Modeling: the problem and the 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 $\mathcal{H}$ VC-dimension.

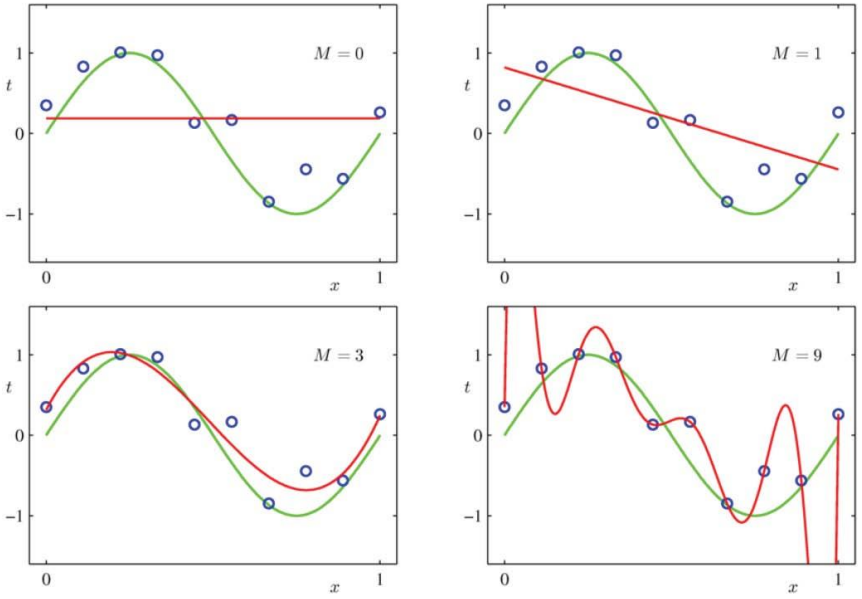
- Why this happen?

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 !



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"

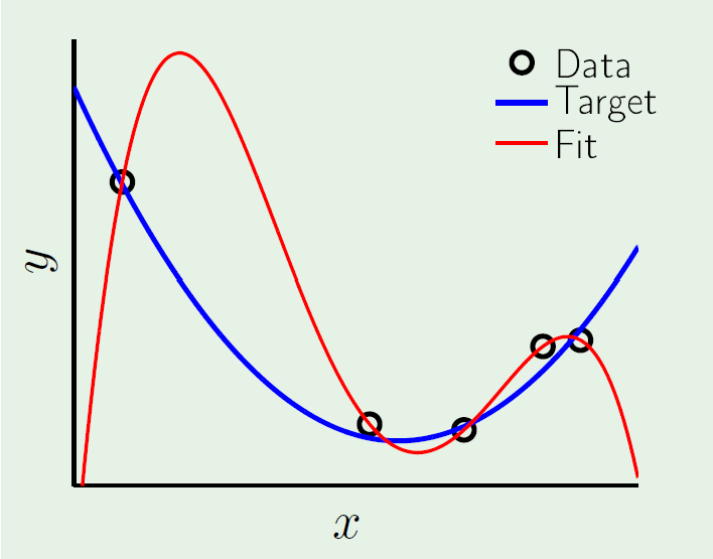
- SOLUTION-2: A softer way is to impose additional conditions on the error function
  - We get a compromise between the best fitting function and its complexity
  - It is soft since the compromise is fixed by a weigthing parameter
  - This technique is called "regularization"

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



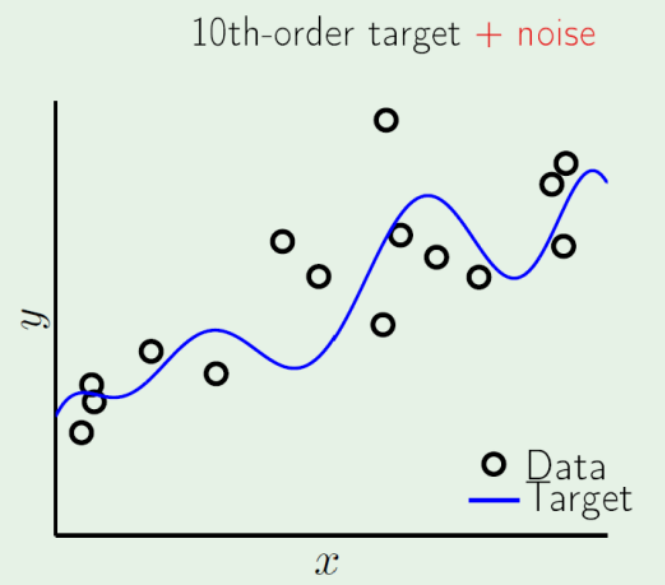
- In blue we show the true function generating the data, 2nd order polynomial
- In red we show the fitted function with zero in-sample-error. A 4th-order polynomial
- The sample have been overfitted !!
- Little noise in the data has mislead the learning

- 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.
  - $\mathbb{E}_D\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \sigma^2 + \mathbf{bias} + \mathbf{variance}$ (for noisy signals)

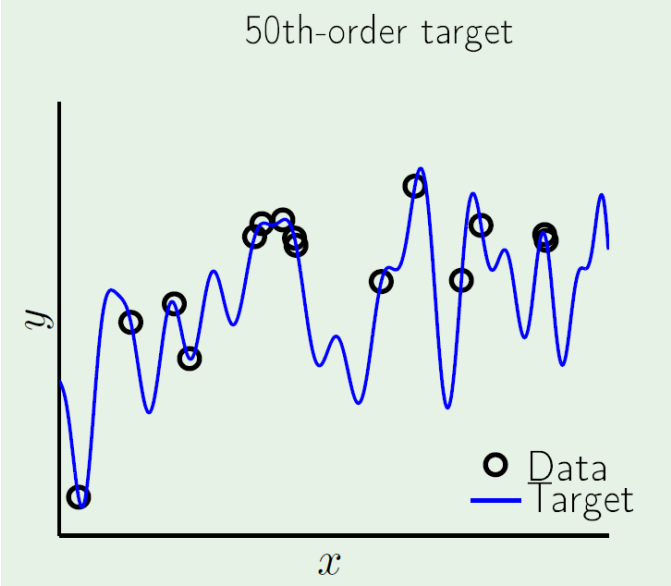
Overfitting: A case study

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

With added noise

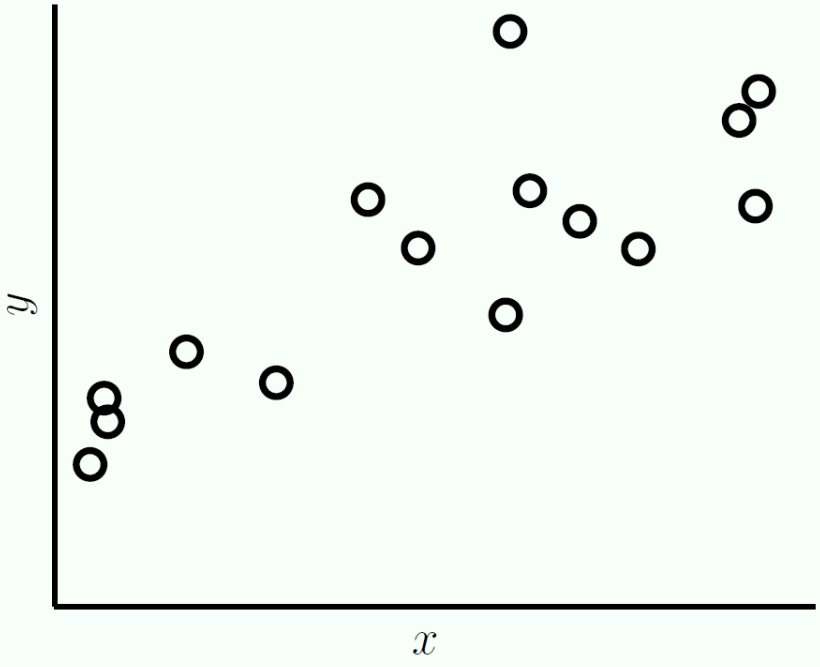


Noiseless

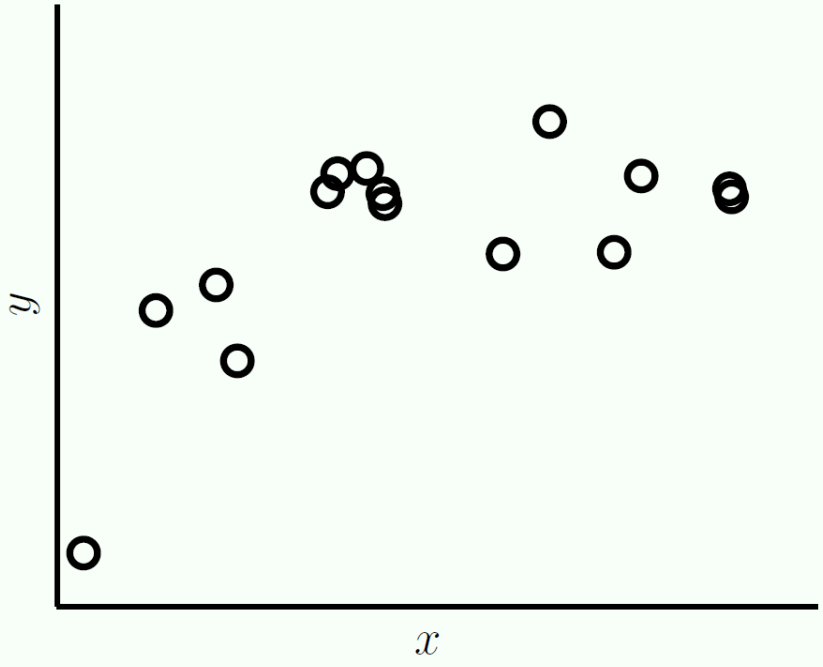


- Let's fit in both cases two polynomial: low and high order (2nd and 10th)
- Let analyze which of both produce lower out-of-sample error

Can the noise be distinguished?



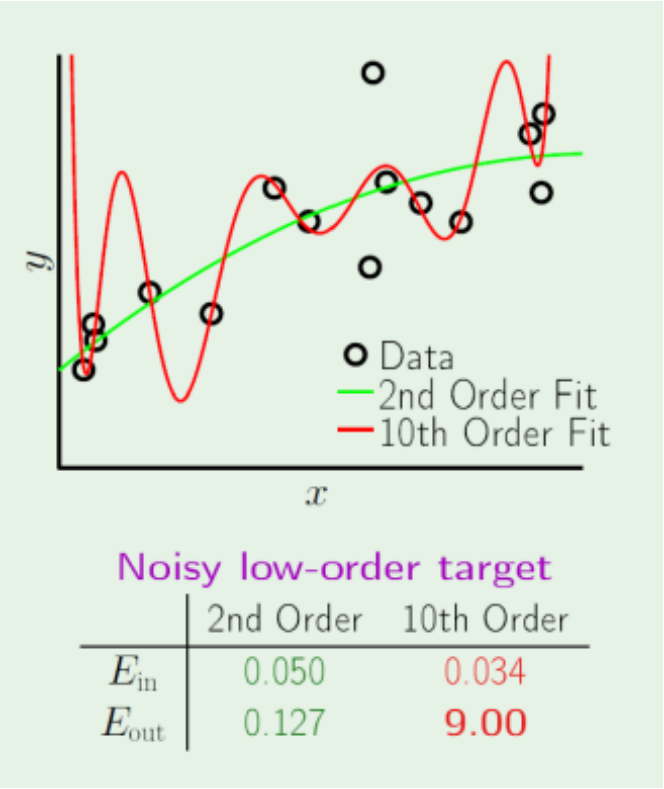
Simple f with noise.



Complex f with no noise.

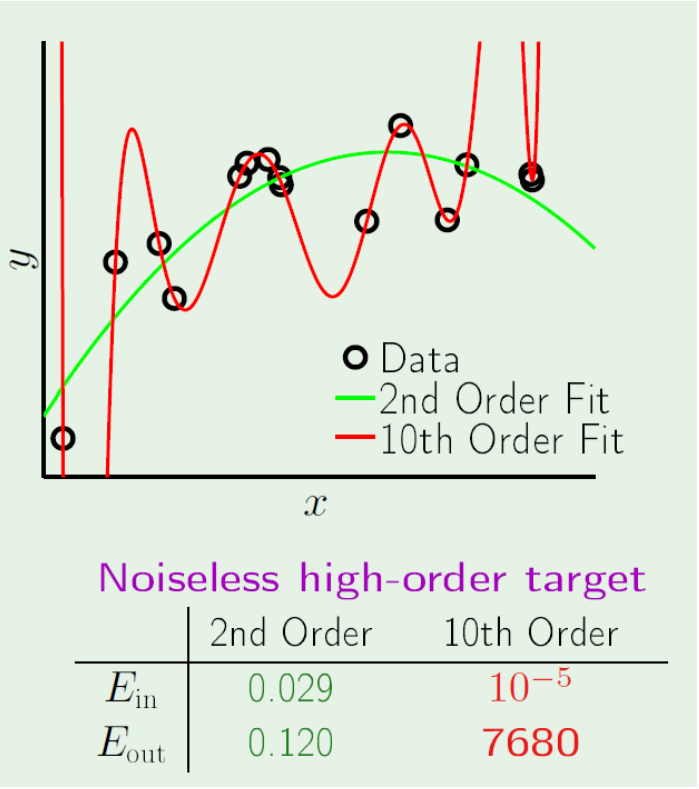
- The learning model should match the quality and quantity of the data NOT f

Overfitting: A case study



Noisy low-order target

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

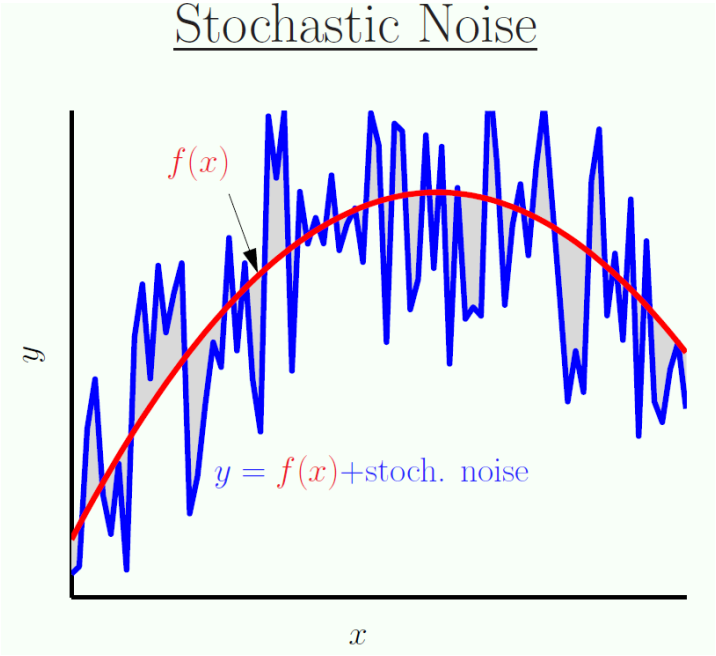


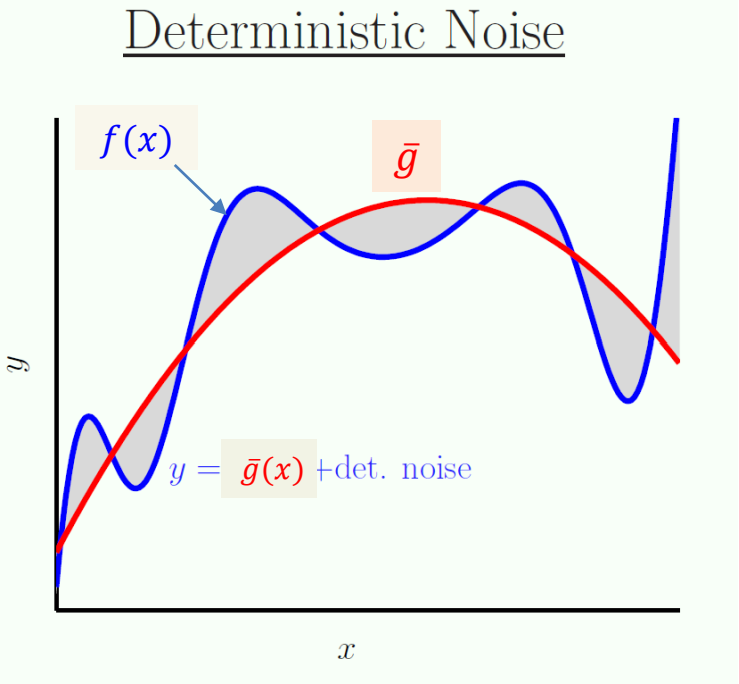
Noiseless high-order target

	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

Noise: what we cannot model





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

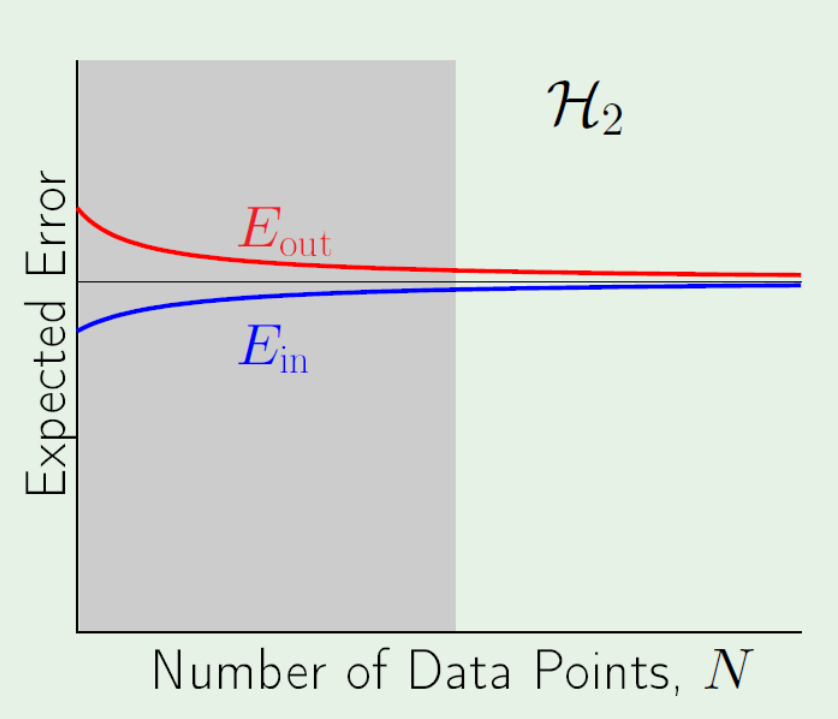
Deterministic noise: The part of the target function outside of the best fit $\bar{g}$

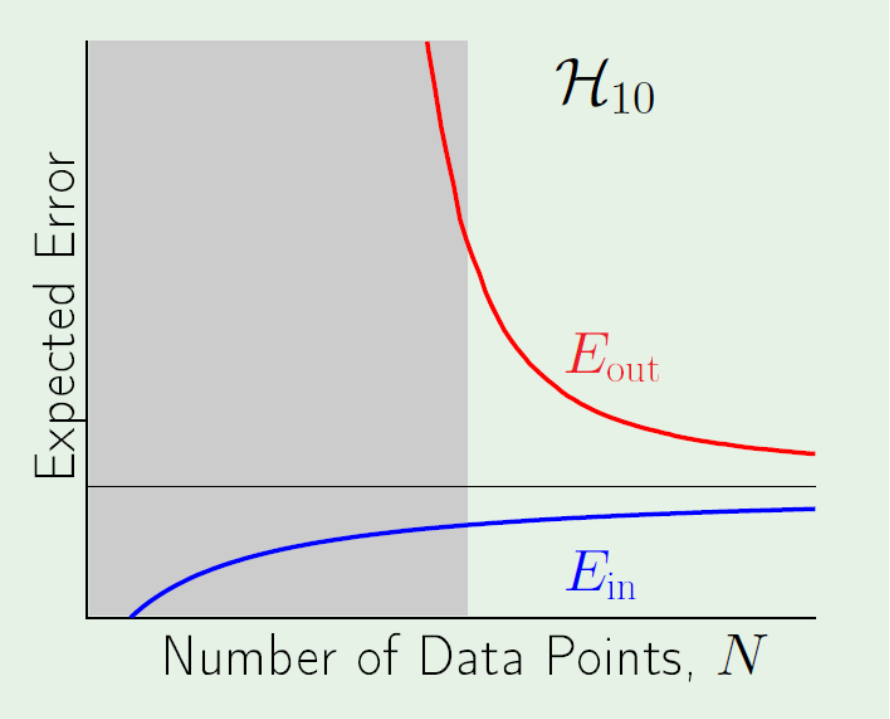
$$y = g_{\mathcal{D}}^{*}(x) + \text{noise} \qquad \text{noise} = \text{stoch. noise} + \text{det. noise}(\mathcal{H})$$

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

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

Learning curves: overfitting



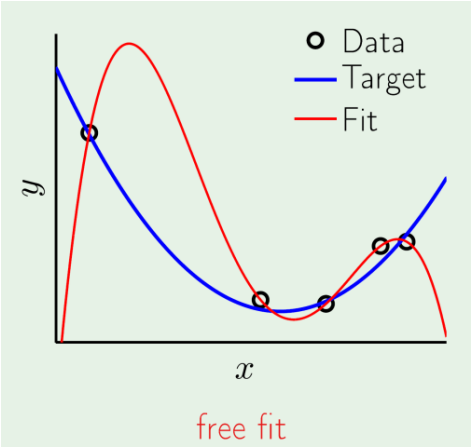


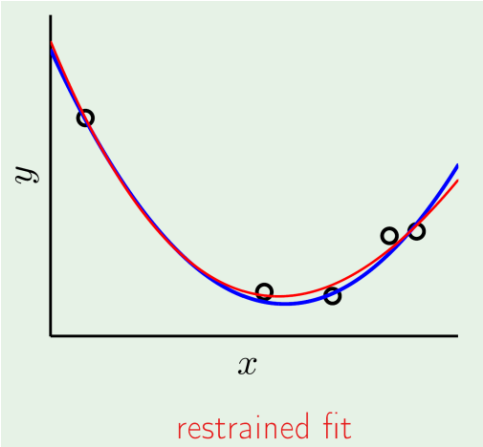
- The gray area shows the range of N values, where $\mathcal{H}_{10}$ has lower E_{in} and higher E_{out}: overfitting is present.
- The learning curves show typical behaviour of a simple and a complex model respectively.
- These pictures show the importance of the data size in the overfitting

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

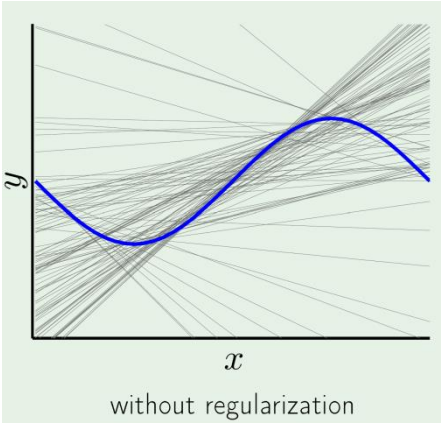
- Regularization is an heuristic approach although is in close connection with the optimization techniques

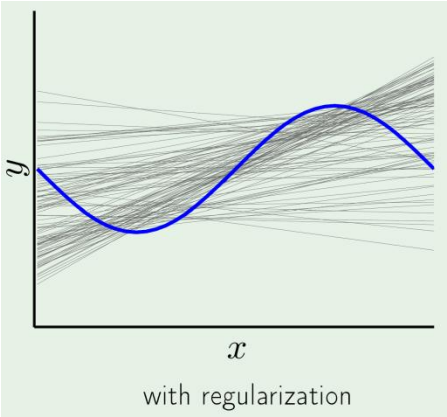
- According to the Approx.-Genera. tradeoff $E_{out}(g) \leq E_{in}(g) + \Omega(\mathcal{H})$, regularization minimizes the right hand of the inequality not only the in-sample error

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

Constraining the model helps

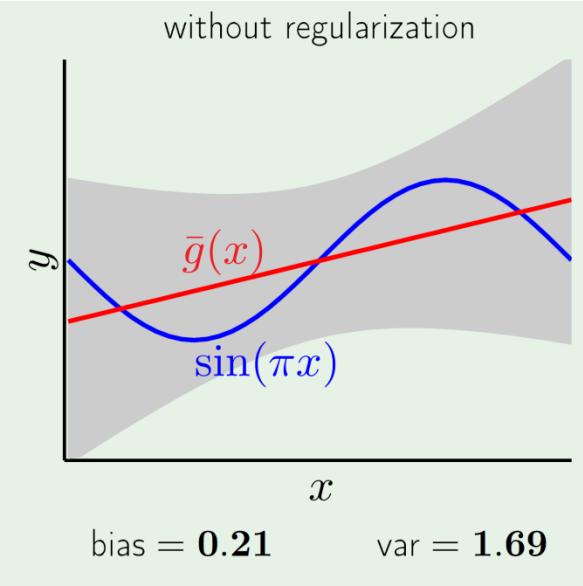
- The **weight decay** technique measures the complexity of a hypothesis h by the size of the coefficients used to represent h.

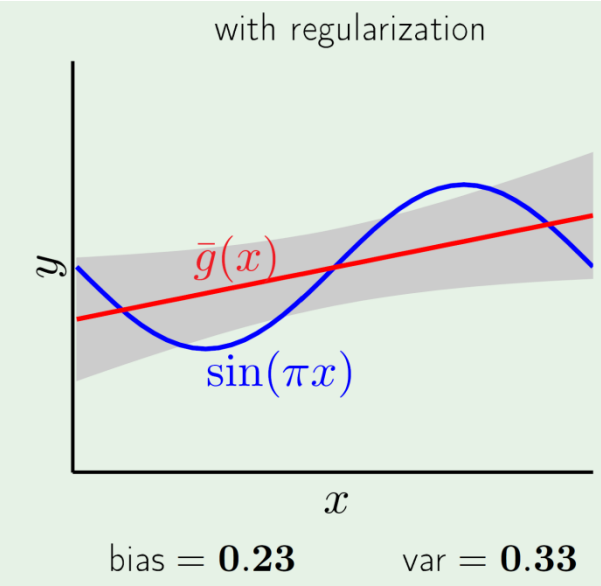




- The figure shows the result of applying weight decay to fit the target $f(x)=\sin(\pi x)$, $x \in [-1,1]$, using samples of N=2 (lines) , x is sampled uniformly in [-1,1]
- Without regularization shows a very high variability in the learning function depending on the sample x
- With regularization (constraining weights to be small) shows how the set of learning functions is much more stable

Constraining the model helps

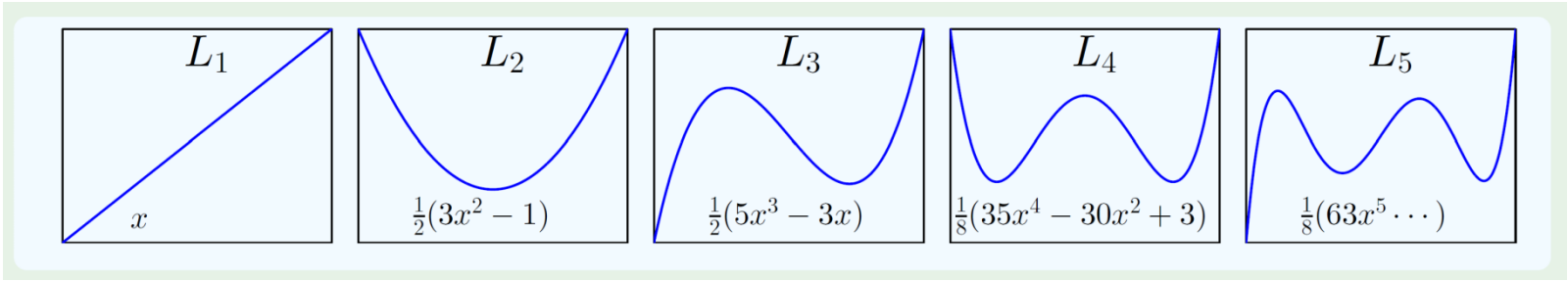




- Let analyze the learning using the Bias-Variace tradeoff
- Without regularization we observe a lower bias and higher variance
- With regularization we observe one light increased bias and a large decrease in variance
- In total the regularization provides a learned function with smaller out-of-sample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

Regularization: Some theory

- Consider a learning model where $\mathcal{H}_Q$ is the set of all polynomials in one variable x∈[-1, 1] until order Q-th.
- The task is to learn the best polynomial function (minimum out-of-sample error) fitting a given data set.



- We will use the Legendre orthogonal polynomial (see figure) in order to simply the derivation.
- The set of function $\{L_i, i=1,\ldots,Q\}$ define an orthogonal base for the rest of polynomials

$$z = \begin{bmatrix} 1 \\ L_1(x) \\ \vdots \\ L_Q(x) \end{bmatrix}, \qquad \mathcal{H}_Q = \left\{ h \mid h(x) = \mathbf{w}^T\mathbf{z} = \sum_{q=0}^{Q} w_q L_q(x) \right\}$$

Regularization: Some theory

- **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 $\mathbf{w}_{lin} = \underset{\mathbf{w}}{\operatorname{argmin}}\, E_{in}(\mathbf{w})$

How to restrict the values of the vector $\mathbf{w}_{lin}$ in order to get a better Bias-Variance tradeoff?

  - Hard constraints: imposes that some weights must be zero
  - 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

Regularization: solving the problem(SRM)

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

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

the learning algorithm chooses the best solution given the total budget *C*.

- Let $\mathrm{H}(C) = \{h \big| h(x) = \mathbf{w}^{\mathrm{T}}\mathbf{z},\ \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C\}$, clearly the *C* value defines a constraint on the class of hypothesis (SRM):
  - If $C_1 < C_2$ then $\mathcal{H}(C_1) \subset \mathcal{H}(C_2)$ and so $d_{VC}(\mathcal{H}(C_1)) < d_{VC}(\mathcal{H}(C_2))$, we expect better generalization error with $\mathcal{H}(C_1)$

Using Lagrange Multipliers

$$\mathbf{w}_{aug} = \operatorname*{argmin}_{\mathbf{w}} \{E_{in}(\mathbf{w}) + \beta(\mathbf{w}^T\mathbf{w} - C)\} = \operatorname*{argmin}_{\mathbf{w}} \{E_{in}(\mathbf{w}) + \lambda_C\, \mathbf{w}^T\mathbf{w}\}$$

- Let's define the augmented error for each hypotesis $\mathbf{w}$:

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

- The λ parameter defines the intensity of the regularization
- $\Omega(\mathbf{w}) = \mathbf{w}^T\mathbf{w}$ defines a complexity measure for each hipothesis

Regularized linear model: 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 $\nabla_{\mathbf{w}} E_{aug}(\mathbf{w}) = \nabla_{\mathbf{w}} (E_{in}(\mathbf{w}) + \lambda \mathbf{w}\mathbf{w}^T) = 0$

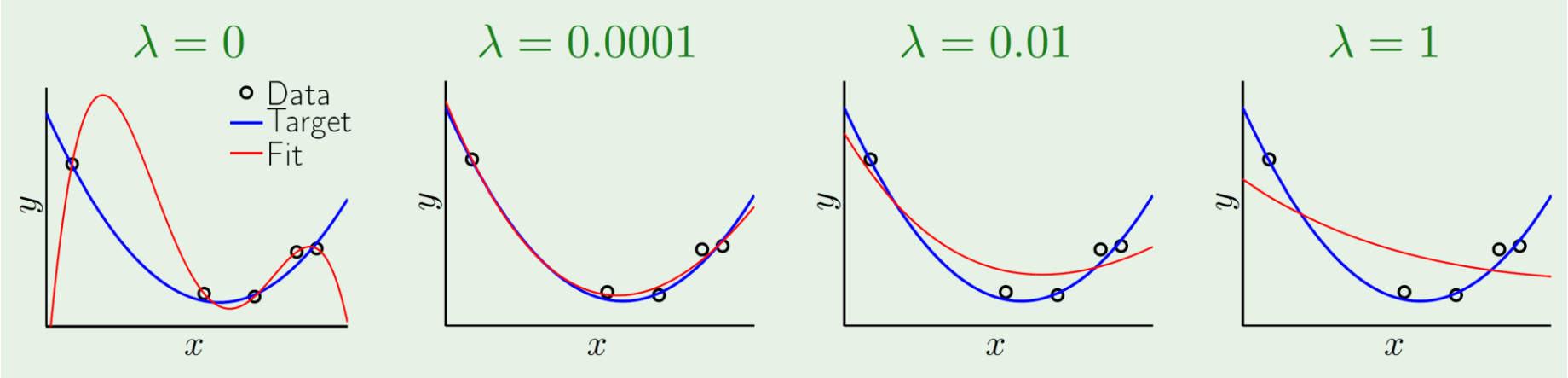
$$\nabla_{\mathbf{w}} E_{aug} = 2\mathrm{Z}^T(\mathrm{Z}\mathbf{w} - \boldsymbol{y}) + \lambda \mathbf{w}^T = 0 \quad \Rightarrow \quad \mathbf{w}_{reg} = (\mathrm{Z}^{\mathrm{T}}\mathrm{Z} + \lambda \mathrm{I})^{-1}\mathrm{Z}^{\mathrm{T}}\boldsymbol{y}$$

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

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

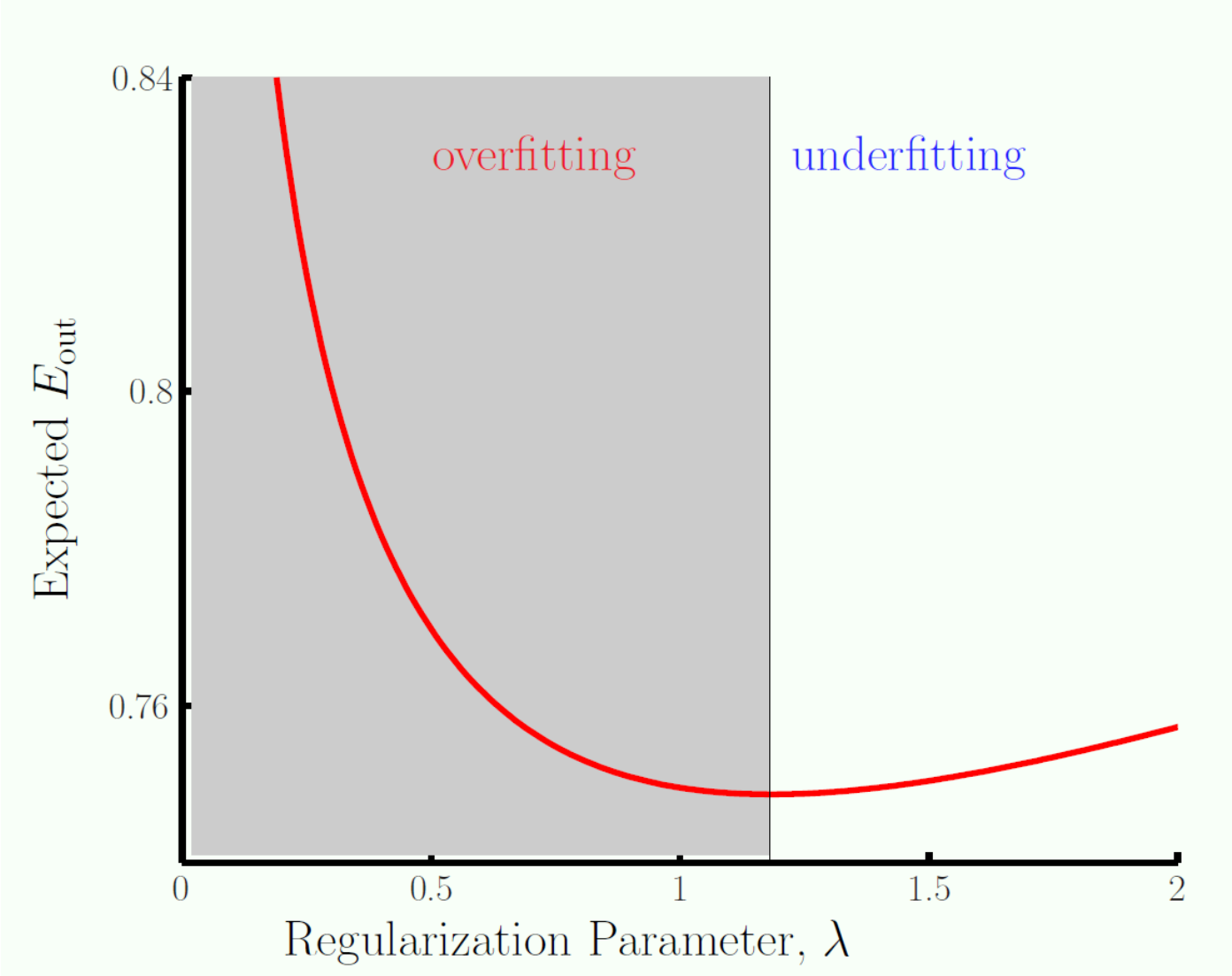
- The matrix hat $\mathrm{H}(\lambda)$ plays a relevant role in defining the efective complexity of the model
  - λ=0, H is the hat-matrix of the linear regression
  - The vector of in-sample errors is : $\boldsymbol{y} - \hat{\boldsymbol{y}} = \left(\mathrm{I} - H(\lambda)\right)\boldsymbol{y}$
  -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

Regularization: Linear models + w.d.

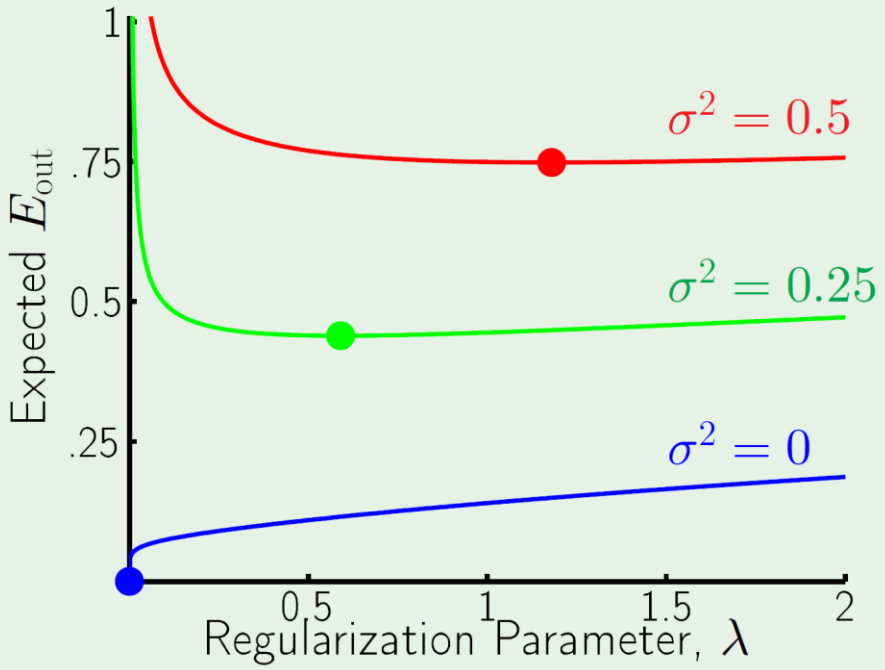


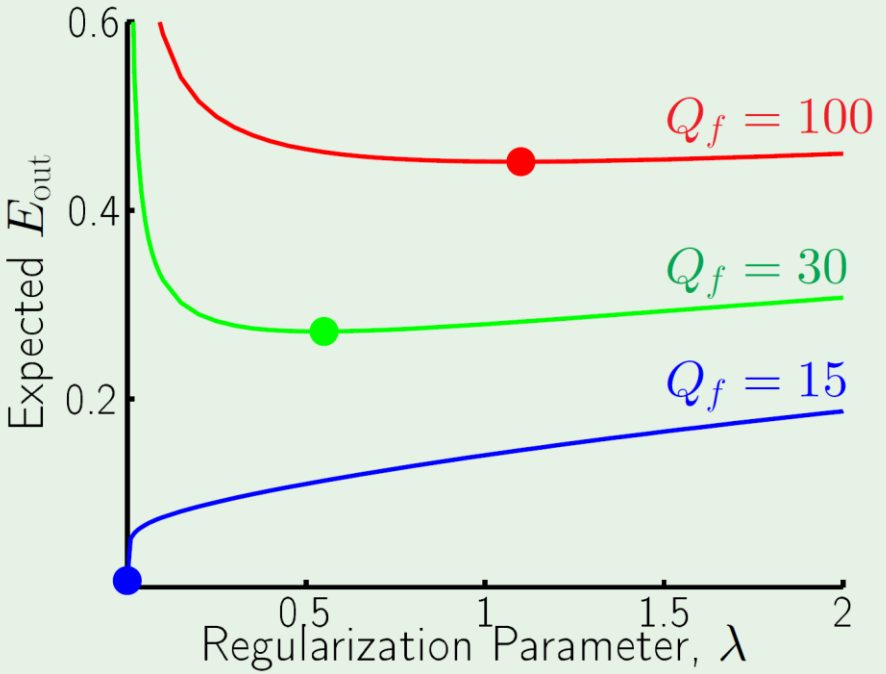
- The figure shows the result of applying different amount of regularization to the same example using weight decay
- It can be seen that non-regularization or too much regularization increases the adjustment error. In the first case due to the variance in the second case due to the bias.

Overfitting & Underfitting



Regularization and noise





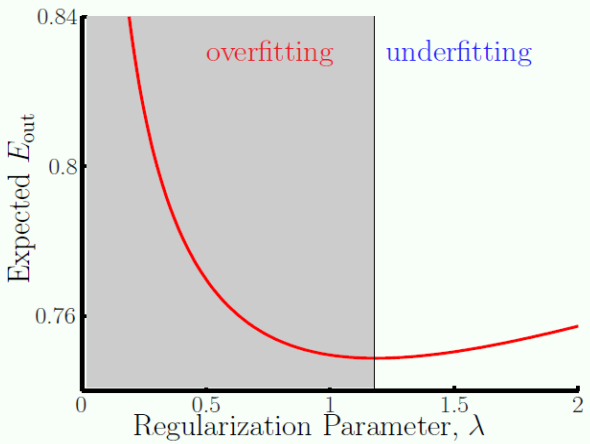
Stochastic noise

Deterministic noise

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

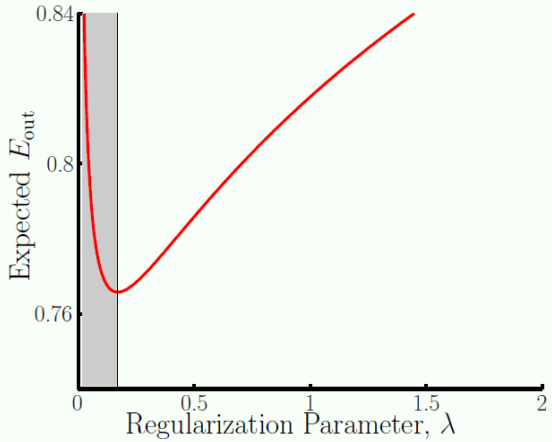
Variations on Weight Decay

Uniform Weight Decay



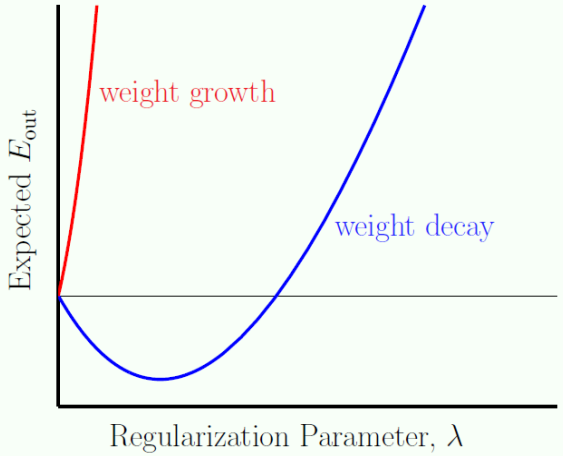
$$\sum_{q=0}^{Q} w_q^2$$

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.**

How Does Regularization Work?

- Stochastic noise −→ nothing you can do about that.
- Good features −→ helps to reduce deterministic noise.
- Regularization:
  - Helps to combat what noise remains, especially when N is small.
  - Typical modus operandi: sacrifice a little **bias** for a huge improvement in **var**.
  - VC angle: you are using a smaller $\mathcal{H}$ without sacrificing too much E_{in}

Regularization and the VC dimension

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

The class $\mathcal{H}$ it is fixed, hence its VC-dimension remains

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

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

The regularization process goes up the fitting error from E_{in} to E_{aug}

$$E_{out}(h) \leq E_{aug}(h) + \Omega(\mathcal{H}^*)$$

$\mathcal{H}^* \subset \mathcal{H}$ is the "effective" class of function due to regularization

When $$\frac{\lambda_C}{N}\Omega(h) \leq \Omega(\mathcal{H}) - \Omega(\mathcal{H}^*)$$

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