Supervised learning

Some remarks on the mathematical foundation

Sören Christensen

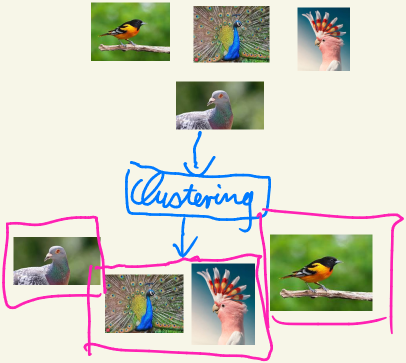
December 13, 2022

Sections

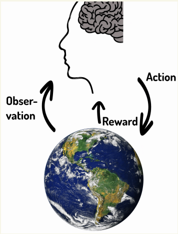
1. What is this lecture about?
2. Framework for learning
3. Some ideas from classical statistical learning
4. Artificial neural networks
5. Statistical learning for ANN

MACHINE LEARNING

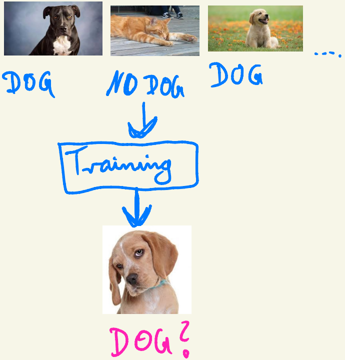
UNSUPERVISED LEARNING

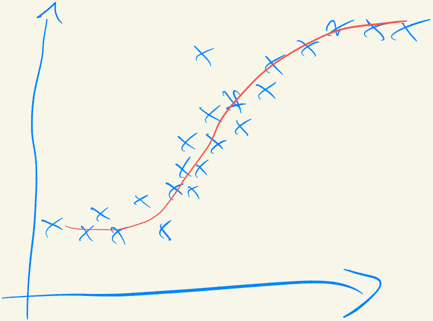


REINFORCEMENT LEARNING



SUPERVISED LEARNING





Question

Problem:

- Have some training data $(X_1, Y_1), \ldots, (X_m, Y_m)$.
- Want to find some function f from some class $\mathcal{F}$ so that for new observed X and unknown Y we have $f(X) \approx Y$.

Question

Problem:

- Have some training data $(X_1, Y_1), \dots, (X_m, Y_m)$.
- Want to find some function f from some class $\mathcal{F}$ so that for new observed X and unknown Y we have $f(X) \approx Y$.

Questions:

- $\mathcal{F}$?
- How to find suitable $f \in \mathcal{F}$?
- $\approx$?

Sections

1. What is this lecture about?
2. Framework for learning
3. Some ideas from classical statistical learning
4. Artificial neural networks
5. Statistical learning for ANN

Framework

Our model is given by

- **domain set** $\mathcal{X}$ (the set of objects we may wish to label), usually represented by a vector of **features**;
- **label set** $\mathcal{M}$, e.g. $\{0,1\}$ or $\mathcal{M} = \mathbb{R}^p$,
- random variables $(X,Y)\colon \Omega \to \mathcal{X} \times \mathcal{M}$, having (unknown) distribution $\mathcal{D}$ under $\mathbb{P}$
- a (known) **training set** $(X_1,Y_1),\dots,(X_m,Y_m)\colon \Omega \to \mathcal{X} \times \mathcal{M}$ of (i.i.d.?) random variables having the same distribution as (X,Y),
- Write $S = (X_1,Y_1,\dots,X_m,Y_m)$ for short.

Definition

A **hypothesis** is a mapping $f\colon \mathcal{X} \to \mathcal{M}$.

Definition

A **hypothesis** is a mapping $f\colon \mathcal{X} \to \mathcal{M}$.

prediction rule (also called **predictor**, **classifier**)

$$f = f_\cdot \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{M}, (x, s) \mapsto f_s(x).$$

The (quadratic) **risk** of f.

$$\mathcal{R}(f_S) \;=\; \mathbb{E}[|f_S(X) - Y|^2 \big| S],$$

Definition

A **hypothesis** is a mapping $f\colon \mathcal{X} \to \mathcal{M}$.

prediction rule (also called **predictor**, **classifier**)

$$f = f_\cdot \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{M}, (x, s) \mapsto f_s(x).$$

The (quadratic) **risk** of f.

$$\mathcal{R}(f_S) \; = \; \mathbb{E}[|f_S(X) - Y|^2 \big| S],$$

empirical risk of h:

$$\widehat{\mathcal{R}}(f) = \frac{1}{m} \sum_{i=1}^{m} |f(X_i) - Y_i|^2$$

Definition

A **hypothesis** is a mapping $f\colon \mathcal{X} \to \mathcal{M}$.

prediction rule (also called **predictor**, **classifier**)

$$f = f_{\cdot}\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{M}, (x, s) \mapsto f_s(x).$$

The (quadratic) **risk** of f.

$$\mathcal{R}(f_S) \ = \ \mathbb{E}[|f_S(X) - Y|^2 \big| S],$$

empirical risk of h:

$$\widehat{\mathcal{R}}(f) = \frac{1}{m} \sum_{i=1}^{m} |f(X_i) - Y_i|^2$$

Learning paradigm

For a hypothesis set $\mathcal{F}$, use the empirical risk minimizer $\hat{f} \in \operatorname{argmin}_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f)$.

Sections

1. What is this lecture about?
2. Framework for learning
3. Some ideas from classical statistical learning
4. Artificial neural networks
5. Statistical learning for ANN

No-free-lunch-Theorem

"Theorem"

For each non-trivial domain set $\mathcal{X}$ and for each prediction rule $f.$, there is a distribution of (X_i, Y_i) such that the risk of $f.$ is high with probability bounded away from 0.

No-free-lunch-Theorem

"Theorem"

For each non-trivial domain set $\mathcal{X}$ and for each prediction rule $f.$, there is a distribution of (X_i, Y_i) such that the risk of $f.$ is high with probability bounded away from 0.

1st Idea

There is no universal learner.

Error decomposition

$$\mathcal{R}(f_S) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \leq \widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + 2\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$$

Error decomposition

$$\mathcal{R}(f_S) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \leq \widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + 2\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$$

$$\mathcal{R}(f_S) \leq \inf_{f\in\mathcal{F}} \mathcal{R}(f) + \widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + 2\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$$

Error decomposition

$$\mathcal{R}(f_S) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \le \widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + 2\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$$

$$\mathcal{R}(f_S) \le \inf_{f\in\mathcal{F}} \mathcal{R}(f) + \widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + 2\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$$

- $\inf_{f\in\mathcal{F}} \mathcal{R}(f)$: approximation error
- $\widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f)$: optimization error
- $\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$: generalization / statistical error

Statistical error $\sup_{f\in\mathcal{F}}|\mathcal{R}(f)-\widehat{\mathcal{R}}(f)|$

Classical bound (for classification)

With probability $\geq 1-\delta$

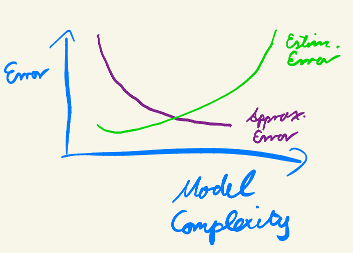
$$\text{statistical error} \leq \sqrt{\frac{\mathrm{VCdim}(\mathcal{F})+\log(1/\delta)}{m}}$$

Statistical error $\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$

Classical bound (for classification)

With probability $\geq 1-\delta$

$$\text{statistical error} \leq \sqrt{\frac{\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)}{m}}$$

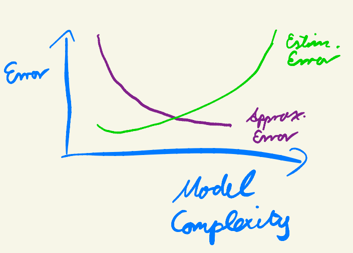


Statistical error $\sup_{f\in\mathcal{F}} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$

Classical bound (for classification)

With probability $\geq 1-\delta$

$$\text{statistical error} \leq \sqrt{\frac{\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)}{m}}$$



2nd Idea

Remember Occam's Razor.

On the optimization error $\widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f)$

Main idea: gradient decent (w.r.t. a parameterized version of f)

On the optimization error $\widehat{\mathcal{R}}(f_S) - \inf_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f)$

Main idea: gradient decent (w.r.t. a parameterized version of f)

$$\widehat{\mathcal{R}}(f_a) = \frac{1}{m}\sum_{i=1}^{m} |f_a(X_i) - Y_i|^2 \quad \to \min_a !$$

On the optimization error $\widehat{\mathcal{R}}(f_S) - \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f)$

Main idea: gradient decent (w.r.t. a parameterized version of f)

$$\widehat{\mathcal{R}}(f_a) = \frac{1}{m}\sum_{i=1}^{m} |f_a(X_i) - Y_i|^2 \quad \to \min_a!$$

- m large! → stochastic gradient decent
- usually not convex (as a function of the parameters)!

3rd Idea

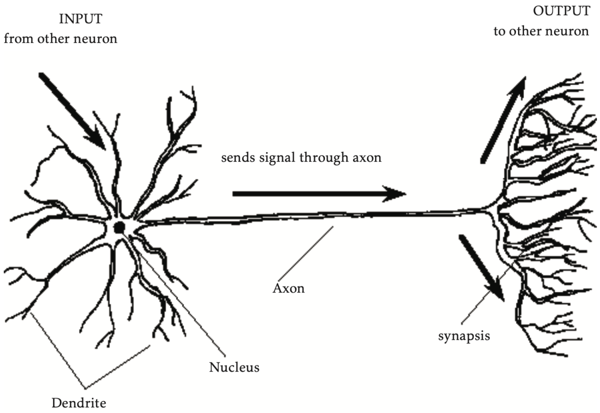
There are some convergence guarantees, but under strong assumptions, e.g. , convexity.

Sections

1. What is this lecture about?
2. Framework for learning
3. Some ideas from classical statistical learning
4. Artificial neural networks
5. Statistical learning for ANN

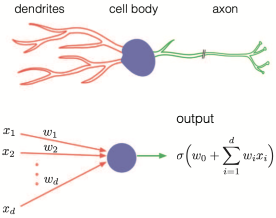
Artificial neural networks (ANN)

- ANNs are inspired by the structure of the (human) brain
- in biology, a **neuron** is an electrically excitable cell that communicates with other cells via specialized connections called **synapses**



Neural networks

- flexible function class for approximation of continuous functions
- incorporates some of the properties of biological neurons



: Source: Michael B. Wolf: “Mathematical Foundations of Supervised Learning”

- can be used for classification and **regression problems**

Illustration of neural networks

- any $g \in \mathcal{F}(L, \mathbf{p})$ is built by alternating matrix-vector multiplications with the action of the non-linear activation function σ
- in any step, the initial value $\mathbf{x}^{(0)} = \mathbf{x} \in \mathbb{R}^d$ is updated via

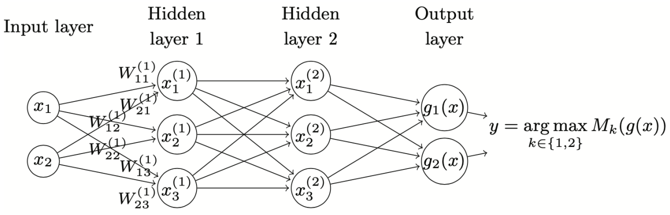
$$\mathbf{x}^{(\ell)} \ = \ \sigma\left(\mathbf{v}^{(\ell)} + W^{(\ell)} \cdot \mathbf{x}^{(\ell-1)}\right), \quad \ell = 1, 2, \dots, L$$

- in the final step: output $y = g(\mathbf{x}) = W^{(L+1)}\mathbf{x}^{(L)}$

Illustration of neural networks

- any $g \in \mathcal{F}(L, \mathbf{p})$ is built by alternating matrix-vector multiplications with the action of the non-linear activation function σ
- in any step, the initial value $\mathbf{x}^{(0)} = \mathbf{x} \in \mathbb{R}^d$ is updated via

$$\mathbf{x}^{(\ell)} \;=\; \sigma\left(\mathbf{v}^{(\ell)} + W^{(\ell)} \cdot \mathbf{x}^{(\ell-1)}\right), \quad \ell = 1, 2, \ldots, L$$



Special types of neural networks

$$f(\mathbf{x}) \;=\; \rho W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \,\cdots\, W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}$$

- network is called **sparse** if the matrices W_i are sparse
- the i-th layer is **fully connected** if W_i is dense (typically, all entries are non-zero)
- for $L = 1$, the network (ρ the identity) coincides with shallow networks

Special types of neural networks

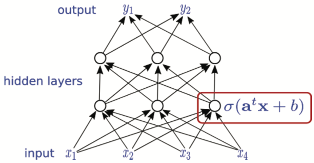
$$f(\mathbf{x}) \;=\; \rho W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \;\cdots\; W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}$$

- network is called **sparse** if the matrices W_i are sparse
- the i-th layer is **fully connected** if W_i is dense (typically, all entries are non-zero)
- for $L = 1$, the network (ρ the identity) coincides with shallow networks
- if $L > 1$, the network is called **deep**

Graph representation

In computer science, neural networks usually are introduced via some **graph representation**:

- nodes in the graph (also called **units**) are arranged in layers, where the first layer is called **input layer**, the last layer **output layer**, and layers that lie in between are referred to as **hidden layers**
- number of hidden layers corresponds to L, number of units in each layer generates the width vector $\mathbf{p}$
- each node/unit in the graph representation stands for the operation $\sigma(\mathbf{a}^\top \ \cdot + b)$



: Source: Johannes Schmidt-Hieber, Statistics for deep neural networks

Universal approximation property

Theorem

Feedforward networks with a single layer are dense in the set of continuous functions on compacts (but the layer may be infeasibly large).

Universal approximation property

Theorem

Feedforward networks with a single layer are dense in the set of continuous functions on compacts (but the layer may be infeasibly large).

4th Idea

For complex enough ANN, the approximation error $\inf_{f\in\mathcal{F}}\mathcal{R}(f)$ small. (But keep 2nd idea in mind!)

Sections

1 What is this lecture about?

2 Framework for learning

3 Some ideas from classical statistical learning

4 Artificial neural networks

5 Statistical learning for ANN

Classical approaches

- **Overparameterization**: Today, ANNs with $> 10^8$ parameters are used.
  - huge VC-dimension!
  - usually: theoretical empirical minimum $\inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) = 0$ with many minimizers.

Classical approaches

- **Overparameterization**: Today, ANNs with $> 10^8$ parameters are used.
    - huge VC-dimension!
    - usually: theoretical empirical minimum $\inf_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f) = 0$ with many minimizers.
- **Non-convexity**: Many local minima
    - Usually very suboptimal solutions

Classical approaches

- **Overparameterization**: Today, ANNs with $> 10^8$ parameters are used.
    - huge VC-dimension!
    - usually: theoretical empirical minimum $\inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) = 0$ with many minimizers.
- **Non-convexity**: Many local minima
    - Usually very suboptimal solutions
- **high dimensional spaces**: Often $dim(\mathcal{X}) > 10^4$
    - No guarantee for small approximation error $\inf_{f\in\mathcal{F}} \mathcal{R}(f)$

Classical approaches

- **Overparameterization**: Today, ANNs with $> 10^8$ parameters are used.
  - huge VC-dimension!
  - usually: theoretical empirical minimum $\inf_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f) = 0$ with many minimizers.
- **Non-convexity**: Many local minima
  - Usually very suboptimal solutions
- **high dimensional spaces**: Often $dim(\mathcal{X}) > 10^4$
  - No guarantee for small approximation error $\inf_{f \in \mathcal{F}} \mathcal{R}(f)$
- **deep networks**: Often deep networks are used ($>$ 100 hidden layers)
  - Classical theory cannot explain why this is beneficial.

(Very partial) mathematical results

- Stochastic gradient decent seems to prefer **nice optima** (effectively smaller hypothesis set).

(Very partial) mathematical results

- Stochastic gradient decent seems to prefer **nice optima** (effectively smaller hypothesis set).
- Despite high-dimensional spaces, the relevant information for typical learning situations seems to be stored in a **low-dimensional submanifold**.

(Very partial) mathematical results

- Stochastic gradient decent seems to prefer **nice optima** (effectively smaller hypothesis set).
- Despite high-dimensional spaces, the relevant information for typical learning situations seems to be stored in a **low-dimensional submanifold**.
- Relevant lass of functions represented by the neural network **increases exponentially in depth**, but only linearly in width (in some sense).

(Very partial) mathematical results

- Stochastic gradient decent seems to prefer **nice optima** (effectively smaller hypothesis set).
- Despite high-dimensional spaces, the relevant information for typical learning situations seems to be stored in a **low-dimensional submanifold**.
- Relevant lass of functions represented by the neural network **increases exponentially in depth**, but only linearly in width (in some sense).
- ...