1. Introduction – Supervised Learning

Deep Learning

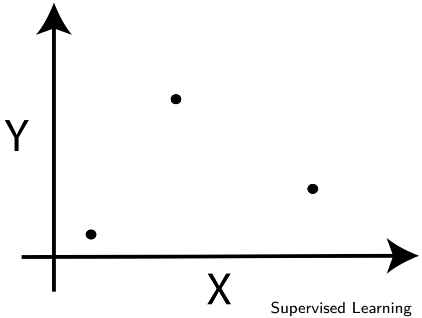
Prof. Dr. Christoph Lippert

based on the material at https://d2l.ai

Data

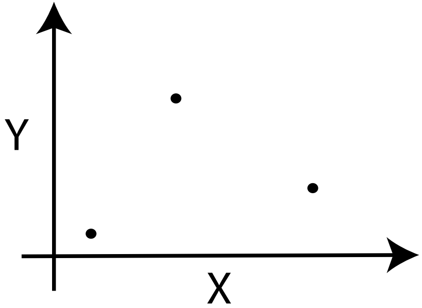
- Let $\mathcal{D}$ denote a dataset, consisting of N datapoints $\mathcal{D} = \{\underbrace{\mathbf{x}_i}_{\text{Inputs}}, \underbrace{y_i}_{\text{Outputs}}\}_{i=1}^{N}$.
- Typical (this course)
  - $\mathbf{x} = \{x_1, \ldots, x_D\}$ multivariate, spanning D features for each observation (age, image pixels etc.).
  - y
    - univariate (disease status, price, etc.).
    - multivariate (image segmentation, bounding box)

- Notation:
  - x: A scalar
  - $\mathbf{x}$: A vector
  - $\mathbf{X}$: A matrix
  - x_i, $[\mathbf{x}]_i$: The i^{th} element of vector $\mathbf{x}$
  - x_{ij}, $[\mathbf{X}]_{ij}$: The element of matrix $\mathbf{X}$ at row i and column j



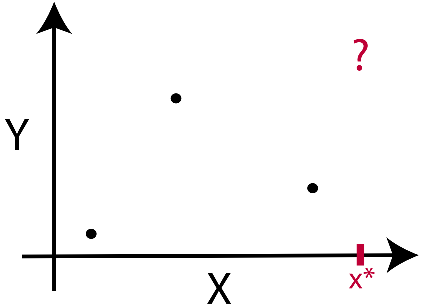
Predictions

- Observed training dataset $\mathcal{D} = \{ \underbrace{\mathbf{x}_n}_{\text{Inputs}}, \underbrace{y_n}_{\text{Outputs}} \}_{n=1}^{N}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?



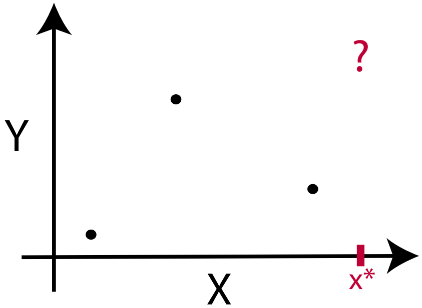
Predictions

- Observed training dataset $\mathcal{D} = \{\underbrace{\mathbf{x}_n}_{\text{Inputs}}, \underbrace{y_n}_{\text{Outputs}}\}_{n=1}^{N}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?



Model

- Observed **dataset** $\mathcal{D} = \{ \underbrace{\mathbf{x}_n}_{\text{Inputs}}, \underbrace{y_n}_{\text{Outputs}} \}_{n=1}^{N}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?
- To make predictions we need to make assumptions.
- A **model** encodes these assumptions and often depends on some parameters $\boldsymbol{\theta}$.

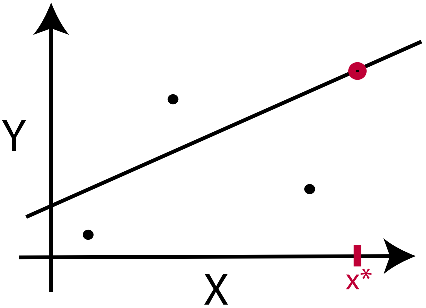


Model

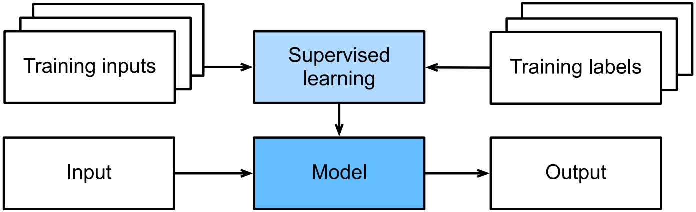
- Observed **dataset** $\mathcal{D} = \{\underbrace{\mathbf{x}_n}_{\text{Inputs}}, \underbrace{y_n}_{\text{Outputs}}\}_{n=1}^{N}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?
- To make predictions we need to make assumptions.
- A **model** encodes these assumptions and often depends on some parameters $\boldsymbol{\theta}$.

- Curve fitting: the model relates $\mathbf{x}$ to y,

$$\begin{aligned} y &= f(x; \underbrace{(w, b)}_{\boldsymbol{\theta}}) \\ &= \underbrace{b + w \cdot x}_{\text{example: a linear model}} \end{aligned}$$



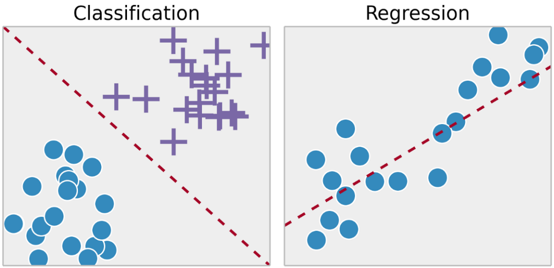
- Observed **dataset** $\mathcal{D} = \{\underbrace{\mathbf{x}_n}_{\text{Inputs}}, \underbrace{y_n}_{\text{Outputs}}\}_{n=1}^{N}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?
- To make predictions we need to make assumptions.
- A **model** encodes these assumptions and often depends on some parameters $\boldsymbol{\theta}$.



Classification vs. regression

Classification:

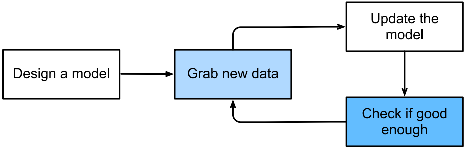
- Categorize data into discrete categories (e.g. disease vs healthy)



Regression:

- Predict continuous output variable (e.g. insulin level)

Training process



- In order to **optimize** our parameters, we need an **objective function** to score different models and parameter settings.
- Based on our training data, we find the model that minimizes the loss using an **optimization algorithm**.
    - Training Error:
      The error on that data on which the model was trained.
    - Test Error:
      This is the error incurred on an unseen test set.
      Can deviate significantly from the training error.

Regression

- L1 loss

$$l(y, y') = \sum_i |y_i - f_i'|$$

- squared loss, or L2 loss

$$l(y, y') = \sum_i (y_i - f_i')^2.$$

Classification

Predict (probability P of) a category given the input

- Negative **log likelihood** of the labels y_i

$$\sum_i -\log P(y_i|x_i;\boldsymbol{\theta})$$

Better known as **cross entropy** loss

$P(y = \text{deathcap}|\text{image}) = 0.2$

$$L(\text{action}|x) = E_{y\sim p(y|x)}[\text{loss}(\text{action}, y)].$$

Summary

For supervised learning, we need

- Labeled training data
- A model with parameters that encodes my assumptions
- An objective function (loss function)
- An optimization algorithm to find the best parameter values