Organization + Introduction

Maschinelles Lernen -
Grundverfahren WS20/21

Prof. Gerhard Neumann
Autonome Lernende Roboter (ALR)
KIT, Institut für Anthrophomatik und Robotik

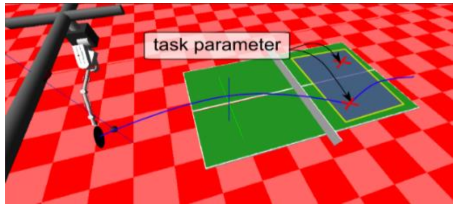
About me and ALR…

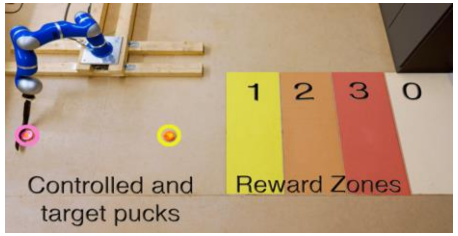
Prof. Gerhard Neumann:

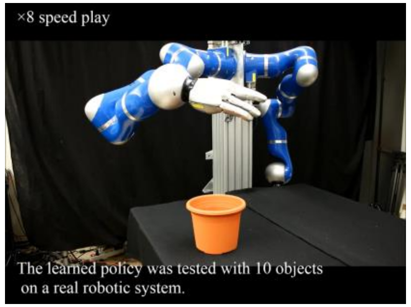
- Institut für Anthrophomatik und Robotik
- Lehrstuhl: Autonome Lernende Roboter (ALR)
- **Email: Gerhard.Neumann@kit.edu**

Research Topics: Machine Learning for Robotics

- Reinforcement Learning
- Probabilistic Machine Learning
- Deep Learning
- Interactive Learning







About me…

Timeline:

- Dissertation 2012 at the TU Graz
- 2014-2016: Junior Professor, TU Darmstadt
- 2016-2019: Professor, University of Lincoln
- 2019: Bosch Group Leader,
  "Information-theoretic Reinforcement Learning"

From 1. January 2020:

- Professorship "Autonome Lernende Roboter", KIT

Introducing the TAs

Onur Celik

celik@kit.edu

Research on:

- Hierarchical RL

Philipp Becker

philipp.becker@kit.edu

Research on:

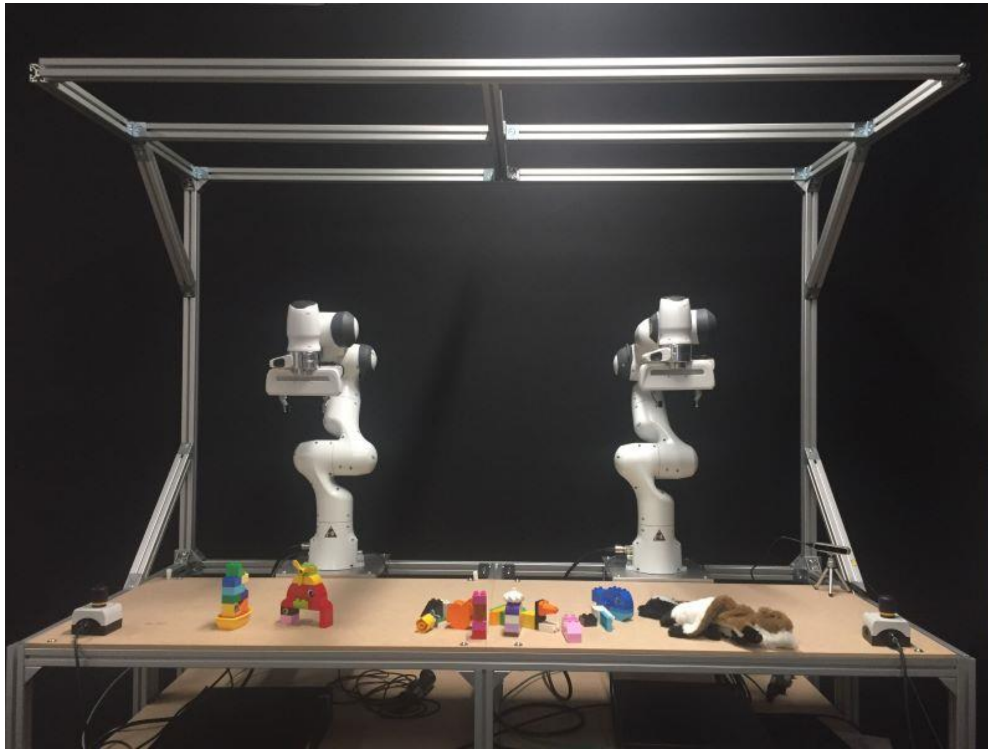
- Time Series Modelling
- Multi-modal Modelling

- Both PhD Students at ALR
- Both started together with Geri in the beginning of 2020
- Feel free to contact us over the forum or mail

A bit of self-advertisment

What else do we offer?

- Interested in a Master-Thesis or Bachelor Thesis?
  - Have a look at https://alr.anthropomatik.kit.edu/
  - We always take motivated students
  - Interesting research topics: Robot Reinforcement Learning, Deep Learning, Imitation Learning, Robotics, Human-Robot Collaboration, Variational Inference
  - Use real robots (Franka Panda arms)
  - Joint supervision of PhD students and me
  - High success-rate of turning your thesis into a paper!



Summer Semester:

- **New Lecture:** Deep Reinforcement Learning

Organization

Start of the lecture: 06.11.2020

- Friday: 14:00 – 16:30 (including 2x5 minutes break)
- Location: Zoom due to Corona
- All lectures will be recorded and put on Illias

Language:

- (Austrian) English
- Why? All the terminology / research papers are in English
- Getting used to English for these technical terms is crucial!

Exam:

- Written
- Date to be announced

Material

Lecture Material:

- Mostly slide-based
- English
- Sometimes additional lecture notes will be available (not part of exam)

Machine Learning is very math heavy!

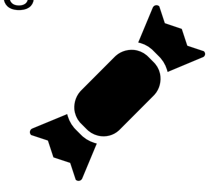
- Understand, not just apply!
- **Math basics:** We will recap the required math before it is used to derive the algorithms
- Math is directly applied… actually quite fun ☺

Exercises - General Info

There will be 6 Exercises

- 1 Exercise every 2 weeks
- Starting 20.11
- Hand in Thursday before next exercise is presented
- Solutions will be presented
- Work in groups of 3

There is a Bonus!!



- You get 0.3 bonus in the exam if you pass and have > 60% of exercise points
- There is only a joint grade of lecture and excercise

Exercises - Format

Mixture of pen and paper as well as coding

- We will use python for coding
- We will use Jupyter notebooks
- Might be a bit challenging, but you work in groups and it's a good preparation for exam

ML @ KIT

Confusion:

- Maschinelles Lernen 1 – Grundverfahren from Prof. Zöllner,
- Based on the old Machine Learning Lecture with Prof. Dillmann
- Fakultät für Wirtschaftwissenschaften, AIFB Institute
- Not applicable for Computer Science students

This lecture: Maschinelles Lernen – Grundverfahren
- New content
- Wahlfach for computer science
- More math, more theory, more programming!

ML @ KIT

Other ML lectures:

- SS: Deep Learning and Neural Networks: Prof. Waibel
- SS: Deep Learning for Computer Vision: Prof. Stiefelhagen
- WS: Optimization Methods for Machine Learning and Engineering
- SS: Cognitive Systems, Prof. Waibel and **Me**
- SS: Pattern Recognition, Prof. Beyerer

New Lecture:

- SS: Deep Reinforcement Learning, **Me**

Ask questions!!!

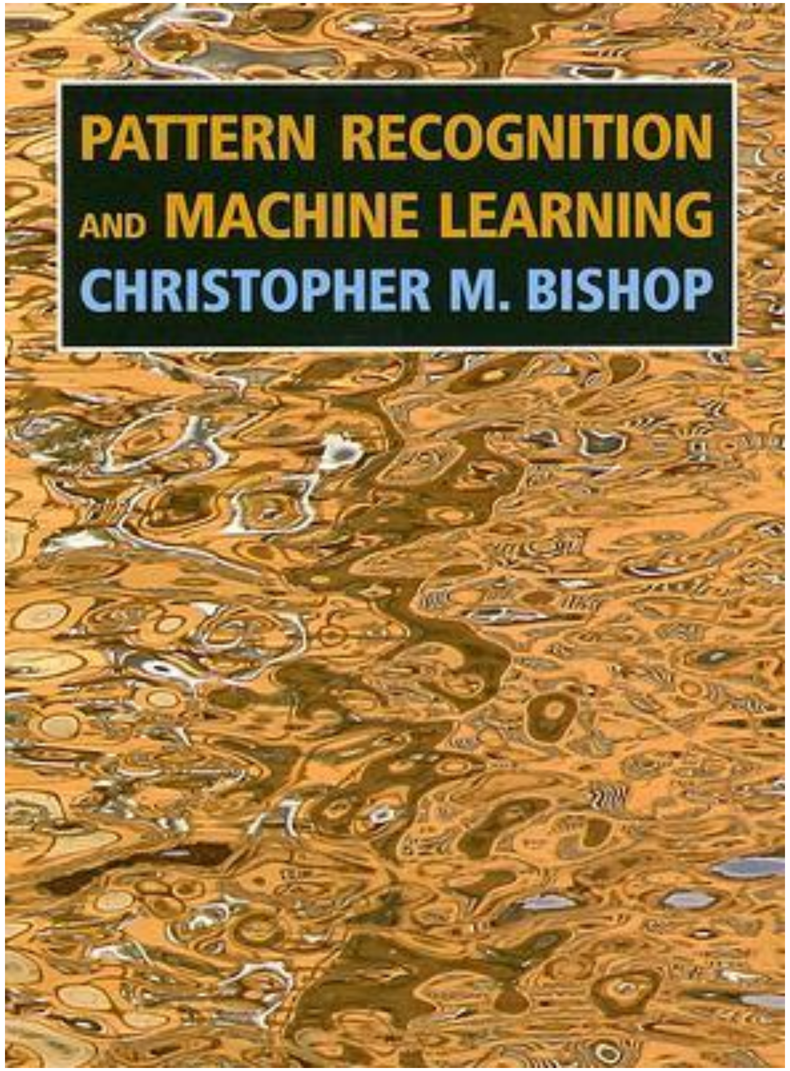
Even though I am Austrian, I am actually a nice guy…

- If it is not clear… tell me!!
- If it is too fast… tell me!!
- If you can not understand „austrian english“ … tell me!

Additional Reading

Pattern Recognition and Machine Learning

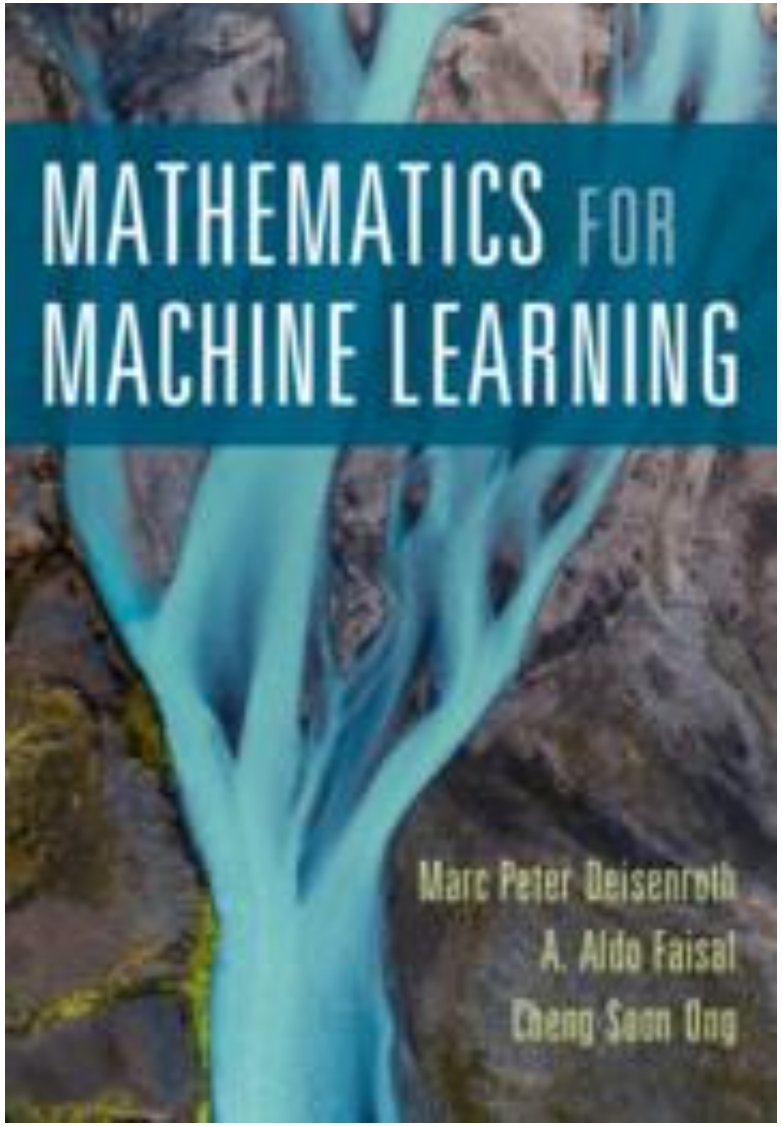
- Christopher Bishop, Springer, 2006
- Very nicely explained fundamentals in classification and regression
- PDF online



Additional Reading

Mathematics for Machine Learning

- Marc Deisenroth, Aldo Faisal and Cheng Ong
- Cambridge Press 2020
- Available as PDF online



A new friend…

The matrix cookbook

- Great collection of matrix identities
- We will use a few of them

Agenda for today

Lets take it easy...

- Introduction in Machine Learning
- Basics: Linear Algebra (essential for next lecture)
- Zeroth Excercise (Philipp)

Introduction in Machine Learning

What is learning?

- *“Learning denotes **changes in a system** that ... enable a system to do the same task … **more efficiently** the next time.”* - Herbert Simon *(Nobel Prize in Economics)*
- *“Learning is constructing or **modifying representations** of what is being **experienced**.”* - Ryszard Michalski *(ML pioneer)*
- *“Learning is making **useful changes** in our minds.”* - Marvin Minsky (MIT)

What is Machine Learning?

Algorithms that can **improve** their **performance** using **training data**

- Typically we have a large number of **parameters**
- Learned from data

Useful if:

- **No expert knowledge available:** industrial/manufacturing control, mass spectrometer analysis, drug design, astronomic discovery
- **Black-box expert knowledge:** face/handwriting/speech recognition, driving a car, flying a plane
- **Fast changing phenomena:** credit scoring, financial modeling, diagnosis, fraud detection
- **Customization/personalization:** personalized news reader, movie/book recommendation

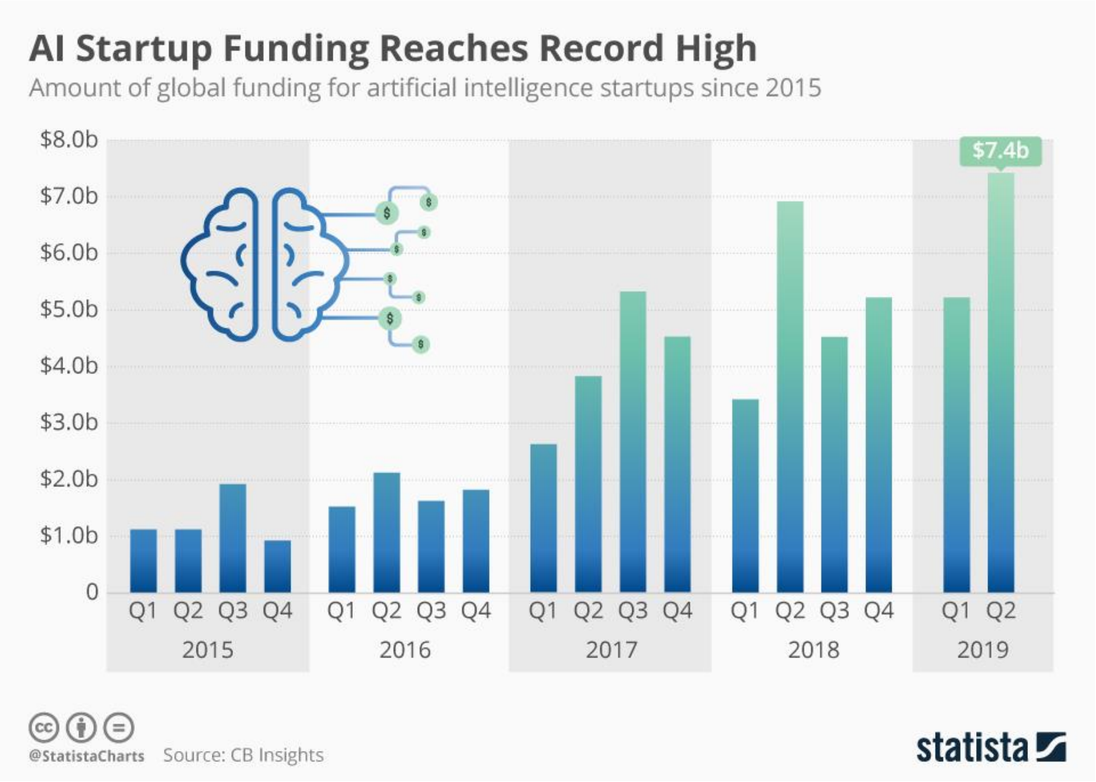
Machine Learning is a hot topic

Do we see a new era of machine learning?

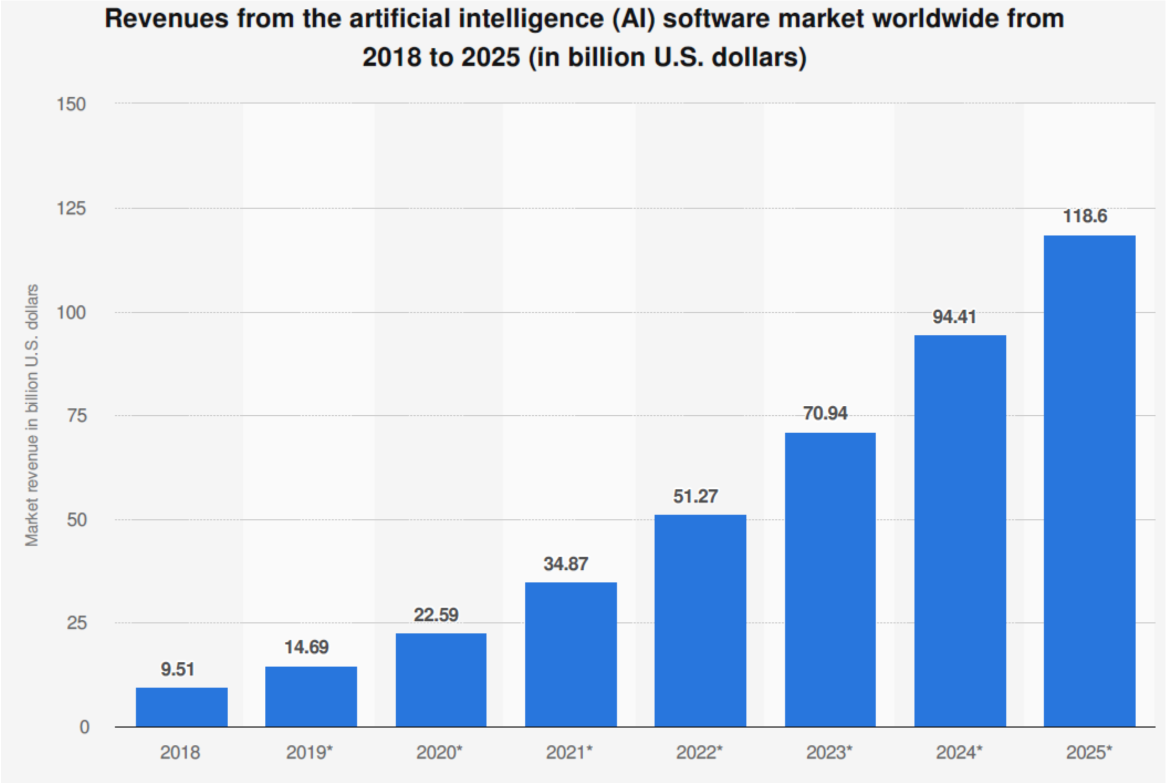
- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)

Commercial Importance

AI startup funding



Predicted Revenue of AI

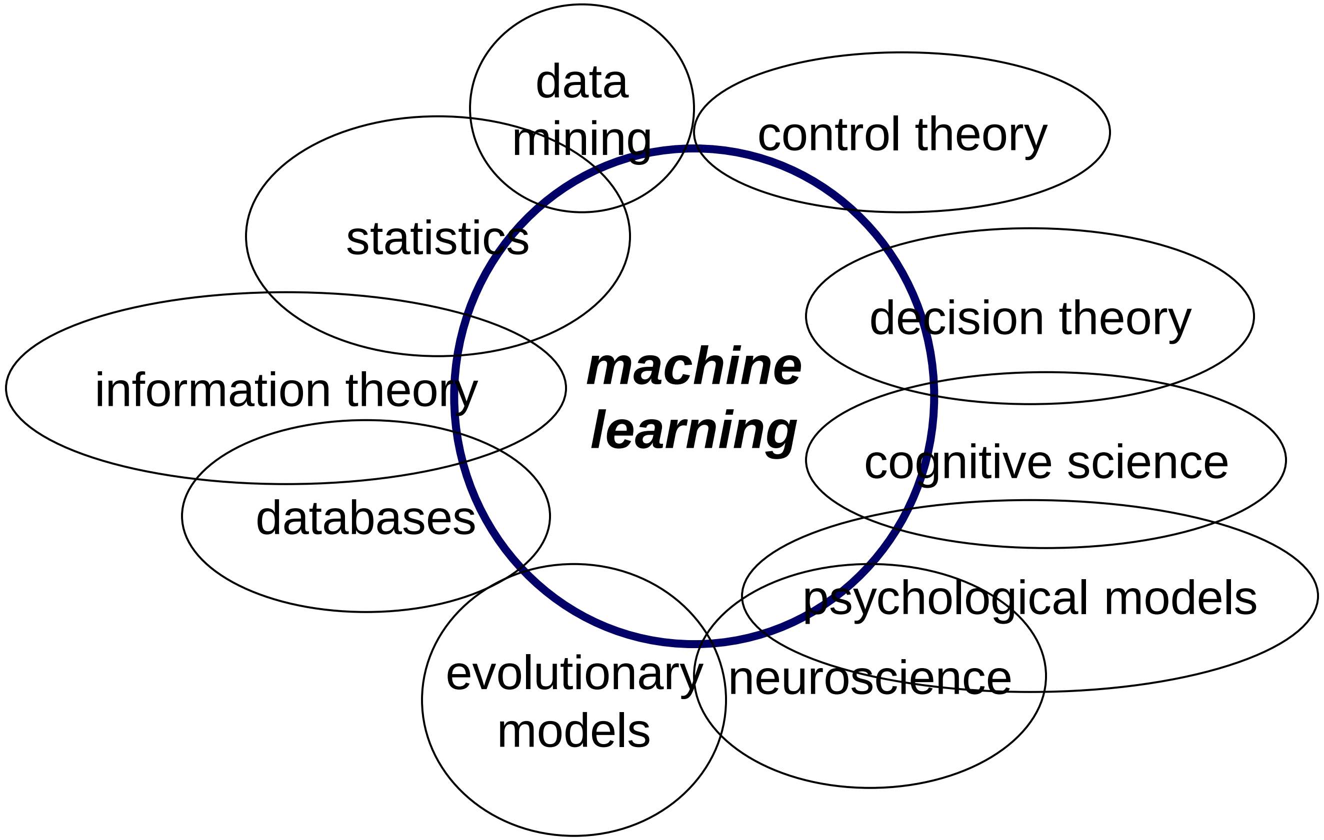


Why do we see this “explosion” now?

- More training data
- More computation power
- New algorithms (Deep Learning)…
  - But: same principles as from 40 years ago are still in use

- "when you go from 10,000 training examples to 10 billion training examples, it all starts to work. Data trumps everything.” Google translate engineer

Machine Learning is interdisciplinary



ML: uses these disciplines to create more accurate and efficient computer systems.

Different Types of Learning

1. **Supervised Learning**
   – Training data includes target values

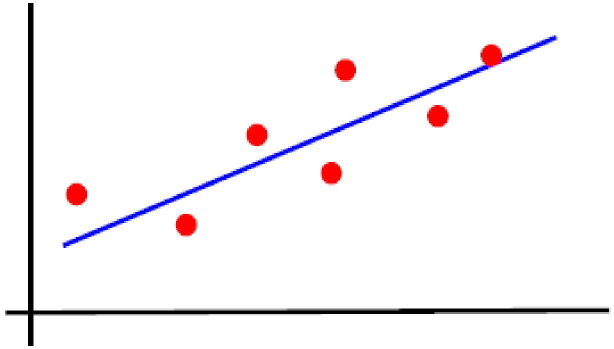
2. **Unsupervised Learning**
   – Training data does not include target values

3. **Reinforcement Learning**
   – No target values, but evaluation (reward) of the output

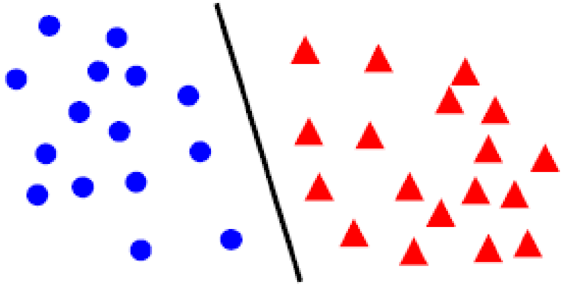
Supervised Learning

Training data includes targets

- **Regression:**
  - Learn continuous function
  - Example: line



- **Classification:**
  - Learn class labels
  - Example: Digit recognition

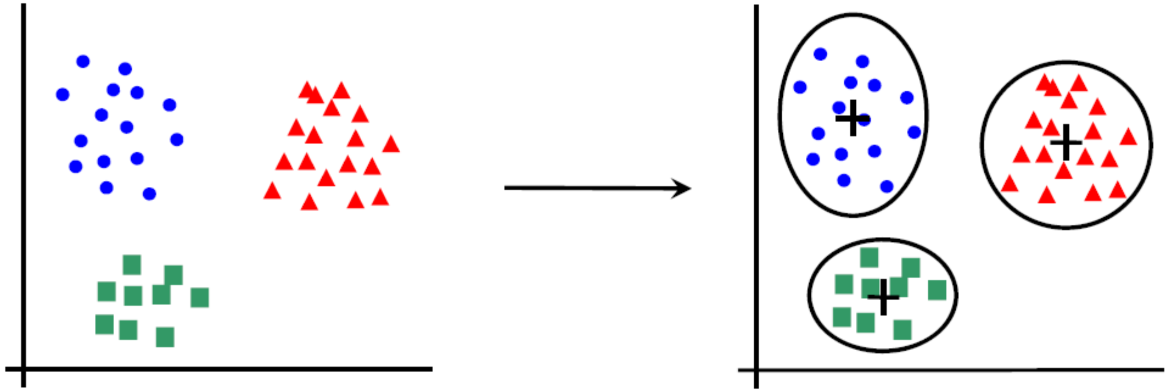


Unsupervised Learning

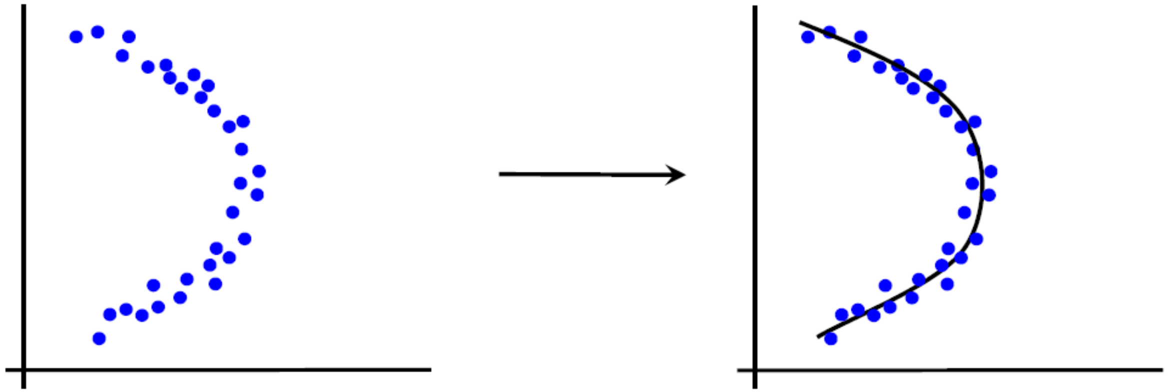
Training data does not include target values

- Model the data

- **Clustering:**



- **Dimensionality reduction:**

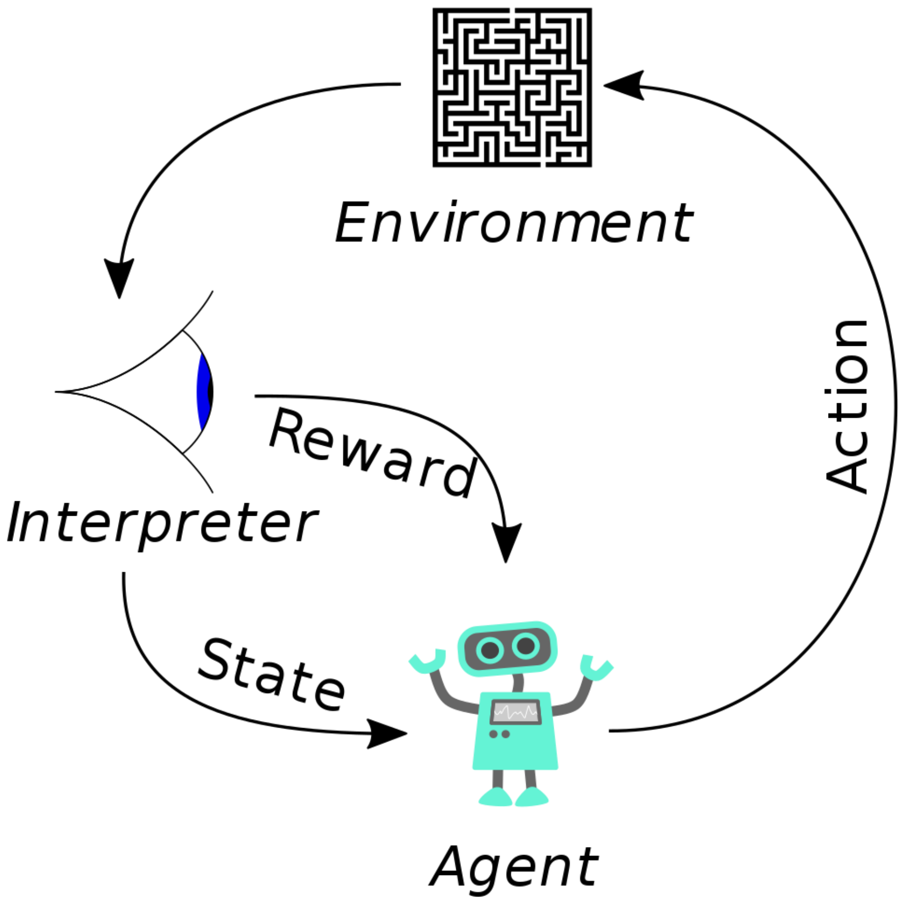


Reinforcement Learning

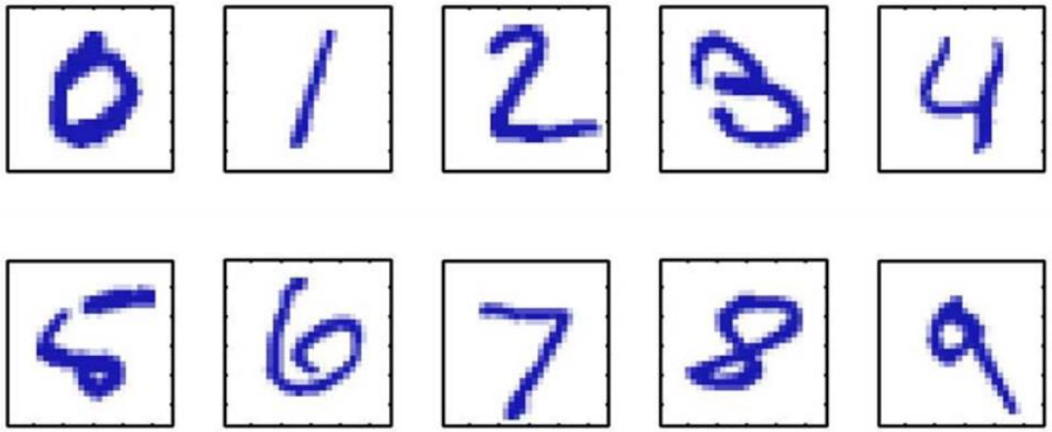
- No supervisor, but reward signal
- Selected actions also influence future states

Not part of this lecture!

… but stay tuned: New lecture in next semester!



Example 1: hand-written digit recognition



Images are 28 x 28 pixels

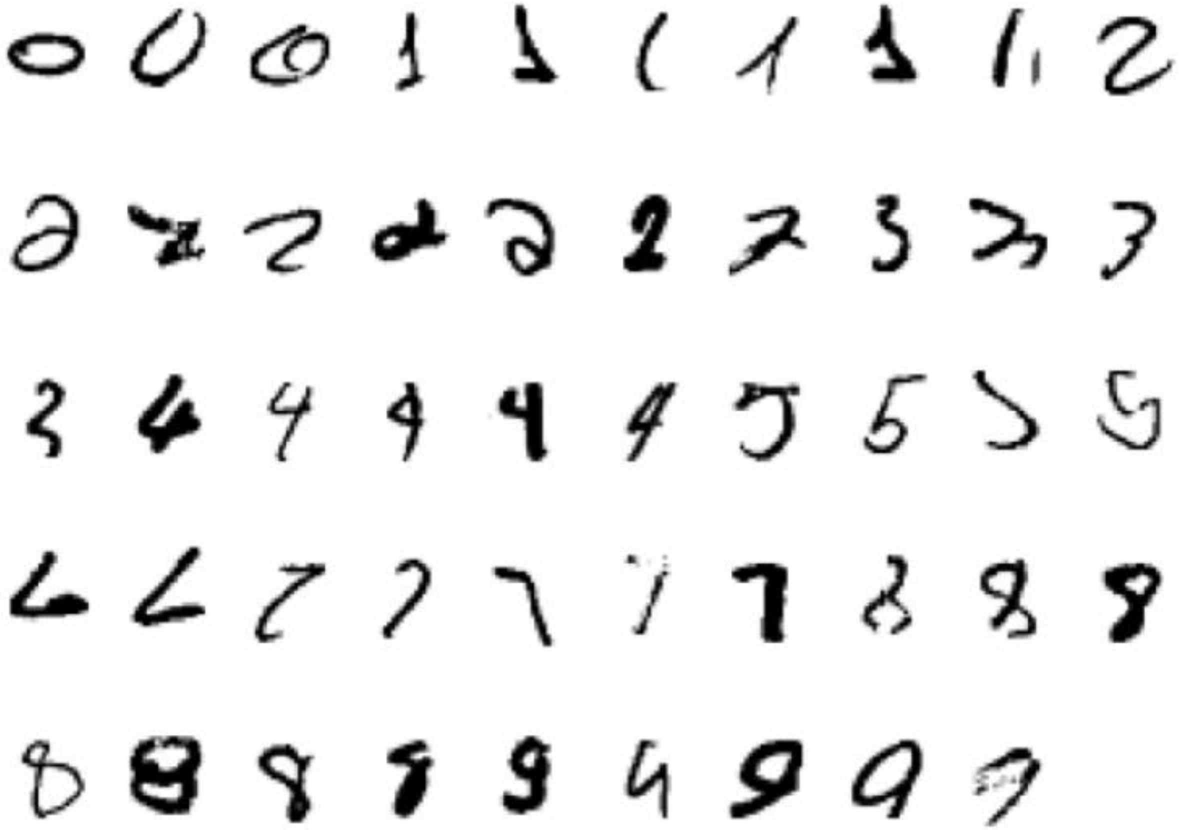
- Represent image as vector
- Learn Classifier

$$\mathbf{x} \in \mathbb{R}^{784}$$

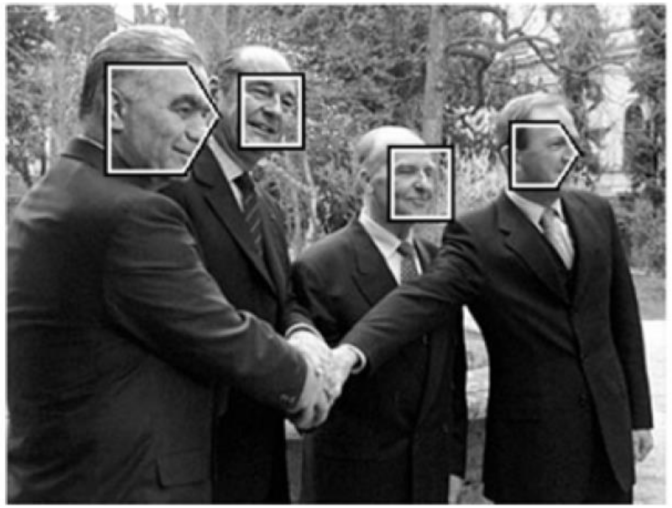
$$f : \mathbf{x} \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Example 1: hand-written digit recognition

- Supervised Classification Problem
- Training-Set: 6000 examples per class
- Error on a test-set: 0.4%
- One of the first applications used in a commercial product (ZIP-Codes, cheques,…)



Example 2: Face Detection



- **Classification Problem**
- **Classify image windows in 3 classes**
  - Non-face
  - Frontal-face
  - Profile-face

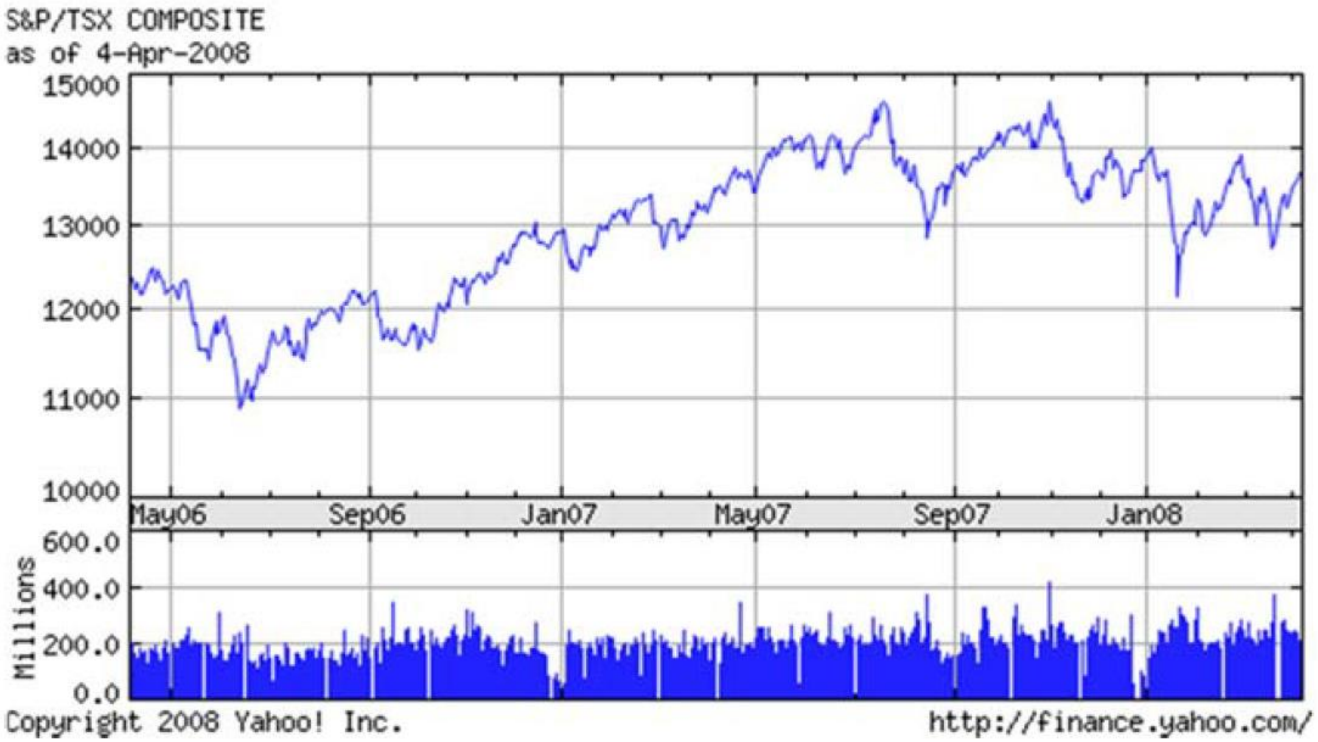
Example 2: Face Detection

Training data for frontal phases

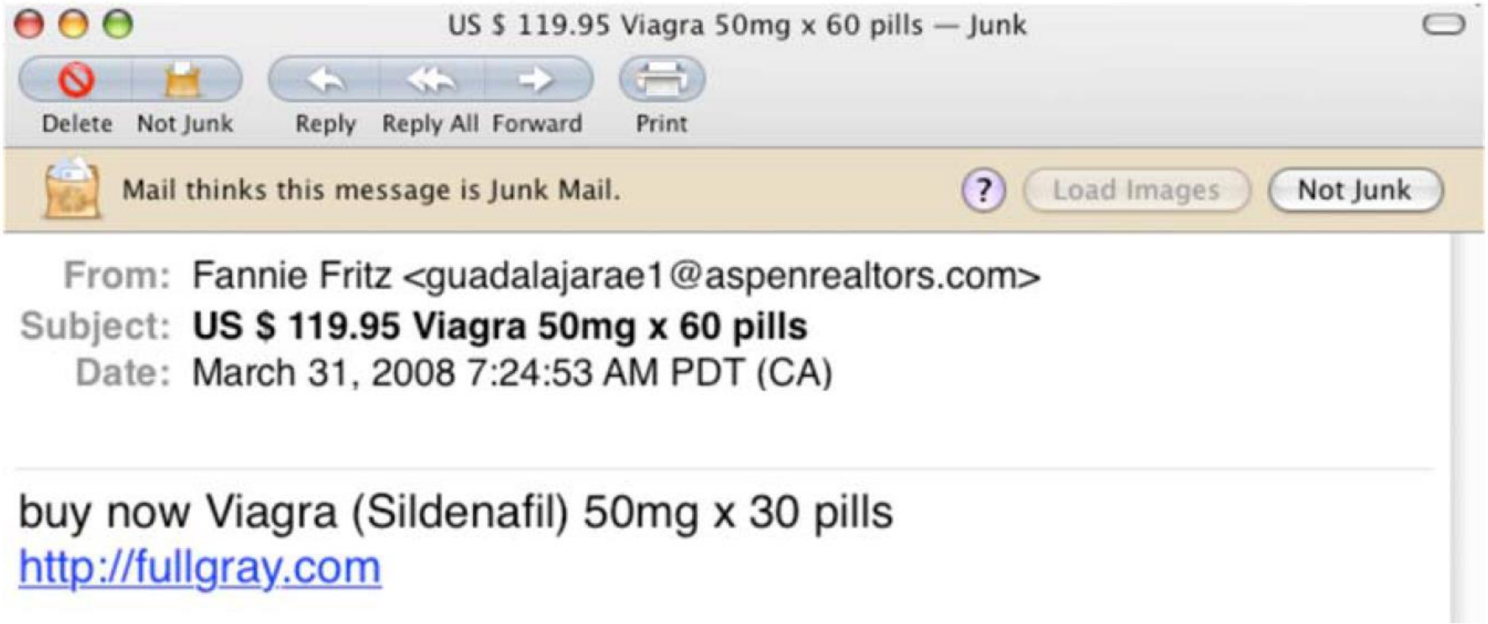
- 5000 faces
  - All near frontal
  - All ages, races, gender, lighting
- 10^8 non-faces
- Normalization (scale, translation)

Example 3: Stock-price prediction

- Prediction of the stock prices
- Regression problem
  continuous outputs)



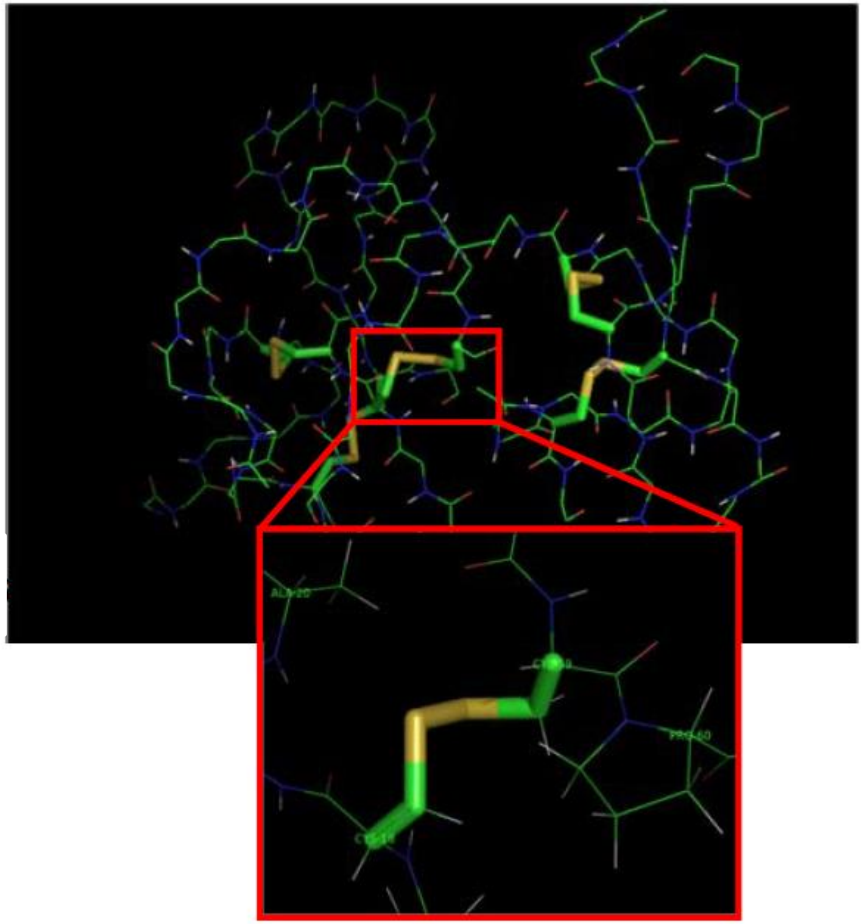
Example 4: Spam Detection



- Classify emails in spam / non-spam
- Data x represented by word count, z.B.: “Viagra”, “outperform”, “you may be surprised to be contacted”…
- Spam strategies change -> we need learning

Beispiel 5: Computational Biology

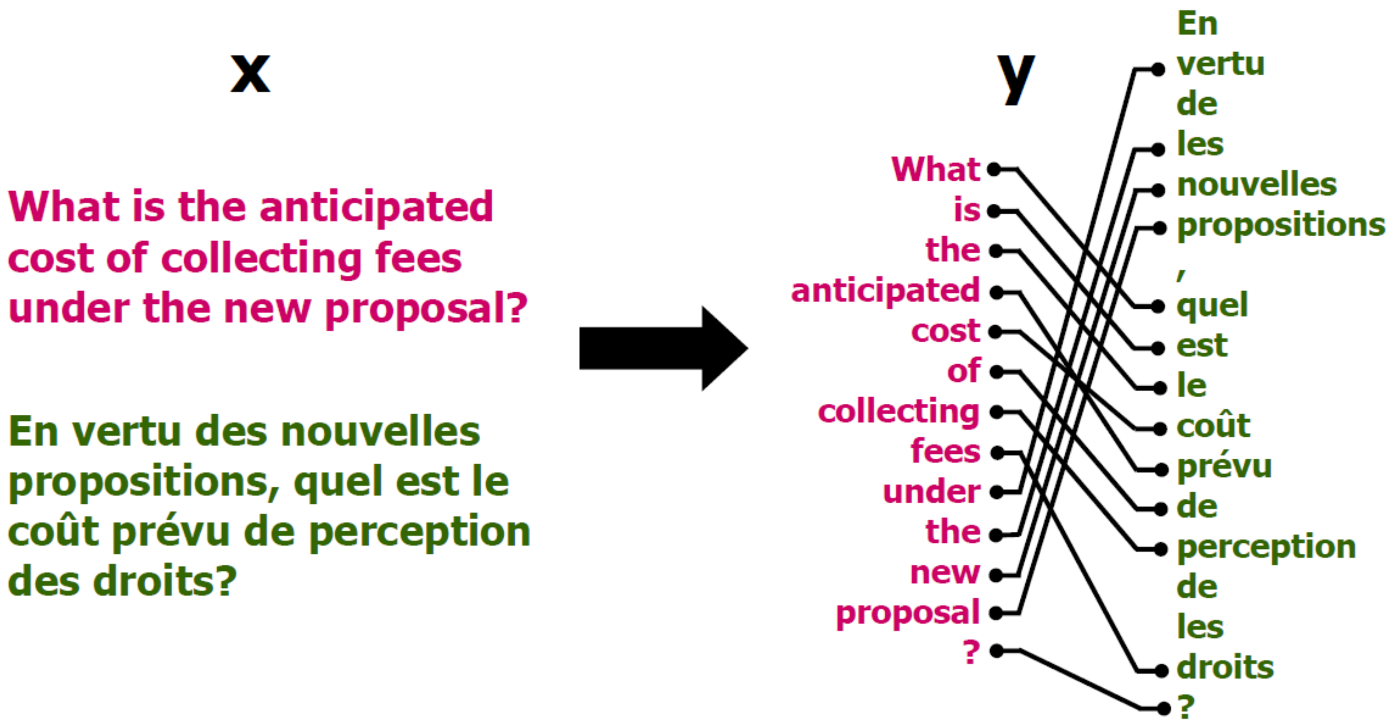
AVITGACERDLQCG
KGTCCAVSLWIKSV
RVCTPVGTSGEDCH
PASHKIPFSGQRMH
HTCPCAPNLACVQT
SPKKFKCLSK



- Given the protein sequence, predict 3D structure

Example 6: Language Translation

- Learn from aligned text translations



e.g. Google translate

Example 6: Language Translation

Example 7: Recommender Systems

Example 8: Robot Manipulation

- Reinforcement Learning: Learning to grasp unknown objects (Google)

Example 9: Tossing objects

- Reinforcement Learning: Learning to throw objects into bins

Example 10: Semantic Segmentation

Example 11: Creating Image Captions

Description:

A falcon is eating during sunset.
The falcon is standing on earth.

Poem:

Like a falcon by the night
Hunting as the black knight
Waiting to take over the fight
With all of it's mind and might

Example 12: Image generation

Figure: synthetic images generated by [pg-GAN](pg-GAN) from Nvidia.
None of these images are real!

More success stories in the last 3 years

- AI beats human in lip reading
- Improved translation with Google Neural Machine Translation System
- OpenAI Bot dominates DOTA 2 Champions
- CMU AI beats top poker players
- Google Deepmind beats world champion in “Go”

More success stories

- Deepmind reduces Google’s Data Center cooling costs by 40%
- Google Duplex: AI System performing every-days phone calls
- Tacotron 2: Generating human speech from text
- State-of-the-art speech recognition via Sequence to Sequence models

Conceptual view on ML

- Thousands of learning algorithms
- Hundreds of new algorithms every year
- We will only look at the fundamental algorithms and discuss the principles that connect them

Every ML algorithm consists of 3 parts:

- **Representation**
- **Evaluation**
- **Optimization**

Representation

How does our model look like?

- Decision trees
- Instances
- Mixture Models
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

Evaluation

What are we optimizing for ?

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- KL divergence
- Etc.

Optimization

How do we optimize?

- Least squares solution
- Gradient descent
- 2^{nd} order methods
- Constraint optimization
- Random search

What will we cover?

We will cover the fundamentals for

- **Representation**
- **Evaluation**
- **Optimization**

With a strong mathematical focus on understanding and deriving the algorithms

More explicitly

- Lecture 1: Introduction
- Lecture 2: Linear Regression
- Lecture 3: Linear Classification
- Lecture 4: Model Selection
- Lecture 5: Nearest Neighbor & Trees
- Lecture 6: Dimensionality Reduction
- Lecture 7: Density Estimation + Clustering
- Lecture 8: Support Vector Machines
- Lecture 9: Kernel Methods
- Lecture 10: Bayesian Learning
- Lecture 11: Neural Nets 1 + (2)
- Lecture 12: Hyperparameter Optimization

What will we not cover?

- Most of Deep Learning
- Reinforcement Learning
- Genetic Algorithms
- Natural Language Processing (NLP)
- Generative Adversarial Networks
- Graphical Models
- Sampling methods
- Recommender Systems
- Topic Models
- …

Please visit the advanced courses

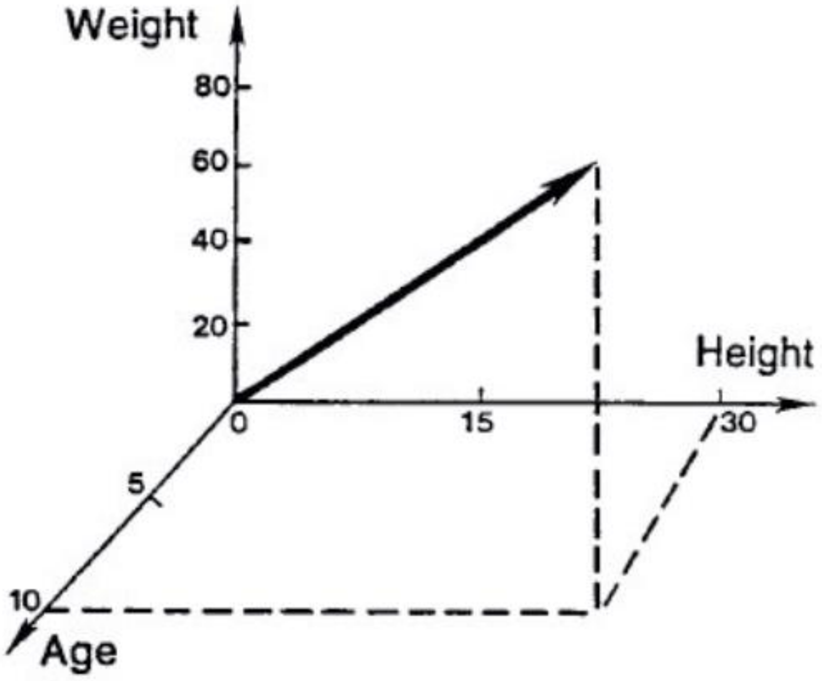
Basics: Linear Algebra

Vectors

- A vector is a multi-dimensional quantity

Joe $\begin{bmatrix} 37 \\ 72 \\ 175 \end{bmatrix}$ Mary $\begin{bmatrix} 10 \\ 30 \\ 61 \end{bmatrix}$

Carol $\begin{bmatrix} 25 \\ 65 \\ 121 \end{bmatrix}$ Brad $\begin{bmatrix} 66 \\ 67 \\ 155 \end{bmatrix}$



- Each dimension contains different information (Age, Height, Weight…)

Some notation

- Vectors will always be represented as bold symbols

$$\underbrace{x = 1}_{\text{scalar}}, \quad \underbrace{\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}}_{\text{vector}}$$

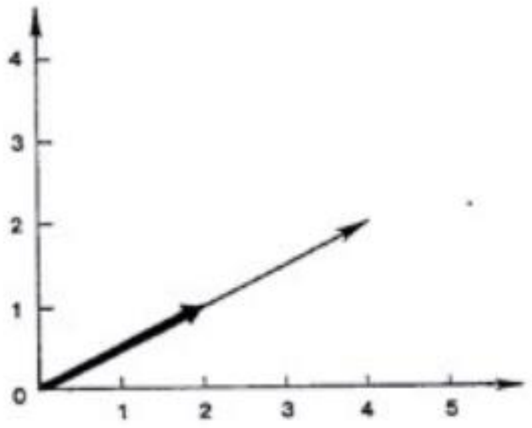
- A vector $\boldsymbol{x}$ is always a **column vector** $\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$

- A transposed vector $\boldsymbol{x}^T$ is always a **row vector** $\boldsymbol{x}^T = \begin{bmatrix} 1 & 2 & 4 \end{bmatrix}$

What can we do with vectors?

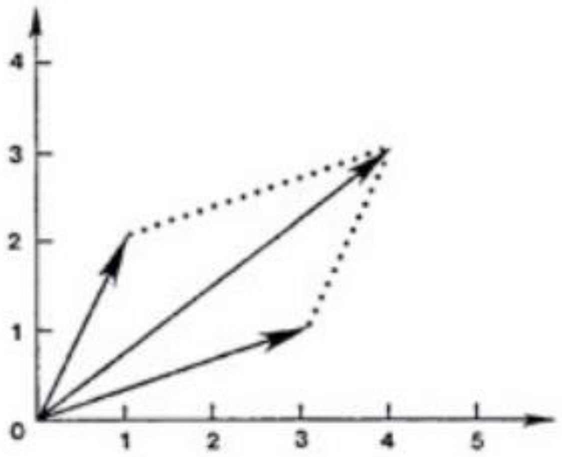
- Multiplication by scalars

$$2\begin{bmatrix}1\\2\\4\end{bmatrix} = \begin{bmatrix}2\\4\\8\end{bmatrix}$$



- Addition of vectors

$$\begin{bmatrix}1\\2\\4\end{bmatrix} + \begin{bmatrix}2\\1\\4\end{bmatrix} = \begin{bmatrix}3\\3\\8\end{bmatrix}$$



Scalar products and length of vectors

- Scalar (Inner) products:
  - Sum the element-wise products

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}$$

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = 1 \cdot 2 + 2 \cdot 4 + 4 \cdot 8 = 42$$

- Length of a vector
  - Square root of the inner product with itself

$$||\boldsymbol{v}|| = \langle \boldsymbol{v}, \boldsymbol{v} \rangle^{\frac{1}{2}} = \left(1^2 + 2^2 + 4^2\right)^{\frac{1}{2}} = \sqrt{21}$$

Matrices

- A matrix is a rectangular array of numbers arranged in rows and columns.

$$\boldsymbol{X} = \begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 4 & 7 \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & 3 & 5 & 4 \\ 2 & 3 & 7 & 2 \end{bmatrix}$$

  – $\boldsymbol{X}$ is a 3 x 2 matrix and $\boldsymbol{A}$ a 2 x 4 matrix
  – Dimension of a matrix is always *num rows times num columns*
  – Matrices will be denoted with bold upper-case letters (***A*,*B*,*W***)
  – Vectors are **special cases of matrices**

$$\underbrace{\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}}_{3\times 1 \text{ matrix}} \qquad \underbrace{\boldsymbol{x}^T = \begin{bmatrix} 1 & 2 & 4 \end{bmatrix}}_{1\times 3 \text{ matrix}}$$

Matrices in Machine Learning

- In many cases, our data set can be represented as matrix, where single samples are vectors

$$\text{Joe: } \boldsymbol{x}_1 = \begin{bmatrix} 37 \\ 72 \\ 175 \end{bmatrix} \quad \text{Mary: } \boldsymbol{x}_2 = \begin{bmatrix} 10 \\ 30 \\ 61 \end{bmatrix} \quad \text{Carol: } \boldsymbol{x}_3 = \begin{bmatrix} 25 \\ 65 \\ 121 \end{bmatrix} \quad \text{Brad: } \boldsymbol{x}_4 = \begin{bmatrix} 66 \\ 67 \\ 175 \end{bmatrix}$$

- **Most typical representation:**
  - Each row represent a data sample (e.g. Joe)
  - Each column represents a data entry (e.g. age)

⇨ ***X*** is a *num samples x num entries* matrix

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \boldsymbol{x}_3^T \\ \boldsymbol{x}_4^T \end{bmatrix} = \begin{bmatrix} 37 & 72 & 175 \\ 10 & 30 & 61 \\ 25 & 65 & 121 \\ 66 & 67 & 175 \end{bmatrix}$$

What can you do with matrices?

- Multiplication with scalar

$$3\boldsymbol{M} = 3\begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 12 & 15 \\ 3 & 0 & 3 \end{bmatrix}$$

- Addition of matrices

$$\boldsymbol{M} + \boldsymbol{N} = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 1 \\ 3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 6 & 6 \\ 4 & 1 & 2 \end{bmatrix}$$

- Matrices can also be transposed

$$\boldsymbol{M} = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix}, \ \boldsymbol{M}^T = \begin{bmatrix} 3 & 1 \\ 4 & 0 \\ 5 & 1 \end{bmatrix}$$

Multiplication of a vector with a matrix

- Matrix-Vector Product: $\boldsymbol{u} = \boldsymbol{W}\boldsymbol{v} = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 4 \cdot 0 + 5 \cdot 2 \\ 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 2 \end{bmatrix} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}$

- Think of it as: $\underbrace{\begin{bmatrix} \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}}_{\boldsymbol{v}} = \underbrace{\begin{bmatrix} v_1 \boldsymbol{w}_1 + \dots + v_n \boldsymbol{w}_n \end{bmatrix}}_{\boldsymbol{u}}$
  - Hence: $\boldsymbol{u} = v_1 \boldsymbol{w}_1 + \dots + v_n \boldsymbol{w}_n = 1 \begin{bmatrix} 3 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 4 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}$
  - We sum over the columns $\boldsymbol{w}_i$ of ***W*** weighted by v_i

- Vector needs to have same **dimensionality as number of columns!**

Multiplication of a matrix with a matrix

- Matrix-Matrix Product:

$$\boldsymbol{U} = \boldsymbol{W}\boldsymbol{V} = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 4 \cdot 0 + 5 \cdot 2 & 3 \cdot 0 + 4 \cdot 3 + 5 \cdot 4 \\ 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 2 & 1 \cdot 0 + 0 \cdot 3 + 1 \cdot 4 \end{bmatrix} = \begin{bmatrix} 13 & 32 \\ 3 & 4 \end{bmatrix}$$

- Think of it as: $\boldsymbol{W} \underbrace{\left[\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\right]}_{\boldsymbol{V}} = \left[\underbrace{\boldsymbol{W}\boldsymbol{v}_1}_{\boldsymbol{u}_1}, \dots, \underbrace{\boldsymbol{W}\boldsymbol{v}_n}_{\boldsymbol{u}_n}\right] = \boldsymbol{U}$

  - Hence: Each column $\boldsymbol{u}_i = \boldsymbol{W}\boldsymbol{v}_i$ in $\boldsymbol{U}$ can be computed by a matrix-vector product

Multiplication of a matrix with a matrix

- **Dimensions:** $$\underbrace{m \times n}_{\boldsymbol{W}} \cdot \underbrace{n \times j}_{\boldsymbol{V}} = \underbrace{m \times j}_{\boldsymbol{U}}$$
  - Number of columns of left matrix must match number of rows of right matri
- **Non-commutative (in general):** $\boldsymbol{V}\boldsymbol{W} \neq \boldsymbol{W}\boldsymbol{V}$
- **Associative:** $\boldsymbol{V}(\boldsymbol{W}\boldsymbol{X}) = (\boldsymbol{V}\boldsymbol{W})\boldsymbol{X}$
- **Transpose Product:** $(\boldsymbol{V}\boldsymbol{W})^T = \boldsymbol{W}^T\boldsymbol{V}^T$

Important special cases

- **Scalar (Inner) product:**

$$\boldsymbol{w}^T\boldsymbol{v} = [w_1, \dots, w_n] \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = w_1 v_1 + \cdots + w_n v_n = \langle \boldsymbol{w}, \boldsymbol{v} \rangle$$

  – The scalar product can be written as vector-vector product

Important special cases

- **Compute row/column averages of matrix**

$$\boldsymbol{X} = \underbrace{\begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & & \vdots \\ X_{n,1} & \dots & X_{n,m} \end{bmatrix}}_{n \text{ (samples) } \times m \text{ (entries)}}$$

  – Vector of row averages (average over all entries per sample)

$$\begin{bmatrix} \frac{1}{m}\sum_{i=1}^{m} X_{1,i} \\ \vdots \\ \frac{1}{m}\sum_{i=1}^{m} X_{n,i} \end{bmatrix} = \boldsymbol{X} \begin{bmatrix} \frac{1}{m} \\ \vdots \\ \frac{1}{m} \end{bmatrix} = \boldsymbol{X}\boldsymbol{a}, \quad \text{with } \boldsymbol{a} = \begin{bmatrix} \frac{1}{m} \\ \vdots \\ \frac{1}{m} \end{bmatrix}$$

  – Vector of column averages (average over all samles per entry)

$$\left[\frac{1}{n}\sum_{i=1}^{n} X_{i,1}, \dots, \frac{1}{n}\sum_{i=1}^{n} X_{i,m}\right] = \left[\frac{1}{n}, \dots, \frac{1}{n}\right] \boldsymbol{X} = \boldsymbol{b}^T\boldsymbol{X}, \text{ with } \boldsymbol{b} = \begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix}$$

Matrix Inverse

scalar **matrices**

- **Definition:** $w \cdot w^{-1} = 1$ $\quad \boldsymbol{W}\boldsymbol{W}^{-1} = \boldsymbol{I}, \quad \boldsymbol{W}^{-1}\boldsymbol{W} = \boldsymbol{I}$

- **Unit Element:** Identity matrix, e.g., 3 x 3: $\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- **Verify it!**

$$\boldsymbol{W} = \begin{bmatrix} 1 & \frac{1}{2} \\ -1 & 1 \end{bmatrix} \qquad \boldsymbol{W}^{-1} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}$$

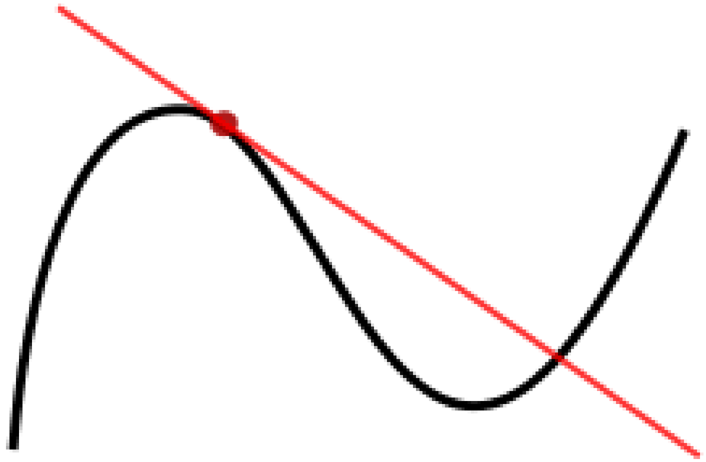
$$\boldsymbol{W}\boldsymbol{W}^{-1} = \begin{bmatrix} 1 & \frac{1}{2} \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- **Note:** We can only invert quadratic matrices (num rows = num cols)

Calculus

We also need to talk about derivatives…

“The derivative of a function of a real variable measures **the sensitivity to change of a quantity** (a function value or dependent variable) which is determined by another quantity (the independent variable)” (Wikipedia)

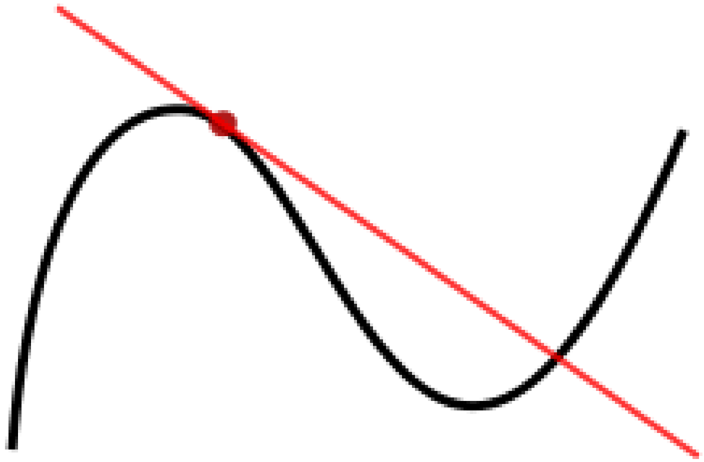


Function: $f(x)$

Derivative: $\frac{\partial f(x)}{\partial x}$

Minimum/Maximum: $\frac{\partial f(x)}{\partial x} = 0$

Derivatives and Gradients



	scalar	vector
Function:	$f(x)$	$f(\boldsymbol{x})$
Derivative:	$\frac{\partial f(x)}{\partial x} = g$	$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left[\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \dots, \frac{\partial f(\boldsymbol{x})}{\partial x_d}\right]^T$
Min/Max:	$\frac{\partial f(x)}{\partial x} = 0$	$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = [0, \dots, 0]^T = \mathbf{0}$

- $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left[\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \dots, \frac{\partial f(\boldsymbol{x})}{\partial x_d}\right]^T$ is called the gradient of function *f* at point ***x***
- We will use the „nabla“ operator as shorthand notation $\nabla f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$

Matrix Calculus

- We need to know some rules from **Matrix Calculus** (see wikipedia)

	scalar	vector
– Linear:	$$\frac{\partial ax}{\partial x} = a$$	$$\nabla_{\boldsymbol{x}} \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T$$
– Quadratic:	$$\frac{\partial x^2}{\partial x} = 2x$$	$$\nabla_{\boldsymbol{x}} \boldsymbol{x}^T\boldsymbol{x} = 2\boldsymbol{x}$$ $$\nabla_{\boldsymbol{x}} \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x}$$

The end…

- Any further questions?

Zeroth Exercise

Starting Today:

- Not Graded
- Register as Team, we created a forum to find teammates
- Familiarize yourself with python, Jupyther notebooks and the hand in process

Presented now