Learning objectives

In this lecture we will...

• Introduce the concept of the validation set
• Explain the relationship between model parameters and hyperparameters
• Introduce the training → validation → test pipeline
In the last few lectures we saw...

- How a **training set** can be used to evaluate model performance on seen data
- How a **test set** can be used to estimate **generalization performance**
- How we can use a **regularizer** to mitigate overfitting
Recap...

In particular, our **regularizer** "trades-off" between model accuracy and model complexity

\[
\frac{1}{N} \sum_{i} (y_i - X_i \cdot \theta)^2 + \lambda \sum_{k} \theta_k^2
\]

- **MSE**
- **regularizer**

• We want a value of our regularization parameter that balances model accuracy (low MSE) with complexity (low sum of squared parameters)
In particular, our regularizer "trades-off" between model accuracy and model complexity.

- If we only cared about \textbf{training error}, we’d always select the smallest possible value of lambda (i.e., \( \lambda = 0 \)).
- We could tune against our \textbf{test set}, but that would mean looking at the test set many times (which would be cheating!)
Recap...

In particular, our regularizer "trades-off" between model accuracy and model complexity

- So, we need a third partition of our data, which is similar to the test set, but which can be used to select hyperparameters like lambda
  - This set is called the validation set
Training and test sets

- Training data
- Validation data
- Testing data

\[
\begin{bmatrix}
X
\end{bmatrix}
\theta = \begin{bmatrix}
y
\end{bmatrix}
\]
Training and test sets

Training data
Used to select $\theta$

Validation data
Used to select $\lambda$

Testing data
Only used once, to evaluate the model
We showed how a **validation set** can be used to tune parameters (or “hyperparameters”) that cannot be selected using the training set (or the test set).

In the following lecture, we’ll explore more how this set can be used to optimize model performance.