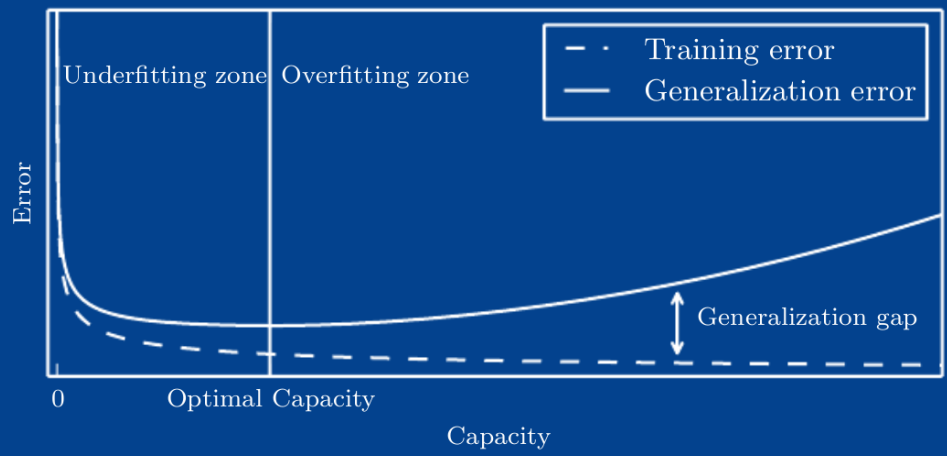
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

LESSON 7: Koncepts II

capacity, under- and overfit, generalization

CARSTEN EIE FRIGAARD

AUTUMN 2019



Supergruppe diskussion

Diskussion ml. Grp A og B:

- ca. 15 min: Grp A genfortæller første halvdel af § 2
  - ”Look at the Big Picture”,
  - ”Get the Data”,

    eksklusiv ”Create the Workspace” og ”Download the Data”,
  - ”Discover and Visualize the Data to Gain Insights”,
  - ”Prepare the Data for Machine Learning Algorithms”,
  - ”Select and Train a Model”,
- ca 15 min: herefter byttes og Grp B genfortæller..
  - ”Fine-Tune Your Model”,
  - ”Launch, Monitor, and Maintain Your System”,
  - ”Try It Out!”.

Inddrag og relater til jeres eget slut-projekt,
Inddrag The Map for Supervised learning
Session I: 8:30..9:15
Session II: 9:15..10:00

The Map

model selection

Pipelines

Brief intro to Scikit-learn pipelines..

Python code from `capacity_under_overfitting.ipynb`

```
1  from sklearn.pipeline import Pipeline
2  from sklearn.preprocessing import PolynomialFeatures
3  from sklearn.linear_model import LinearRegression
4  from sklearn.model_selection import cross_val_score
5
6  [..]
7
8  polynomial_features = PolynomialFeatures(degree=degrees[i], [..]
9  linear_regression = LinearRegression()
10
11 pipeline = Pipeline([
12         ("polynomial_features", polynomial_features),
13         ("linear_regression", linear_regression)
14     ])
15 pipeline.fit(X[:, np.newaxis], y)
16
17 scores = cross_val_score(pipeline, X[:, np.newaxis], y, scoring=
       "neg_mean_squared_error", cv=10)
```

Model capacity

Exercise: `capacity_under_overfitting.ipynb`

Dummy and Paradox classifier:
capacity fixed ~ 0, cannot generalize at all!

Linear regression for a polynomial model:
capacity $\sim$ degree of the polynomial, x^n

Neural Network model:
capacity $\propto$ number of neurons/layers

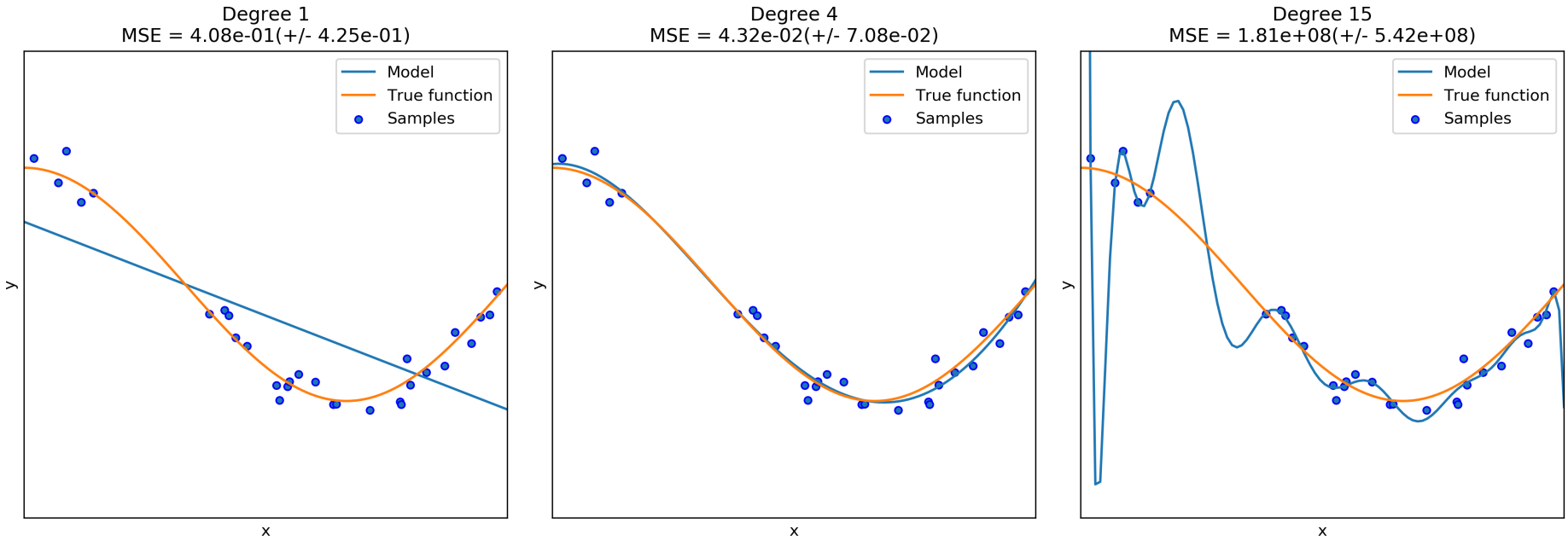
$\Rightarrow$ **Capacity** can be hard to express as a quantity for some models, but you need to choose..

$\Longrightarrow$ how to choose the **optimal** capacity?

Under- and overfitting

Exercise: `capacity_under_overfitting.ipynb`

Polynomial linear reg. fit for underlying model: `cos(x)`



- underfitting: capacity of model too low,
- overfitting: capacity to high.

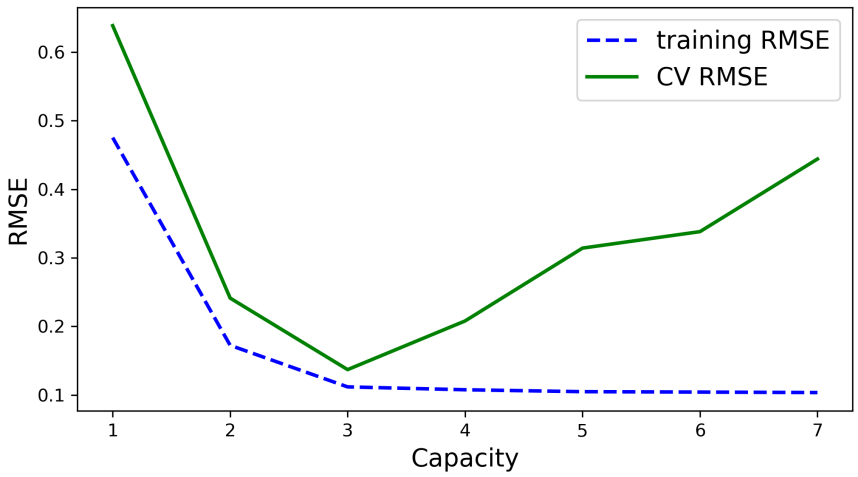
$\Longrightarrow$ how to choose the **optimal** capacity?

Generalization Error

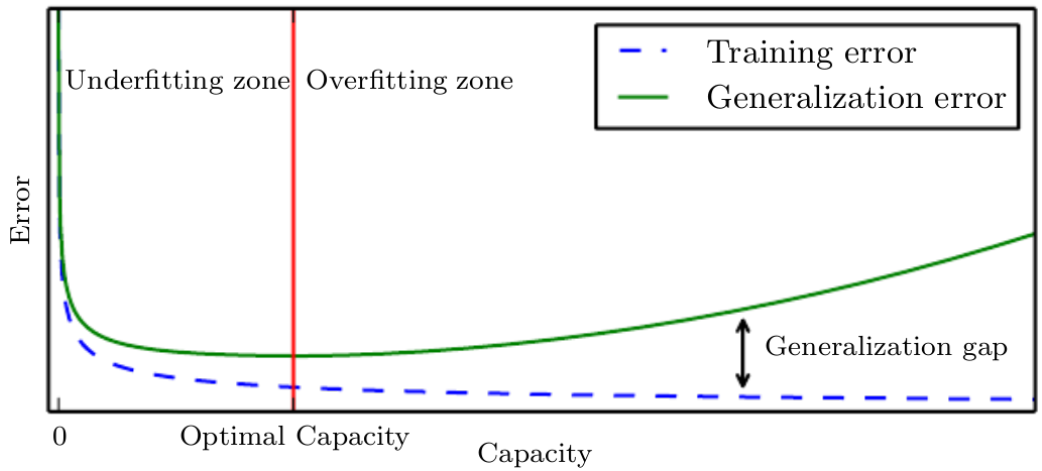
Exercise: `generalization_error.ipynb`

RMSE-capacity plot for lin. reg. with polynomial features

(capacity $\sim$ degree of poly)



(Figure 5.3 from [DL])



Inspecting the plots from the exercise (`.ipynb`) and [DL], extracting the concepts:

- training/generalization error,
- generalization gab,
- underfit/overfit zone,
- optimal capacity (best-model, early stop),
- (and the two axes: x/capacity, y/error.)

Generalization Error

Exercise: `generalization_error.ipynb`

NOTE: three methods/plots:

i) via **learning curves** as in [HOML],
ii) via an **error-capacity** plot as in [GITHOLM] and[DL],
ii) via an **error-epoch** plot as in [GITHOML].