Physics-Informed Machine-Learning & Computer-Vision

Seminar Computer Vision by Deep Learning

Lecturer: Mahdi Naderi
m.naderibeni@tudelft.nl

26 Feb 2025

TUDelft

Intro

Mahdi Naderi

PhD student at the Pattern Recognition Lab

Interested in Physics-Informed Machine Learning:

ML for Fluid dynamics and Dynamical systems

Outline

- The quest for patterns
- “The right coordinate system” in ML
- Physics and ML
- Embedding Physics into ML
- Summary

Outline

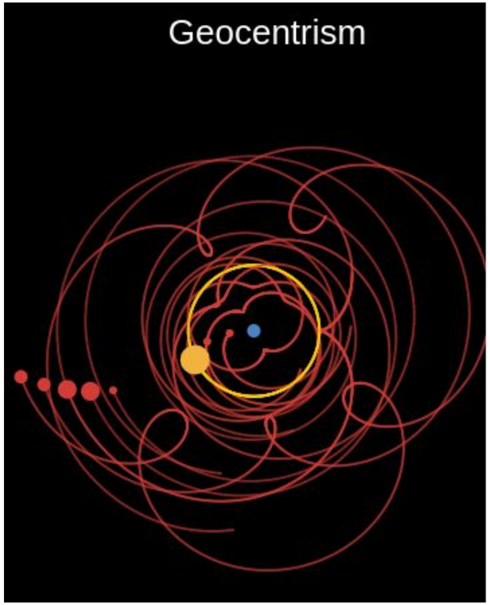
- The quest for patterns
- “The right coordinate system” in machine learning
- Physics and Machine Learning
- Embedding Physics into Machine Learning
- Summary

Finding the right coordinate system

Geocentric model of the Universe:

"Earth occupies the central position of the world system"

Data scientists at that time must have had difficult life finding patterns in the motion of celestial bodies!

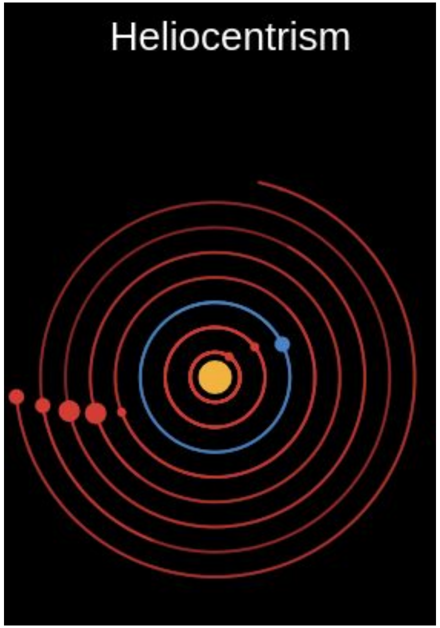


https://www.malinc.se/math/trigonometry/geocentrismen.php

Finding the right coordinate system

Heliocentric model of the Universe:

"Earth and planets orbit around the Sun at the center of the universe."



This coordinate transformation accelerated progress in finding celestial patterns.

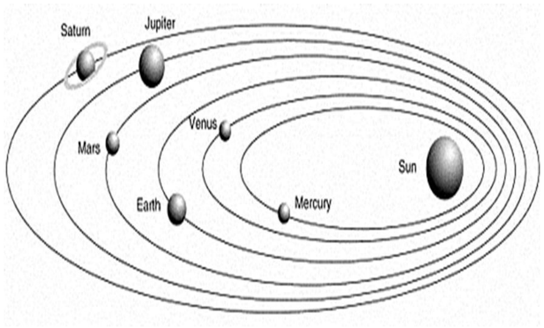
https://www.malinc.se/math/trigonometry/geocentrismen.php

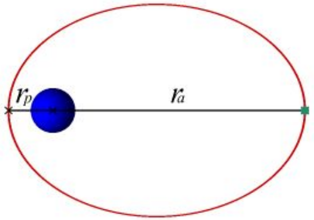
Finding the right coordinate system

Kepler's elliptical orbits

Kepler developed a data-driven model for planetary motion, resulting in his famous elliptic orbits.

However, this model did not explain the fundamental dynamic relationships that give rise to planetary orbits.





Small object in space orbits a larger object (such as a planet around the Sun) along an elliptical path.

Finding the right coordinate system

Newton's laws of motion

In contrast to his predecessors, Newton discovered the underlying relationship between the motion of an object and the forces acting on it.

$$\vec{F} = m\,\vec{a}$$

This described the processes responsible for the elliptic orbits and also many other phenomena.

Finding the right coordinate system

Newton's laws of motion

In contrast to his predecessors, Newton discovered the underlying relationship between the motion of an object and the forces acting on it.

This described the processes responsible for the elliptic orbits and also many other phenomena.

$$\vec{F} = m\,\vec{a}$$

Newton's Law of Universal Gravitation:

$$F = G\frac{m_1 m_2}{r^2}$$

Kinetic Energy:

$$K = \frac{1}{2}mv^2$$

Momentum

$$\vec{p} = m\vec{v}$$

Torque and Rotational Motion

$$\tau = I\alpha$$

Angular Momentum

$$\vec{L} = I\vec{\omega}$$

Vibrations and Oscillations

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

Fluid Dynamics and Continuum Mechanics

$$\rho\left(\frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u}\right) = -\nabla p + \mu\nabla^2\vec{u} + \vec{f}$$

...

TUDelft

Nowadays, how do we discover physical laws with Machine Learning?

For those who are interested:

Wang, H., Fu, T., Du, Y. et al. Scientific discovery in the age of artificial intelligence. Nature 620, 47–60 (2023). https://doi.org/10.1038/s41586-023-06221-2

S.L. Brunton, J.L. Proctor, & J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. U.S.A. 113 (15) 3932-3937, https://doi.org/10.1073/pnas.1517384113 (2016). https://arxiv.org/pdf/1509.03580

https://pysindy.readthedocs.io/en/latest/examples/2_introduction_to_sindy/example.html#

https://kks32-courses.github.io/sciml/README.html

Outline

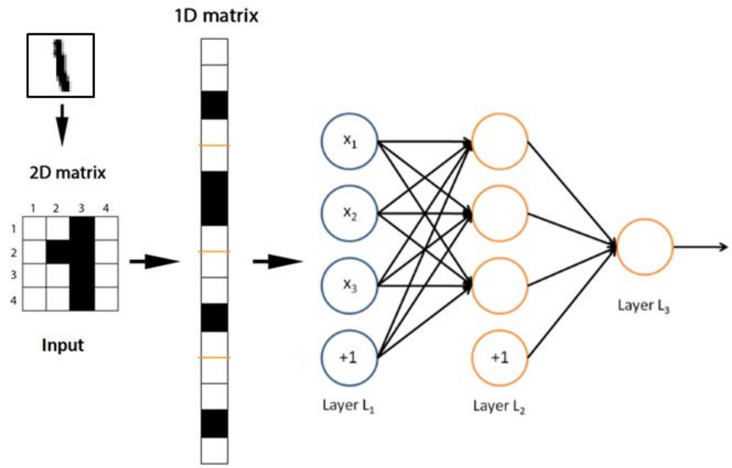
- The quest for patterns
- “The right coordinate system” in machine learning
- Physics and Machine Learning
- Embedding Physics into Machine Learning
- Summary

Machine Learning Stages

1. **Problem**
2. **Data**
3. **Model/Architecture**
4. **Loss Function**
5. **Optimization**

How to find “the right coordinate system” in a ML problem?

1. **Directly input the data features into the model**



2. **Use feature Maps**

   feature transformation (e.g., CNN feature maps, Fourier Transform, ...?)

3. **?**

What about the data itself? What is the best way of representing it?

TUDelft

Representation of the data

1. **As a discrete grid of intensities** (e.g., pixels,)

$$I = \sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}\mathbf{e}_i\mathbf{e}_j = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ I_{21} & I_{22} & \cdots & I_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{m1} & I_{m2} & \cdots & I_{mn} \end{bmatrix}, \quad I \in R^{m\times n}$$

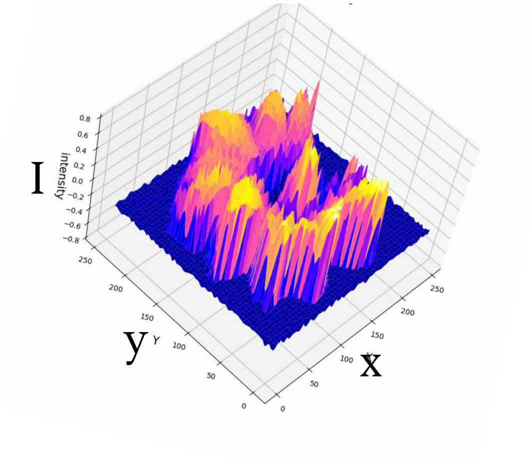
Representation of the data

1. **As a discrete grid of intensities (e.g., pixels,)**

$$I = \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \mathbf{e}_i \mathbf{e}_j = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ I_{21} & I_{22} & \cdots & I_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{m1} & I_{m2} & \cdots & I_{mn} \end{bmatrix}, \quad I \in R^{m \times n}$$

2. **Representing data as continuous functions**

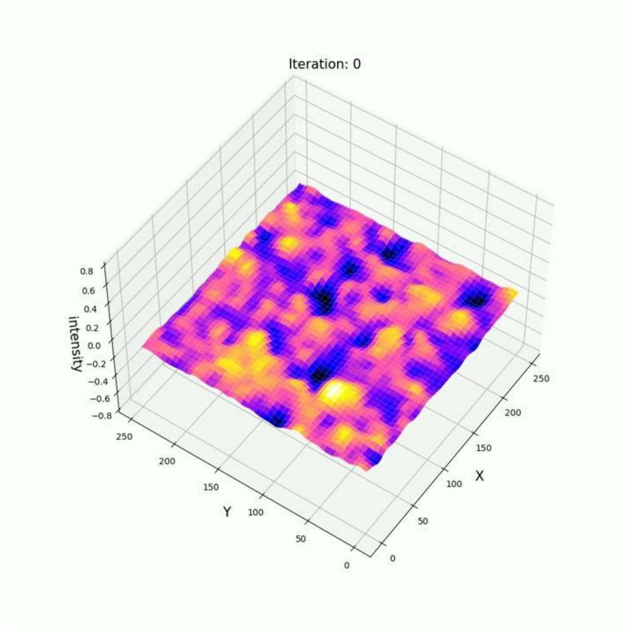
$$I = f(x, y), \quad f : R^2 \rightarrow R$$



Implicit Neural Representations (INRs)

Learning an image as a continuous 2D function

f_i :  $\longrightarrow$ $f_\theta(x_i)$:



$$\min_\theta \mathcal{L}(f_\theta, \{\mathbf{x}_i, \mathbf{f}_i\}_{i\in\mathcal{I}}) = \min_\theta \sum_{i\in\mathcal{I}} \|f_\theta(\mathbf{x}_i) - \mathbf{f}_i\|_2^2.$$

Use these representations for your ML tasks (e.g., classification, generative modeling, regression, …)

For those who are interested:

https://mnaderib.gtihub.io/files/Implicit_Neural_Representations.pdf

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, *33*, 7462-7473.
https://www.vincentsitzmann.com/siren/
https://www.youtube.com/watch?v=Or9J-DCDGko&list=PLat4GgaVK09e7aBNVlZelWWZIUzdq0RQ2&index=4&ab_channel=AndreasGeiger

Dupont, E., Kim, H., Eslami, S. M., Rezende, D., & Rosenbaum, D. (2022). From data to functa: Your data point is a function and you can treat it like one. arXiv preprint arXiv:2201.12204.
https://github.com/google-deepmind/functa

Outline

- The quest for patterns
- "The right coordinate system" in machine learning
- Physics and Machine Learning
- Embedding Physics into Machine Learning
- Summary

Physics is about gradients (rates of change)

Our approach in computing gradients determines our approach in embedding physics into Machine Learning.

as an example:

$$\overrightarrow{F} = m\,\overrightarrow{a}$$

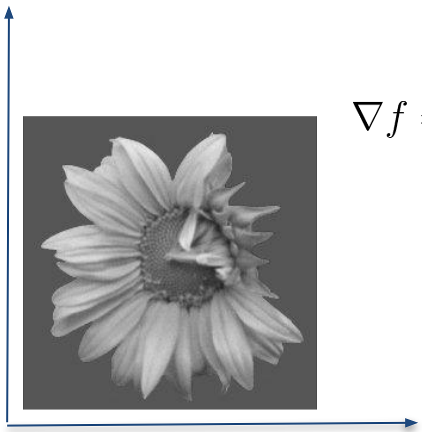
$$\mathit{Force} = \mathit{Mass} \cdot \mathit{Acceleration}$$

$$\mathit{Acceleration} = \frac{\mathrm{d}(\mathit{velocity})}{\mathrm{d}t}$$

$$\mathit{velocity} = \frac{\mathrm{d}(\mathit{position})}{\mathrm{d}t}$$

Computing gradients on image/videos

y



x

$f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$

$$\nabla f = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

Gradient Operator

?

$$\frac{\partial f}{\partial x}$$

?

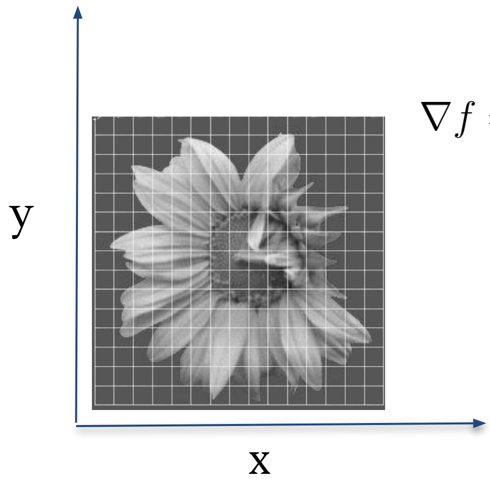
$$\frac{\partial f}{\partial y}$$

Computing gradients on image/videos

1. Numerical differentiation
   (Suitable for conventional grid based representations)

(e.g., finite differences)

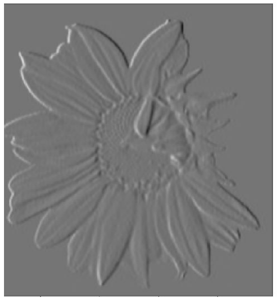
$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+h,y)-f(x,y)}{h}$$



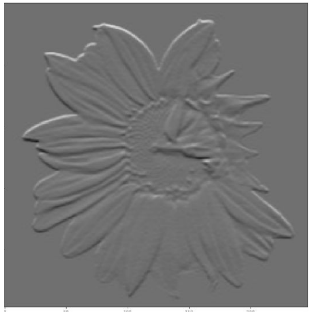
$$f(x,y) : \mathbb{R}^2 \to \mathbb{R}$$

$$\nabla f = \left\{ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\} : \mathbb{R}^2 \to \mathbb{R}^2$$

Gradient Operator



$$\frac{\partial f}{\partial x}$$



$$\frac{\partial f}{\partial y}$$

TU Delft

Computing gradients on image/videos

1. Numerical differentiation
   (Suitable for conventional grid based representations)

   (e.g., finite differences)

   $$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+h,y)-f(x,y)}{h}$$

2. Automatic Differentiation
   (Suitable for continuous representations of data)

   (using dual numbers)

3. ?

Expected Physics in Computer Vision

e.g.,

- **Geometric Consistency**
  (Principles of perspective projection)

- **Photometric Invariance**
  (The law of light reflection, shading, and it's interaction with surfaces)

- **Temporal and Kinematic Constraints**

  (Consistency and basic laws of motion

- **Boundary and Material Constraints**

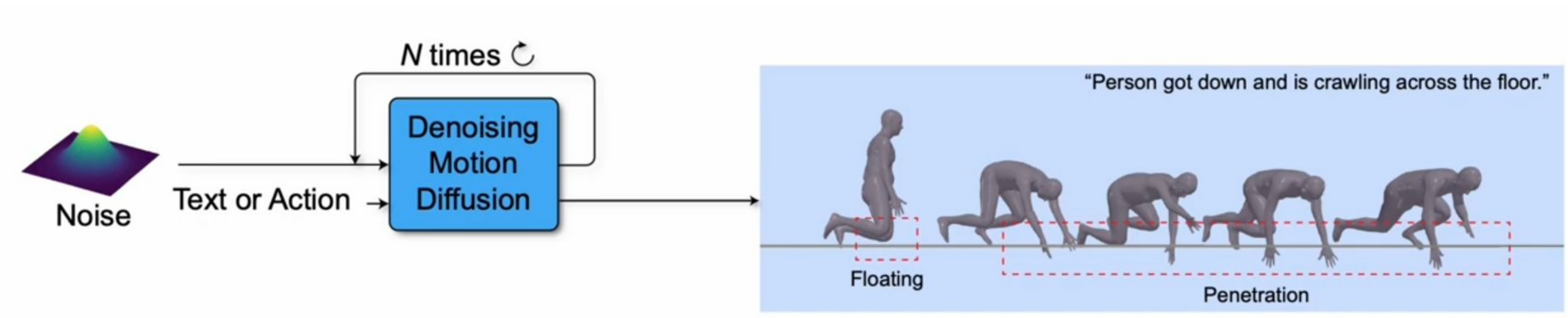
  Natural boundaries, (e.g., in image segmentation or reconstruction)

- **?**

TUDelft

Example: generative modeling of human motion

- Even simple physical constraints (e.g., rigid impenetrable ground) get violated with current generative models



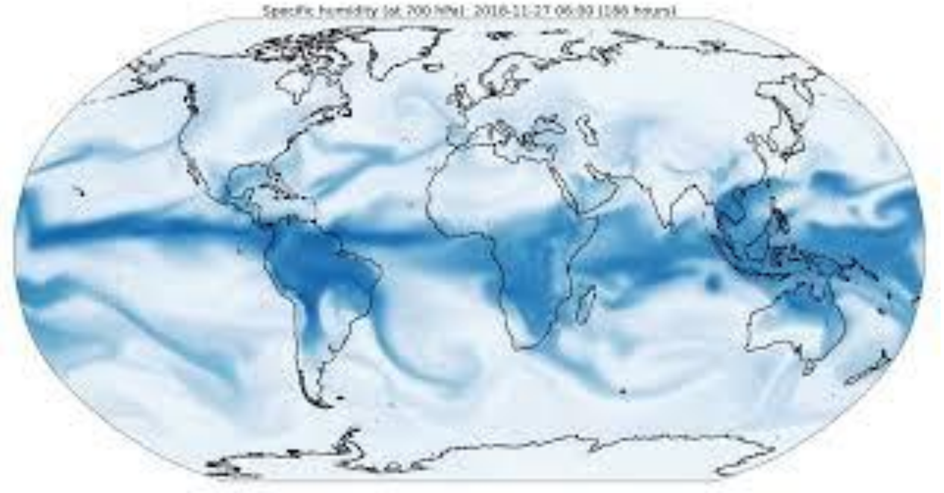
https://nvlabs.github.io/PhysDiff/

Example: Weather Forecasting

- There is tremendous amounts of weather data available from local and satellite measurements

- Climate is a very complex dynamical system
  (The Butterfly effect,
  https://en.wikipedia.org/wiki/Butterfly_effect)

- Modern weather models use atmospheric physics together with historical data.



TU Delft

Outline

- The quest for patterns
- "The right coordinate system" in machine learning
- Physics and Machine Learning
- Embedding Physics into Machine Learning
- Summary

Machine Learning Stages

1. **Problem**
2. **Data**
3. **Model/Architecture**
4. **Loss Function**
5. **Optimization**

How to embed physics into

1. Problem
2. **Data**
3. Model/Architecture
4. Loss Function
5. Optimization

- Use the right coordinates to represent your data
- If the system is translationally or rotationally invariant, augment dataset with rotated and translated copies.
- ?

Are these enough to make a model that obeys physics?

TU Delft

How to embed physics into

1. Problem
2. **Data**
3. Model/Architecture
4. Loss Function
5. Optimization

OpenAI's Sora https://openai.com/index/video-generation-models-as-world-simulators/ *February 15, 2024*

We explore large-scale training of generative models on video data. Specifically, we train text-conditional diffusion models jointly on videos and images of variable durations, resolutions and aspect ratios. We leverage a transformer architecture that operates on spacetime patches of video and image latent codes. Our largest model, Sora, is capable of generating a minute of high fidelity video. Our results suggest that scaling video generation models is a promising path towards building general purpose simulators of the physical world.

How strong is this argument?

How to embed physics into

1. Problem
2. **Data**
3. Model/Architecture
4. Loss Function
5. Optimization

How does Open AI's text-to-video model considers physics in generating videos?
Prompt: *A flock of paper airplanes flutters through a dense jungle, weaving around trees as if they were migrating birds.*

Do some paper planes go missing? What are the possible causes?

A technical report is not published, however you can read this:

https://arxiv.org/pdf/2402.17177

TU Delft

How to embed physics into

1. Problem
2. **Data**
3. Model/Architecture
4. Loss Function
5. Optimization

What are the limitations?

For those who are interested:
Benton, G., Finzi, M., Izmailov, P., & Wilson, A. G. (2020). Learning invariances in neural networks from training data. *Advances in neural information processing systems, 33*, 17605-17616.

How to embed physics into

1. Problem
2. Data
3. **<u>Model/Architecture</u>**
4. Loss Function
5. Optimization

Designing architectures that are physically consistent is very rewarding but also very difficult.

e.g.: Build classification models that are rotation or translation invariant.
does this sound familiar?

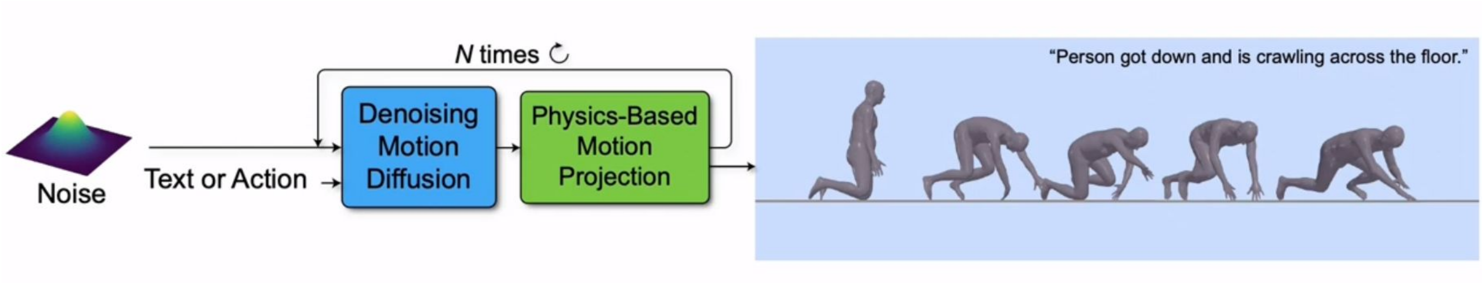
Advantages?

What about tasks like regression or generative modeling? How to design models that output predictions that are physically consistent?

TU Delft

Example: generative modeling of human motion

Combining the diffusion step with a physics simulator can make huge difference



Yuan, Y., Song, J., Iqbal, U., Vahdat, A., & Kautz, J. (2023). Physdiff: Physics-guided human motion diffusion model. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 16010-16021). https://arxiv.org/pdf/2212.02500

TU Delft

How to embed physics into

1. Problem
2. Data
3. **Model/Architecture**
4. Loss Function
5. Optimization

For those who are interested:

Cohen, T., & Welling, M. (2016, June). Group equivariant convolutional networks. In International conference on machine learning (pp. 2990-2999). PMLR. https://arxiv.org/pdf/1602.07576

Richter-Powell, J., Lipman, Y., & Chen, R. T. (2022). Neural conservation laws: A divergence-free perspective. Advances in Neural Information Processing Systems, 35, 38075-38088. https://openreview.net/pdf?id=prQkA_NjuuB

Greydanus, S., Dzamba, M., & Yosinski, J. (2019). Hamiltonian neural networks. Advances in neural information processing systems, 32. https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf

How to embed physics into

1. Problem
2. Data
3. Model/Architecture
4. **Loss Function**
5. Optimization

- Add additional term to the loss function as a physical regularizer

$$\mathcal{L}(\theta) = \mathcal{L}_{data}(\theta) + \mathcal{L}_{physics}(\theta)$$

Very interesting topic in the field of scientific machine learning:

Physics-Informed Neural Networks.

- On the labeled samples you can minimize both terms of the loss function.
- On the unlabeled samples, you can minimize the physics loss term
(Generate these samples if possible)

Example: Weather Forecasting

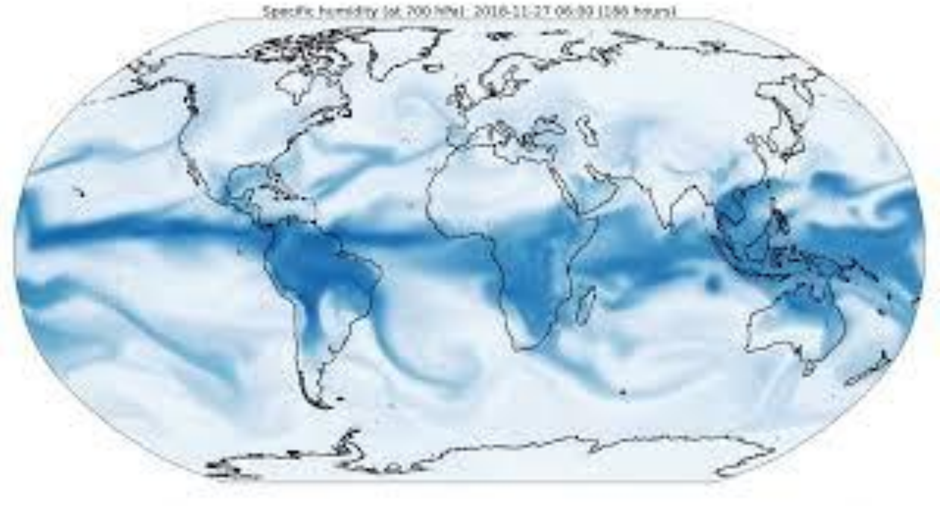
$$\mathcal{L}(\theta) = \mathcal{L}_{data}(\theta) + \mathcal{L}_{physics}(\theta)$$

$\mathcal{L}_{data}(\theta)$: Error in prediction of

- Temperature
- Wind speed
- Humidity

$\mathcal{L}_{physics}(\theta)$: Error in satisfying the related physical laws

- Conservation of Mass
- Conservation of Energy
- Conservation of Momentum



GraphCast: AI model for faster and more accurate global weather forecasting: https://deepmind.google/discover/blog/graphcast-ai-model-for-faster-and-more-accurate-global-weather-forecasting/

TU Delft

How to embed physics into

1. Problem
2. Data
3. Model/Architecture
4. Loss Function
5. **Optimization**

Constrained optimization problem:

$$\min_{\theta} \sum_{i=1}^{N} \|f(\mathbf{x}_i;\theta) - \hat{f}_i\|^2$$

$$s.t.\ Physical\ constraint, \forall x \in R$$

Is this different than having additional term in the loss function as physics loss?

$$\mathcal{L}(\theta) = \mathcal{L}_{data}(\theta) + \mathcal{L}_{physics}(\theta)$$

Here the aim is to enforce this term to be zero rather than minimizing it:

$$\mathcal{L}_{physics}(\theta) = 0$$

TUDelft

How to embed physics into

1. Problem
2. Data
3. Model/Architecture
4. Loss Function
5. **Optimization**

For those who are interested:
Otto, S. E., Zolman, N., Kutz, J. N., & Brunton, S. L. (2023). A unified framework to enforce, discover, and promote symmetry in machine learning. arXiv preprint https://arxiv.org/pdf/2311.00212

Balestriero, R., & LeCun, Y. (2023, June). POLICE: Provably optimal linear constraint enforcement for deep neural networks. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE. https://arxiv.org/abs/2211.01340

Machine Learning Stages

1. **Problem**
2. **Data**
3. **Model/Architecture**
4. **Loss Function**
5. **Optimization**

TUDelft

Outline

- The quest for patterns
- "The right coordinate system" in machine learning
- Physics and Machine Learning
- Embedding Physics into Machine Learning
- Summary

Summary

1. Finding the right coordinate system can simplify the learning task

2. Symmetries or physical constraints can be embedded at every stage of ML
   (Data, Architecture, Loss Function, & optimization algorithm)

Choose your weapon! (reference #3)

As a machine learning researcher, if you don't have enough computational resources to assume that the data will bring the symmetries/physics, search for better ways to embed symmetries/physics into machine learning!

Read more:

1. Lectures on Physics Informed Machine Learning: by Steve Brunton: https://youtu.be/JoFW2uSd3Uo?feature=shared
2. Gradient-domain image processing: http://graphics.cs.cmu.edu/courses/15-463/2019_fall/lectures/lecture9.pdf
3. Togelius, J., & Yannakakis, G. N. (2023). Choose your weapon: Survival strategies for depressed AI academics. arXiv preprint arXiv:2304.06035. https://arxiv.org/pdf/2304.06035v2