4.2. Generalization and Regularization

Lecture based on "Dive into Deep Learning" **http://D2L.AI** (Zhang et al., 2020)

and C.M. Bishop, Pattern Recognition and Machine Learning

Prof. Dr. Christoph Lippert

Digital Health & Machine Learning

Training and generalization error

- *Training error*: The error calculated on the training data set
- *Generalization error*: The expectation of our model's error on additional data points drawn from the same underlying data distribution.
- Problem: *we can never calculate the generalization error exactly*
- $\rightarrow$ apply model to an independent test set, withheld from training

Example (Coin Toss)

- Dataset $\{0, 1, 1, 1, 0, 1\} \rightarrow$ Predict the *majority class* (here: 1) with error of only $\frac{1}{3}$.
- With more samples it would go to $\frac{1}{2}$.

Generalization is the fundamental problem in machine learning.

Drawing training and validation samples

In **supervised learning**, we usually assume that both the training data and the test data are drawn *independently* from *identical* distributions (i.i.d.).

- Sampling process has no memory
- 2nd and 3rd samples are no more correlated than 2nd and nth sample

Example

Covid-19 mortality risk predictor on data collected from patients from Charité Berlin, and apply it on patients from Mt. Sinai.

Example

Face recognition trained only on *students* and then applied in an *elderly home*

Goal: Find a function that fits training set well, but generalizes well on unseen data

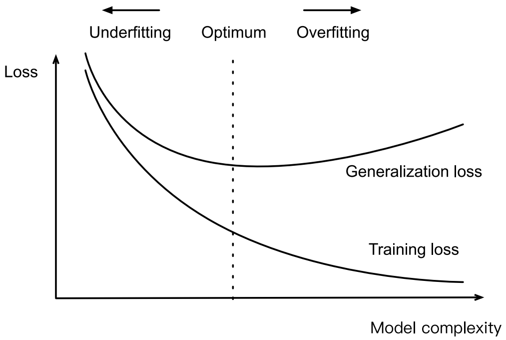
Underfitting or overfitting

Aim: Model, where training error and validation error are both substantial but there is a little gap between them.

- Model too simple: unable to reduce the training error $\rightarrow$ **Underfitting**
- Model too complex: Training error «validation error $\rightarrow$ **Overfitting**
- Over- or underfitting depends on size of training data and model complexity

Care more about validation error and find tradeoff

Underfitting or overfitting



Example (Polynomial regression)

Given training data consisting of a single feature x and a corresponding real-valued label y

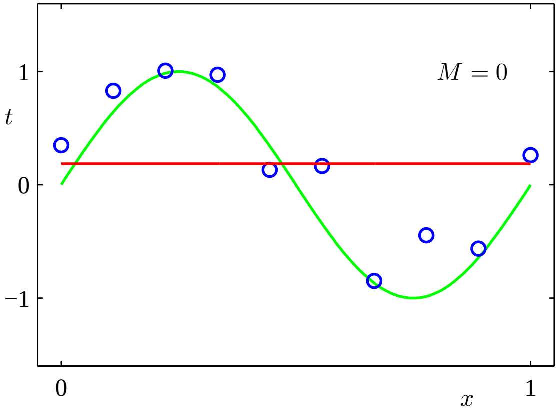
- find the polynomial of degree M

$$\begin{aligned}\hat{y} &= \sum_{j=0}^{M} x^j w_j \\ &= \sum_{j=0}^{M} \phi_j(x) w_j \\ &= \mathbf{w}^\top \mathbf{x} + b\end{aligned}$$

- Higher-order polynomial: More model parameters, lower training error

Polynomial curve fitting

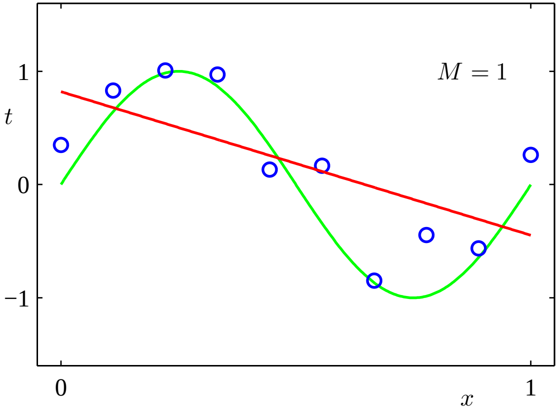
The degree M of the polynomial is crucial.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

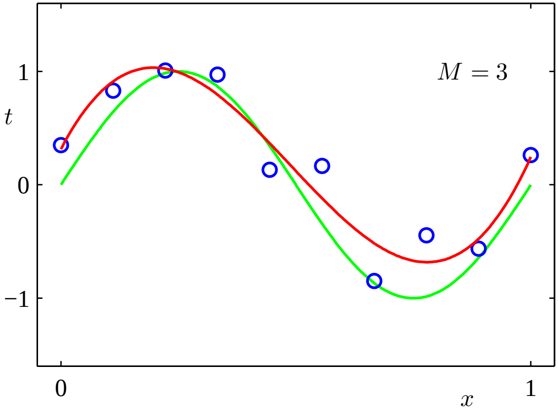
The degree M of the polynomial is crucial.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

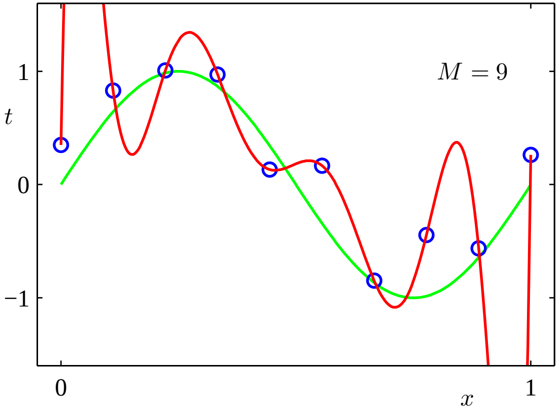
The degree M of the polynomial is crucial.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Polynomial curve fitting

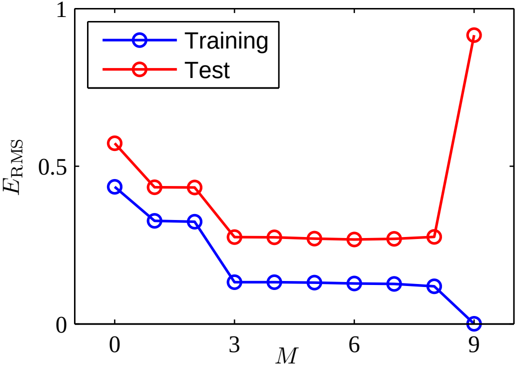
The degree M of the polynomial is crucial.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Train vs. Test error

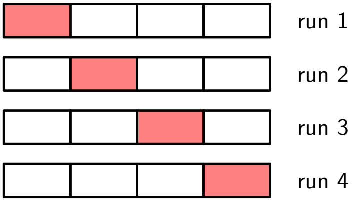
- Split sample in training and test set.
- Choose M based on test error.



(C.M. Bishop, Pattern Recognition and Machine Learning)

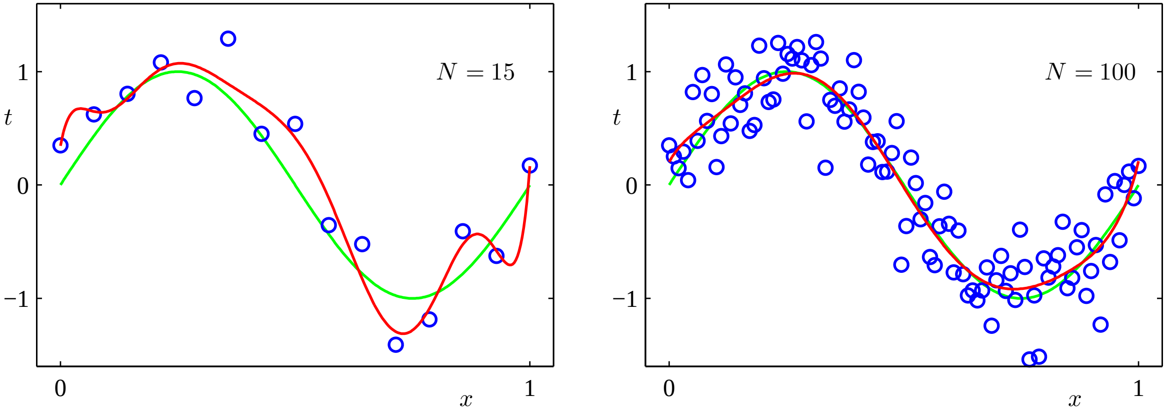
K-fold Cross validation:

- randomly assign samples $(\mathbf{x}^{(i)}, y^{(i)}$ to K sets of equal size
- for each set $s \in S$ (e.g. $K = 4$ sets) and $m \in [1, \ldots, M]$:
  - train model on $K - 1$ remaining sets
  - predict on s and compute loss.
- compute average MSE for degree m.
- pick m with lowest loss.



(C.M. Bishop, Pattern Recognition and Machine Learning)

Get more data



Others ways to avoid overfitting and fit complex model for limited number of observations?

Regularization of weights

Regularization

Regularize the regression weights.

Loss function:
$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\mathbf{w}^\top\mathbf{x}^{(i)} + b - y^{(i)})^2}_{l(\mathbf{w},b)} \quad + \quad \underbrace{\frac{\lambda}{2}\sum_{j=1}^{d} w_j^2}_{l_2\text{-norm regularizer}}$$

where $\sqrt{\sum_{j=1}^{d} w_j^2}$ is the l_2-norm of $\mathbf{w}$.

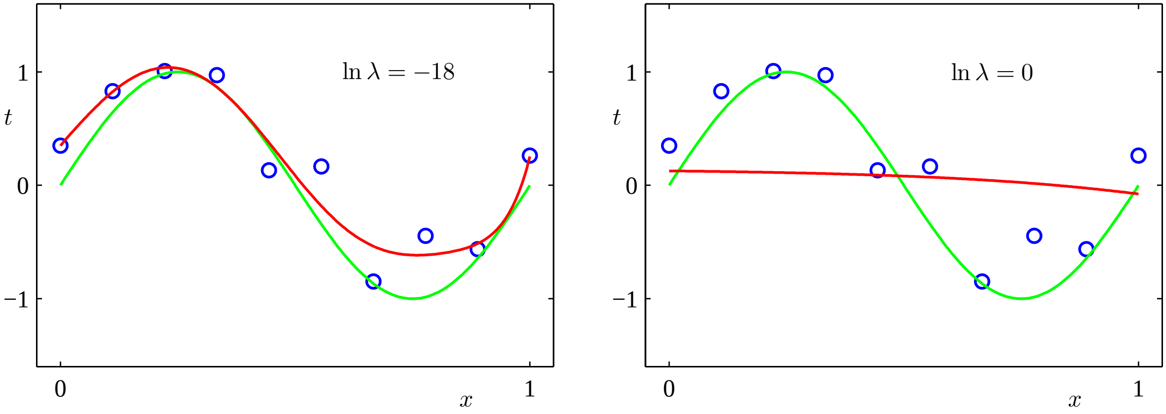
What effect does this have? How will the learned weights be different?

- Penalizes large weights.
- Reduces the complexity of the function that associates $\mathbf{x}$ with $\mathbf{y}$, i.e. learn parsimonious model.
- Also known as **shrinkage** or **weight decay**.

Regularization

Loss function:

$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\mathbf{w}^\top\mathbf{x}^{(i)}+b-y^{(i)})^2}_{l(\mathbf{w},b)} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{d}w_j^2}_{l_2\text{-norm regularizer}}$$



The stochastic gradient descent updates for L2-regularised regression are as follows:

$$\mathbf{w} \leftarrow \left(1 - \frac{\eta\lambda}{|\mathcal{B}|}\right)\mathbf{w} - \frac{\eta}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \mathbf{x}^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right),$$

Weight decay

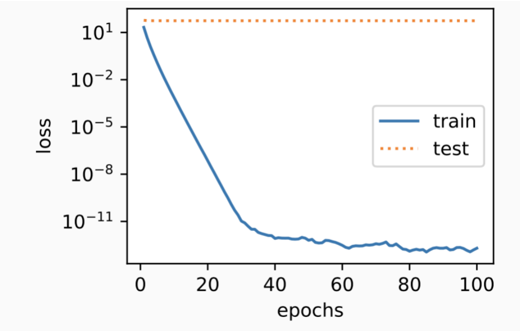


Figure: Training without regularization

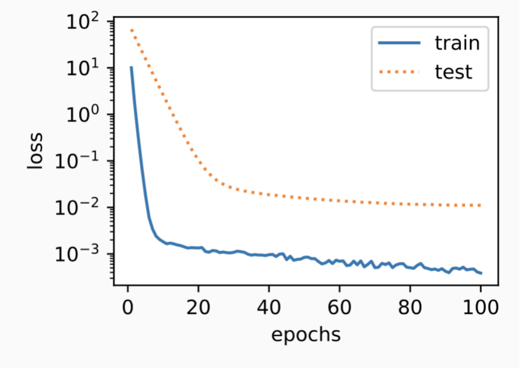


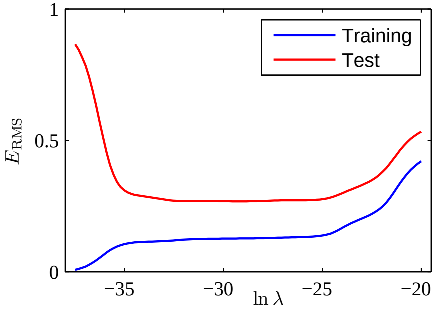
Figure: Training with L2-regularization

Regularization

$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)})^2}_{l(\mathbf{w},b)} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{d} w_j^2}_{l_2\text{-norm regularizer}}$$

Question: How to chose an optimal λ?

Answer: Look at the test error!

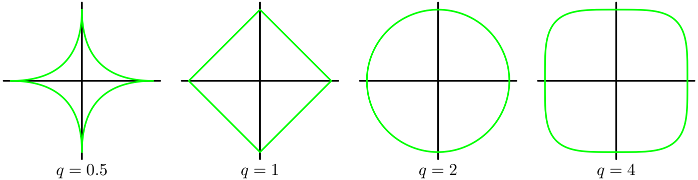


(C.M. Bishop, Pattern Recognition and Machine Learning)

Regularization

A more general regularization:

$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\mathbf{w}^\top\mathbf{x}^{(i)}+b-y^{(i)})^2}_{l(\mathbf{w},b)}+\underbrace{\frac{\lambda}{2}\sum_{j=1}^{d}|w_j|^q}_{l_q\text{-norm regularizer}}$$



(C.M. Bishop, Pattern Recognition and Machine Learning)