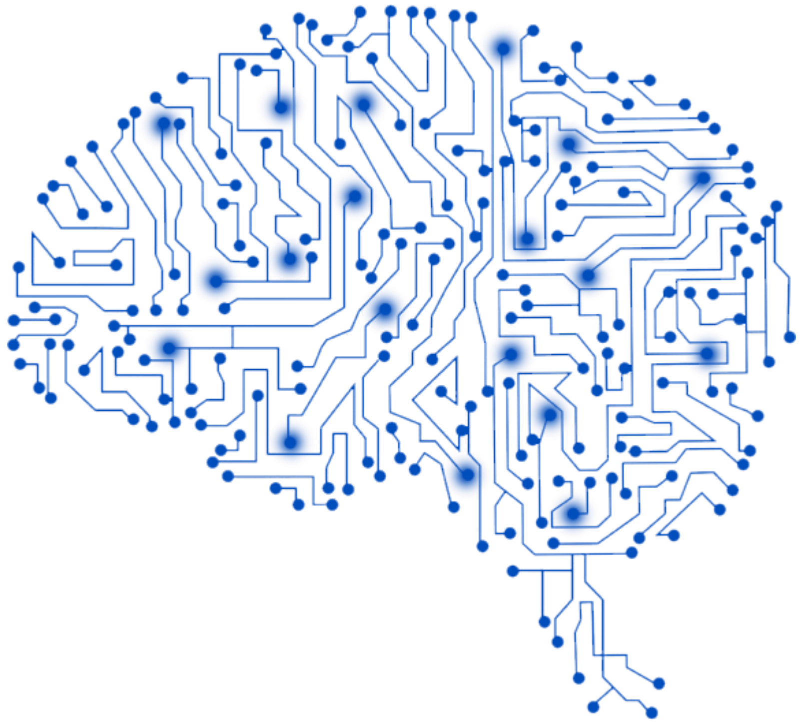
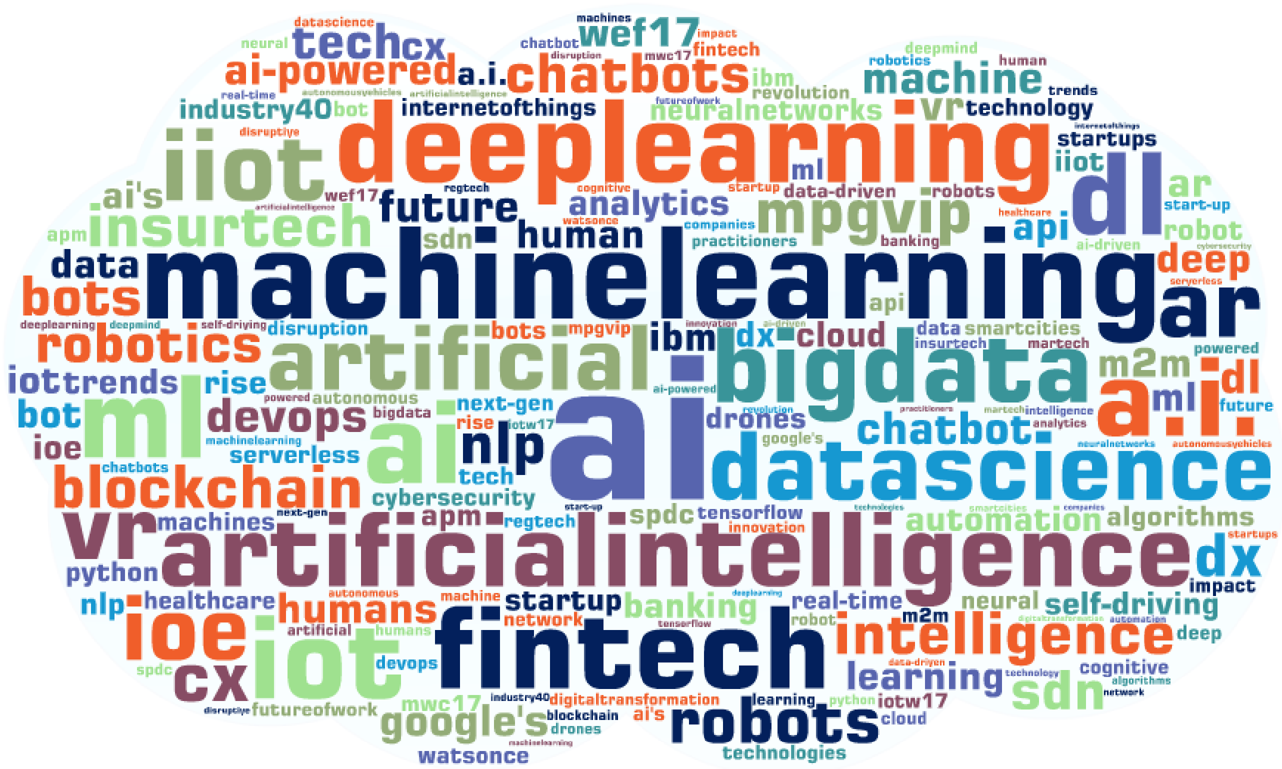
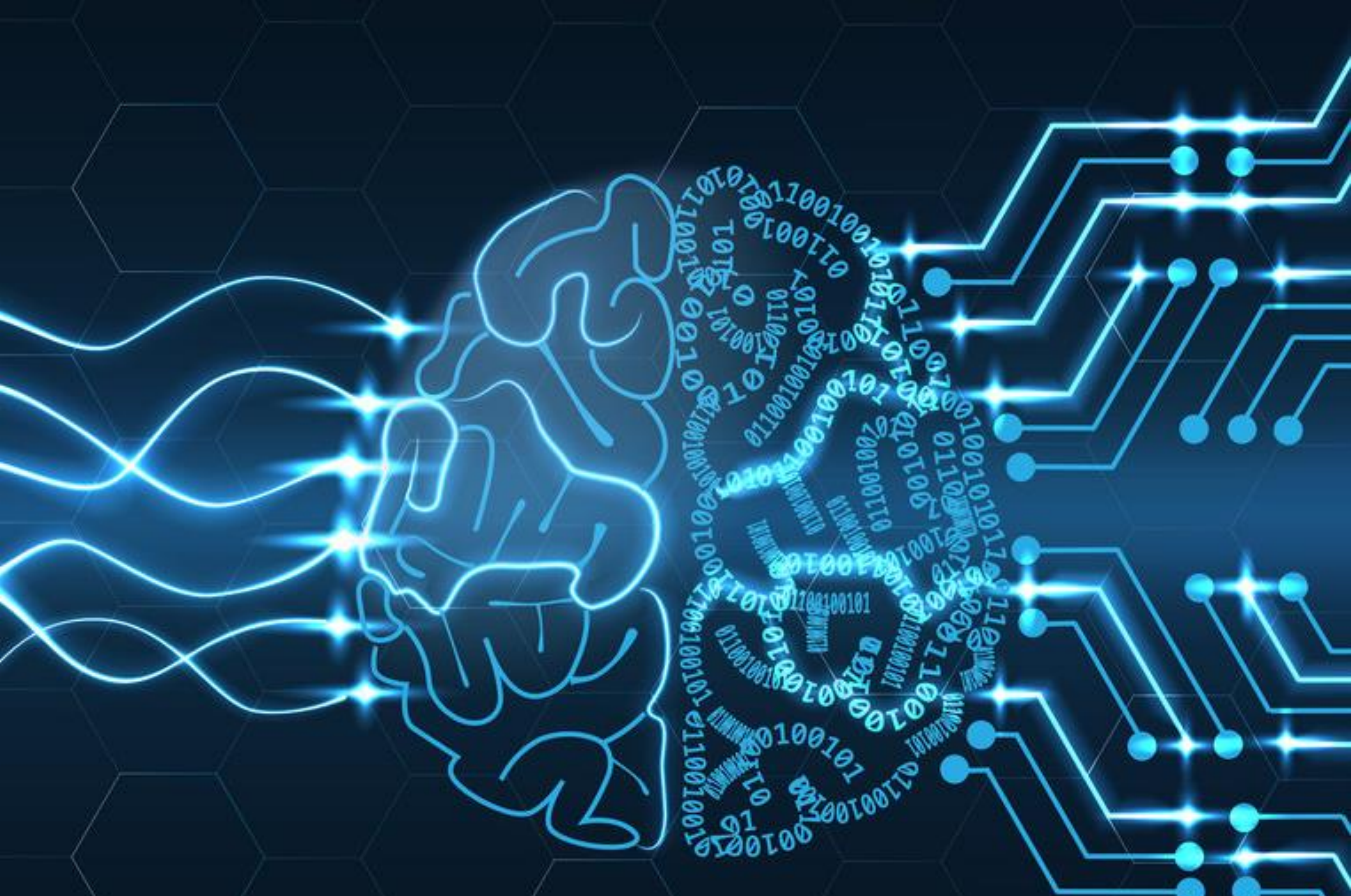
Dynamic Model Order Reduction by Machine Learning



Francesco Regazzoni

VendrediSIM - 4 février 2018


machinelearning
ai
artificialintelligence
deeplearning
fintech
bigdata
datascience
artificial
iiot
dl
a.i.
ml
vr
ar
iot
ioe
robots
intelligence
blockchain
robotics
insurtech
mpgvip
chatbots
ai-powered
techcx
wef17
machine
bots
devops
nlp
chatbot
cx
humans
learning
sdn
google's
future
human
analytics
cloud
ibm
dx
automation
algorithms
deep
robot
api
data
iottrends
bot
rise
drones
serverless
cybersecurity
machines
python
healthcare
startup
banking
real-time
self-driving
neural
cognitive
industry40
internetofthings
neuralnetworks
technology
startups
ai's
apm
technologies
watsonce
futureofwork
mwc17
digitaltransformation
iotw17
spdc
tensorflow
regtech
next-gen
autonomous
m2m
smartcities
disruption
impact
network
innovation
revolution
deepmind
datascience
neural
a.i.
tech
bigdata
powered



1 / Artificial Intelligence

Artificial Intelligence (AI)

Techniques enabling computers to mimic human intelligence

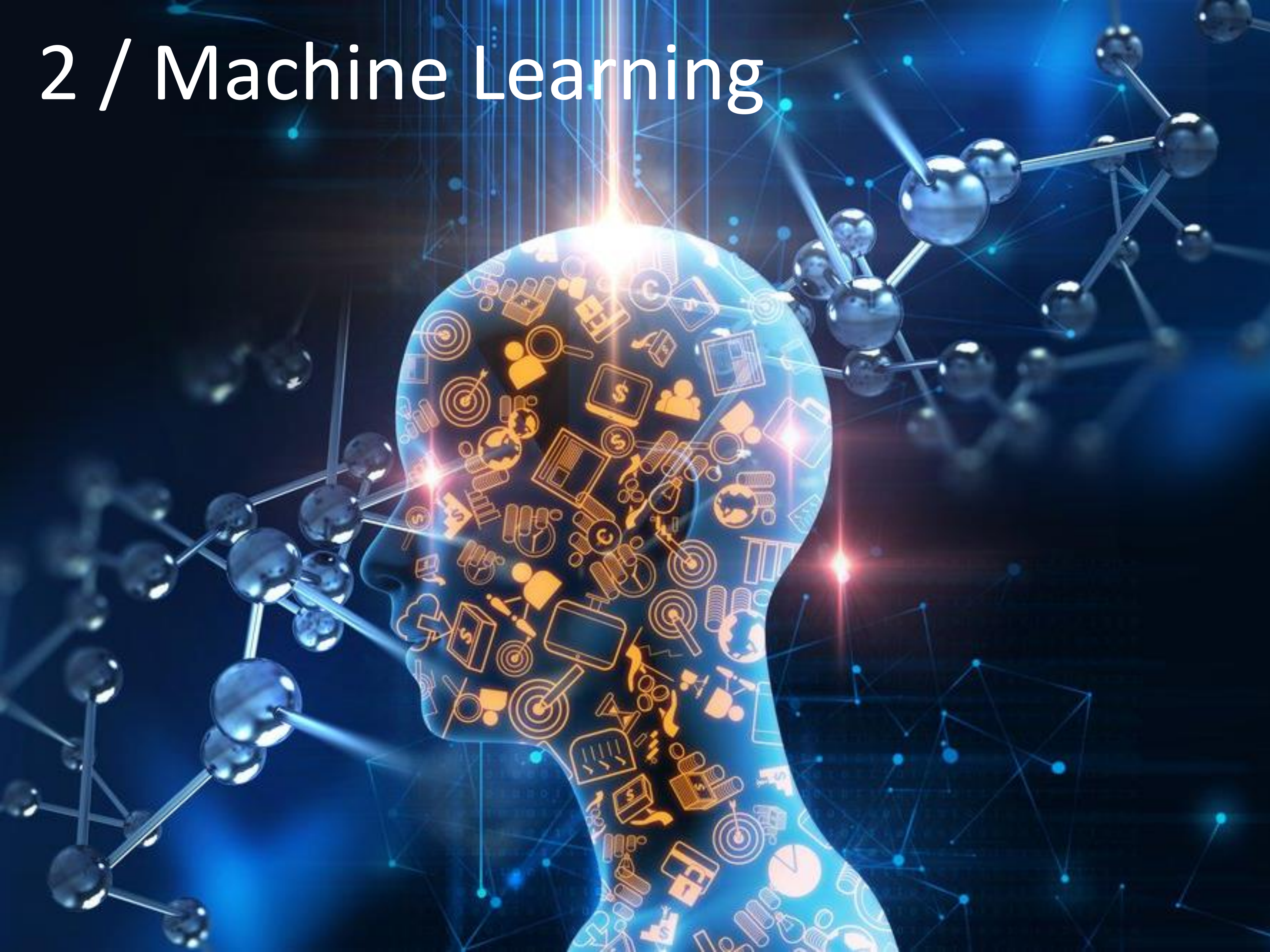
Garry Gasparov vs Deep Blue,
10 February 1997

AI Effe

Philoso

- Tur
- Chi
- ELI
- Marovec’s paradox
- …

2 / Machine Learning



Machine Learning

«Fields of study that gives computers the ability to learn without being explicitly programmed»

(Arthur Samuel, 1959)

It is a possible (and promising) approach to AI

Problem:

find the *best* function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ where $\mathbf{f}: X \to Y$

Problem rewritten:

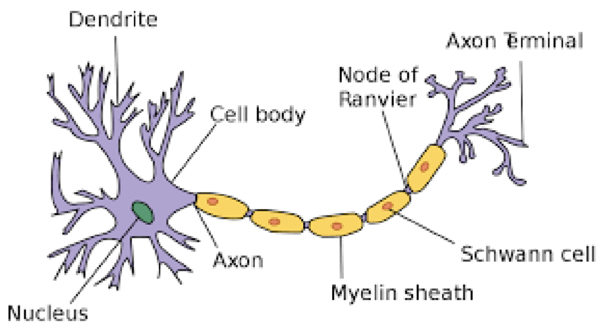
1. Select a class of functions $\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$ where $\mathbf{f}: X \times \mathcal{M} \to Y$
2. Find («learn») the *best* $\boldsymbol{\mu} \in \mathcal{M}$

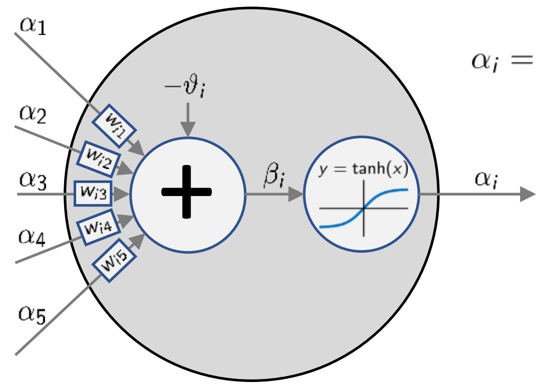
Learning machines	Learning approaches
• Decision trees / forests • Support Vector Machines (SVM) • Artificial Neural Networks (ANN) • Deep learning • Gaussian Processes (Kriging) •	• Supervised learning *(e.g. classification, regression)* • Unsupervised learning *(e.g. clustering)* • Reinforcement learning

3 /
Artificial Neural Networks

Biological vs Artificial

Neuron

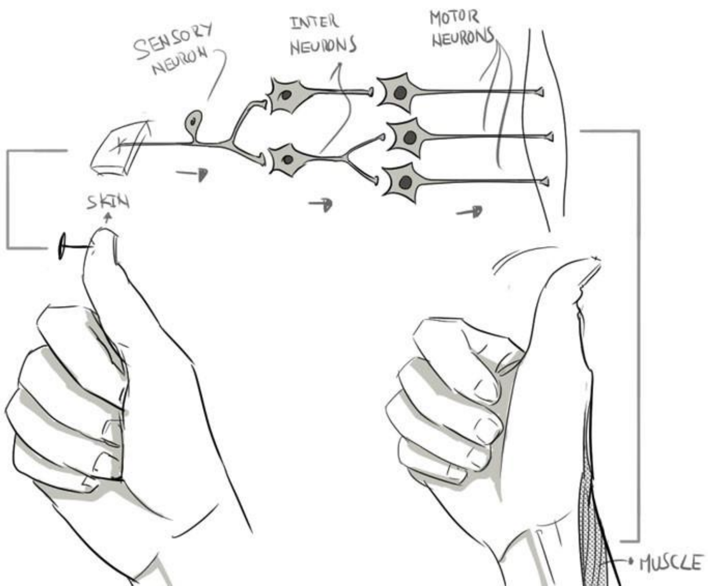




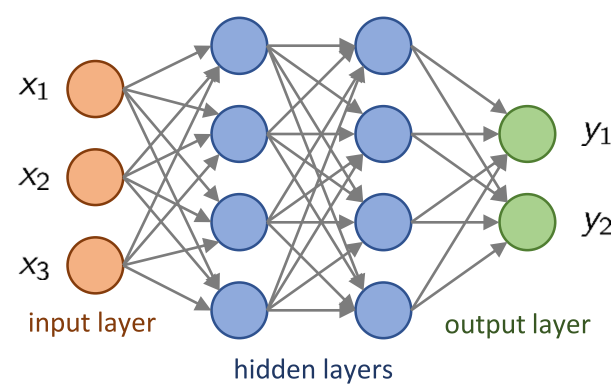
$$\beta_i = \sum_{j \in I_i} w_{ij}\alpha_j - \vartheta_i$$

$$\alpha_i = \tanh(\beta_i)$$

Neural Network

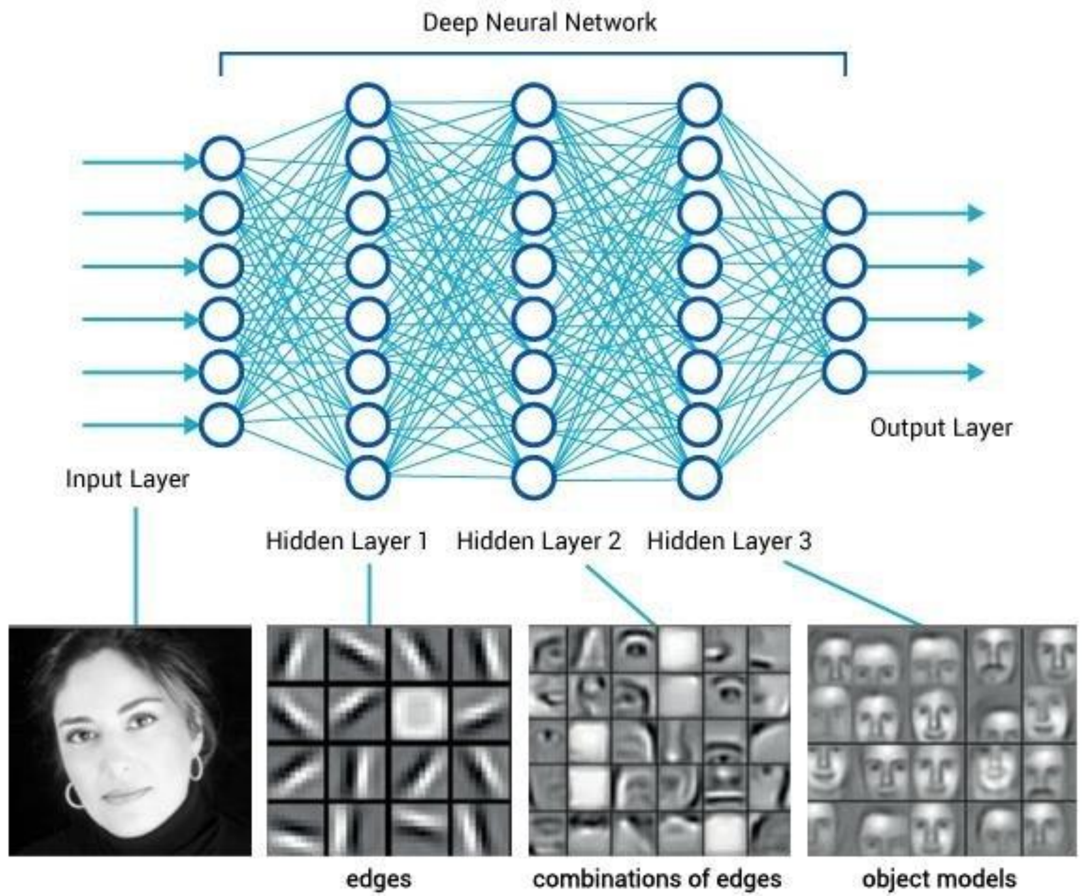


Multilayer Perceptron (MLP)



$$\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$$

Deep Learning



Universal Approximation Theorems

«A MLP with one hidden layer can approximate any *continuous* function»

Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions", Mathematics of Control, Signals, and Systems, 2(4), 303-314.

«A MLP with two hidden layer can approximate any function»

Cybenko, G. (1988). "Continuous valued neural networks with two hidden layers are sufficient". Technical Report, Department of Computer Science, Tufts University

Definition. We say that σ is *discriminatory* if for a measure $\mu \in M(I_n)$

$$\int_{I_n} \sigma(y^{\mathrm{T}}x + \theta)\, d\mu(x) = 0$$

for all $y \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ implies that $\mu = 0$.

Definition. We say that σ is *sigmoidal* if

$$\sigma(t) \to \begin{cases} 1 & \text{as} \quad t \to +\infty, \\ 0 & \text{as} \quad t \to -\infty. \end{cases}$$

Theorem 1. *Let σ be any continuous discriminatory function. Then finite sums of the form*

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(y_j^{\mathrm{T}} x + \theta_j) \tag{2}$$

are dense in $C(I_n)$. In other words, given any $f \in C(I_n)$ and $\varepsilon > 0$, there is a sum, $G(x)$, of the above form, for which

$$|G(x) - f(x)| < \varepsilon \qquad \textit{for all} \quad x \in I_n.$$

Lemma 1. *Any bounded, measurable sigmoidal function, σ, is discriminatory. In particular, any continuous sigmoidal function is discriminatory.*

Machine Learning pipeline (supervised learning)

1. **Collect train data** $\hat{\mathbf{x}}_i \quad \hat{\mathbf{y}}_i \qquad \text{for } i = 1, \dots, N$

2. **Select the hyper-parameters (nr layers, nr neurons, …)** $\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\mu})$

3. **Train the ANN** $\min_{\boldsymbol{\mu} \in \mathcal{M}} \frac{1}{2} \sum_{i=1}^{N} \|\hat{\mathbf{y}}_i - \mathbf{f}(\hat{\mathbf{x}}_i; \boldsymbol{\mu})\|^2 + \text{regularization}$

https://playground.tensorflow.org

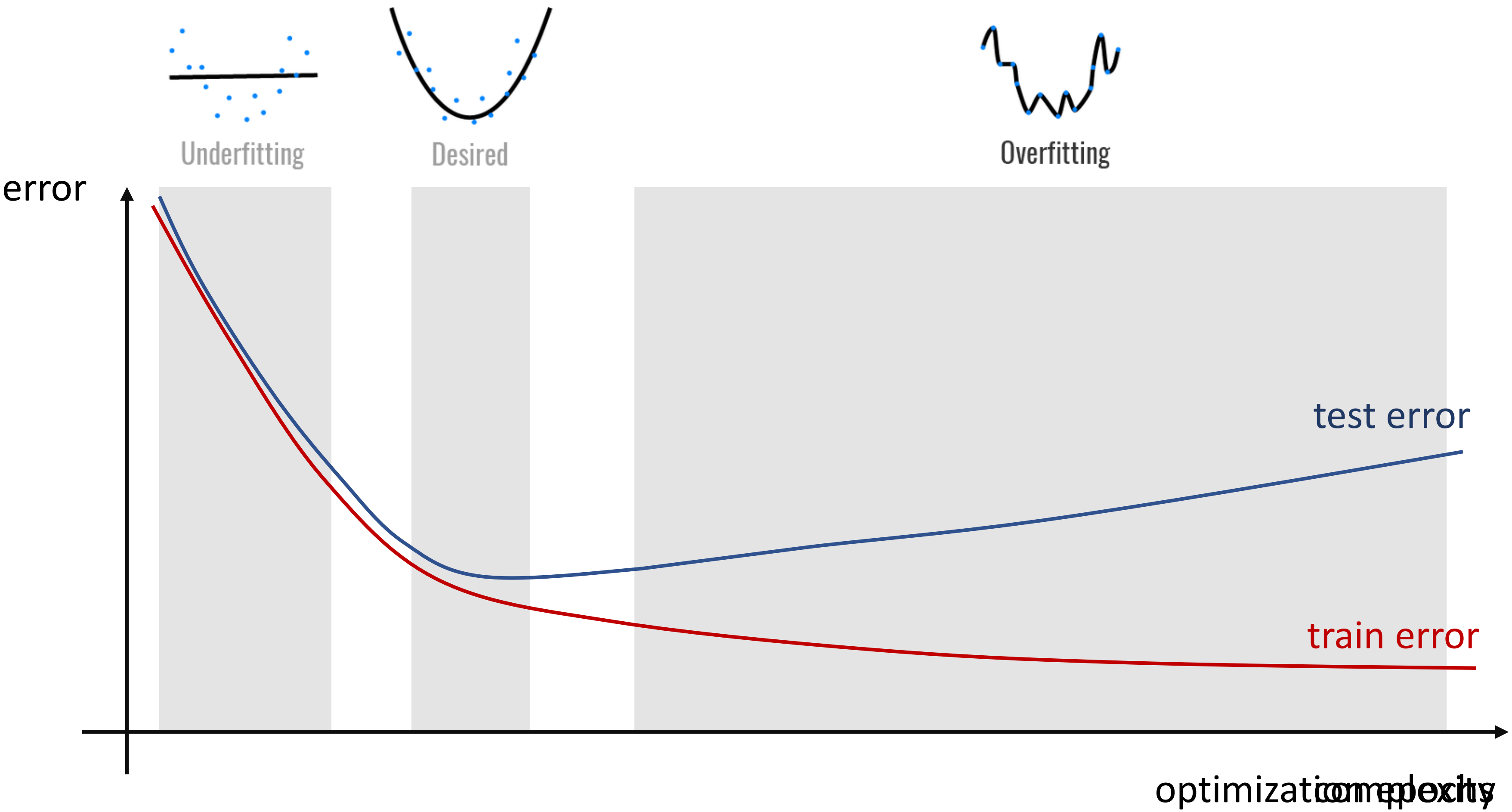
William of Ockham

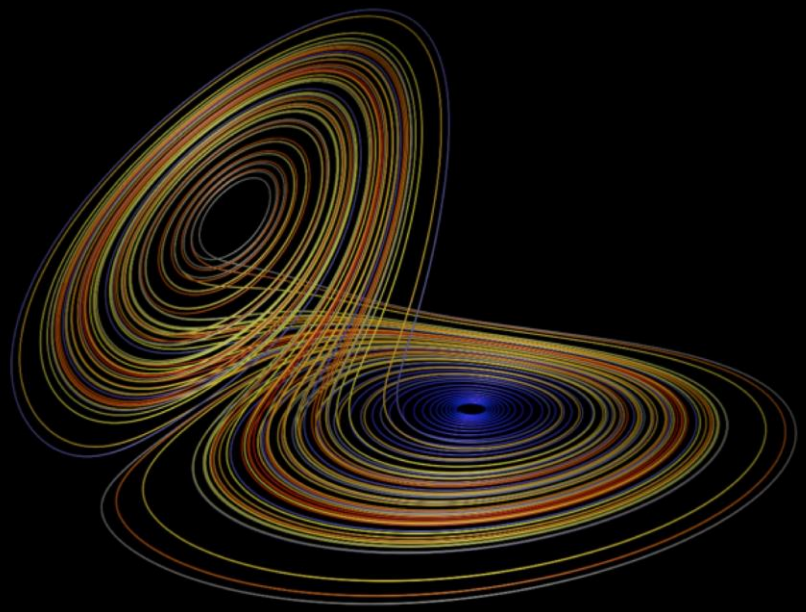
theologician, philosopher and franciscan friar

1285 – 1347

« Frustra fit per plura
quod fieri potest per pauciora »

Ockham’s Razor in action





4 / Reducing complexity in dynamical models through ANNs

Model Order Reduction (MOR) of Dynamical Systems

High-Fidelity Model ($\mathbf{X} \in \mathbb{R}^N$)

$$\begin{cases} \dot{\mathbf{X}}(t) &= \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t)), \qquad t \in (0, T] \\ \mathbf{X}(0) &= \mathbf{X}_0 \end{cases}$$
$$\mathbf{y}(t) = \mathbf{G}(\mathbf{X}(t)), \qquad t \in (0, T],$$

$\xrightarrow[n \ll N]{\text{MOR}}$

Reduced Order Model ($\mathbf{x} \in \mathbb{R}^n$)

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \qquad t \in (0, T] \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{cases}$$
$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)), \qquad t \in (0, T],$$

- **Model-based MOR**
  - Projection-based MOR (Moment-matching, Padè approximation, Balanced Truncation, POD, PCA, Karhunen-Loeve expansion, Reduced Basis, …)
  - Hierarchical surrogates (simplified geometries, coarser grids, simplified physical assumptions, Geometric Multiscale, …)

- **Data-driven MOR**
  - *Linear case*: Loewner framework, Orthonormal Vector Fitting
  - *Nonlinear case*: Dynamic Mapping Kriging (DMK), Sparse Identication of Nonlinear Dynamics (SINDy)

Data-driven approach to dynamic MOR

$\mathbf{u}$

$$\begin{cases} \dot{\mathbf{x}} & = \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} & = \mathbf{g}(\mathbf{x}) \end{cases}$$

High fidelity model

$\mathbf{x} \in \mathbb{R}^n$ (with n small)

$\mathbf{y}$

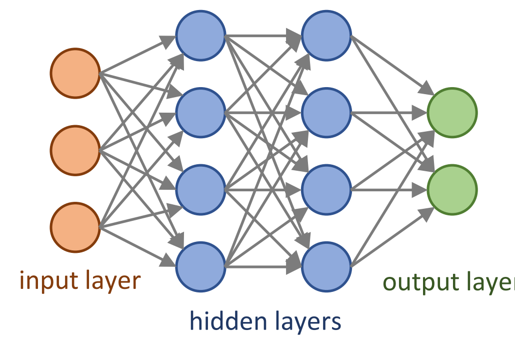
Observations:

$\hat{\mathbf{u}}_j(t), \hat{\mathbf{y}}_j(t) \quad j = 1, \dots, k$

$$\begin{cases} \min_{\mathbf{f},\mathbf{g}} & \frac{1}{2}\sum_{j=1}^{k}\int_0^T \|\hat{\mathbf{y}}_j - \mathbf{g}(\mathbf{x}_j)\|^2 dt \\ \text{s.t.} & \dot{\mathbf{x}}_j = \mathbf{f}(\mathbf{x}_j, \hat{\mathbf{u}}_j) \\ & \mathbf{x}_j(0) = \mathbf{0} \end{cases}$$

Discretize $\mathbf{f}$ and $\mathbf{g}$ through a finite set of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, by means of Artificial Neural Networks

Multilayer Perceptron (MLP)



First-Order Optimality Conditions (Lagrange multipliers method)

$$\begin{cases} \dot{\mathbf{x}}_j & = \mathbf{f}(\mathbf{x}_j, \hat{\mathbf{u}}) \\ \mathbf{x}_j(0) & = \mathbf{0} \end{cases}$$

Primal (forward) system

$$\begin{cases} -\dot{\mathbf{z}}_j & = \nabla_{\mathbf{x}}^T \mathbf{g}\,\left(\hat{\mathbf{y}}_j - \mathbf{g}(\mathbf{x}_j)\right) + \nabla_{\mathbf{x}}^T \mathbf{f}\,\mathbf{z}_j \\ \mathbf{z}_j(T) & = \mathbf{0} \end{cases}$$

Dual (backward) system

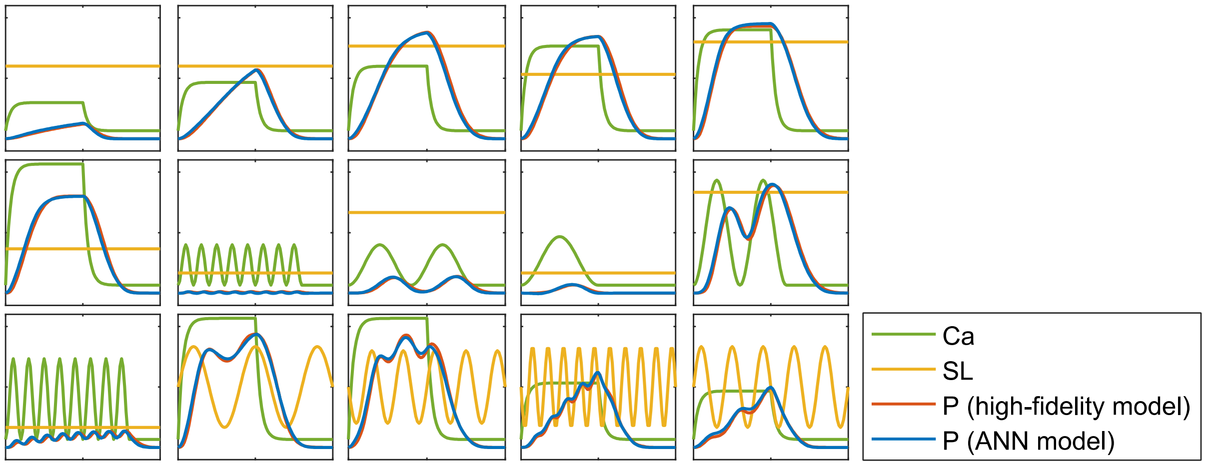
$$\begin{cases} \sum_{j=1}^{k}\int_0^T \nabla_{\boldsymbol{\mu}}^T \mathbf{f}\,\mathbf{z}_j dt & = \mathbf{0} \\ \sum_{j=1}^{k}\int_0^T \nabla_{\boldsymbol{\nu}}^T \mathbf{g}\,\left(\hat{\mathbf{y}}_j - \mathbf{g}(\mathbf{x}_j)\right) dt & = \mathbf{0} \end{cases}$$

Vanishing gradient condition

Numerical resolution (Levenberg-Marquardt)

An example: sarcomere RU model

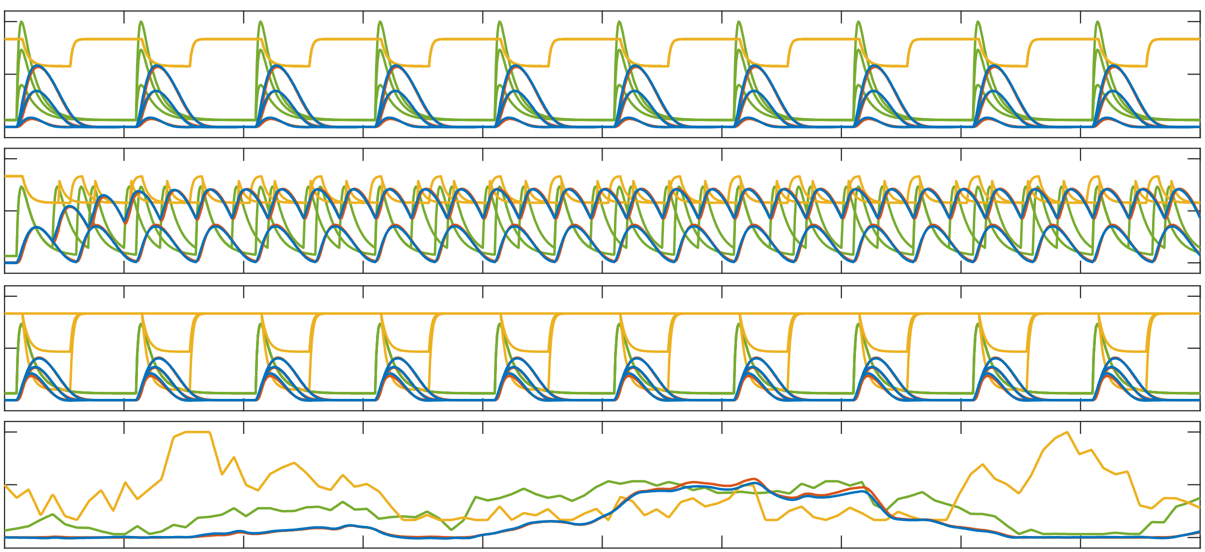
1. Train (~1 day)



- 103 samples
- 1 s duration each
- Fundamentals functions: (constants, steps, sinusoidals)

L^2 error: $1.8 \cdot 10^{-2}$

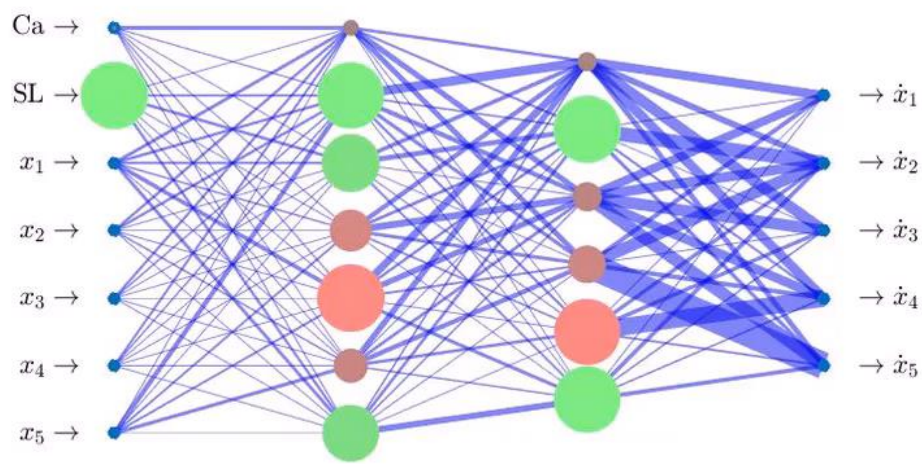
2. Test

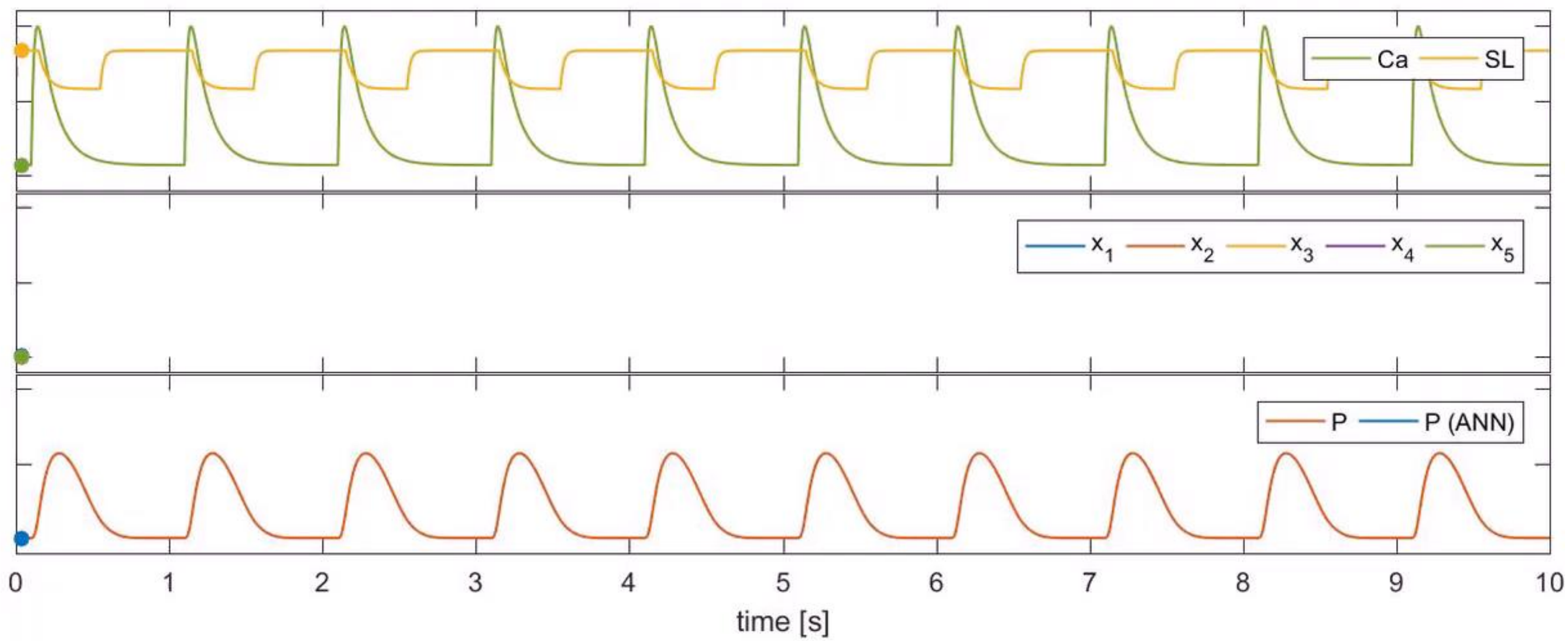


- Train-test correlation while learning: 99,8%
- Reliable also over long-term horizons

L^2 error: $3.0 \cdot 10^{-2}$

Neural Network Model in action





Merci pour votre attention.