Don't believe the hype? A hands-on introduction to machine-learning in Python

Open Workshops on Computer Based Systems Modelling

Johannes Schmidt, Johann Baumgartner

Institute for Sustainable Economic Development, BOKU, Vienna

7.05.2019

Contents - Workshop

- Day 1: Introduction to Machine Learning
- Day 2: Understanding backpropagation and Neural Networks in Python
- Day 3: An example of a practical application of Neural Networks for image recognition in Python and Reinforcement learning in Python
- Day 4: Bring your own data!

Literature

- Introduction to Machine Learning by Andrew Ng (Coursera.org)
- Deep Learning with Python by Francois Chollet

Contents - Today

- The basic concept of machine learning
- Some examples of machine learning
- Supervised, unsupervised, reinforcement learning: practical exercises
- An introduction to neural networks

Simple and hard tasks

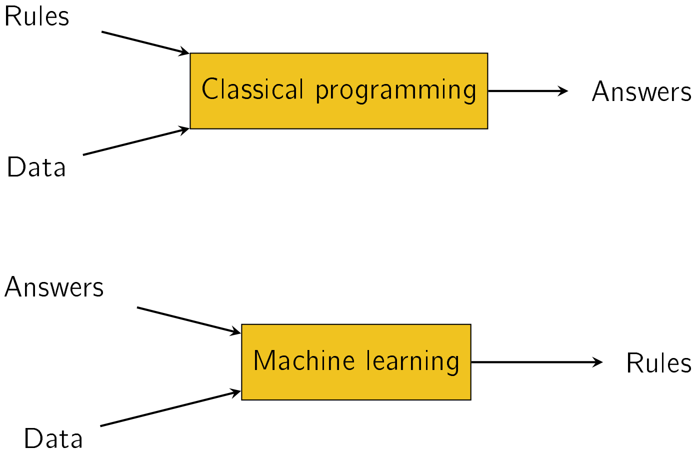
Traditionally simple tasks for a computer (but hard for humans)

- $\log \sqrt{x^3 + y * z^7}, x = 329$
- Search for all occurrences of "European Union" in a (superlong) text
- Generate a 3D scene from a mathemtical description, using raytracing
- Compile a computer program

Traditionally hard tasks for a computer (but simple for humans)

- Who is on that photo?
- Is this a cat or a dog?
- Generate a realistic image of a person
- Generate human readable text
- Translate one language to another one

The basic concept of machine learning



Why is this cool or frightening?

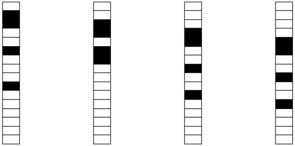
- Image recognition with captionbot.ai
- Computer plays Atari computer games
- The moment alphaGo wins against Lee Sedol
- This person does not exist
- Minority report
- Supervising Oktoberfest waiters
- Write a (boring) book
- Amazon automatically tracks and fires warehouse workers
- AI Lie detector for border control
- Twitter content filter going wrong
- Arbeitsmarktservice predicts probability of finding a job

A typology of machine learning

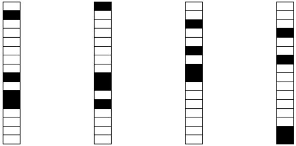
- Supervised learning: Given inputs and outputs, find the rules that link the two
- Unsupervised learning: Find structure in data
- Reinforcement learning: learning by doing.

Supervised learning - Exercise (I)

Positives



Negatives

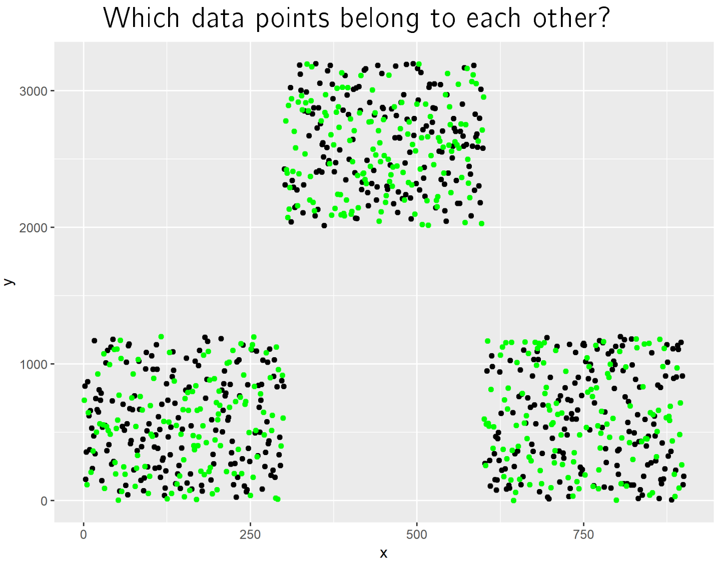


Positive or Negative?

Supervised learning needs labeled datasets

- Classification. Input features: Photos of people. Labels: smiling or not.
- Classification. Input features: Photos of waiters carrying plates and glasses. Labels: number of plates and glasses.
- Classification. Input features: German text. Labels: English text.
- Regression. Input features: wind speeds. Labels: measured wind power generation.
- Therefore: Inmates in Finland are training AI as part of prison labor

Unsupervised learning - Exercise



Unsupervised learning - Find structure in data

- Clustering (which data belongs together?)
- Anomaly detection (which data is somehow strange?)
- Generative adversial networks (Generate photos, sounds, and text, also involves supervised learning)

Reinforcement learning - Exercise

See Netlogo

Reinforcement learning - Learning by doing

- Propose new videos on video platform to user
- Play Computer Games or Go
- Simulation of heuristic optimization of agents (e.g. on markets)

What is it?

- Google translate
- Amazon product suggestion
- Face recognition
- Facebook face recognition
- Bidding for advertisements on Taobao

An introduction to artificial neural networks

- Why neural networks?
- Basic concepts and terminology
- Backpropagation algorithm and its computational complexity
- Under- and overfitting of neural networks
- Practical aspects of machine learning

Why artificial neural networks?

- Current state of the art in pattern recognition, and language processing
- Used in supervised learning, unsupervised learning, and reinforcement learning
- Understanding neural networks will help you to understand other algorithms too.

Are they new?

- Well... not really
- 1940ies first work
- Short lived hypes in 60ies and 80ies
- 1997: US Mail uses ANNs for recognizing handwritten addresses.
- New hype starting around 2010

What are they used for?

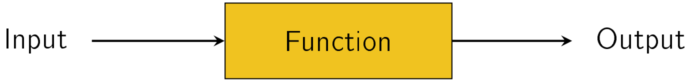
- Classification
- Regression
- Text and Image generation
- Anything, that needs the approximation of a function.

Wait... regression? Classification?

You've heart it before, but no ANNs were involved? Well, ordinary least squares regression is around since the 19th century.
Ok, so what's the difference?

- Ordinary Least Squares regression needs assumptions on functional relationships. (blackboard example)
- ANNs are *Universal Function Approximators* for linear, bounded functions. For a nice visual proof of the theorem, see here.
- ANNs **do not** allow deriving impact of input parameters on output (such as statistical significance)
- ANNs are therefore useful for prediction, but not for statistical inference
- Machine learning uses way cooler vocabulary.

Universal Function Approximator?

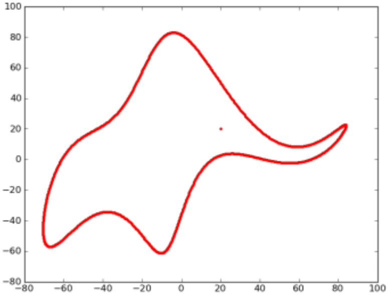


Possible Input Output relations

- German Language -> English Language
- Photos of cats and dogs -> Classification "Cat", "Dog"
- x -> x^2
- Weather data -> Wind power generation

Elephants

Wait... but what about "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." (John von Neumann)



Fit the elephant in python with four complex numbers

So true: the art of ANNs lies in achieving reasonable fits without **overfitting**.

Basic terminology

- Input layer (features)
- Hidden layer
- Output layer (labels)
- Nodes
- Connections
- Weights
- Bias node
- Activation function
- Error measure

Backpropagation algorithm

- Weights are adapted, so that the prediction and the observation error is minimized.
- Non-convex optimization due to non-convex activation functions: heuristic approach
- Training neural networks is NP-Complete in its most general form.

Terminology data sets and training

- Sample
- Training data
- Validation data
- Test data
- Epoch
- Batch, batch size
- Online training, micro-batch training, batch training

Underfitting

- Both training and validation data do not fit well.
- Problem: not enough data to fit to your network
- Solution: Smaller network. Or more data.

Overfitting

- Training data fits excellent, validation or test data does not.
- Problem: Network is adapted perfectly to your training data, but does not generalize.
- Solution: larger network. Add leaky ReLu layers.

Practical aspects

- Always assess variance-bias trade-off to understand under- and overfitting
- Get a GPU!
- Play around (get a GPU!)
- Choose your problems wisely. (If OLS regression works, use it, it gives much more insights.)
- Use feature engineering. With and without neural networks.

Next workshops

What we do

- Play around in netlogo to better understand backpropagation
- How to predict windpower from wind speeds
- Practical exercises in python

What to bring

- A laptop
- Python installed with the following packages:
- Netlogo installed

Thank you!

For slides and source-code, check
https://github.com/joph/Machine-Learning-Workshop
mail: johannes.schmidt@boku.ac.at