Deep Learning

Lecture 1: Fundamentals of machine learning

Prof. Gilles Louppe
g.louppe@uliege.be

LIÈGE université

Today

A recap on statistical learning:

- Supervised learning
- Empirical risk minimization
- Under-fitting and over-fitting
- Bias-variance dilemma

Statistical learning

Supervised learning

Consider an unknown joint probability distribution $p_{X,Y}$.

Assume training data

$$(\mathbf{x}_i, y_i) \sim p_{X,Y},$$

with $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, ..., N$.

- In most cases,
  - $\mathbf{x}_i$ is a p-dimensional vector of features or descriptors,
  - y_i is a scalar (e.g., a category or a real value).
- The training data is generated i.i.d.
- The training data can be of any finite size N.
- In general, we do not have any prior information about $p_{X,Y}$.

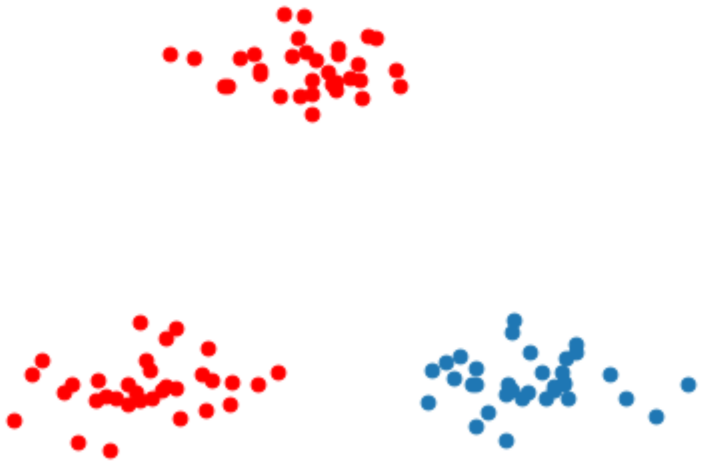
Supervised learning is usually concerned with the two following inference problems:

- Classification: Given $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} = \mathbb{R}^p \times \triangle^C$, for $i = 1, ..., N$, we want to estimate for any new $\mathbf{x}$,

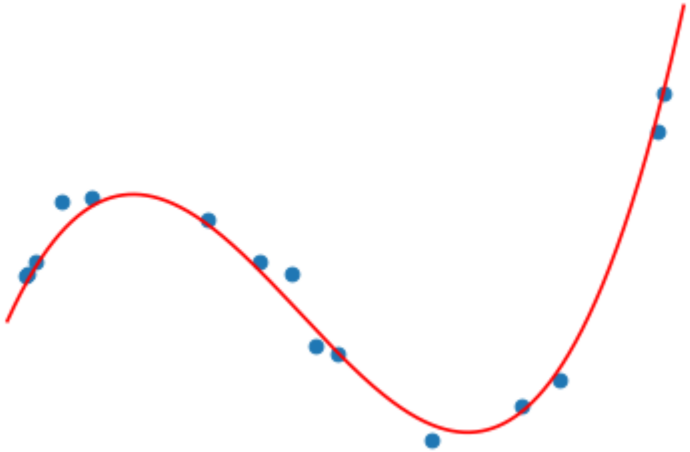
$$\arg\max_y p(Y = y | X = \mathbf{x}).$$

- Regression: Given $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} = \mathbb{R}^p \times \mathbb{R}$, for $i = 1, ..., N$, we want to estimate for any new $\mathbf{x}$,

$$\mathbb{E}\left[Y | X = \mathbf{x}\right].$$



Classification consists in identifying
a decision boundary between objects of distinct classes.



Regression aims at estimating relationships among (usually continuous) variables.

Probabilistic perspective

Supervised learning can be framed as probabilistic inference, where the goal is to estimate the conditional distribution

$$p(Y = y|X = \mathbf{x})$$

for any new $(\mathbf{x}, y)$.

This is the framing we will adopt in this course (starting from Lecture 2).

Empirical risk minimization

The traditional perspective on supervised learning is empirical risk minimization.

Consider a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ produced by some learning algorithm. The predictions of this function can be evaluated through a loss

$$\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R},$$

such that $\ell(y, f(\mathbf{x})) \geq 0$ measures how close the prediction $f(\mathbf{x})$ from y is.

Examples of loss functions

Classification:	$\ell(y, f(\mathbf{x})) = \mathbf{1}_{y \neq f(\mathbf{x})}$
Regression:	$\ell(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$

Let $\mathcal{F}$ denote the hypothesis space, i.e. the set of all functions f that could be produced by the learning algorithm (e.g., the set of all neural networks with a given architecture).

We are looking for a function $f \in \mathcal{F}$ with a small expected risk (or generalization error)

$$R(f) = \mathbb{E}_{(\mathbf{x},y)\sim p_{X,Y}} \left[\ell(y, f(\mathbf{x})) \right].$$

This means that for a given data generating distribution $p_{X,Y}$ and for a given hypothesis space $\mathcal{F}$, the optimal model is

$$f_* = \arg\min_{f \in \mathcal{F}} R(f).$$

Since $p_{X,Y}$ is unknown, the expected risk cannot be evaluated and the optimal model cannot be determined.

However, if we have i.i.d. training data $\mathbf{d} = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N\}$, we can compute an estimate, the empirical risk (or training error)

$$\hat{R}(f, \mathbf{d}) = \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{d}} \ell(y_i, f(\mathbf{x}_i)).$$

This estimator is unbiased and can be used for finding a good enough approximation of f_*. This results into the empirical risk minimization principle:

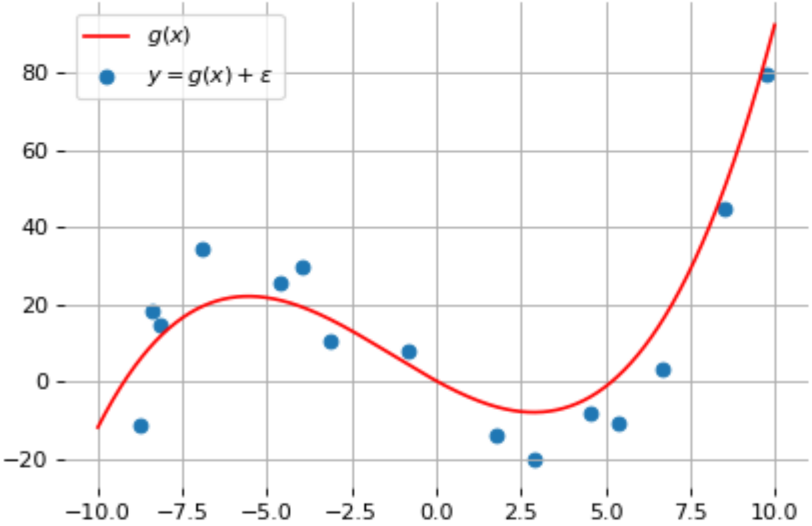
$$f_*^{\mathbf{d}} = \arg\min_{f \in \mathcal{F}} \hat{R}(f, \mathbf{d})$$

Most machine learning algorithms, including neural networks, implement empirical risk minimization.

Under regularity assumptions, empirical risk minimizers converge:

$$\lim_{N \to \infty} f_*^{\mathbf{d}} = f_*$$

Polynomial regression



Consider the joint probability distribution $p_{X,Y}$ induced by the data generating process

$$(x, y) \sim p_{X,Y} \Leftrightarrow x \sim U[-10; 10], \epsilon \sim \mathcal{N}(0, \sigma^2), y = g(x) + \epsilon$$

where $x \in \mathbb{R}$, $y \in \mathbb{R}$ and g is an unknown polynomial of degree 3.

Our goal is to find a function f that makes good predictions on average over $p_{X,Y}$.

Consider the hypothesis space $f \in \mathcal{F}$ of polynomials of degree 3 defined through their parameters $\mathbf{w} \in \mathbb{R}^4$ such that

$$\hat{y} \triangleq f(x; \mathbf{w}) = \sum_{d=0}^{3} w_d x^d$$

For this regression problem, we use the squared error loss

$$\ell(y, f(x; \mathbf{w})) = (y - f(x; \mathbf{w}))^2$$

to measure how wrong the predictions are.

Therefore, our goal is to find the best value $\mathbf{w}_*$ such that

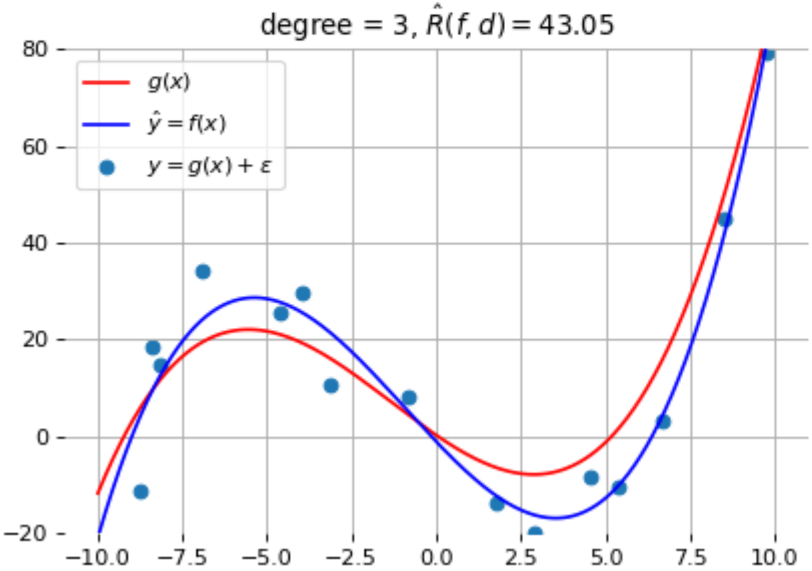
$$\begin{aligned} \mathbf{w}_* &= \arg\min_{\mathbf{w}} R(\mathbf{w}) \\ &= \arg\min_{\mathbf{w}} \mathbb{E}_{(x,y)\sim p_{X,Y}} \left[(y - f(x; \mathbf{w}))^2 \right] \end{aligned}$$

Given a large enough training set $\mathbf{d} = \{(x_i, y_i) | i = 1, \ldots, N\}$, the empirical risk minimization principle tells us that a good estimate $\mathbf{w}_*^{\mathbf{d}}$ of $\mathbf{w}_*$ can be found by minimizing the empirical risk:

$$\begin{aligned}
\mathbf{w}_*^{\mathbf{d}} &= \arg\min_{\mathbf{w}} \hat{R}(\mathbf{w}, \mathbf{d}) \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{(x_i, y_i) \in \mathbf{d}} (y_i - f(x_i; \mathbf{w}))^2 \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{(x_i, y_i) \in \mathbf{d}} (y_i - \sum_{d=0}^{3} w_d x_i^d)^2 \\
&= \arg\min_{\mathbf{w}} \frac{1}{N} \left\| \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{pmatrix}}_{\mathbf{y}} - \underbrace{\begin{pmatrix} x_1^0 \ldots x_1^3 \\ x_2^0 \ldots x_2^3 \\ \ldots \\ x_N^0 \ldots x_N^3 \end{pmatrix}}_{\mathbf{X}} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix} \right\|^2
\end{aligned}$$

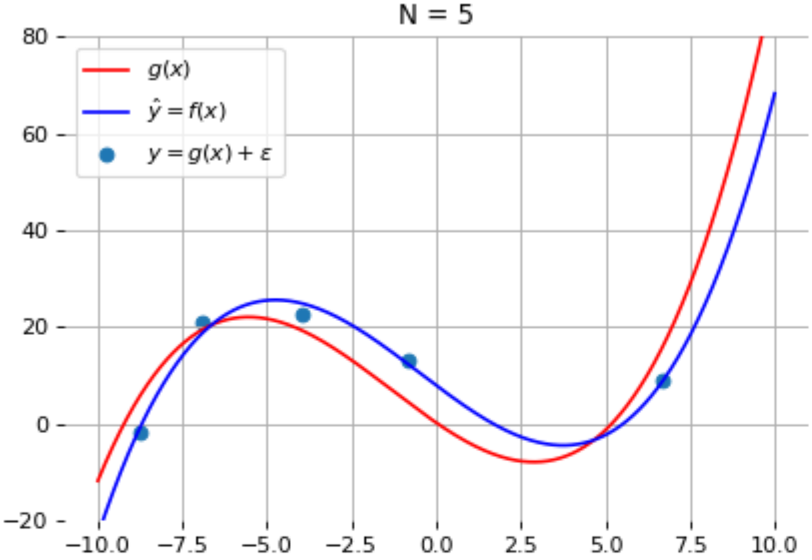
This is ordinary least squares regression, for which the solution is derived as

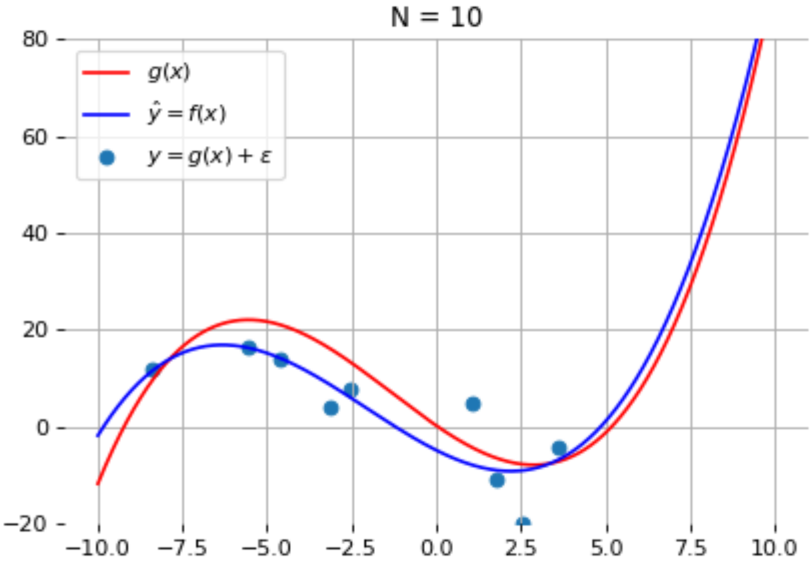
$$\mathbf{w}_*^{\mathbf{d}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

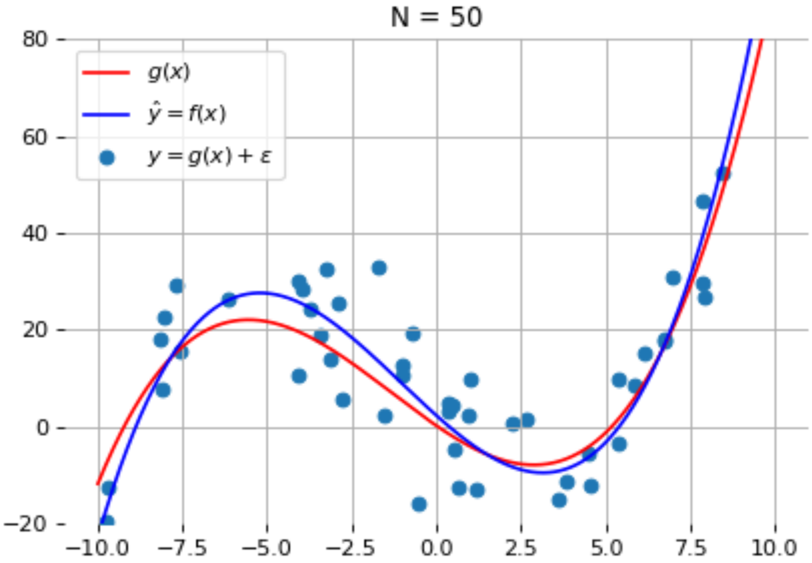


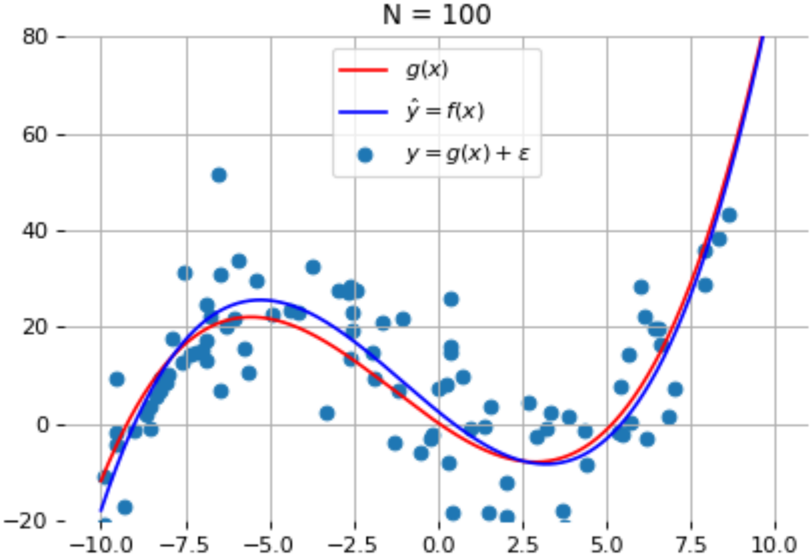
The expected risk minimizer $\mathbf{w}_*$ within our hypothesis space is g itself.

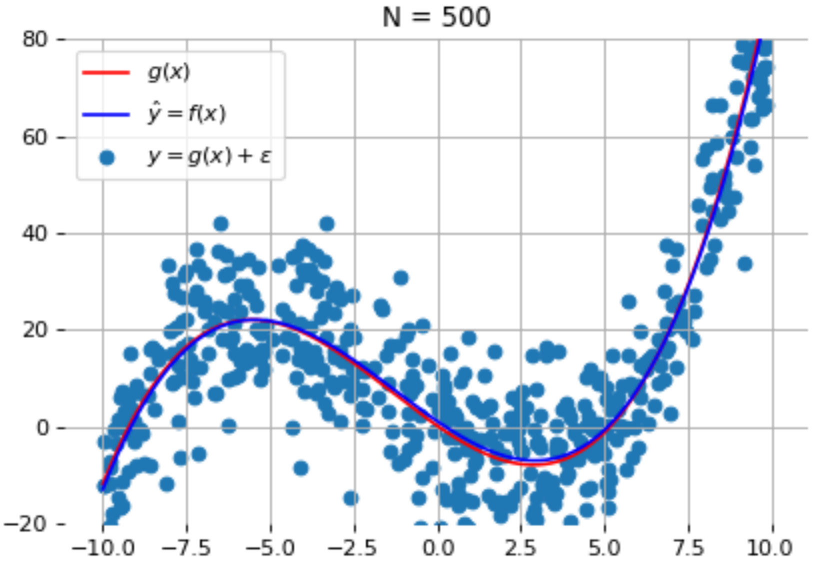
Therefore, on this toy problem, we can verify that $f(x; \mathbf{w}_*^{\mathbf{d}}) \rightarrow f(x; \mathbf{w}_*) = g(x)$ as $N \rightarrow \infty$.


N = 5
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
$g(x)$
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$


N = 10
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
$g(x)$
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$

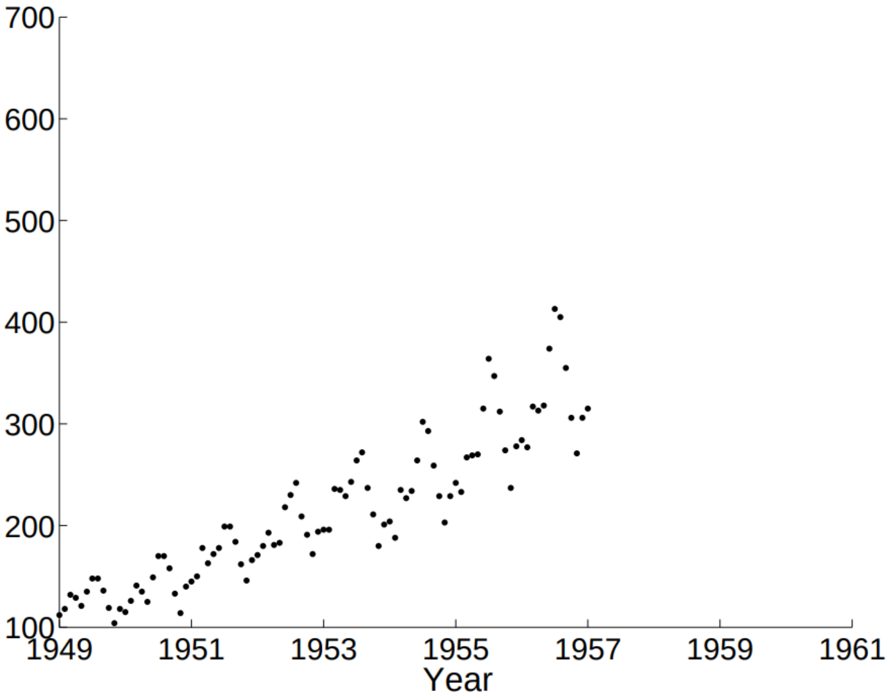

N = 50
$g(x)$
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


N = 100
$g(x)$
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


N = 500
g(x)
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

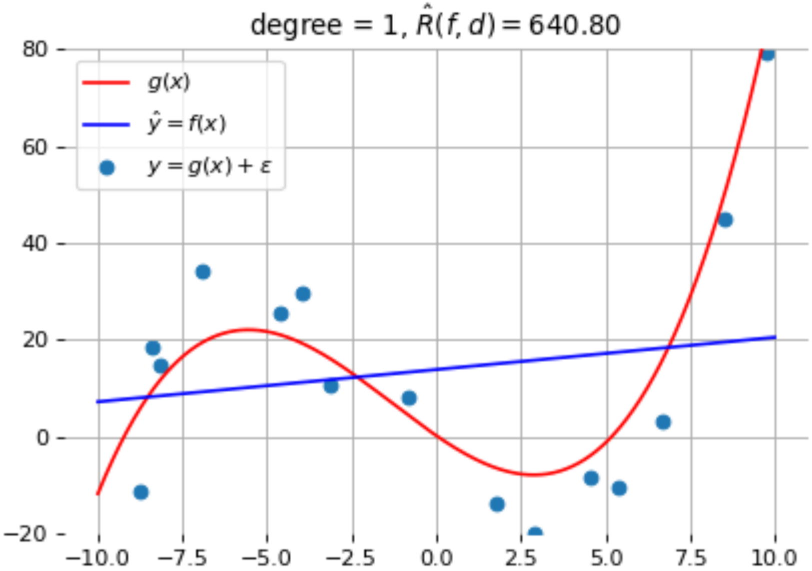
Under-fitting and over-fitting

What if we consider a hypothesis space $\mathcal{F}$ in which candidate functions f are either too "simple" or too "complex" with respect to the true data generating process?

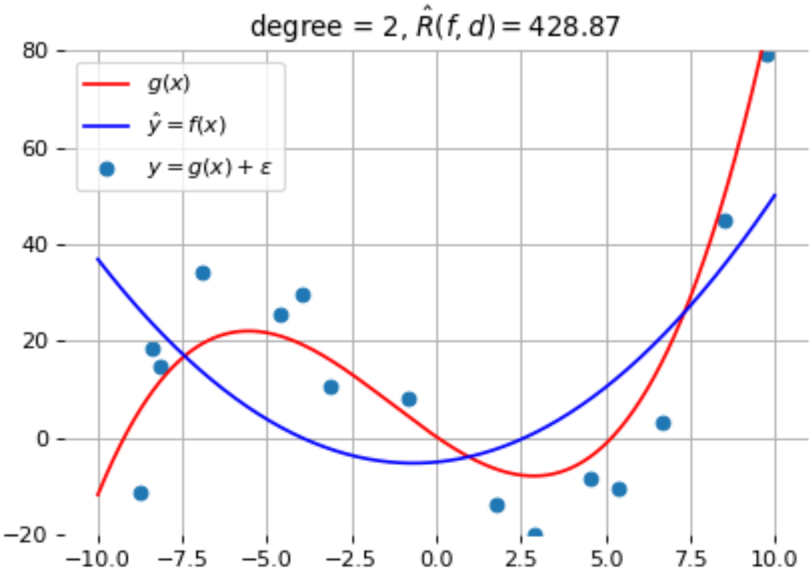


Which model would you choose?

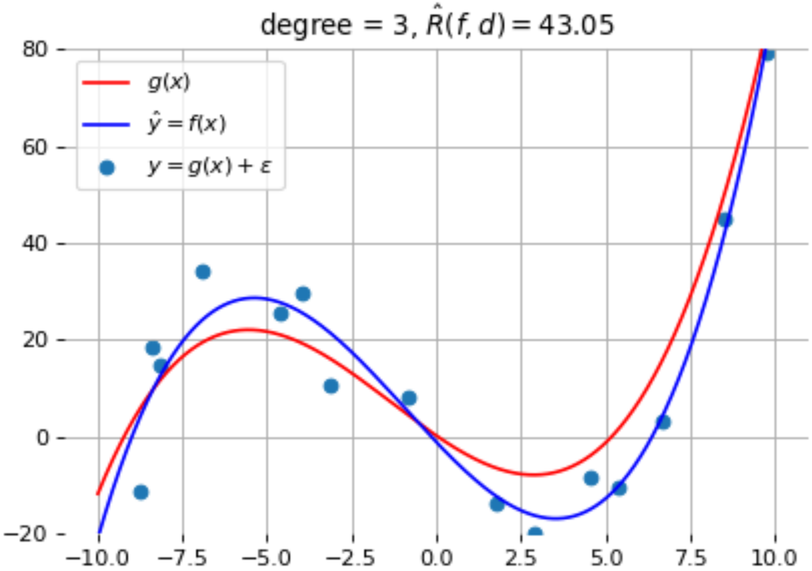
$f_1(x) = w_0 + w_1 x$ $\qquad f_2(x) = \sum_{j=0}^{3} w_j x^j$ $\qquad f_3(x) = \sum_{j=0}^{10^4} w_j x^j$



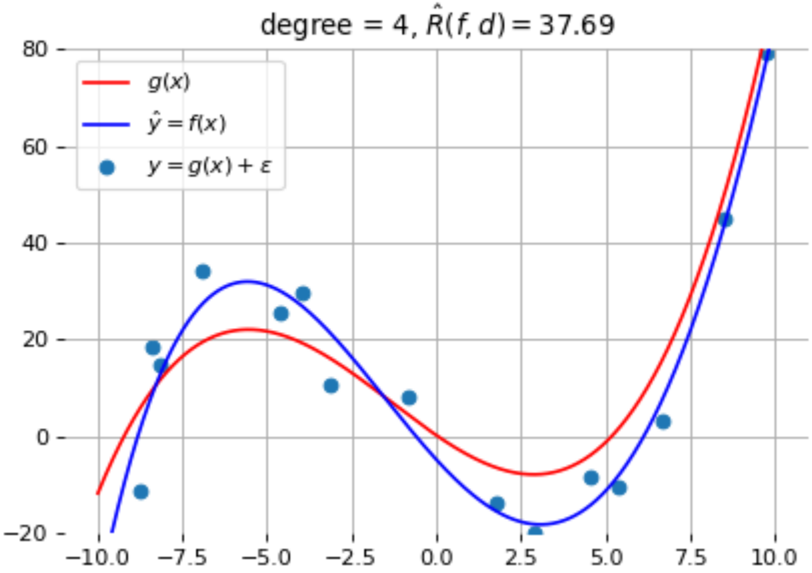
$\mathcal{F}$ = polynomials of degree 1



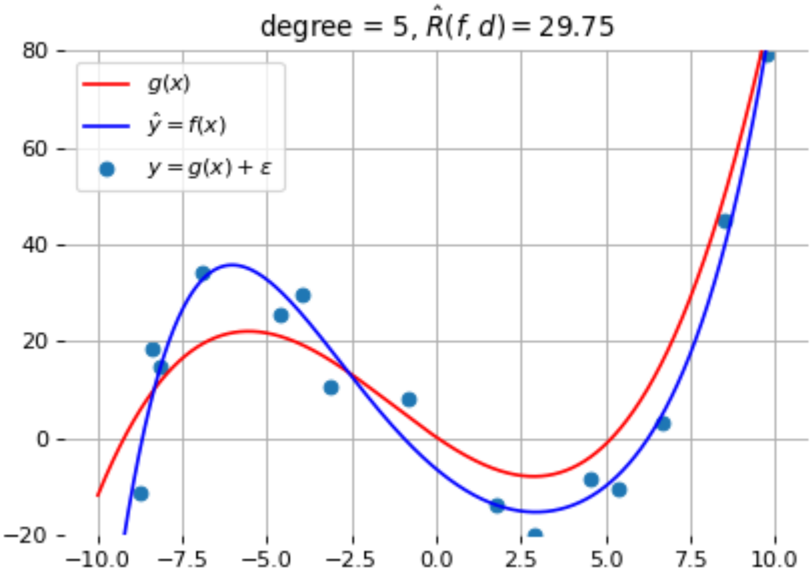
$\mathcal{F}$ = polynomials of degree 2



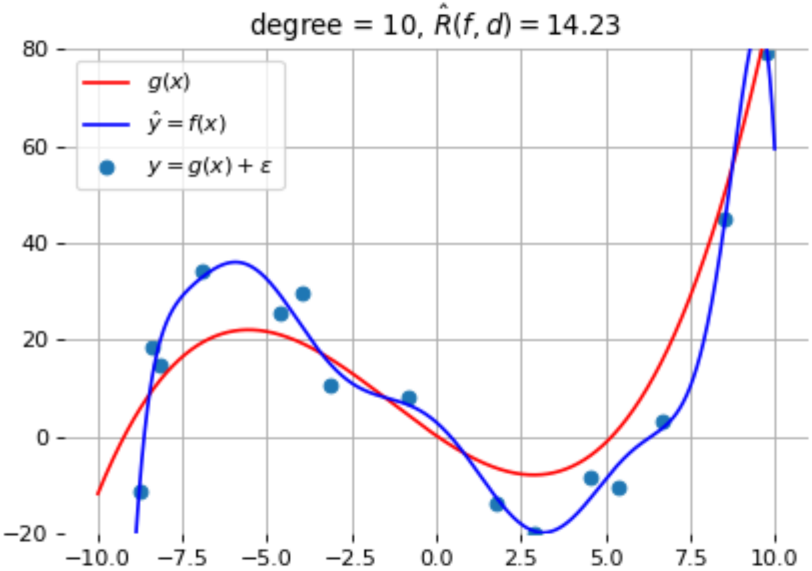
$\mathcal{F}$ = polynomials of degree 3



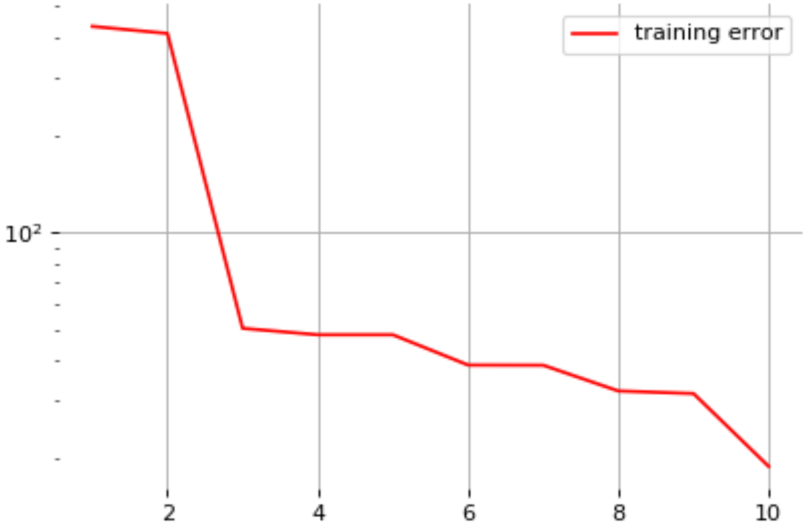
$\mathcal{F}$ = polynomials of degree 4



$\mathcal{F}$ = polynomials of degree 5



$\mathcal{F}$ = polynomials of degree 10



Degree d of the polynomial VS. error.

The Bayes optimal model

Let $\mathcal{Y}^{\mathcal{X}}$ be the set of all functions $f : \mathcal{X} \rightarrow \mathcal{Y}$.

We define the Bayes risk as the minimal expected risk over all possible functions,

$$R_B = \min_{f \in \mathcal{Y}^{\mathcal{X}}} R(f),$$

and call the Bayes optimal model the model f_B that achieves this minimum.

No model f can perform better than f_B.

Capacity of a hypothesis space

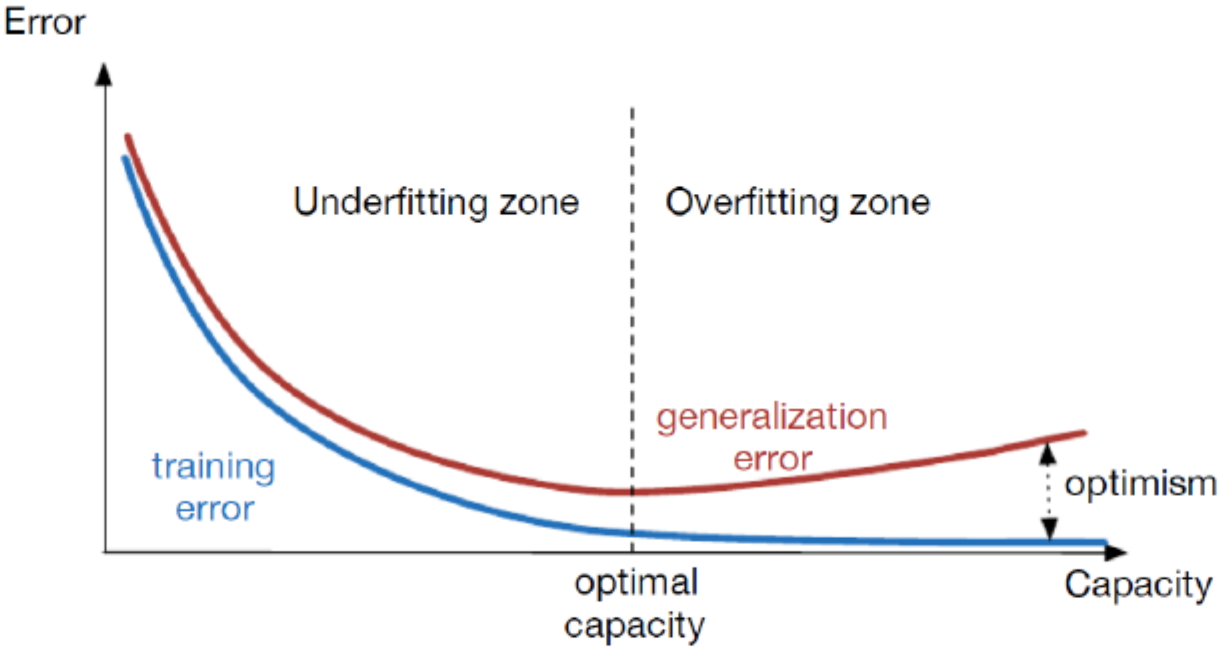
The capacity (or complexity) of a hypothesis space $\mathcal{F}$ characterizes how well functions in $\mathcal{F}$ can approximate arbitrary functions from $\mathcal{Y}^{\mathcal{X}}$.

- If the capacity of $\mathcal{F}$ is too low, then $f_B \notin \mathcal{F}$ and $R(f) - R_B$ is large for any $f \in \mathcal{F}$, including f_* and $f_*^{\mathbf{d}}$. Such models f are said to underfit the data.
- If the capacity of $\mathcal{F}$ is too high, then $f_B \in \mathcal{F}$ or $R(f_*) - R_B$ is small. However, because of the high capacity of the hypothesis space, the empirical risk minimizer $f_*^{\mathbf{d}}$ could fit the training data arbitrarily well such that

  $$R(f_*^{\mathbf{d}}) \geq R_B \geq \hat{R}(f_*^{\mathbf{d}}, \mathbf{d}) \geq 0.$$

  In this situation, $f_*^{\mathbf{d}}$ becomes too specialized with respect to the true data generating process and a large reduction of the empirical risk (often) comes at the price of an increase of the expected risk of the empirical risk minimizer $R(f_*^{\mathbf{d}})$. In this situation, $f_*^{\mathbf{d}}$ is said to overfit the data.

Therefore, our goal is to adjust the capacity of the hypothesis space such that the expected risk of the empirical risk minimizer gets as low as possible.



When overfitting,

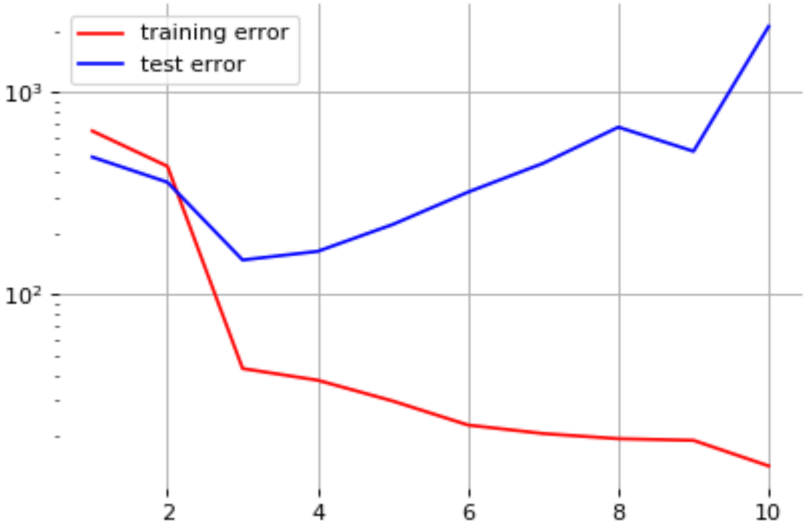
$$R(f_*^{\mathbf{d}}) \geq R_B \geq \hat{R}(f_*^{\mathbf{d}}, \mathbf{d}) \geq 0.$$

This indicates that the empirical risk $\hat{R}(f_*^{\mathbf{d}}, \mathbf{d})$ is a poor estimator of the expected risk $R(f_*^{\mathbf{d}})$.

Nevertheless, an unbiased estimate of the expected risk can be obtained by evaluating $f_*^{\mathbf{d}}$ on data $\mathbf{d}_{\text{test}}$ independent from the training samples $\mathbf{d}$:

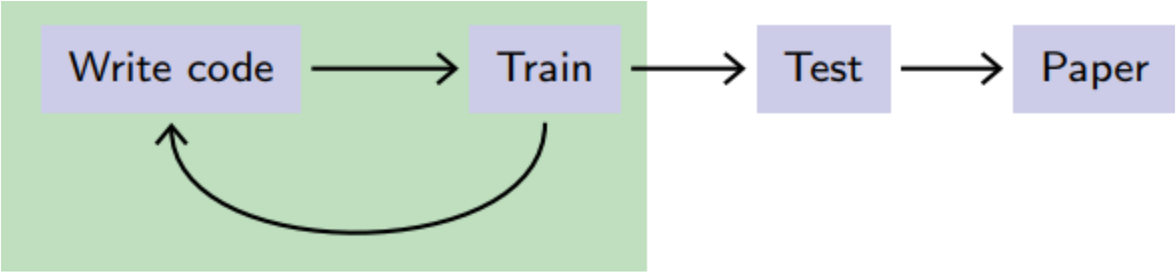
$$\hat{R}(f_*^{\mathbf{d}}, \mathbf{d}_{\text{test}}) = \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{d}_{\text{test}}} \ell(y_i, f_*^{\mathbf{d}}(\mathbf{x}_i))$$

This test error estimate can be used to evaluate the actual performance of the model. However, it should not be used, at the same time, for model selection.



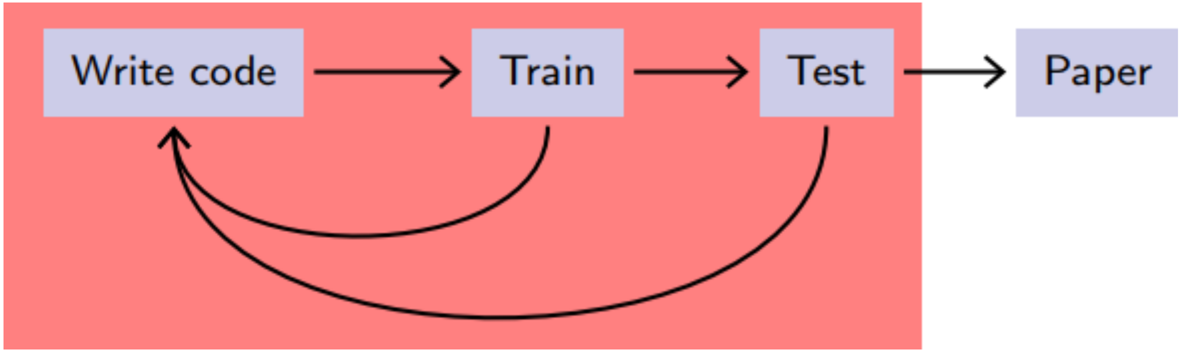
Degree d of the polynomial VS. error.

(Proper) evaluation protocol



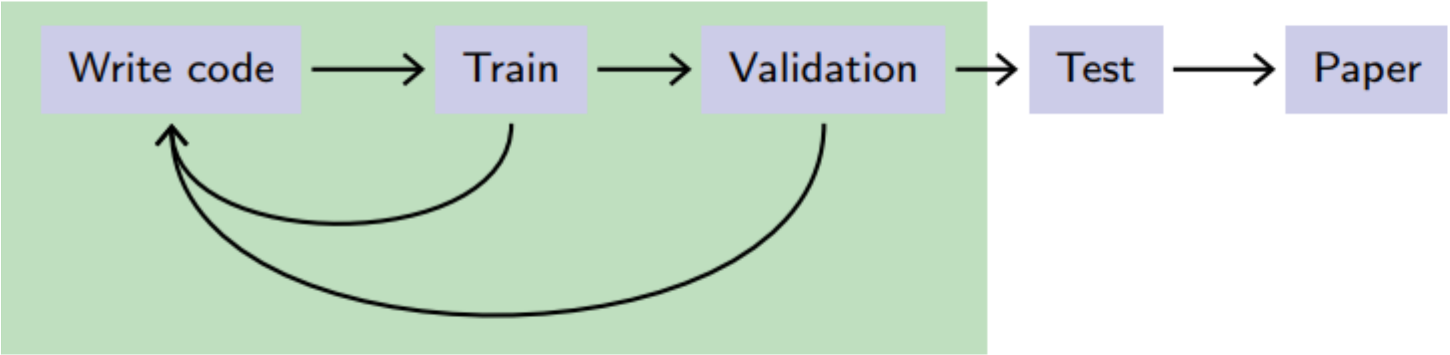
There may be over-fitting, but it does not bias the final performance evaluation.

Credits: Francois Fleuret, EE559 Deep Learning, EPFL.



This should be avoided at all costs!

Credits: Francois Fleuret, EE559 Deep Learning, EPFL.



Instead, keep a separate validation set for tuning the hyper-parameters.

Credits: Francois Fleuret, EE559 Deep Learning, EPFL.

Bias-variance decomposition

Consider a fixed point x and the prediction $\hat{Y} = f_*^{\mathbf{d}}(x)$ of the empirical risk minimizer at x.

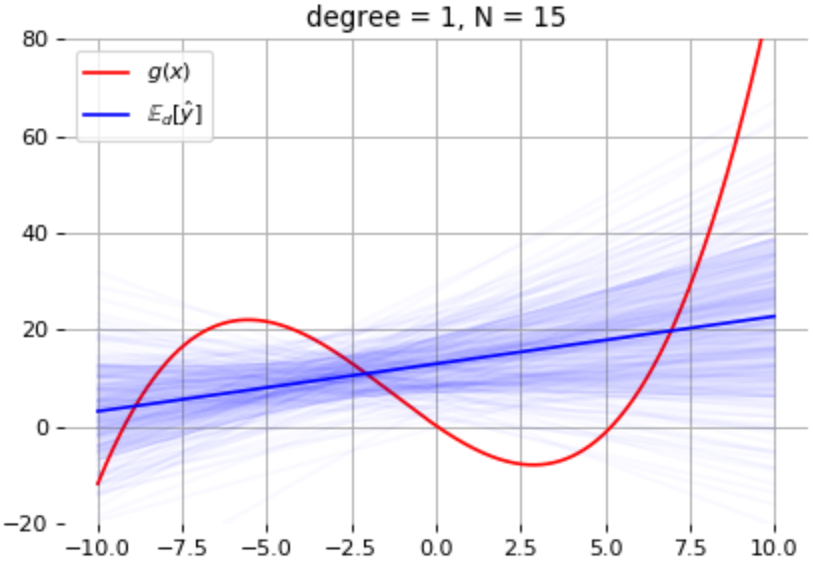
Then the local expected risk of $f_*^{\mathbf{d}}$ is

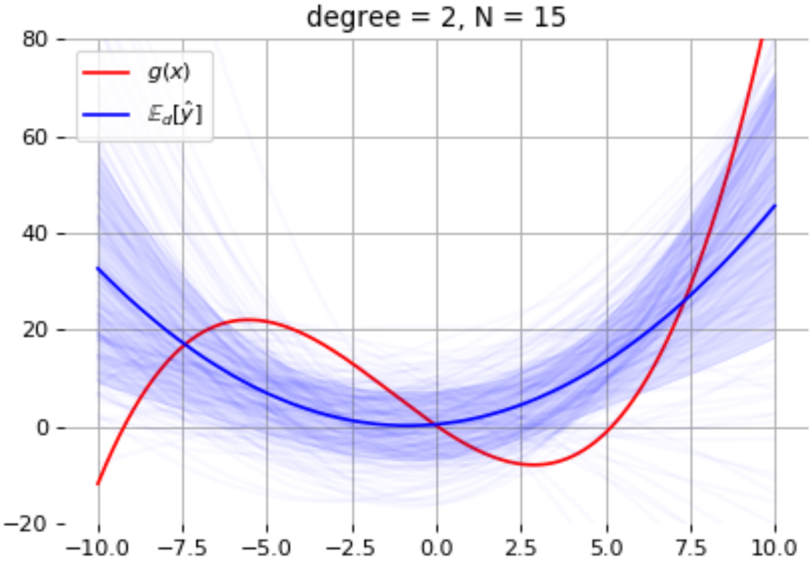
$$\begin{aligned}
R(f_*^{\mathbf{d}}|x) &= \mathbb{E}_{y\sim p_{Y|x}}\left[(y - f_*^{\mathbf{d}}(x))^2\right] \\
&= \mathbb{E}_{y\sim p_{Y|x}}\left[(y - f_B(x) + f_B(x) - f_*^{\mathbf{d}}(x))^2\right] \\
&= \mathbb{E}_{y\sim p_{Y|x}}\left[(y - f_B(x))^2\right] + \mathbb{E}_{y\sim p_{Y|x}}\left[(f_B(x) - f_*^{\mathbf{d}}(x))^2\right] \\
&= R(f_B|x) + (f_B(x) - f_*^{\mathbf{d}}(x))^2
\end{aligned}$$

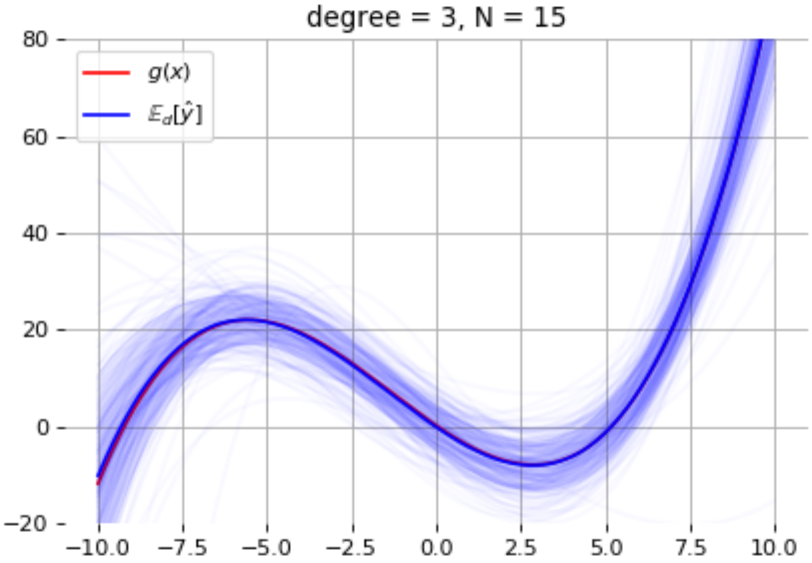
where

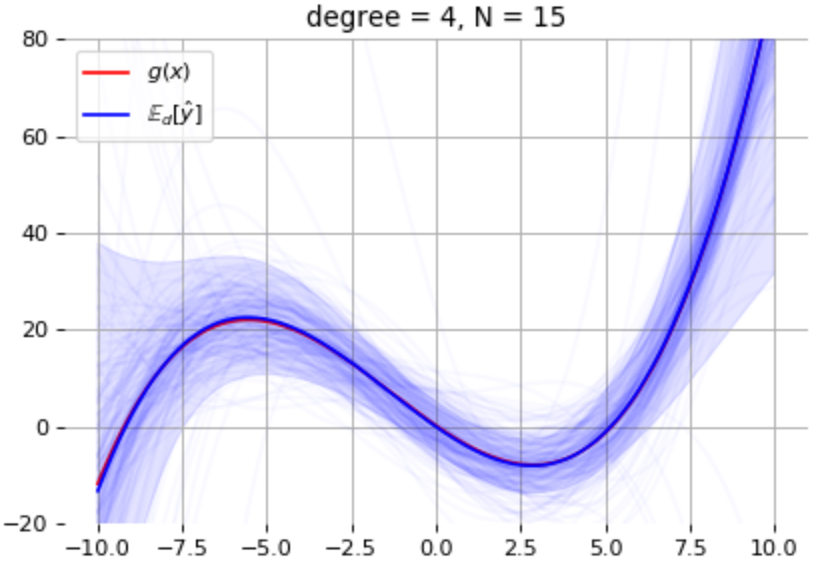
- $R(f_B|x)$ is the local expected risk of the Bayes model. This term cannot be reduced.
- $(f_B(x) - f_*^{\mathbf{d}}(x))^2$ represents the discrepancy between f_B and $f_*^{\mathbf{d}}$.

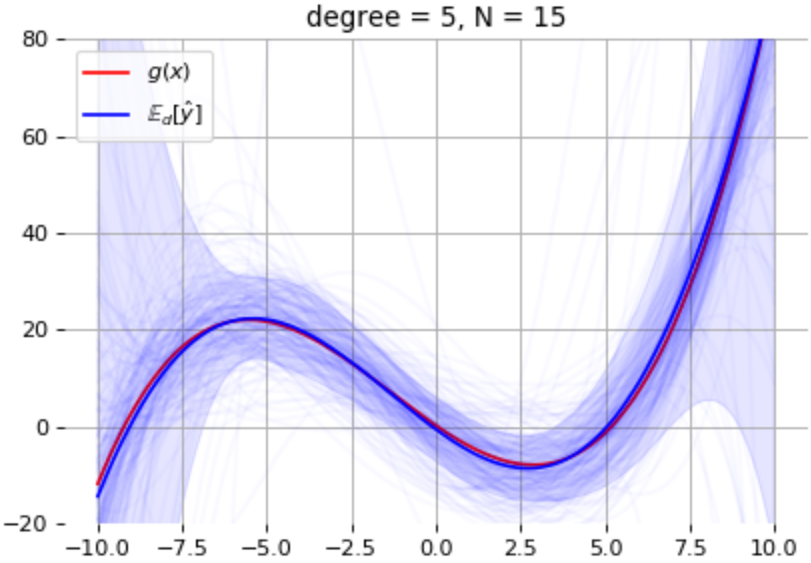
If $\mathbf{d} \sim p_{X,Y}$ is itself considered as a random variable, then $f_*^{\mathbf{d}}$ is also a random variable, along with its predictions $\hat{Y}$.


degree = 1, N = 15
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
$g(x)$
$\mathbb{E}_d[\hat{y}]$


degree = 2, N = 15
$g(x)$
$\mathbb{E}_d[\hat{y}]$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


degree = 3, N = 15
$g(x)$
$\mathbb{E}_d[\hat{y}]$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


degree = 4, N = 15
$g(x)$
$\mathbb{E}_d[\hat{y}]$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


degree = 5, N = 15
$g(x)$
$\mathbb{E}_d[\hat{y}]$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Formally, the expected local expected risk yields to:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{d}}\left[R(f_*^{\mathbf{d}}|x)\right] \\
&= \mathbb{E}_{\mathbf{d}}\left[R(f_B|x) + (f_B(x) - f_*^{\mathbf{d}}(x))^2\right] \\
&= R(f_B|x) + \mathbb{E}_{\mathbf{d}}\left[(f_B(x) - f_*^{\mathbf{d}}(x))^2\right] \\
&= \underbrace{R(f_B|x)}_{\text{noise}(x)} + \underbrace{(f_B(x) - \mathbb{E}_{\mathbf{d}}\left[f_*^{\mathbf{d}}(x)\right])^2}_{\text{bias}^2(x)} + \underbrace{\mathbb{E}_{\mathbf{d}}\left[(\mathbb{E}_{\mathbf{d}}\left[f_*^{\mathbf{d}}(x)\right] - f_*^{\mathbf{d}}(x))^2\right]}_{\text{var}(x)}
\end{aligned}
$$

This decomposition is known as the bias-variance decomposition.

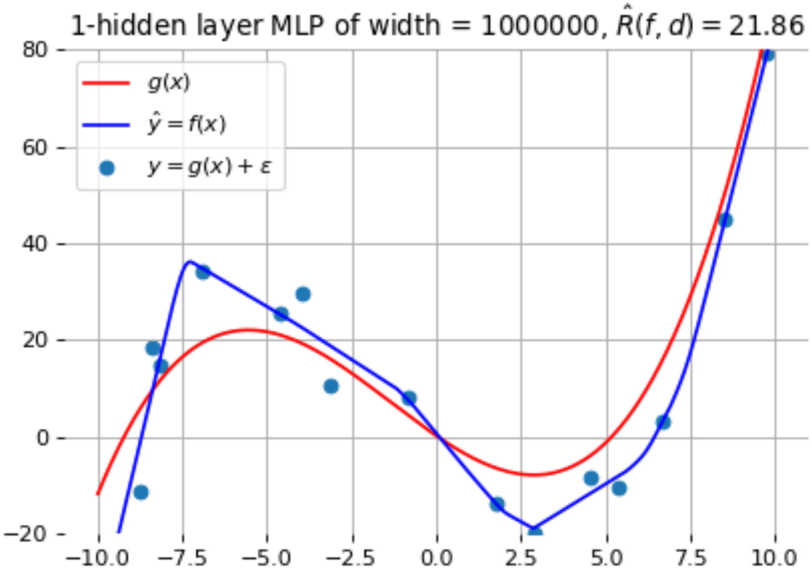
- The noise term quantifies the irreducible part of the expected risk.
- The bias term measures the discrepancy between the average model and the Bayes model.
- The variance term quantities the variability of the predictions.

Bias-variance trade-off

- Reducing the capacity makes $f_*^{\mathbf{d}}$ fit the data less on average, which increases the bias term.
- Increasing the capacity makes $f_*^{\mathbf{d}}$ vary a lot with the training data, which increases the variance term.

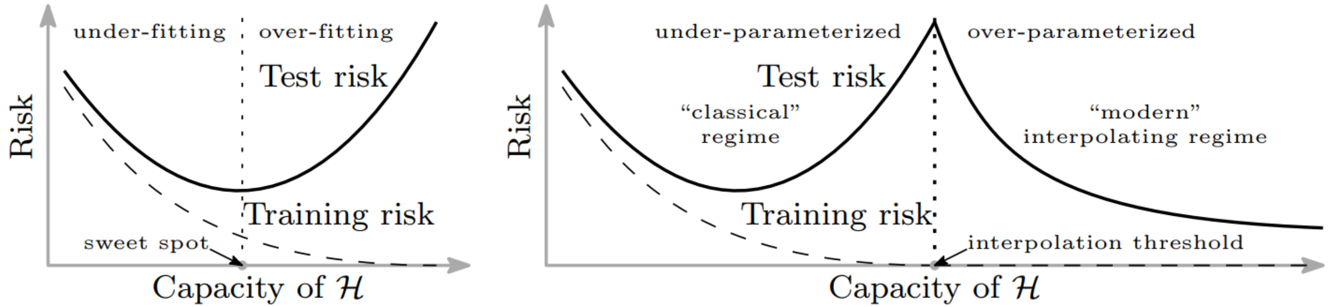
Credits: Francois Fleuret, EE559 Deep Learning, EPFL.

What about a neural network with **millions** of parameters?


1-hidden layer MLP of width = 1000000, $\hat{R}(f, d) = 21.86$
$g(x)$
$\hat{y} = f(x)$
$y = g(x) + \varepsilon$
80
60
40
20
0
−20
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

While classical learning theory suggests that large capacity models should overfit the data, **deep neural networks with millions of parameters often generalize well in practice.**

In fact, increasing the number of parameters can keep decreasing the test error, even after the training error has reached zero!



(a) **(b)**

Figure 1: **Curves for training risk (dashed line) and test risk (solid line).** (**a**) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (**b**) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

Credits: Belkin et al, 2018.

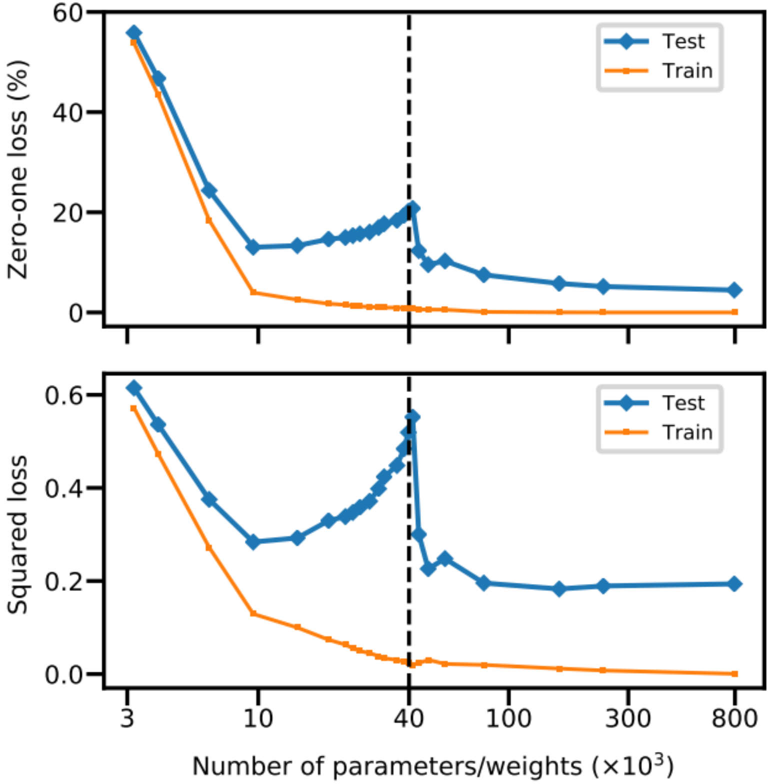


Figure 4: **Double descent risk curve for fully connected neural network on MNIST.** Training and test risks of network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d+1) \cdot H + (H+1) \cdot K$. The interpolation threshold (black dotted line) is observed at $n \cdot K$.

Credits: Belkin et al, 2018.

Effective hypothesis space

Because of the way they are trained, the **effective** hypothesis space of deep neural networks has a much lower capacity than what could be expected from the number of parameters alone.

- The optimization algorithm (e.g., SGD) introduces an implicit bias towards "simpler" models within the large hypothesis space defined by the neural network architecture. Not all functions that can be represented by the network are equally likely to be found by the optimization process!
- Regularization techniques (e.g., dropout, weight decay) further constrain the effective hypothesis space.

The end.