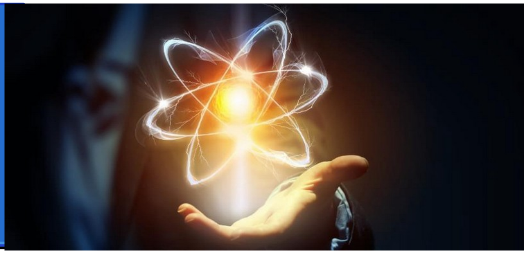
Jakovác Antal

Dept. of Computational Sciences

Representation learning in (artificial) intelligence

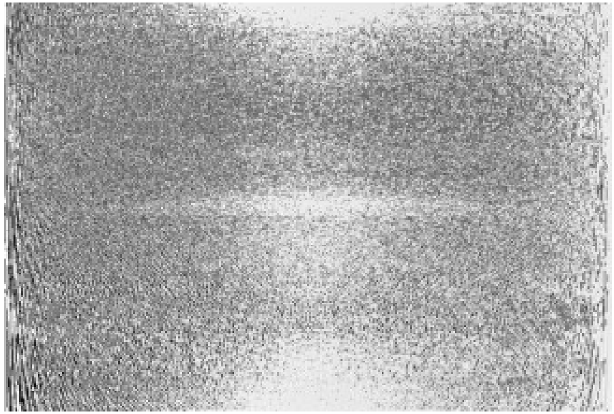
GPU day 2022, June 20-21, 2022.

Motivation: representation matters!



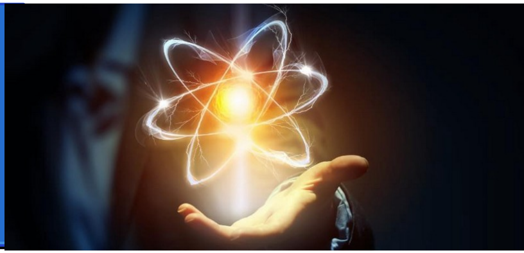
What is in the images?

First seems to be noise … although it is just a transformed variant of the second!



Human visual system uses a recognition function class that relies on the specific properties of the natural images (eg. solid bodies, forms).

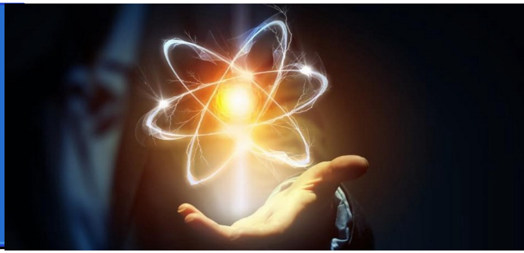
Motivation: representation matters!



No “general learning machine”

- $I = \{ N \times M \text{ color images } \}$, for 1Mpx images $|I| \approx 10^{7000000}$
- a class can be any subset: number of subsets $2^{|I|} \approx 10^{10^{7000000}}$
- information in 1Pbyte $\approx 10^{10^{15}}$
- we can describe only a vast minority of all possible classes
- **for success we must exploit the specific properties of the observed class!**
  e.g. in images: important details are slowly changing, shapes, textures, translation and scale invariance
  → included in the Convolutional Neural Network (CNN) architecture

Motivation: representation matters!



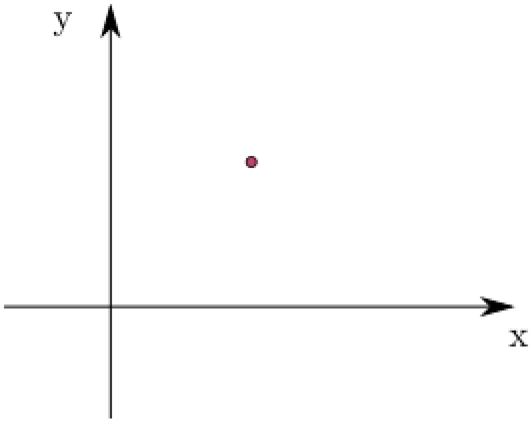
Difference between understanding and training:

- neural network $f(x, \alpha) = y$ maps input to output using a parametrizable function class
- **training**: in a given function class we refine the parametrization to fit to the external requirements (*supervision*)
- **understanding**: find the function class that best fits to the set of the inputs (*unsupervised, data-driven*)
- understanding should precede training! (*representation learning*)

Examples of data modeling

The most elementary, but generic task is to tell if an item is element of a set.

Continuous examples: single 2D data point: S={p} one element set.



We can represent it with the (x,y) coordinates.

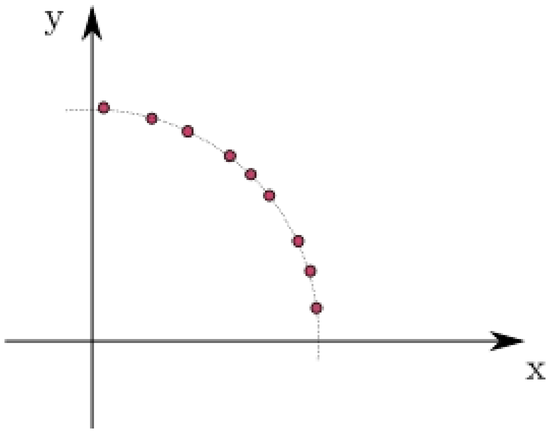
Other representations are also appropriate.

For a single data all representations are equivalent.

Examples of data modeling

The most elementary, but generic task is to tell if an item is element of a set.

Continuous examples: multiple 2D data points



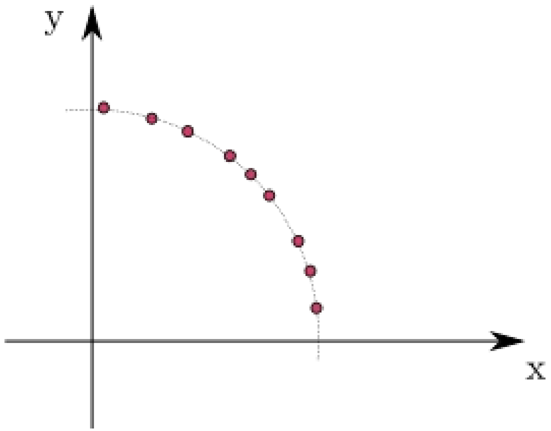
In the (x,y) representation the coordinates are not independent.

In the polar coordinate system (r,φ) we find r=R for all data points! The r and φ coordinates are independent.

Examples of data modeling

The most elementary, but generic task is to tell if an item is element of a set.

Continuous examples: multiple 2D data points



In the (x,y) representation the coordinates are not independent.

In the polar coordinate system (r,φ) we find r=R for all data points! The r and φ coordinates are independent.

In a well-chosen coordinate system the data coordinates are independent, and they are either constant (**relevant** or **selective** coordinates, or **laws**),
or variable (**irrelevant** or **descriptive** coordinates).

Coordination and understanding

If we understand a system well, elementary training is trivial!

Features: independent coordinates over C, either selective or descriptive

Let ξ be the common features for $C_{1,} C_{2,} \ldots, C_a, C = \cup_i C_i$

- **classification:** $x \in C_i$ iff selective bits of $\xi(x) =$ selective bits of C_i
- **decoding:** to produce $x \in C_i$ we have to chose the relevant bits characteristic to C_i and the irrelevant bits independently, uniform randomly

$$\xi^{-1}(\sigma_{relevant} = C_{i,relevant}, \sigma_{irrelevant} = \mathrm{random}) \in C_i$$

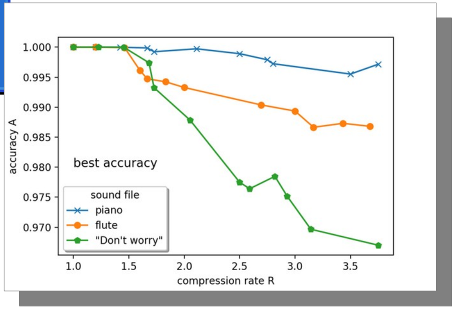
- **lossless data compression:** if we know that $x \in C_i$, the relevant bits can be built into the static part of the code, and we have to store the *irrelevant bits*.

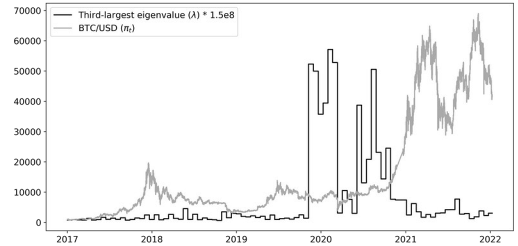
All the AI tasks can be solved by inspecting certain bits.

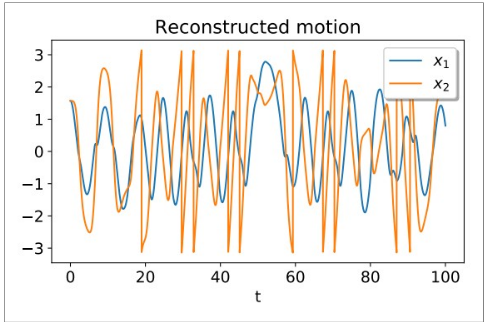
Publications in the topic

Using this technique we studied some topics:

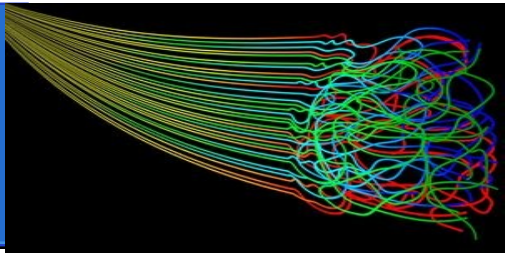
- [D.Berenyi, AJ, P. Pósfay, 2020]: paper about the theoretical basics
- [AJ, 2021]: treating linear laws, application for musical data compression
- [TS. Biró, AJ, 2022] : entropy associated to representations
- [M. Kurbucz, P. Pósfay, AJ, 2022] using linear laws we examined Bitcoin prices and identified potential external influence
- [M. Kurbucz, P. Pósfay, AJ, 2022]: reconstruction of mechanical motions using nonlinear laws
- … more in preparation







Entropy of the intelligence



Intelligence or understanding is the choice of correct representation.

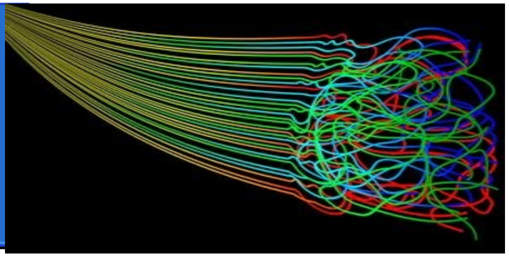
Is there a universal measure to decide, how good a given representation is?

entropy of a representation with respect to a subset

- **Shannon entropy:** $$S_{SH} = \sum_{\sigma \in B^N} p_C(\xi = \sigma) \log_2 p_C(\xi = \sigma) = \log_2 |C|$$
  - independent of the representation
  - yields the true information content of the set (i.e. the number of necessary bits)
- **representation entropy:** ξ coordination implies $p_C(\xi_i = \sigma_i)$ **bitwise distribution**

$$S_{repr} = \sum_{i=1}^{N} \left[\sum_{\sigma \in 0,1} p_C(\xi_i = \sigma_i) \log_2 p_C(\xi_i = \sigma) \right]$$

Entropy of the intelligence



Representation entropy

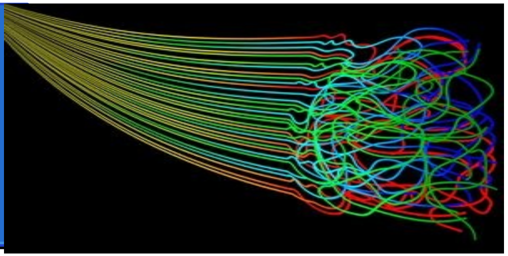
$$S_{repr} = \sum_{i=1}^{N} [\sum_{\sigma \in 0,1} p_C(\xi_i = \sigma_i) \log_2 p_C(\xi_i = \sigma)]$$

Mathematical properties

- $S_{repr} \geq S_{SH}$, equality if the coordination is independent
- minimality of S_{repr} implies independence, and the least # of descriptive coordinates
- $Loss = S_{repr} + \lambda \alpha + \mu \beta$ can be used in practice, with type one and two errors (false negative and false positive)

representation entropy is a general unsupervised loss function:
in a general learning process, by minimizing the representation entropy, we get closer to the learning of the proper representation

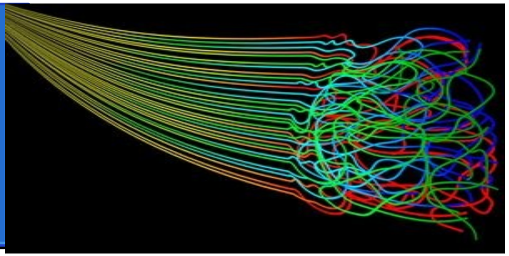
Reconstruction of mechanical motions



Task:

- observe a motion $\{x_n \in \mathbb{R}^D \mid n \in \{0,\ldots,N\}\}$
  - ‘n’ is a (discrete) time variable for $t = n\Delta$, maximal observed time $T = N\Delta t$
  - D dimensional motion
- describe/characterize the motion
- continue for t>T in a ”plausible” way

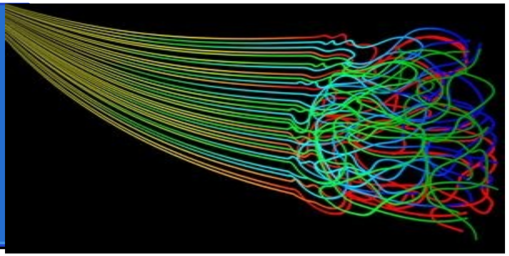
Reconstruction of mechanical motions



Method:

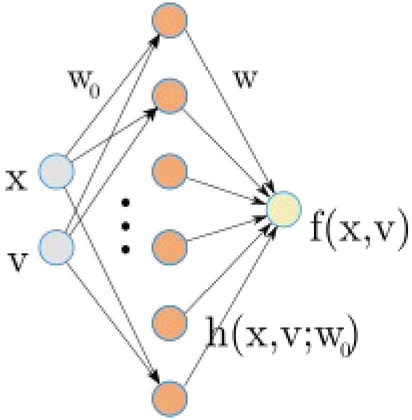
- local characterization of the motion: $v_n = \frac{x_n - x_{n-1}}{\Delta t}, \quad a_n = \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2}$
- look for different level “laws”/constraints:
  - level 0, holonomic contraints: $C^{(0)}(x_n) = C^{(0)}(x_0)$
  - level 1: anholonomic constraints, conserved quantities: $C^{(1)}(x_n, v_n) = C^{(1)}(x_0, v_0)$
  - level 2: lows for acceleration / discrete Newton’s laws: $a_n = f(x_n, v_n)$
- In a consistent mechanical system Newton’s laws are compatible with lowest order constraints, but in numerical observations they are independent.
- from discrete Newton’s laws a recursion can be obtained $(x_{n-1}, x_n) \Rightarrow x_{n+1}$

Reconstruction of mechanical motions



Numerical implementation:

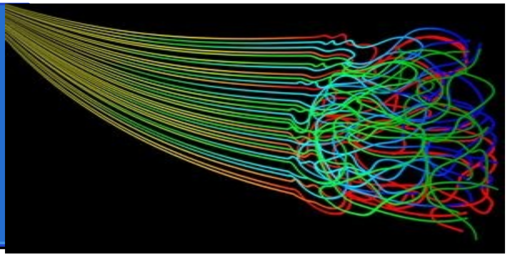
- input: (x_n, v_n) 2D dimensional
- output: $C^{(1)}(x_n, v_n)$ conserved quantity or $f(x_n, v_n)$ force function
- network: Extreme Learning Machine, 1 hidden layer, only last weights are trained, nonlinear activation function ensures smoothness of output



Issues:

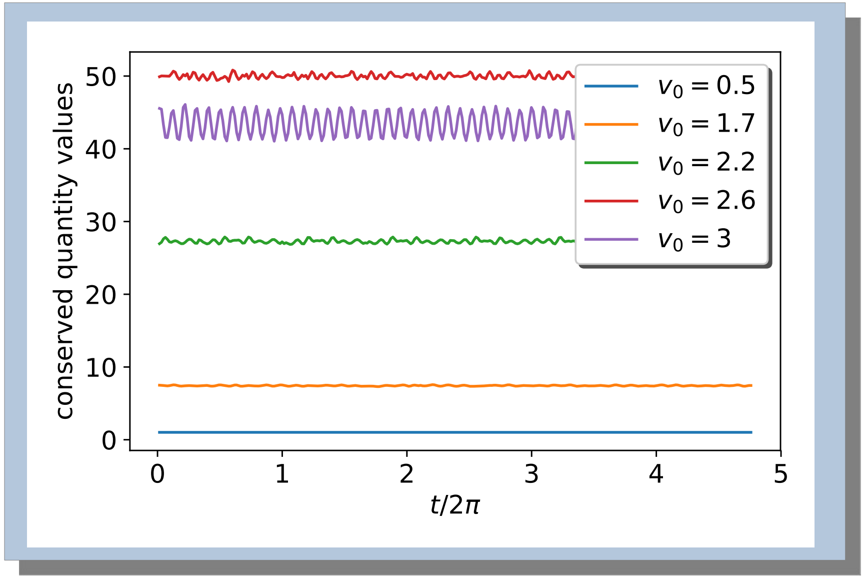
- chaoticity: if nearby motions diverge fast, even “exact” methods give different results. Comparison: force and qualitative features
- renormalization: recursion can be determined for different Δt, multi-step algorithms are possible.

Reconstruction of mechanical motions

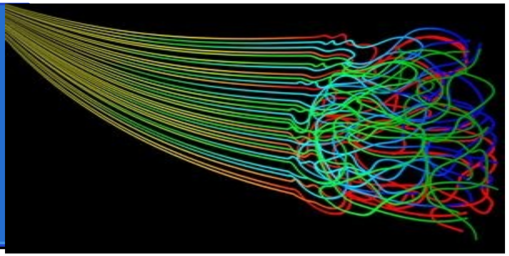


Results:

- gravity pendulum: integrable motion $\ddot{x} = -\sin x$

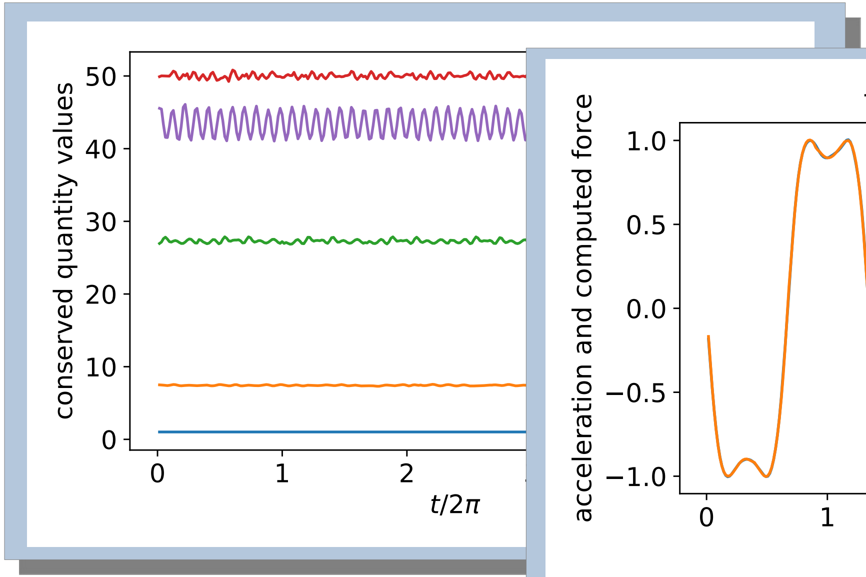


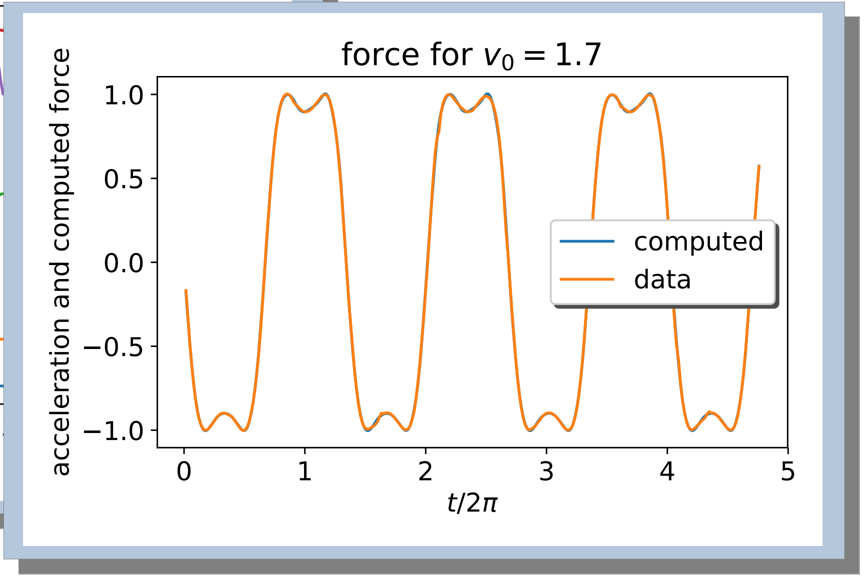
Reconstruction of mechanical motions



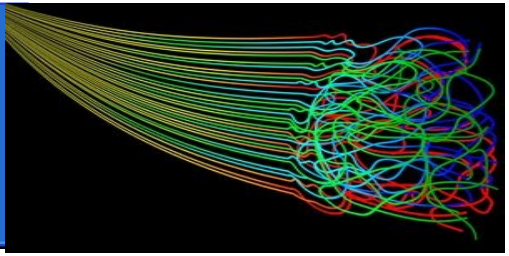
Results:

- gravity pendulum: integrable motion $\ddot{x} = -\sin x$



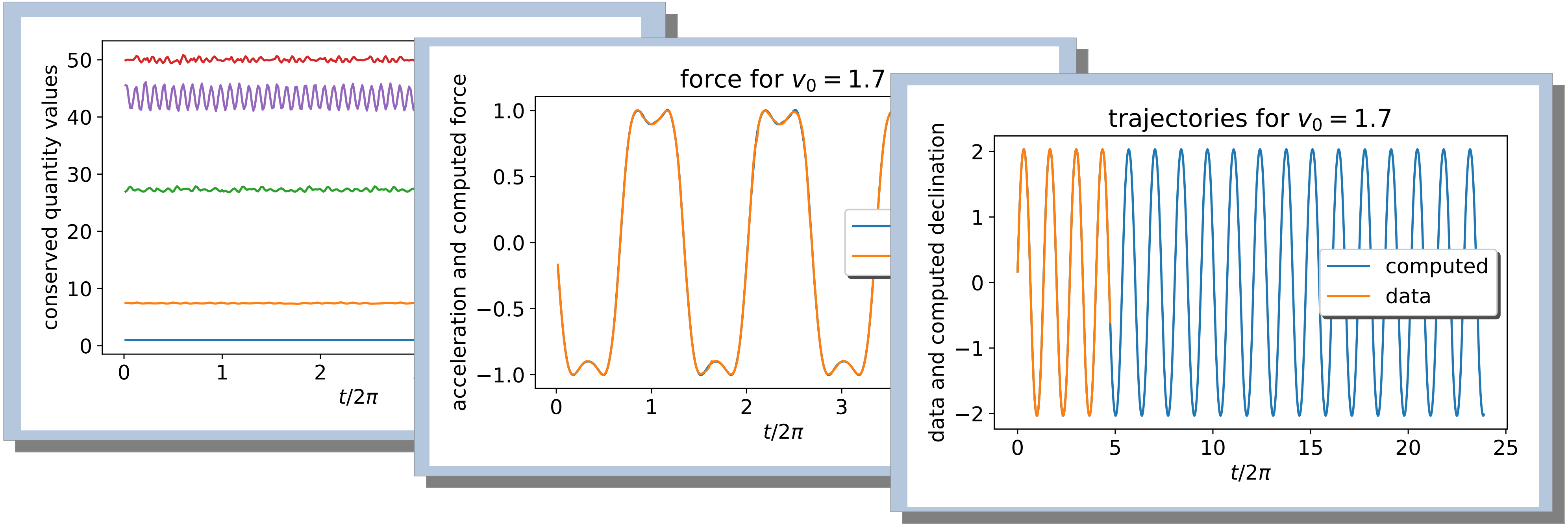


Reconstruction of mechanical motions

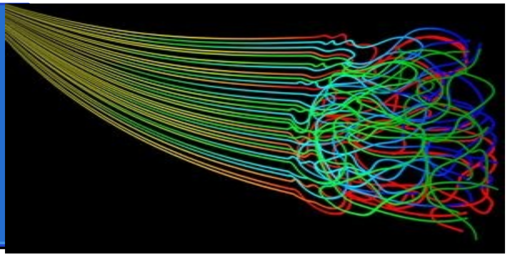


Results:

- gravity pendulum: integrable motion $\ddot{x} = -\sin x$

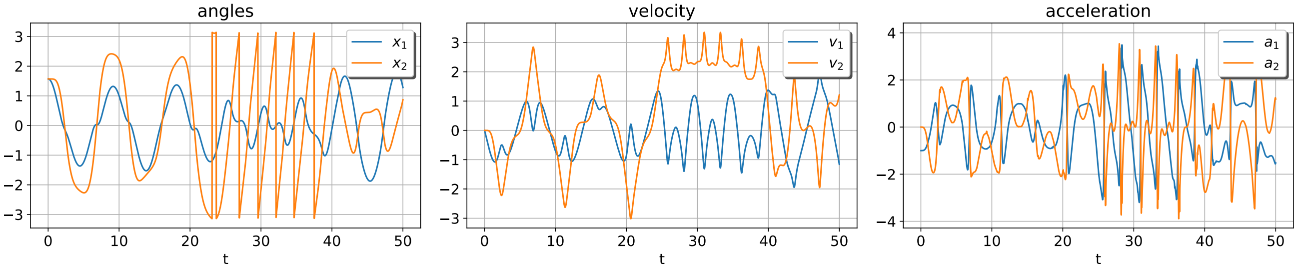


Reconstruction of mechanical motions

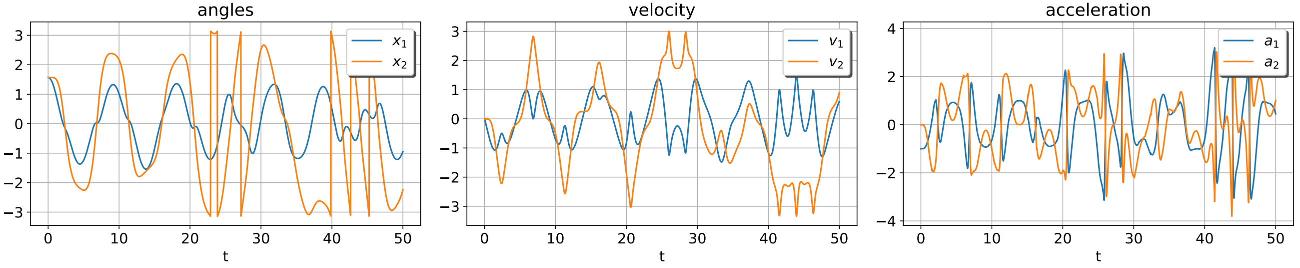


Results:

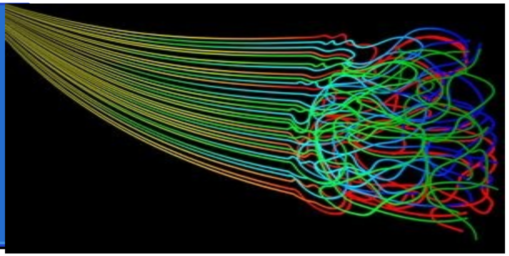
- double pendulum: 2D chaotic motion
- “exact solution” can not be found
  - Python scipy DOP853



  - Python scipy RK45

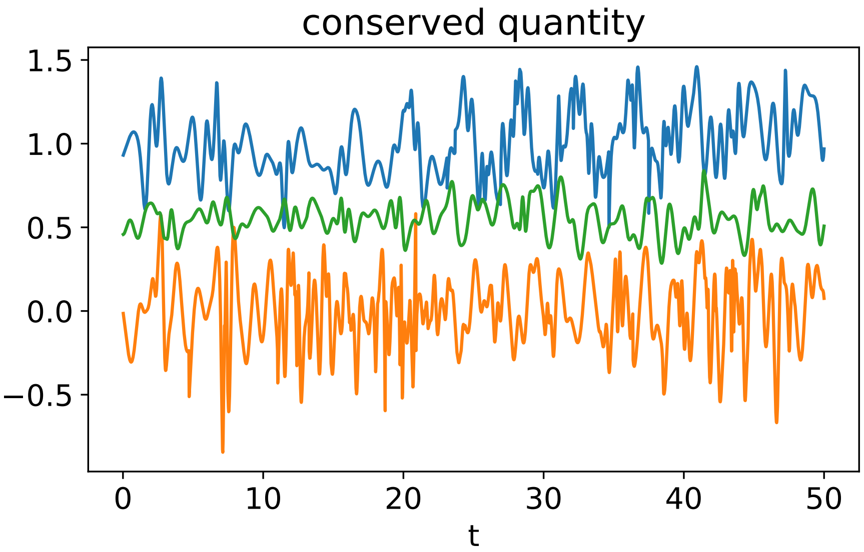


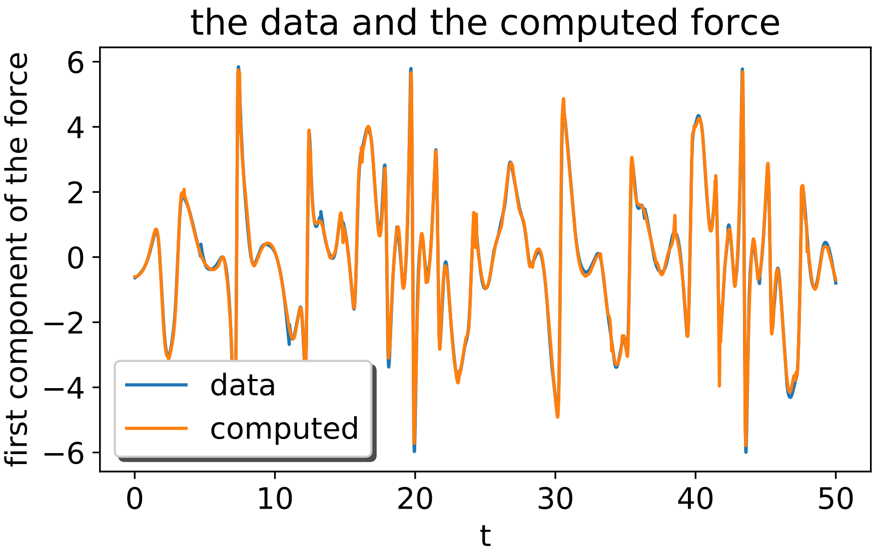
Reconstruction of mechanical motions



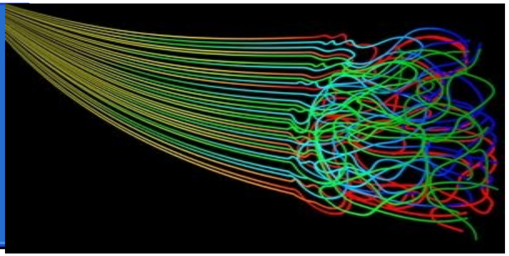
Results:

- double pendulum: 2D chaotic motion



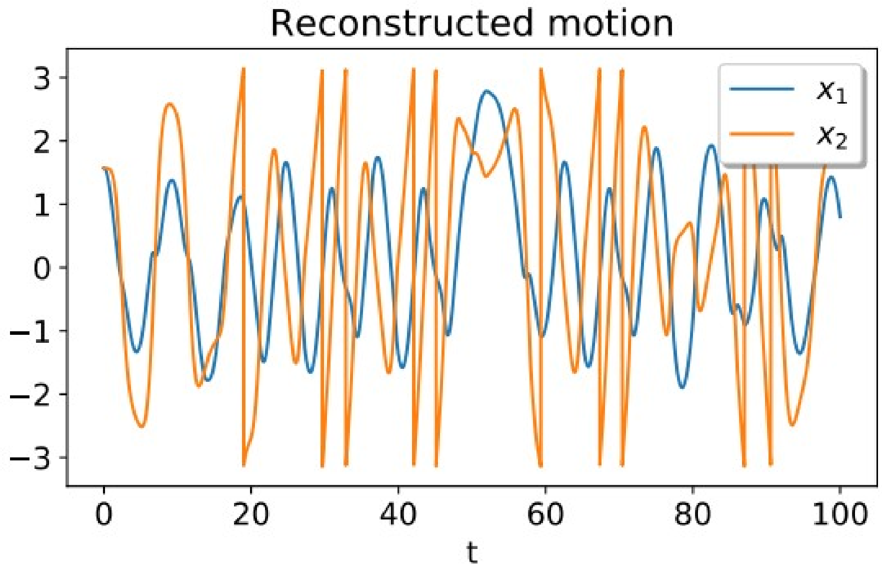


Reconstruction of mechanical motions



Results:

- double pendulum: 2D chaotic motion reconstructed force with 93% accuracy
- motion qualitatively correct, no runaway solutions

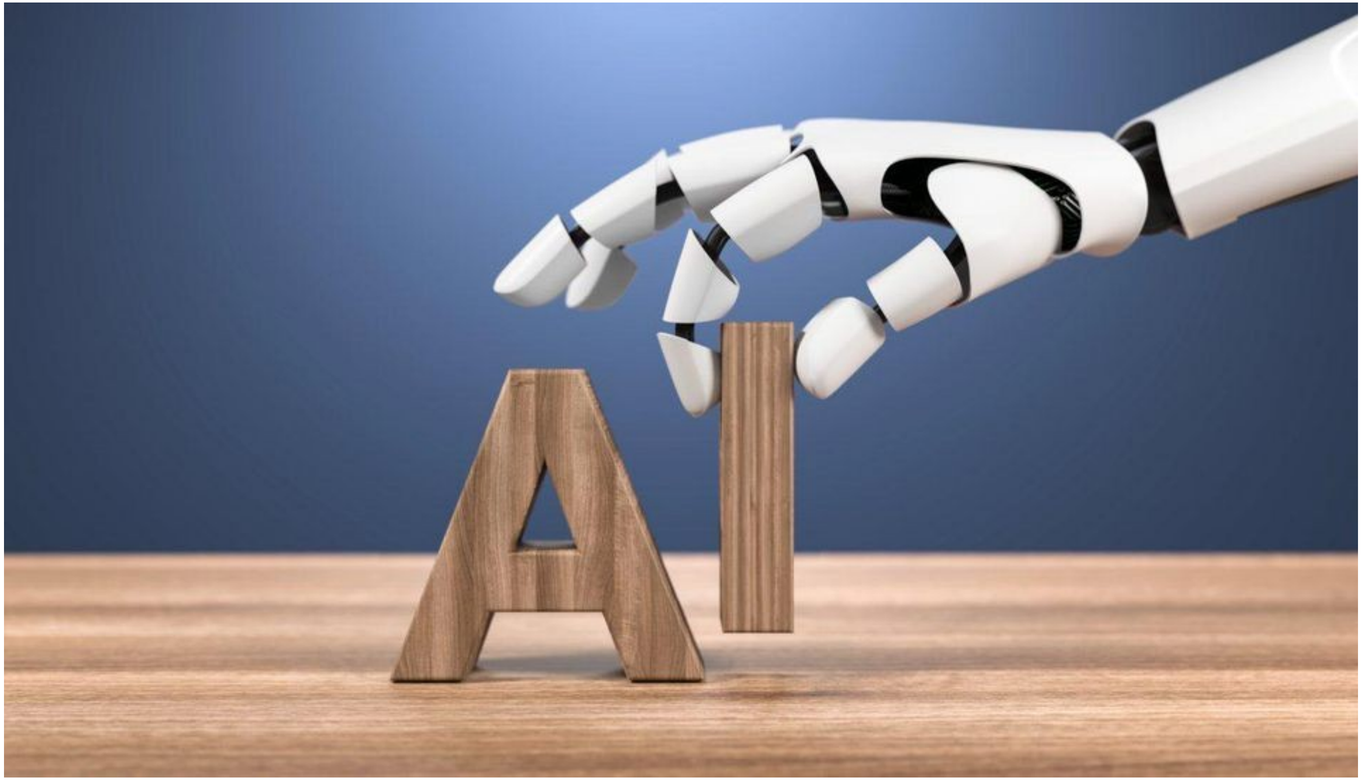


Conclusions

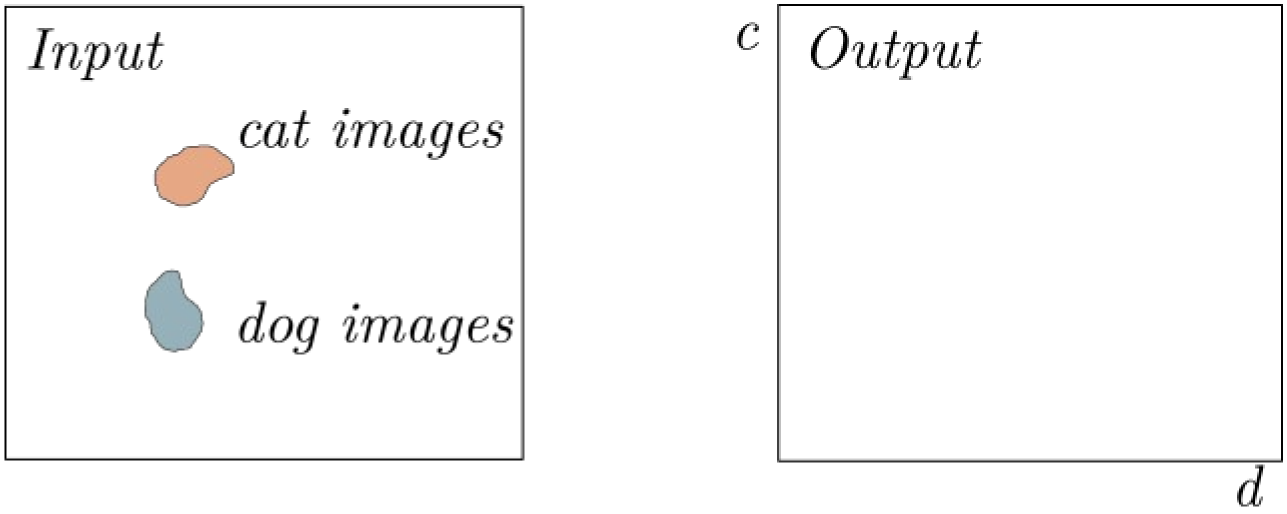
understanding ≡ best representation of data

- independent features (coordinates) over a set C: either selective or descriptive
- selective/relevant features: constant over C, good for classification
- descriptive/irrelevant features: variable over C, good for compression
- representation entropy: universal unsupervised loss function, by minimizing it we improve understanding
- in mechanical systems laws ≡ conserved quantities & Newton’s law
  - good reconstruction for integrable systems
  - qualitatively correct reconstruction for chaotic motions

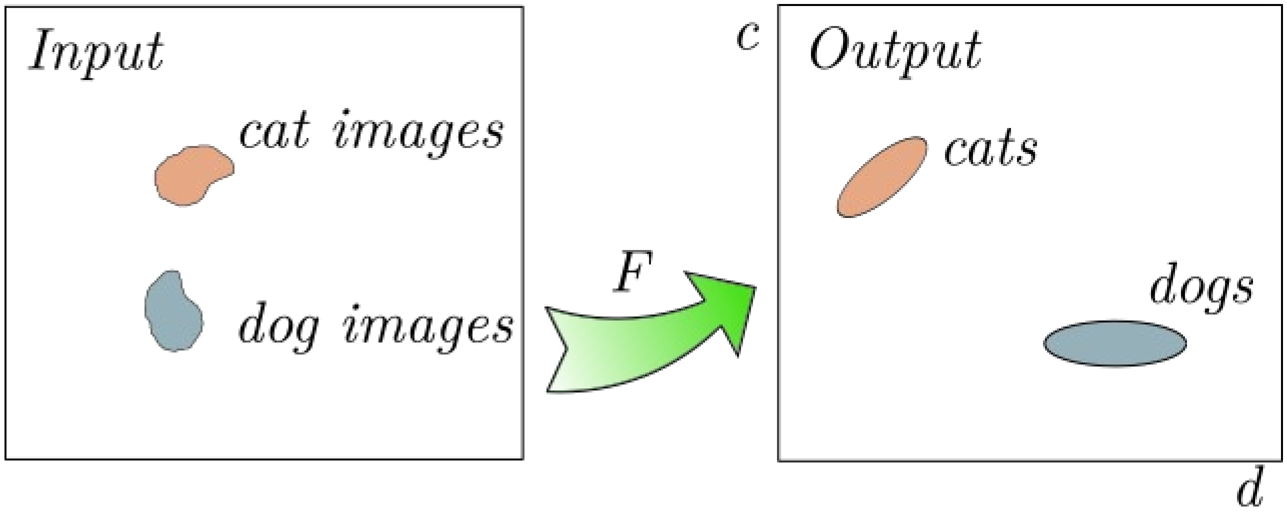
The end



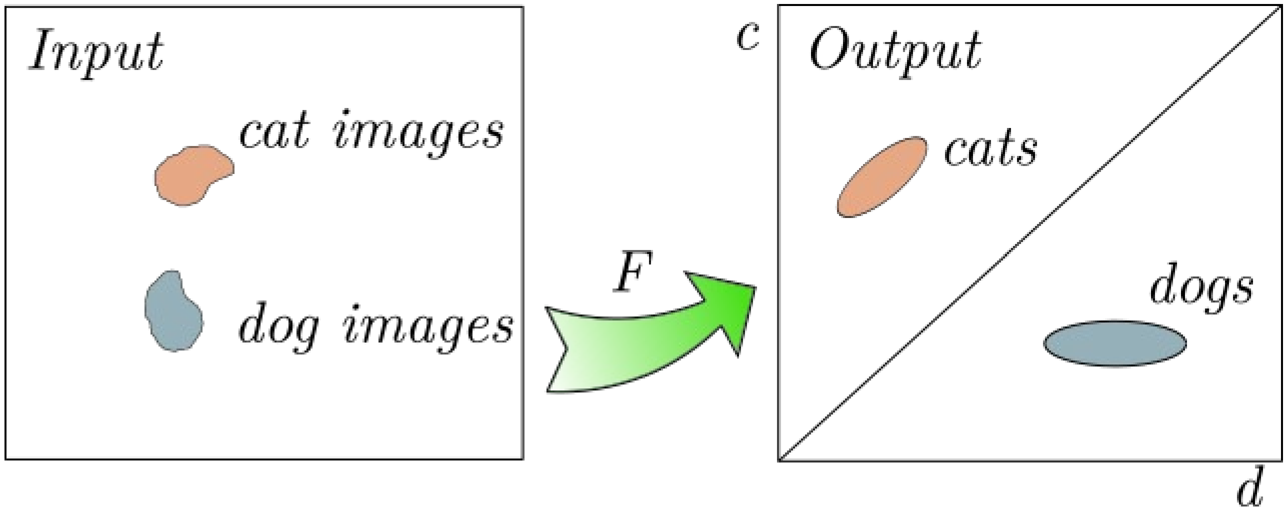
Interpretation of AI



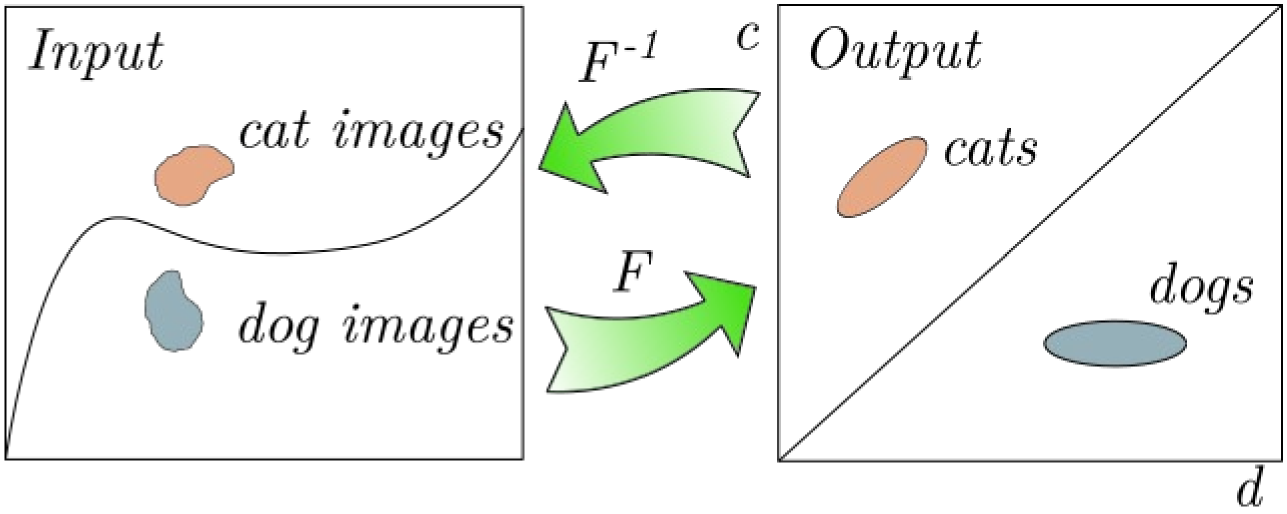
Interpretation of AI



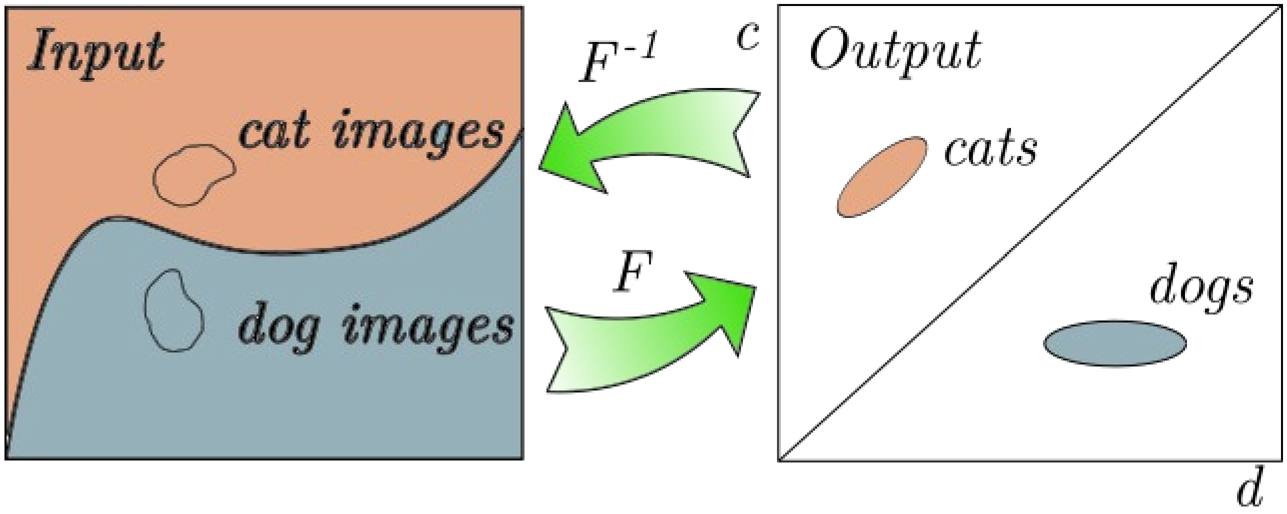
Interpretation of AI



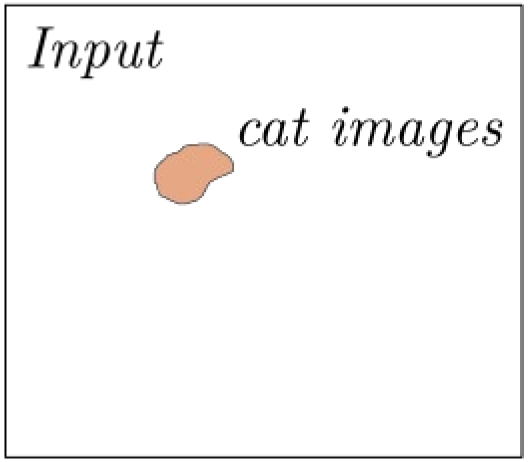
Interpretation of AI

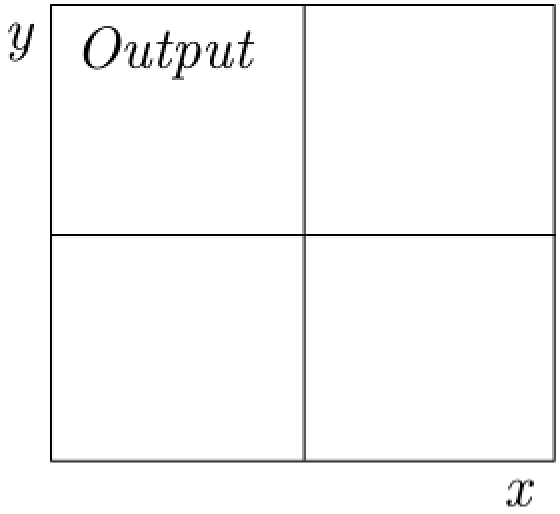


Interpretation of AI

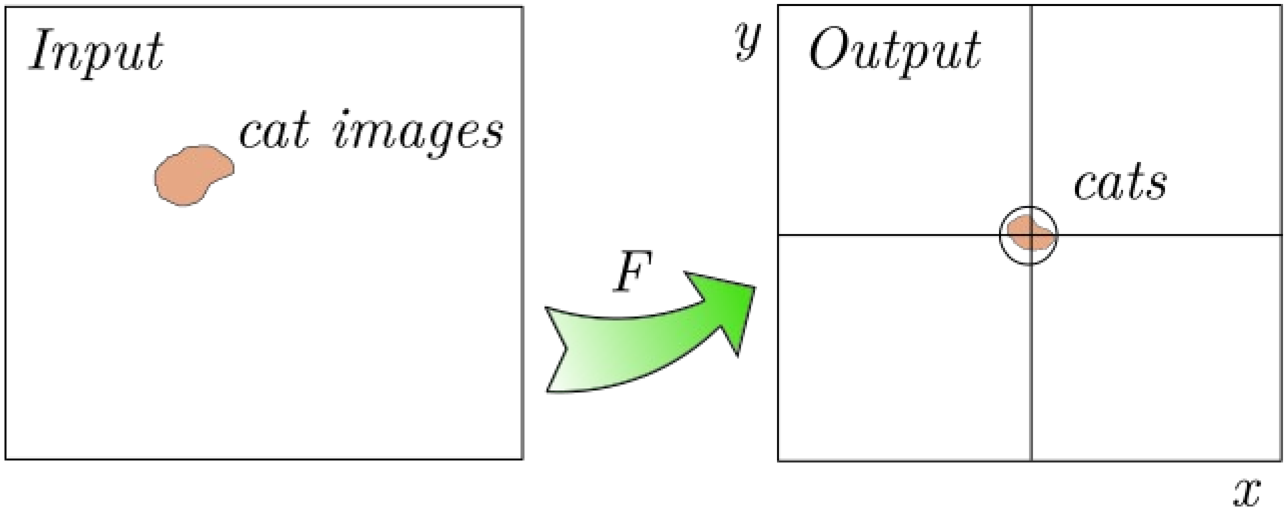


Interpretation of AI

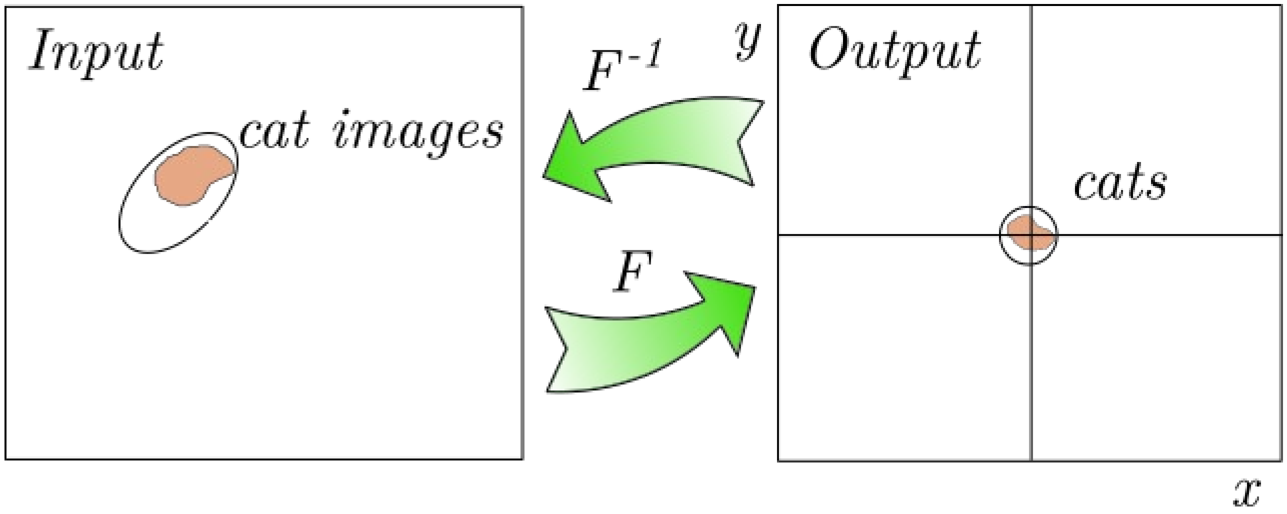




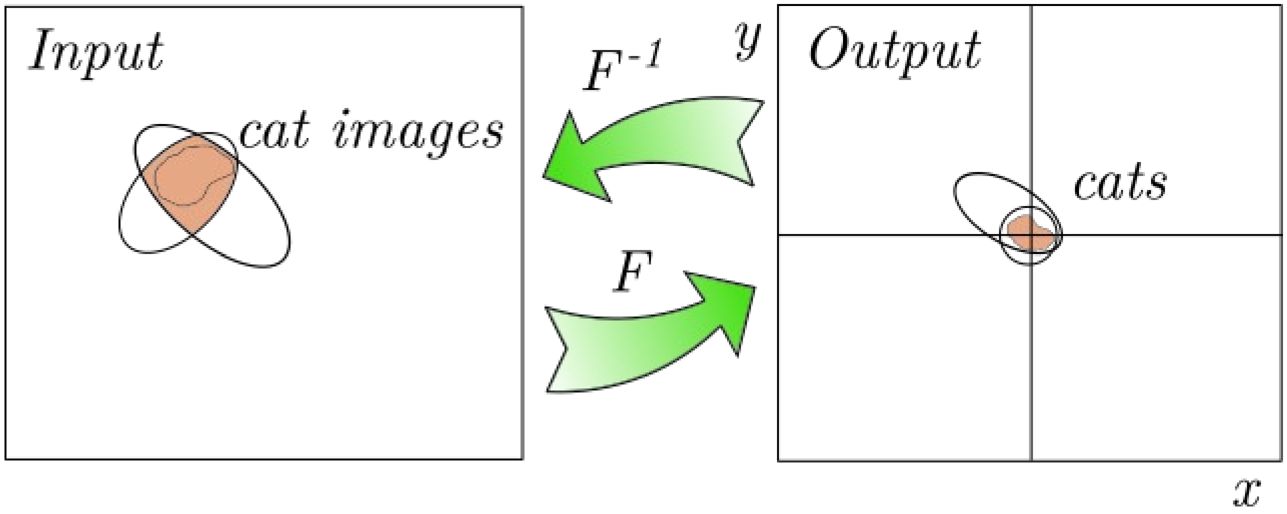
Interpretation of AI



Interpretation of AI



Interpretation of AI

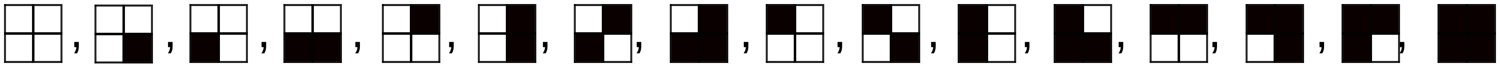


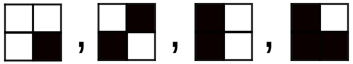
Examples of data modeling

The most elementary, but generic task is to tell if an item is element of a set.

Discrete examples: consider *2x2 bitmap "images"*, and choose a subset. Can we find the proper representation of the set where the identification of the subset is easy?

We can list all images:

X = {  }

choose an arbitrary subset, our abstract "cat images": C={  }

- the pixel-wise coordination C={0001,0110,1010,1011} : no regularity
- the pixels are not independent in C:

$$P(\xi_1=0, \xi_2=0)=1/4 \neq P(\xi_1=0)P(\xi_2=0)=1/2*3/4$$

Examples of data modeling

Find a coordination that fits the best to the problem!

X = {→0100, →0000, →0101, →0110, →0111, →1000, →0001, →1001,
→1010, →1011, →0010, →0011, →1100, →1101, →1110, →1111}

This is *not the original bit coordinates,* but it fits well to our chosen C subset!

In the new coordinates: C = {0000,0001,0010,0011}

- first two bits are 0 for elements of C: these are the relevant (selective) coordinates:
  $x \in C \Leftrightarrow x_0 = x_1 = 0$:appropriate to select the elements of C
- last two bits are variable: these are the irrelevant (descriptive) coordinates:
  to tell apart elements of C (compression) we need to consider only these coordinates