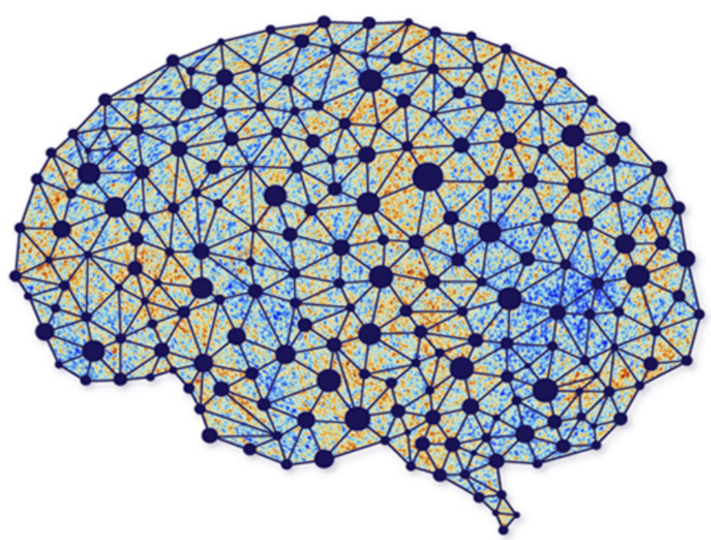
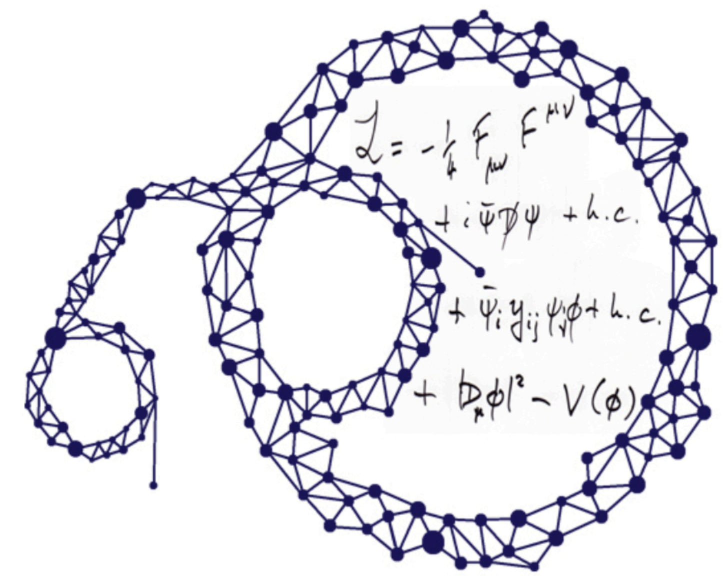
Physics 361 - Machine Learning in Physics

Lecture 5 – Basics of Machine Learning

Feb. 6th 2024



AI
∩
Universe



Moritz Münchmeyer

Unit 2: Machine Learning Basics

2.1 Machine Learning concepts using the example of Linear Regression (cont.)

Linear regression (cont'd)

Very simple ML model:

$$\hat{\vec{y}} = \vec{w}^T \vec{x} \qquad \vec{x}: \text{input} \quad \vec{y}: \text{output}$$

training data:

$$X^{train} = \overbrace{\left(\qquad \right)}^{\text{features}} \Big\} \text{examples}$$

$$\vec{y}^{\,train} = \text{desired output} = \text{label}$$

Loss function:

$$MSE = \frac{1}{m} \| \underbrace{\hat{\vec{y}}}_{\text{estimate}} - \underbrace{\vec{y}}_{\text{training label}} \|_2^2$$

Can be minimized w.r.t. to $\vec{w}$:

$$\vec{w} = \left(X^{train\,T} X^{train} \right)^{-1} X^{train\,T} \vec{y}^{\,train}$$

Last slide: y was a 1-dimensional output.

We can also have multidimensional regression targets:

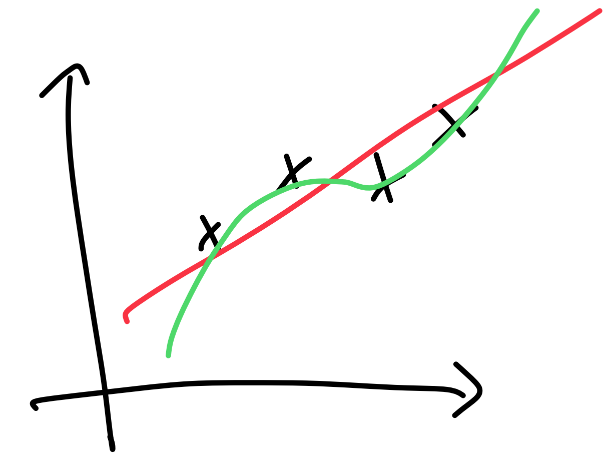
$$\vec{y} = M\vec{x} + b$$

vector output

(before we had $y = \vec{w}^T \vec{x}$)

- These linear transformations are called Linear Layers in machine learning.
- Side note: While linear models are analytically solvable (for invertible M), for very high dimensional problems it is more efficient to solve them by optimization.

Polynomial regression



- In polynomial regression we fit a higher order polynomial.

  E.g. 2nd order pol. $\hat{y} = b + w_1 x + w_2 x^2$

  General: $\hat{y} = b + \sum_{i=1}^{N} w_i x^i$

- This model can be solved for $\vec{w}$ with the same equations as linear regression because it is still linear in the parameters w_i.

Capacity, Overfitting, Underfitting

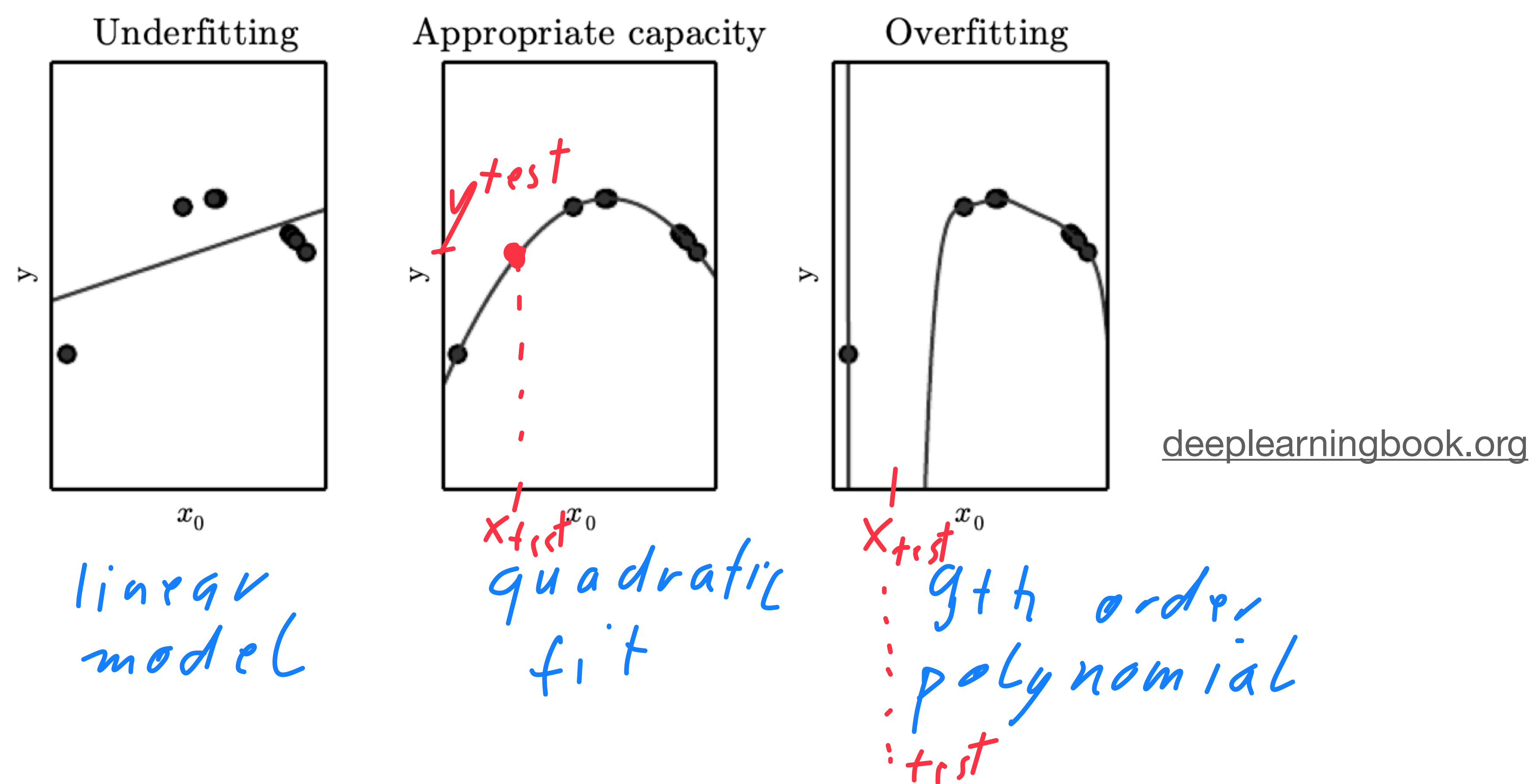
- How well does a model trained/fitted to the training data work on novel data.
  $\Rightarrow$ Generalización error

- We usually assume that training data and test data are drawn from the same distribution. „i.i.d." data : independent identically distributed

- Generalización is closely related to overfitting and underfitting.

  underfit : training error is large
  overfit : training error is small but test error is large.

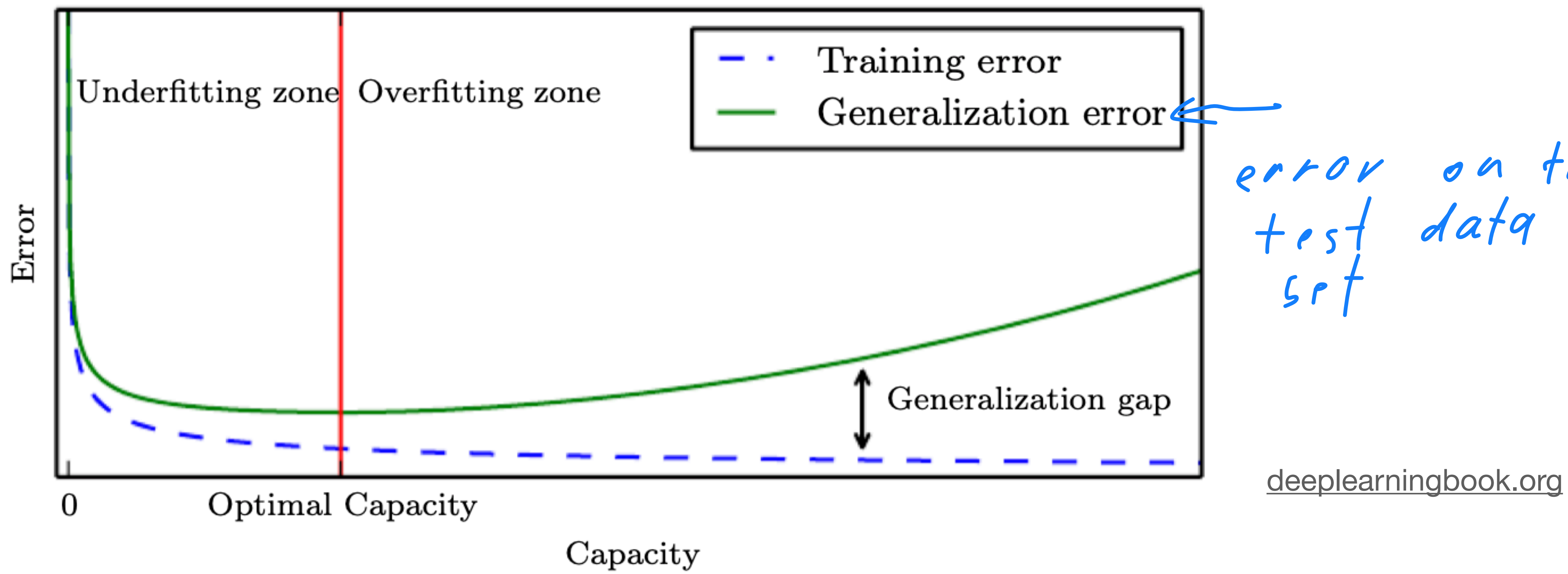
Example of polynomial regression:



- The space of functions that the model can represent is called the capacity.
- Occam's razor: chose the simplest „that fits"
- Lower capacity → tends to generalize better
  Higher capacity → reduces training error.

⇒ Need to optimize the capacity or regularize.

Typical behavior of error vs capacity
(after training out the model)



error on the test data set

We could e.g train models with various numbers of neural network layers.
Usually this is not the main approach.
Instead we use „regularization" to avoid overfitting.

Regularization

- Idea: Keep the model capacity fix, but encourage simpler solutions.

- Common method: weight decay

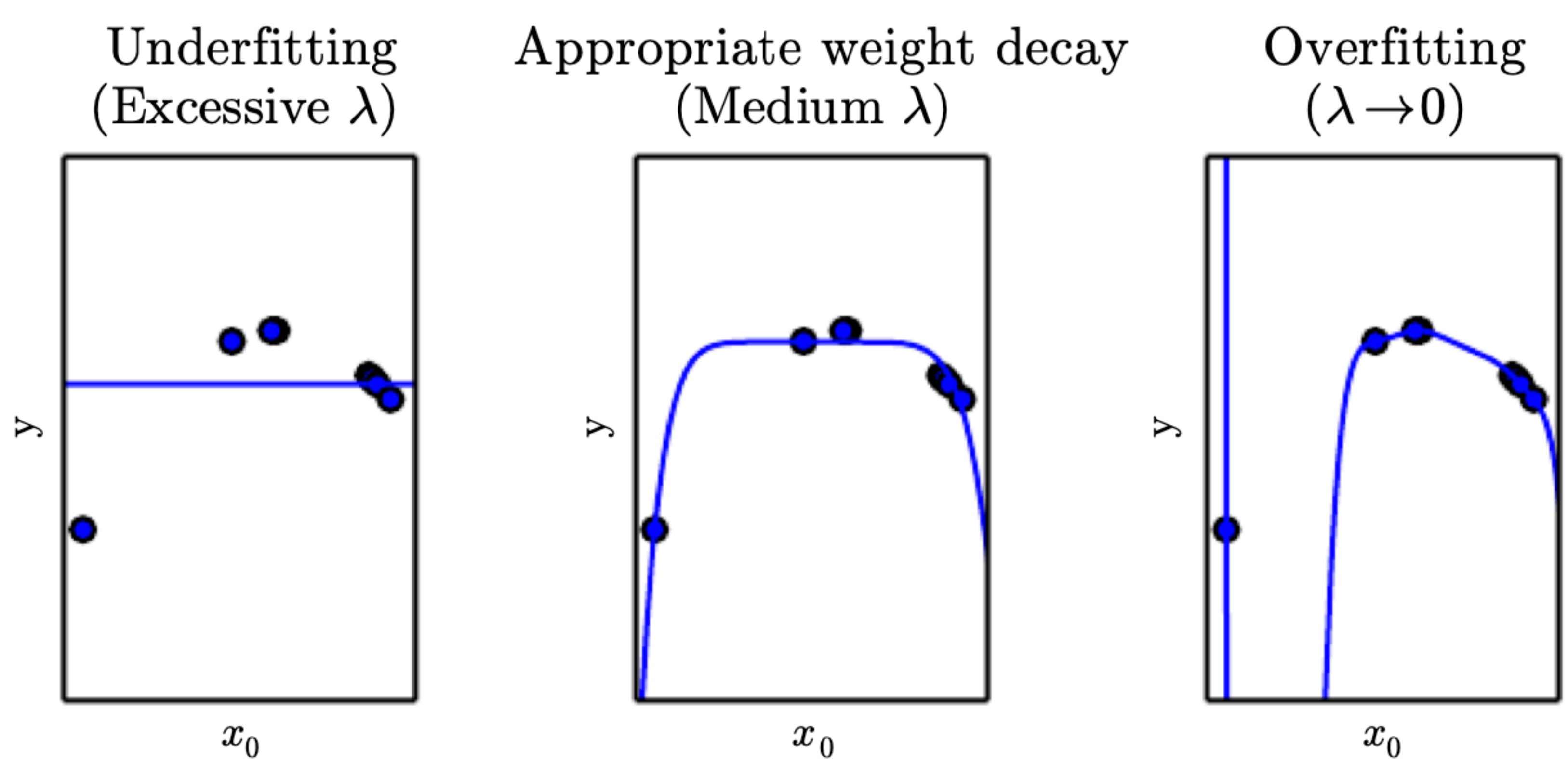
New loss to minimize $\quad J(\vec{w}) = MSE_{train} + \lambda \underbrace{\vec{w}^T \vec{w}}$

parameter to chose „lambda"

L-2 norm

L-2 regularization

This discourages large weights



deeplearningbook.org

L-2 regularization is one of several popular methods of regularization

Regularization : any method meant to prevent overfitting without changing the model capacity.

Two other popular methods which we discuss later:

- drop-out
- early stopping.

Hyperparameters

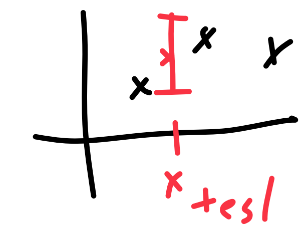
- Hyperparameters are parameters that are NOT part of the fitting/optimization procedure.

- Examples:
  - model architecture
    e.g. - maximum order of polynomials
    - depth of a neural network
  - Learning parameters
    e.g. learning rate
  - Regularization parameters e.g. λ

- To chose the best hyperparameters one often splits off a validation data set from the training data (~20% thereof).

Relation of MSE and maximum Likelihood

- In Linear regression we were Learning y from x. Instead now we want to Learn $p(y|x)$.
  $\Rightarrow$ we get an error bar.



- If we assume that the error is Gaussian we want to Learn: $p(y|x) = \mathcal{N}(y;\ \underbrace{\hat{y}(\vec{x}, \vec{w})}_{\text{mean of prediction}},\ \underbrace{\sigma^2}_{\text{width of prediction error.}})$

- The Loss is now the Likelihood of the training data:

plug in

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p(y^{(i)} | x^{(i)}; \theta)$$

model parameters, here $\vec{\theta} = \vec{w}$ in Linear regression.

MSE Loss

$$= -m \log \sigma - \frac{m}{2} \log(2\pi) - \boxed{\sum_{i=1}^{N} \frac{|\hat{y}^{(i)}_{(\theta)} - y^{(i)}|^2}{\sigma^2}}$$

- We want to maximize this with respect to θ.

Maximize $\mathcal{L}$ is the same as minimizing MSE.

Course logistics

- **Reading for this lecture:**
  - **For example:** Deeplearningbook.org 5.

- **Problem set**: First problem set end of this week.