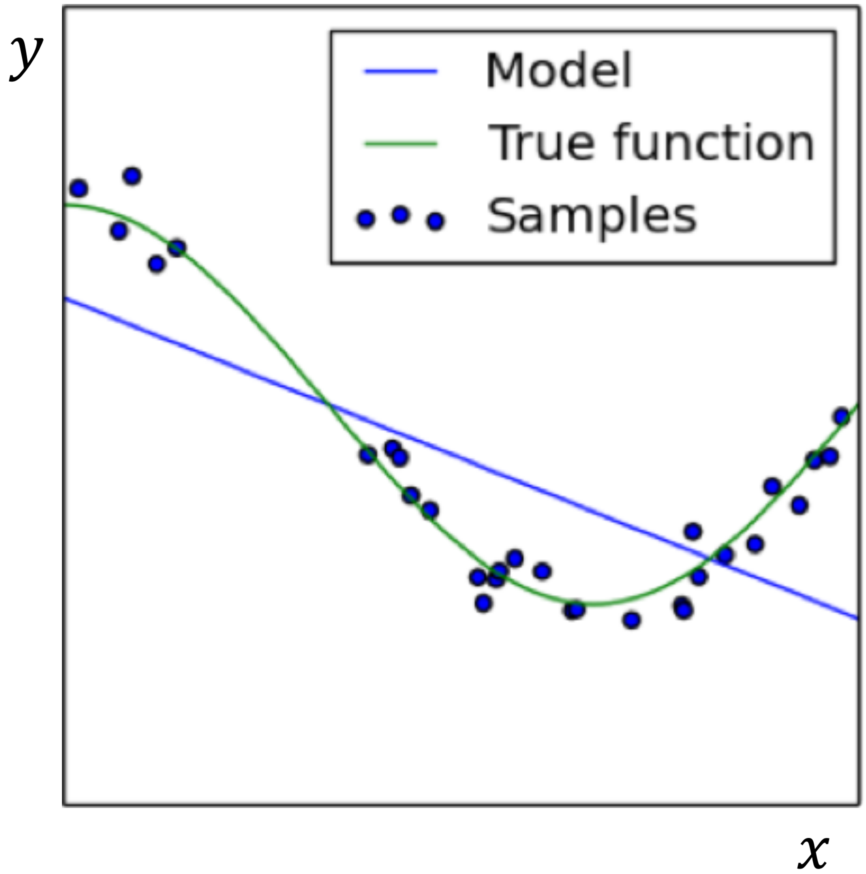
Generalization

- Consider a model with accuracy 80% on training set
- How it will perform on the new data?
- Does the model generalize well?

Underfitting and overfitting example

Training set: $X \subset \mathbb{R}$

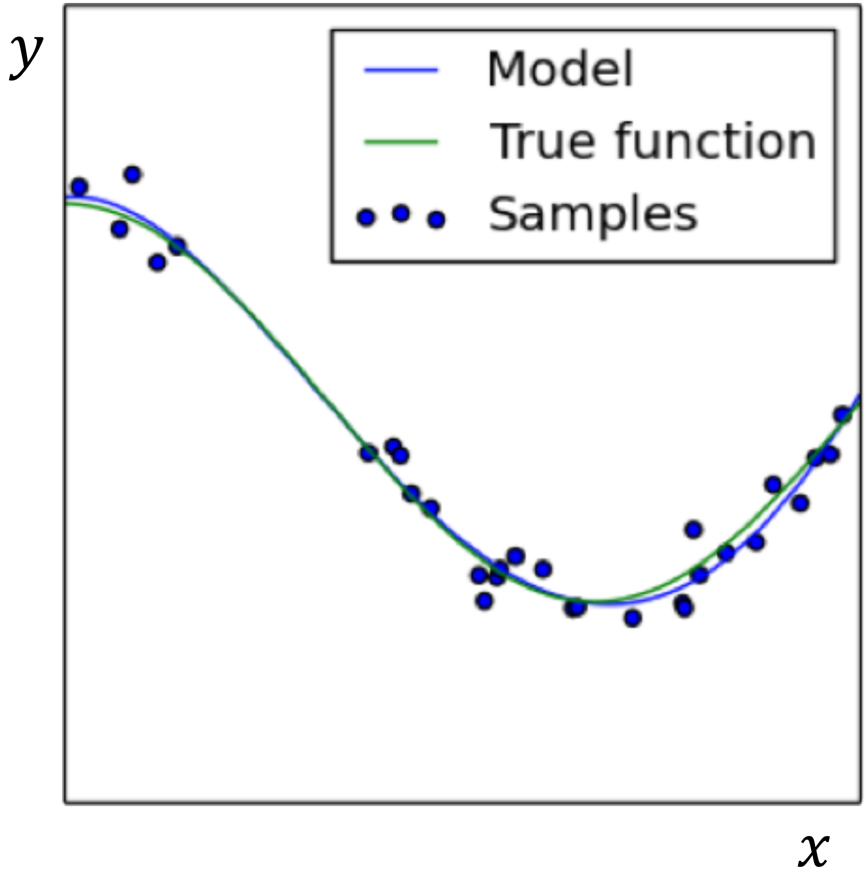
Model: $a(x) = b + w_1 x$



Underfitting and overfitting example

Training set: $X \subset \mathbb{R}$

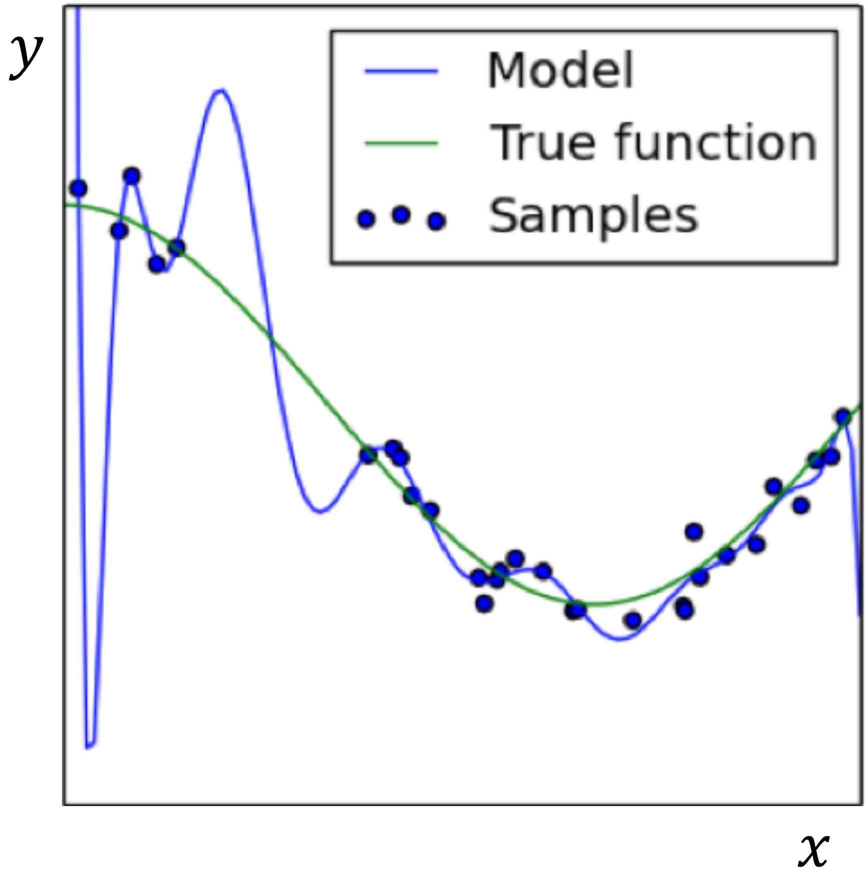
Model: $a(x) = b + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$



Underfitting and overfitting example

Training set: $X \subset \mathbb{R}$

Model: $a(x) = b + w_1 x + w_2 x^2 + \cdots + w_{15} x^{15}$



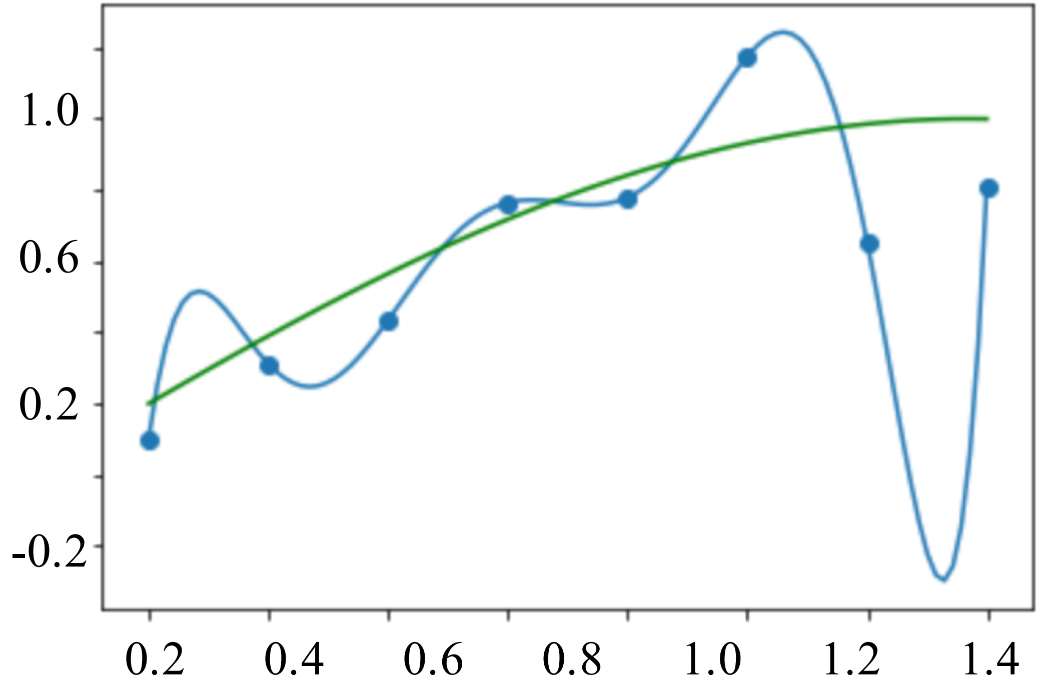
Overfitting example 2

Training set: $\{0.2, 0.4, \ldots, 1.6\}$, $y = \sin(x) + \epsilon$

Model: $a(x) = b + w_1 x + w_2 x^2 + \cdots + w_8 x^8$

Parameters: $(130.0, -525.8, \ldots, 102.6)$

Model just incorporates target into parameters!



Holdout set

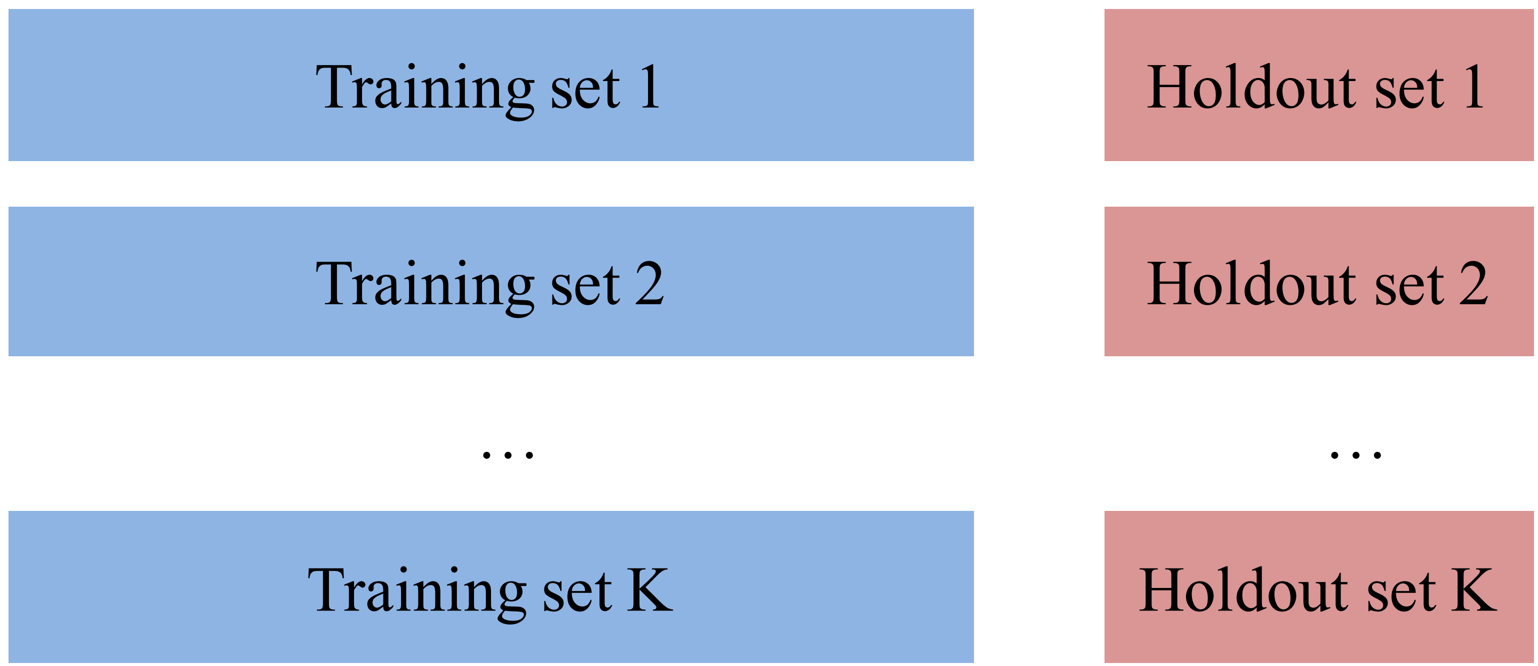
Small holdout set:

- Training set is representative
- Holdout quality has high variance

Large holdout set:

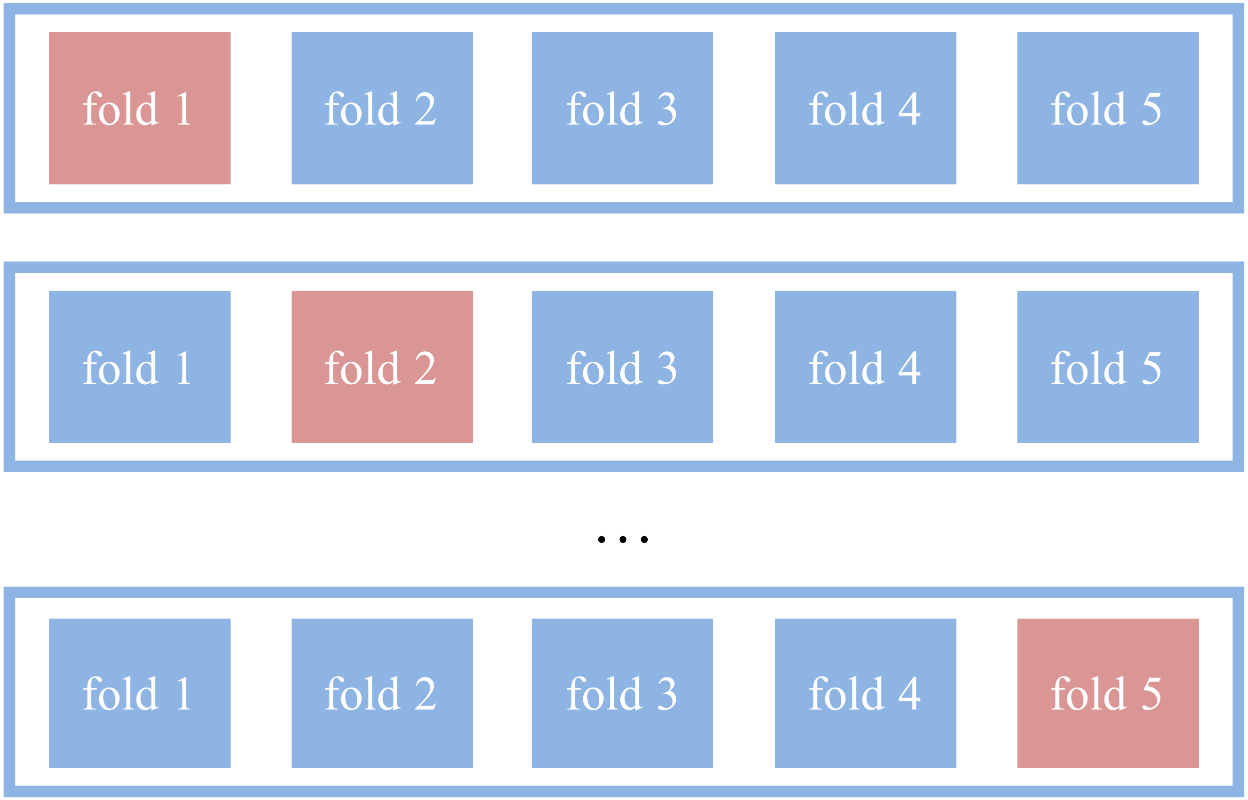
- Holdout quality has low variance
- Holdout quality has high bias

Holdout set



No guarantees that each object will be in holdout part at least once

Cross-validation



Cross-validation

- Requires to train models K times for K-fold CV
- Useful for small samples
- In deep learning holdout samples are usually preferred

Summary

- Models can easily overfit with high number of parameters
- Overfitted model just remembers target values for training set and doesn't generalize
- Holdout set or cross-validation can be used to estimate model performance on new data