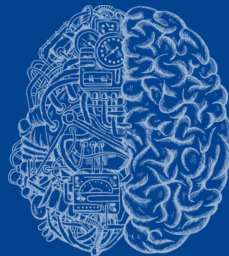


MACHINE LEARNING

LESSON 1: Introduction

CARSTEN EIE FRIGAARD
HENRIK DANIEL KJELDSSEN
AUTUMN 2019



Undervisere

Carsten Eie Frigaard:
kursusholder,
rum: E311,
email: cef@ase.au.dk



Henrik Daniel Kjeldsen:
kursusholder,
rum: E301,
email: hdk@ase.au.dk

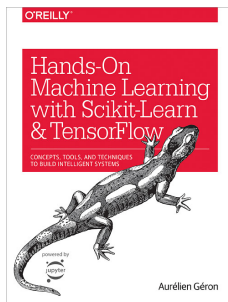


Peter Ahrendt
hjælpeleærer,
rum: E308,
email: pah@ase.au.dk



Litteratur

"Hands-On Machine Learning with Scikit-Learn and TensorFlow", Aurélien Géron, O'Reilly, 2017



- ▶ [HOML] → udtales (Brian) Holm!
- ▶ [HOML] i ny version, forår 2019, som *ikke* bruges.
- ▶ Plus yderligere materiale (brug links i BB).
- ▶ NB: **ITMAL helt nyt kursus**; fejl og mangler påpeges!



Læringsmål

Når kurset er afsluttet forventes det at den studerende kan:

- ▶ **Gengive** udvalgt machine learning historie og teori, og dets betydning for praktisk anvendelse.
- ▶ **Redegøre** hardwarens betydning for machine learning algoritmer.
- ▶ **Anvende** udvalgte machine learning teknikker i praktiske opgaver og projekter.
- ▶ **Anvende** udvalgte kodebiblioteker (frameworks) og værktøjer til machine learning.
- ▶ **Beskrive** betydningen af data kvalitet i machine learning, samt anvende udvalgte data-behandlingsteknikker.
- ▶ **Sammenligne og vurdere** forskellige algoritmer og teknikkers anvendelighed i forbindelse med praktiske projekter.
- ▶ **Diskutere** litteratur om machine learning og vurdere materialets teoretiske og praktiske anvendelses muligheder.

Eksamen

Prøveform

- ▶ Aflevering og godkendelse af alle journaler.

Bedømmelse

- ▶ Godkendt/Ikke godkendt, ingen censur.

Forudsætninger for prøvedeltagelse

- ▶ For at kunne bestå kurset skal der i løbet af semesteret være afleveret et antal obligatoriske opgaver. Der vil være deadlines for afleveringen af de enkelte opgaver.

Bemærkninger

- ▶ Beståelsen af kurset sker på baggrund af én samlet vurdering af de afleverede opgaver, hvor der vil blive lagt vægt på, om den studerende opfylder punkterne i kvalifikationsbeskrivelsen. Bedømmelsen foretages kun af eksaminator (underviser).

Reeksamen

- ▶ Næste ordinære eksamen. Samme procedure som ved den ordinære eksamen. Der skal afleveres nye opgaver til eksaminationen.

Syllabus

Preliminary...

- L01: Intro.
- L02: End-to-end demo.
- L03: Classification.
- L04: Training.
- L05: Regularization and Searching.
- L06: Reverse engineering of Learning.
- L07: Breaking the curse of dimensionality.
- L08: Deep learning I.
- L09: Adversarial examples.
- L10: Deep learning II.
- L11: Frameworks and Hardware + J3 Project.
- L12: J3 Project (until L16).

ITMAL Nomenklatur

[HOML]: Hands-On Machine Learning bog, aka (B.)Holm.

[GITHOML]: Git repository for [HOML].

[GITMAL]: Git repository for ITMAL kursus opgaver,
opdater for hver ny lektion!

[G]: ITMAL gruppe, med tre studerende, (evt. fire).

[SG]: ITMAL super-gruppe, ved nogle af opgaverne.

[J1]: Journal 1, osv. (J2/J3).

[L01]: Lektion 1, osv.

NOTE: se fuld liste på '*BB / General / Nomenclature*'.

Journalafleveringer: J1, J2 og J3

J1: Q-Opgavesæt (jupyter notebooks).

J2: Q-Opgavesæt (jupyter notebooks).

J3: Mini-projekt:

- ▶ For the final journal, you must design and implement a full machine learning system. You have relative free hands...

Criteria (extract):

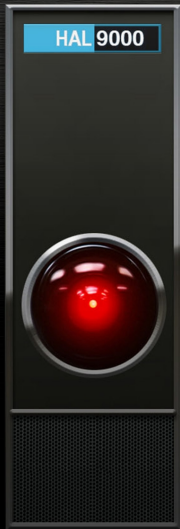
- ▶ Data must be split in a training-test set...
- ▶ Your machine learning algorithm must be described in depth...
- ▶ The system must be evaluated via a suitable performance metric-..

NOTE₀: Afleveringsformat frit (PDF, .ipynb, etc.).

NOTE₁: J3 vil blive specificeret på BB, med projektforslag.

NOTE₂: J3 konflikt med BA projekter?

END Kursus intro/BEGIN ML intro



python Introduction

- ▶ Python is an **interpreted** high-level programming language for general-purpose programming. Created by **Guido van Rossum** and first released in 1991, Python has a design philosophy that emphasizes **code readability**, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.
- ▶ Python features a **dynamic type system and automatic memory management**. It supports multiple programming paradigms, including **object-oriented**, **imperative**, **functional** and **procedural**, and has a large and comprehensive standard library.
- ▶ Python interpreters are available for many operating systems.



Anaconda and Jupyter Introduction



- ▶ Anaconda: a python distribution [<https://www.anaconda.com>].
- ▶ Jupyter notebook: interactive python development environment (GUI IDE), distributed with the Anaconda package.
- ▶ Jupyter is an anagram of: Julia, Python, and R
- ▶ Jupyter notebook method:
 - ✓ polyglot environment, mixing source code, markdown test and formulas (LaTeX),
 - ✓ interactive trial-and-error environment,
 - ÷ not good at source-code level debugging.
- ▶ Other IDE's:
 - ▶ Spyder (Anaconda),
 - ▶ VSCode (Microsoft),
 - ▶ and many others...

Scikit-learn Introduction

- ▶ Scikit-learn: a page/site for machine Learning in python.
- ▶ <http://scikit-learn.org>
- ▶ [git@github.com:scikit-learn/scikit-learn.git](https://github.com/scikit-learn/scikit-learn)



scikit learn

Home Installation Documentation Examples

Google Custom Search

Fork me on GitHub

scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Vores videnskabelige framework

Sat sammen...

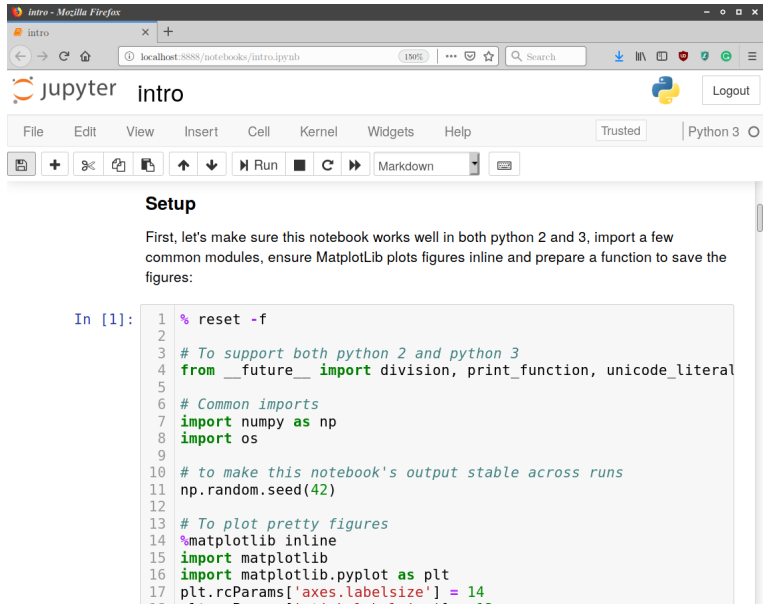


Gode hjælpe og dokumentations-systemer..

Alternativer kunne være...



Anaconda and Jupyter Demo



intro - Mozilla Firefox

intro

localhost:8888/notebooks/intro.ipynb

jupyter intro

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Setup

First, let's make sure this notebook works well in both python 2 and 3, import a few common modules, ensure Matplotlib plots figures inline and prepare a function to save the figures:

```
In [1]: 1 % reset -f
2
3 # To support both python 2 and python 3
4 from __future__ import division, print_function, unicode_literals
5
6 # Common imports
7 import numpy as np
8 import os
9
10 # to make this notebook's output stable across runs
11 np.random.seed(42)
12
13 # To plot pretty figures
14 %matplotlib inline
15 import matplotlib
16 import matplotlib.pyplot as plt
17 plt.rcParams['axes.labelsize'] = 14
```

Anaconda and Jupyter Demo: Highlights...

- ▶ Polyglot miljø:
 - ▶ lidt ala Matlab IDE,
 - ▶ markdown (HTML+LaTeX)-og-Python-i-éen = polyglot,
 - ▶ alt kører i browser, lokalt eller på server.
- ▶ Quickstart:
 - ▶ åbn via `http://localhost:8888` (efter launch),
 - ▶ ENTER på celle: editér celle,
 - ▶ CTRL+ENTER: kør celle.
 - ▶ SHIFT+TAB: hjælp på funktion.
- ▶ Magics:
 - ▶ nulstil vars: `% reset -f`,
 - ▶ inline plots: `% matplotlib inline`.
- ▶ Hints:
 - ▶ Pas på globale vars (igen scopes ml. `.ipynb` celler),
 - ▶ Brug menu *'Help'* og
find shortcuts i *'open command palette'*n,
 - ▶ Hvis du er C++ haj: alt er anderledes!

Machine learning taksonomi

- ▶ Læringsstyper:
 - ▶ supervised (mest om dette i ITMAL),
 - ▶ unsupervised,
 - ▶ [semisupervised], [reinforced learning].
- ▶ Output klasser:
 - ▶ classification (ham/spam),
 - ▶ regression ($h(x) = y$),
- ▶ Læring via data:
 - ▶ batch læring (al data),
 - ▶ [inkrementel læring (on-the-fly)].
- ▶ Prediktions/generaliserings model:
 - ▶ model-based (pattern-detection, byg intern model),
 - ▶ [instance-based (lær al data udenad)],
- ▶ Typiske ML fejl klasser:
 - ▶ for lidt trænings data (small-data, brug cross-validation),
 - ▶ sampling noise, sampling bias (ved manglende stratificering),
 - ▶ outliers og dårlig data (i big-data),
 - ▶ model og algoritme fejl: underfitting/overfitting.

Machine learning terminologi

\mathbf{X}, \mathbf{x} : input data matrix og vektor,

\mathbf{y}, y : output data vektor og skalar,

θ : model parametre,

h : hypothesis funktion; typer af ML algos:

Bayes classifier, k-Nearest Neighbors, Linear Reg., Logistic Reg., SVM,
Decision Trees, Random Forest, Neural Networks, k-Means, ...

y_{true} : ground truth, til supervised learning,

y_{pred} : predikteret værdi, aka \hat{y} ,

attribut: data type, f.eks. salgspris, dog anvendes
'feature' typisk i stedet for attribut!

feature: data attribut plus value, f.eks. $\lambda_{\text{salgspris}} = \42 ,

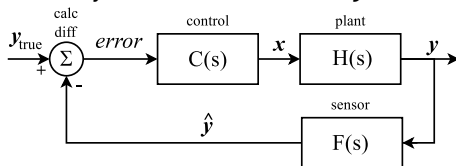
loss fun.: loss/cost/error/objective funktion, som
minimeres i fitting, jo lavere jo bedre et fit,

score fun.: score/fitness/goodness funktion, jo højere jo
bedre, bruges typisk efter fit-minimeringen
til model inspektion og eftervalidering.

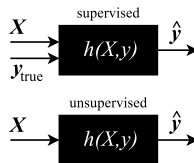
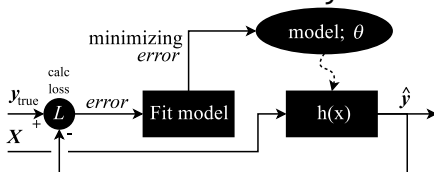
Supervised learning, blok diagram

Fra white-box til black-box

Almindelig white-box negativ feedback control block diagram, som for linear og tids-uafhængige lineære funktioner kan Laplace analyseres 'i det uendelige':



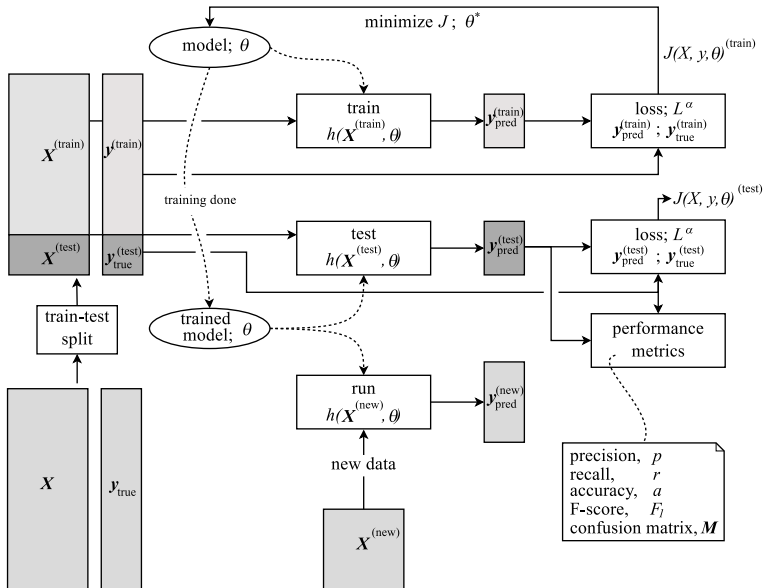
Supervised machine learning block diagram:



Valg af: Loss funktion, model/hypothesis funktion, that's is! (excl. hyperparametre). Alt er nu black-box.

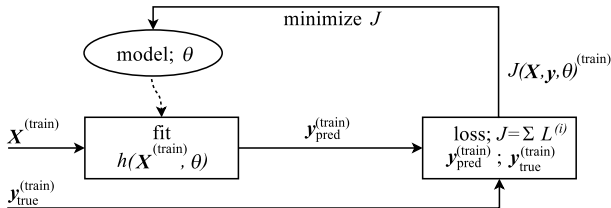
Supervised learning, blok diagram

Sneak-preview of 'the full monty'...



Q-Øvelse

ML supervised learning data flow model: Training (fit).

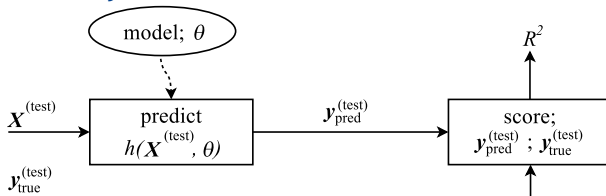


- $\mathbf{X}^{(\text{train})}$: trænings matrix input data,
 $\mathbf{x}^{(\text{train})}$: data input vector; $\mathbf{x} = [x_1, x_2, \dots, x_d]$,
 $\mathbf{y}_{\text{true}}^{(\text{train})}$: trænings input ground truth vektor,
 $\mathbf{y}_{\text{pred}}^{(\text{train})}$: predikteret værdi for y , aka \hat{y}
 θ : model parametre,
 h : hypothesis funktion, aka. ML algoritmen,
 $L^{(i)}$: loss funktion (indivduel), $L^{(i)}(y_{\text{pred}}^{(i)}, y_{\text{true}}^{(i)})$
 J : loss funktion (summeret), $J = \frac{1}{n} \sum_i L^{(i)}$.

med \mathbf{x} havende dimensionalitet d ... mere om denne og loss funktioner i L02.

Q-Øvelse

ML supervised learning data flow model: Prediction



Øvelse:

- ▶ træne en lineær regressions model, (Scikit-learn fit-predict interface),
- ▶ gå i detaljen med R^2 score funktionen, (NOTE: test data er lig train data for denne øvelse),
- ▶ check k-Nearest Neighbors modellen ud på data, sammenlign kNN-score med lineær regression-score.