# CS3244 : Machine Learning

Semester 1 2023/24

## Lecture 5 : Bias and Variance


Xavier Bresson

https://twitter.com/xbresson



Department of Computer Science

National University of Singapore (NUS)


# Material used for preparation

- Prof Kilian Weinberger, CS4780 Cornell, Machine Learning, 2018
  - https://www.cs.cornell.edu/courses/cs4780/2018fa
- Prof Min-Yen Kan, CS3244 NUS, Machine Learning, 2022
  - https://knmnyn.github.io/cs3244-2210
- Prof Xavier Bresson, CS6208 NUS, Advanced Topics in Artificial Intelligence, 2023

# Outline

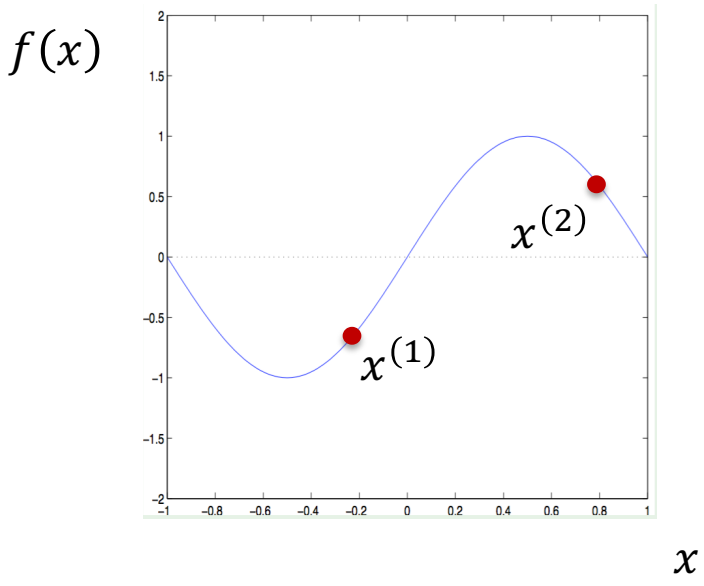
- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

# Outline

- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

# An example of bias and variance

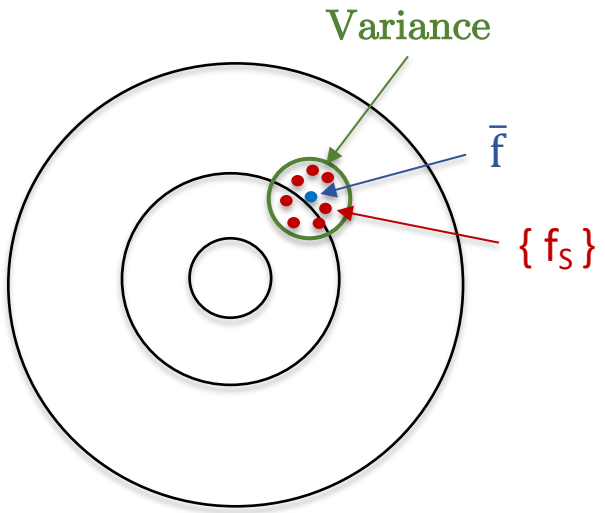
- We consider a simple example to compute the bias and the variance.
- Example : Predict the sine function
  - Regression task
  - Training set : 2 data points, $(\mathbf{X}, \boldsymbol{y}) = \left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right)$, i.e. n=2 and $x^{(k)} \in \mathrm{R}$, d=1
  - Two hypothesis spaces :
    - $\mathcal{H}_0 : \mathrm{f}_\theta(x) = \theta_0$
    - $\mathcal{H}_1 : \mathrm{f}_\theta(x) = \theta_0 + \theta_1 x_1$
- Q: Which predictive class of functions is better, $\mathcal{H}_0$ or $\mathcal{H}_1$ ?
  - Hypothesis $\mathcal{H}_1$ seems better because the model is more expressive than $\mathcal{H}_0$.

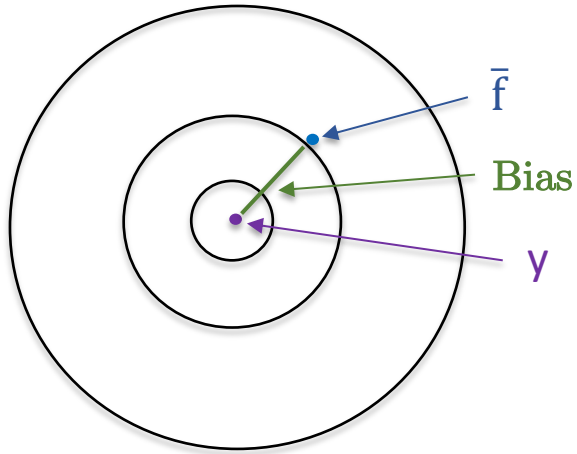



$f(x) = \sin(\pi x) : [-1,1] \to \mathbb{R}$

# An example of bias and variance

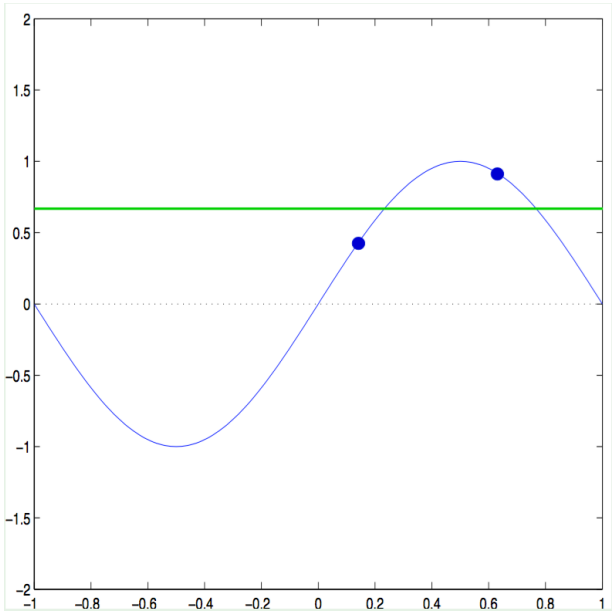
- How to define a good hypothesis?
  - When an hypothesis has low bias and low variance (week 3).
  - Variance : The variability of the prediction model.
  - Bias : The difference between the average prediction model and the true regression value.



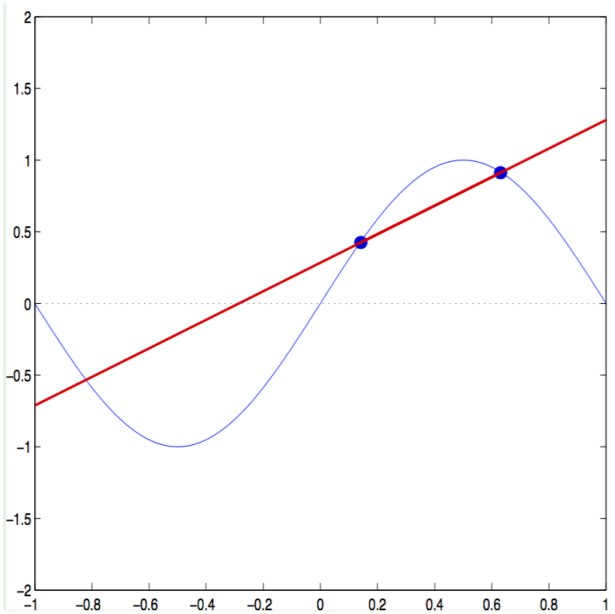



# An example of bias and variance

- Example of predictive function for each hypothesis given a training set :



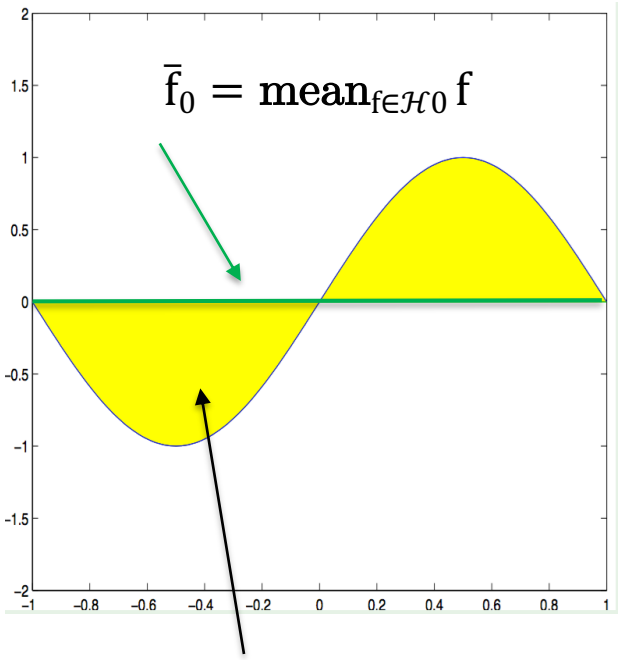

$\mathcal{H}_0 : \mathrm{f}_\theta(x) = \theta_0$



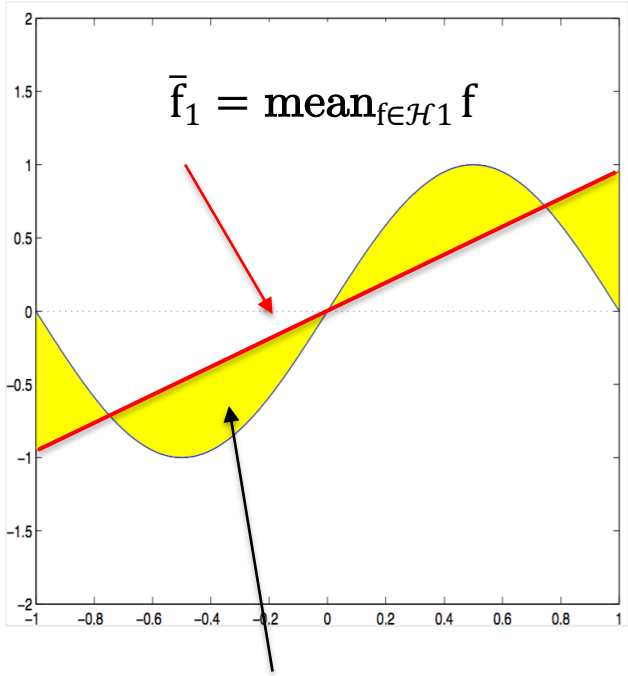

$\mathcal{H}_1 : \mathrm{f}_\theta(x) = \theta_0 + \theta_1 x_1$

# An example of bias and variance

- $\text{Bias}(\mathcal{H}) = \text{L}_{\text{test}}\,(\,\text{mean}_{\text{f}\in\mathcal{H}}\,\text{f}\,)$



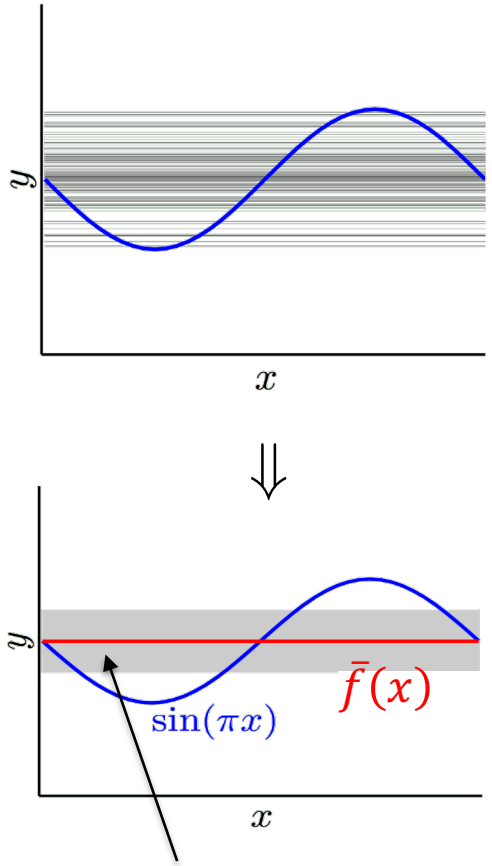

$\text{Bias}(\mathcal{H}_0) = 0.50$



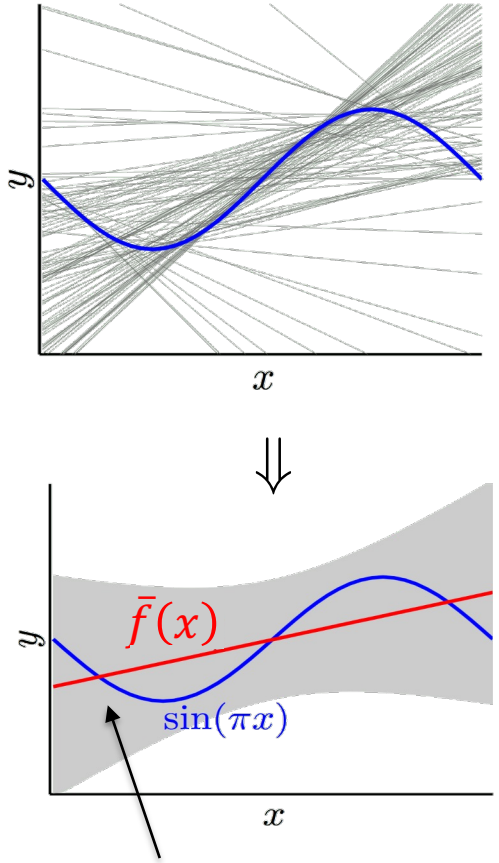

$\text{Bias}(\mathcal{H}_1) = 0.20$

# An example of bias and variance

- $\text{Variance}(\mathcal{H}) = \text{mean}_{f \in \mathcal{H}} (f - \text{mean}_{f \in \mathcal{H}} f)^2$
  - Formally, the mean is replaced by $\mathbb{E}$ (expectation operator)



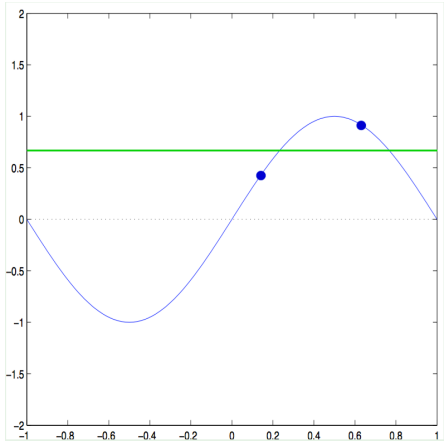

$\text{Var}(\mathcal{H}_0) = 0.25$




$\text{Var}(\mathcal{H}_1) = 1.69$

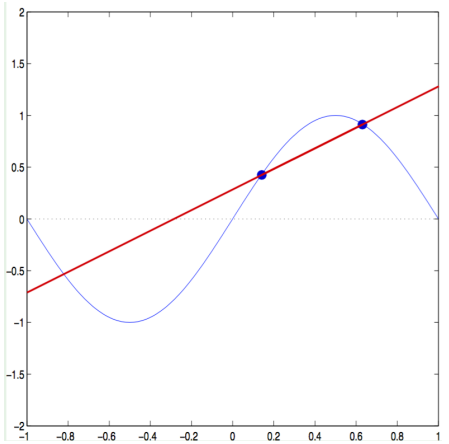
# An example of bias and variance

- How to define a good hypothesis?
  - When a hypothesis has low bias and low variance.
    - $\mathcal{H}_0 : \mathrm{f}_\theta(x) = \theta_0 \quad \Rightarrow$ Bias + Var = 0.50 + 0.25 = 0.75 ✓
    - $\mathcal{H}_1 : \mathrm{f}_\theta(x) = \theta_0 + \theta_1 x_1 \Rightarrow$ Bias + Var = 0.20 + 1.69 = 1.89
- In the case of low-data, it is not the model with the most expressivity that performs the best.



$\mathcal{H}_0 : \mathrm{f}_\theta(x) = \theta_0$

✓



$\mathcal{H}_1 : \mathrm{f}_\theta(x) = \theta_0 + \theta_1 x_1$

# Outline

- An example of bias and variance
- **Random variable**
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

# Random variable

- A random variable $(X,\Omega,P)$ is a variable X that can have different values $x \in \Omega$ controlled by a probability distribution, $P(X = x)$.
  - For example, a coin can be represented by a random variable X.
  - The set $\Omega$ of all outcomes of the coin is $\Omega = \{ \text{Head} , \text{Tail} \}$.
  - The probabilities of the outcomes are $P(X = H) = 1/2$ and $P(X = T) = 1/2$ if the coin is fair, i.e. not biased to any particular side.
- Expected value of X : Average value over all possible outcomes of a random variable X weighted by the probability of the outcome, i.e. $\mathbb{E}(X) = \sum_{x \in X} x \Pr(X = x)$.
  - Example: Let X be the outcome of a fair dice roll. Then, the expected value is $\mathbb{E}(X)=1/6\ (1+2+3+4+5+6)=3.5$.
- Variance of X : Variation of a random variable from the expected value, i.e. $\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}(X))^2]$.
  - Example: Let X be the outcome of a fair dice roll. Then, the variance is $\mathrm{Var}(X)=1/6\ .\ ( (1-3.5)^2+...+(6-3.5)^2 )=35/12$.

# Outline

- An example of bias and variance
- Random variable
- **Bias, variance, noise decomposition**
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

# Bias-variance-noise decomposition

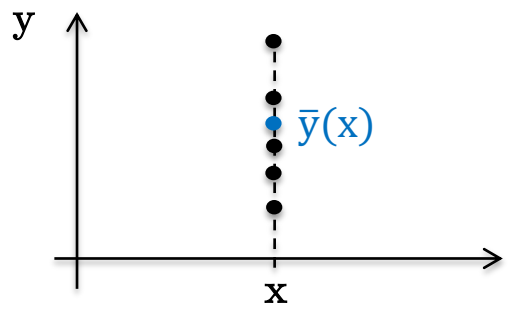
- We consider a training dataset $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$ of n data points sampled i.i.d. from a data-label distribution P(X,Y), where X and Y are random variables.
- Our goal is to decompose the generalization error of a learner (classifier or regressor) into three fundamental terms, i.e. bias, variance and data noise.
- Let us assume the regression task, i.e. $y \in R$, with the MSE loss, i.e. $L = (f(x)-y)^2$ (decomposition is easier to prove for regression than classification).
- Let us first define the following quantities :
  - Expected label given x
  - Expected test error given $f_D$
  - Expected learner given algorithm A
  - Expected test error given algorithm A

# Bias-variance-noise decomposition

- For any any given input $x \in R^d$, there may not exist a unique label y, but multiple labels y.
- For example, if your input x describes house features, e.g. #bedrooms, square footage, etc, and label y its price, it is likely that 2 houses with identical features can be sold with different prices. Hence, for any given feature vector x, there is a distribution P(y|x) over possible labels y.
- We therefore define the following expected label given $x \in R^d$ :

$$\bar{y}(x) = \mathbb{E}_{y|x \sim P(y|x)}[Y] = \int_y y \; P(y|x) dy$$

- The expected label is the label we would expect to obtain given a feature vector x.



# Bias-variance-noise decomposition

- Given a training set D of n inputs i.i.d. sampled from the data-label distribution $P \sim P^n$, we aim at learning a function (classifier or regressor) that solves a machine learning task.
  - This is known as the training process.
- Formally, for a given training set D and an algorithm A, we can compute the learner $f_D = A(D)$.
- Given the trained learner $f_D = A(D)$, we define the expected test error or generalization error as follows :

$$\mathbb{E}_{x,y\sim P(x,y)}\left[(f_D(x)-y)^2\right] = \int_x \int_y (f_D(x)-y)^2 \; P(x,y)dxdy$$

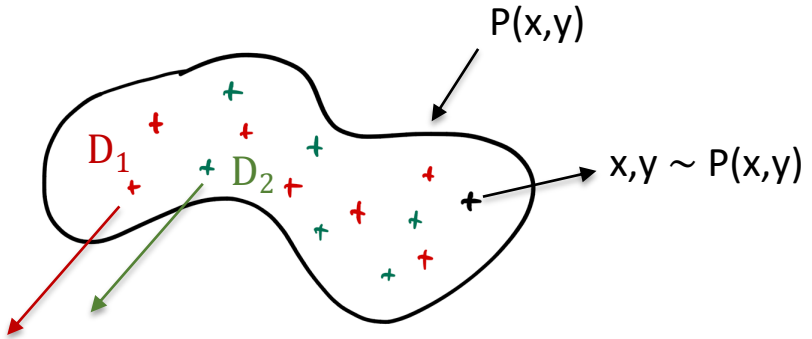
  where (x,y) are test data points and D consists of training data points.
- Note that any other supervised loss function can be used, e.g. regression and classification losses.

# Bias-variance-noise decomposition

- The expected test error is defined for a specific training set D.
- This implies that the trained learner $f_D = A(D)$ will be different when it is computed with a different training set D.
- We can compute the expected learner given an algorithm A as :

$$\bar{f} = \mathbb{E}_{D \sim P^n}[f_D] = \int_D f_D \; P(D) dD$$

where P(D) is the probability of drawing D from $P^n$.




$f_{D_1}(x),\ p(D_1)$

$f_{D_2}(x),\ p(D_2)$

$\vdots$

$\bar{f}(x) = \mathbb{E}_{D \sim Pn(x,y)}\,(\,f_D\,(x)\,)$

# Bias-variance-noise decomposition

- Function $\bar{f}$ is a weighted average over all trained functions, i.e. the mean predictor.
  - It is independent of D as it was integrated over all possible training sets D.
  - Function $\bar{f}$ can be approximated as $\bar{f}(x) \approx \frac{1}{K}\sum_{k=1}^{K} f_{\mathcal{D}_k}(x)$ but collecting multiple $D_k$ is usually never done (time consuming). New data points are generally added to the original D.

# Bias-variance-noise decomposition

- Finally, we can also compute the expected test error for an algorithm A w.r.t. all training sets D sampled from $P^n$ as :

$$\mathbb{E}_{x,y\sim P(x,y),\ D\sim P^n}\left[(f_D(x)-y)^2\right] = \int_D \int_x \int_y (f_D(x)-y)^2\ P(x,y)P(D)dxdydD$$

  where (x,y) are test data points and D consisting of training data points.

- This error evaluates the quality of a machine learning algorithm A given a data distribution P.
- We will decompose this generalization error into quantities called bias, variance and data noise.

# Bias-variance-noise decomposition

- Decomposition of expected test error :

$$\mathbb{E}_{x,y,D}\left[(f_D(x)-y)^2\right] = \mathbb{E}_{x,y,D}\left[\left(\left((f_D(x)-\underbrace{\bar{f}(x)}_{\text{Do not change anything}})+(\bar{f}(x)-y)\right)\right)^2\right]$$

$$= \mathbb{E}_{x,D}\left[(f_D(x)-\bar{f}(x))^2\right] + \mathbb{E}_{x,y}\left[(\bar{f}(x)-y)^2\right] + 2\,\underbrace{\mathbb{E}_{x,y,D}\left[(f_D(x)-\bar{f}(x))(\bar{f}(x)-y)\right]}$$

$$= \mathbb{E}_{x,D}\left[(f_D(x)-\bar{f}(x))^2\right] + \underbrace{\mathbb{E}_{x,y}\left[(\bar{f}(x)-y)^2\right]}_{\text{Let us develop this term in the next slide.}}$$

$$\begin{aligned}\mathbb{E}_{x,y,D}\left[(f_D(x)-\bar{f}(x))(\bar{f}(x)-y)\right] &= \mathbb{E}_{x,y}\left[\mathbb{E}_D\left[(f_D(x)-\bar{f}(x))(\bar{f}(x)-y)\right]\right]\\ &= \mathbb{E}_{x,y}\left[(\mathbb{E}_D\left[f_D(x)\right]-\bar{f}(x))(\bar{f}(x)-y)\right]\\ &= \mathbb{E}_{x,y}\left[(\bar{f}(x)-\bar{f}(x))(\bar{f}(x)-y)\right]\\ &= 0\end{aligned}$$

# Bias-variance-noise decomposition

- Decomposition of second term :

$$\mathbb{E}_{x,y}\left[(\bar{f}(x)-y)^2\right] = \mathbb{E}_{x,y}\left[\left((\bar{f}(x)-\underbrace{\bar{y}(x)}_{\text{Do not change anything}})+(\bar{y}(x)-y)\right)^2\right]$$

$$= \mathbb{E}_x\left[(\bar{f}(x)-\bar{y}(x))^2\right] + \mathbb{E}_{x,y}\left[(\bar{y}(x)-y)^2\right] + 2\,\underbrace{\mathbb{E}_{x,y}\left[(\bar{f}(x)-\bar{y}(x))(\bar{y}(x)-y)\right]}$$

$$= \mathbb{E}_x\left[(\bar{f}(x)-\bar{y}(x))^2\right] + \mathbb{E}_{x,y}\left[(\bar{y}(x)-y)^2\right]$$

$$\begin{aligned}
\mathbb{E}_{x,y}\left[(\bar{f}(x)-\bar{y}(x))(\bar{y}(x)-y)\right] &= \mathbb{E}_x\left[\mathbb{E}_{y|x}\left[(\bar{f}(x)-\bar{y}(x))(\bar{y}(x)-y)\right]\right] \\
&= \mathbb{E}_x\left[(\bar{f}(x)-\bar{y}(x))\mathbb{E}_{y|x}\left[(\bar{y}(x)-y)\right]\right] \\
&= \mathbb{E}_x\left[(\bar{f}(x)-\bar{y}(x))(\bar{y}(x)-\mathbb{E}_{y|x}\left[y\right])\right] \\
&= \mathbb{E}_x\left[(\bar{f}(x)-\bar{y}(x))\underbrace{(\bar{y}(x)-\bar{y}(x))}_{0}\right] \\
&= 0
\end{aligned}$$

# Bias-variance-noise decomposition

- Decomposition of expected test error :

$$\underbrace{\mathbb{E}_{x,y,D}\left[(f_D(x)-y)^2\right]}_{\text{Expected test error}} = \underbrace{\mathbb{E}_{x,D}\left[(f_D(x)-\bar{f}(x))^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_{x}\left[(\bar{f}(x)-\bar{y}(x))^2\right]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{x,y}\left[(\bar{y}(x)-y)^2\right]}_{\text{Data noise}}$$

- This is the most fundamental equation of supervised machine learning.
- This is also called the bias-variance-error trade-off.

# Outline

- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- **Fundamental equation of supervised machine learning**
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

# Fundamental equation of supervised learning

- Decomposition of expected test error :

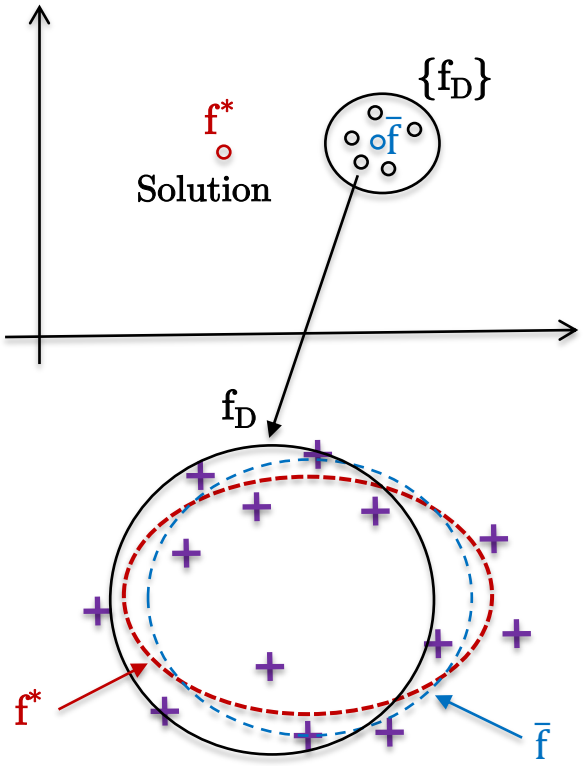
$$\underbrace{\mathbb{E}_{x,y,D}\left[(f_D(x)-y)^2\right]}_{\text{Expected test error}} = \underbrace{\mathbb{E}_{x,D}\left[(f_D(x)-\bar{f}(x))^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_{x}\left[(\bar{f}(x)-\bar{y}(x))^2\right]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{x,y}\left[(\bar{y}(x)-y)^2\right]}_{\text{Data noise}}$$

# Fundamental equation of supervised learning

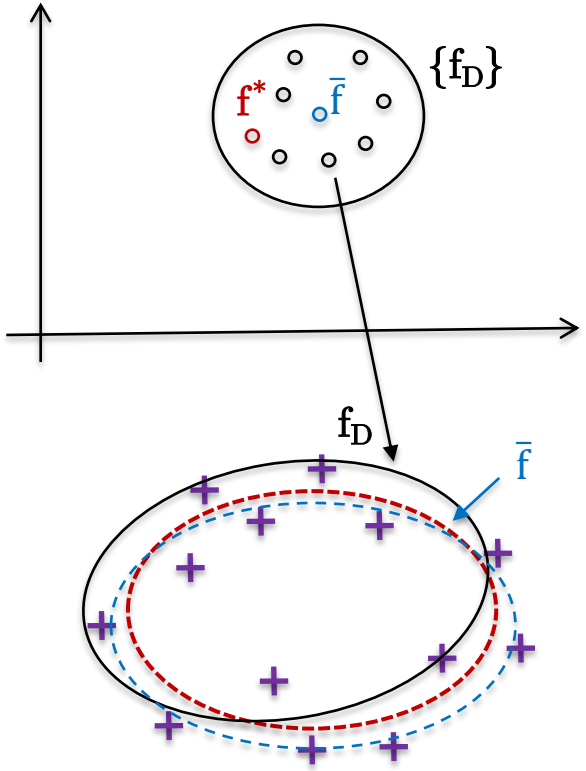
- **Variance :** $\mathrm{Var}(f) = \mathbb{E}_{x,D}\left[(f_D(x) - \bar{f}(x))^2\right]$
- Given an algorithm A, the variance captures how much the learner changes when it is computed on different training sets.
- The variance indicates the expressivity of algorithm A.
  - Complex algorithms have high variance and simple algorithms have low variance.
- Over-fitting : A learner $f_D(x)$ which has zero prediction error on a training set D.
  - A learner $f_D(x)$ that is capable of overfitting various training sets must exhibit significant variability, indicating a complex algorithm.
- Under-fitting : A learner $f_D(x)$ which is not able to predict correctly a training set D.
  - A learner $f_D(x)$ that exhibits underfitting has limited variability, signifying a simple algorithm.

# Fundamental equation of supervised learning

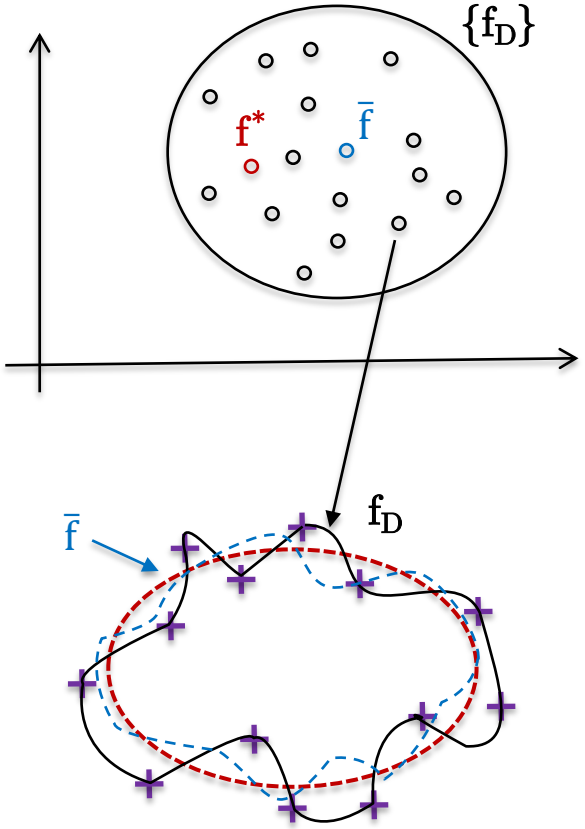
- **Variance :** $\mathrm{Var}(f) = \mathbb{E}_{x,D}\left[(f_D(x) - \bar{f}(x))^2\right]$




Under-fitting
Simple model (circles)
Small variance



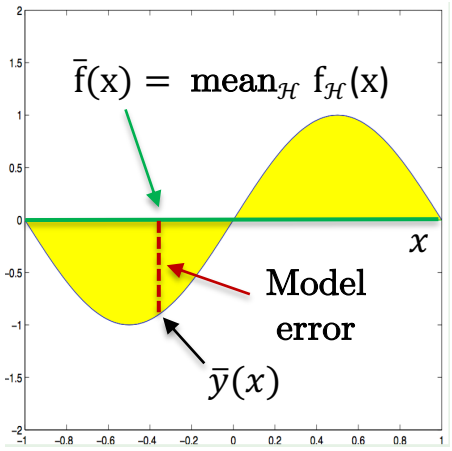

Right-fitting
Right model (ellipsoids)
Right variance



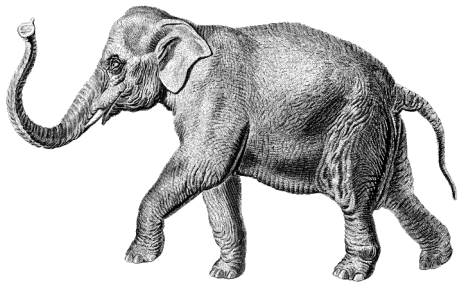

Over-fitting
Complex model (polynomials)
Large variance

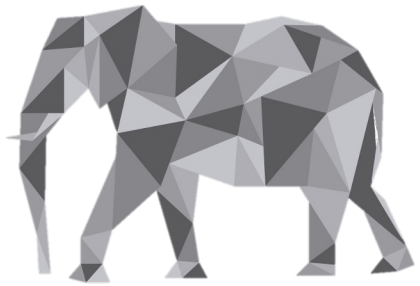
# Fundamental equation of supervised learning

- **Bias :** $\text{Bias}(f, y) = \mathbb{E}_x\left[(\bar{f}(x) - \bar{y}(x))^2\right]$
- Given an algorithm A, the bias is the intrinsic error of a learner with infinite training data.
- Any learner has some bias toward a particular class of solutions, e.g. linear models with hyper-plans or decision trees with piece-constant functions.
- Bias is inherent to the model and independent of data.
- Bias is a.k.a. the model error, i.e. the lack of capacity of the model to perfectly capture the data distribution.



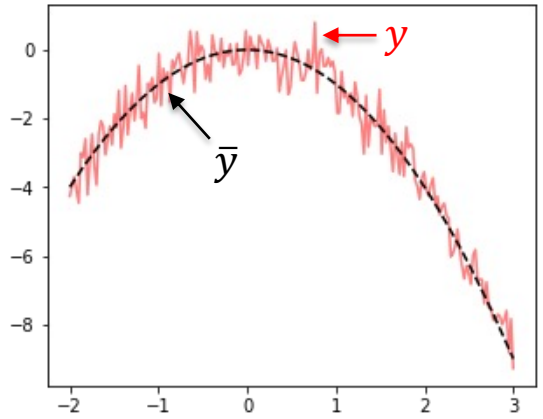

$\text{Bias}(\mathcal{H}) = 0.50$

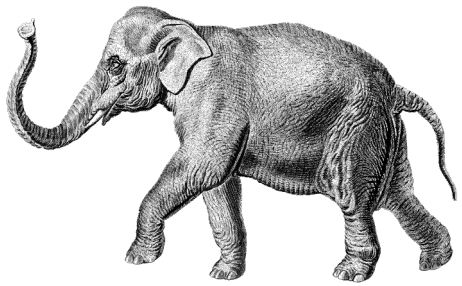


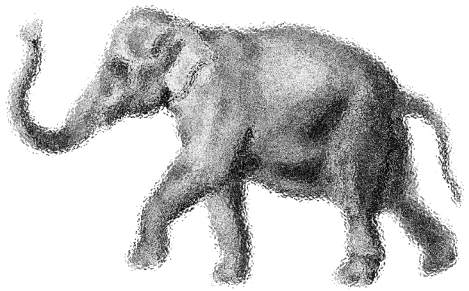


# Fundamental equation of supervised learning

- **Data-intrinsic noise :** $\text{Noise}(y) = \mathbb{E}_{x,y}\left[(\bar{y}(x) - y)^2\right]$
- This error measures the ambiguity inherent to the data distribution and feature representation.
- It is impossible to get rid of this noise, it is a part of data.
- Noise is often modeled as a stochastic process, that is added to the "clean" data, i.e. $y = \bar{y} + \varepsilon$.





# Outline

- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

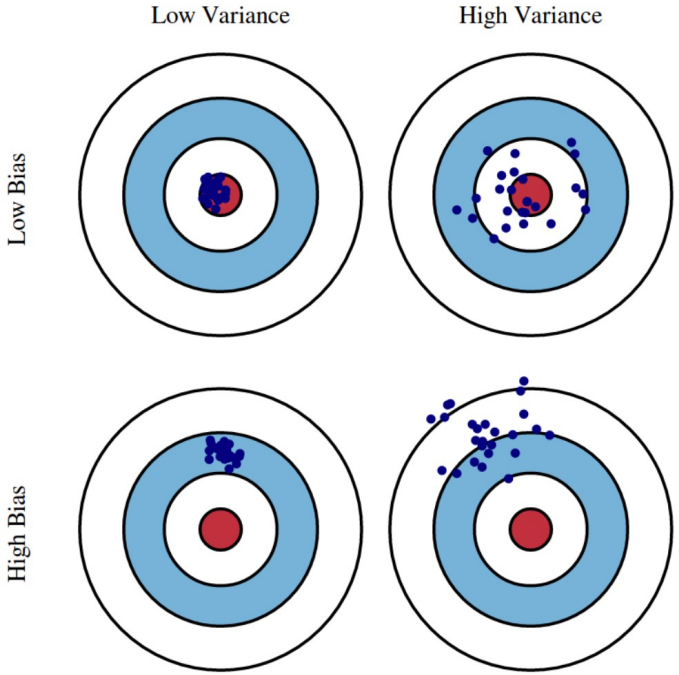
# Bias-variance trade-off

- Quality of predictive models are evaluated by the bias-variance trade-off.
- For example, consider a model that can predict the red center of the target below.

Low variance and low bias
The perfect models

Low variance and high bias
The model favors some solutions, far from the true ones.

Boosting helps reducing bias (week 3)




Bagging helps reducing variance (week 3)

High variance and low bias
The model is able to find the correct solution on average.

For examples, decision tree with large depth and deep learning.

High variance and high bias
The worst models
The model has not only bad bias but also large variance.

For examples, decision tree with small depth and linear model.

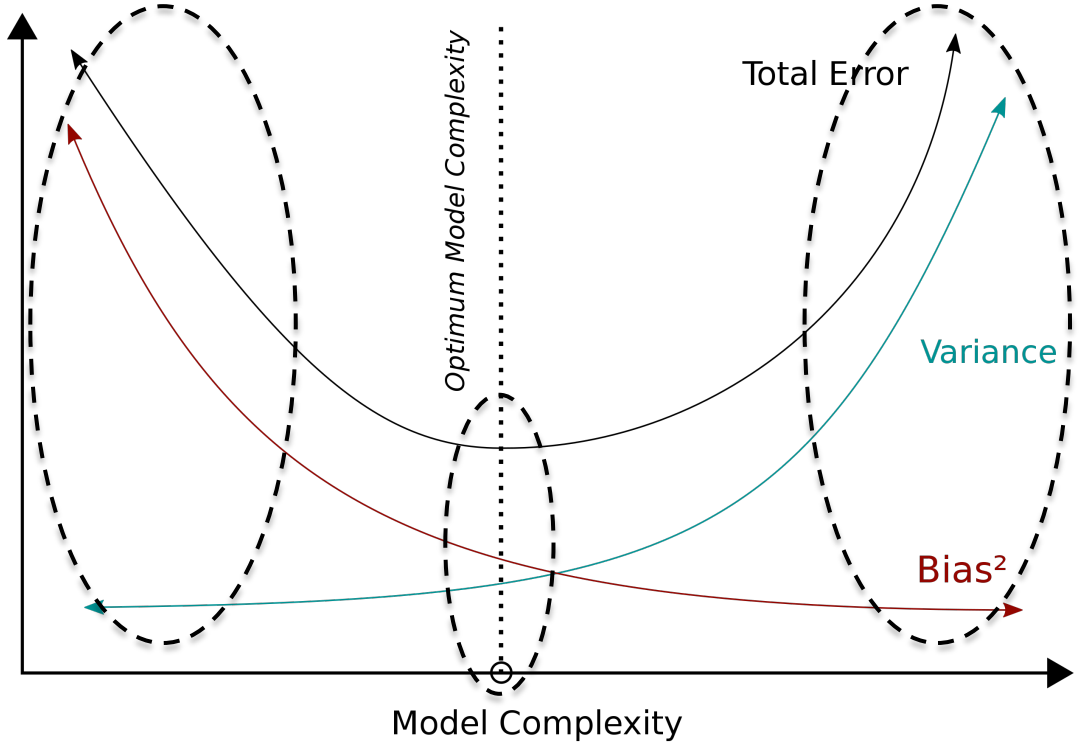
# Bias-variance trade-off

- Test error vs. model complexity

Low complexity

Simple models e.g. linear ones have low variance but high bias, making test error high.

Under-fitting




High complexity

Complex models e.g. deep learning have low bias but high variance, making test error high.

Over-fitting

Right complexity

Models that minimizes both variance and bias, making test error low.

Right-fitting

# Bias-variance trade-off

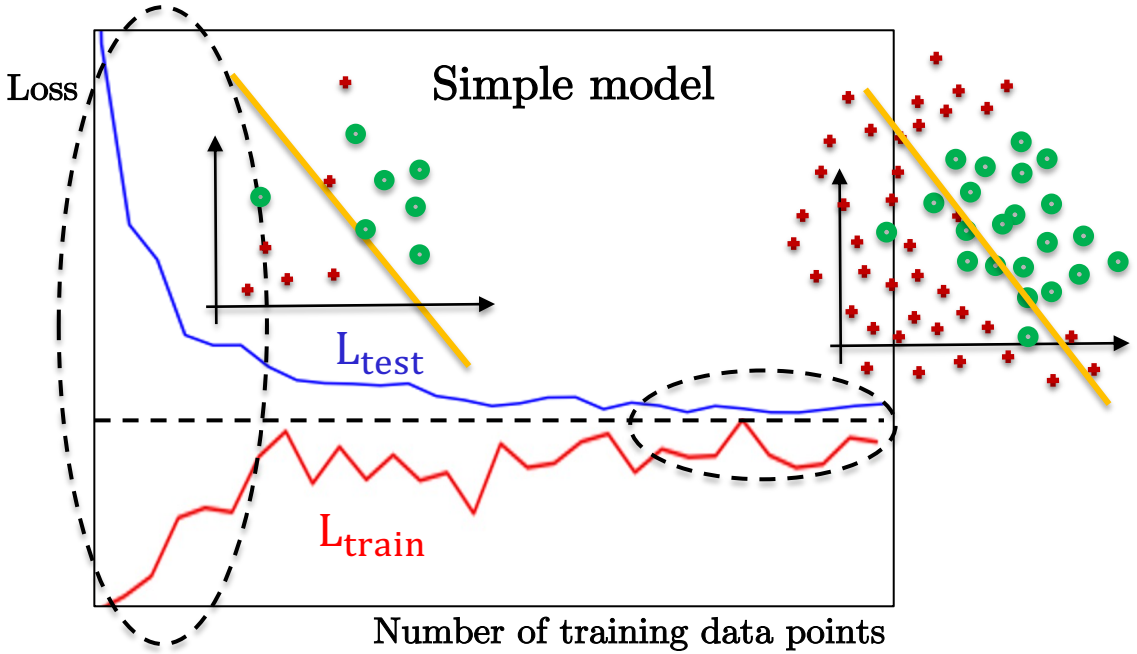
- How to reduce high variance?
  - Reduce model complexity : Lower model expressivity or use regularizers (week 6)
  - Remove non-informative/bad data features : E.g. house features such house color, back door, etc but challenging to decide bad features (not done in practice).
  - Bagging (week 3) : Averaging weak high-variance learners produces strong learner with low variance (requires fast computation of weak learners in practice).
- How to reduce high bias?
  - Increase model complexity : More expressive models (deep learning)
  - Add more informative data features : E.g. house features such as storage space, garden, security system, etc (effective but time and money consuming).
  - Boosting (week 3) : Adding weak high-bias learners produces strong learner with low bias (requires fast computation of weak learners in practice s.a. small-depth decision trees).
- Add more training data or use data augmentation to reduce both bias and variance.

# Outline

- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- Conclusion

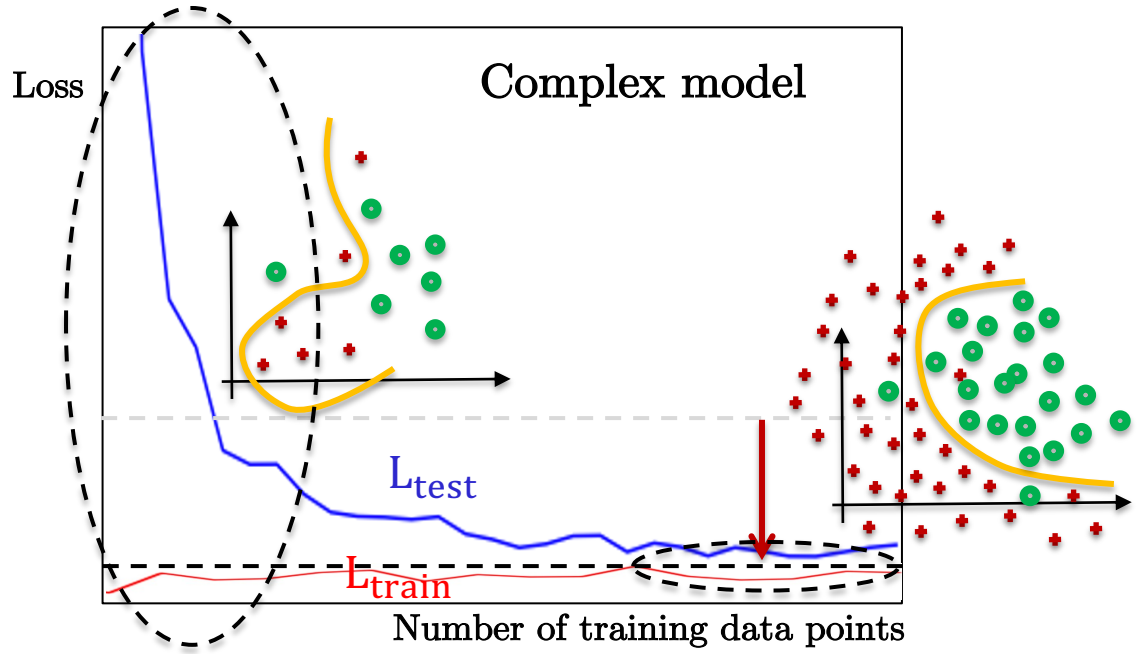
# Best-case scenario

- Test/train errors vs. number of training data points




Small #data
Low train error and
high test error
(high variance)

Large #data
Loss(train) = Loss(error)
but high error for simple
model (high bias)




Small #data
Low train error and
high test error
(high variance)

Large #data
Loss(train) = Loss(error)
and small error for
complex model (low bias)
Best case !

# Best-case scenario

- The scenario in which supervised learning provides the best results
  - Highly expressive learner $f_D(x)$, such as deep learning models (e.g. Transformers)
  - Infinite number of training data – In practice, the more, the better
- A good example is ChatGPT/GPT3.5-4
  - Number of parameters features $|\boldsymbol{\theta}| = 175B$ (GPT-3)
  - Number of training data $n = 300B$ tokens (GPT-3)
  - (Self-)supervised learning delivers excellent results.

# Best-case scenario

- From a theoretical perspective, what does it mean when the training set D is infinite or very large?
  - When $D \approx \{ x \sim P(X) \}$, it implies that the learner $f_D(x)$ can successfully predict any data point sampled from the probability distribution $P(X)$.
  - This means that $f_D(x)$ has captured the underlying patterns of the data distribution.
- For ChatGPT, which has been trained on a large corpus of (almost) all English texts, the question is whether it represents the best of what can be achieved with supervised learning?
  - In my opinion, yes.
- Then, does it mean that ChatGPT has successfully passed the Turing test, i.e. designing an AI capable of conversation on any topic while remaining indistinguishable from humans?
  - No, learning all linguistic patterns and knowledge is not sufficient to achieve human-level intelligence. Although ChatGPT can address first-order logic tasks, it lacks the capability for real-world complex reasoning, long-term hierarchical planning and it also produces "hallucinations", i.e. making things up.

# Outline

- An example of bias and variance
- Random variable
- Bias, variance, noise decomposition
- Fundamental equation of supervised machine learning
- Understanding bias-variance trade-off
- Best-case scenario
- **Conclusion**

# Conclusion

- Main goal of machine learning is to minimize test/generalization error.
- Test error can be decomposed into three fundamental parts; bias, variance and data noise.
  - Bias reveals the limit of the learner to predict correctly with infinite training data.
  - Variance captures how much the learner can change.
  - Noise is the inherent uncertainty present in data.
- Best case : Expressive model and large training set
- Under-fitting is not an issue in practice : Easy to increase the model expressivity, e.g. from linear to polynomial functions.
- Over-fitting is one of the most important practical problems : Models s.a. deep learning are usually too expressive and overfit easily, which prevents generalization.
  - How to reduce over-fitting for successful generalization? (Week 6)

Questions?