Physics-enhanced Machine Learning
Part one: Physics-informed neural networks

M. Martinez, L. Tesan, P. Urdeitx, C. Bermejo, A. Tierz, **I. Alfaro**, *D. Gonzalez, F. Chinesta, E. Cueto*
ESI Group-UZ Chair of the National Strategy on AI

UKACM-SEMNI Autumn School

España | digital 2026

Universidad Zaragoza 1542

Aragon Institute of Engineering Research

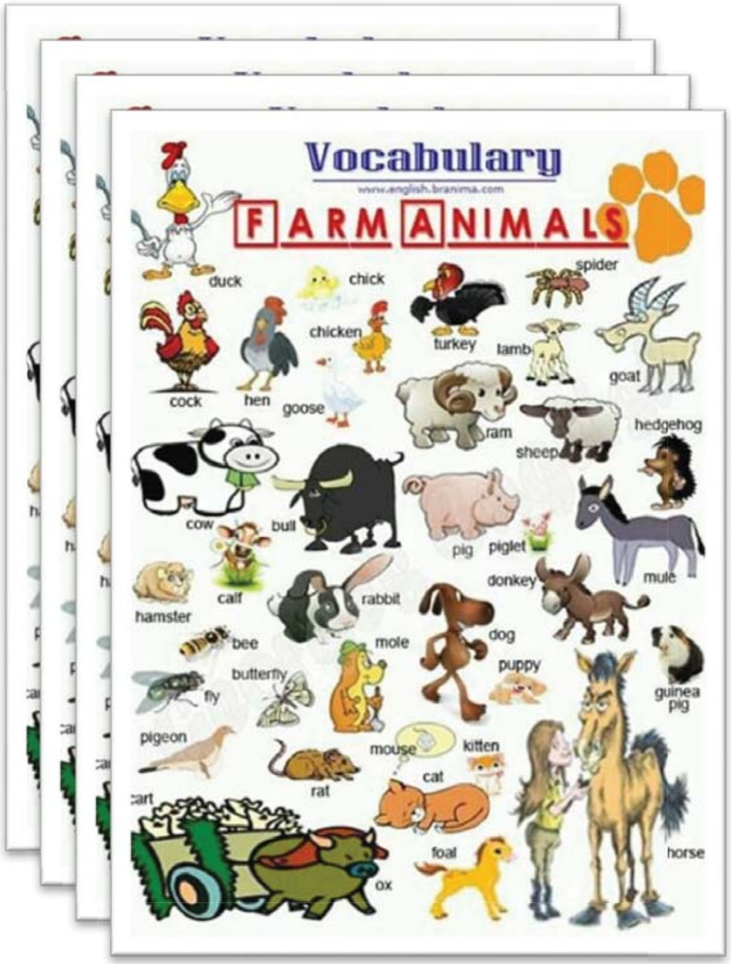
Table of context

Table of context

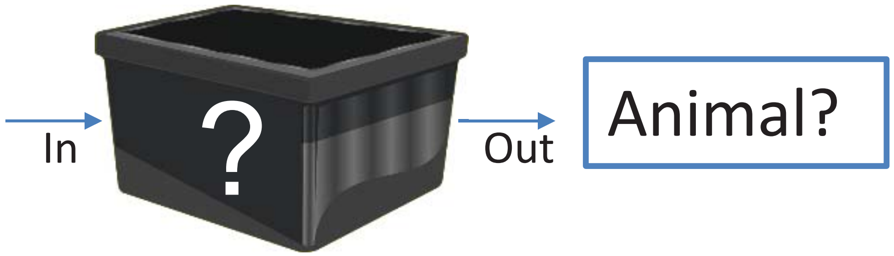
Motivation

- Neural Networks (NNs) can be seen as a black box

image taken from Attanatta in https://www.flickr.com/



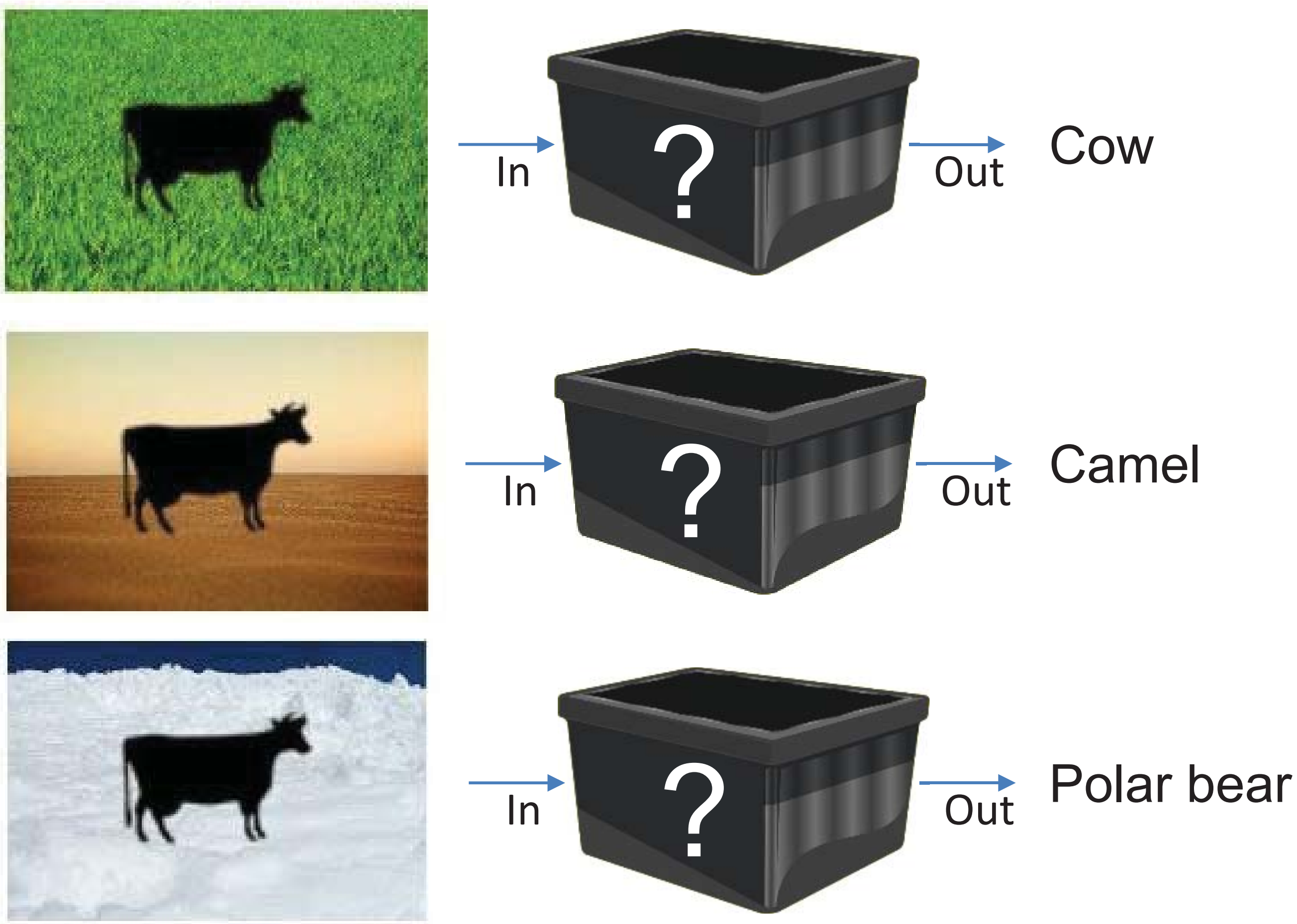
Lots of data



Black Box

Motivation

- Neural Networks (NNs) are seen as a black box



Not always reliable

Motivation

- Need robust and fast solvers

Simulator	Physics	Cost
Traditional (CFD, FEM)	✓	✗
Game engines (Havok, Unreal)	~	✓
Deep learning (NNs)	✗	✓

Motivation

- Need robust and fast solvers

Simulator	Physics	Cost
Traditional (CFD, FEM)	✓	✗
Game engines (Havok, Unreal)	~	✓
Deep learning (NNs)	✗	✓

Incorporate physics consistency to Deep Learning!

Motivation

- Inductive biases: set of assumptions to improve generalization [1]

No physics | Little physics | Lots of physics

NN — Neural Networks

TINNs — Thermodynamics Informed Neural Networks

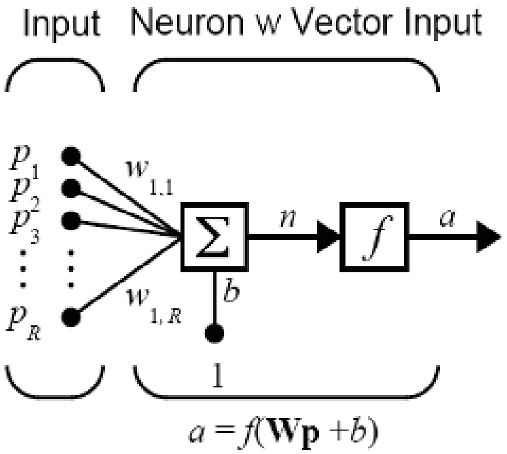
PINNs — Physics Informed Neural Networks

[1] Battaglia, P. et al. (2018) **Relational inductive biases, deep learning, and graph networks.**

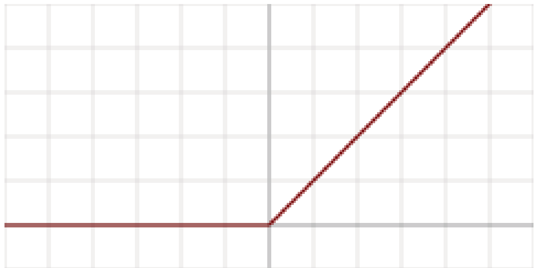
Table of context

Introduction to NN

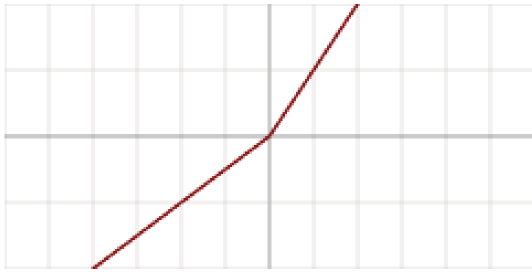
- Artificial neuron



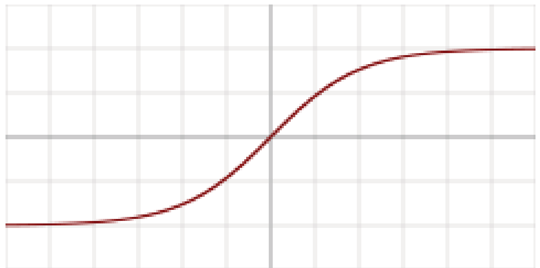
- Activation functions f



RELU



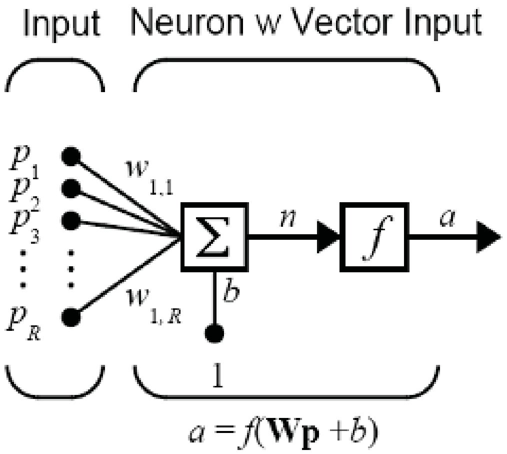
Leaky RELU



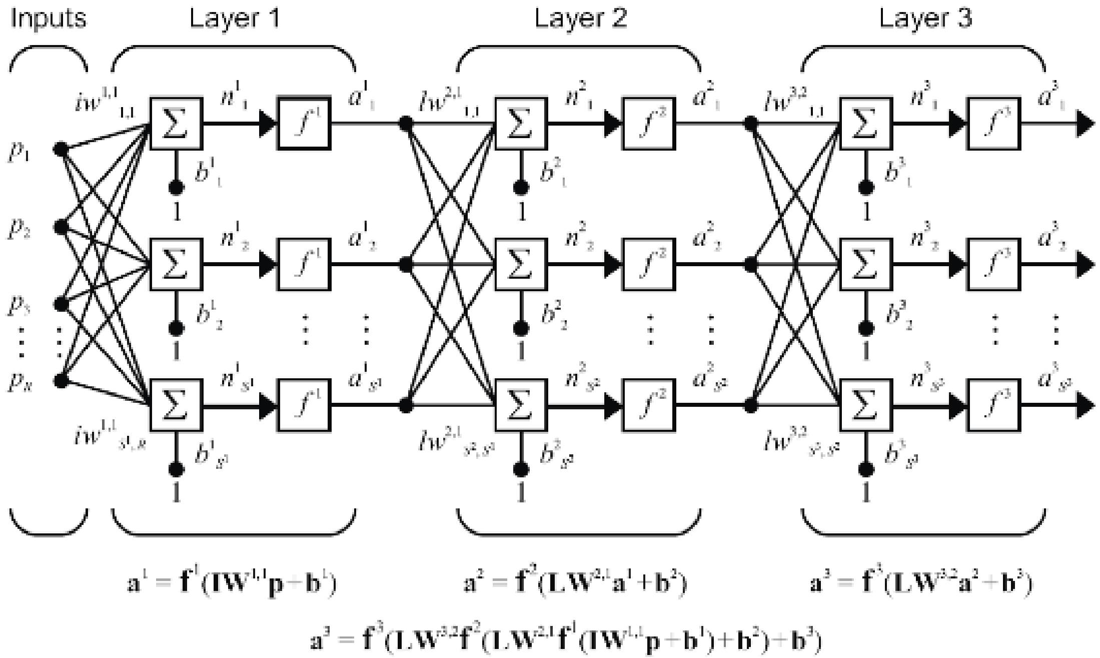
Hyperbolic tangent

Introduction to NN

- Artificial neuron

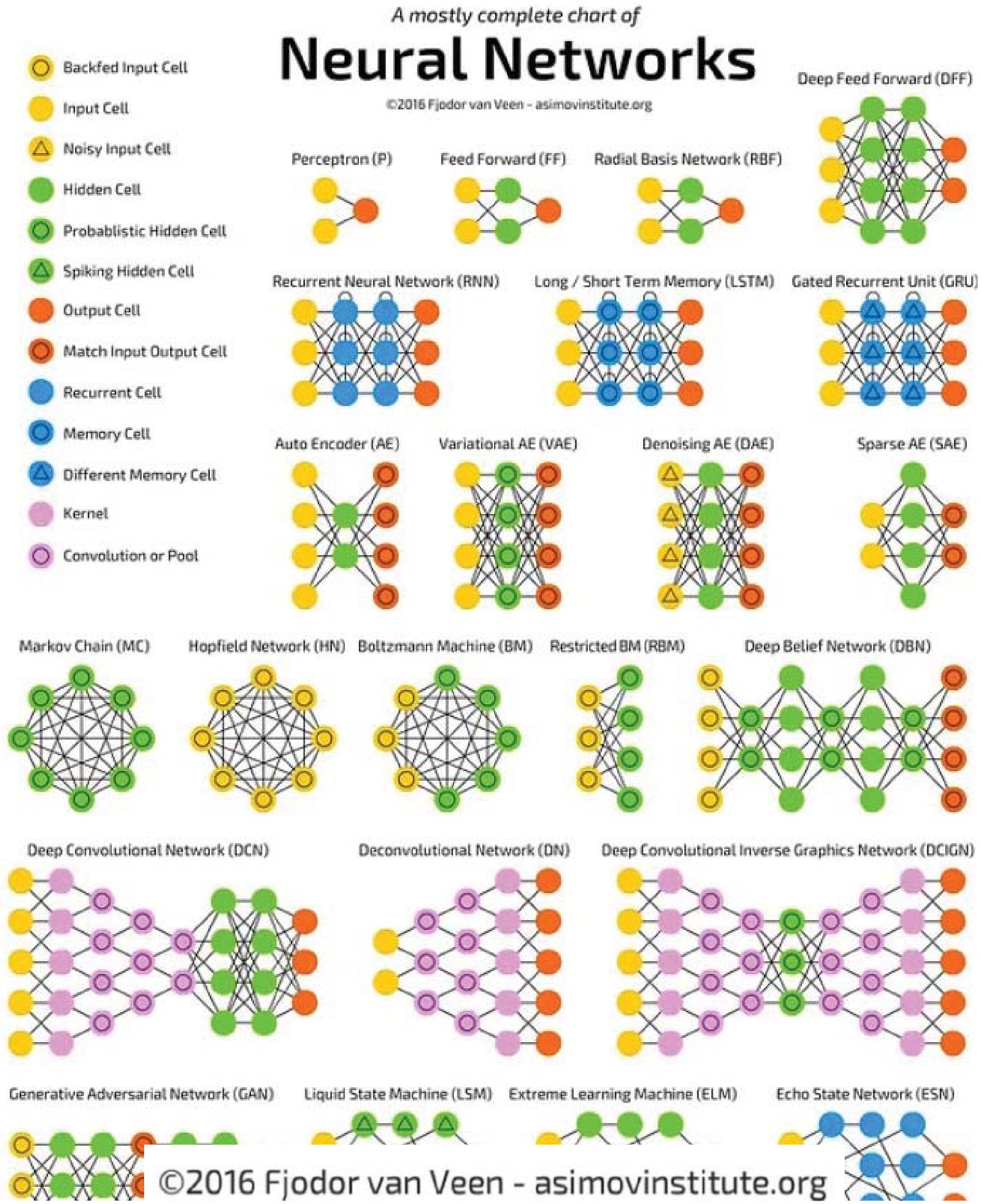


- Multilayer Perceptron (MLP)



Introduction to NN

- Many different architectures



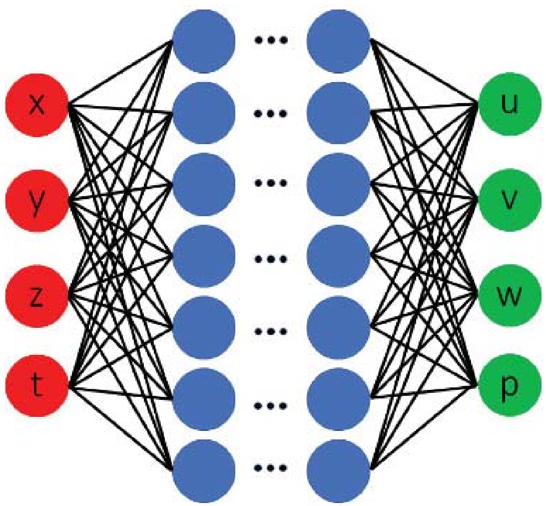
https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

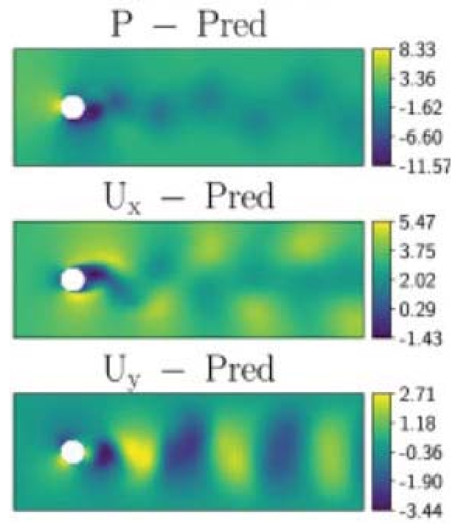
Introduction to NN

- Neural Networks (NNs) provide great versatility

$$\boldsymbol{a}^{nLayers} = f^{nLayers}(\dots(L \cdot W^{4,3} \cdot f^{3}\,(L \cdot W^{3,2} \cdot f^{2}(L \cdot W^{2,1} \cdot f^{1}(L \cdot W^{1,1} \cdot \boldsymbol{p} + b^{1}) + b^{2}) + b^{3}) + \cdots) + b^{nLayers})$$

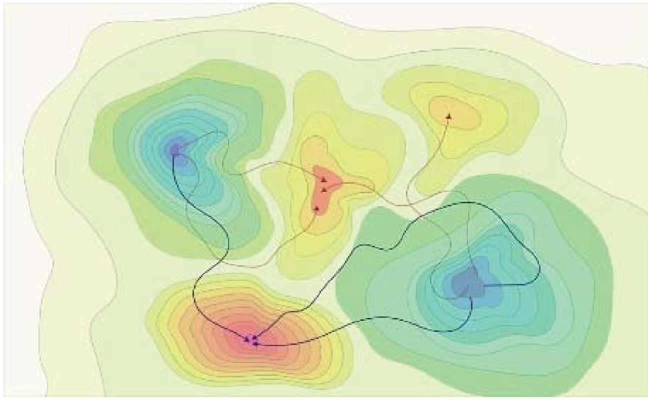
- For example, NNs can give us physical results for input coordinates:





Introduction to NN

- Training Process:
  - Iterative process (number of epochs) until a minimum error (loss function) is reached.
  - Loss function depends on the problem or application to be performed.



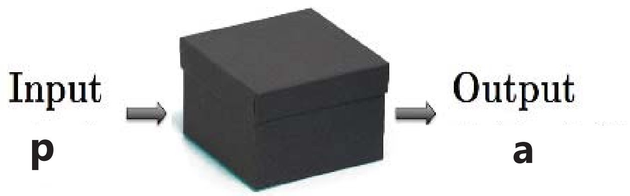
  - Optimization problem:
    Given
      - Data
      - net architecture
      - loss definition

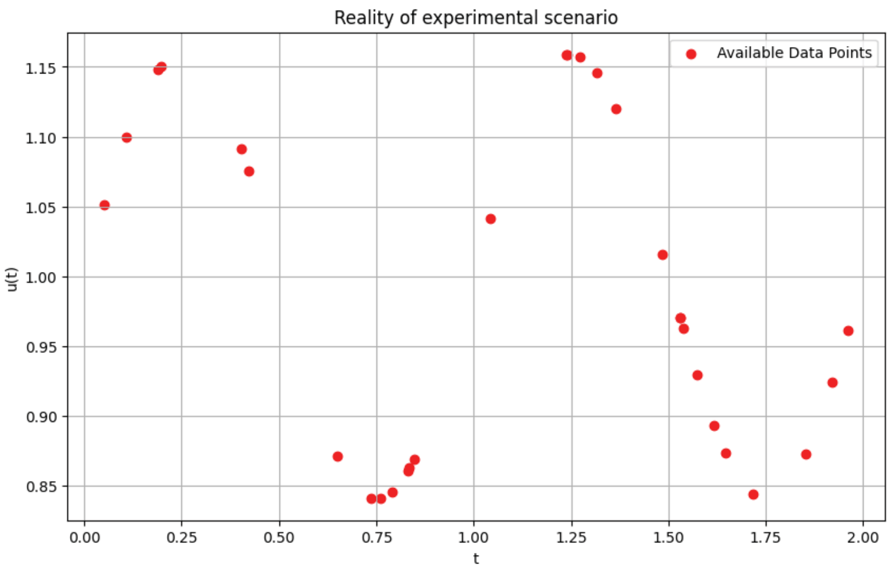
    calculate all weights (W) and biases (b) that minimizes the loss

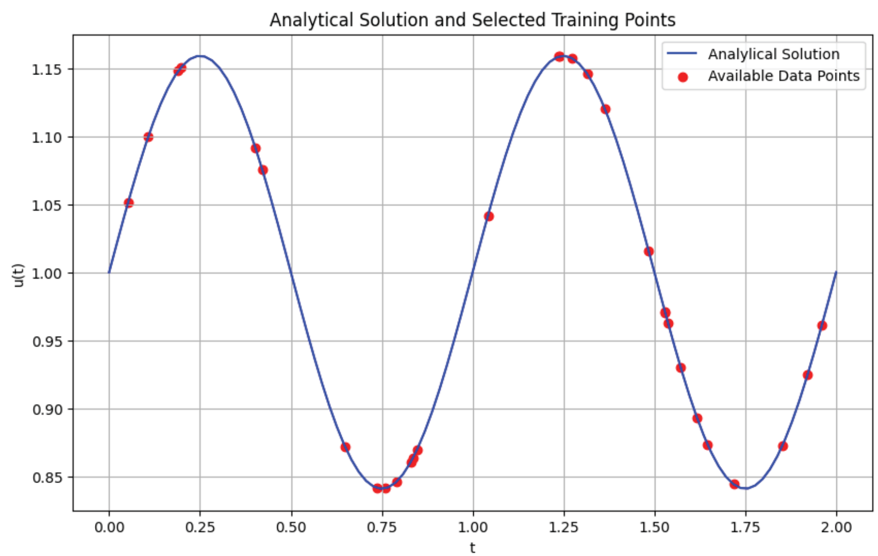
  - Backpropagation computes the gradient in the weight space of a NN, with respect to the loss function.
  - Automatic differentiation is used to calculate the derivatives.

Supervised Learning: Learning from data

- Based only in data:
  - Black box
  - Data: pairs of $\left(\boldsymbol{p}^{input}, \boldsymbol{a}^{output}\right)$
  - Data loss: $L_{data} = \left\|\boldsymbol{a}^{real} - \boldsymbol{a}^{net}\right\|^2$
  - Many data pairs are needed
  - Data must be very good

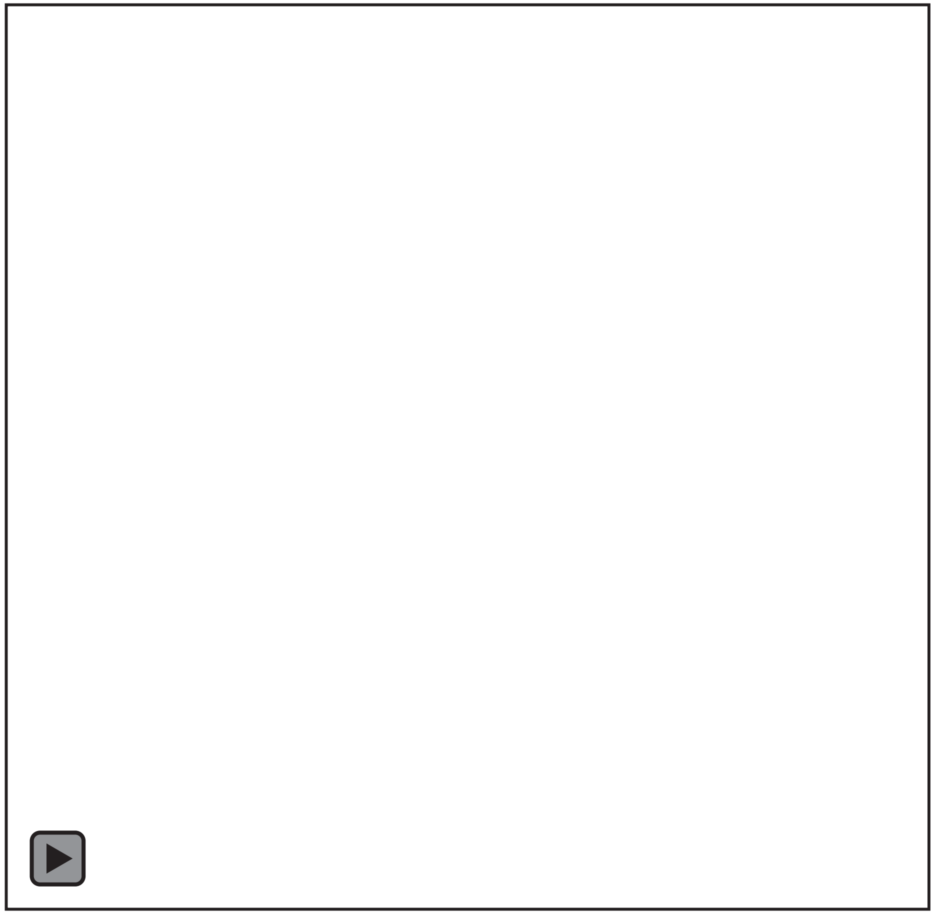






Supervised Learning: Learning from data

- Based only in data:
  - Black box



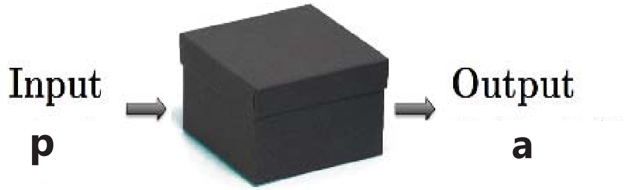
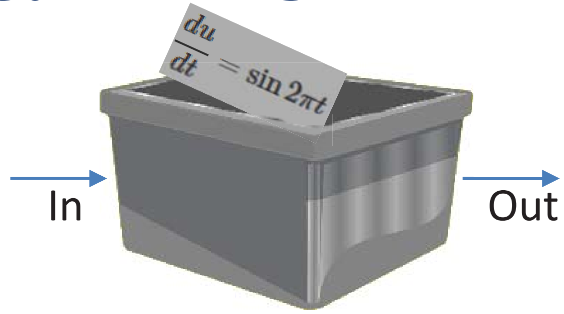


Table of context

Physics-informed neural networks: PINNs

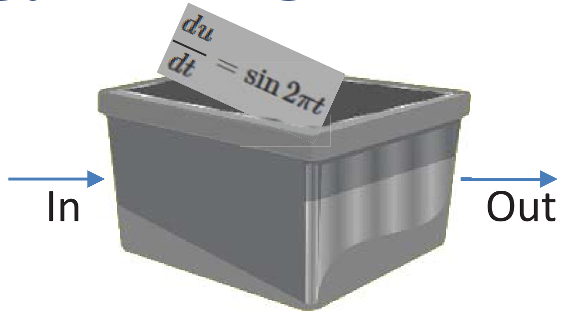
- Include physics information
- Force PDE fulfilment in the loss term



Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Physics-informed neural networks: PINNs

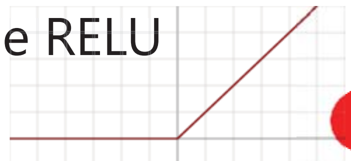
- Include physics information



- Force PDE fulfilment in the loss term

- Deep Learning is differentiable by default. The order of derivation of the activation functions has to be enough to solve the PDE.

  - Don't use RELU



- Self-supervised:
  no need explicit data
  (collocation points+PDE)

- Semi-supervised:
  also experiental data

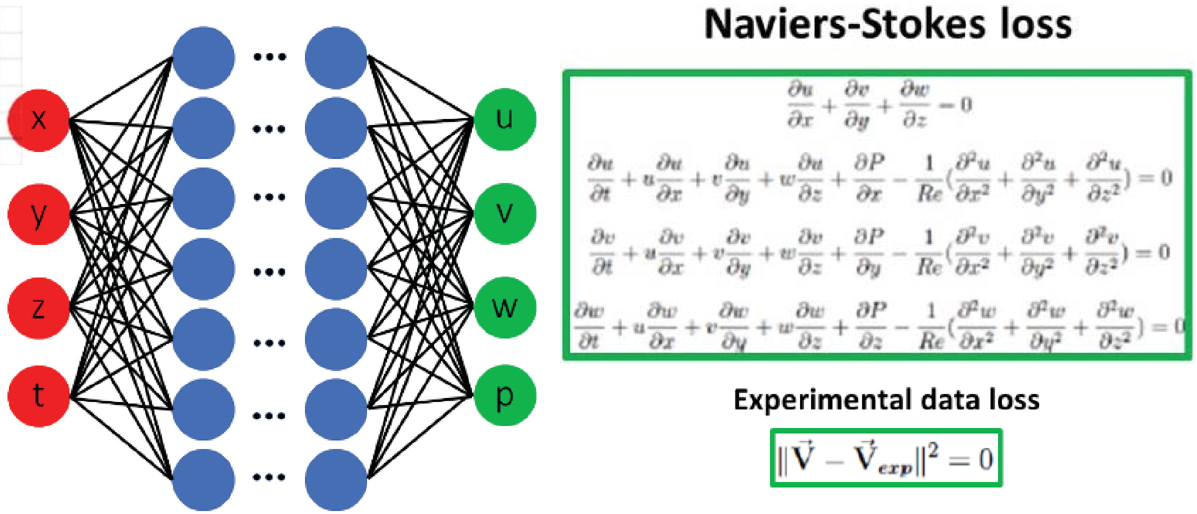


Figure: Physics-informed neural for the solution of the Navier-Stokes equations. Source: Wikipedia

Physics-informed neural networks: PINNs

- Loss can be composed of:

- $L_{data} = \left\|\boldsymbol{a}^{real} - \boldsymbol{a}^{net}\right\|^2$
  - Pairs of $\left(\boldsymbol{p}^{input}, \boldsymbol{a}^{output}\right)$
  - Experimental datapoints (sensors) or simulated datapoints, (real data-virtual data)

- L_{PDE}
  - Check if the PDE system is satisfied

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} + \frac{\partial P}{\partial x} - \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + \frac{\partial P}{\partial y} - \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) = 0$$

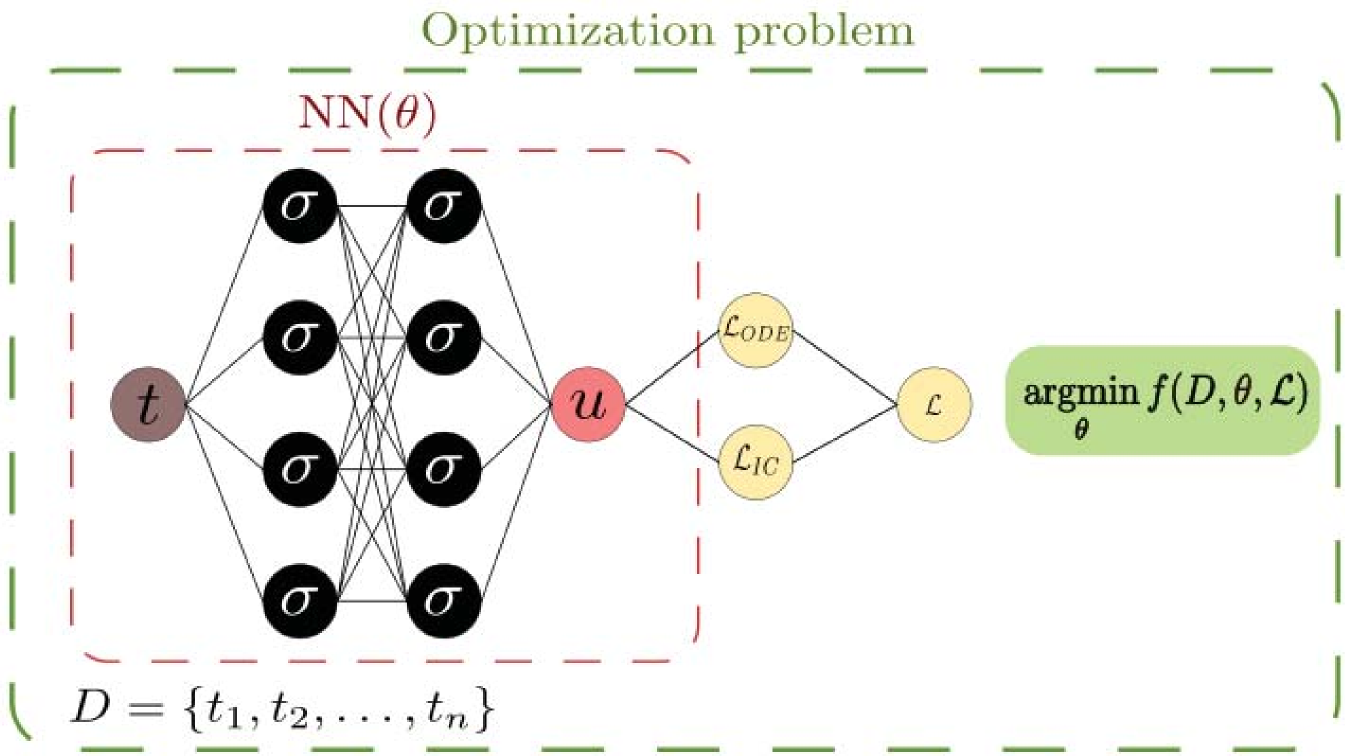
$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} + \frac{\partial P}{\partial z} - \frac{1}{Re}\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) = 0$$

- L_{IC} , L_{BC}
  - Check if initial and boundary conditions are satisfied

- Total loss= weighted sum of these partial losses

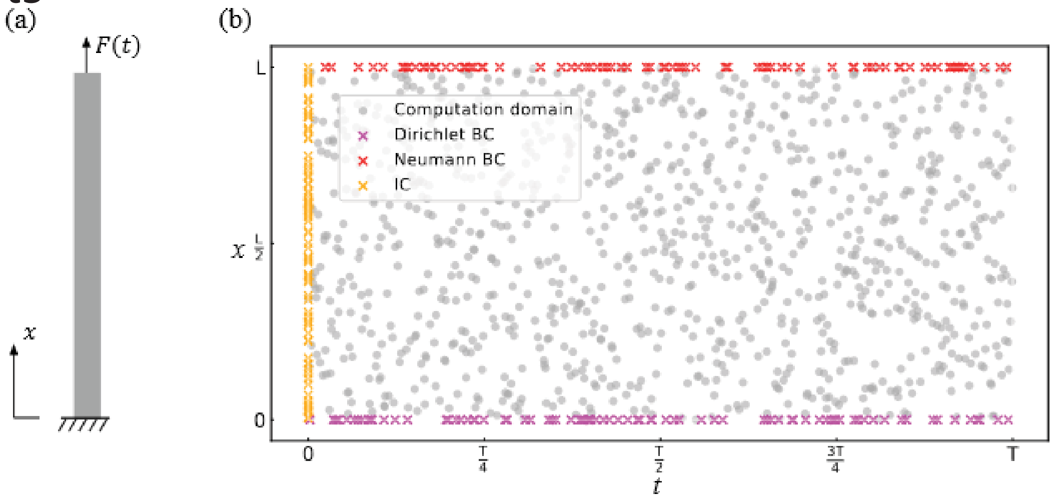
Physics-informed neural networks: PINN

- Advantages: we like to work with our preferred behavior model
  - Very simple way to add physics to the net



Physics-informed neural networks: PINN

- Advantages:
  - Very simple way to add physics to the net
  - Don't need real data or only few points (sensors)
  - Choose the collocation points



Technical Papers | May 21, 2021

Physics-Informed Deep Learning for Computational Elastodynamics without Labeled Data

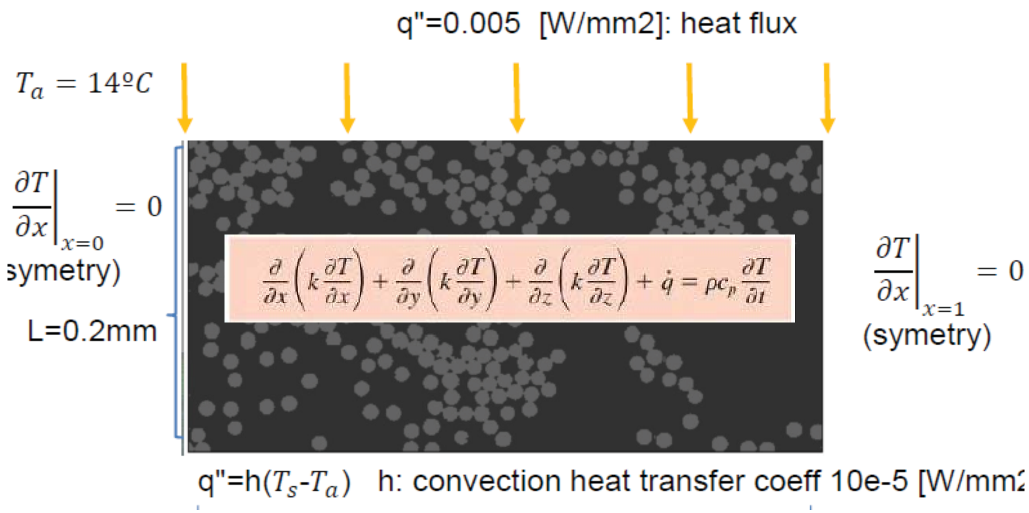
Authors: Chengping Rao, Hao Sun, A.M.ASCE, and Yang Liu, A.M.ASCE | AUTHOR AFFILIATIONS

Publication: Journal of Engineering Mechanics • Volume 147, Issue 8 • https://doi.org/10.1061/(ASCE)EM.1943-7889.0001947

Physics-informed neural networks: PINN

- Advantages:
  - Very simple way to add physics to the net
  - Don't need real data or only few points (sensors)
  - Choose the collocation points
  - Can be used for Forward simulation and for Inverse problems

$$K = f(\%matrix\ \%fiber,\ imperfections(type, size), \dots)$$



Physics-informed neural networks: PINN

- Disadvantages:
  - Difficult to train (loss composed of real and virtual data)

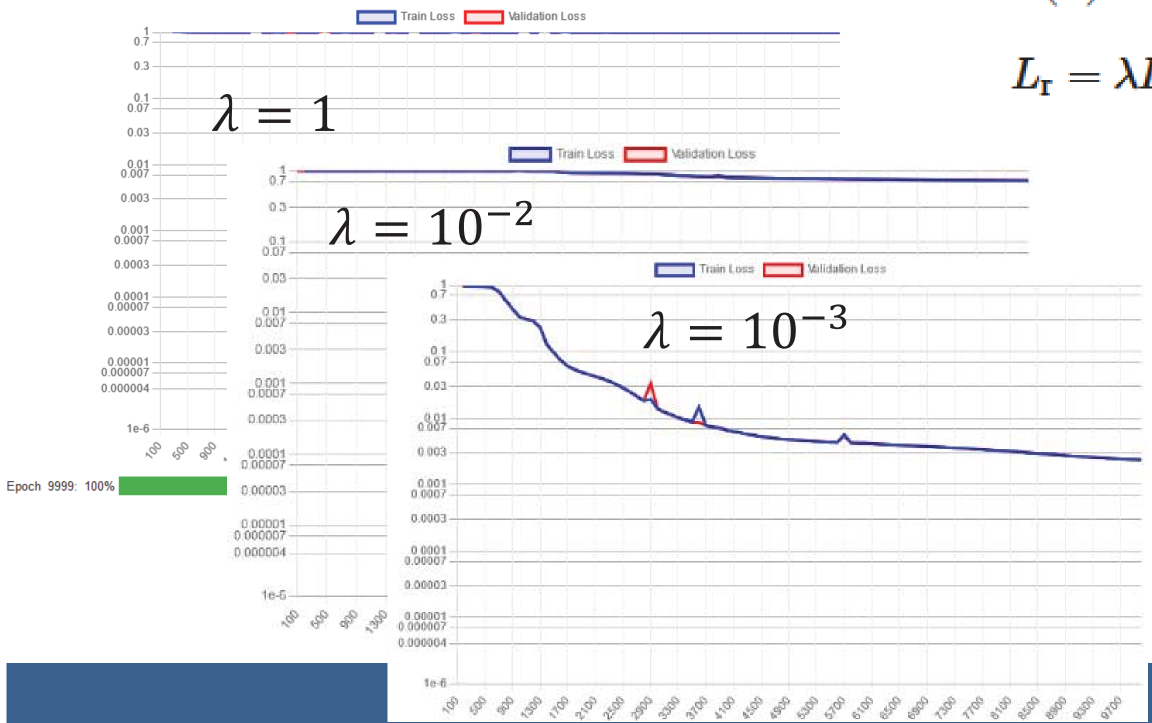
Physics-informed neural networks: PINN

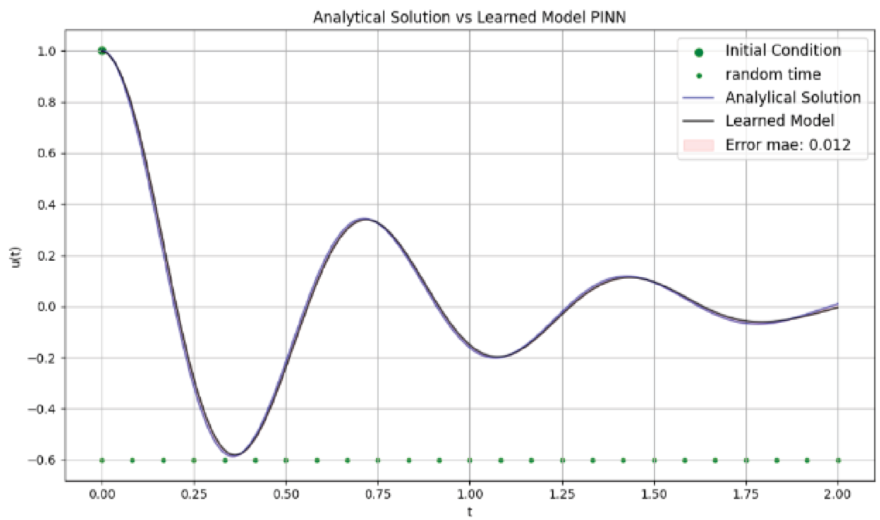
- Disadvantages:
  - Difficult to train (loss composed of real and virtual data)
  - Tunning hyperparameters

$$L_{\text{Total}} = L_{\text{data}} + \lambda(L_{\text{ODE}} + L_{\text{IC}})$$

$$\mathcal{L}(\theta) = \lambda_{ic}\mathcal{L}_{ic}(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta) + \lambda_r\mathcal{L}_r(\theta),$$

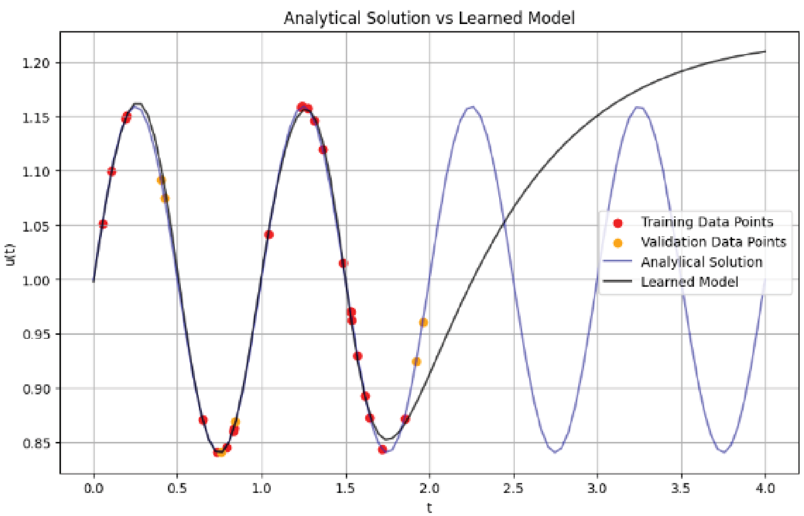
$$L_{\text{r}} = \lambda L_{\text{ODE}} + L_{\text{IC}_{position}} + L_{\text{IC}_{velocity}}$$

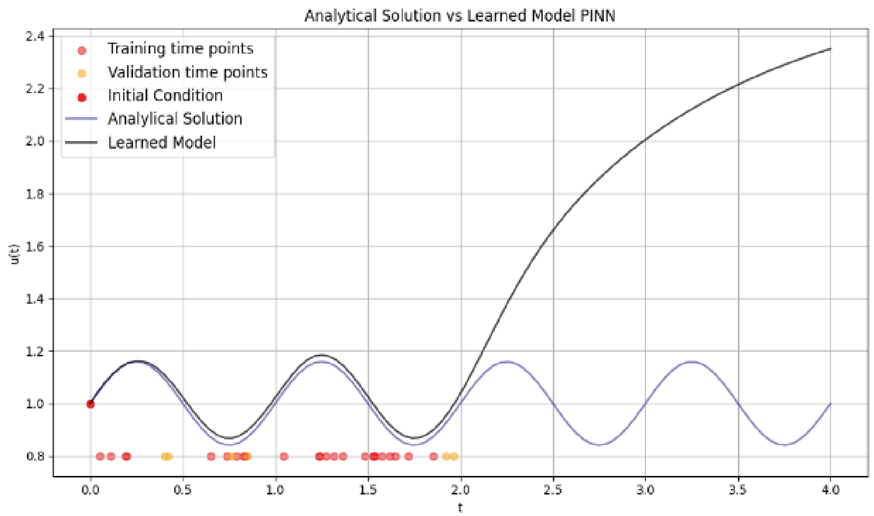




Physics-informed neural networks: PINN

- Disadvantages:
  - Difficult to train (loss composed of real and virtual data)
  - Tunning hyperparameters
  - Inference is not guaranteed (Tip: Generate more collocation points)





Physics-informed neural networks: PINN

- Disadvantages:
  - Difficult to train (loss composed of real and virtual data)
  - Tunning hyperparameters
  - Inference is not guarantee
  - Self supervised can return trivial solution to homogeneous PDE

$$m\frac{d^2x}{dt^2} + \mu\frac{dx}{dt} + kx = 0$$

Physics-informed neural networks: PINN

- Few recommendations:
  - Balance the loss terms $\mathcal{L}(\theta) = \lambda_{ic}\mathcal{L}_{ic}(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta) + \lambda_r\mathcal{L}_r(\theta),$

(c) Compute the global weights by

$$\hat{\lambda}_{ic} = \frac{\|\nabla_\theta \mathcal{L}_{ic}(\theta)\| + \|\nabla_\theta \mathcal{L}_{bc}(\theta)\| + \|\nabla_\theta \mathcal{L}_r(\theta)\|}{\|\nabla_\theta \mathcal{L}_{ic}(\theta)\|}, \tag{2.12}$$

$$\hat{\lambda}_{bc} = \frac{\|\nabla_\theta \mathcal{L}_{ic}(\theta)\| + \|\nabla_\theta \mathcal{L}_{bc}(\theta)\| + \|\nabla_\theta \mathcal{L}_r(\theta)\|}{\|\nabla_\theta \mathcal{L}_{bc}(\theta)\|}, \tag{2.13}$$

$$\hat{\lambda}_{r} = \frac{\|\nabla_\theta \mathcal{L}_{ic}(\theta)\| + \|\nabla_\theta \mathcal{L}_{bc}(\theta)\| + \|\nabla_\theta \mathcal{L}_r(\theta)\|}{\|\nabla_\theta \mathcal{L}_{r}(\theta)\|}, \tag{2.14}$$

where $\|\cdot\|$ denotes the L^2 norm.

(d) Update the global weights $\lambda = (\lambda_{ic}, \lambda_{bc}, \lambda_r)$ using a moving average of the form

$$\lambda_{\text{new}} = \alpha\lambda_{\text{old}} + (1-\alpha)\hat{\lambda}_{\text{new}}. \tag{2.15}$$

where the parameter α determines the balance between the old and new values

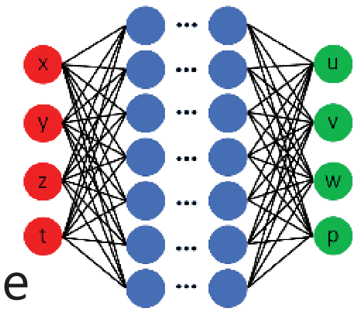
DOI: 10.48550/arXiv.2308.08468 • Corpus ID: 260925531

An Expert's Guide to Training Physics-informed Neural Networks

Sifan Wang, Shyam Sankaran, Hanwen Wang, P. Perdikaris less • Published in arXiv.org 16 August 2023 • Physics, Computer Science

Physics-informed neural networks: PINN

- Few recommendations:
  - Balance the loss terms $\mathcal{L}(\theta) = \lambda_{ic}\mathcal{L}_{ic}(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta) + \lambda_r\mathcal{L}_r(\theta),$
  - Non dimensionalization
    - Not traditional input normalization
    - Ensure the target output vary within a reasonable value
    - Transform the problem in an equivalent dimensionless problem



DOI: 10.48550/arXiv.2308.08468 • Corpus ID: 260925531

An Expert's Guide to Training Physics-informed Neural Networks

Sifan Wang, Shyam Sankaran, Hanwen Wang, P. Perdikaris less • Published in arXiv.org 16 August 2023 •
Physics, Computer Science

Physics-informed neural networks: PINN

- Few recommendations:
  - Balance the loss terms $\mathcal{L}(\theta) = \lambda_{ic}\mathcal{L}_{ic}(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta) + \lambda_r\mathcal{L}_r(\theta),$
  - Non dimensionalization
    - Not traditional input normalization
    - Ensure the target output vary within a reasonable value
    - Transform the problem in an equivalent dimensionless problem
  - Add specific information to the problem
    - Frequency domain

M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, R. Ng
Fourier features let networks learn high frequency functions in low dimensional domains
Adv Neural Inf Process Syst, 33 (2020), pp. 7537-7547

Acknowledges

sustAInable

https://eniachair.unizar.es

- Ministry of Digital Transformation and Public Function, through the call for ENIA 2022 Chairs, project TSI-100930-2023-1.
- ESI Group
- University of Zaragoza.

GOBIERNO DE ESPAÑA

España digital 2026

esi get it right

Universidad Zaragoza 1542