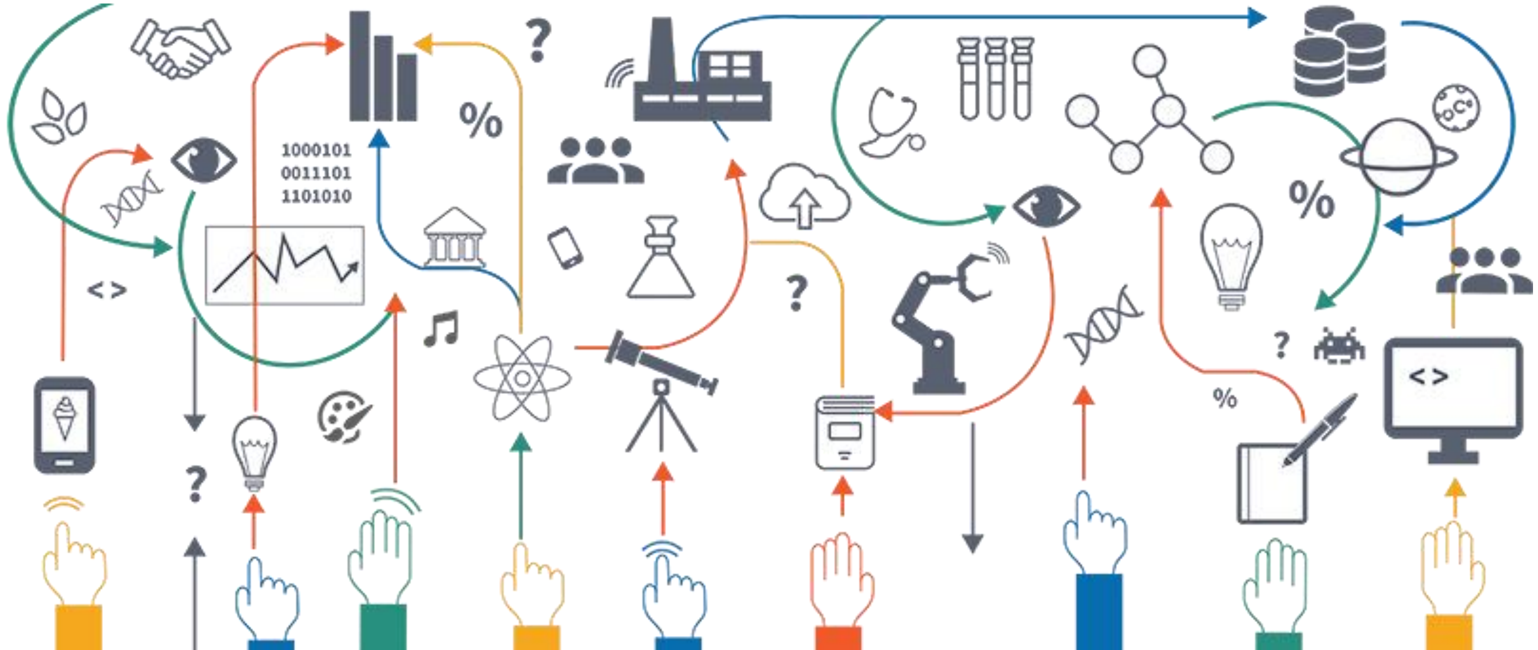
6 – Generalization bounds

Mathematics of Data Science

Lecturer: Dominik Dold



University of Vienna, WiSe 2025

Master’s programme in Data Science

Motivation

In the lecture block on *Function Approximation and Supervised Learning*, we identified the following inequality for the risk:

$$R(\hat{h}) \leq \sup_{h\in\mathcal{H}} \left| R(h) - \hat{R}(h) \right| + \inf_{g\in\mathcal{H}} \hat{R}(g)$$

The second term is the **interpolation error**. In the *Deep Neural Networks* block, we introduced the concept of **affine pieces** to bound it!

But how can we bound the first term: **the generalization error**?

Content

- Reminder: Hoeffding’s inequality
- PAC learning
- Covering numbers
- Overfitting in the under-and overparametrized regime
- Appendix: VC dimension

Return of the Hoeffding's inequality

In the lecture block *Foundations of Probability Theory*, we discussed Hoeffding's inequality:

Hoeffding's inequality: Assume independent random variables $x_1, x_2, \ldots, x_m$ from the same distribution and finite support $x \in [a, b]$. Then:

$$P\left(\left| \frac{1}{m} \sum_i x_i \ - \mathrm{E}(x_1) \right| \geq \epsilon \right) \leq 2\, e^{-\frac{2m\epsilon^2}{(b-a)^2}} \quad \forall \epsilon > 0$$

Observation:
The risk $R(h)$ is an **expectation value**,
and the empirical risk the corresponding **arithmetic mean**!
If we restrict the loss function to be between, e.g., [0,1], then the risk is an expectation value of a random variable with **finite support**!

→ **Hoeffding's inequality applies! :)**

Learning bound: finite hypothesis set

Equipped with this knowledge, lets assume we have a finite hypothesis set $\mathcal{H}$. We start with the probability that the supremum is larger than some certain error $\epsilon > 0$:

$$p\left(\sup_{h\in\mathcal{H}}\left|R(h)-\hat{R}(h)\right| \geq \epsilon\right) \leq \sum_{h\in\mathcal{H}} p\left(\left|R(h)-\hat{R}(h)\right| \geq \epsilon\right) \leq \sum_{h\in\mathcal{H}} 2\,e^{-2m\epsilon^2} \leq |\mathcal{H}|2e^{-2m\epsilon^2} = \delta$$

Union bound:
$p(\text{largest one} \geq \epsilon) \leq p(\text{at least one} \geq \epsilon)$

Hoeffding

# training samples

With this, we get:

Let $\mathcal{H}$ be a finite hypothesis set. Then for every $\delta > 0$, the following inequality holds with probability at least $1-\delta$ for all $h \in \mathcal{H}$:

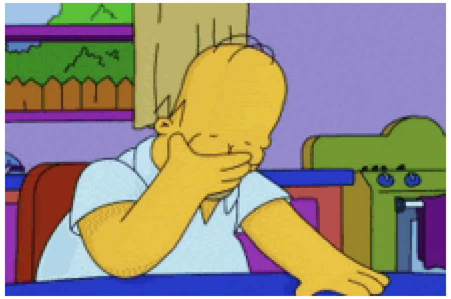
$$\left|R(h)-\hat{R}(h)\right| \leq \sqrt{\frac{\ln(|\mathcal{H}|)+\ln\left(\frac{2}{\delta}\right)}{2m}} = \epsilon$$

PAC Learning

Learning algorithms that have such a learning bound are also known as **PAC learnable**:

Probably Approximately Correct

$1 - \delta$ ϵ



We can squeeze out some more by solving ϵ for the number of training samples m:

Let $\mathcal{H}$ be a finite hypothesis set. Then for every $\delta > 0$ and $\epsilon > 0$, we have for all $h \in \mathcal{H}$

$$p\left(\left|R(h) - \hat{R}(h)\right| \leq \epsilon\right) \geq 1 - \delta$$

if we have at least $m \geq \frac{\ln(|\mathcal{H}|) + \ln\left(\frac{2}{\delta}\right)}{2\epsilon}$ training samples.

Problem: what happens in the limit of infinite hypotheses?

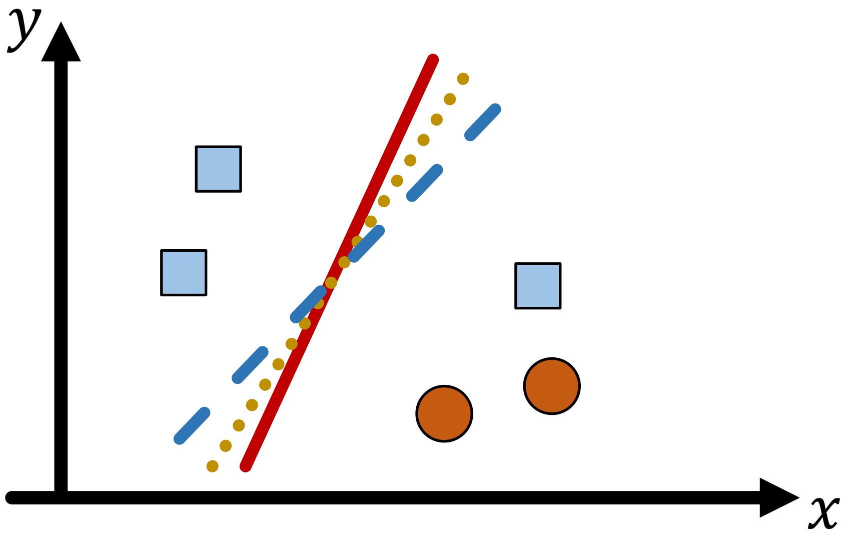
$$\sup_{h\in\mathcal{H}}\left|R(h)-\hat{R}(h)\right| \le \sqrt{\frac{\ln(|\mathcal{H}|)+\ln\left(\frac{2}{\delta}\right)}{2m}}$$

The bound goes to infinity!

But why? Obviously linear regression, neural networks, etc. all generalize to some degree!

Reason: we assumed independence of the hypotheses! But what if two hypotheses have very similar outputs for all possible inputs? We shouldn't count them separately!

Illustration: classification



All lines shown here produce similar outputs!

Thus, instead of counting them individually, we should count them as **"one class" of "equivalent" functions**!

Covering number

Let's formalize this!

Assume we have our hypothesis set $\mathcal{H}$. We will now decompose the set into "equivalence" classes. Per class $C_k \subset \mathcal{H}$, we have one representative function $h_k \in \mathcal{H}$ such that for all $h \in C_k, |h - h_k|_\infty < \kappa$ for $\kappa > 0$.

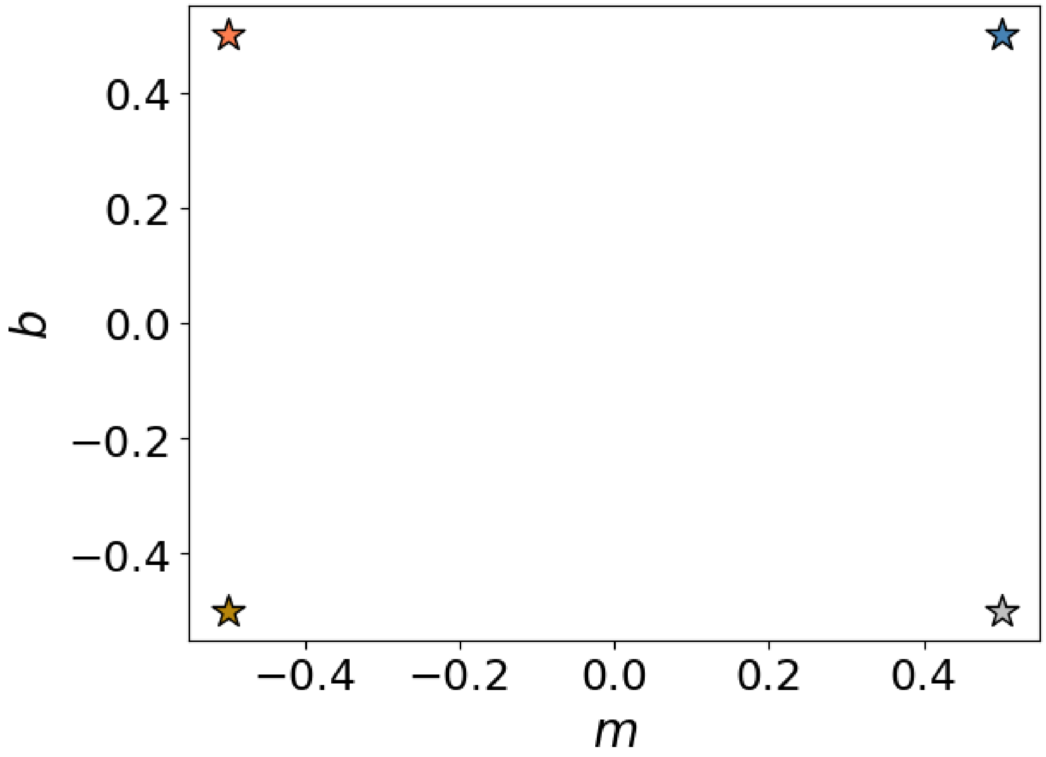
The covering number $d_C(\kappa)$ is the minimum number of classes k required to **fully cover** $\mathcal{H}$, i.e., such that "$\bigcup_{i=1}^{k} C_k = \mathcal{H}$" up to error κ.

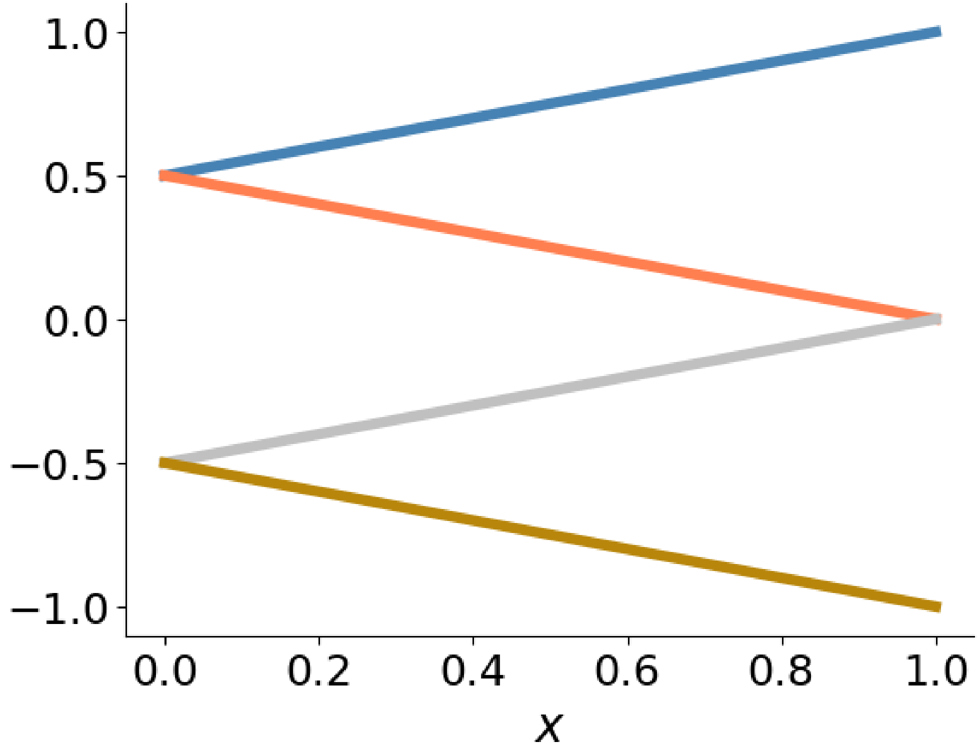
Example

Linear regression: $f_{m,b}(x) = mx + b$ with $m \in [-0.5, 0.5], b \in [-0.5, 0.5], x \in [0,1]$.

Assume $\kappa = 0.5$. Then the following four functions cover all realizations of f:

$$f_{0.5,0.5}, f_{-0.5,0.5}, f_{0.5,-0.5}, f_{-0.5,-0.5}$$



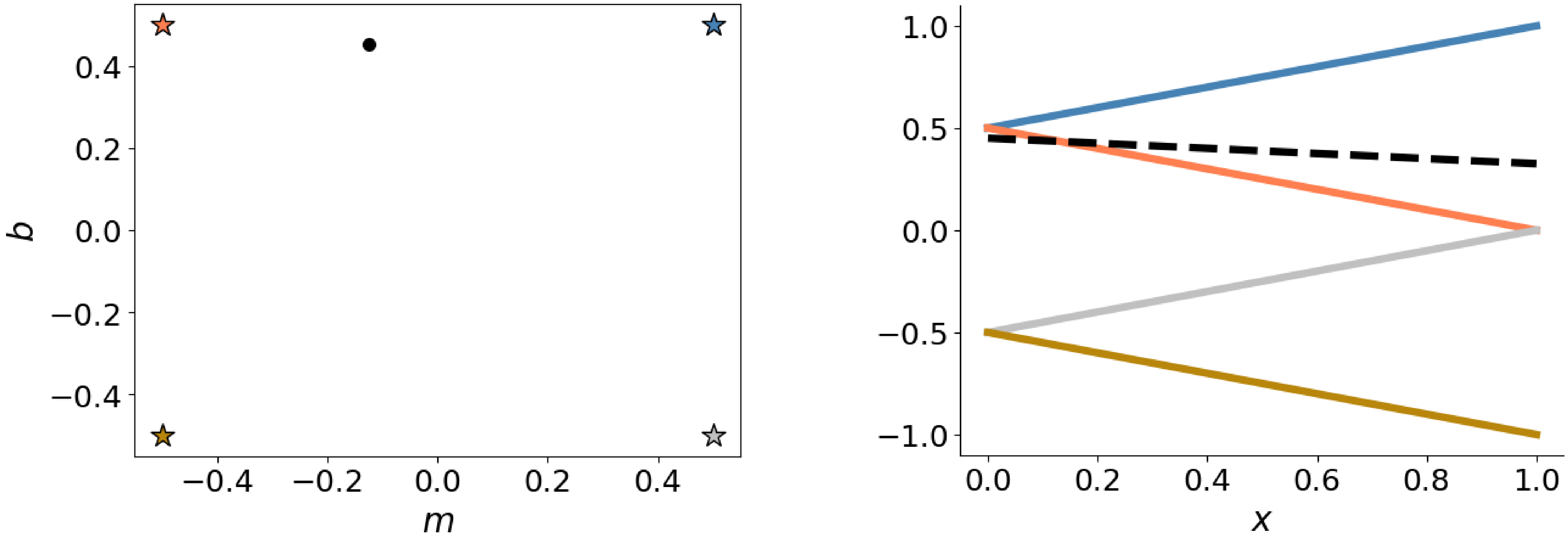


Example

Linear regression: $f_{m,b}(x) = mx + b$ with $m \in [-0.5, 0.5], b \in [-0.5, 0.5], x \in [0,1]$.

Assume $\kappa = 0.5$. Then the following four functions cover all realizations of f:

$$f_{0.5,0.5}, f_{-0.5,0.5}, f_{0.5,-0.5}, f_{-0.5,-0.5}$$

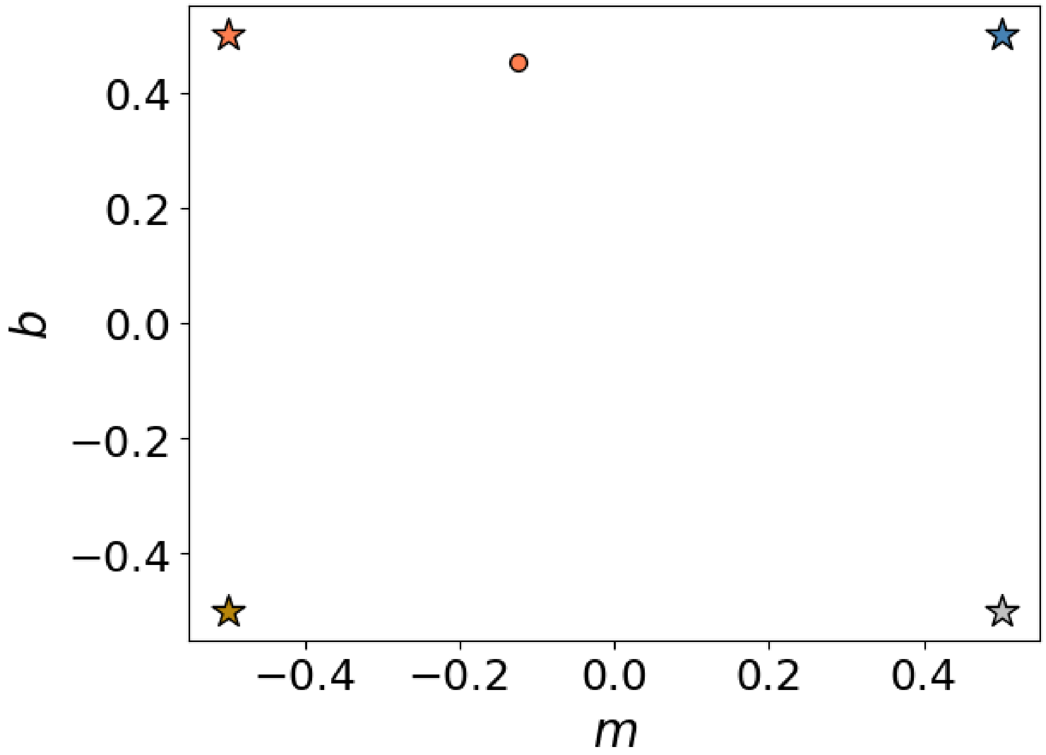


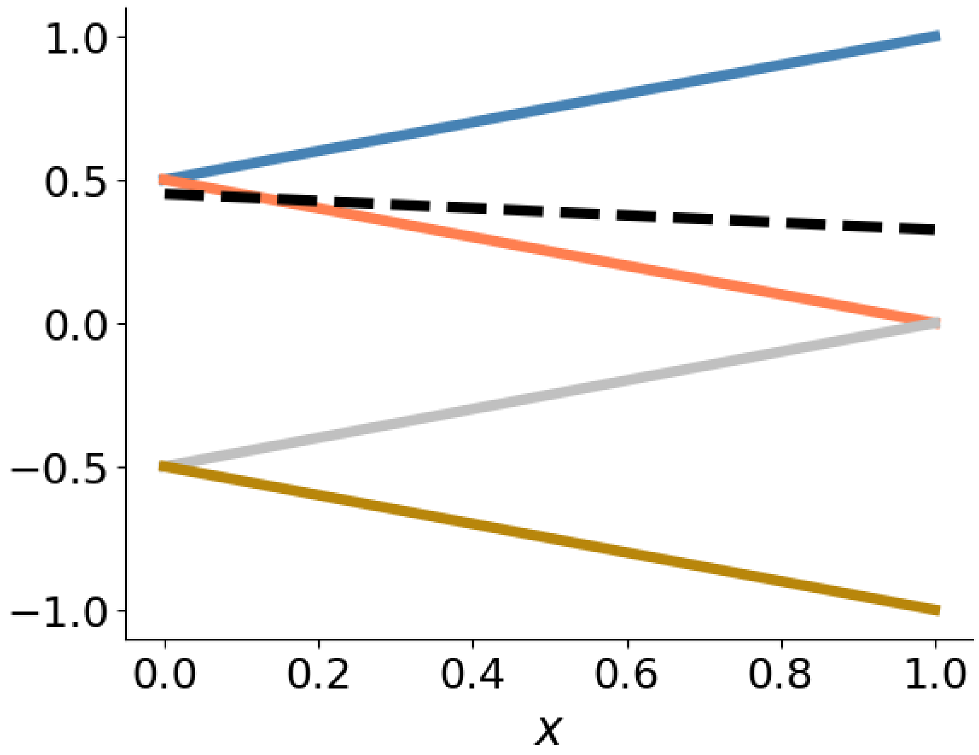
Example

Linear regression: $f_{m,b}(x) = mx + b$ with $m \in [-0.5, 0.5], b \in [-0.5, 0.5], x \in [0,1]$.

Assume $\kappa = 0.5$. Then the following four functions cover all realizations of f:

$$f_{0.5,0.5}, f_{-0.5,0.5}, f_{0.5,-0.5}, f_{-0.5,-0.5}$$





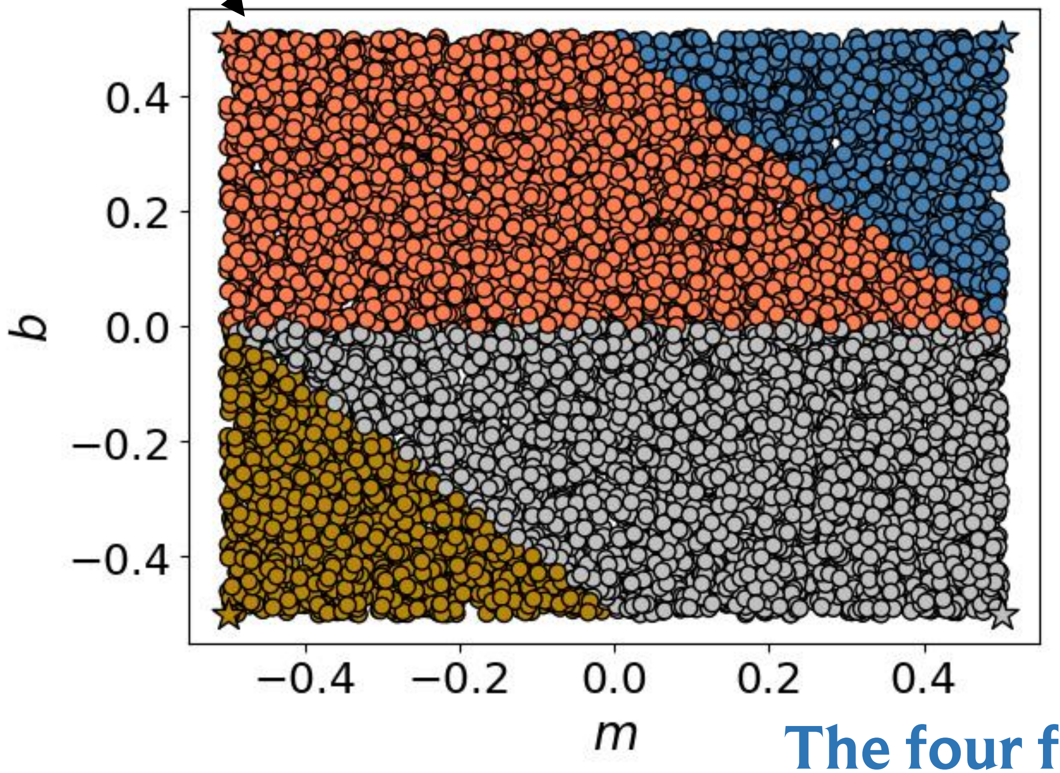
Example

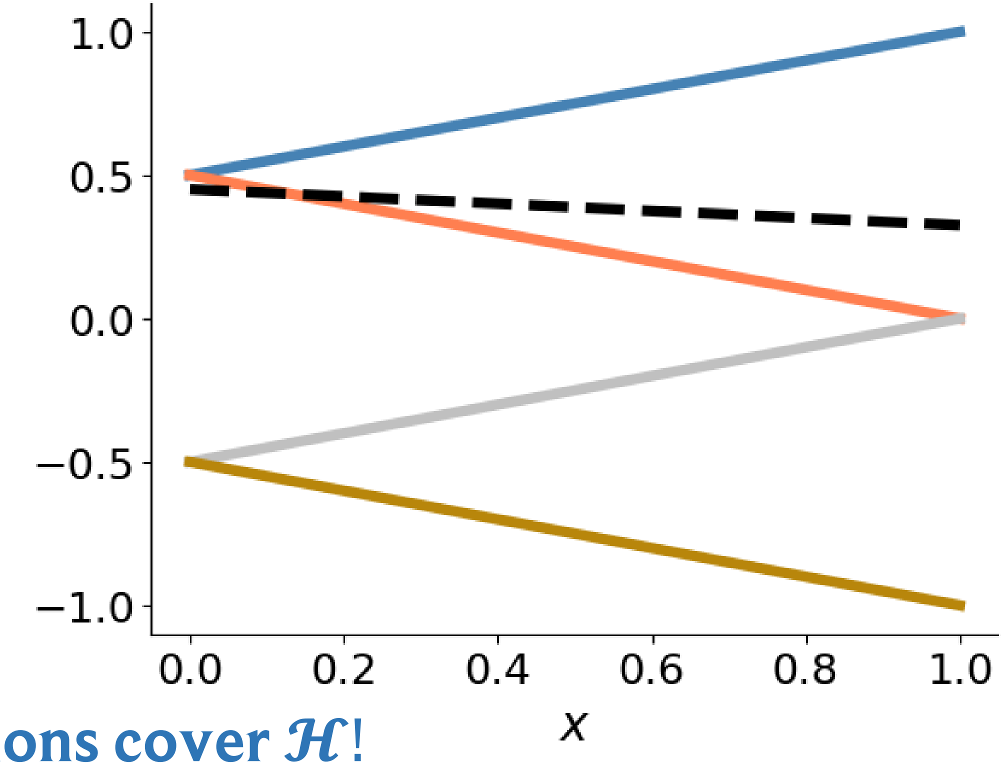
Linear regression: $f_{m,b}(x) = mx + b$ with $m \in [-0.5, 0.5], b \in [-0.5, 0.5], x \in [0,1]$.

Assume $\kappa = 0.5$. Then the following four functions cover all realizations of f:

assign random functions to one of the four classes such that the distance is smaller than κ

$$f_{0.5,0.5}, f_{-0.5,0.5}, f_{0.5,-0.5}, f_{-0.5,-0.5}$$





The four functions cover $\mathcal{H}$!

Learning bound: covering numbers

Previously, we had:

Let $\mathcal{H}$ be a finite hypothesis set. Then for every $\delta > 0$, the following inequality holds with probability at least $1 - \delta$ for all $h \in \mathcal{H}$:

$$\left|R(h) - \hat{R}(h)\right| \leq \sqrt{\frac{\ln(|\mathcal{H}|) + \ln(2/\delta)}{2m}}$$

Using covering numbers, we get *(using a rather similar derivation)*:

Let $\mathcal{H}$ be a hypothesis set with covering number $\mathcal{G}$. Then for every $\delta > 0$, and for C_L-Lipschitz loss function with output range $[-C, C]$, we have with probability at least $1 - \delta$ for all $h \in \mathcal{H}$ and m training data samples:

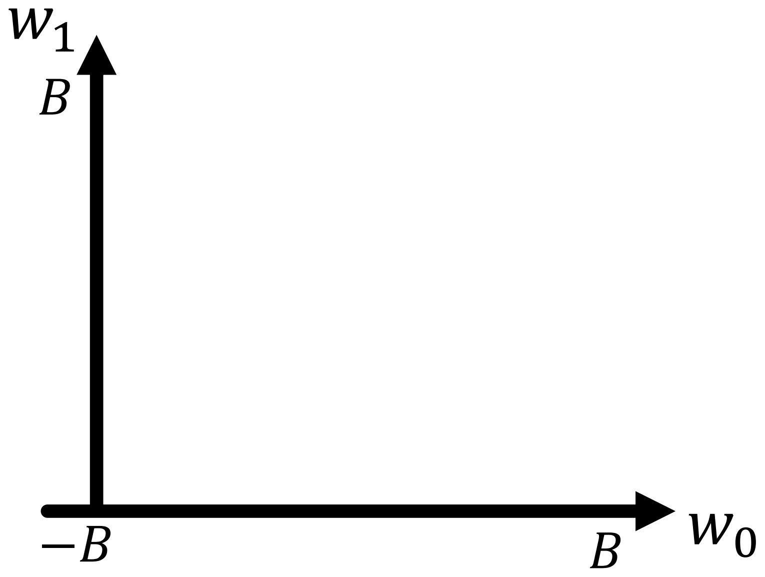
$$\left|R(h) - \hat{R}(h)\right| \leq 4C \cdot C_L \cdot \sqrt{\frac{\ln \mathcal{G} + \ln(2/\delta)}{m}} + \frac{2C_L}{m^{\alpha}}$$

size of the "balls" κ we use for covering $\mathcal{H}$

Covering numbers from Lipschitz constants

If our neural network is ***C*-Lipschitz in the parameters** *(which it is for most activation functions used in practice),* then there is a clever way of **bounding the covering number**!

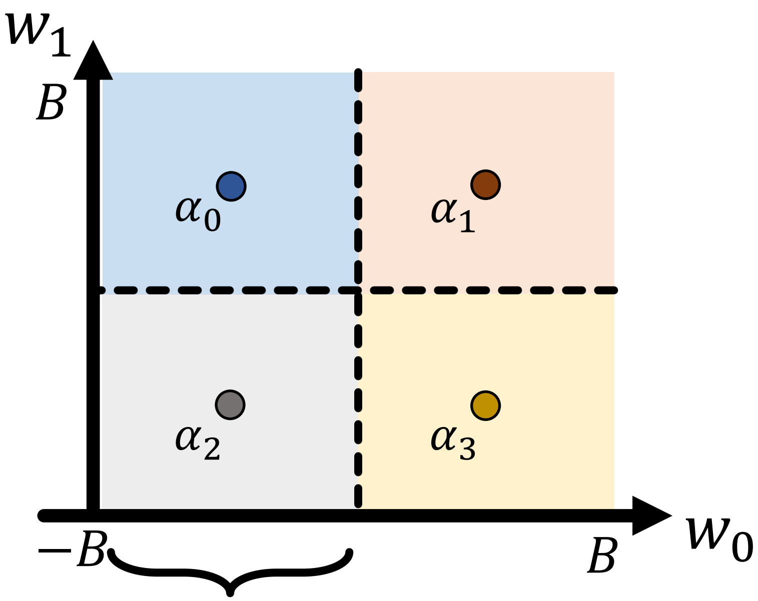
Taking the reverse route: cover the parameter space!
→ Find w_i such that for all $w \in [-B, B]^n$, there is at least one α_i with $|w - \alpha_i|_\infty \leq \frac{\kappa}{C}$.



Covering numbers from Lipschitz constants

If our neural network is ***C*-Lipschitz in the parameters** *(which it is for most activation functions used in practice),* then there is a clever way of **bounding the covering number**!

Taking the reverse route: cover the parameter space!
→ Find w_i such that for all $w \in [-B, B]^n$, there is at least one α_i with $|w - \alpha_i|_\infty \leq \frac{\kappa}{C}$.



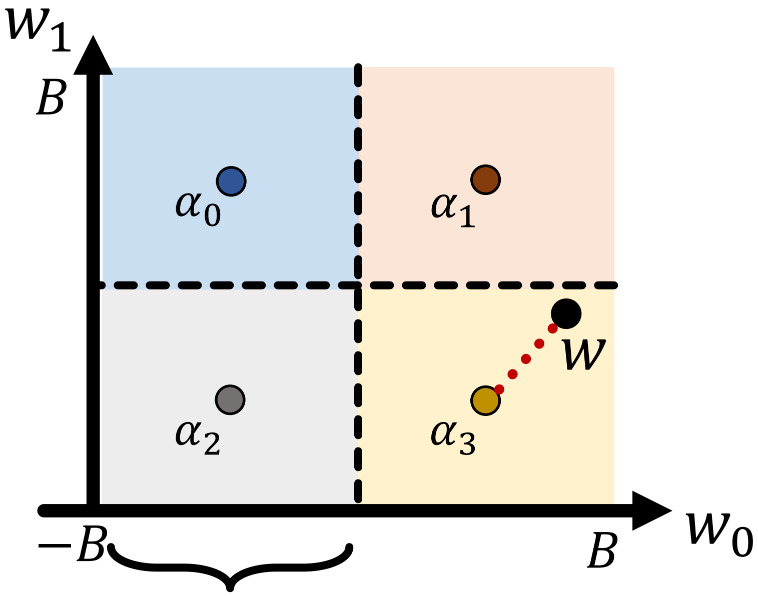
Size: κ/C

Thus: $\mathcal{G} \leq (B \cdot C/\kappa)^n$ = # of boxes here

Covering numbers from Lipschitz constants

If our neural network is ***C*-Lipschitz in the parameters** *(which it is for most activation functions used in practice),* then there is a clever way of **bounding the covering number**!

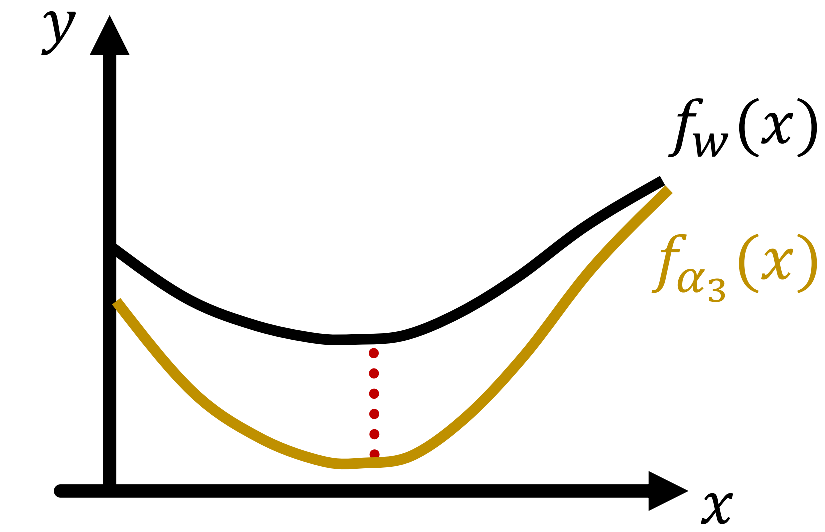
Taking the reverse route: cover the parameter space!
→ Find w_i such that for all $w \in [-B, B]^n$, there is at least one α_i with $|w - \alpha_i|_\infty \leq \frac{\kappa}{C}$.



Size: κ/C

Thus: $\mathcal{G} \leq (B \cdot C/\kappa)^n = \#$ of boxes here

Look at any w!



Via Lipschitz property, we have
$\left|f_w(x) - f_{\alpha_i}(x)\right|_\infty \leq C|w - \alpha_i|_\infty = \kappa$,
→ **We also covered $\mathcal{H}$**!

Finale: Lipschitz constant of ReLU neural networks

A ReLU neural network with n parameters, max. width d, L layers, an activation function with Lipschitz constant C_ϕ, and parameters constrained to $[-B, B]$, has Lipschitz constant:

1. If $B \geq 1$: $C = n \cdot \left(2\, C_\phi B\, d\right)^L$
2. If B > 0: $C = C_\phi^L \cdot (B\, d)^{L+1}$

From which we get the covering numbers, e.g.,:

1. If $B \geq 1$: $\mathcal{G} \leq (n/\kappa)^n \cdot \left(2\, C_\phi B\, d\right)^{n \cdot L}$
2. If B > 0: $\mathcal{G} \leq \left(C_\phi^L/\kappa\right)^n \cdot (B\, d)^{nL+n}$

These can be simply inserted into our generalization bounds!

Outro: data science in the overparametrized regime

Classical piece of wisdom: the more parameters your model has, the more expressive it is. However, with many more parameters than training data, it will start overfitting!

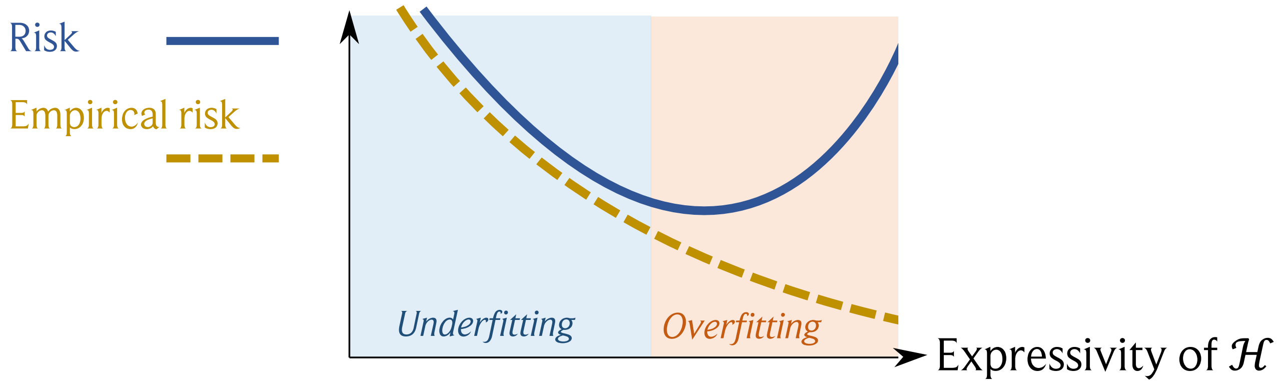
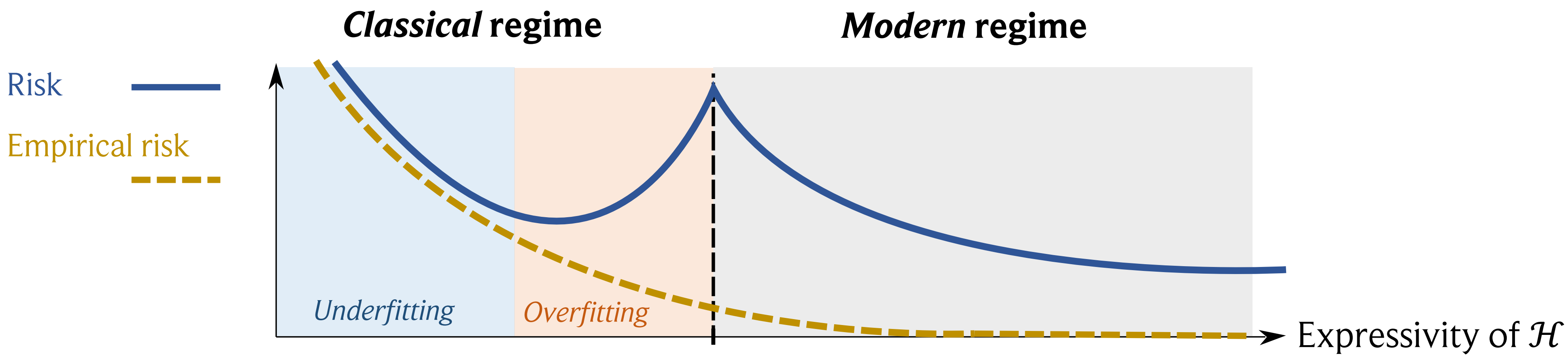


Image adapted from: Philipp Petersen, Jakob Zech. Mathematical theory of deep learning (2025)

Outro: data science in the overparametrized regime

Classical piece of wisdom: the more parameters your model has, the more expressive it is. However, with many more parameters than training data, it will start overfitting!



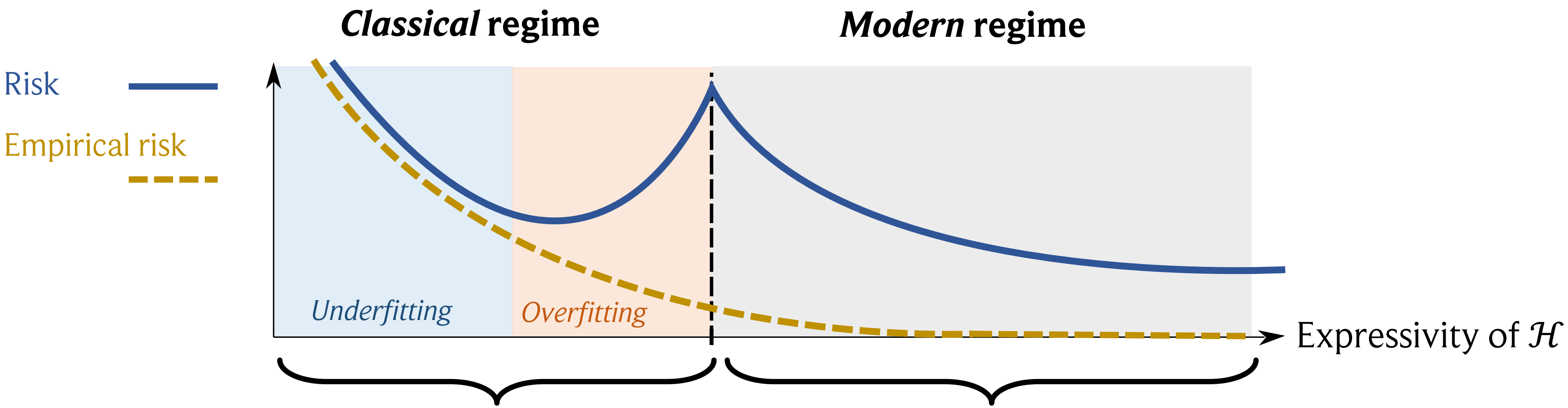
However, in practice we observe that highly overparametrized models can generalize well! A famous phenomena is the above curve, called **"double descent"** .

Note that this phenomenon is **not exclusive to neural networks**:
one can recreate it for linear models, polynomials, regression with basis functions, etc.!

There are several explanations for this behaviour … we will briefly touch on one of them.

Image adapted from: Philipp Petersen, Jakob Zech. Mathematical theory of deep learning (2025)

One explanation: the power of many, small parameters



Only few parameters, which potentially have to be large to fit the data. We have:

$$\mathcal{G} \leq (n/\kappa)^n \cdot \left(2\, C_\phi B\, d\right)^{n \cdot L}$$

Generalization bound depends on # parameters!

Many parameters, so they can be quite small*

(large values are obtained by summing up many inputs!)

Thus, we can choose $B \leq \left(d\, C_\phi\right)^{-1}$, leading to $C \approx 1$.
The hypothesis set of C-Lipschitz functions (here: $C = 1$) has a covering number** that only depends on the # input features d_0:

$$\log \mathcal{G} \leq \alpha \kappa^{-d_0} \qquad \text{with constant } \alpha > 0$$

Generalization bound is constant w.r.t. # parameters!
We see a decrease since the empirical risk still improves.

* *Note that most types of regularization ensure small weights (L1, L2, etc.)! We also often initialize neural networks with weight scales that depend on the width!*
** *Both bounds shown here are valid for both regimes. We get the "double" curve by choosing the bound that is smaller!*

Image adapted from: Philipp Petersen, Jakob Zech. Mathematical theory of deep learning (2025)