Introduction

A. Maier, K. Breininger, L. Mill, N. Ravikumar, T. Würfl, M. Hoffman, S. Gündel, F.Ḋenzinger, F. Thamm
Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
October 16, 2018

Who are we?

Andreas Maier

Tobias Würfl

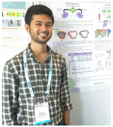
Leonid Mill

Mathis Hoffman

Sebastian Gündel

Florian Thamm

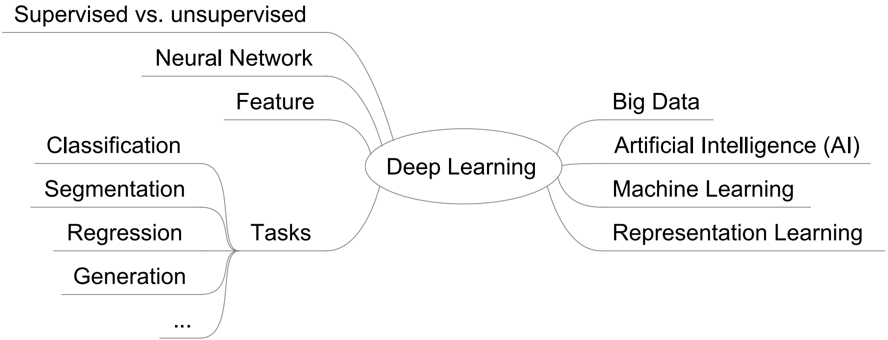
Katharina Breininger



Nishant Ravikumar

Felix Denzinger

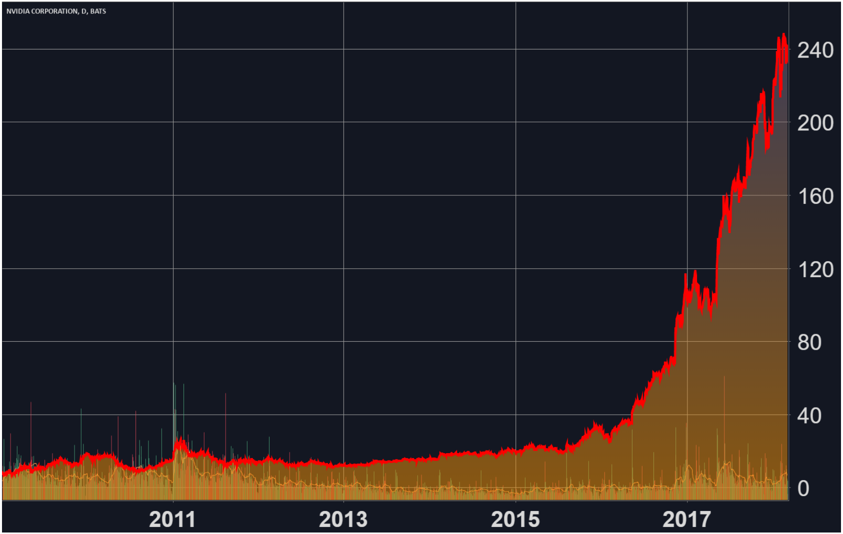
Deep Learning - Buzzwords



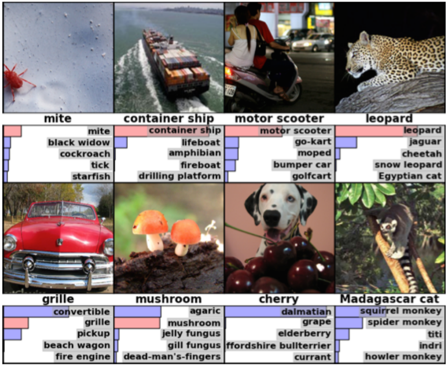
Outline

Motivation

NVIDIA Stock Market



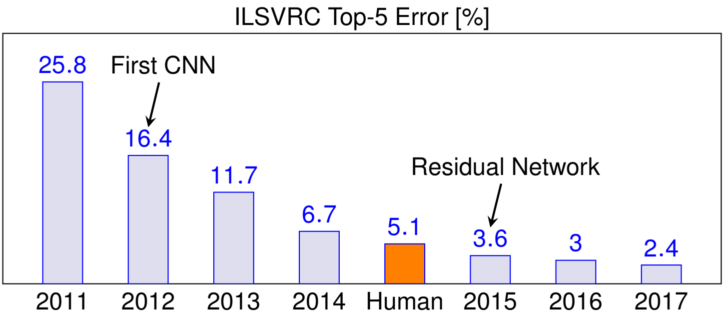
The Big Bang of Deep Learning



ImageNet [15] Dataset

- ≈ 14 mio. images, labeled into ≈ 20.000 **synonym sets**
- **I**mageNet **L**arge **S**cale **V**isual **R**ecognition **C**hallenge using ≈ 1000 classes
- **2012: Breakthrough** by Krizhevsky et al. [6]

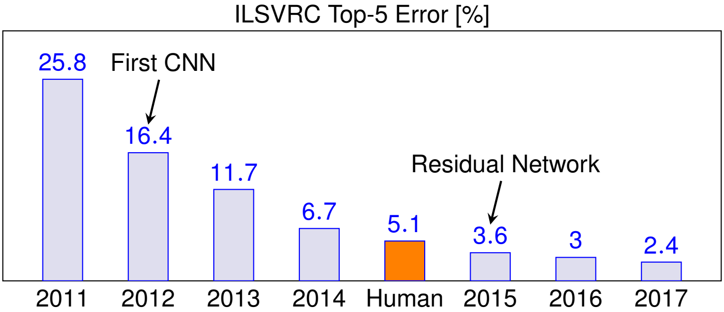
ImageNet Large Scale Visual Recognition Challenge



- First CNN approach now famous as **AlexNet** [6]

Source: image-net.org, Russakovsky et al. 2015

ImageNet Large Scale Visual Recognition Challenge



- First CNN approach now famous as **AlexNet** [6]
- “Superhuman” should be Super-Karpathy-an performance

Source: image-net.org, Russakovsky et al. 2015

Deep Learning Users

NETFLIX
DAIMLER
Lunit
SIEMENS
Google
DeepMind
IBM
xerox
Microsoft
SAMSUNG

Playing Go

- 1997: Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor

Source: https://commons.wikimedia.org/wiki/File:FloorGoban.jpg

Playing Go

- 1997: Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor
- 2016: AlphaGo [11] beats a professional

Source: https://commons.wikimedia.org/wiki/File:FloorGoban.jpg

Playing Go

- 1997: Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor
- 2016: AlphaGo [11] beats a professional
- 2017: AlphaGoZero [12] **surpasses every human** in Go purely by **self-play**

Source: https://commons.wikimedia.org/wiki/File:FloorGoban.jpg

Playing Go

- 1997: Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor
- 2016: AlphaGo [11] beats a professional
- 2017: AlphaGoZero [12] **surpasses every human** in Go purely by **self-play**
- 2017: AlphaZero [13] **generalizes** to a number of other board games

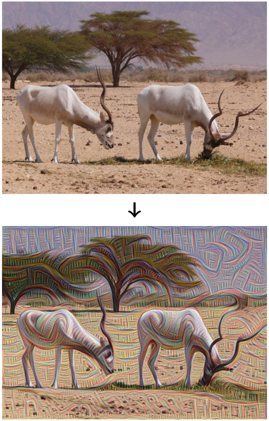
Source: https://commons.wikimedia.org/wiki/File:FloorGoban.jpg

Google DeepDream

Attempt to understand the inner workings of the network: What it "dreams" about when presented with images

Idea:

- Arbitrary image or noise as input
- Instead of adjusting network parameters, tweak image towards high activations
- Different layers enhance different features (low or high level)

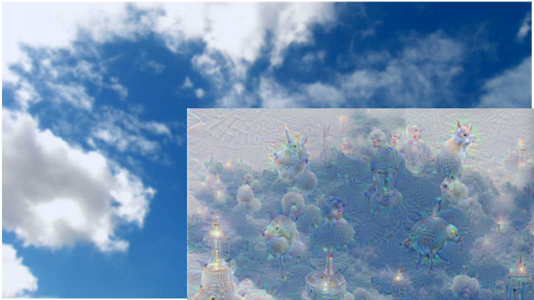


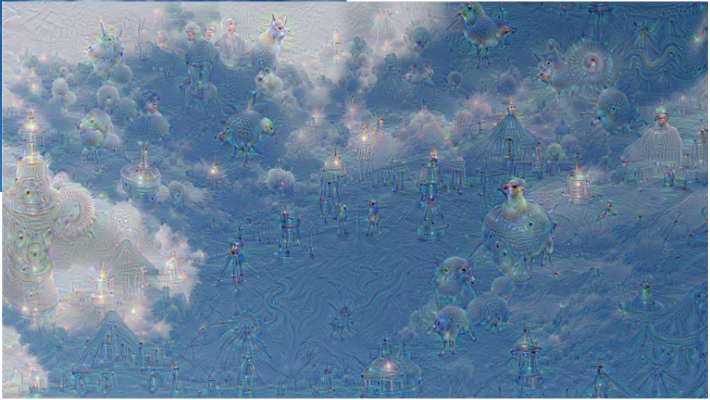
Source: https://research.googleblog.com

Google DeepDream

Source: https://research.googleblog.com

Google DeepDream





Source: https://research.googleblog.com

Google DeepDream

Looking for new animals in the clouds

"Admiral Dog!"

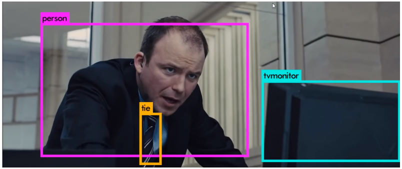
"The Pig-Snail"

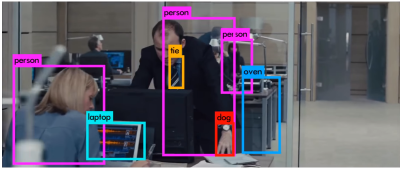
"The Camel-Bird"

"The Dog-Fish"

Source: https://research.googleblog.com

Real-Time Object Detection: YOLO & YOLO 9000 [7], [8]





Click for video

- YOLO: You only ~~live~~ look once
- Prior systems ➔ Use classifiers at multiple locations and scales
- YOLO ➔ Simultaneous regression of bounding box and label
- FAST: 40-90 frames/second on a NVIDIA Titan X

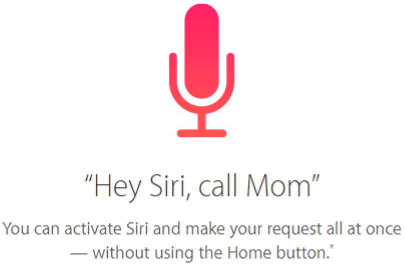
Source: www.youtube.com, Redmon and Farhadi 2016

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Every Day Use

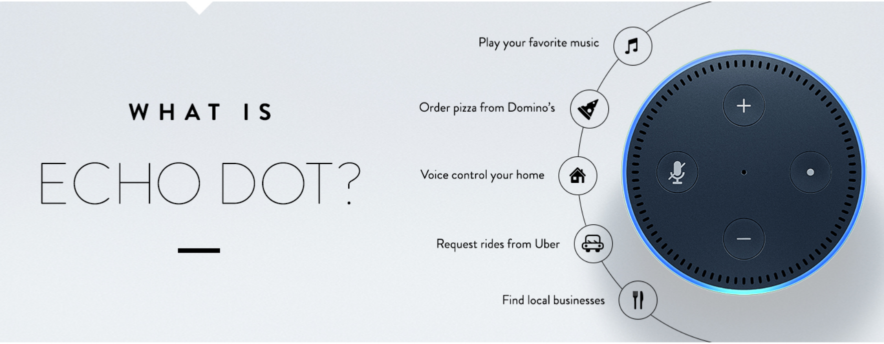
Siri

Siri: Speech Interpretation and Recognition Interface



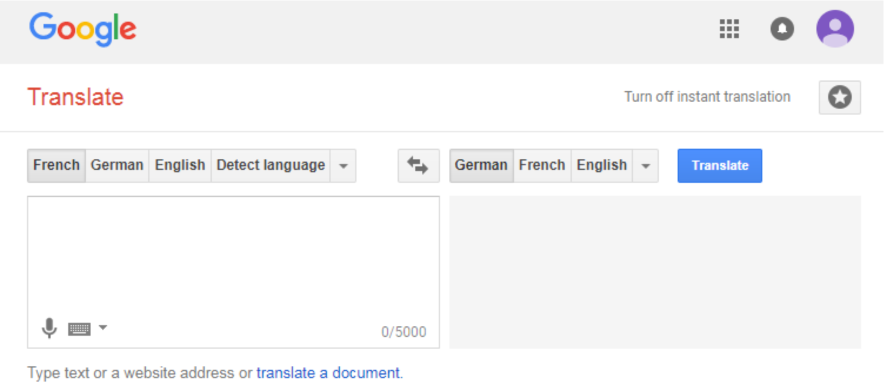
Source: www.apple.com/ios/siri/

Google Echo & Amazon Alexa Voice Service



Source: www.amazon.com

Google Translate



Source: translate.google.de

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

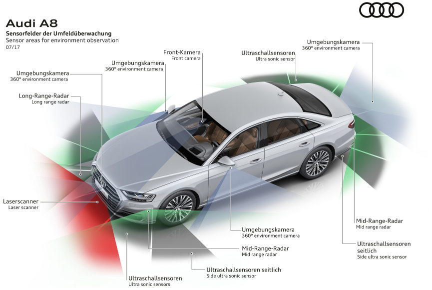
Research at the Pattern Recognition Lab

Assisted and Automated Driving

Goal

Find new ways to train and update deep learning mechanisms in environments with high safety requirements

- Assisted and automatic driving relies on sensor data
- Cameras to detect dynamic objects, driving lanes and free space
- Detection and segmentation tasks
  ➔ deep learning



Source: Audi AG

Assisted and Automated Driving

- Currently: neural networks trained and thoroughly tested before deployment

→ Requires huge amounts of manually labeled data

- Regular test drives cannot verify system reliability in all traffic scenarios

Click for video

Source: Mobileye N.V.

Assisted and Automated Driving

- Currently: neural networks trained and thoroughly tested before deployment

→ Requires huge amounts of manually labeled data

- Regular test drives cannot verify system reliability in all traffic scenarios
- **Challenge**: New ways to test algorithms in simulated environments and utilize data collected in production cars equipped with appropriate hardware

Click for video

Source: Mobileye N.V.

Smart Devices

Problem statement

Renewable energy power $\neq$ energy demand

- Underproduction ➔ backup power plants
- Overproduction ➔ energy lost

➔ Real-Time-Pricing to match energy demand and supply

- Needs *smart devices* to shift workload automatically

Smart Devices

Goal

Establish energy equilibrium by predicting energy consumption

- Example: Interrupt fridge cooling cycle when price is high, start washing machine when price is low
- Dependencies between tasks, user information and action necessary (e.g., washer/dryer)
- Task: Identify time-shiftable loads and assess appropriate time frame
- Approach: Train **recurrent neural networks** to identify usage patterns and dependencies between devices

Cloud Detection for Power Forecast [2]

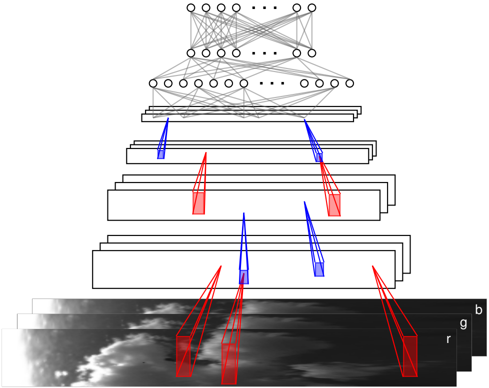
Goal

Power forecast for solar power plants with a high temporal and spatial resolution

Approach

1. Monitor the sky
2. Detect clouds
3. Estimate the cloud motion
4. Establish power forecasts

Cloud Detection for Power Forecast [2]



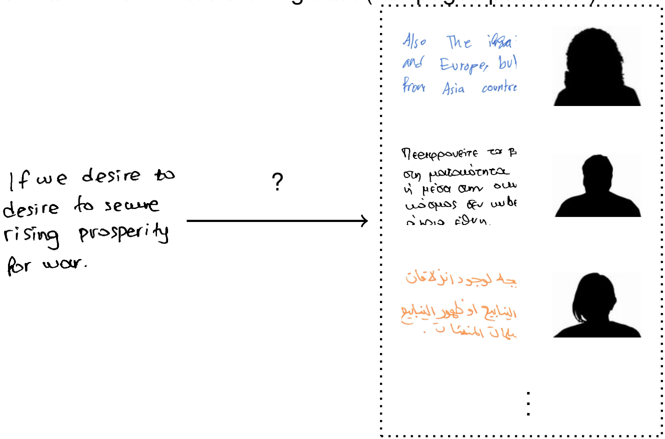
Input: Sky moving towards the sun

Output: Clear Sky Index = values betw. 0 (overcast sky) to 1 (clear sky)

Writer Recognition

Goal

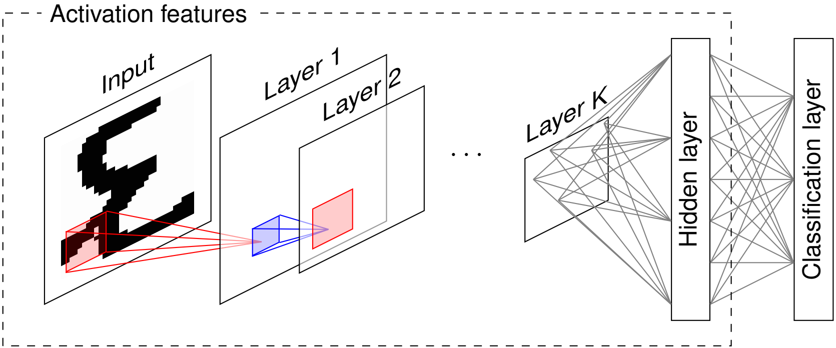
Writer identification with limited training data (few pages per writer)



Source: ICDAR'13 dataset, QUWI'15 dataset, freepik.com

Writer Recognition using CNN Activation Features [3]

Use Neuronal Network for feature extraction



Medical Applications

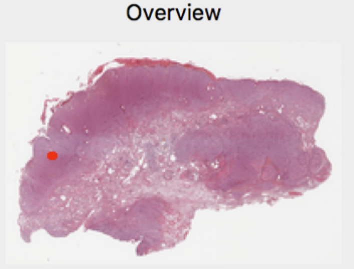
Cell Classification for Tumor Diagnostics [1]

Goal

Identify cells undergoing mitosis to asses tumor proliferation and aggressiveness in histological images

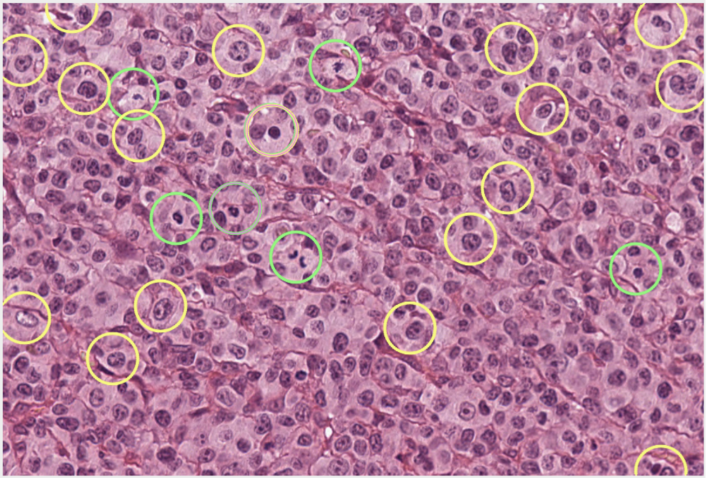
Challenge

- Histological images: large number of cells
- Full annotations not feasible
- Sparse annotations
- Cells vary significantly in size/shape/etc



Source: Aubreville et al. 2017

Cell Classification for Tumor Diagnostics [1]

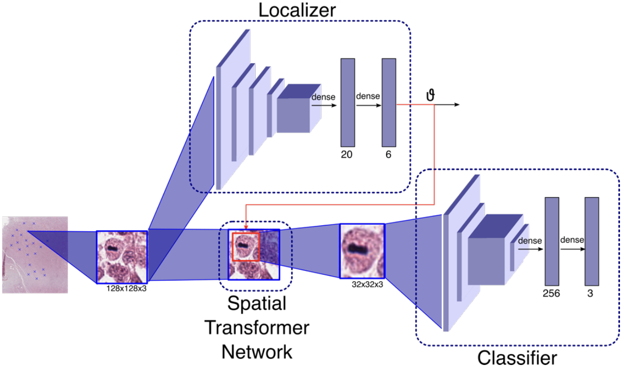


Source: Aubreville et al. 2017

Cell Classification for Tumor Diagnostics [1]

Approach

Use *spatial transformer networks* (STNs) to learn affine transformation **and** classification



Source: Aubreville et al. 2017

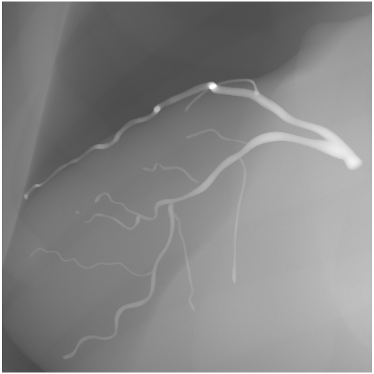
Defect Pixel Interpolation

Goal

- Reconstruction of coronaries based on truncated X-ray images
- Create “virtual” digital subtraction angiography

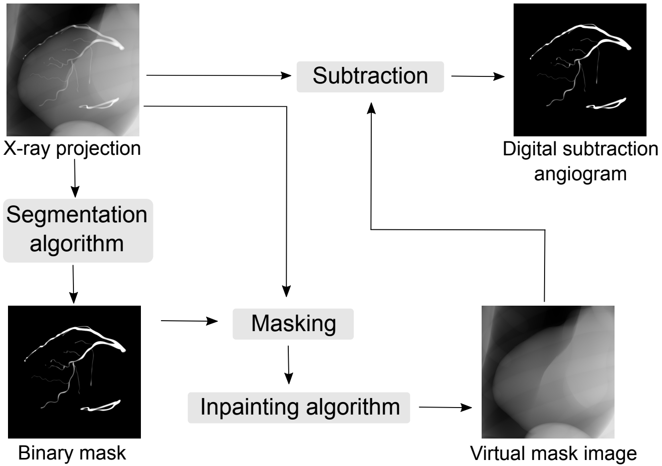
Approach

1. Segment coronary vessels
2. Mask fluoroscopic image
3. Inpaint using U-net
4. Subtract inpainted image to get untruncated data



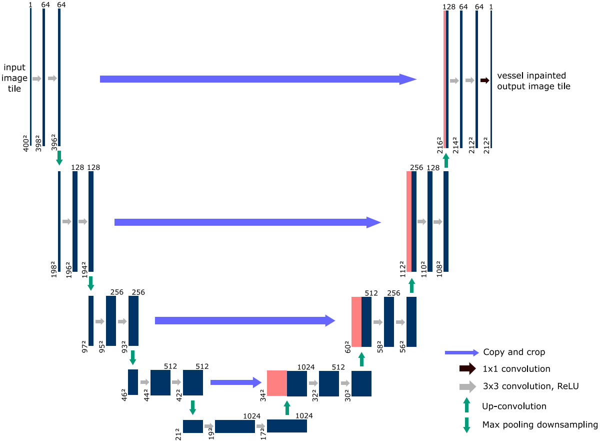
Defect Pixel Interpolation

Processing pipeline



Defect Pixel Interpolation

Deep learning for inpainting



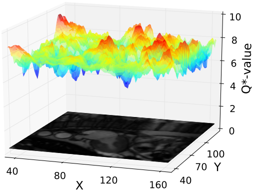
Organ Search [4]

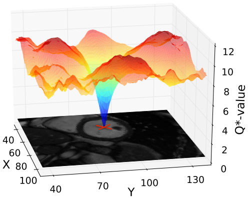
Goal

Locate anatomic structures automatically

Approach

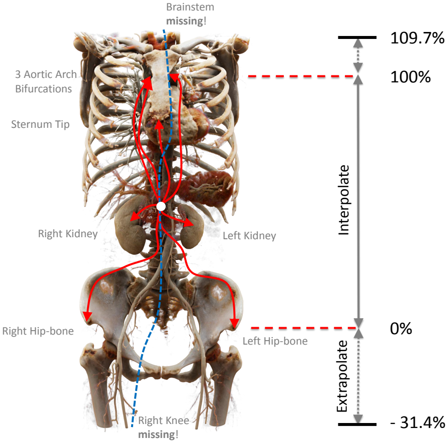
- Deep reinforcement learning
- Learn strategies how to search objects
  ➔ Learn optimal shortest search through image volume to different landmarks
- Hierarchical approach to improve speed and robustness





Source: Ghesu et al. 2016, Ghesu et al. 2017

Organ Search [4]



3D search trajectories
starting from image center

Body Range

Organ Search [4]

X-ray-transform Invariant Anatomical Landmark Detection

Goal

- Detect landmarks in X-ray images
- Knowing correspondences enables symbolic reconstruction

➔ Classic computervision reconstruction

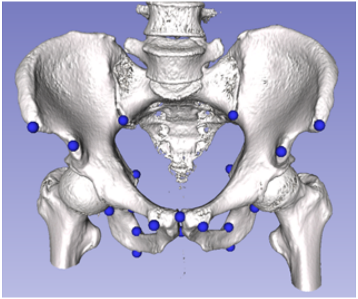
Challenge

- Transmission imaging

➔ Overlap/superposition of structures

➔ High variance due to projection

➔ Artifacts e.g. interventional devices



Source: Bier et al. 2018

X-ray-transform Invariant Anatomical Landmark Detection

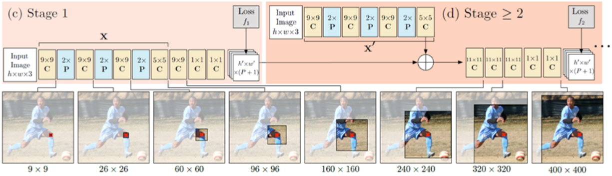
Approach: Convolutional Pose Machine (CPM) [17]

- Sequential prediction framework to detect landmarks

→ Yields 2D belief maps

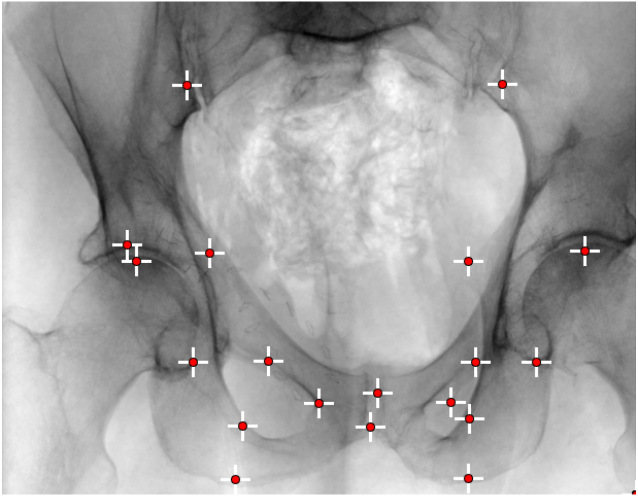
Properties

- Large receptive fields enable learning of configurations
- Estimation is refined over stages



Source: Wei et al. 2016

X-ray-transform Invariant Anatomical Landmark Detection

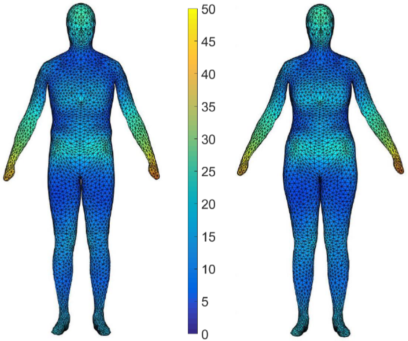


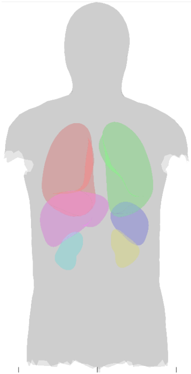
Source: Bier et al. 2018

Organ Prediction

Goal

Estimation of body and organ shapes based on patient's height and weight for X-ray exposure estimation.

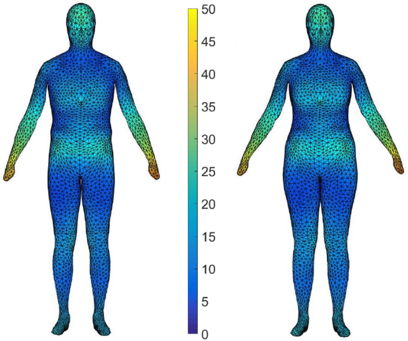


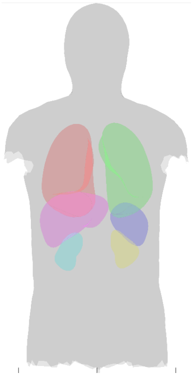


Organ Prediction

Goal

Estimation of body and organ shapes based on patient's height and weight for X-ray exposure estimation.





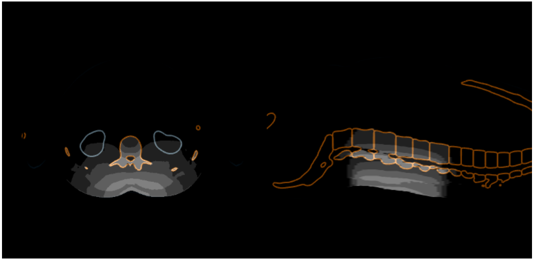
Could we achieve more if we had old CT data of a patient?

Action Learning for 3D Point Cloud Based Organ Segmentation

Goal: Versatile organ segmentation for:

- Use it in computer aided diagnosis
- Treatment planning
- Dose management

Dose estimation in interventions with overlays

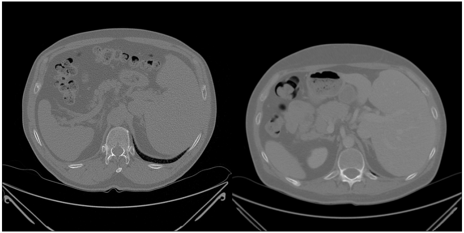


Action Learning for 3D Point Cloud Based Organ Segmentation

Challenges for clinical applications

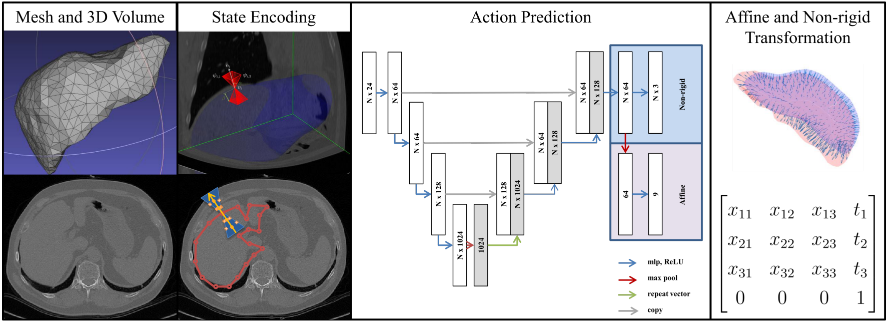
- Robustness w.r.t.
  1. Individual anatomy
  2. Scan protocols
- Time constraints

Pre-operative CT (left) and contrast enhanced CT (right)



Action Learning for 3D Point Cloud Based Organ Segmentation

- Reinforcement learning
- Predict the transformation at given state



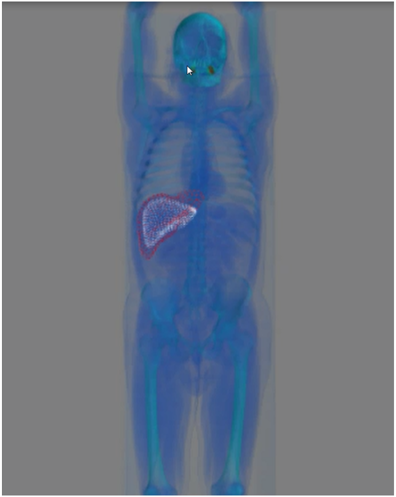
$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & t_1 \\ x_{21} & x_{22} & x_{23} & t_2 \\ x_{31} & x_{32} & x_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Action prediction pipeline for 3D point cloud based organ segmentation

Source: Zhong et al. 2018

Action Learning for 3D Point Cloud Based Organ Segmentation

- Runtime:
  1. **0.3 - 2.6s per volume**
  2. **50 - 100 speedup** from U-net [16]
- Very accurate
- Robust to:
  1. scan protocol
  2. contrast agent
  3. organ initialization



Source: Zhong et al. 2018

Limitations

Image Captioning

Image captioning (e.g., Karpathy et al. 2014 [5]) often yields impressive results:

"baseball player is throwing ball in game."

"girl in pink dress is jumping in air."

"man in black shirt is playing guitar."

Source: http://cs.stanford.edu/people/karpathy/deepimagesent

Image Captioning

“Straightforward” errors:

"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"black cat is sitting on top of suitcase."

Source: http://cs.stanford.edu/people/karpathy/deepimagesent

Image Captioning

Plainly wrong:

"a horse is standing in the middle of a road."

"a woman holding a teddy bear in front of a mirror."

Source: http://cs.stanford.edu/people/karpathy/deepimagesent

Challenges with Training Data

- Deep learning applications often rely on **huge**, manually-annotated data sets
- Hard to obtain, time-consuming, expensive, ambiguous
- To err is human: Mislabeled ground-truth annotation

➔ May cause a significant drop in performance

Challenges with Training Data

- Deep learning applications often rely on **huge**, manually-annotated data sets
- Hard to obtain, time-consuming, expensive, ambiguous
- To err is human: Mislabeled ground-truth annotation

→ May cause a significant drop in performance

- Question: How far can we get with simulations?

Challenges with Trust and Reliability

- Verification is mandatory for high risk applications

➔ End-to-end learning prohibits verification of parts

➔ Largely unsolved

Challenges with Trust and Reliability

- Verification is mandatory for high risk applications

➔ End-to-end learning prohibits verification of parts

➔ Largely unsolved

- Possible solution: Reformulate classical algorithms

Future Directions

Learning of Algorithms

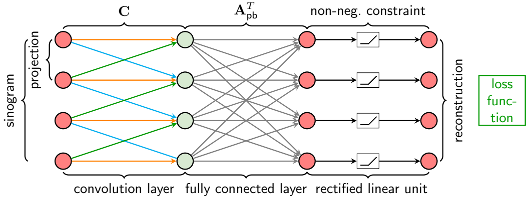
- Computed Tomography
- Efficient solution via filtered back-projection:

$$f(x,y) = \int_0^{\pi} p(s,\theta) * h(s)\big|_{s=x\cos\theta + y\sin\theta} d\theta$$

- Three steps:
  - Convolution along s
  - Back-projection along θ
  - Suppress negative values

Reconstruction Networks

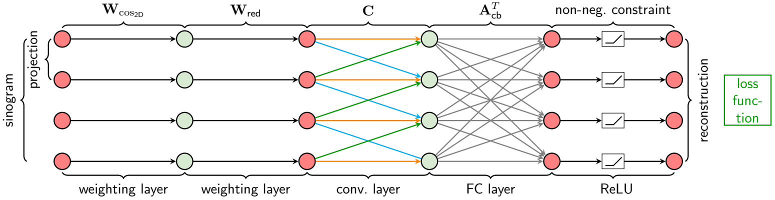
- All three steps can be modeled as a neural network:



- All weights are known from FBP

Reconstruction Networks

- Reconstruction Networks can be expanded

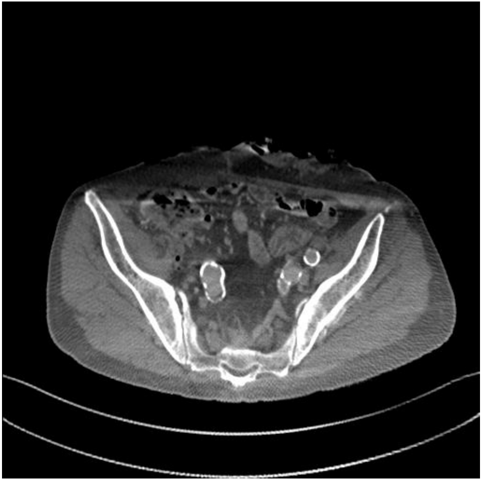


- Embedding of "heuristics" for artifact reduction possible

Application to Incomplete Scans [14]

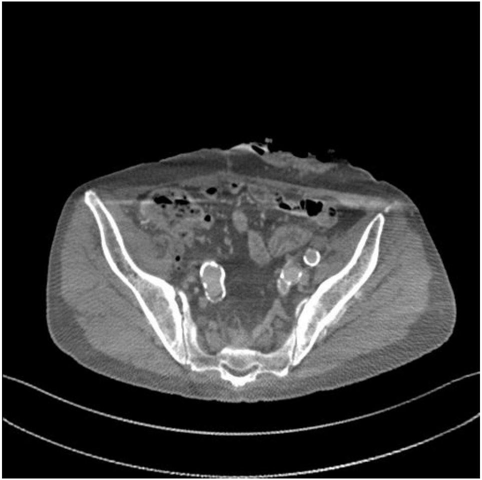
Reconstruction with 360°

Application to Incomplete Scans [14]



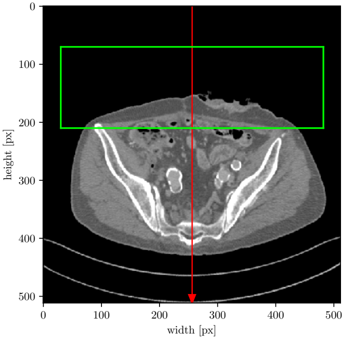
Reconstruction with 180° (FBP)

Application to Incomplete Scans [14]

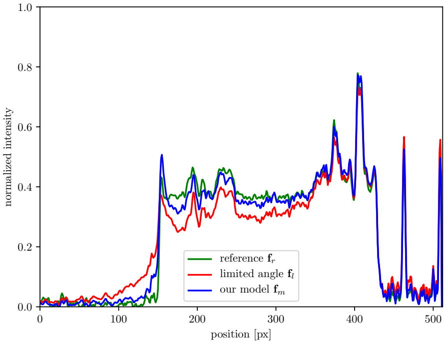


Reconstruction with 180° (NN)

Application to Incomplete Scans [14]

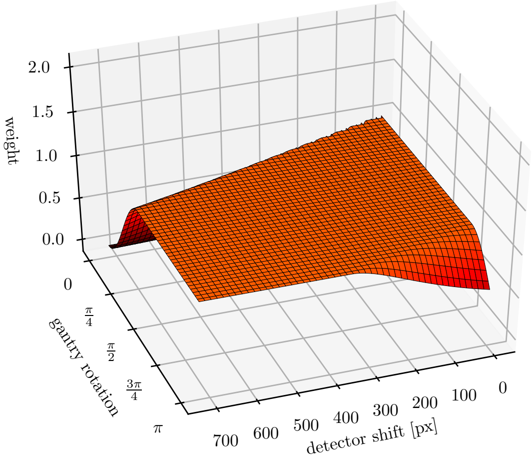


Location of the lineplot



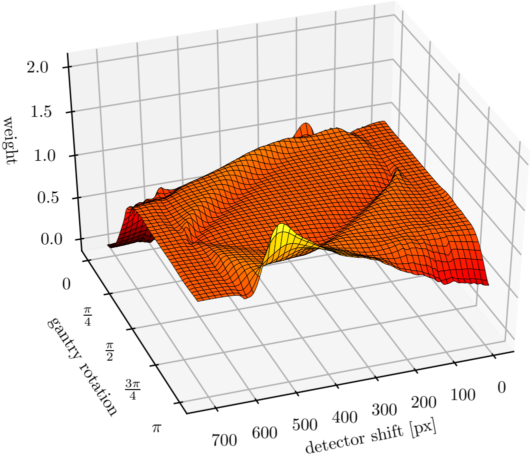
Lineplot

Parker Weights



Parker weights before learning

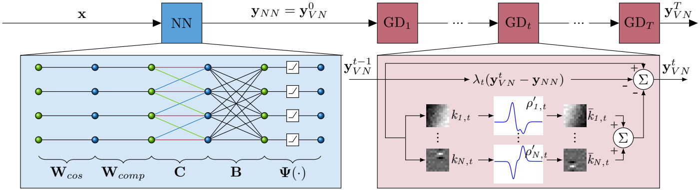
Parker Weights



Parker weights after learning

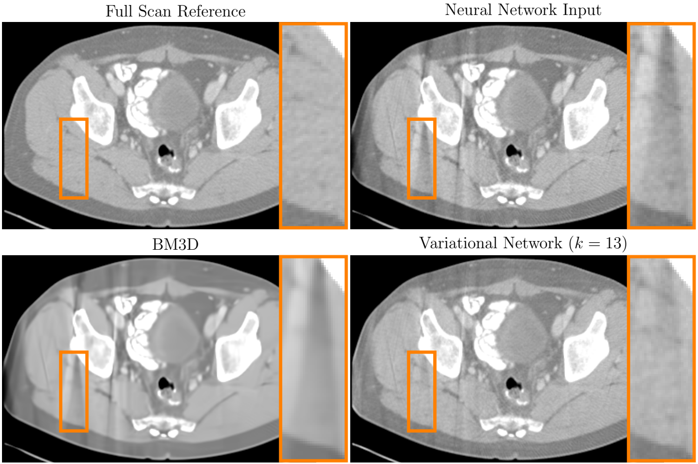
Further Extensions

- Add non-linear de-streaking and de-noising step:



Step 1: Neural network CT reconstruction **Step 2: Variational network non-linear filtering**

Further Extensions



Machine Learning and Pattern Recognition

Terminology and Notation

Throughout these slides, we will use the following notation:

- Matrices: bold, uppercase, e.g., $\mathbf{M}$, $\mathbf{A}$
- Vectors: bold, lowercase, e.g., $\mathbf{v}$, $\mathbf{x}$
- Scalars: italic, lowercase, e.g., y, w, α
- Gradient of a function: ∇, partial derivative: ∂

Terminology and Notation

Throughout these slides, we will use the following notation:

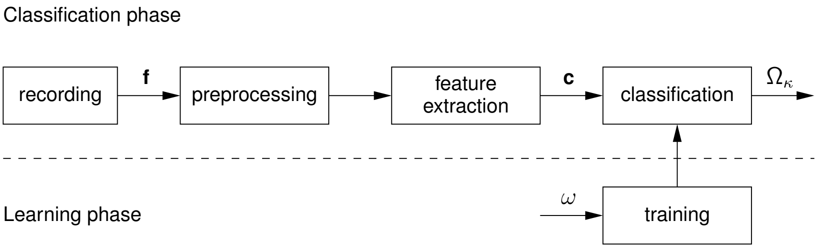
- Matrices: bold, uppercase, e.g., $\mathbf{M}$, $\mathbf{A}$
- Vectors: bold, lowercase, e.g., $\mathbf{v}$, $\mathbf{x}$
- Scalars: italic, lowercase, e.g., y, w, α
- Gradient of a function: ∇, partial derivative: ∂

Notation regarding deep learning:

- Trainable parameters ("weights"): w
- Features/input: $\mathbf{x}$
- Ground truth label/target: y
- Estimated output: $\hat{y}$
- Index denoting iteration will be in superscript, e.g., $\mathbf{x}^{(i)}$

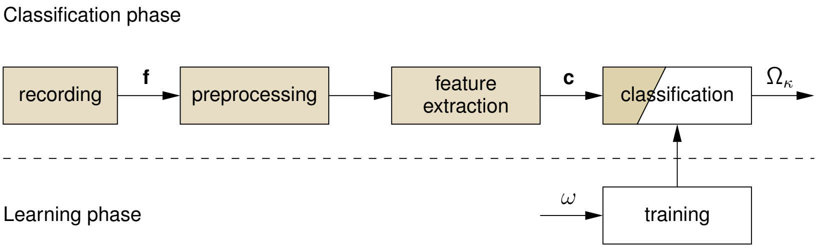
The notation and the terminology will be further developed throughout the lecture.

“Classical” Image Processing Pipeline



“Classical” Image Processing Pipeline

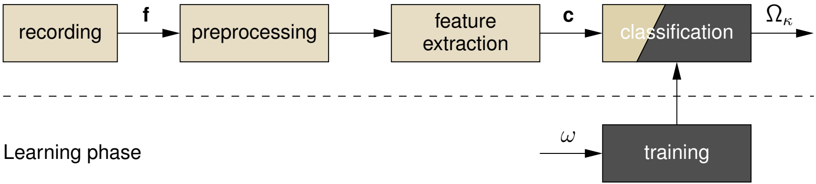
Lecture Introduction to Pattern Recognition



“Classical” Image Processing Pipeline

Lecture Introduction to Pattern Recognition

Classification phase



Learning phase

Lecture Pattern Recognition

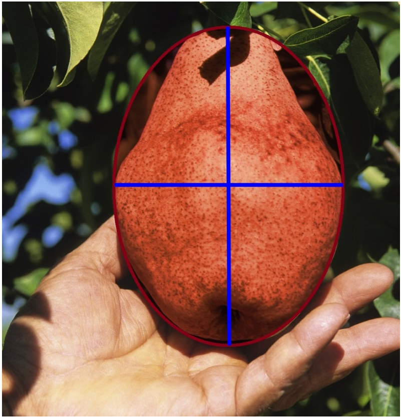
“Classical” Image Processing Pipeline: Apple vs. Pears

Source: https://commons.wikimedia.org

“Classical” Image Processing Pipeline: Apple vs. Pears

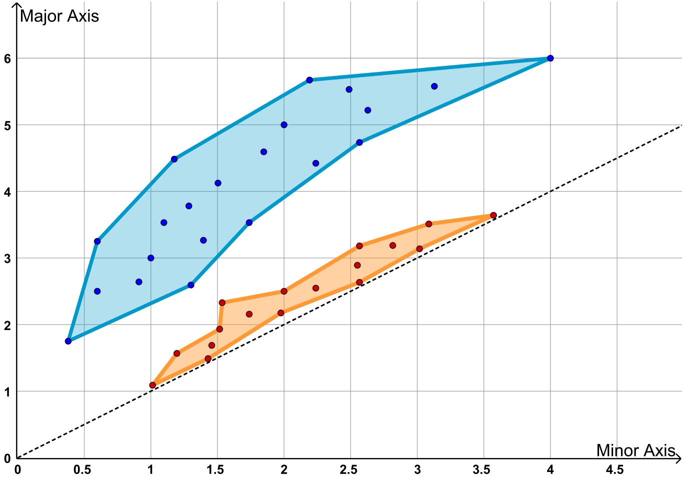
Source: https://commons.wikimedia.org

“Classical” Image Processing Pipeline: Apple vs. Pears

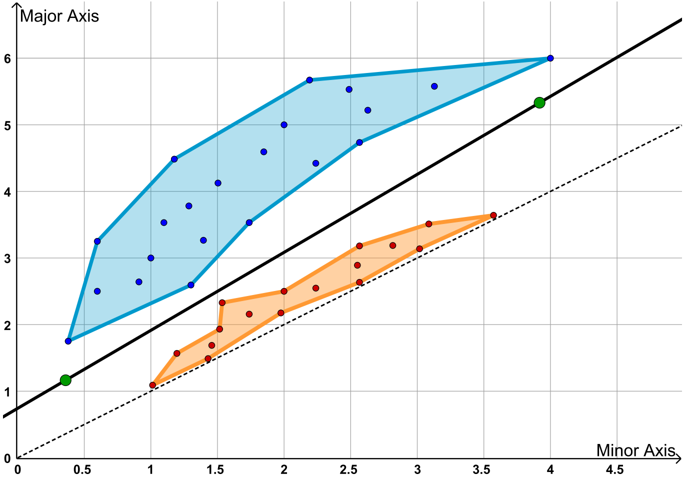


Source: https://commons.wikimedia.org

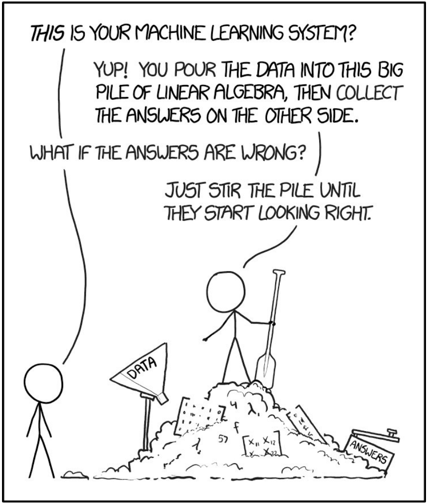
“Classical” Image Processing Pipeline: Apple vs. Pears



“Classical” Image Processing Pipeline: Apple vs. Pears



Pipeline in Deep Learning



Source: https://xkcd.com/1838/

Postulates for Pattern Recognition

6 Postulates:

1. Availability of a **representative sample** ω of **patterns** ${}^{i}\mathbf{f}(\mathbf{x})$ for the given field of problems Ω

$$\omega = \{{}^{1}\mathbf{f}(\mathbf{x}), \ldots, {}^{N}\mathbf{f}(\mathbf{x})\} \subseteq \Omega$$

Postulates for Pattern Recognition

6 Postulates:

1. Availability of a **representative sample** ω of **patterns** ${}^{i}\mathbf{f}(\mathbf{x})$ for the given field of problems Ω

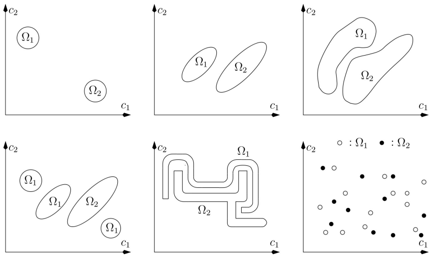
$$\omega = \{{}^{1}\mathbf{f}(\mathbf{x}), \ldots, {}^{N}\mathbf{f}(\mathbf{x})\} \subseteq \Omega$$

2. A (simple) pattern has **features**, which characterize its membership in a certain class Ω_κ.

Postulates for Pattern Recognition (cont.)

3. Compact domain of features of the same class; domains of different classes are (reasonably) separable.
   - small **intra-class distance**
   - high **inter-class distance**

Example of an increasingly less compact domain in the feature space:



Postulates for Pattern Recognition (cont.)

4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.

Postulates for Pattern Recognition (cont.)

4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.

5. A (complex) pattern $\mathbf{f}(\mathbf{x}) \in \Omega$ has a certain **structure**. Not any arrangement of simple constituents is a valid pattern. Many patterns may be represented with relatively few constituents

Postulates for Pattern Recognition (cont.)

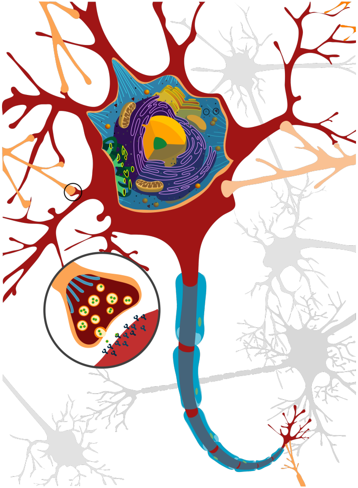
4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.

5. A (complex) pattern $\mathbf{f}(\mathbf{x}) \in \Omega$ has a certain **structure**. Not any arrangement of simple constituents is a valid pattern. Many patterns may be represented with relatively few constituents
6. Two patterns are **similar** if their features or simpler constituents differ only slightly

Perceptron

Perceptron Biology - Neural Excitation (simplified)

- Neurons are **connected** by synapses / dendrites
- If the **sum** of incoming (excitatory and inhibitory) **activations** is large enough, an action potential is created
- The action potential activates synapses to other neurons, “transmitting” information
- All-or-none response: A **higher** stimulus does **not** cause a **higher** response
  ➔ “binary classifier”



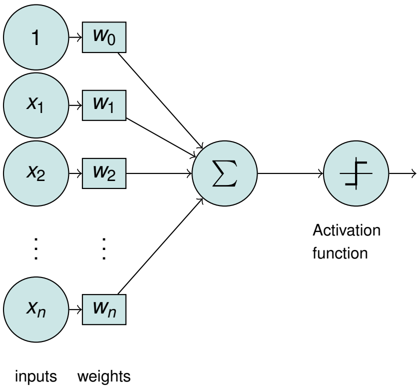
Source: https://commons.wikimedia.org

Rosenblatt's Perceptron

- In 1957, Frank Rosenblatt [9] invented the Perceptron
- Binary classification $y \in \{-1, 1\}$.
- It computes the function

$$\hat{y} = \text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x}),$$

where
$\mathbf{w} = (w_0, \ldots, w_n)$: set of weights ($w_0$=bias)
$\mathbf{x} = (1, x_1, \ldots, x_n)$: input feature vector



Perceptron Objective Function

Task: Find weights that minimize the distance of misclassified samples to the decision boundary

Assumptions

- Let $S = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_m}, y_m)\}$ be a training data set
- Let $\mathcal{M}$ be the set of misclassified feature vectors $y_i \neq \hat{y}_i = \text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x_i})$ according to a given set of weights $\mathbf{w}$
- Optimization problem:

$$\underset{\mathbf{w}}{\text{argmin}} \left\{ D(\mathbf{w}) = -\sum_{\mathbf{x}_i \in \mathcal{M}} y_i \cdot (\mathbf{w}^\mathsf{T}\mathbf{x}_i) \right\}$$

Perceptron Objective Function – Observations

- Objective function depends on misclassified feature vectors $\mathcal{M}$ ➔ iterative optimization
- In each iteration, the cardinality and composition of $\mathcal{M}$ may change
- The gradient of the objective function is:

$$\nabla D(\mathbf{w}) = -\sum_{x_i \in \mathcal{M}} y_i \cdot \mathbf{x}_i$$

Perceptron Training

- Strategy 1: Process all samples, then perform weight update
- Strategy 2: Take an update step right after each misclassified sample
- Update rule in iteration $(k+1)$ for the misclassified sample $\mathbf{x}_i$ simplifies to:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y_i \cdot \mathbf{x}_i$$

- Optimization until convergence or for a predefined number of iterations

Organizational Matters

Dates

- Lecture: Tuesday, 14:15 - 15:45.
- Exercises:
  - Monday, 12:00-14:00 **(Start: October 22)**
  - Tuesday, 8:00 - 10:00 **(Start: October 23)**
  - Wednesday, 10:00 - 12:00 **(Start: October 24)**
  - Thursday, 14:00-16:00 **(Start: October 18)**
  - Friday, 8:00 - 10:00 **(Start: October 19)**
- First exercise: Python recap (no submission)
- Please sign-up in StudOn by October 16 (today), 17:00, for the course and by October 17, 09:00, for the exercises.
- If you decide to drop the course, please **inform us as soon as possible!**

Grading

- Module consists of lecture **and** exercises (together 5 ECTS)
- 30 min. oral exams in the semester break, determines grade
- Exercises are **mandatory**, each exercise has to be completed successfully to pass the module

Exercise Content

- Python introduction
- Developing a neural network framework from scratch
  - Feed Forward Neural Networks
  - Convolutional Neural Networks
  - Regularization
  - Recurrent Networks
- Using the Tensorflow framework
  - Large scale classification

Exercise Requirements

- Basic knowledge of Python and Numpy
- Linear algebra, -
- Image processing, -
- Pattern recognition fundamentals
- Passion for coding
- Attention to detail
- Time

How it works

- New exercise every two weeks
- One session explanation
- One session assistance
- Submission opportunity and new exercise
- First four exercises will have unit tests
- Demonstration of every exercise mandatory

Summary

- Deep learning more and more present in day to day life
- Huge support from industry
- **Very** active area of research!
- Perceptron as binary classifier motivated by biological neurons

NEXT TIME

ON DEEP LEARNING

Next Time

- Extending the Perceptron to obtain a universal function approximator
- Gradient based training algorithm for these models
- Efficient automatic computation of gradients

Comprehensive Questions

- What are the six postulates of pattern recognition?
- What is the Perceptron objective function?
- Can you name three applications successfully tackled by deep learning?

Further Reading

- Link - Deep learning book
- Link - Research and publications at the Pattern Recognition Lab
- Link - Google Research Blog with posts on e.g. Deep dream or Alpha Go

Questions?

References

References I

[1] M. Aubreville, M. Krappmann, C. Bertram, et al. "A Guided Spatial Transformer Network for Histology Cell Differentiation". In: ArXiv e-prints (July 2017). arXiv: 1707.08525 [cs.CV].

[2] David Bernecker, Christian Riess, Elli Angelopoulou, et al. "Continuous short-term irradiance forecasts using sky images". In: Solar Energy 110 (2014), pp. 303–315.

[3] Vincent Christlein, David Bernecker, Florian Hönig, et al. "Writer Identification Using GMM Supervectors and Exemplar-SVMs". In: Pattern Recognition 63 (2017), pp. 258–267.

[4] Florin Cristian Ghesu, Bogdan Georgescu, Tommaso Mansi, et al. "An Artificial Agent for Anatomical Landmark Detection in Medical Images". In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016 Athens, 2016, pp. 229–237.

References II

[5] A. Karpathy and L. Fei-Fei. “Deep Visual-Semantic Alignments for Generating Image Descriptions”. In: ArXiv e-prints (Dec. 2014). arXiv: 1412.2306 [cs.CV].

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[7] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: CoRR abs/1506.02640 (2015).

[8] J. Redmon and A. Farhadi. “YOLO9000: Better, Faster, Stronger”. In: ArXiv e-prints (Dec. 2016). arXiv: 1612.08242 [cs.CV].

References III

[9] Frank Rosenblatt. The Perceptron–a perceiving and recognizing automaton. 85-460-1. Cornell Aeronautical Laboratory, 1957.

[10] Olga Russakovsky, Jia Deng, Hao Su, et al. "ImageNet Large Scale Visual Recognition Challenge". In: International Journal of Computer Vision 115.3 (2015), pp. 211–252.

[11] David Silver, Aja Huang, Chris J. Maddison, et al. "Mastering the game of Go with deep neural networks and tree search". In: Nature 529.7587 (Jan. 2016), pp. 484–489.

[12] David Silver, Julian Schrittwieser, Karen Simonyan, et al. "Mastering the game of go without human knowledge". In: Nature 550.7676 (2017), p. 354.

[13] David Silver, Thomas Hubert, Julian Schrittwieser, et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: arXiv preprint arXiv:1712.01815 (2017).

References IV

[14] Tobias Würfl, Florin C Ghesu, Vincent Christlein, et al. "Deep learning computed tomography". In: International Conference on Medical Image Computing and Computer-Assisted Springer International Publishing. 2016, pp. 432–440.

[15] Jia Deng, Wei Dong, Richard Socher, et al. "Imagenet: A large-scale hierarchical image database". In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conferenc IEEE. 2009, pp. 248–255.

[16] Patrick Ferdinand Christ, Mohamed Ezzeldin A Elshaer, Florian Ettlinger, et al. "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields". In: International Conference on Medical Image Computing and Computer-Assisted Springer. 2016, pp. 415–423.

References V

[17] S. E. Wei, V. Ramakrishna, T. Kanade, et al. “Convolutional Pose Machines”. In: CVPR. 2016, pp. 4724–4732.