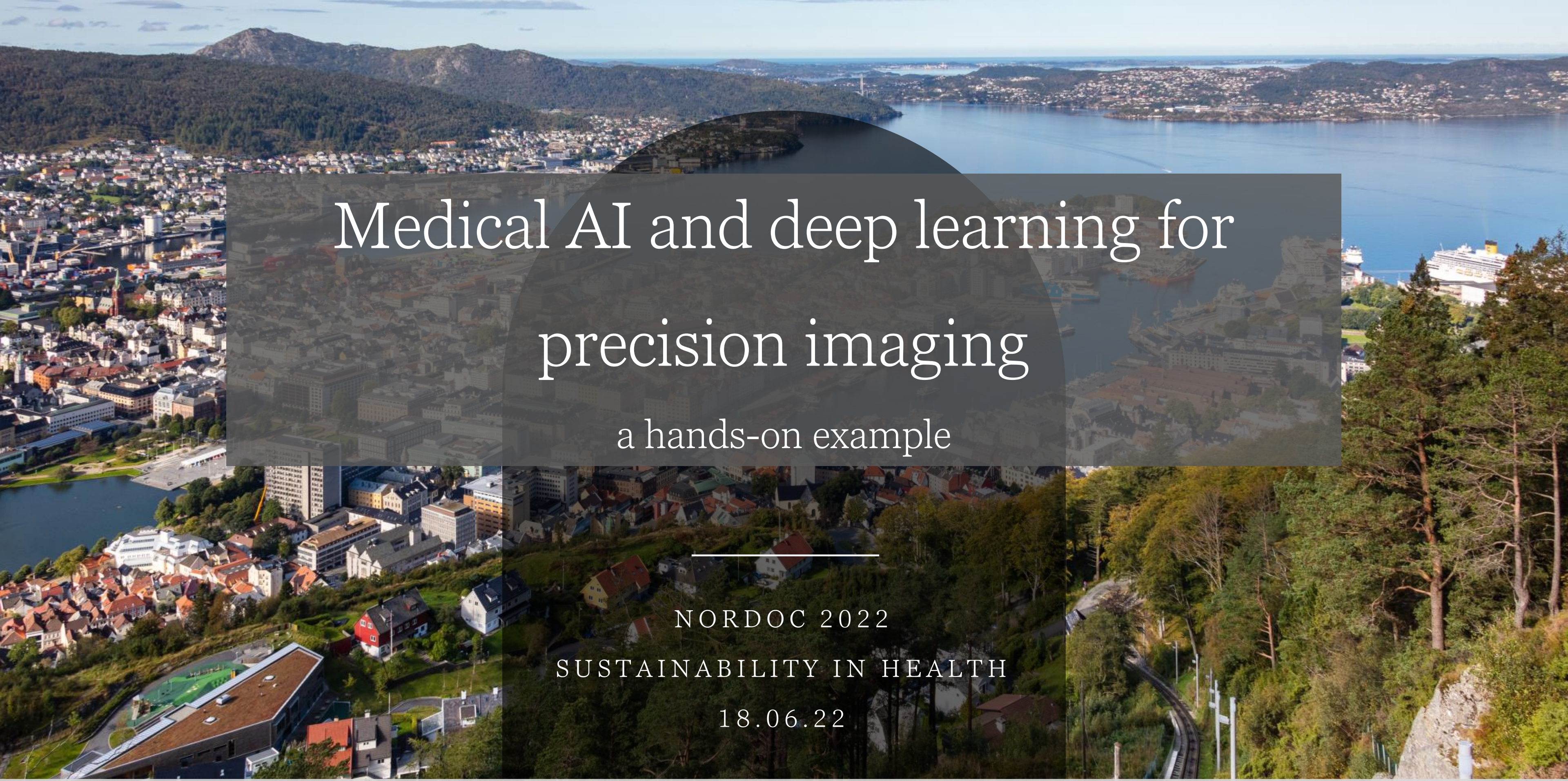


Medical AI and deep learning for precision imaging

a hands-on example

NORDOC 2022

SUSTAINABILITY IN HEALTH

18.06.22

Western Norway University of Applied Sciences

MMIV

Haukeland University Hospital

Alexander Selvikvåg Lundervold

allu@hvl.no; lundervold.net

Medical AI and deep learning for precision imaging

a hands-on example

NORDOC 2022

SUSTAINABILITY IN HEALTH

18.06.22

Western Norway University of Applied Sciences

MMIV

Haukeland University Hospital

Alexander Selvikvåg Lundervold

allu@hvl.no; lundervold.net

Plan

Hands-on!

What *is* deep learning?

How do you *do* deep learning?

Some perspectives, opportunities and challenges

Plan

Hands-on!

Will only have time for a quick peek at the field!

Objectives:

(i) to *demystify*

(ii) make sure everyone understands what deep learning is about at a fundamental level, and how you *do* deep learning as a practicioner

(iii) hopefully trigger some ideas and potentially a desire to explore further

...t *is* deep learning?

How do you *do* deep learning?

Some perspectives, opportunities and challenges

Plan

Hands-on!

What *is* deep learning?

How do you *do* deep learning?

Some perspectives, opportunities and challenges

Deep learning

searching for good hierarchical geometric representations

Deep learning

searching for good hierarchical geometric representations

Deep learning

searching for good hierarchical geometric representations

Deep learning

searching for good hierarchical geometric representations

Deep learning

searching for good hierarchical geometric representations

01

02

03

04

05

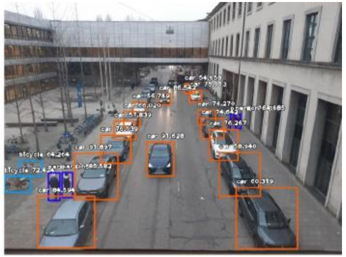
Function approximation

Function approximation

an image

$$y \approx f(x; \theta)$$

what's in the image

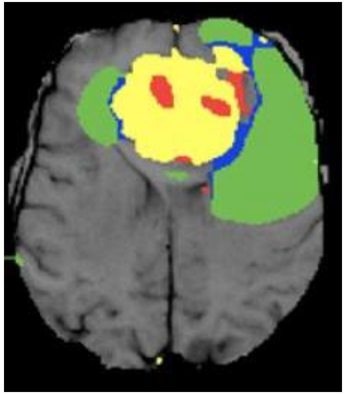


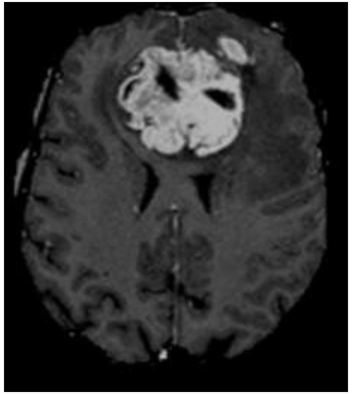
Function approximation

an image

$$y \approx f(x;\theta)$$

what’s in the image



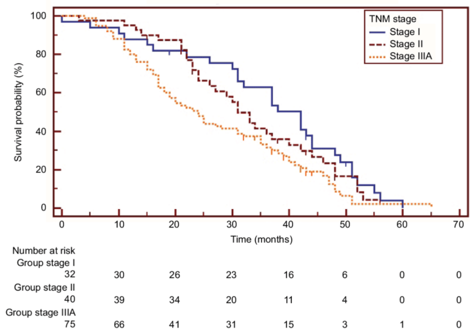


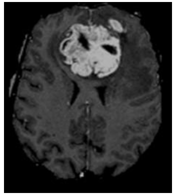
Function approximation

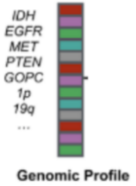
an image

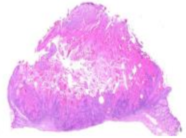
$$y \approx f(x;\theta)$$

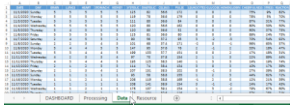
what's in the image











Function approximation

heterogeneous information

$$y \approx f(x;\theta)$$

outcome (e.g., survival)

Training

$$(X, y)$$

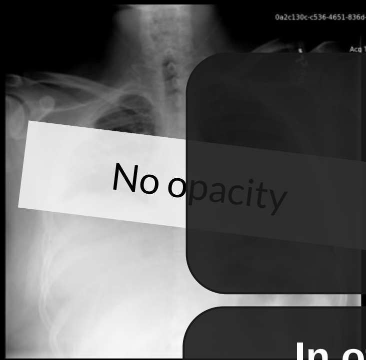
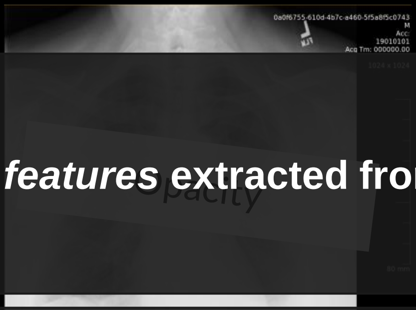
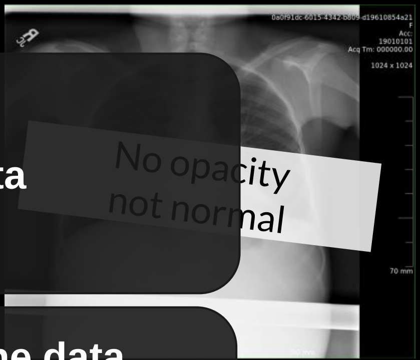
Training

$$(X, y)$$

Input data

Labels

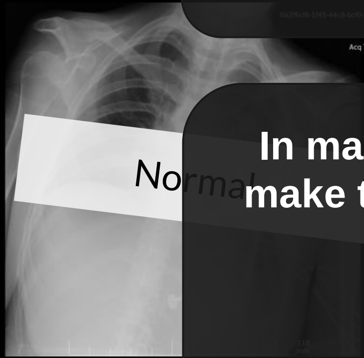
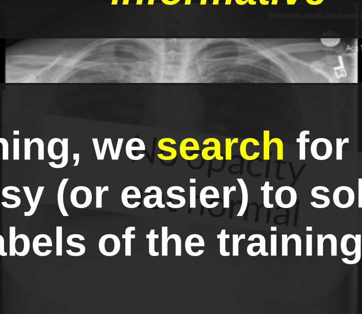
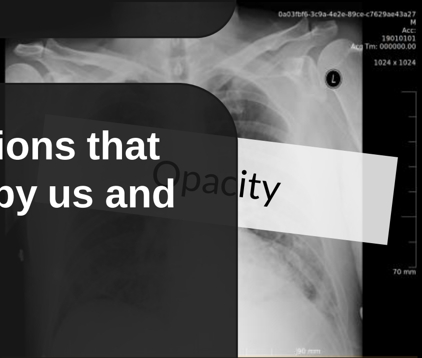
This is the setup in what's called ***supervised learning***

Train on *features* extracted from the data

In other words, *representations* or *views* of the data

Important that these representations are sufficiently informative

In machine learning, we search for representations that make the task easy (or easier) to solve, guided by us and the labels of the training data set.

$$y \approx f(x\,;\,\theta)$$

01 Machine learning: **function approximation** based on **training**

features of a disease,
process, patient or
system

$$y \approx f(x\,;\,\theta)$$

healthy or not healthy /
outcome

Models and parameters

$$y \approx f(x;\theta)$$

model **parameters**

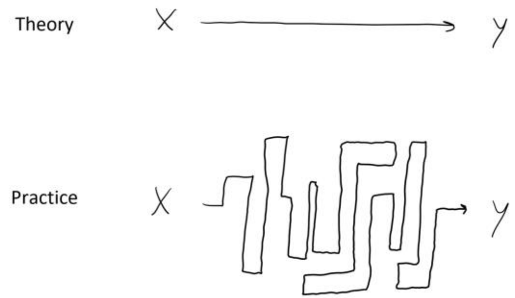
Training **is a search for useful representations through a space defined by a family Ƒ of parametrized models**

01 Machine learning: **function approximation** based on **training**

02 We pick the model family Ƒ. The parameters Θ are found automatically through training

$$y \approx f(x; \theta)$$

Theory versus practice



In theory:

can get arbitrarily good approximations by training simple models
(*universal approximation* and *no free lunch theorems*)

In practice:

which **f** we use and how the training is done makes a huge difference

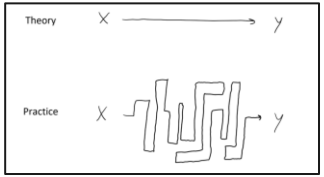
The family Ƒ of models considered, i.e., the *hypothesis space*, should be (i) efficiently searchable, (ii) leading to expressive models that can (iii) generalize beyond the training data distribution (i.e., work well on new data)

01 Machine learning: **function approximation** based on **training**

02 We pick the model family Ƒ. The parameters Θ are found automatically through training

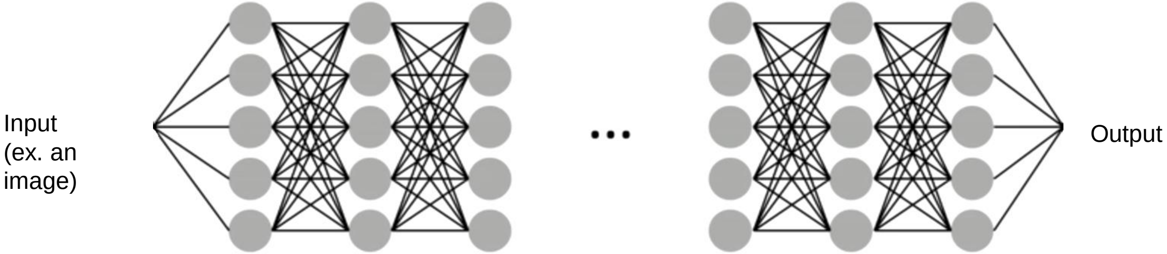
03 In practice: some models are better suited to some tasks than others.

$$y \approx f(x; \theta)$$



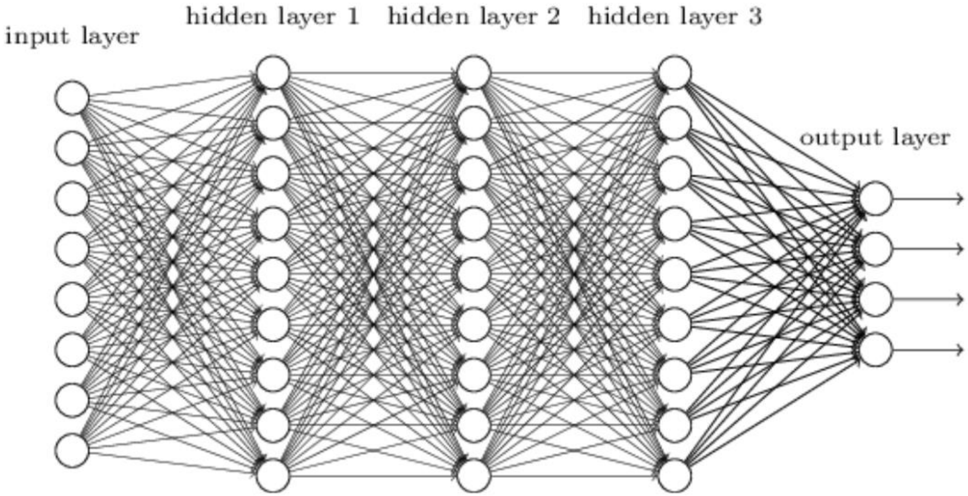
Deep learning

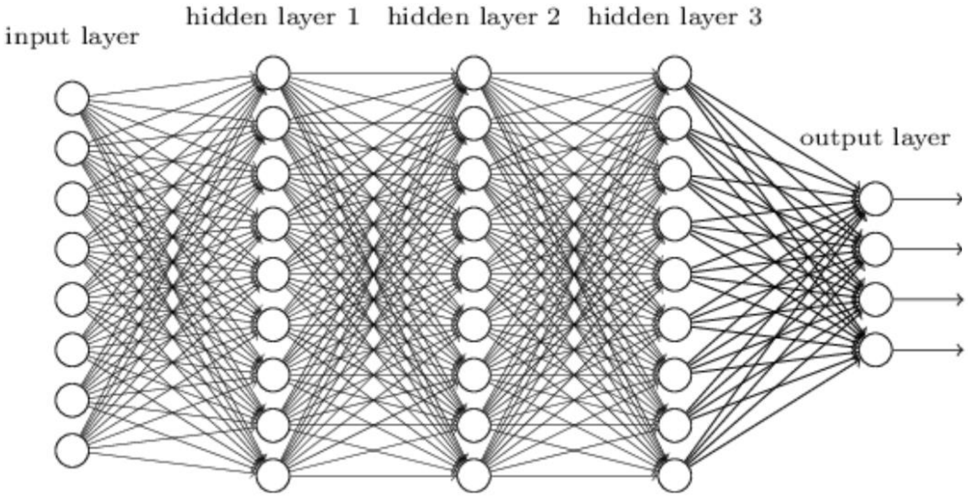
$$y \approx f(x;\theta)$$



Artificial neural networks

Computational graphs of simple units (“neurons”) connected in various specific ways, parametrized by parameters associated to each layer.


hidden layer 1
hidden layer 2
hidden layer 3
input layer
output layer



A geometric analogy: uncrumpling paper

This is a fancy way to draw what's really just a composition of simple functions:

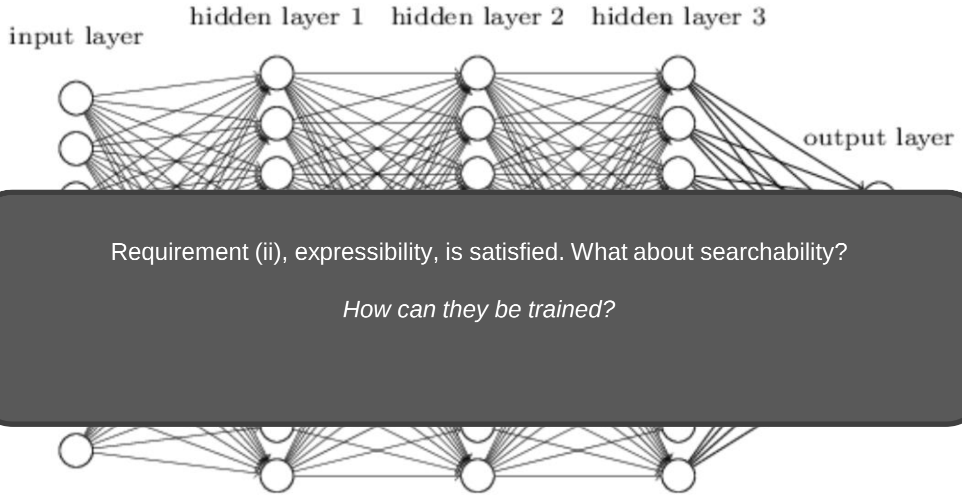
$$y = \sigma_{k+1}(W_{k+1}^T \sigma_k (W_k^T \left(\cdots \left(\sigma_1 (W_1^T x) \right) \cdots \right) = f(x; \theta)$$

Input x is sent through several consecutive **layers** (functions). An output is produced.

Think of each layer as a (geometric) **transformation** of its input.

Every layer has a bunch of **parameters**, stored in the matrices W

Each family member of Ƒ is complex transformation broken down into a composition of simpler ones, parametrized by the layer weights.



Requirement (ii), expressibility, is satisfied. What about searchability?

How can they be trained?

This is a fancy way to draw what's really just a composition of simple functions:

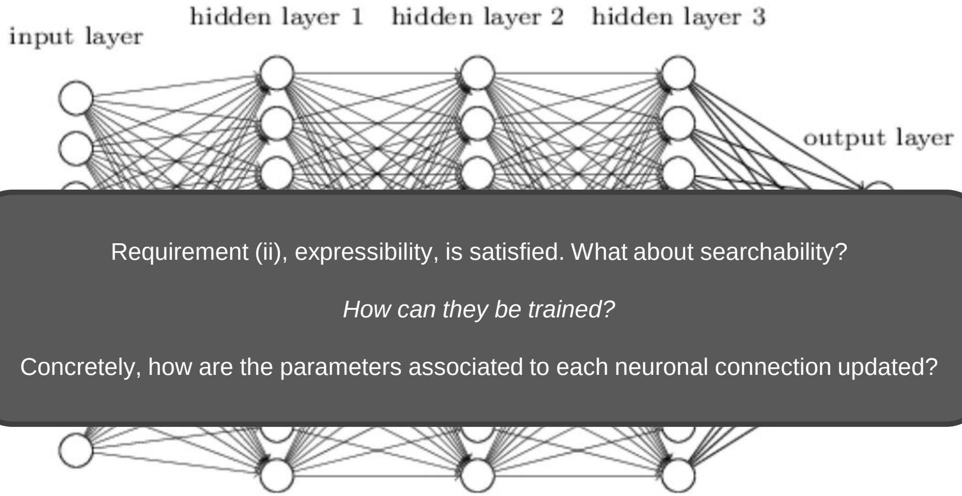
$$y = \sigma_{k+1}(W_{k+1}^T \sigma_k (W_k^T \left(\cdots \left(\sigma_1 (W_1^T x) \right) \cdots \right) = f(x; \theta)$$

Input x is sent through several consecutive **layers** (functions). An output is produced.

Think of each layer as a (geometric) **transformation** of its input.

Every layer has a bunch of **parameters**, stored in the matrices W

Each family member of $\mathcal{F}$ is complex transformation broken down into a composition of simpler ones, parametrized by the layer weights.



Requirement (ii), expressibility, is satisfied. What about searchability?

How can they be trained?

Concretely, how are the parameters associated to each neuronal connection updated?

This is a fancy way to draw what's really just a composition of simple functions:

$$y = \sigma_{k+1}(W_{k+1}^T \sigma_k (W_k^T \left(\cdots \left(\sigma_1 (W_1^T x) \right) \cdots \right) = f(x;\theta)$$

Input x is sent through several consecutive **layers** (functions). An output is produced.

Think of each layer as a (geometric) **transformation** of its input.

Every layer has a bunch of **parameters**, stored in the matrices W

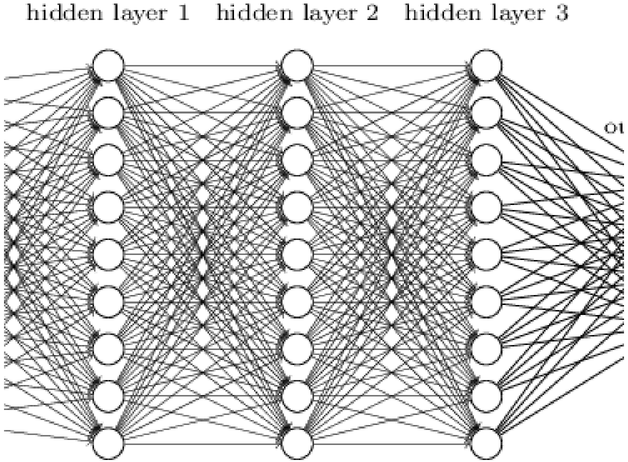
Each family member of $\mathcal{F}$ is complex transformation broken down into a composition of simpler ones, parametrized by the layer weights.

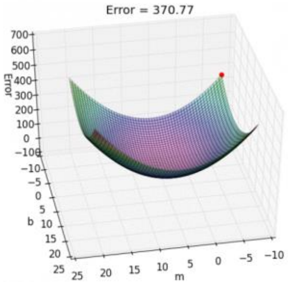
Three basic ingredients:

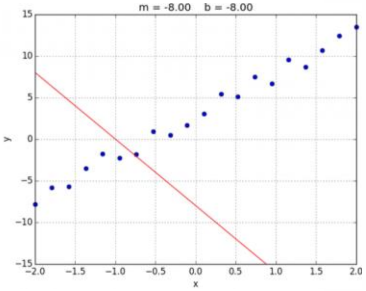
1. A **loss function**: Used to measure the network's performance on the training data. A **feedback signal** for the training process.

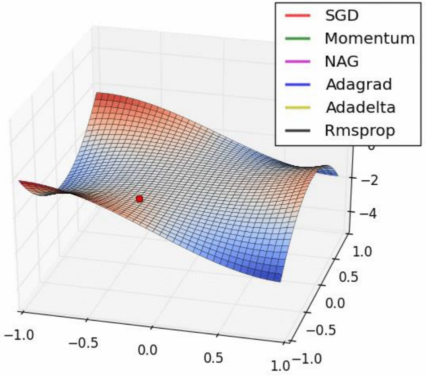
2. An **optimizer**: Used to update the *parameters* of the network to increase performance as measured by the loss function. **Gradient descent** and **backpropagation**.

3. One or more **metrics**: A way to score the model. For example *accuracy* or *mean squared error.*









Define a neural network model with an architecture and a set of trainable parameters (weights and biases)

1. Initialize the parameters

2.Collect a batch of inputs `(X,y)` and use the model with its current parameters to make predictions `y_pred`

3. Calculate how good the model is using the loss function to measure the discrepancy between the known labels `y_true` and the predictions `y_pred` for this batch

4. Calculate the gradients for each parameter using backpropagation. This measures how changing each parameter would change the loss. In other words, each parameter's contribution to the loss.

5. Move each weight a little bit in the opposite direction of its gradient using gradient descent. I.e., some variant of

```
parameter = parameter - learning_rate * gradient
```

6. Go back to step 2 and repeat the process

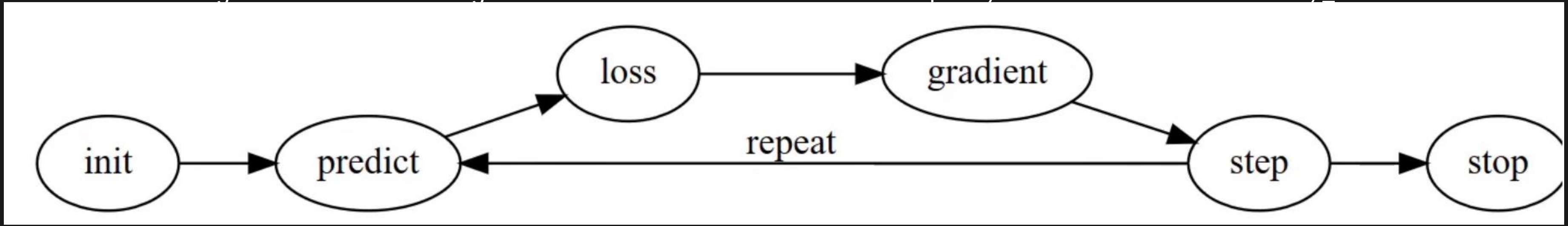
7. Do this over and over for several epochs (one epoch is one pass through all the training data), until you meet some chosen *stopping criteria*

Define a neural network model with an architecture and a set of trainable parameters (weights and biases)

1. Initialize the parameters

2.Collect a batch of inputs `(X,y)` and use the model with its current parameters to make predictions `y_pred`

3. Calculate how good the model is using the loss function to measure the discrepancy between the known labels `y_true` and the



5. Move each weight a little bit in the opposite direction of its gradient using gradient descent. I.e., some variant of
   `parameter = parameter - learning_rate * gradient`

6. Go back to step 2 and repeat the process

7. Do this over and over for several epochs (one epoch is one pass through all the training data), until you meet some chosen *stopping criteria*

Define a neural network model with an architecture and a set of trainable parameters (weights and biases)

1. Initialize the parameters

2.Collect a batch of inputs `(X,y)` and use the model with its current parameters to make predictions `y_pred`

3. Calculate how good the model is using the loss function to measure the discrepancy between the known labels `y_true` and the predictions `y_pred` fo [redacted]

What is gradient descent?

4. Calculate the gradi [redacted] er would change the loss. In other words, e [redacted]

5. Move each weight a little bit in the opposite direction of its gradient using gradient descent. I.e., some variant of

```
parameter = parameter - learning_rate * gradient
```

6. Go back to step 2 and repeat the process

7. Do this over and over for several epochs (one epoch is one pass through all the training data), until you meet some chosen *stopping criteria*

Define a neural network model with an architecture and a set of trainable parameters (weights and biases)

1.Init

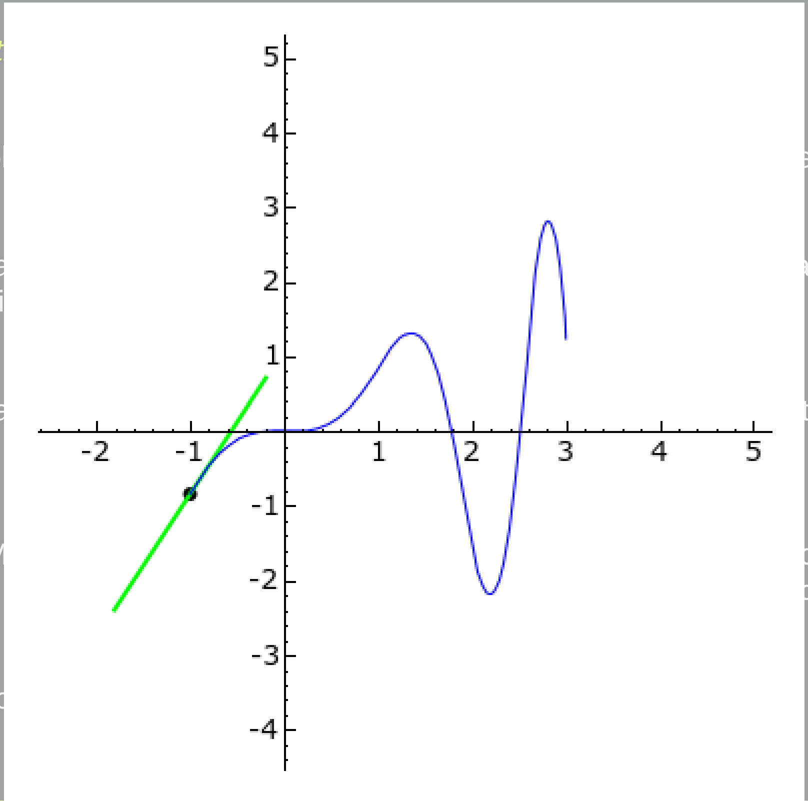
2.Co

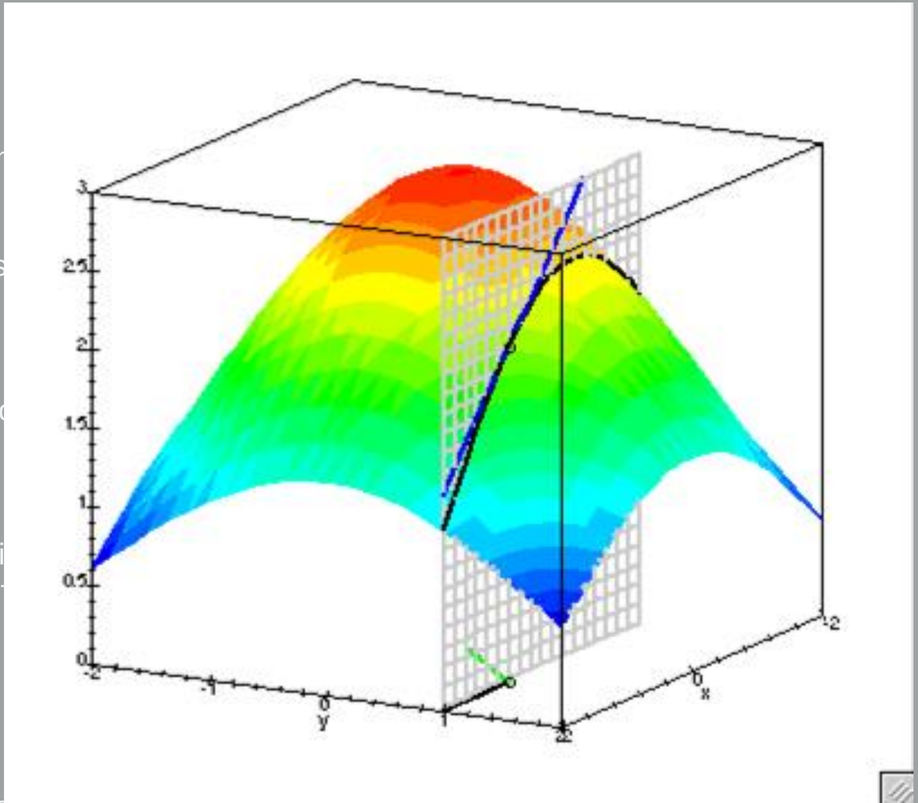
3. Ca
predi

4. Ca
loss.

5. M

6. Go





7. Do this over and over for several epochs (one epoch is one pass through all the training data), until you meet some chosen *stopping criteria*

Define a neural network model with an architecture and a set of trainable parameters (weights and biases)

1.Init

2.Co

3. Ca
predi

4. Ca
loss.

5. M

6. Go

7. Do this over an ... e chosen *stopping criteria*

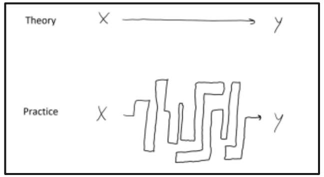
01 Machine learning: **function approximation** based on **training**

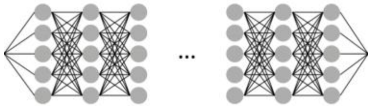
02 We pick the model family Ƒ. The parameters Θ are found automatically through training

03 In practice: some models are better suited to some tasks than others.

04 In deep learning: a particular choice of Ƒ. A **composition** of simple functions, trained jointly, typically using GPUs or other accelerators.

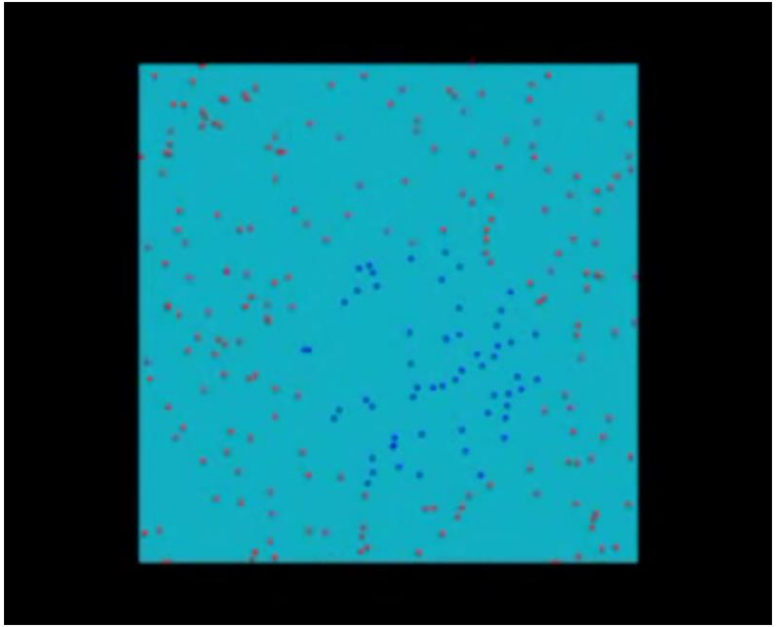
$$y \approx f(x;\theta)$$



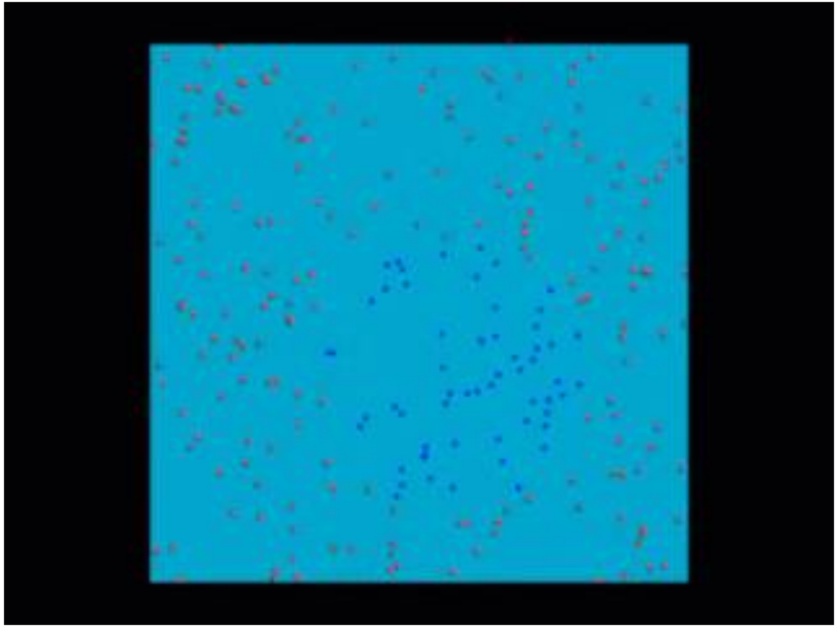


Representations and the power of transformations

Transformations

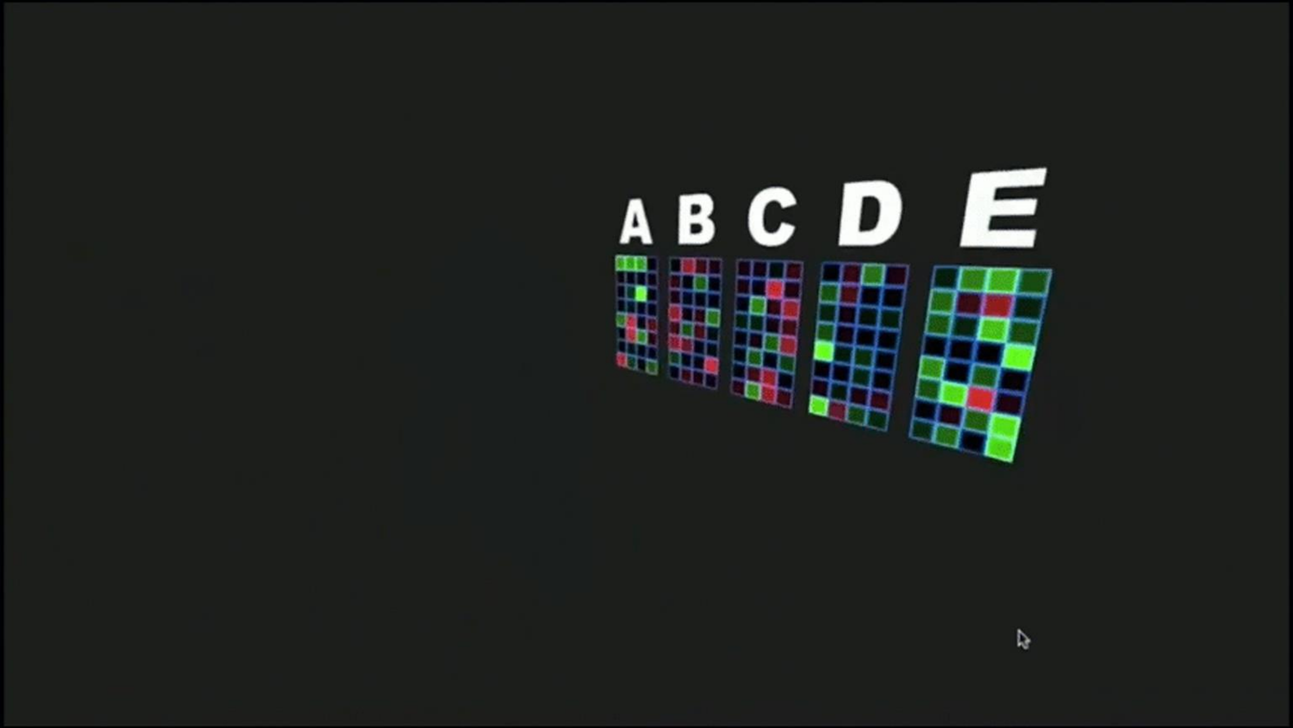


Transformation

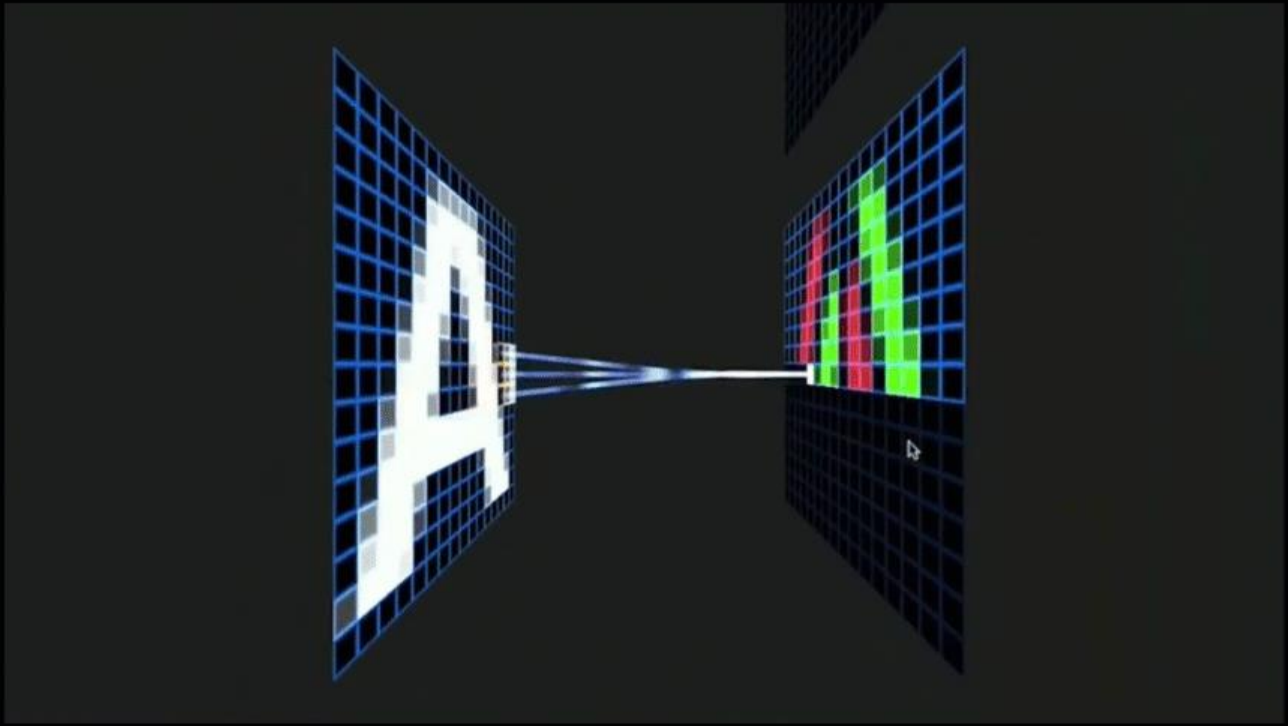


What transformations should be applied?

In deep learning: ***transformations providing useful representations automatically* found via training**



Source: Otavio Good



Source: Otavio Good

01 Machine learning: **function approximation** based on **training**

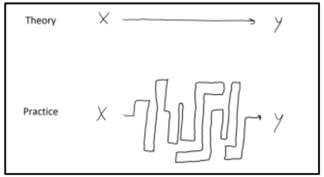
02 We pick the model family Ƒ. The parameters Θ are found automatically through training

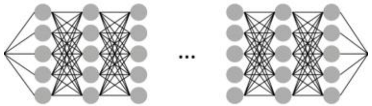
03 In practice: some models are better suited to some tasks than others.

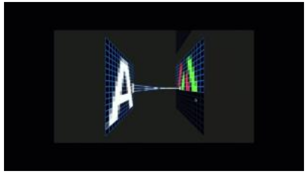
04 In deep learning: a particular choice of Ƒ. A **composition** of simple functions, trained jointly, typically using GPUs or other accelerators.

05 In deep learning: finding a **hierarchical set of geometric transformations**, ending in a representation that makes the desired task easy to solve.

$$y \approx f(x; \theta)$$







Deep learning

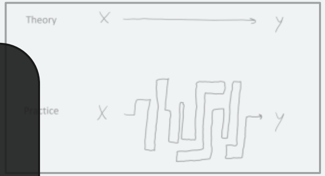
searching for good hierarchical geometric representations

01 Machine learning: **function approximation** based on **training**

$$y \approx f(x; \theta)$$

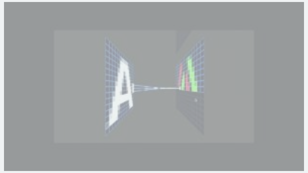
02 We pick the model family F. The parameters Θ are found automatically through training

03 In practice: some models are better suited to some tasks than others.



04 In deep learning: a particular choice of F. A **composition** of simple functions, trained jointly, typically using GPUs or other accelerators.

05 In deep learning: finding a **hierarchical set of geometric transformations**, ending in a representation that makes the desired task easy to solve.

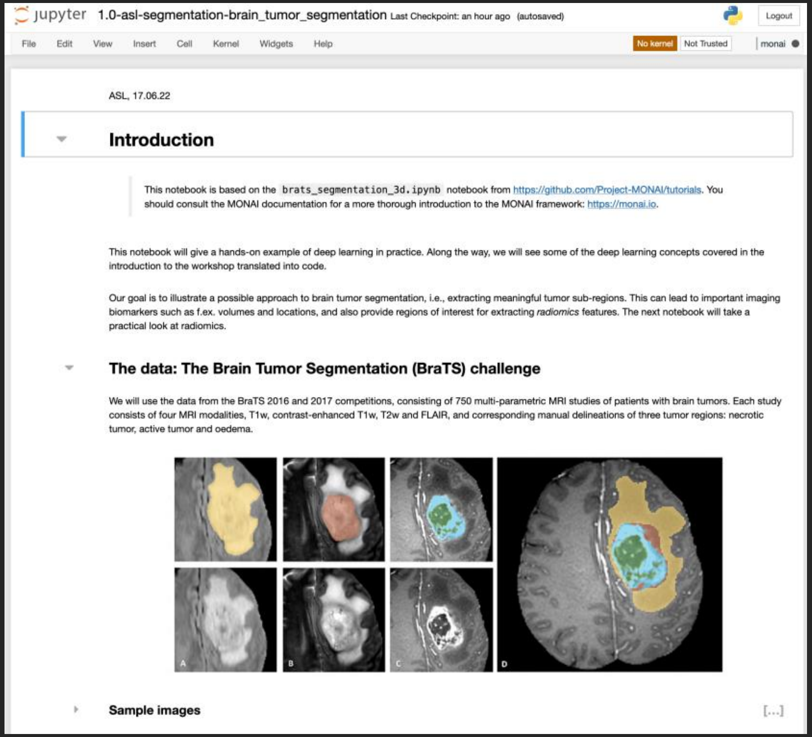


Plan

What *is* deep learning?

How do you *do* deep learning?

Some perspectives, opportunities and challenges



jupyter 1.0-asl-segmentation-brain_tumor_segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

No kernel | Not Trusted | monai

ASL, 17.06.22

Introduction

> This notebook is based on the `brats_segmentation_3d.ipynb` notebook from https://github.com/Project-MONAI/tutorials. You should consult the MONAI documentation for a more thorough introduction to the MONAI framework: https://monai.io.

This notebook will give a hands-on example of deep learning in practice. Along the way, we will see some of the deep learning concepts covered in the introduction to the workshop translated into code.

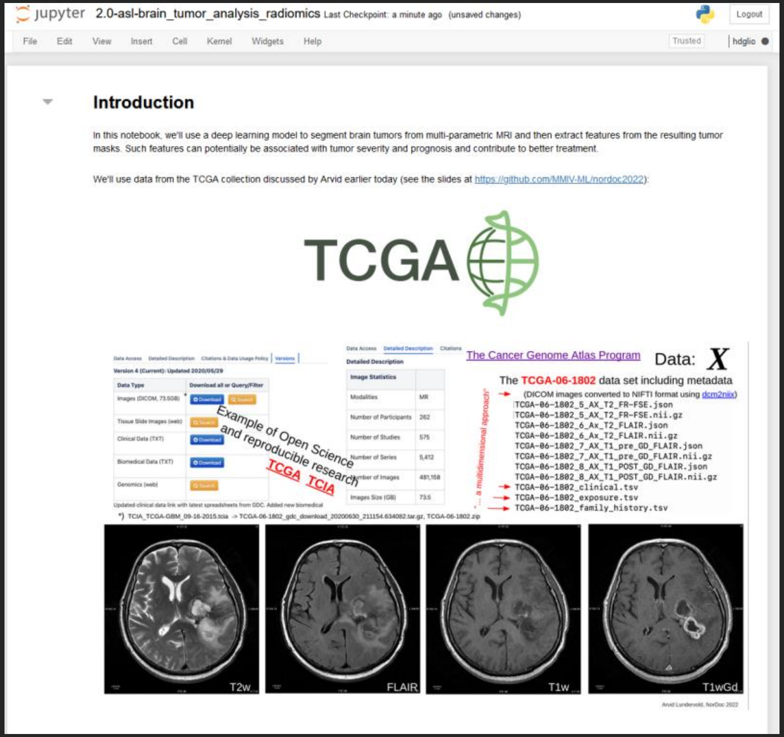
Our goal is to illustrate a possible approach to brain tumor segmentation, i.e., extracting meaningful tumor sub-regions. This can lead to important imaging biomarkers such as f.ex. volumes and locations, and also provide regions of interest for extracting *radiomics* features. The next notebook will take a practical look at radiomics.

The data: The Brain Tumor Segmentation (BraTS) challenge

We will use the data from the BraTS 2016 and 2017 competitions, consisting of 750 multi-parametric MRI studies of patients with brain tumors. Each study consists of four MRI modalities, T1w, contrast-enhanced T1w, T2w and FLAIR, and corresponding manual delineations of three tumor regions: necrotic tumor, active tumor and oedema.

Sample images

[...]



jupyter 2.0-asl-brain_tumor_analysis_radiomics Last Checkpoint: a minute ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | hdglio

Introduction

In this notebook, we'll use a deep learning model to segment brain tumors from multi-parametric MRI and then extract features from the resulting tumor masks. Such features can potentially be associated with tumor severity and prognosis and contribute to better treatment.

We'll use data from the TCGA collection discussed by Arvid earlier today (see the slides at https://github.com/MMIV-ML/nordoc2022):

Plan

What *is* deep learning?

How do you *do* deep learning?

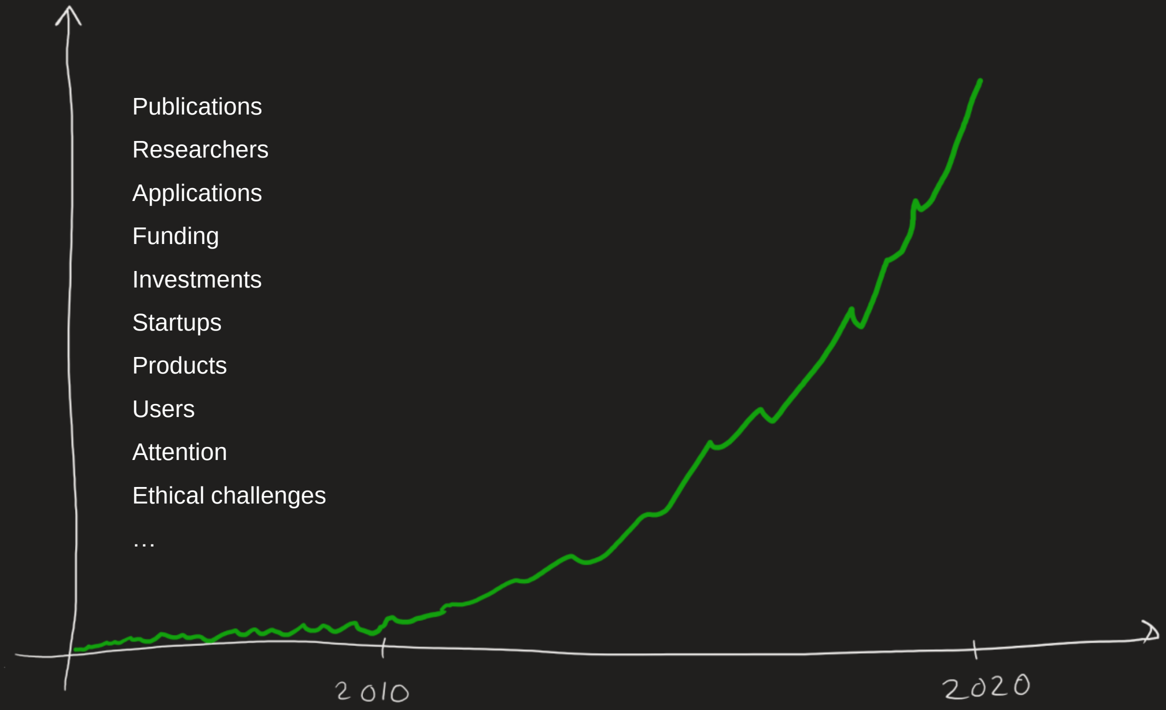
Some perspectives, opportunities and challenges

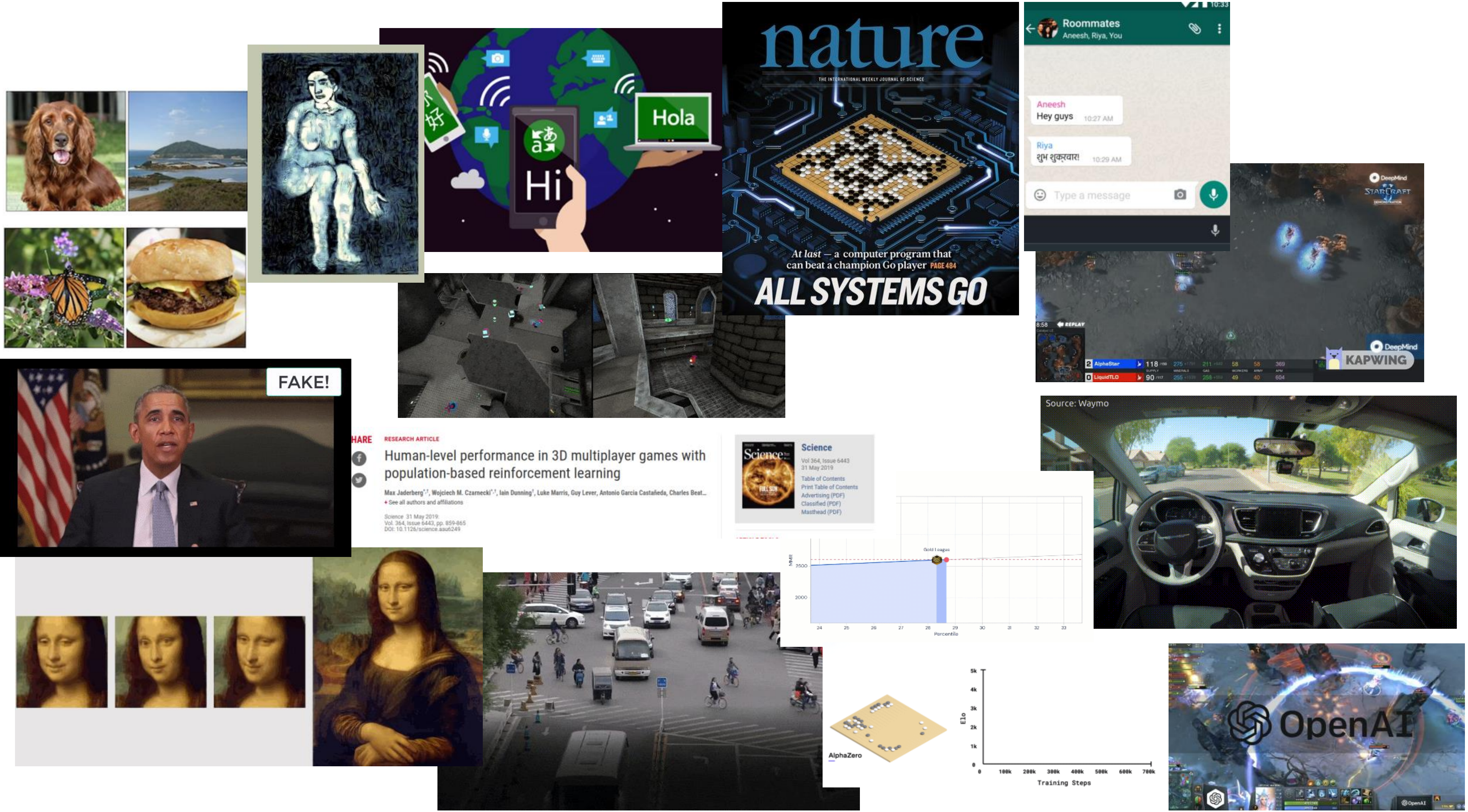
Artificial intelligence

Machine learning

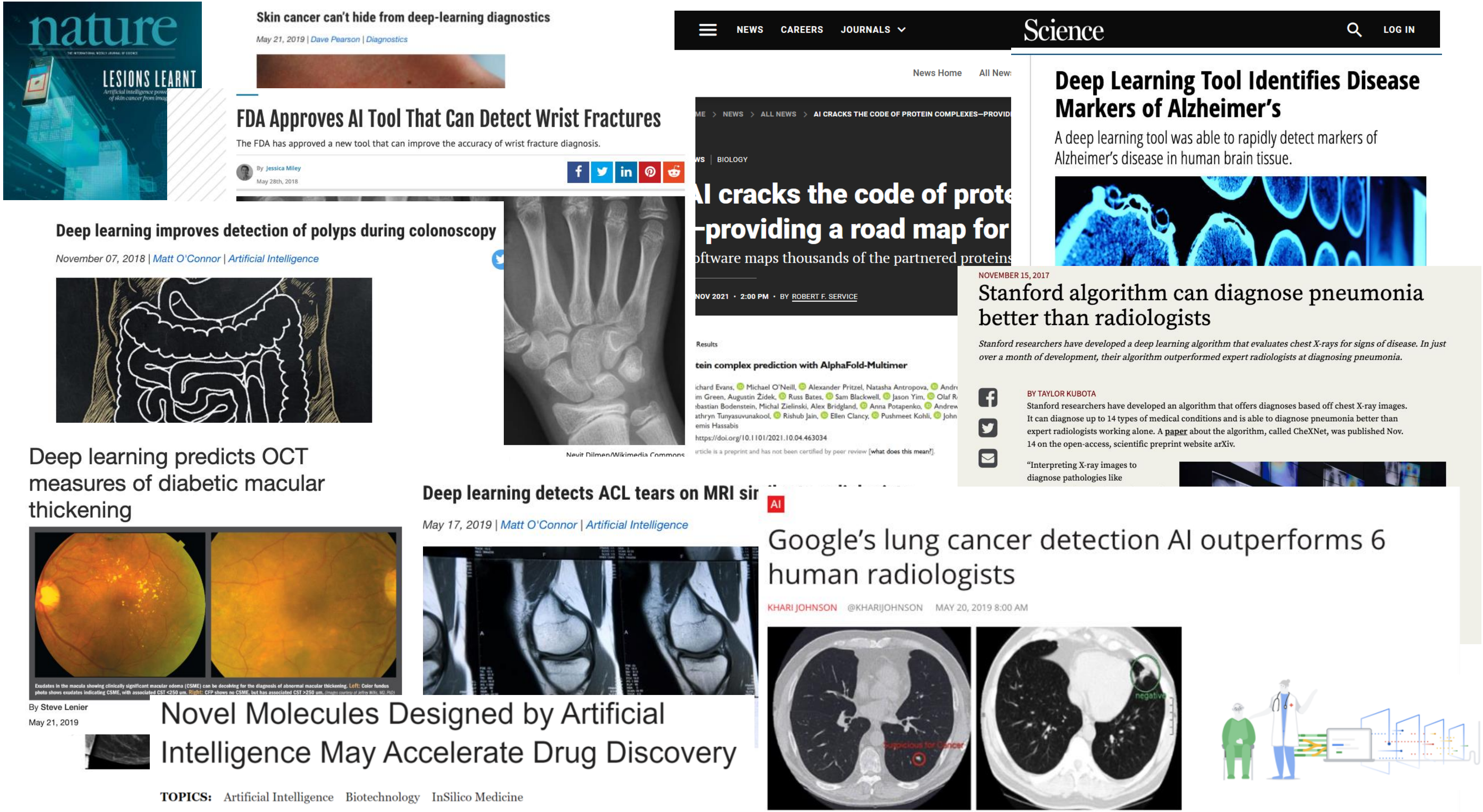
Deep learning

Representation learners





Hola
Hi
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
Roommates
Aneesh, Riya, You
Aneesh
Hey guys 10:27 AM
Riya
शुभ शुक्रवार! 10:29 AM
Type a message
DeepMind
KAPWING
FAKE!
RESEARCH ARTICLE
Human-level performance in 3D multiplayer games with population-based reinforcement learning
Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beat...
See all authors and affiliations
Science 31 May 2019:
Vol. 364, Issue 6443, pp. 859-865
DOI: 10.1126/science.aau6249
Science
Vol 364, Issue 6443
31 May 2019
Table of Contents
Print Table of Contents
Advertising (PDF)
Classified (PDF)
Masthead (PDF)
Source: Waymo
Gold League
Percentile
AlphaZero
Elo
Training Steps
OpenAI

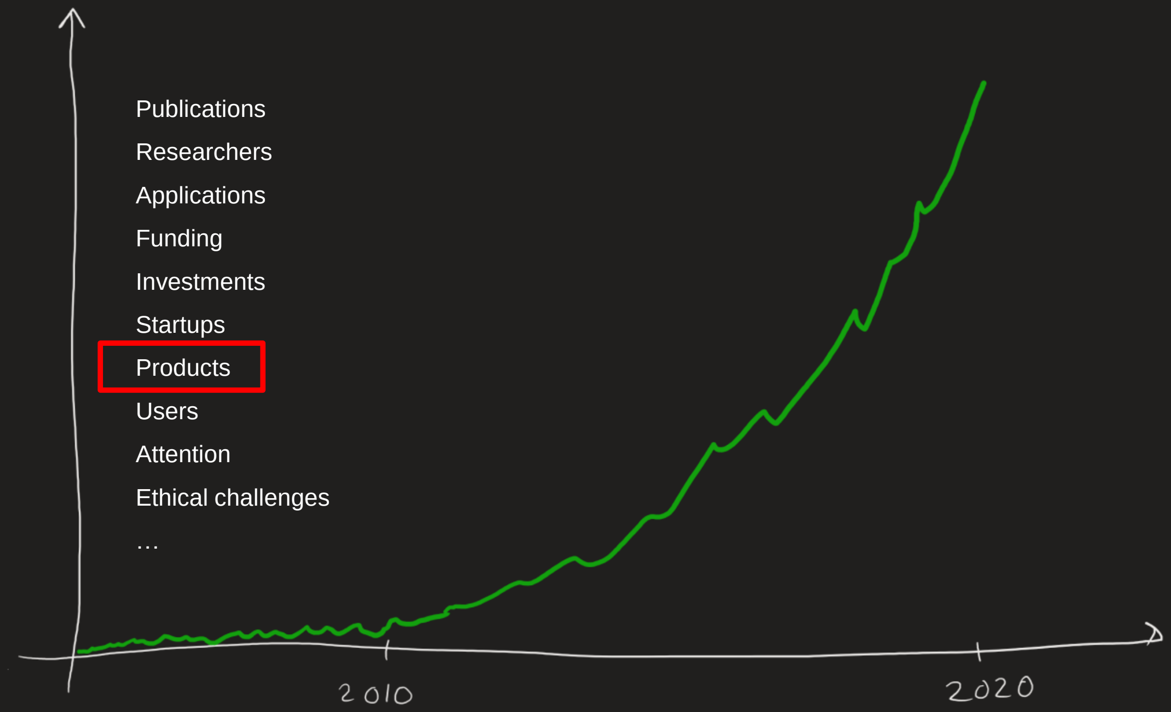

nature
LESIONS LEARNT
Skin cancer can't hide from deep-learning diagnostics
May 21, 2019 | Dave Pearson | Diagnostics
FDA Approves AI Tool That Can Detect Wrist Fractures
The FDA has approved a new tool that can improve the accuracy of wrist fracture diagnosis.
By Jessica Miley
May 28th, 2018
NEWS
CAREERS
JOURNALS
Science
LOG IN
News Home
All News
NEWS
ALL NEWS
AI CRACKS THE CODE OF PROTEIN COMPLEXES—PROVID
BIOLOGY
AI cracks the code of prote
-providing a road map for
oftware maps thousands of the partnered proteins
NOV 2021 · 2:00 PM · BY ROBERT F. SERVICE
Deep Learning Tool Identifies Disease Markers of Alzheimer's
A deep learning tool was able to rapidly detect markers of Alzheimer's disease in human brain tissue.
Deep learning improves detection of polyps during colonoscopy
November 07, 2018 | Matt O'Connor | Artificial Intelligence
Nevit Dilmen/Wikimedia Commons
Results
tein complex prediction with AlphaFold-Multimer
https://doi.org/10.1101/2021.10.04.463034
NOVEMBER 15, 2017
Stanford algorithm can diagnose pneumonia better than radiologists
Stanford researchers have developed a deep learning algorithm that evaluates chest X-rays for signs of disease. In just over a month of development, their algorithm outperformed expert radiologists at diagnosing pneumonia.
BY TAYLOR KUBOTA
Stanford researchers have developed an algorithm that offers diagnoses based off chest X-ray images. It can diagnose up to 14 types of medical conditions and is able to diagnose pneumonia better than expert radiologists working alone. A paper about the algorithm, called CheXNet, was published Nov. 14 on the open-access, scientific preprint website arXiv.
"Interpreting X-ray images to diagnose pathologies like
Deep learning predicts OCT measures of diabetic macular thickening
By Steve Lenier
May 21, 2019
Deep learning detects ACL tears on MRI si
May 17, 2019 | Matt O'Connor | Artificial Intelligence
AI
Google's lung cancer detection AI outperforms 6 human radiologists
KHARI JOHNSON @KHARIJOHNSON MAY 20, 2019 8:00 AM
Novel Molecules Designed by Artificial Intelligence May Accelerate Drug Discovery
TOPICS: Artificial Intelligence Biotechnology InSilico Medicine

Artificial intelligence

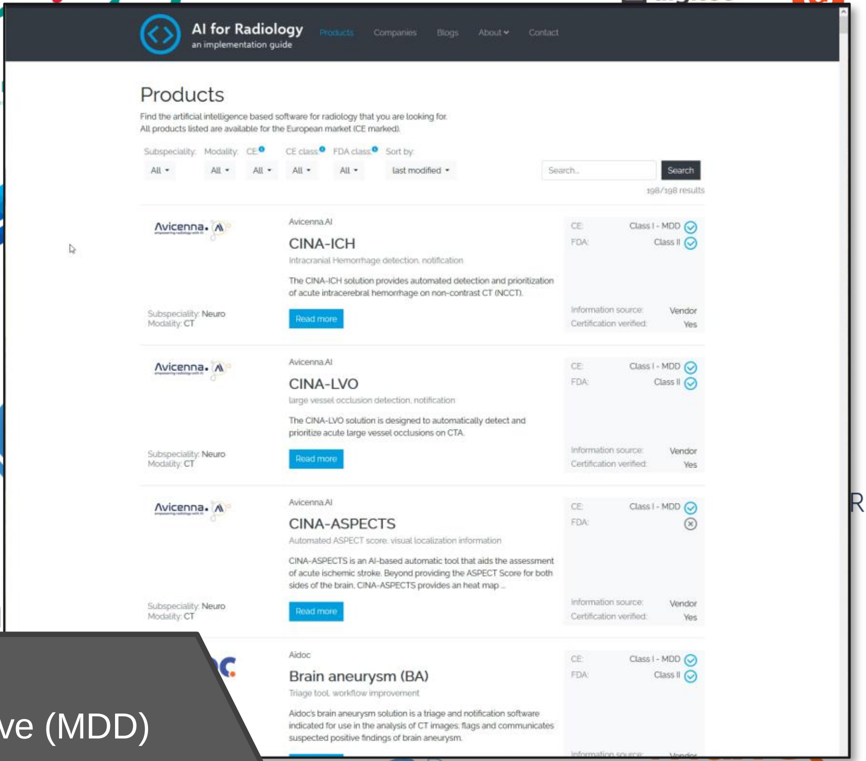
Machine learning

Deep learning

Representation learners



All of these are available on the European market per May 2022 (CE marked)



OBS:
Medical Device Directive (MDD)
versus
Medical Device Regulation (MDR)

Narrow

Tasks

Modalities

Anatomical regions

…

Data availability

Competency availability

Medical feasibility

Technical feasibility

Organizational feasibility

Regulatory feasibility

Theoretical limitations

Incentives

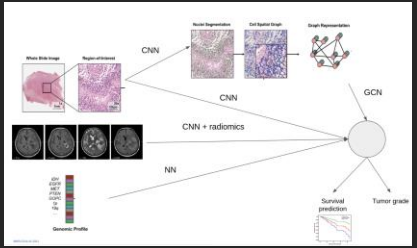
…

The research side is somewhat broader…

…trying to investigate how to integrate heterogeneous data about patients and disease processes.

Two examples:

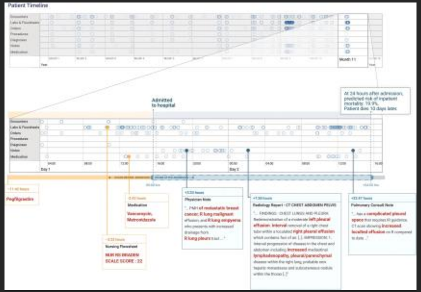
Integrated diagnostics



From brain tumor imaging and genetics to survival prediction

Chen et.al. IEEE Trans Med Imaging, 2022, 10.1109/TMI.2020.3021387

Deep learning and electronic health records



From EHR to length-of-stay and in-hospital mortality

Rajkomar et.al., NPJ Digit Med, 2018, 10.1038/s41746-018-0029-1

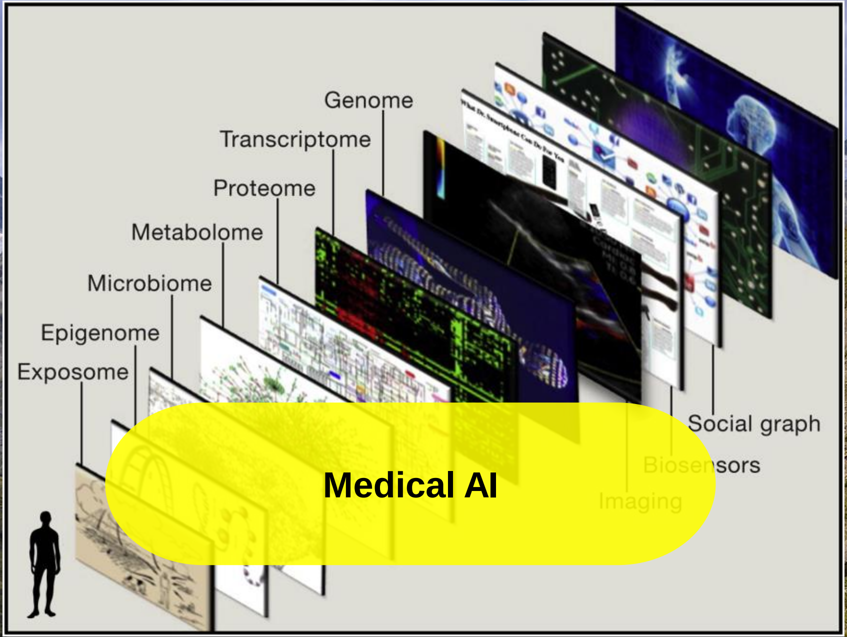


Illustration from E. Topol, Cell, 157(1), 2014

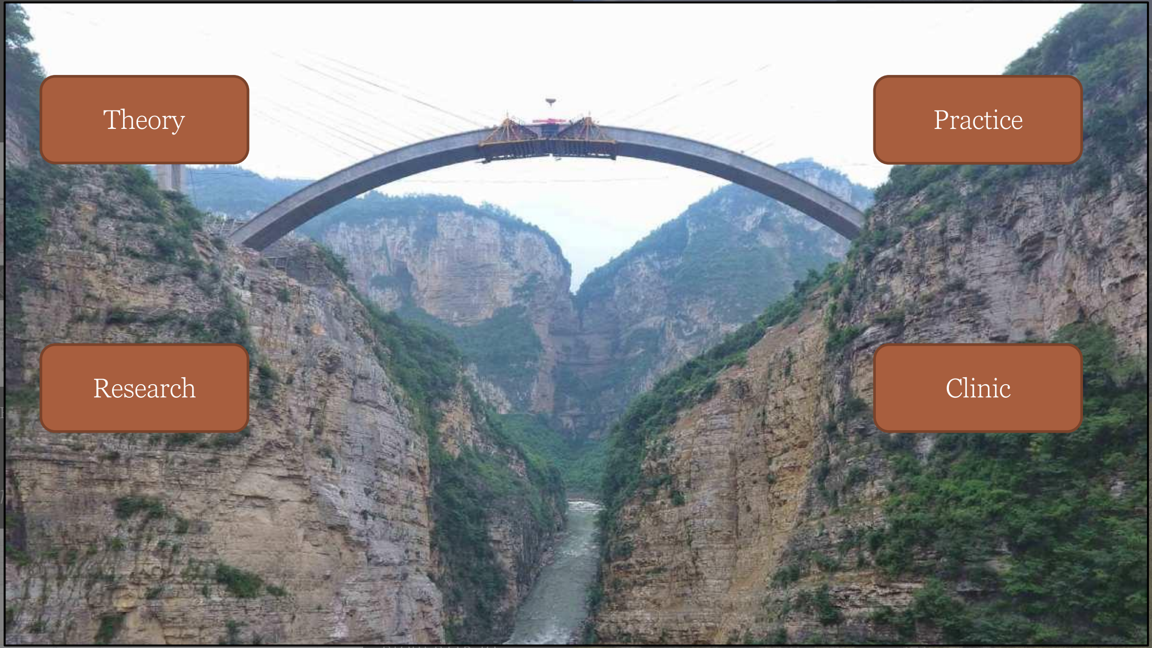
The research side is somewhat broader…

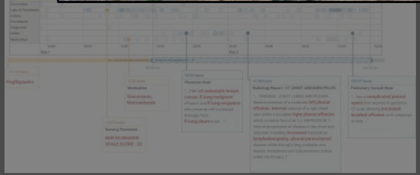
…trying to investigate how to integrate heterogeneous data about patients and disease processes.

Two examples:

Chen et.al. I…

Deep lea…





From EHR to length-of-stay and in-hospital mortality

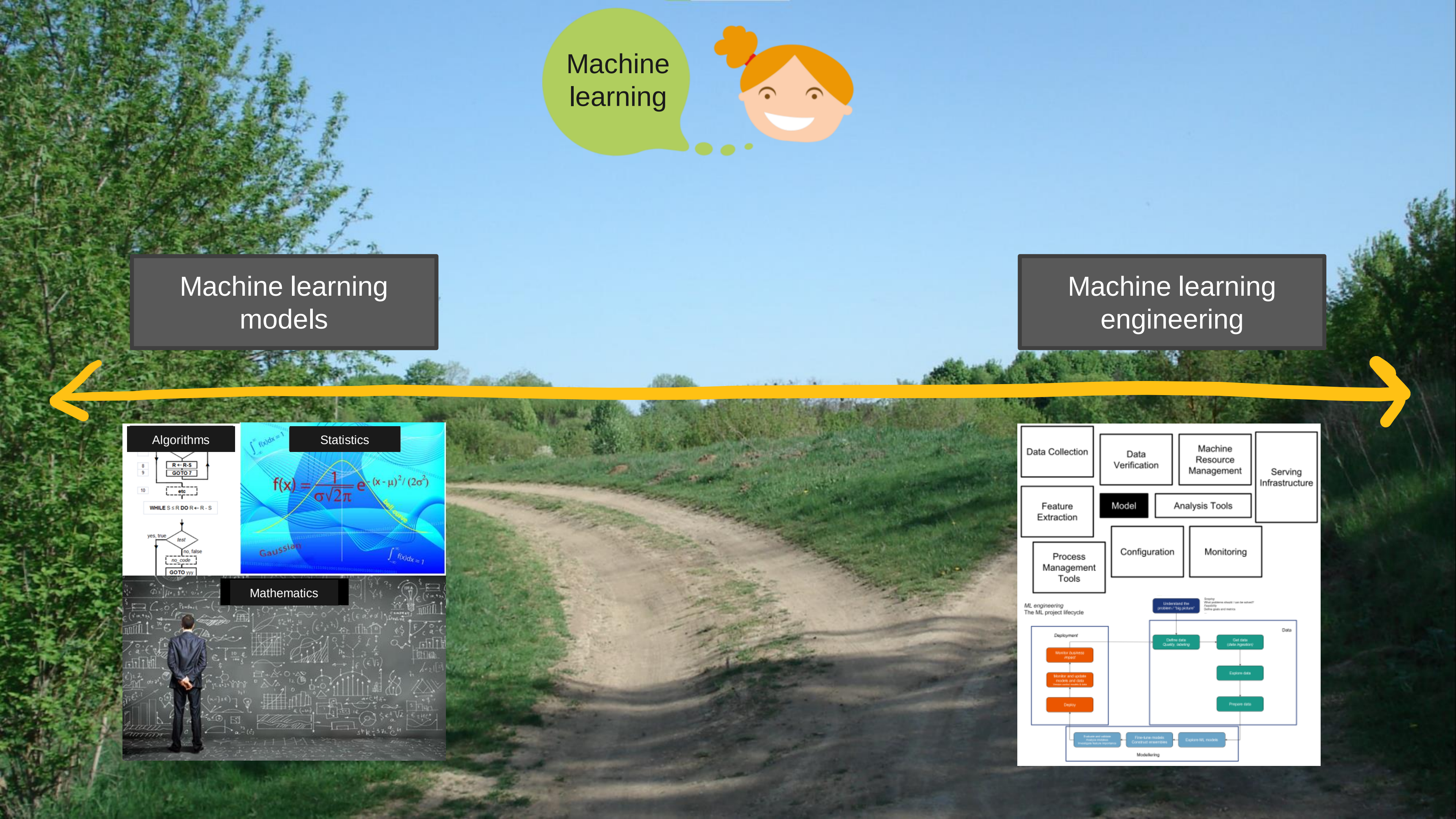
Rajkomar et.al., NPJ Digit Med, 2018, 10.1038/s41746-018-0029-1

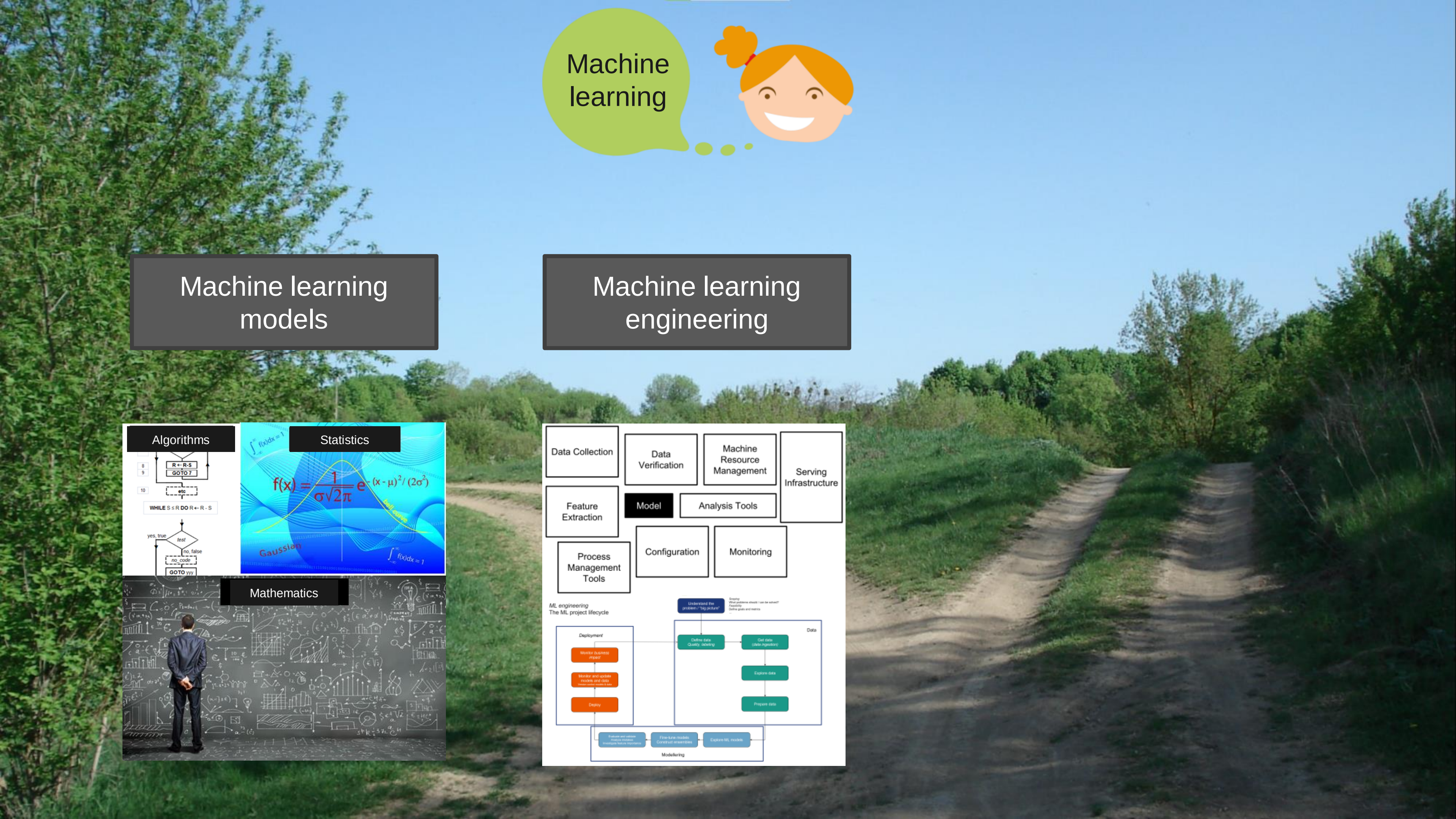
Machine learning in medicine

Practical machine learning
&
Machine learning engineering

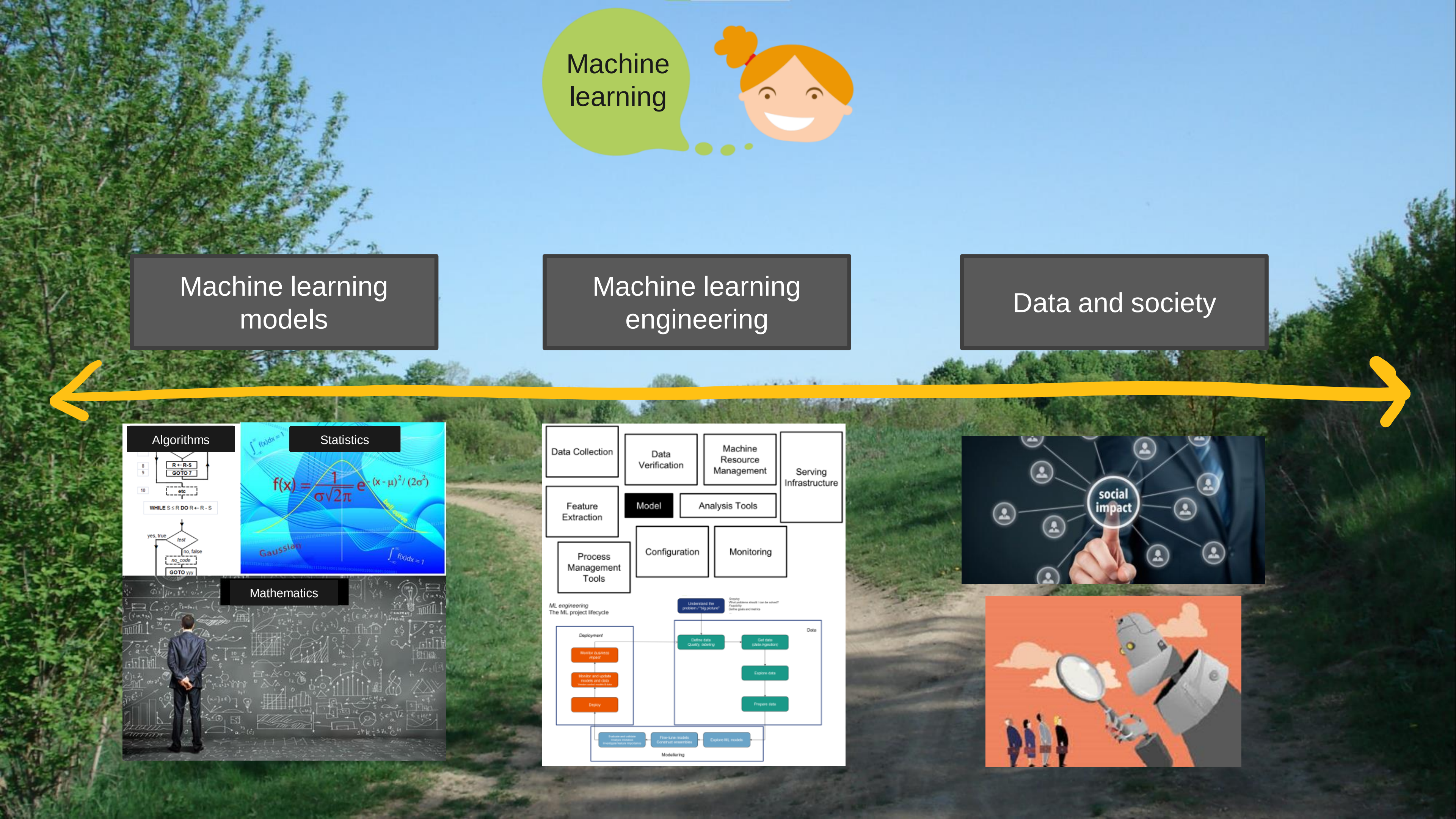

Machine learning
Machine learning
Machine learning


Machine learning
Machine learning models
Machine learning engineering


Machine learning
Machine learning models
Machine learning engineering
Algorithms
Statistics
Mathematics
$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$
Gaussian
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Feature Extraction
Model
Analysis Tools
Process Management Tools
Configuration
Monitoring
ML engineering
The ML project lifecycle
Deployment
Data
Modelling



Machine learning
Machine learning models
Machine learning engineering
Algorithms
Statistics
Mathematics
$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$
Gaussian
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Feature Extraction
Model
Analysis Tools
Process Management Tools
Configuration
Monitoring
ML engineering
The ML project lifecycle
Understand the problem / "big picture"
Data
Define data
Quality, labeling
Get data
(data ingestion)
Explore data
Prepare data
Deployment
Monitor business impact
Monitor and update models and data
Deploy
Fine-tune models
Construct ensembles
Explore ML models
Modellering



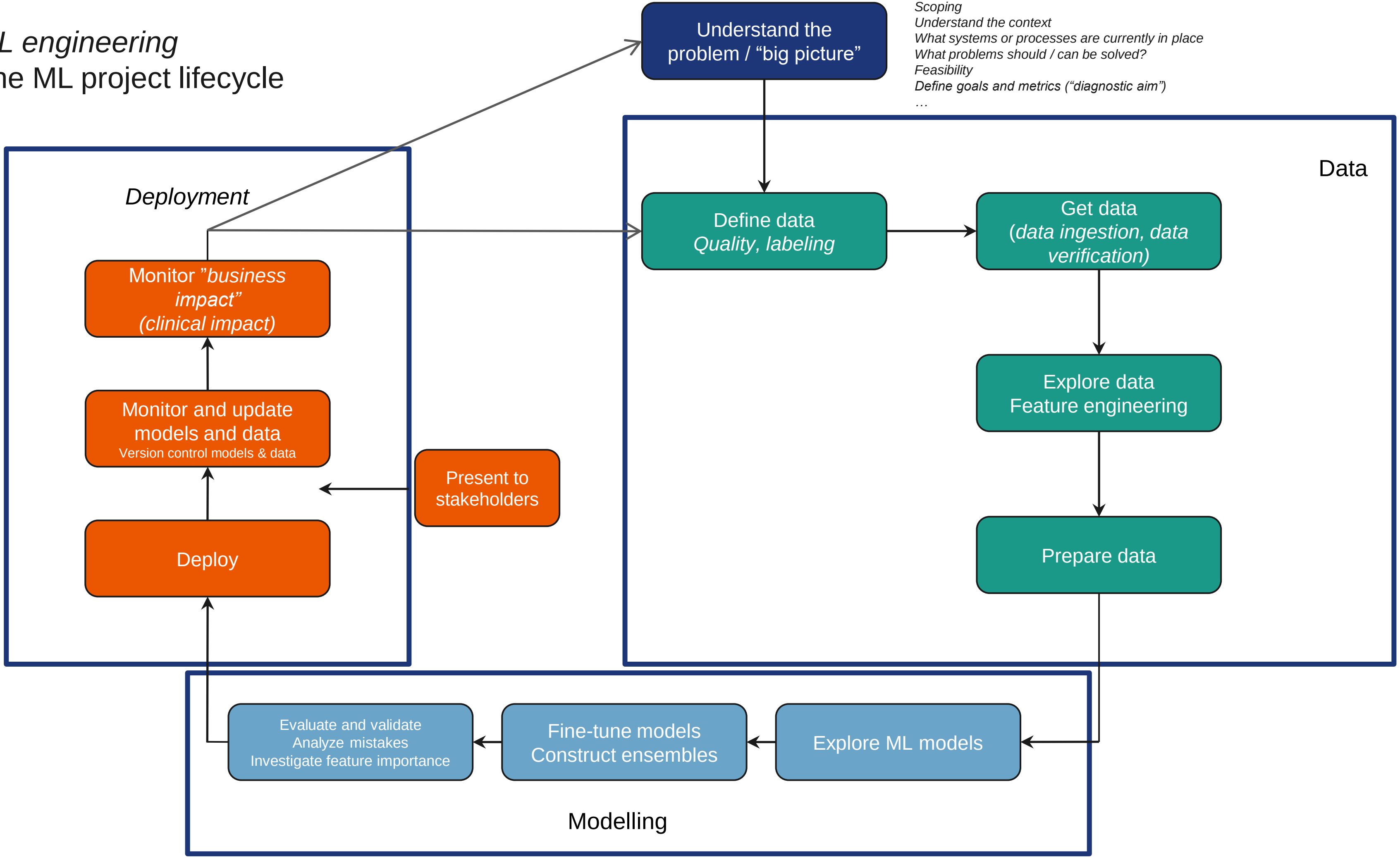
Machine learning
Machine learning models
Machine learning engineering
Data and society
Algorithms
Statistics
Mathematics
Gaussian
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Feature Extraction
Model
Analysis Tools
Configuration
Monitoring
Process Management Tools
ML engineering
The ML project lifecycle
Deployment
Data
social impact

How do you make a machine learning-based solution?

ML engineering

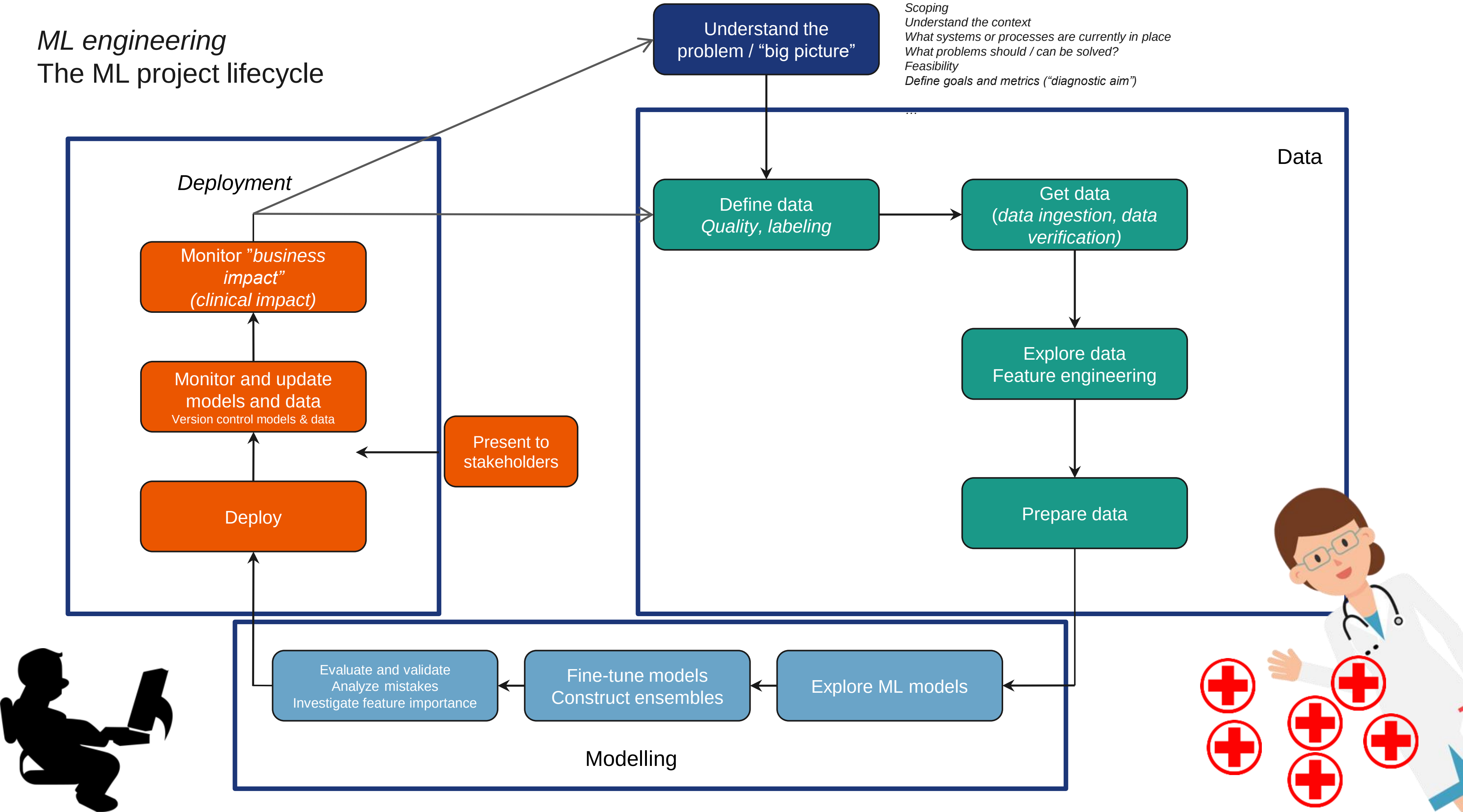
The ML project lifecycle

Scoping
Understand the context
What systems or processes are currently in place
What problems should / can be solved?
Feasibility
Define goals and metrics ("diagnostic aim")
…



ML engineering

The ML project lifecycle



Model

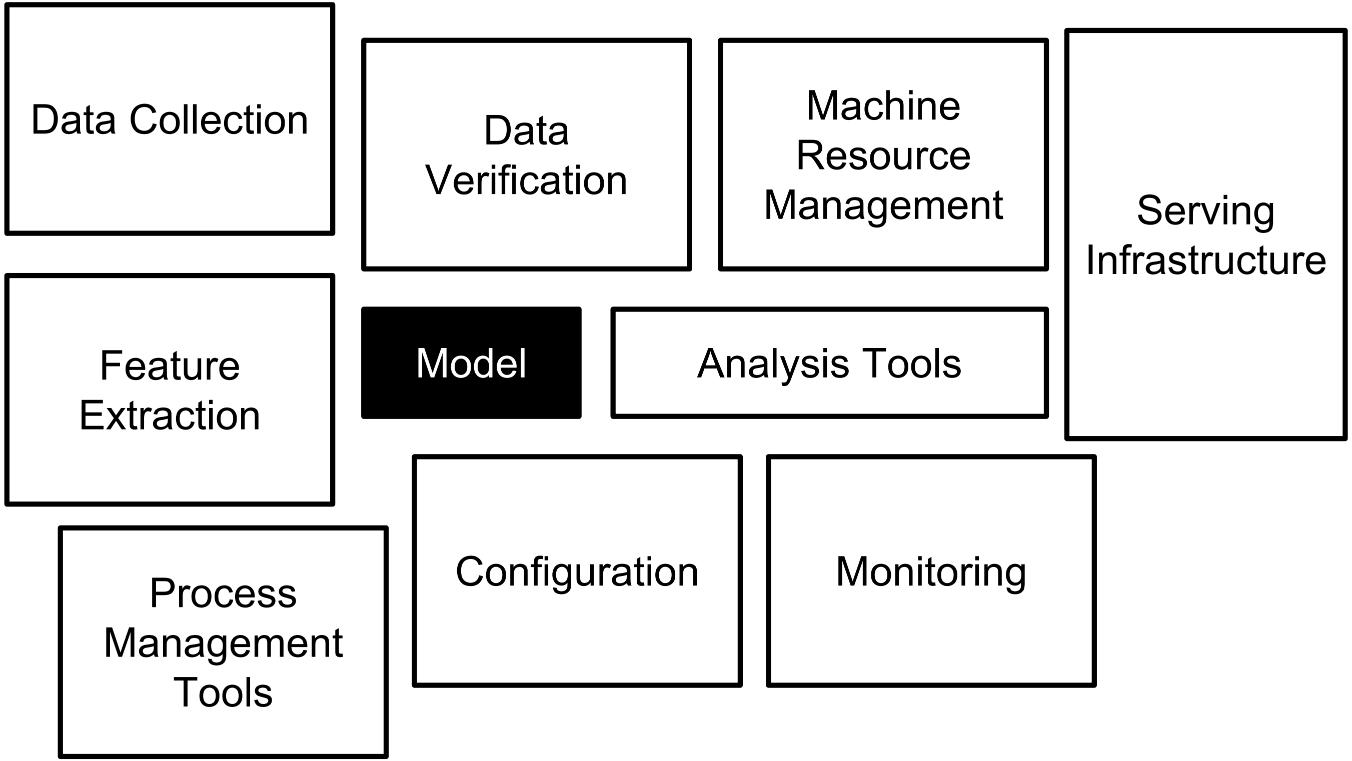
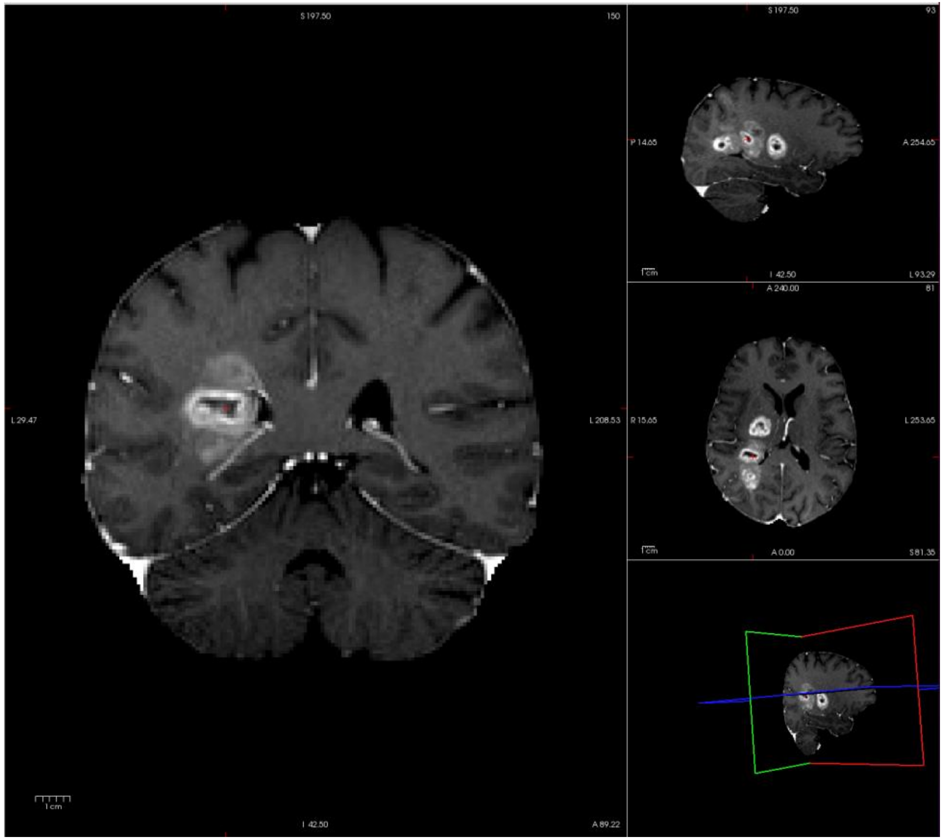


Illustration from https://developers.google.com/machine-learning/testing-debugging/pipeline/overview

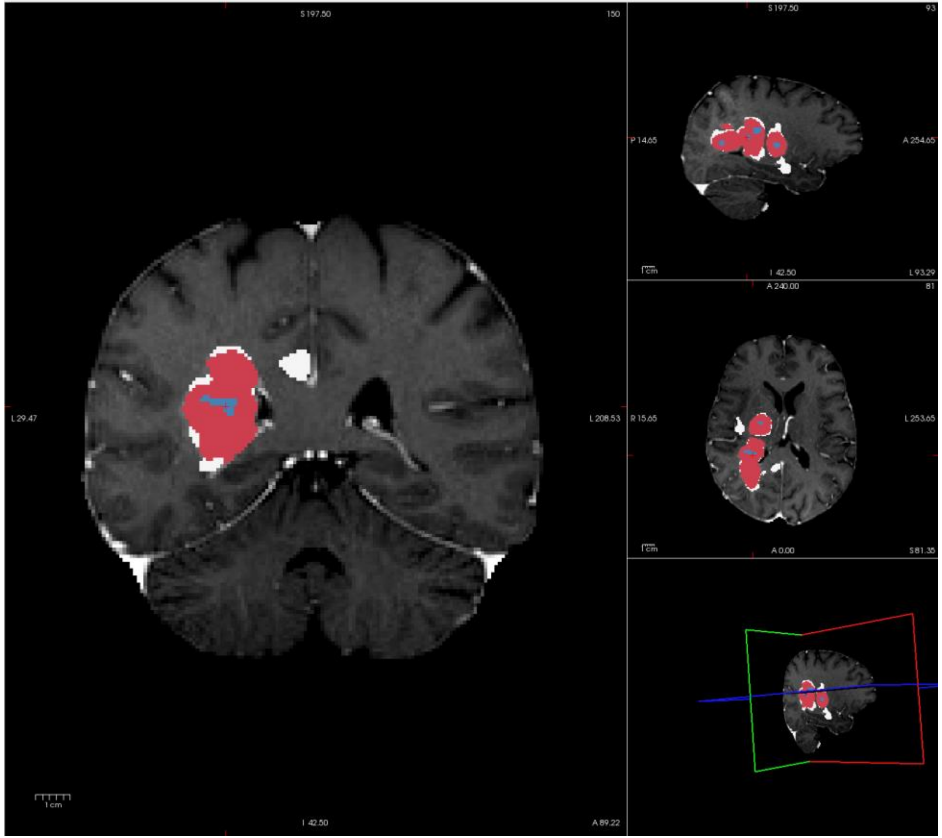

Machine learning in medicine

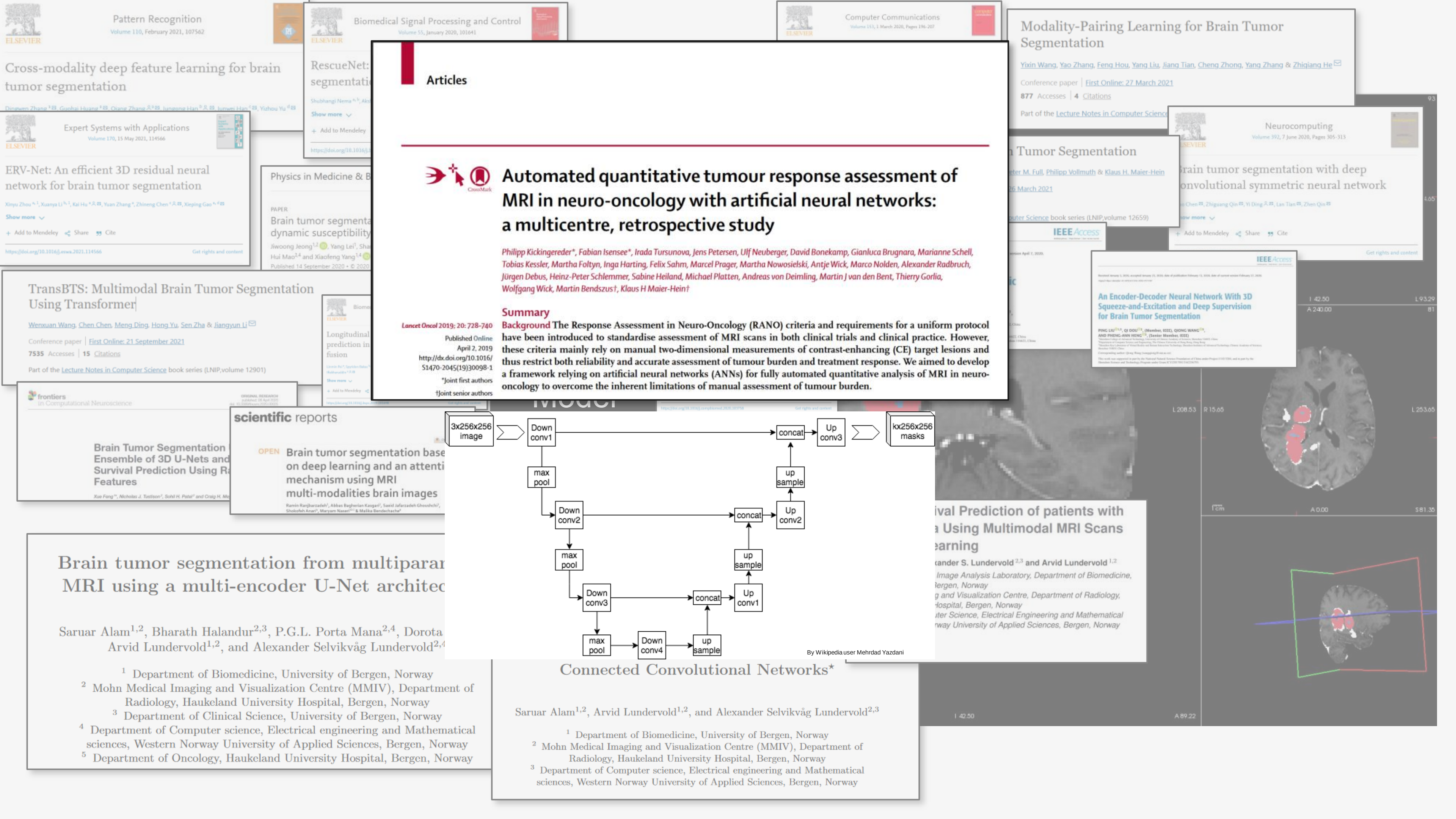
Model

Model

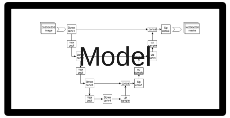


Model


S 197.50
150
L 29.47
L 208.53
1 cm
I 42.50
A 89.22
S 197.50
93
P 14.65
A 254.65
I 42.50
L 93.29
A 240.00
81
R 15.65
L 253.65
A 0.00
S 81.35



Pattern Recognition
Volume 110, February 2021, 107562
Cross-modality deep feature learning for brain tumor segmentation
Biomedical Signal Processing and Control
Volume 55, January 2020, 101641
Computer Communications
Modality-Pairing Learning for Brain Tumor Segmentation
Yixin Wang, Yao Zhang, Feng Hou, Yang Liu, Jiang Tian, Cheng Zhong, Yang Zhang & Zhiqiang He
Conference paper | First Online: 27 March 2021
877 Accesses | 4 Citations
Expert Systems with Applications
Volume 170, 15 May 2021, 114566
ERV-Net: An efficient 3D residual neural network for brain tumor segmentation
Neurocomputing
Physics in Medicine & B
TransBTS: Multimodal Brain Tumor Segmentation Using Transformer
Wenxuan Wang, Chen Chen, Meng Ding, Hong Yu, Sen Zha & Jiangyun Li
Conference paper | First Online: 21 September 2021
7535 Accesses | 15 Citations
Part of the Lecture Notes in Computer Science book series (LNIP, volume 12901)
Articles
Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study
Philipp Kickingereder*, Fabian Isensee*, Irada Tursunova, Jens Petersen, Ulf Neuberger, David Bonekamp, Gianluca Brugnara, Marianne Schell, Tobias Kessler, Martha Foltyn, Inga Harting, Felix Sahm, Marcel Prager, Martha Nowosielski, Antje Wick, Marco Nolden, Alexander Radbruch, Jürgen Debus, Heinz-Peter Schlemmer, Sabine Heiland, Michael Platten, Andreas von Deimling, Martin J van den Bent, Thierry Gorlia, Wolfgang Wick, Martin Bendszus†, Klaus H Maier-Hein†
Summary
Background The Response Assessment in Neuro-Oncology (RANO) criteria and requirements for a uniform protocol have been introduced to standardise assessment of MRI scans in both clinical trials and clinical practice. However, these criteria mainly rely on manual two-dimensional measurements of contrast-enhancing (CE) target lesions and thus restrict both reliability and accurate assessment of tumour burden and treatment response. We aimed to develop a framework relying on artificial neural networks (ANNs) for fully automated quantitative analysis of MRI in neuro-oncology to overcome the inherent limitations of manual assessment of tumour burden.
Lancet Oncol 2019; 20: 728–740
Published Online April 2, 2019
http://dx.doi.org/10.1016/S1470-2045(19)30098-1
*Joint first authors
†Joint senior authors
An Encoder-Decoder Neural Network With 3D Squeeze-and-Excitation and Deep Supervision for Brain Tumor Segmentation
scientific reports
Brain Tumor Segmentation Ensemble of 3D U-Nets and Survival Prediction Using Features
OPEN Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images
3x256x256 image
Down conv1
max pool
Down conv2
max pool
Down conv3
max pool
Down conv4
up sample
concat
Up conv1
up sample
concat
Up conv2
up sample
concat
Up conv3
kx256x256 masks
By Wikipedia user Mehrdad Yazdani
Brain tumor segmentation from multiparametric MRI using a multi-encoder U-Net architecture
Saruar Alam[1,2], Bharath Halandur[2,3], P.G.L. Porta Mana[2,4], Dorota
Arvid Lundervold[1,2], and Alexander Selvikvåg Lundervold[2,4]
1 Department of Biomedicine, University of Bergen, Norway
2 Mohn Medical Imaging and Visualization Centre (MMIV), Department of Radiology, Haukeland University Hospital, Bergen, Norway
3 Department of Clinical Science, University of Bergen, Norway
4 Department of Computer science, Electrical engineering and Mathematical sciences, Western Norway University of Applied Sciences, Bergen, Norway
5 Department of Oncology, Haukeland University Hospital, Bergen, Norway
Connected Convolutional Networks*
Saruar Alam[1,2], Arvid Lundervold[1,2], and Alexander Selvikvåg Lundervold[2,3]
1 Department of Biomedicine, University of Bergen, Norway
2 Mohn Medical Imaging and Visualization Centre (MMIV), Department of Radiology, Haukeland University Hospital, Bergen, Norway
3 Department of Computer science, Electrical engineering and Mathematical sciences, Western Norway University of Applied Sciences, Bergen, Norway



Model
3x256x256 image
Down conv1
max pool
concat
Up conv3
up sample
Up conv2
max pool
Down conv3
concat
Up conv1
up sample
max pool
Down conv4
up sample

In radiology, solutions must be compatible with existing infrastructure!

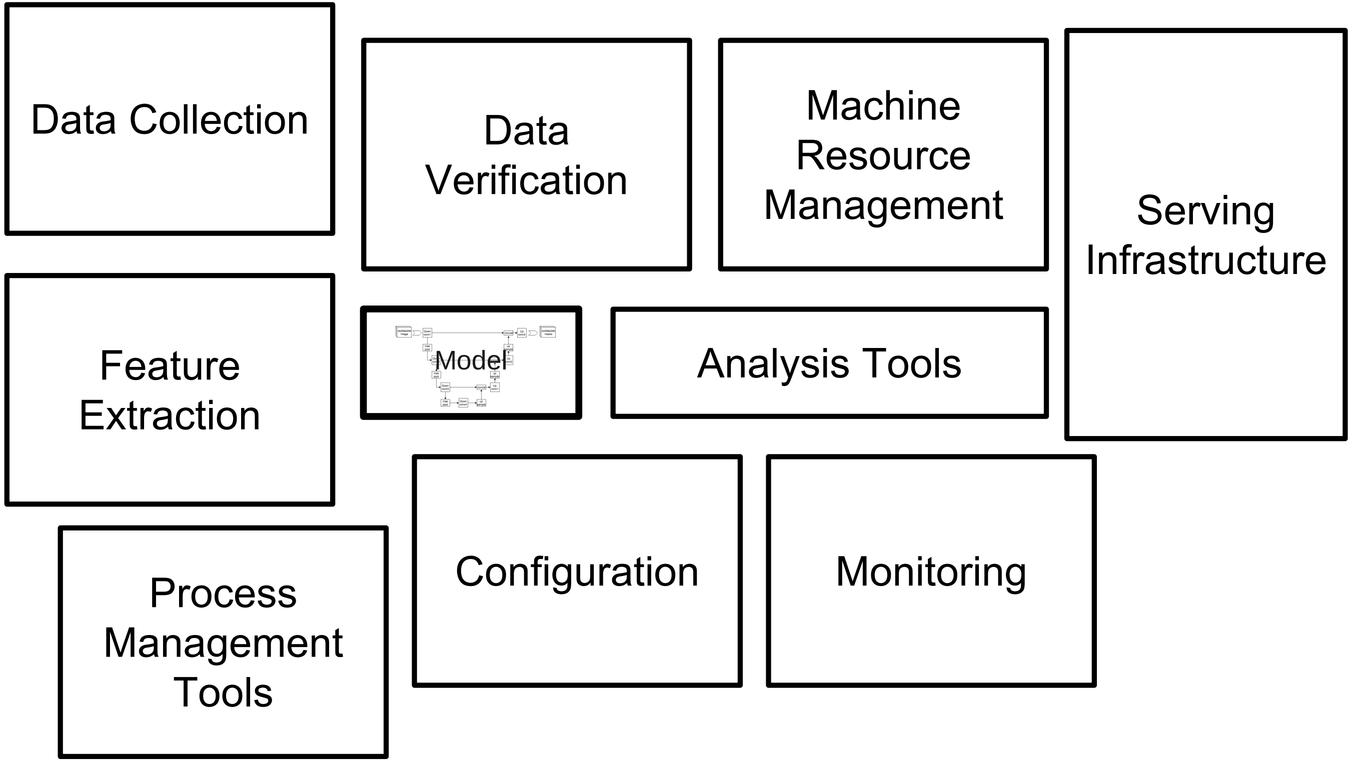


Illustration from https://developers.google.com/machine-learning/testing-debugging/pipeline/overview

In radiology, solutions must be compatible with existing infrastructure!

Hauke Bartsch
Computer Science

Zhanbolat Satybaldinov
Software engineering

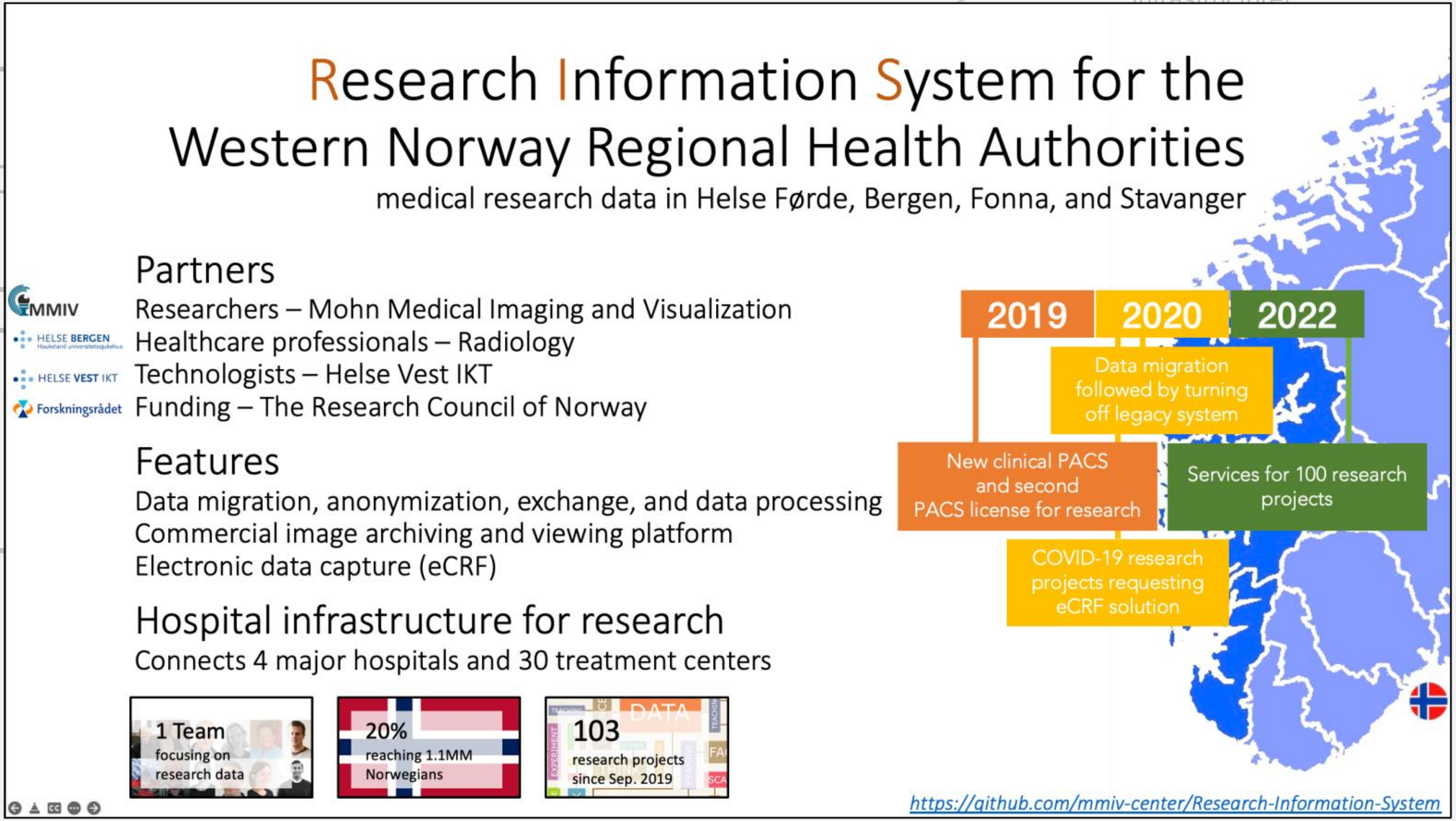


Illustration from https://developers.google.com/machine-learning/testing-debugging/pipeline/overview

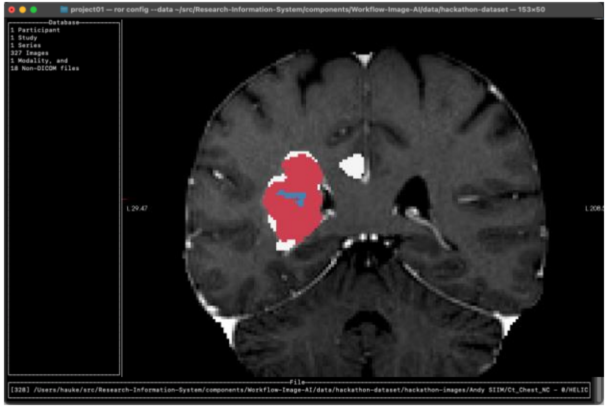
Slide by Hauke Bartsch

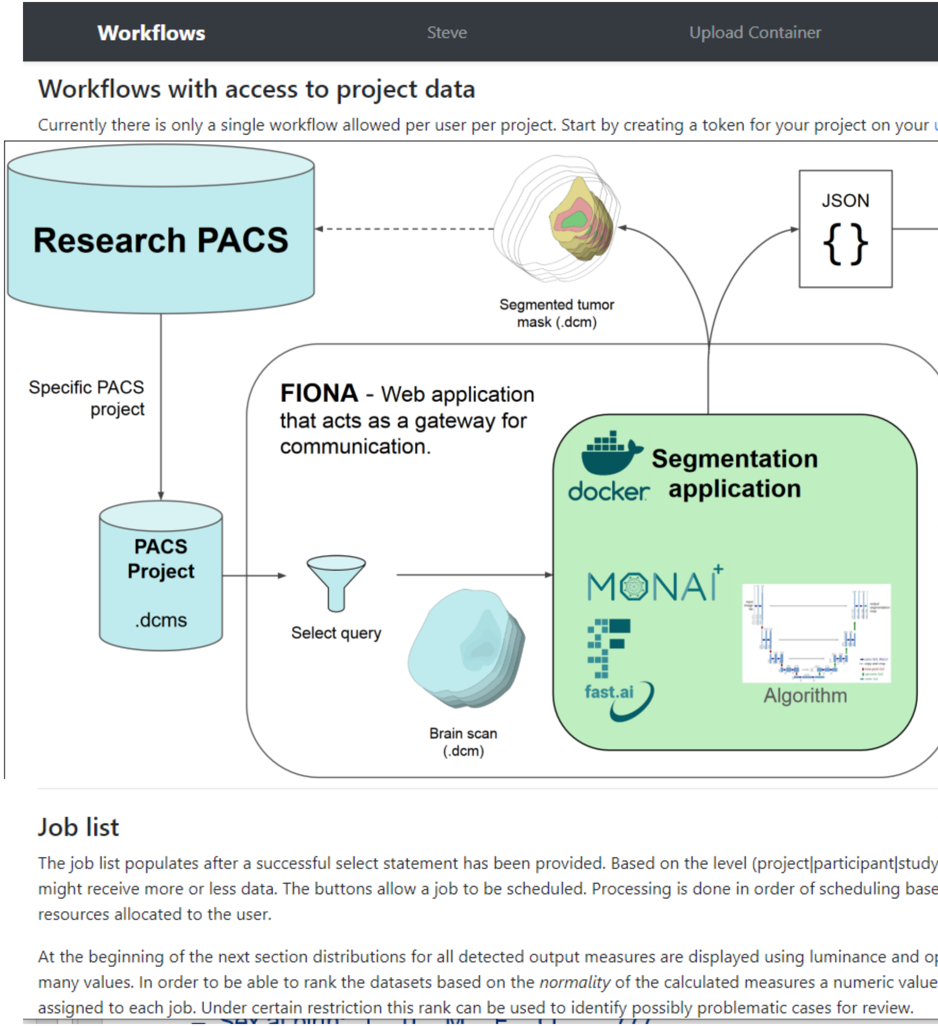
AI workflows

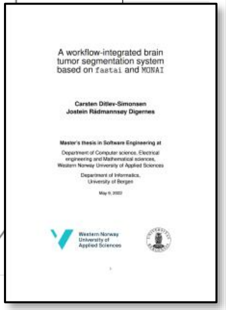
User driven activity supporting deep learning on medical images with training and prediction inside the hospital system.

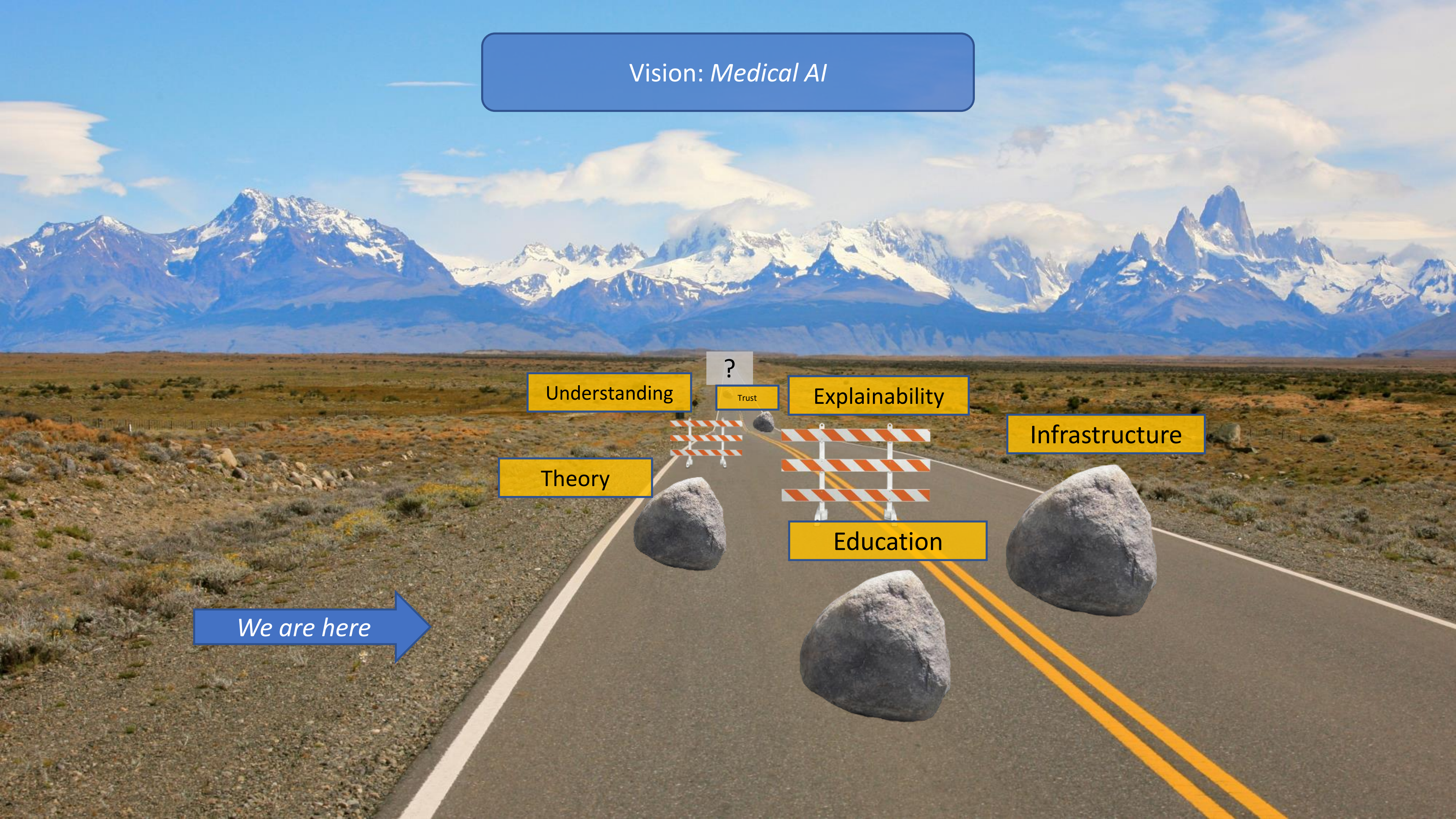
Open-source code development see:

github.com / mmiv-center / Research-Information-System / tree / master / components / Workflow-Image-AI








Vision: *Medical AI*
?
Understanding
Trust
Explainability
Infrastructure
Theory
Education
We are here

Vision: *Medical AI*

The development of Medical AI should happen with medical experts in the driver's seat *

in tight collaboration with (all the right kind of) scientists and technologists

We are here

* at least in the front seat

Vision: *Medical AI*

The development of Medical AI should happen with medical experts in the driver's seat *

in tight collaboration with (all the right kind of) scientists and technologists

We are here

* at least in the front seat

Vision: *Medical AI*

The development of Medical AI should happen with medical experts in the driver's seat *

in tight collaboration with (all the right kind of) scientists and technologists

Western Norway University of Applied Sciences
MMIV
Haukeland University Hospital
Alexander Selvikvåg Lundervold
allu@hvl.no; lundervold.net