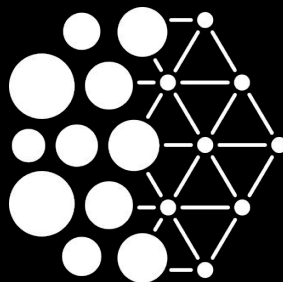


Quebec
Artificial
Intelligence
Institute



Mila

Introduction to Machine Learning

Gaétan Marceau Caron

Applied research scientist, Mila

gaetan.marceau.caron@mila.quebec

Different types of learning

- Supervised learning
 - Semi-supervised learning
 - Active learning
- Unsupervised learning
 - Self-supervised learning
- Reinforcement learning
- ...

Supervised learning

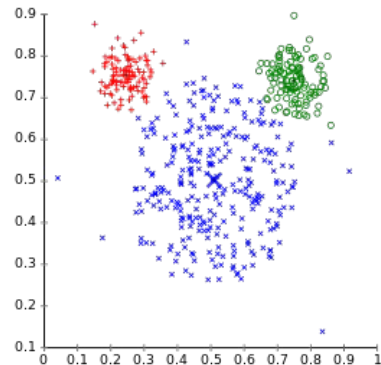
For each input in the dataset, we have the optimal output. $z = (x, y)$

We want to model the conditional probability: $p(y|x)$

- **Regression**: targets are real-valued variables.
- **Classification**: targets are categorical variables.
 - **Multi-class**: choose only one class among a predefined set.
 - **Multi-label**: choose all relevant classes among a predefined set.

Unsupervised learning

- In the dataset, we only have the inputs. $z = (x)$
- We want to model the marginal probability: $p(x)$
- Applications:
 - Dimensionality reduction
 - Clustering, anomaly detection
 - Data generation



Source: Kingma, Durk P., and Prafulla Dhariwal.
"Glow: Generative flow with invertible 1x1
convolutions." In Advances in Neural Information
Processing Systems, pp. 10215-10224. 2018.

Self-supervised learning

- In the dataset, we only have the inputs. $z = (x)$
- We want to model the conditional probability: $p(g(x)|x)$
where g is a function that creates *pseudo-labels* from the inputs.
- The goal is to learn good representations that transfer well to any task.
- Examples of pseudo-labels:
 - Is the video forward or backward?
 - Is the image upside-down?
 - Is the first sentence contradicts the second sentence?

Semi-supervised learning

- We consider two datasets:
 - Unlabeled examples (many examples)
 - Labeled examples (few examples)
- The goal is to learn good representations from the unlabeled examples that helps to train the model with the labeled examples for a specific task.
- Main idea: use the unlabeled examples to regularize the model.

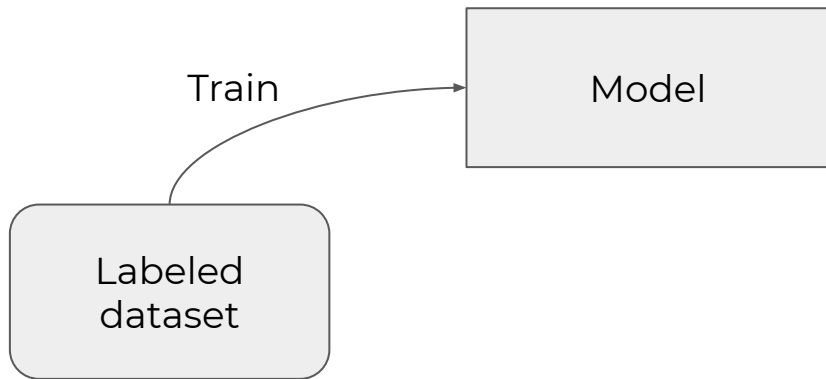
Learning with a teacher

The concept of **feedback** from a teacher is central in ML:

1. the model makes a prediction,
2. a teacher compares the model prediction with its prediction and gives back a feedback of how right is the prediction,
3. the model uses this feedback to improve its prediction.

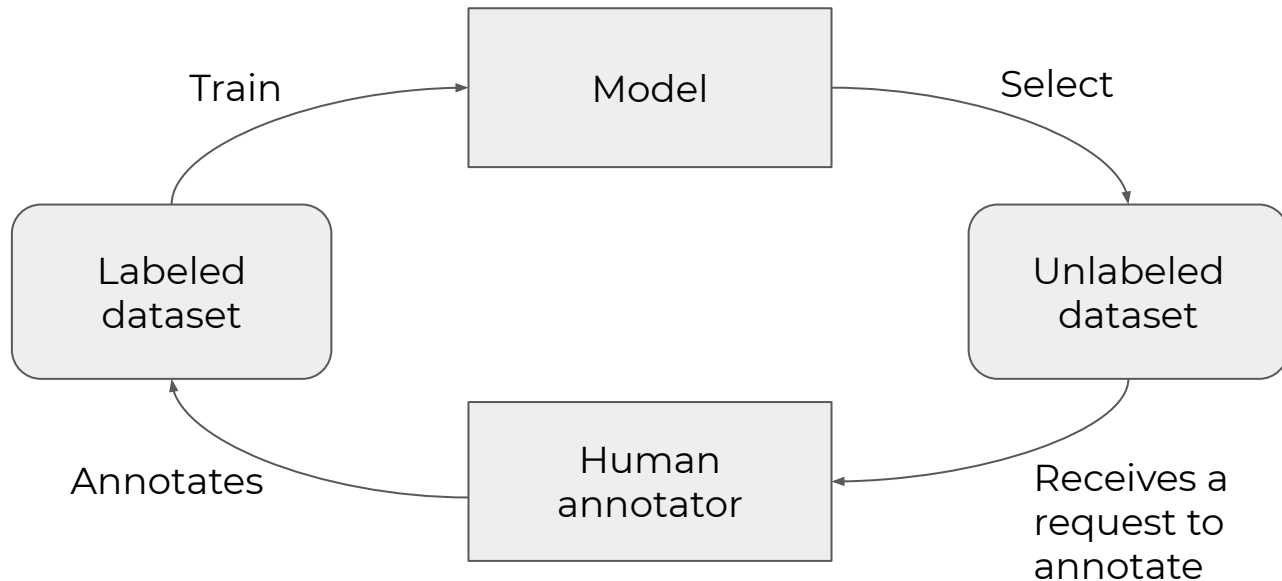
Active vs. passive learning

Passive learning: a teacher provides datasets independently of the learner.



Active vs. passive learning

Active learning: the learner can ask the teacher to annotate specific data.



Reinforcement learning

- Sequential decision making problem.
- Learning from interactions(examples) with the environment.
- The feedback (reward) concerns only the action taken in a given state.
- The agent builds its own dataset.

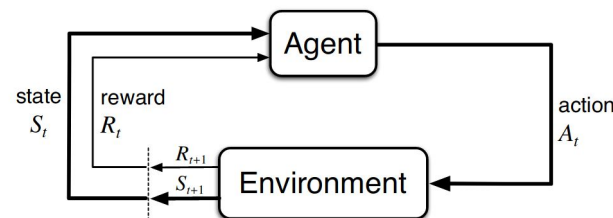
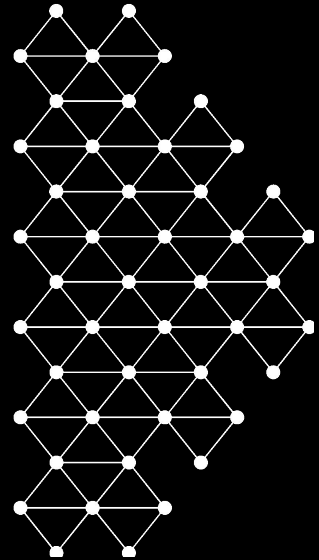


Figure 3.1: The agent–environment interaction in a Markov decision process.

Source: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018., P. 38

Supervised learning



Statistical learning framework

- Domain set \mathcal{X} : set of objects we want to annotate.
- Label set \mathcal{Y} : set of possible labels.
- Training data: finite sequence of pairs.

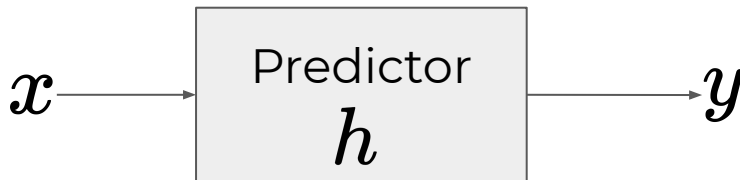
$$S = ((x_1, y_1), \dots, (x_m, y_m)) \quad (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \forall i$$

Statistical learning framework

- Learner's output $h \in \mathcal{H}$

A model, prediction rule, predictor, hypothesis or classifier.

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$



Statistical learning framework

Data generation model: $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

← The set of all possible probability distributions over domain and target spaces.

Independent and identically distributed (iid)

$$S \sim \mathcal{D}^m$$

$$(x_i, y_i) \sim \mathcal{D}$$

$$P_{\mathcal{D}}(S) = \prod_{i=1}^m P_{\mathcal{D}}(x_i, y_i)$$

Statistical learning framework

- Measure of success: loss function

$$l : \underbrace{(\mathcal{X} \times \mathcal{Y})}_{\text{Example}} \times \underbrace{\mathcal{H}}_{\text{Predictor}} \rightarrow \underbrace{\mathbb{R}^+}_{\text{Loss}}$$

- Examples:

- 0/1 loss (Classification): $l_{0-1}((x, y), h) = \begin{cases} 0, & \text{if } h(x) = y \\ 1, & \text{if } h(x) \neq y \end{cases}$

- Square loss (Regression): $l_{sq}((x, y), h) := (h(x) - y)^2$

Definition of risk

$$\underbrace{L_{\mathcal{D}}}_{\text{Data distribution}}(\underbrace{h}_{\text{Predictor}}) := \underbrace{\mathbb{E}_{(x,y) \sim \mathcal{D}}}_{\text{Expectation}} [\underbrace{l((x,y), h)}_{\text{Loss}}]$$

The risk is a weighted sum of the loss where a weight is the probability of the example. However, \mathcal{D} is unknown.

Definition of the empirical risk

$$L_S(h) := \underbrace{\frac{1}{m} \sum_{i=1}^m}_{\text{Average}} \underbrace{[l((x_i, y_i), h)]}_{\text{Loss}}$$

Predictor
Dataset

The empirical risk is the average of the loss evaluated on our dataset, not all possible examples.

Empirical risk minimization

Find the predictor that minimizes the empirical risk:

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

where

$$L_S(h) := \frac{1}{m} \sum_{i=1}^m [l((x_i, y_i), h)]$$

Main question in machine learning

Will the model perform the same in production than on the training set?

$$L_{\mathcal{D}}(h_S) \stackrel{?}{\approx} L_S(h_S)$$

where

$$L_{\mathcal{D}}(h) := \mathbb{E}_{(x,y) \sim \mathcal{D}} [l((x,y), h)]$$

$$L_S(h) := \frac{1}{m} \sum_{i=1}^m [l((x_i, y_i), h)]$$

What can go wrong?

- We only have access to a finite dataset:

$$S = ((x_1, y_1), \dots, (x_m, y_m)) \quad (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \forall i$$

- We are approximating an expectation:

$$L_{\mathcal{D}}(h) := \mathbb{E}_{(x,y) \sim \mathcal{D}} [l((x, y), h)]$$

$$L_S(h) := \frac{1}{m} \sum_{i=1}^m [l((x_i, y_i), h)]$$

What can go wrong? Wrong hypothesis class

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

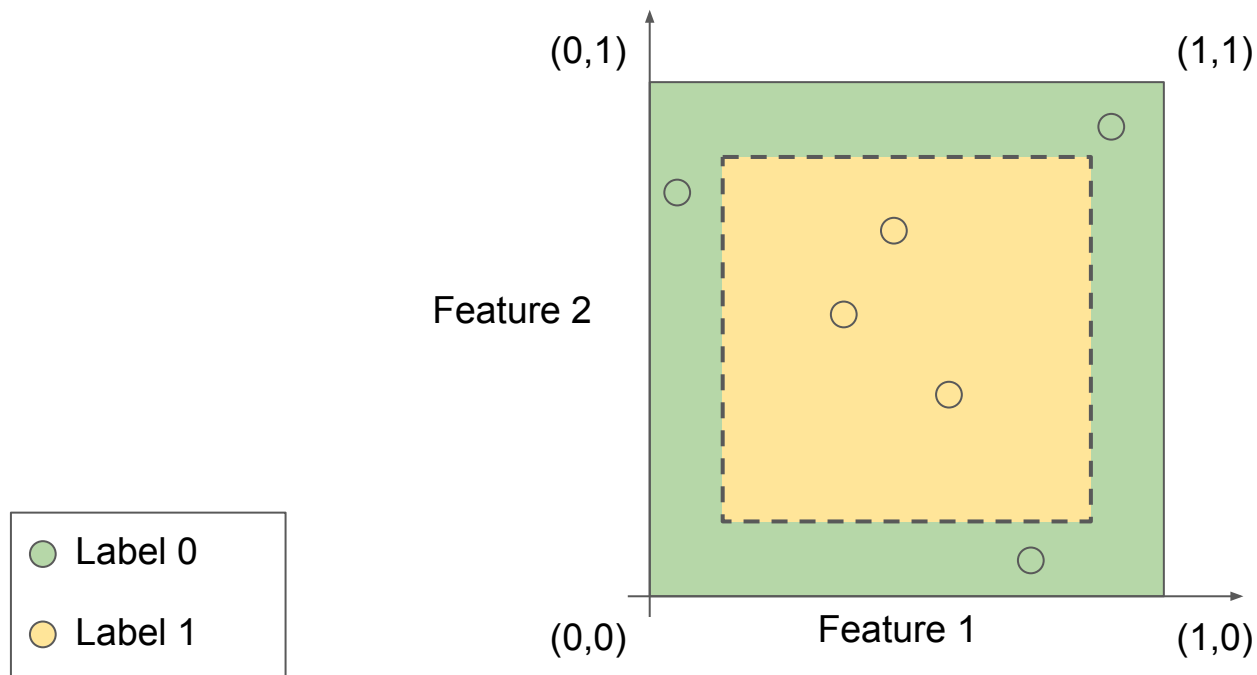
What is the space of predictors?

Optimal predictor for empirical risk: **a lookup table!**

$$h(x) = \begin{cases} y_i, & \text{if } \exists i \text{ st } x = x_i \\ 0, & \text{otherwise} \end{cases}$$

Example: 2D classification problem

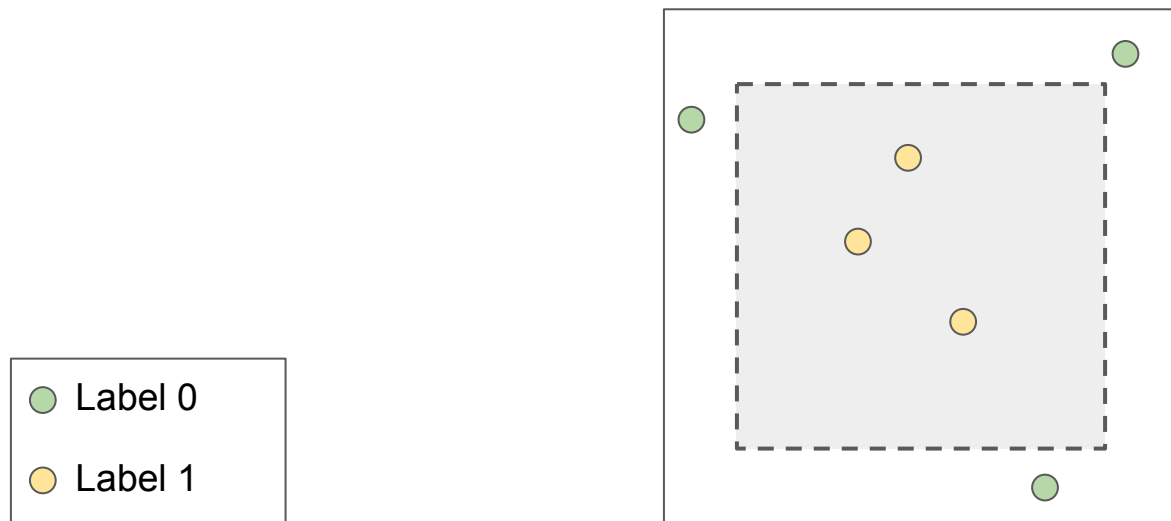
$$\mathcal{D} = \text{Uniform}([0, 1]^2)$$



$$l_{0-1}((x, y), h) = \begin{cases} 0, & \text{if } h(x) = y \\ 1, & \text{if } h(x) \neq y \end{cases}$$

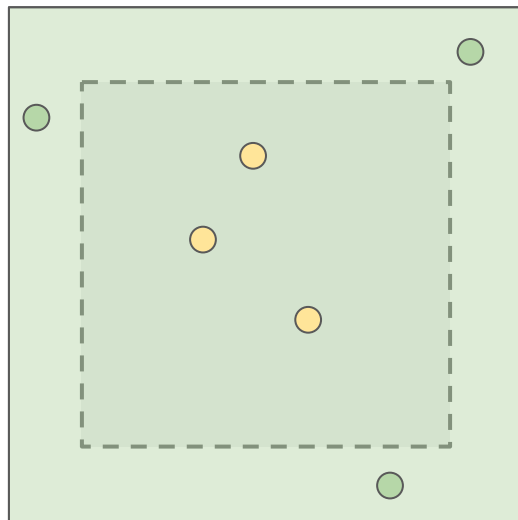
Lookup table predictor

$$h_S(x) = \begin{cases} y_i, & \text{if } \exists i \text{ st } x = x_i \\ 0, & \text{otherwise} \end{cases}$$



Lookup table predictor: decision boundary

$$h_S(x) = \begin{cases} y_i, & \text{if } \exists i \text{ st } x = x_i \\ 0, & \text{otherwise} \end{cases}$$



$$L_{\mathcal{D}}(h_S) \approx 0.5$$

$$L_S(h_S) = 0$$

What can go wrong? Overfitting

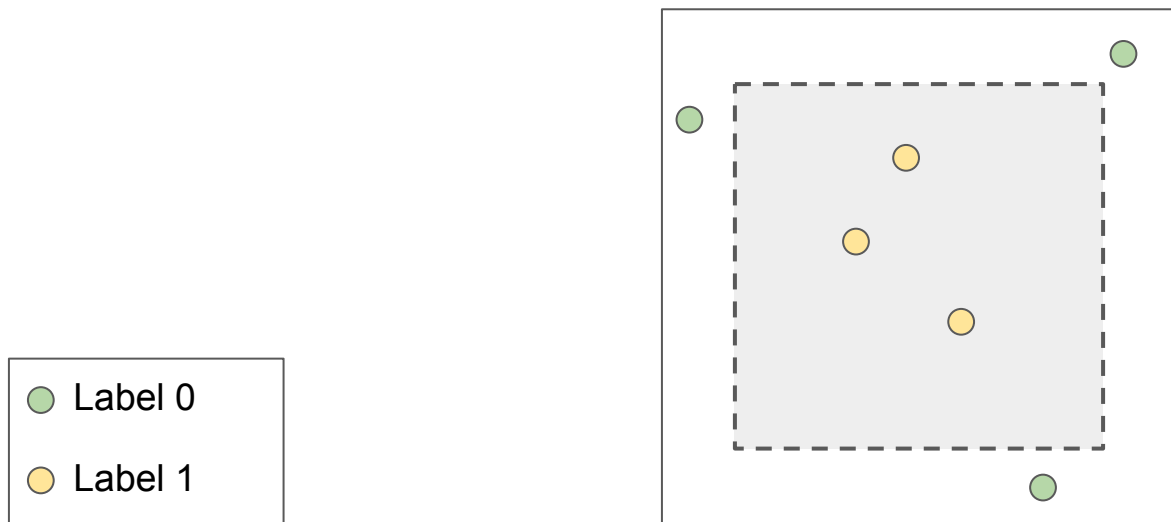
$$\underbrace{L_{\mathcal{D}}(h_S)}_{\text{Data distribution}} \gg \underbrace{L_S(h_S)}_{\text{Dataset}}^{\text{ERM Predictor}}$$

ERM: Empirical Risk Minimization

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

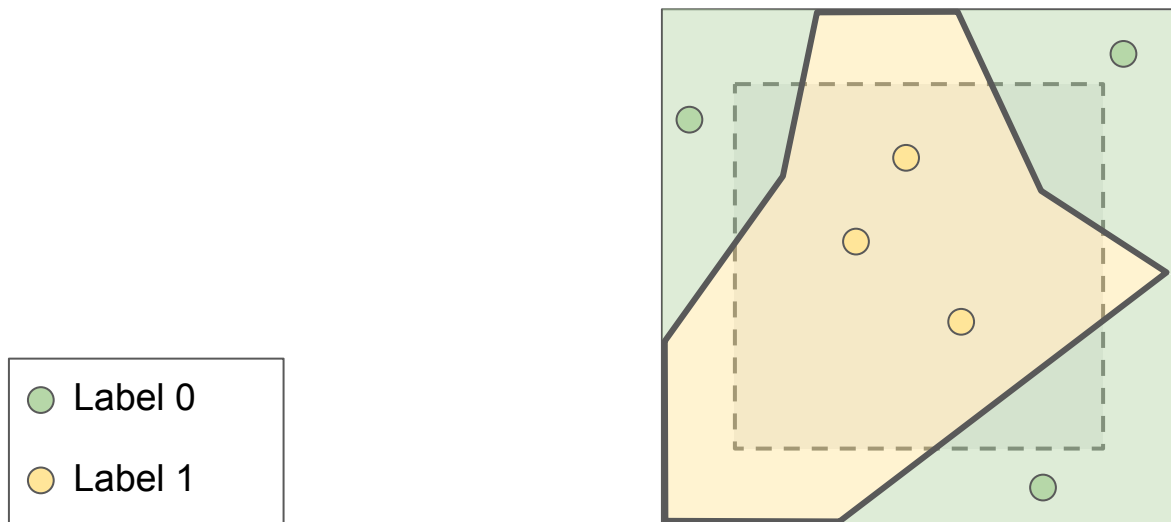
Small improvement: nearest neighbor

$$h_S(x) = y_i \text{ s.t. } i = \arg \min_i d(x, x_i)$$



Small improvement: nearest neighbor

$$h_S(x) = y_i \text{ s.t. } i = \arg \min_i d(x, x_i)$$



$$L_{\mathcal{D}}(h_S) \approx 0.35$$

$$L_S(h_S) = 0$$

Nearest neighbor: high variance

$$h_S(x) = y_i \text{ s.t. } i = \arg \min_i d(x, x_i)$$



$$L_{\mathcal{D}}(h_S) \approx 0.25$$

$$L_S(h_S) = 0$$

What can go wrong? Wrong hypothesis class

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

What is the space of predictors?

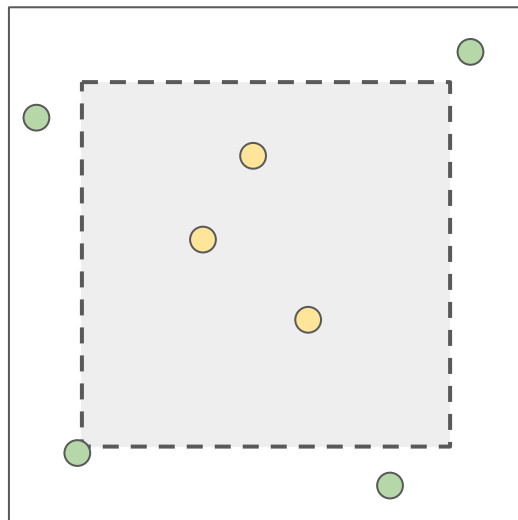
Let \mathcal{H} be the space of linear classifier.

$$h_{w,b}(x) = \text{sign}(\langle w, x \rangle + b)$$

Linear classifier

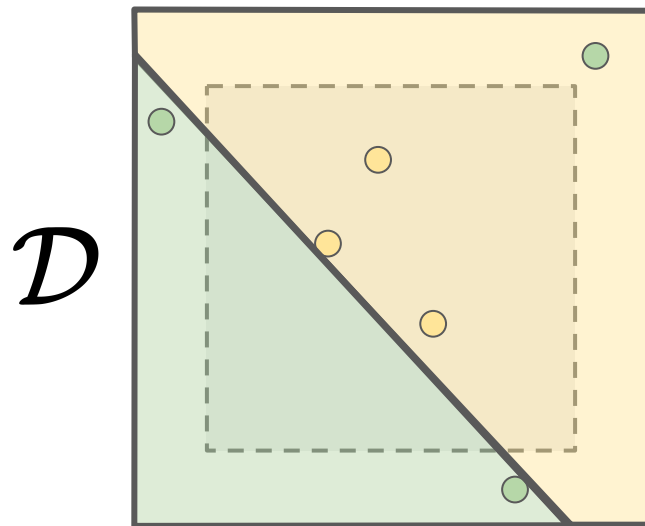
$$h_{w,b}(x) = \text{sign}(w_1 x_1 + w_2 x_2 + b)$$

\mathcal{D}



Linear classifier

$$h_{w,b}(x) = \text{sign}(w_1 x_1 + w_2 x_2 + b)$$

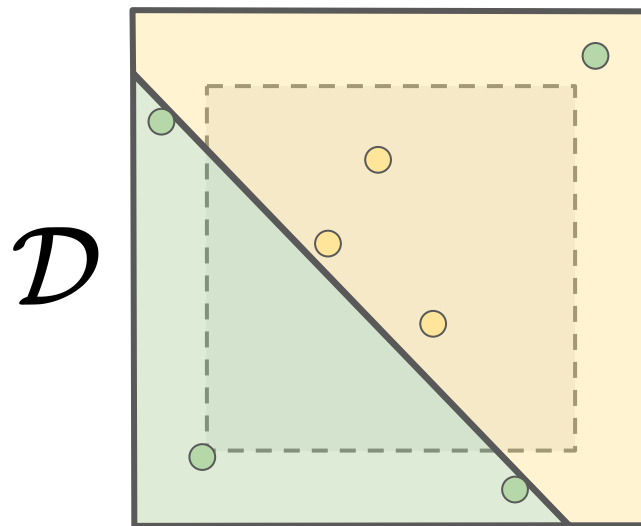


$$L_{\mathcal{D}}(h_S) \approx 0.70$$

$$L_S(h_S) \approx 0.14$$

Linear classifier

$$h_{w,b}(x) = \text{sign}(w_1 x_1 + w_2 x_2 + b)$$



$$L_{\mathcal{D}}(h_S) \approx 0.70$$

$$L_S(h_S) \approx 0.14$$

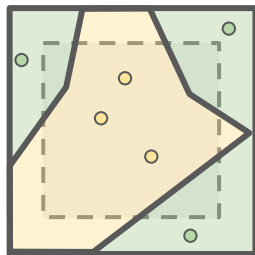
● Label 0
● Label 1

Two different hypothesis classes

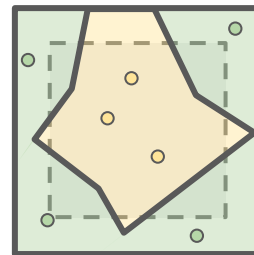
- Class of nearest neighbor classifiers
 - Instance-based learning (all the dataset is kept in memory).
 - Decision boundaries are complex and sensitive to new examples.
- Class of linear classifiers
 - Parametric model.
 - Decision boundaries are simple, and robust to new examples.

Bias vs. variance

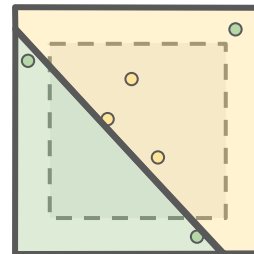
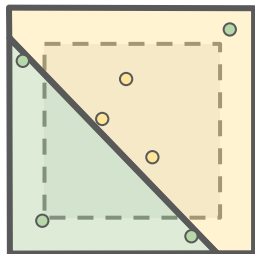
Estimation error
(High variance)



Sensitivity to
new examples

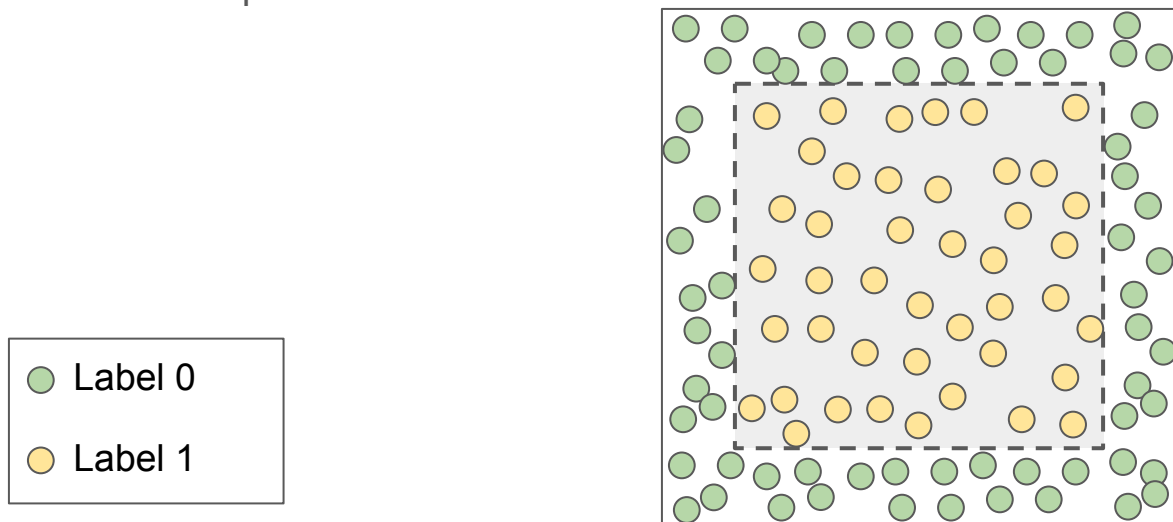


Approximation error
(High bias)



Can more data be the solution?

- Theoretically yes! By using the nearest neighbor algorithm with a large number of examples, the empirical risk minimizer will be close to the best predictor.



Can more data be the solution?

In practice, **No**! The number of examples to cover the domain space \mathcal{X} grows too fast with respect to the dimension of \mathcal{X} .

Data is **necessary**, but **not sufficient**.

Can more data be the solution?

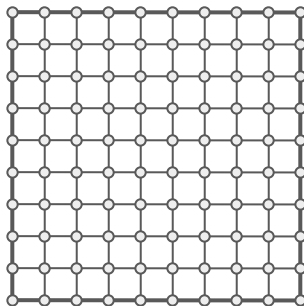
Intuition: suppose $\mathcal{X} = [0, 1]^d$, we want to cover \mathcal{X} with a regular grid with step $\epsilon = 0.1$.

$$d = 1$$



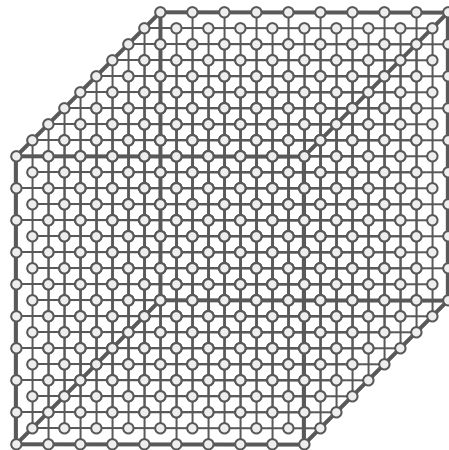
$$m = 10$$

$$d = 2$$



$$m = 100$$

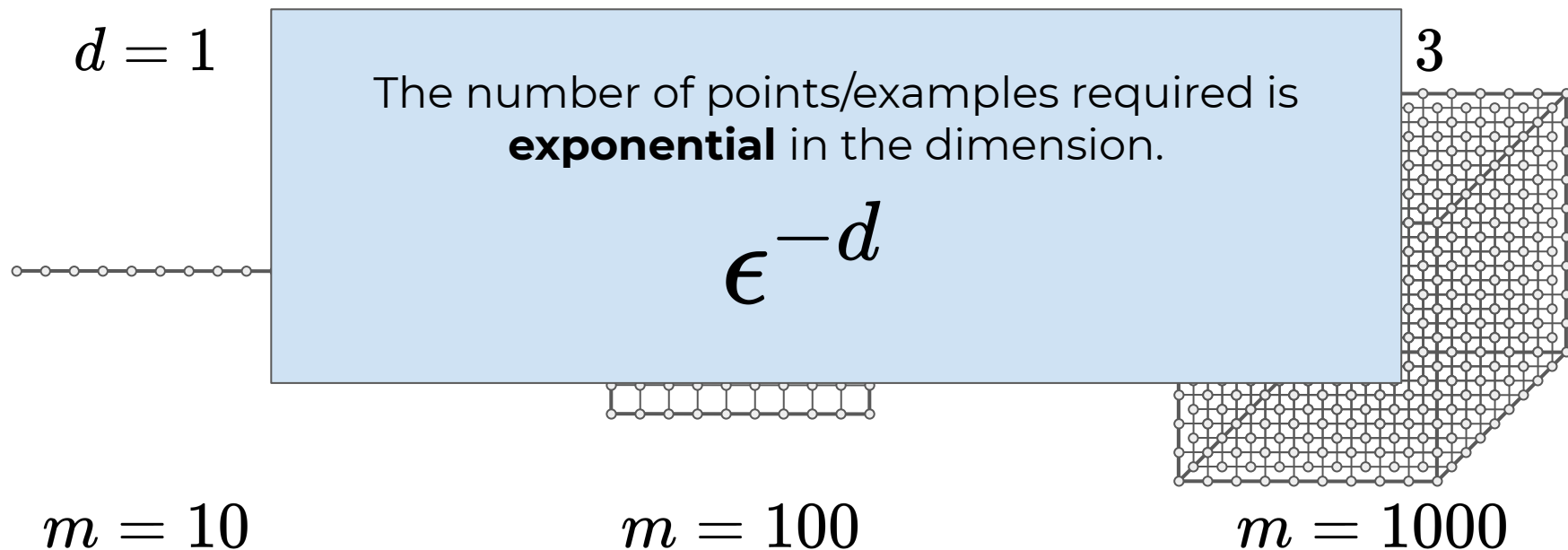
$$d = 3$$



$$m = 1000$$

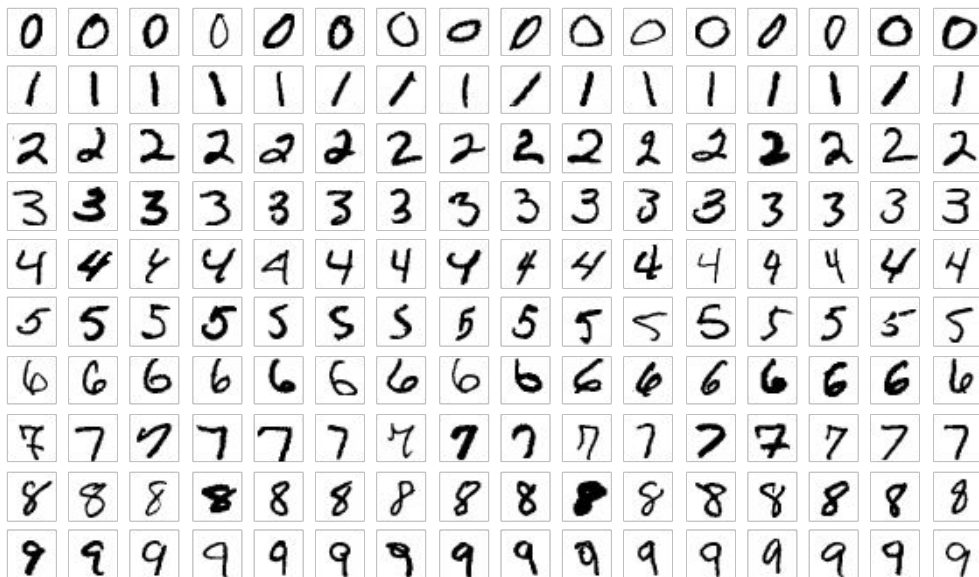
Can more data be the solution?

Intuition: suppose $\mathcal{X} = [0, 1]^d$, we want to cover \mathcal{X} with a regular grid with step $\epsilon = 0.1$.



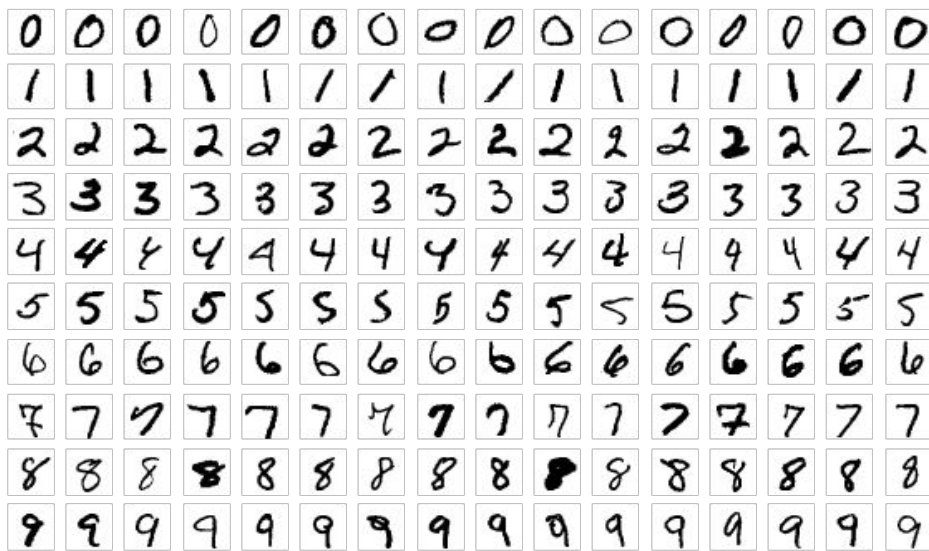
Simple example: MNIST

Image classification: image size $28 \times 28 = 784$.



Simple example: MNIST

If we sample uniformly from $\mathcal{X} = [0, 1]^{784}$, the probability of obtaining a realistic digit is close to 0.



Manifold assumption

In high-dimension, a lower-dimensional manifold supports the data distribution. The data **representation** has too many degrees of freedom compared to the underlying system.



The underlying system has 43 degrees of freedom (facial muscles) + some deformable parts (glasses, hair, ...)

The image representation has $960 \times 720 \times 3 = 2,073,600$ degrees of freedom (RGB pixels).

Simple example: MNIST

Even the notion of distance is hard to define in pixel space.

$$d(\text{0}, \text{0}) \approx 0$$

What is the best hypothesis class?

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

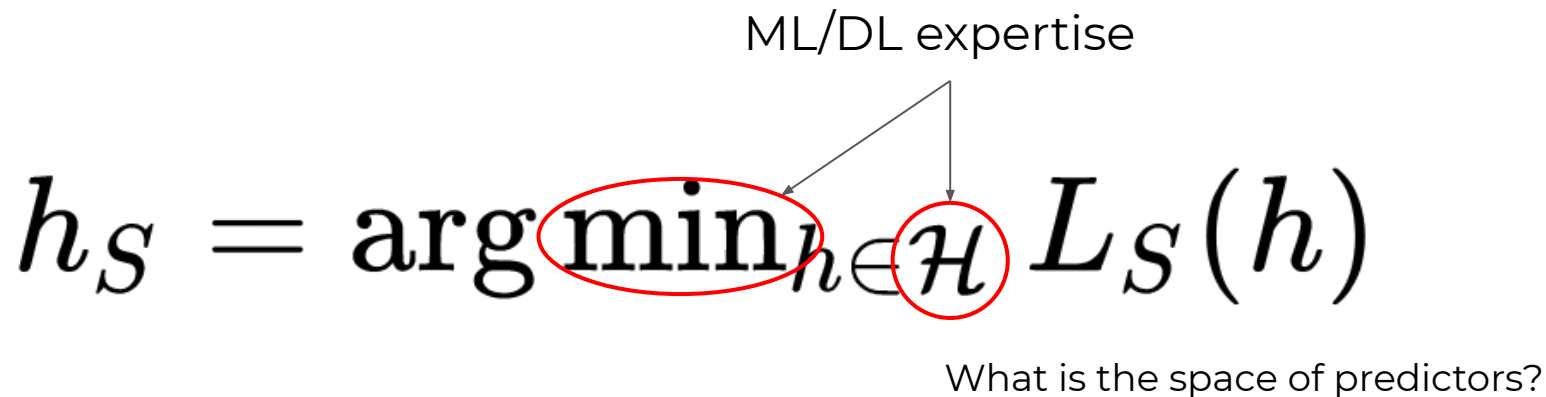
Inductive bias

We choose the hypothesis class and how to navigate in it with our prior knowledge on the task.

ML/DL expertise

$$h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$$

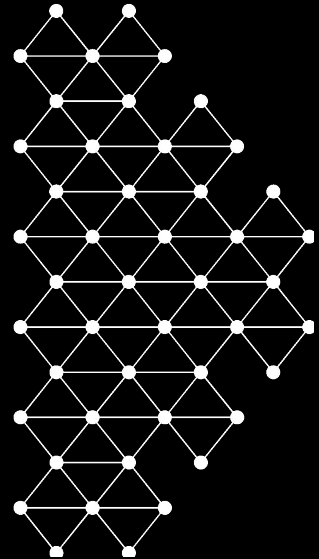
What is the space of predictors?

The diagram illustrates the concept of inductive bias in machine learning. It features the equation $h_S = \arg \min_{h \in \mathcal{H}} L_S(h)$. Above the equation, the text "ML/DL expertise" has two arrows pointing down to the \min operator and the hypothesis space \mathcal{H} . Both the \min operator and \mathcal{H} are circled in red. Below the equation, the text "What is the space of predictors?" is positioned under the \mathcal{H} circle.

How to choose \mathcal{H} ?

- **Deep learning** is a powerful way to describe parametric models in terms of computational modules.
- We can also restrict the values taken by the parameters in order to reduce the complexity of \mathcal{H} . We call this restriction **regularization**.

Hyperparameter tuning and model selection



How to diagnose overfitting?

Can we detect when

$$L_{\mathcal{D}}(h_S) \gg L_S(h_S) ?$$

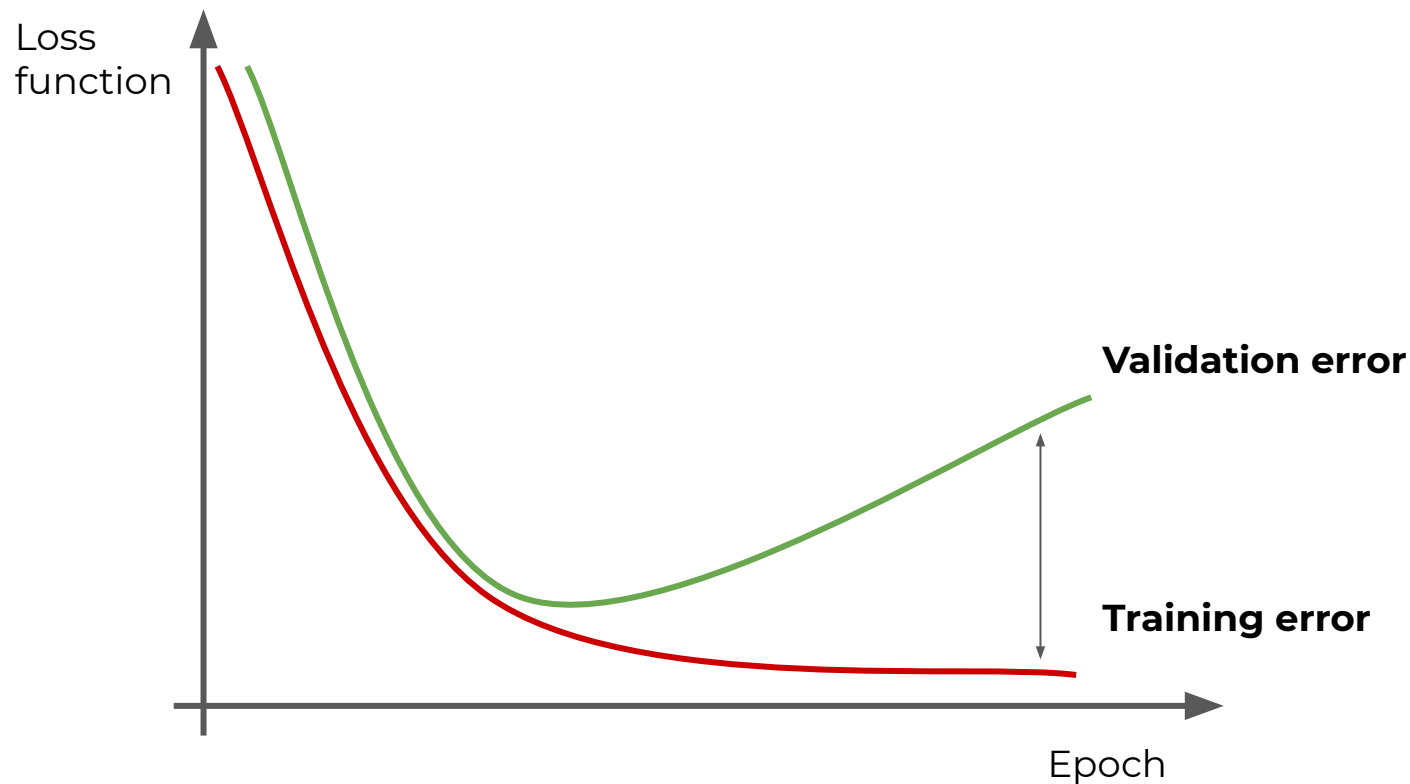
Hold out method (Validation set)

Monitor the loss function on **an independent set** of examples.

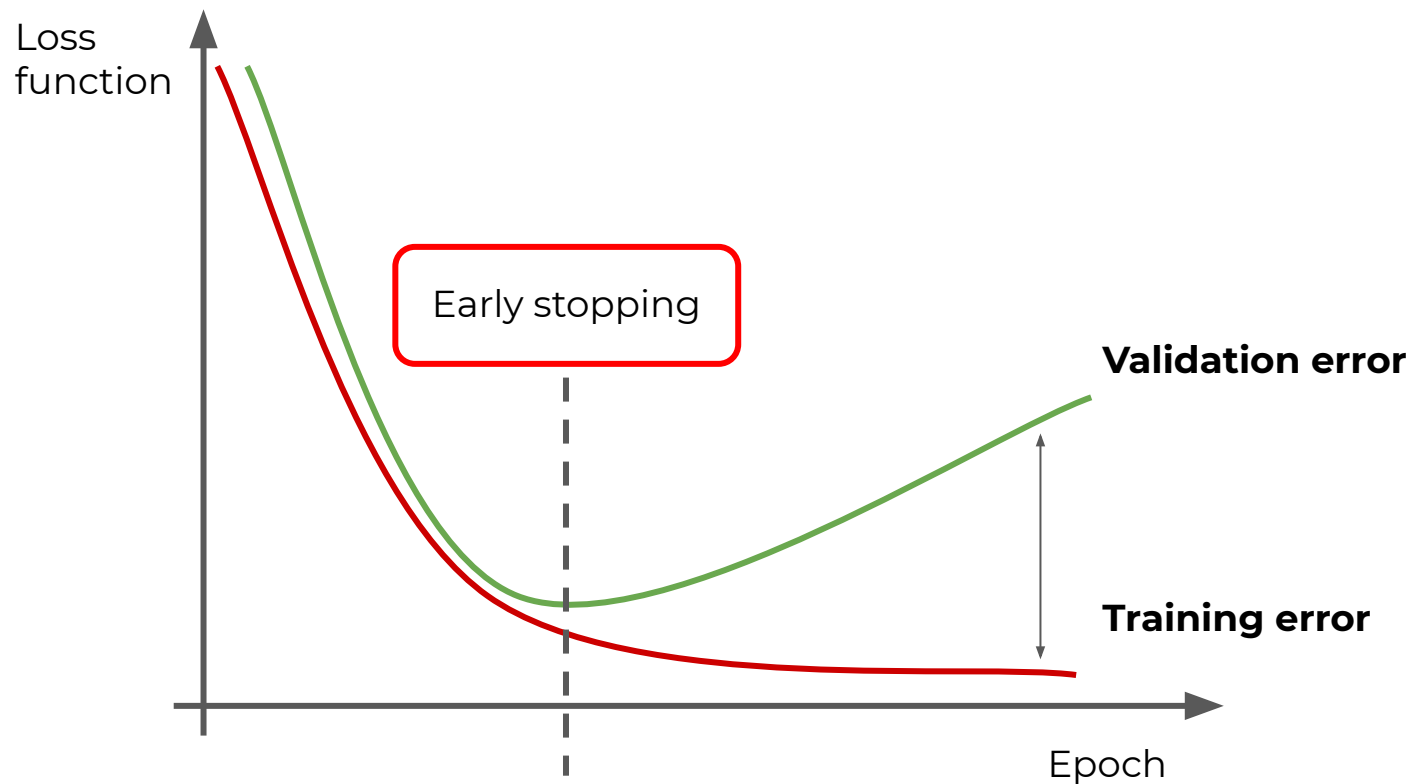
$$V = ((x_1, y_1), \dots, (x_{m_v}, y_{m_v})) \quad (x_i, y_i) \sim \mathcal{D}$$

$$L_V(h) := \frac{1}{m_v} \sum_{i=1}^{m_v} [l((x_i, y_i), h)]$$

How to diagnose overfitting?

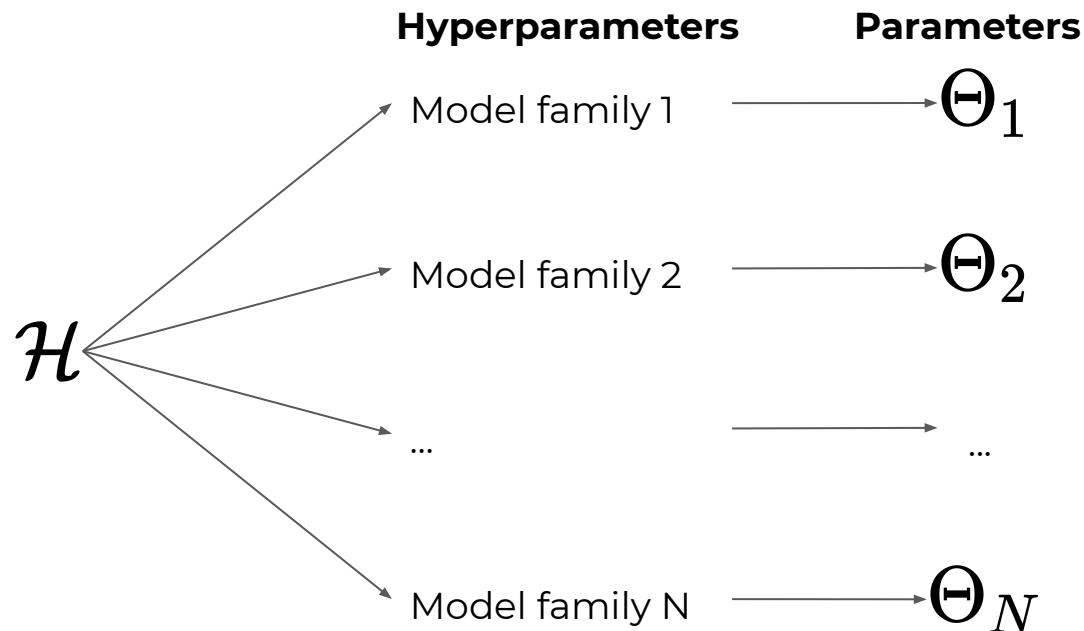


How to diagnose overfitting?



Hyperparameters tuning and model selection

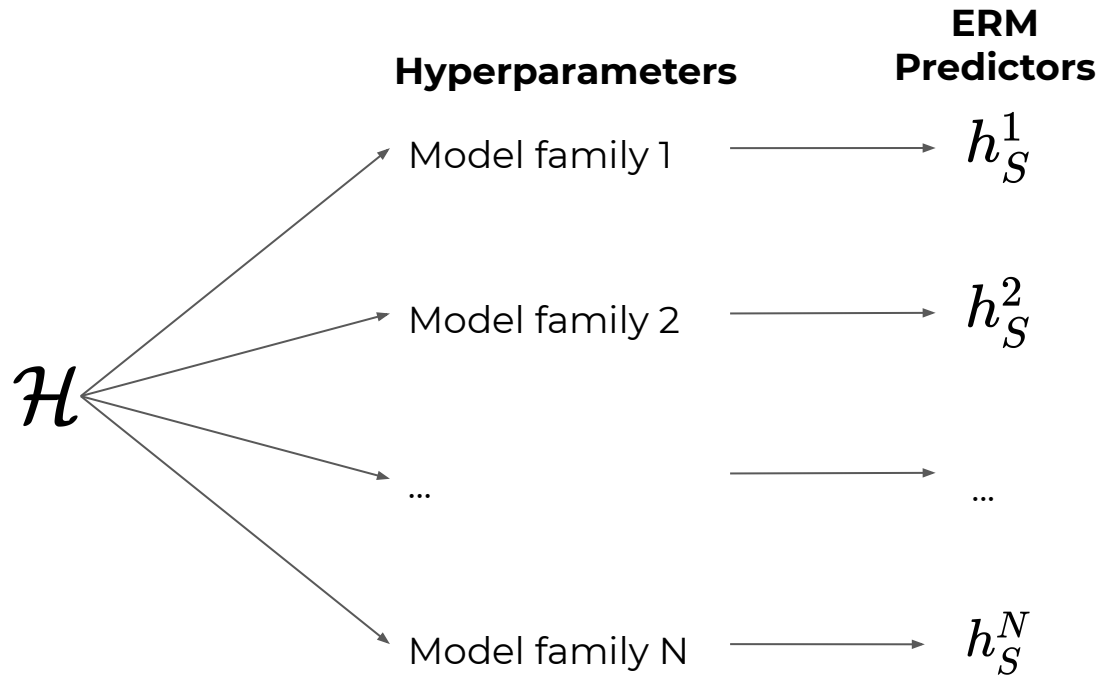
\mathcal{H} can have a complex structure.



Each model family has its own parameter space and optimization algorithm.

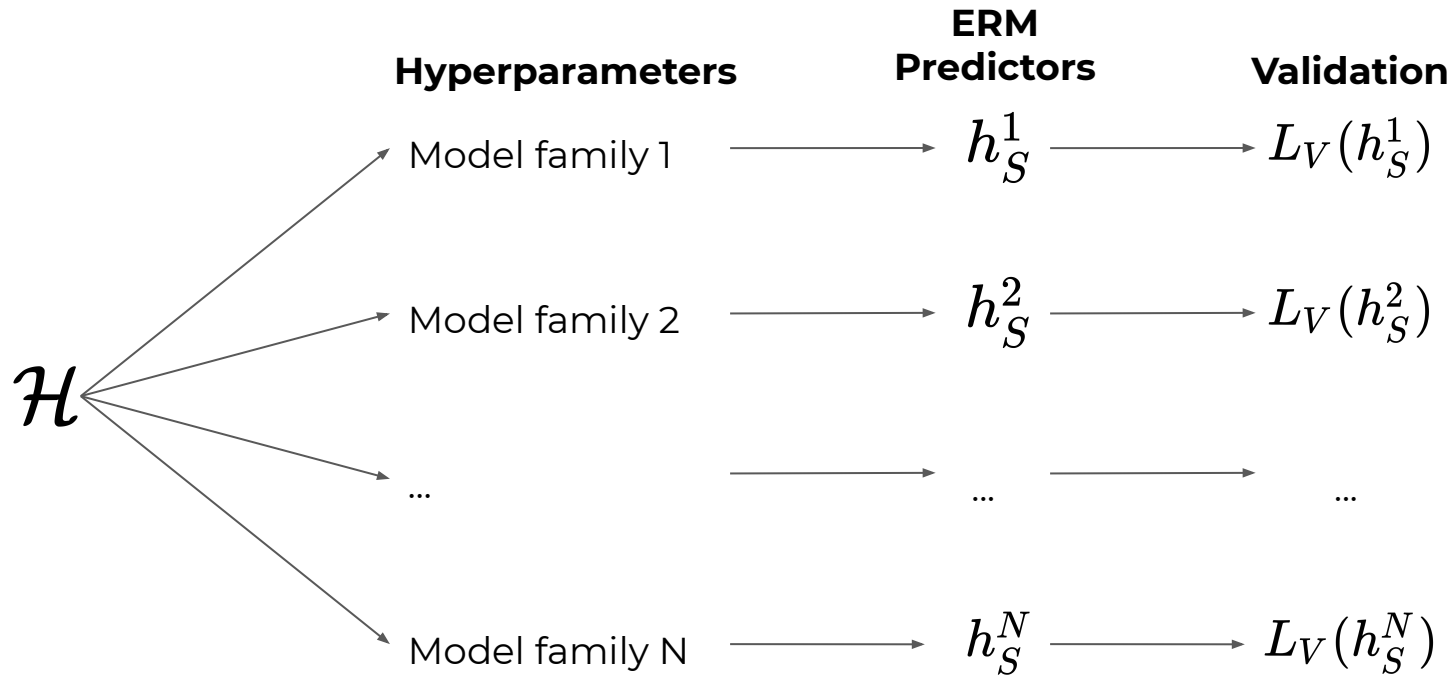
Hyperparameters tuning and model selection

Find the model with minimal risk for each family with **the training set**.



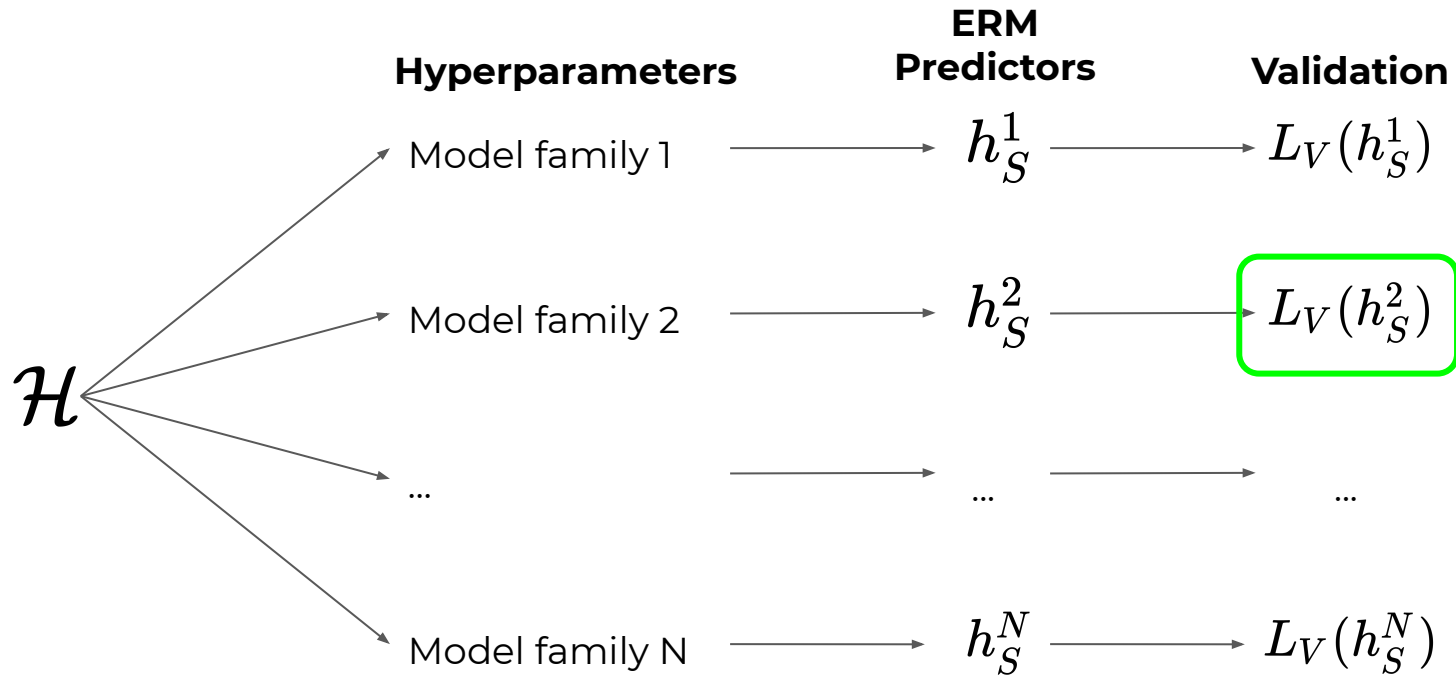
Hyperparameters tuning and model selection

Evaluate the ERM predictors on **the validation set**.



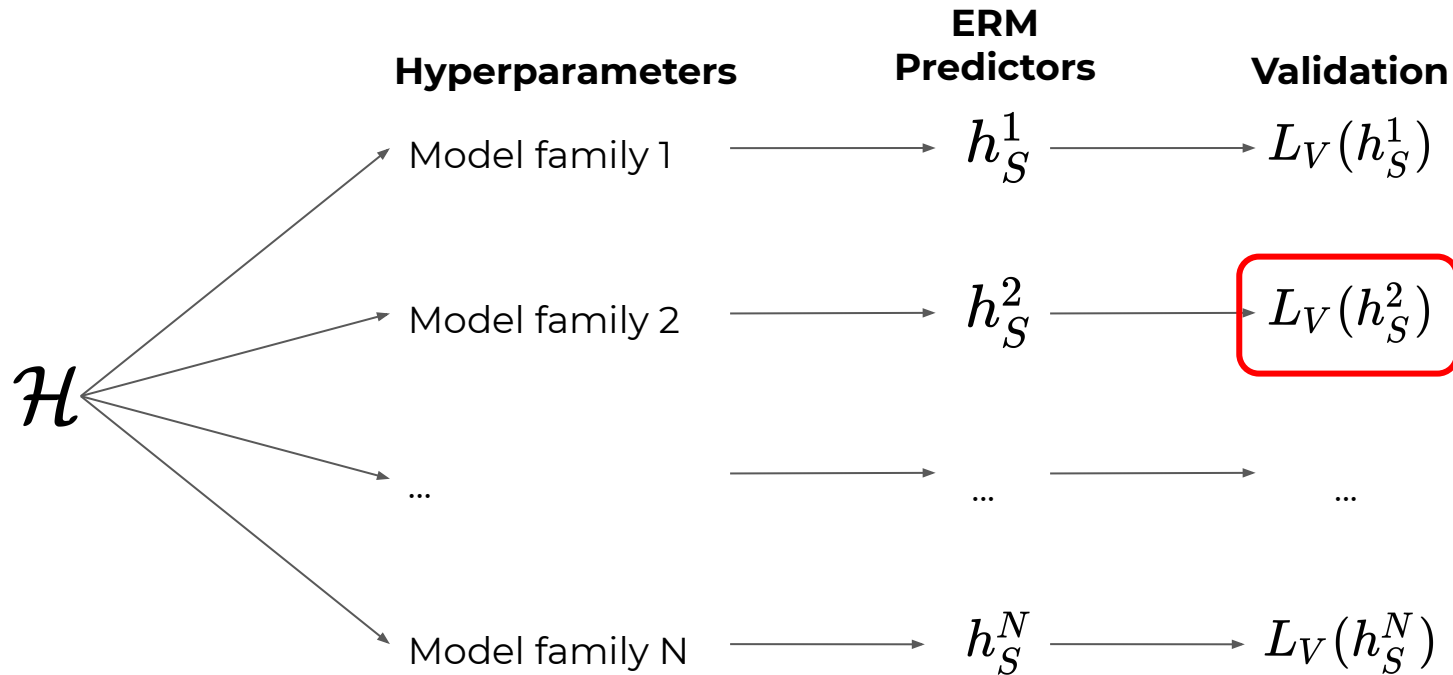
Hyperparameters tuning and model selection

Choose the predictor with the lowest risk on the validation set.



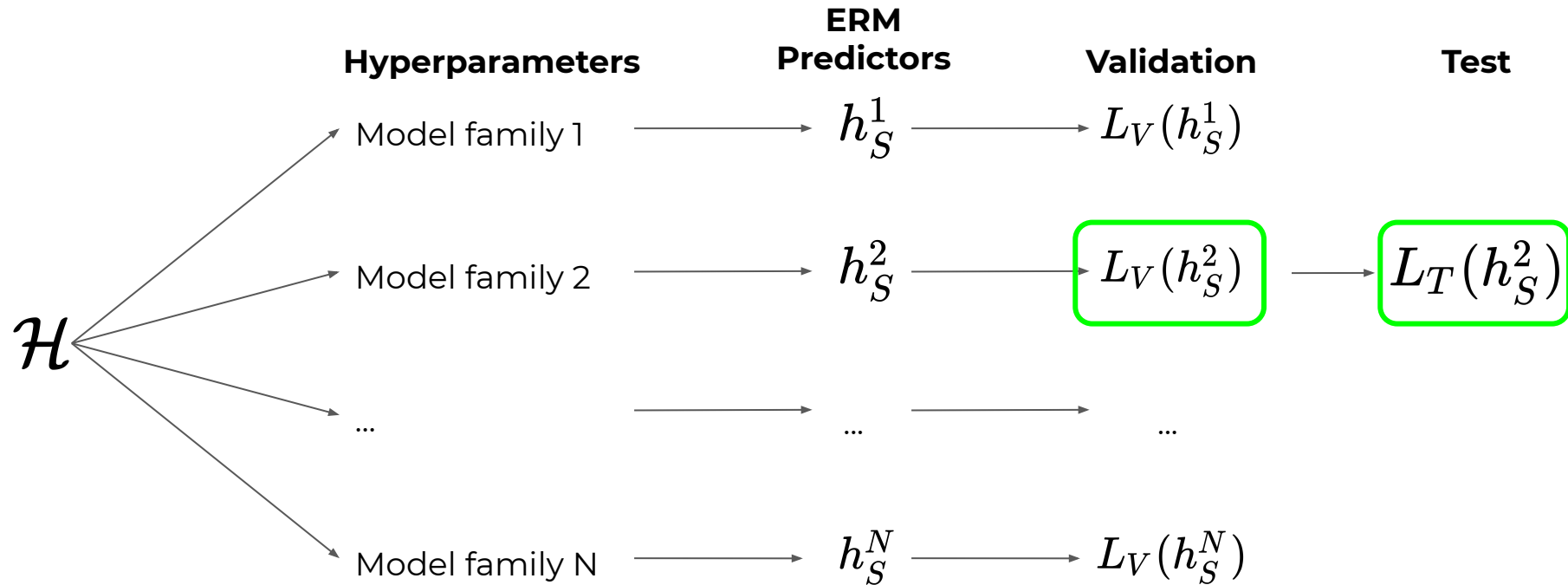
Hyperparameters tuning and model selection

If the number of model families is high, we can **overfit the validation set**



Hyperparameters tuning and model selection

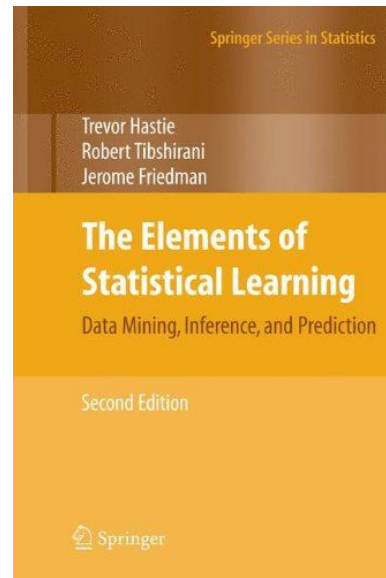
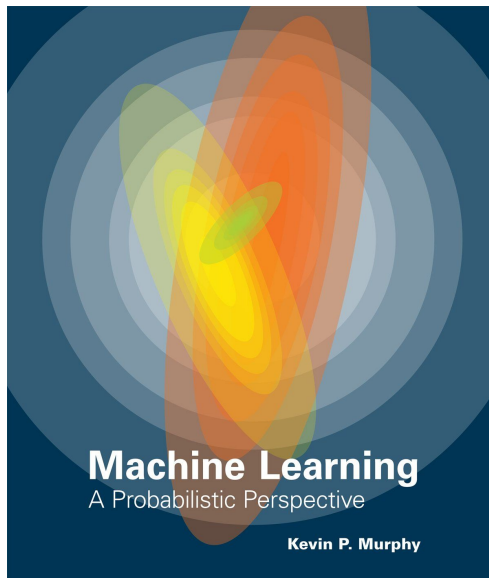
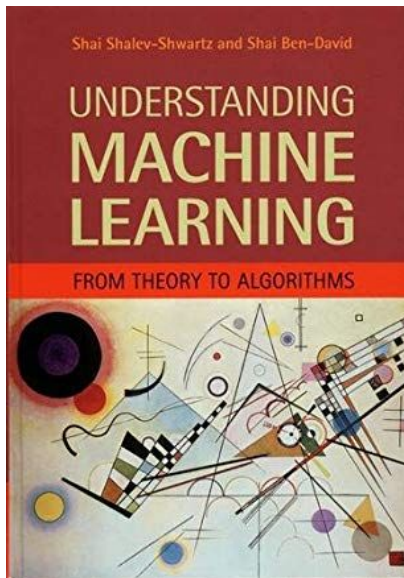
So, we use another set called the test set T



Take-home message

- Different types of learning have been studied in the literature.
- Statistical learning framework helps us to understand overfitting.
- Models can be too biased or have too much variance.
- Data is necessary, but not sufficient.
- Deep learning is an efficient way to define hypothesis classes.

References



Quebec
Artificial
Intelligence
Institute

