Knowledge- and data-driven modeling for inverse problems

Ozan Öktem* Jonas Adler*†

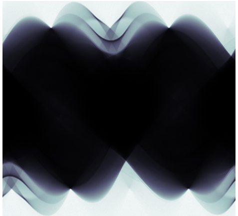
*Department of Mathematics
KTH - Royal Institute of Technology, Stockholm

†Research and Physics
Elekta

Inverse Problems

$$y = \mathcal{A}(x_{\text{true}}) + e.$$

- $y \in Y$ Data
- $x_{\text{true}} \in X$ Image
- $\mathcal{A} : X \to Y$ Forward operator
- $e \in Y$ Noise

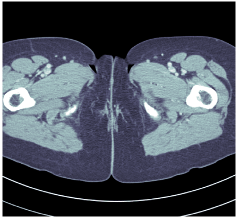


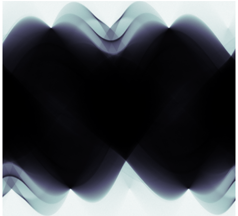
Find $\mathcal{R} : Y \to X$ such that $\mathcal{R}(y) \approx x_{\text{true}}$

Inverse Problems

$$y = \mathcal{A}(x_{\text{true}}) + e.$$

- $y \in Y$ Data
- $x_{\text{true}} \in X$ Image
- $\mathcal{A} : X \to Y$ Forward operator
- $e \in Y$ Noise



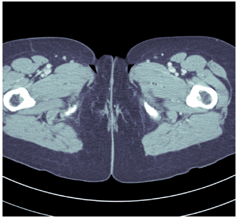


Find $\mathcal{R} : Y \to X$ such that $\mathcal{R}(y) \approx x_{\text{true}}$

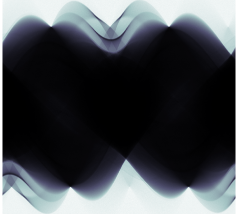
Inverse Problems

$$y = \mathcal{A}(x_{\text{true}}) + e.$$

- $y \in Y$ Data
- $x_{\text{true}} \in X$ Image
- $\mathcal{A} : X \to Y$ Forward operator
- $e \in Y$ Noise



$\xrightarrow{\mathcal{A}}$

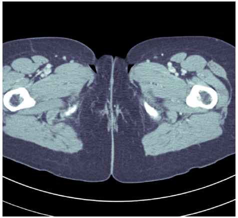
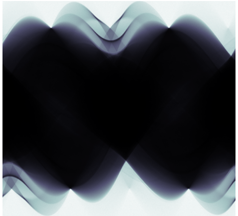


Find $\mathcal{R} : Y \to X$ such that $\mathcal{R}(y) \approx x_{\text{true}}$

Inverse Problems

$$y = \mathcal{A}(x_{\text{true}}) + e.$$

- $y \in Y$ Data
- $x_{\text{true}} \in X$ Image
- $\mathcal{A} : X \to Y$ Forward operator
- $e \in Y$ Noise

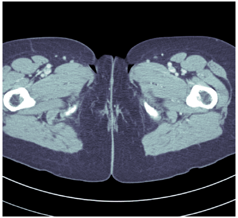
 $\xrightarrow{\mathcal{A}}$ 

Find $\mathcal{R} : Y \to X$ such that $\mathcal{R}(y) \approx x_{\text{true}}$

Inverse Problems

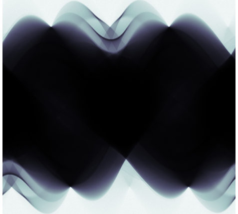
$$y = \mathcal{A}(x_{\text{true}}) + e.$$

$y \in Y$	Data
$x_{\text{true}} \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



$\xrightarrow{\mathcal{A}}$

$\xleftarrow[\mathcal{R}]{}$



Find $\mathcal{R} : Y \to X$ such that $\mathcal{R}(y) \approx x_{\text{true}}$

Modelling paradigms

Knowledge-driven: Model is prescribed beforehand using reductionistic approach, data is used to calibrate model.

+ Model design strongly guided by first principles (explicit laws and dependencies encoded by equations) that can be tested and validated independently.
+ Rather moderate data requirements.
+ Provides conceptual simplification aiding understanding.
+ Highly successful in natural and engineering sciences.

Modelling paradigms

Knowledge-driven: Model is prescribed beforehand using reductionistic approach, data is used to calibrate model.

+ Model design strongly guided by first principles (explicit laws and dependencies encoded by equations) that can be tested and validated independently.
+ Rather moderate data requirements.
+ Provides conceptual simplification aiding understanding.
+ Highly successful in natural and engineering sciences.
− Requires explicitly describing all causal relations, so less successful when first principles are unavailable or unreliable (life sciences, social and behavioral sciences, finance, . . .).
− Difficult to account for statistical properties in data (uncertainty quantification).

Modelling paradigms

Data-driven: Model is learnt from data without resorting to any first principles using assumptions about its statistical properties.

+ Requires a lot of sufficiently informative data.
+ Can capture complicated causal relations without making strong limiting assumptions.
+ Based from outset on a model for statistical properties in data.

Modelling paradigms

Data-driven: Model is learnt from data without resorting to any first principles using assumptions about its statistical properties.

+ Requires a lot of sufficiently informative data.
+ Can capture complicated causal relations without making strong limiting assumptions.
+ Based from outset on a model for statistical properties in data.
– Does not provide any conceptual simplification.
– Not easy to incorporate a priori knowledge.

Knowledge driven modelling

- Analytic pseudoinverse (FBP, FDK)
  $$x = \mathcal{A}^{\dagger}(y)$$
- Iterative methods (ART, SART)
  $$x_{i+1} = x_i - \omega \mathcal{A}^*(\mathcal{A}(x_i) - y)$$
- Variational methods (TV, TGV, Huber)
  $$x = \arg\min_x ||\mathcal{A}(x) - y||_Y^2 + \lambda ||\nabla x||_1$$

Knowledge driven modelling

- Analytic pseudoinverse (FBP, FDK)
  $$x = \mathcal{A}^\dagger(y)$$
- Iterative methods (ART, SART)
  $$x_{i+1} = x_i - \omega\mathcal{A}^*(\mathcal{A}(x_i) - y)$$
- Variational methods (TV, TGV, Huber)
  $$x = \arg\min_x ||\mathcal{A}(x) - y||_Y^2 + \lambda||\nabla x||_1$$

Knowledge driven modelling

- Analytic pseudoinverse (FBP, FDK)
  $$x = \mathcal{A}^{\dagger}(y)$$
- Iterative methods (ART, SART)
  $$x_{i+1} = x_i - \omega \mathcal{A}^*(\mathcal{A}(x_i) - y)$$
- Variational methods (TV, TGV, Huber)
  $$x = \arg\min_x ||\mathcal{A}(x) - y||_Y^2 + \lambda ||\nabla x||_1$$

What are analytic methods?

Pseudo-inverse methods are analytic (e.g.) one-pass methods. For example, if $\mathcal{A}$ is an *invertible* matrix, an analytic inverse is simply the matrix inverse.

Problem 1: For many problems an inverse *does not exist* or is not unique.

Solution 1: We can use a pseudo-inverse $\mathcal{R} = \mathcal{A}^\dagger$

Problem 2: For many problems an inverse *exists* but it is not *bounded*! Example $\mathcal{A} : \ell^2 \to \ell^2$

$$\mathcal{A}([x_1, x_2, x_3, \ldots]) = [x_1, x_2/2, x_3/3, \ldots]$$
$$\mathcal{A}^{-1}([y_1, y_2, y_3, \ldots]) = [y_1, 2y_2, 3y_3, \ldots]$$

Solution 2: We need to use regularization!

What are analytic methods?

Pseudo-inverse methods are analytic (e.g.) one-pass methods. For example, if $\mathcal{A}$ is an *invertible* matrix, an analytic inverse is simply the matrix inverse.

Problem 1: For many problems an inverse *does not exist* or is not unique.

Solution 1: We can use a pseudo-inverse $\mathcal{R} = \mathcal{A}^{\dagger}$

Problem 2: For many problems an inverse *exists* but it is not *bounded*! Example $\mathcal{A} : \ell^2 \to \ell^2$

$$\mathcal{A}([x_1, x_2, x_3, \dots]) = [x_1, x_2/2, x_3/3, \dots]$$
$$\mathcal{A}^{-1}([y_1, y_2, y_3, \dots]) = [y_1, 2y_2, 3y_3, \dots]$$

Solution 2: We need to use regularization!

What are analytic methods?

Pseudo-inverse methods are analytic (e.g.) one-pass methods. For example, if $\mathcal{A}$ is an *invertible* matrix, an analytic inverse is simply the matrix inverse.

Problem 1: For many problems an inverse *does not exist* or is not unique.

Solution 1: We can use a pseudo-inverse $\mathcal{R} = \mathcal{A}^{\dagger}$

Problem 2: For many problems an inverse *exists* but it is not *bounded*! Example $\mathcal{A} : \ell^2 \to \ell^2$

$$\mathcal{A}([x_1, x_2, x_3, \ldots]) = [x_1, x_2/2, x_3/3, \ldots]$$
$$\mathcal{A}^{-1}([y_1, y_2, y_3, \ldots]) = [y_1, 2y_2, 3y_3, \ldots]$$

Solution 2: We need to use regularization!

What are analytic methods?

Pseudo-inverse methods are analytic (e.g.) one-pass methods. For example, if $\mathcal{A}$ is an *invertible* matrix, an analytic inverse is simply the matrix inverse.

Problem 1: For many problems an inverse *does not exist* or is not unique.

Solution 1: We can use a pseudo-inverse $\mathcal{R} = \mathcal{A}^\dagger$

Problem 2: For many problems an inverse *exists* but it is not *bounded*! Example $\mathcal{A} : \ell^2 \to \ell^2$

$$\mathcal{A}([x_1, x_2, x_3, \ldots]) = [x_1, x_2/2, x_3/3, \ldots]$$
$$\mathcal{A}^{-1}([y_1, y_2, y_3, \ldots]) = [y_1, 2y_2, 3y_3, \ldots]$$

Solution 2: We need to use regularization!

What are iterative methods?

Iterative methods define a sequence of iterates

$x_1, x_2, \ldots$

Example: Landwebers method (ART, SIRT)

$$x_{i+1} = x_i - \omega \mathcal{A}^*(\mathcal{A}(x_i) - y)$$

$\mathcal{A}^*$ is the *adjoint* operator

$$\langle \mathcal{A}(x), y \rangle_Y = \langle x, \mathcal{A}^*(y) \rangle_X$$

Problem: These methods are typically not convergent

What are iterative methods?

Iterative methods define a sequence of iterates
$x_1, x_2, \ldots$

Example: Landwebers method (ART, SIRT)

$$x_{i+1} = x_i - \omega \mathcal{A}^*(\mathcal{A}(x_i) - y)$$

$\mathcal{A}^*$ is the *adjoint* operator

$$\langle \mathcal{A}(x), y \rangle_Y = \langle x, \mathcal{A}^*(y) \rangle_X$$

Problem: These methods are typically not convergent

What are iterative methods?

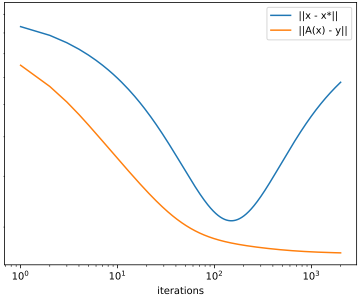
Iterative methods define a sequence of iterates $x_1, x_2, \ldots$

Example: Landwebers method (ART, SIRT)

$$x_{i+1} = x_i - \omega \mathcal{A}^*(\mathcal{A}(x_i) - y)$$

$\mathcal{A}^*$ is the *adjoint* operator

$$\langle \mathcal{A}(x), y \rangle_Y = \langle x, \mathcal{A}^*(y) \rangle_X$$



Problem: These methods are typically not convergent

What are variational methods?

Variational methods are methods based on the calculus of variations. They are formulated as minimization problems, for inverse problems typically of the form

$$\min_f \left[D(f;g) + R(f)\right]$$

with

- given **data** g according to a forward model,
- **data discrepancy** functional $D(\cdot;g)$,
- **regularization** functional R

Typical example: Total Variation (TV) regularization with L^2 data term:

$$\min_f \left[\|Af - g\|_2^2 + \alpha\|\nabla f\|_1\right]$$

The solution of variational problems usually requires an optimization method (smooth or non-smooth, convex or non-convex).

What are variational methods?

Variational methods are methods based on the calculus of variations. They are formulated as minimization problems, for inverse problems typically of the form

$$\min_f \left[D(f; g) + R(f)\right]$$

with

- given **data** g according to a forward model,
- **data discrepancy** functional $D(\cdot; g)$,
- **regularization** functional R

Typical example: Total Variation (TV) regularization with L^2 data term:

$$\min_f \left[\|Af - g\|_2^2 + \alpha\|\nabla f\|_1\right]$$

The solution of variational problems usually requires an optimization method (smooth or non-smooth, convex or non-convex).

Optimization example: linearized ADMM

Example: The Alternating Direction Method of Multipliers (ADMM) is a popular method for the solution of problems

$$\min_f \left[F(f) + G(Lf)\right]$$

with convex functionals F and G and a linear operator L. The linearized variant uses the following iteration (τ and σ are parameters, $z^{(0)} = u^{(0)} = 0$):

$$f^{(k+1)} = \mathrm{prox}_{\tau F}\left[f^{(k)} - \frac{\tau}{\sigma} L^* \left(Lf^{(k)} + u^{(k)} - z^{(k)}\right)\right]$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma G}\left[Lf^{(k+1)} + u^{(k)}\right]$$

$$u^{(k+1)} = Lf^{(k+1)} + u^{(k)} - z^{(k+1)}$$

Here, prox is the proximal operator of a functional,

$$\mathrm{prox}_{\tau F}(f) = \arg\min_v \left[\tau F(v) + \frac{1}{2}\|f - v\|_2^2\right].$$

Optimization example: linearized ADMM

Example: The Alternating Direction Method of Multipliers (ADMM) is a popular method for the solution of problems

$$\min_f \left[F(f) + G(Lf)\right]$$

with convex functionals F and G and a linear operator L. The linearized variant uses the following iteration (τ and σ are parameters, $z^{(0)} = u^{(0)} = 0$):

$$f^{(k+1)} = \mathrm{prox}_{\tau F}\left[f^{(k)} - \frac{\tau}{\sigma} L^* \left(Lf^{(k)} + u^{(k)} - z^{(k)}\right)\right]$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma G}\left[Lf^{(k+1)} + u^{(k)}\right]$$

$$u^{(k+1)} = Lf^{(k+1)} + u^{(k)} - z^{(k+1)}$$

Here, prox is the proximal operator of a functional,

$$\mathrm{prox}_{\tau F}(f) = \arg\min_v \left[\tau F(v) + \frac{1}{2}\|f - v\|_2^2\right].$$

Implementation of linearized ADMM

```
def admm_linearized(f, F, G, L, tau, sigma, niter):
    z = u = L.range.zero()
    for i in range(niter):
        f[:] = F.proximal(tau)(f - tau / sigma * L.adjoint(L(f) + u - z))
        z = G.proximal(sigma)(L(f) + u)
        u = L(f) + u - z
```

$$z^{(0)}) = u^{(0)} = 0$$

$$f^{(k+1)} = \mathrm{prox}_{\tau F}\left[f^{(k)} - \frac{\tau}{\sigma} L^* \left(Lf^{(k)} + u^{(k)} - z^{(k)}\right)\right]$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma G}\left[Lf^{(k+1)} + u^{(k)}\right]$$

$$u^{(k+1)} = Lf^{(k+1)} + u^{(k)} - z^{(k+1)}$$

Implementation of linearized ADMM

```
def admm_linearized(f, F, G, L, tau, sigma, niter):
    z = u = L.range.zero()
    for i in range(niter):
        f[:] = F.proximal(tau)(f - tau / sigma * L.adjoint(L(f) + u - z))
        z = G.proximal(sigma)(L(f) + u)
        u = L(f) + u - z
```

Notable:

- Implementation uses **ODL**, a Python framework for inverse problems
- `L` is an `Operator` that can be queried for its `domain` and `range` (actually codomain).
- `F` and `G` are `Functionals` that can be queried for their `proximal` operators.
- `L.adjoint` is again an `Operator`.

Limits to knowledge driven modeling

Task: Does the image contain a rabbit?

Limits to knowledge driven modeling

Task: Does the image contain a rabbit?

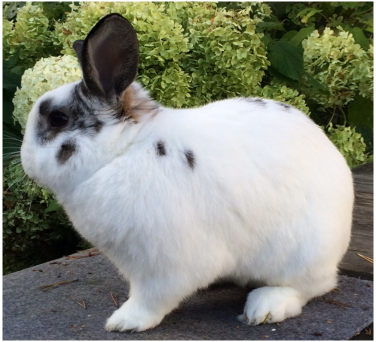
Knowledge driven modelling:

Connected structure consisting of fused ellipsoids, two specific elongated structures near each other (ears), fibre-like texture (fur) with specific range of colours, . . . then it is a rabbit!

Limits to knowledge driven modeling

Task: Does the image contain a rabbit?

Knowledge driven modelling:
Connected structure consisting of fused ellipsoids, two specific elongated structures near each other (ears), fibre-like texture (fur) with specific range of colours, . . . then it is a rabbit!

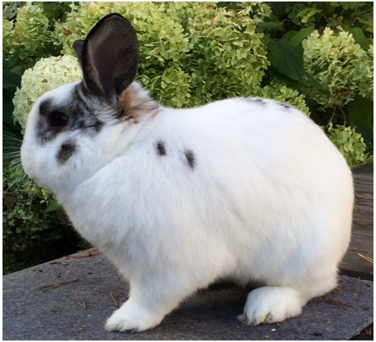


Limits to knowledge driven modeling

Task: Does the image contain a rabbit?

Knowledge driven modelling:

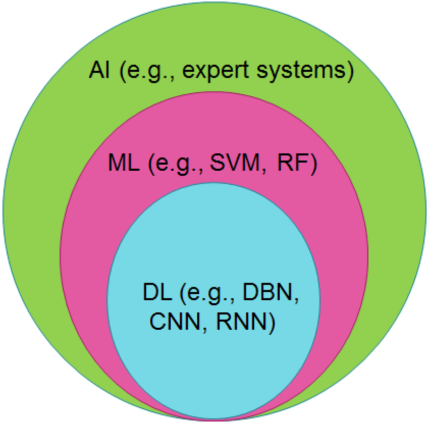
Connected structure consisting of fused ellipsoids, two specific elongated structures near each other (ears), fibre-like texture (fur) with specific range of colours, . . . then it is a rabbit!



Data-driven modeling

Machine learning: Terminology

- Artificial Intelligence (AI): Any technique that enables computers to mimic human intelligence
- Machine learning (ML): A subset of AI using statistical techniques to enable machines to learn from data
- Deep Learning (DL): A subset of ML where complicated tasks are performed by breaking them down into several layers



- Early history (1950–1990)
  - First artificial neural network computational machines in 1954
  - The perceptron for pattern recognition introduced in 1958.
  - First functional networks with many layers introduced in 1965.

  Stagnation: Basic perceptrons cannot process the exclusive-or circuit (1969) and computers not powerful enough.
- Deep convolutional neural networks (2000)

  Series of papers in early 2000s showing state-of-the-art results for many tasks.

  Key aspects
  - Many network layers & huge amount of training data.
  - Massively paralleled GPU based implementations.
  - Optimisation algorithms (effective initializations & stochastic gradient descent).

- Applications: Began with written digit recognition (classification), moving towards tasks associated with vision, speech recognition & natural language processing.
    - Denoising: BM2D performed better than any known image denoising algorithm in 2012.
    - Single image super-resolution
    - Demosaicing
    - Deblurring
    - Segmentation
    - Image annotation
    - Face recognition
    - ...

- Applications: Began with written digit recognition (classification), moving towards tasks associated with vision, speech recognition & natural language processing.
  - Denoising: BM2D performed better than any known image denoising algorithm in 2012.
  - Single image super-resolution
  - Demosaicing
  - Deblurring
  - Segmentation
  - Image annotation
  - Face recognition
  - ...

Deep learning currently addresses many image processing tasks with unsurpassed results.

Data-driven modeling

Machine learning: In the news

Forbes

Tech

FEB 19, 2015 @ 01:06 PM 11,281 VIEWS

Microsoft's Deep Learning Project Outperforms Humans In Image Recognition

MIT Technology Review

Deep Learning Machine Beats Humans in IQ Test

Computers have never been good at answering the type of verbal reasoning questions found in IQ tests. Now a deep learning machine unveiled in China is changing that.

June 12, 2015

Bloomberg Technology

Google's DeepMind Achieves Speech-Generation Breakthrough

by Jeremy Kahn

September 9, 2016 — 7:29 AM EDT

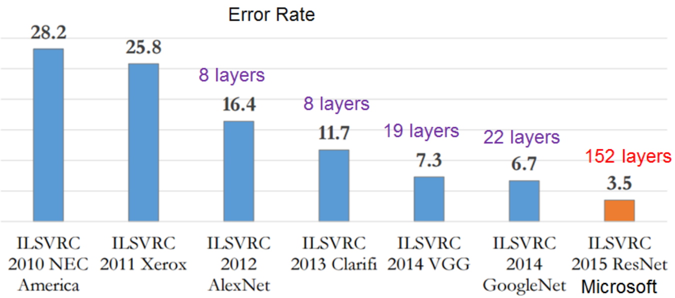
Microsoft

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

October 18, 2016

Data-driven modeling

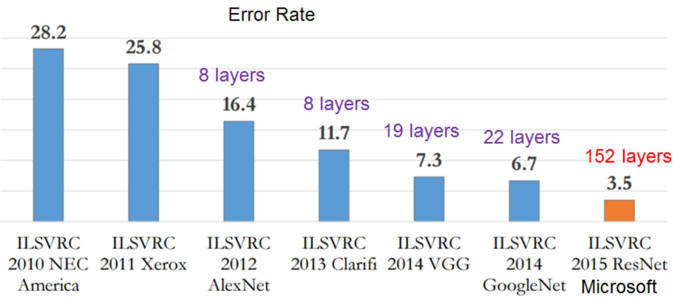
Machine learning: In the news



Human Performance ~ 5% error

Data-driven modeling

Machine learning: In the news

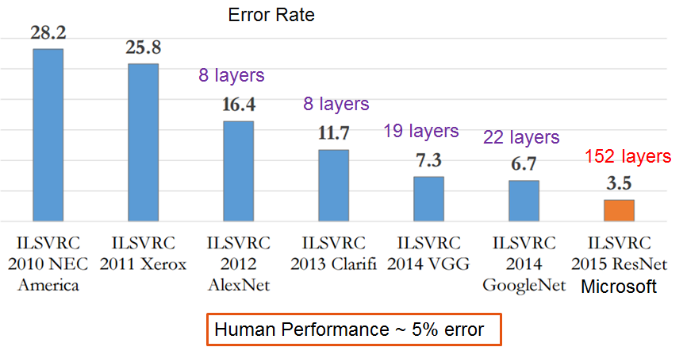


Human Performance ~ 5% error

Achievements obtained without theoretical understanding:

- *Optimisation solved during learning is highly non-convex and seems intractable from a theoretical viewpoint.*
- *Unclear what characterises good training data.*
- *Stability properties of network architectures (adversarial attacks).*

Data-driven modeling
Machine learning: In the news

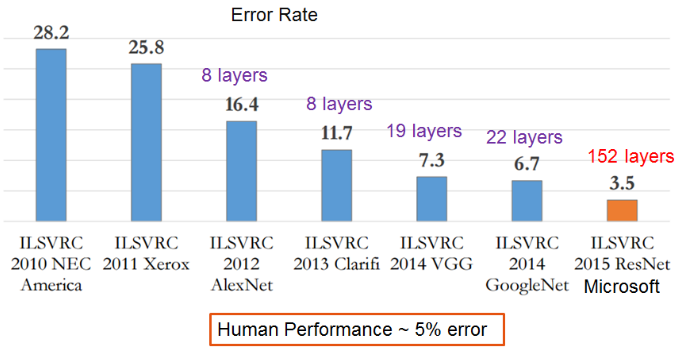


Success stories of deep learning this far confined to tasks that do not require knowledge about how data is generated.

> *Entirely data-driven tomographic image reconstruction is unfeasible.*
> *As an example, a typical 3D CB-CT problem* ($512 \times 512 \times 512$ *voxel) requires a network with at least* $\approx 2 \cdot 10^{16}$ *connections!!*

Data-driven modeling

Machine learning: In the news



IMAGENET

Hybrid: Combine elements of knowledge- and data-driven approaches for modelling.

Can a hybrid approach be useful for solving ill-posed inverse problems?
In particular, can one incorporate the knowledge of a forward operator when designing a neural network for reconstruction?

Thank you for your attention!