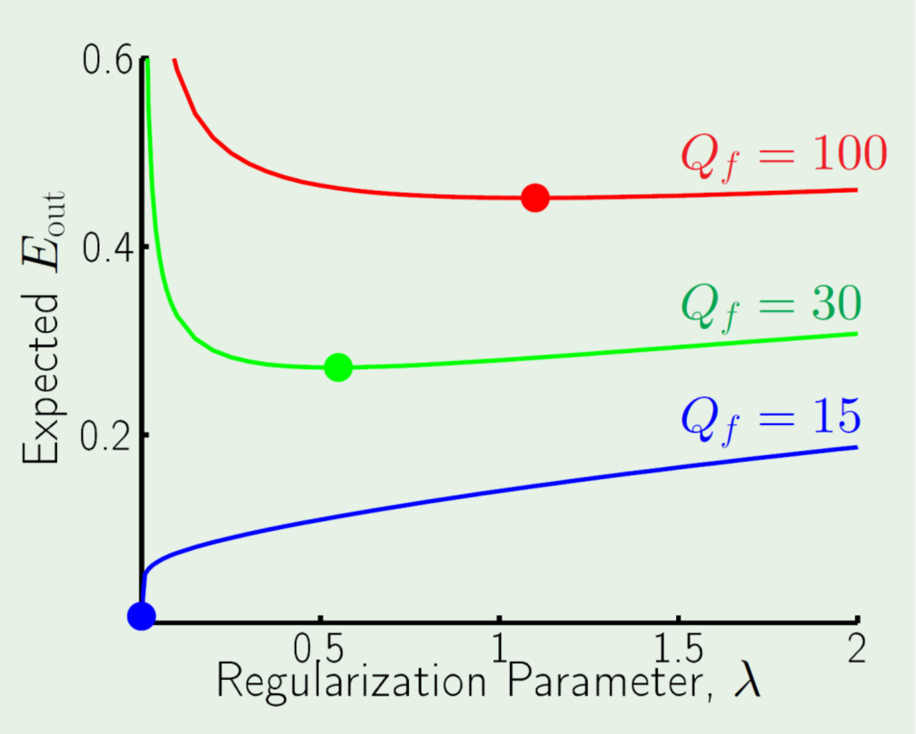
Overfitting & Underfitting



Regularization and noise



Stochastic noise

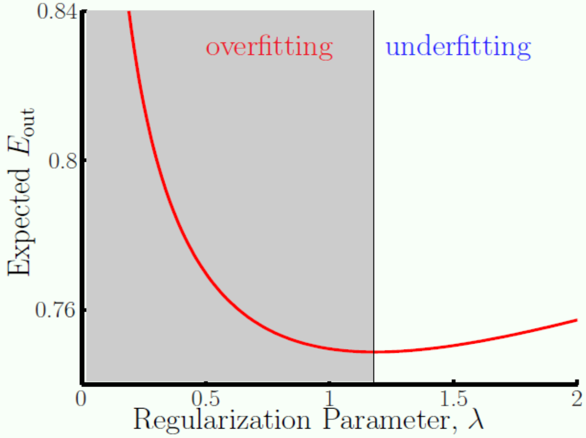


Deterministic noise

Uniform regularizer: $\Omega(\mathbf{w}) = \sum_{q=0}^{15} w_q^2$

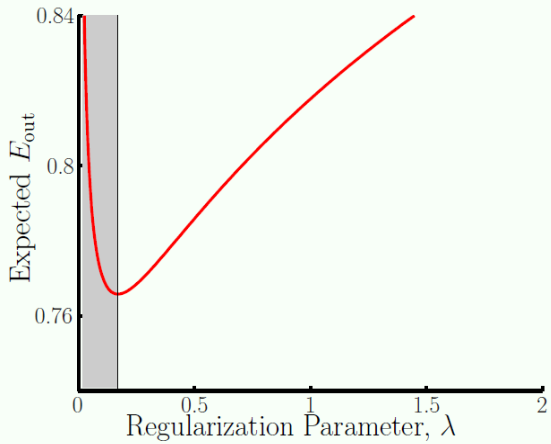
Weight Decay Influence

Uniform Weight Decay



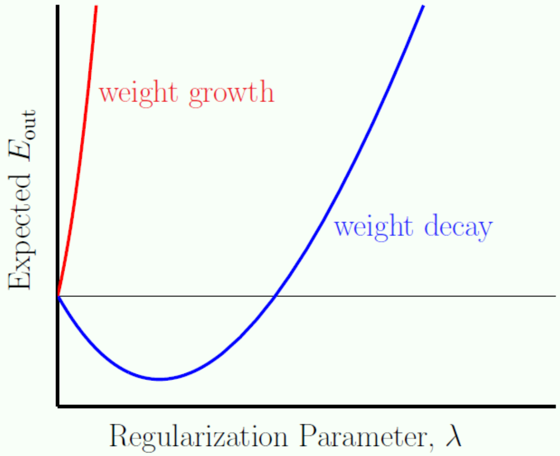
$$\sum_{q=0}^{Q} w_q^2$$

Low Order Fit



$$\sum_{q=0}^{Q} q w_q^2$$

Weight Growth!



$$\sum_{q=0}^{Q} \frac{1}{w_q^2}$$

Choosing a Regularized: A Practitioner's Guide….

- Leasson learned: Some form of regularization is necessary

- The perfect regularizer: does not exist
  - constrain in the 'direction' of the target function.
  - target function is **unknown** (going around in circles).

- The guiding principle:
  - constrain in the 'direction' of smoother (usually simpler) hypotheses
  - hurts your ability to fit the 'high frequency' noise
  - smoother and simpler usually means −→ weight decay not weight growth.

- What if you choose the wrong regularizer?
  - You still have λ to play with — **validation.**