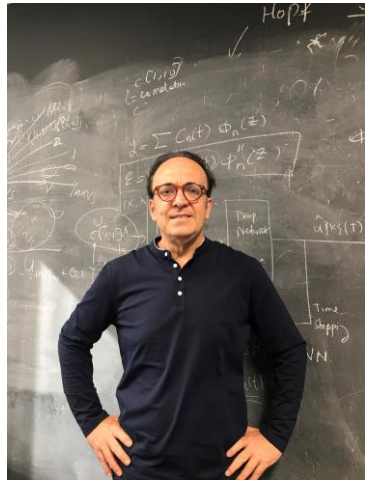
Deep Learning for Science and Engineering Teaching Kit

Deep Learning for Scientists and Engineers

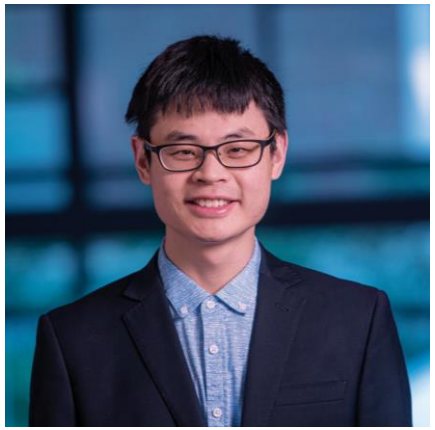
Lecture 1: Introduction to Deep Learning

Instructors: George Em Karniadakis, Khemraj Shukla, Lu Lu
Teaching Assistants: Vivek Oommen and Aniruddha Bora

Instructors and Teaching Assistants



George Em Karniadakis
The Charles Pitts Robinson & John Palmer Barstow Professor of Applied Mathematics and Engineering, Brown University



Lu Lu
Assistant Professor of Chemical & Biomolecular Engineering
University of Pennsylvania

Khemraj Shukla
Assistant Professor, Division of Applied Mathematics
Brown University

Vivek Oommen
Graduate Student,
School of Engineering, Brown University

Aniruddha Bora
Postdoctoral Research Associate
Division of Applied Mathematics, Brown University

Course Roadmap

Module-1 (Basics)

- Lecture 1: Introduction
- Lecture 2: A primer on Python, NumPy, SciPy and *jupyter* notebooks
- Lecture 3: Deep Learning Networks
- Lecture 4: A primer on TensorFlow and PyTorch
- Lecture 5: Training and Optimization
- Lecture 6: Neural Network Architectures

Module-2 (PDEs and Operators)

- Lecture 7: Machine Learning using Multi-Fidelity Data
- Lecture 8: Physics-Informed Neural Networks (PINNs)
- Lecture 9: PINN Extensions
- Lecture 10: Neural Operators

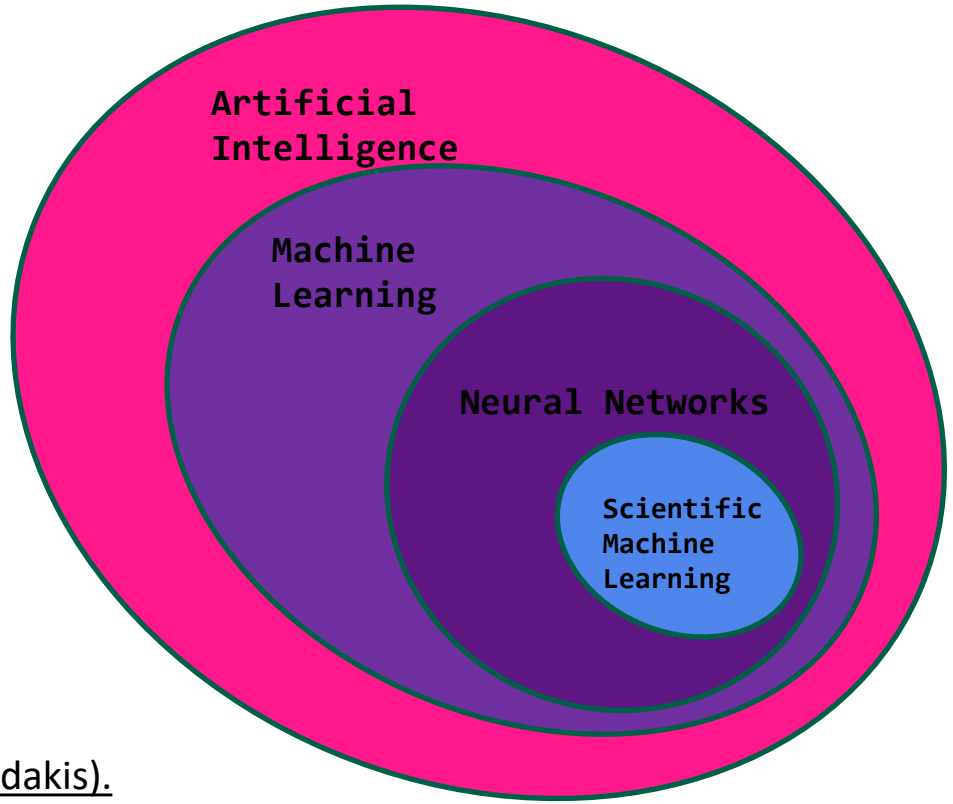
Module-3 (Codes & Scalability)

- Lecture 11: Multi-GPU Scientific Machine Learning

Contents

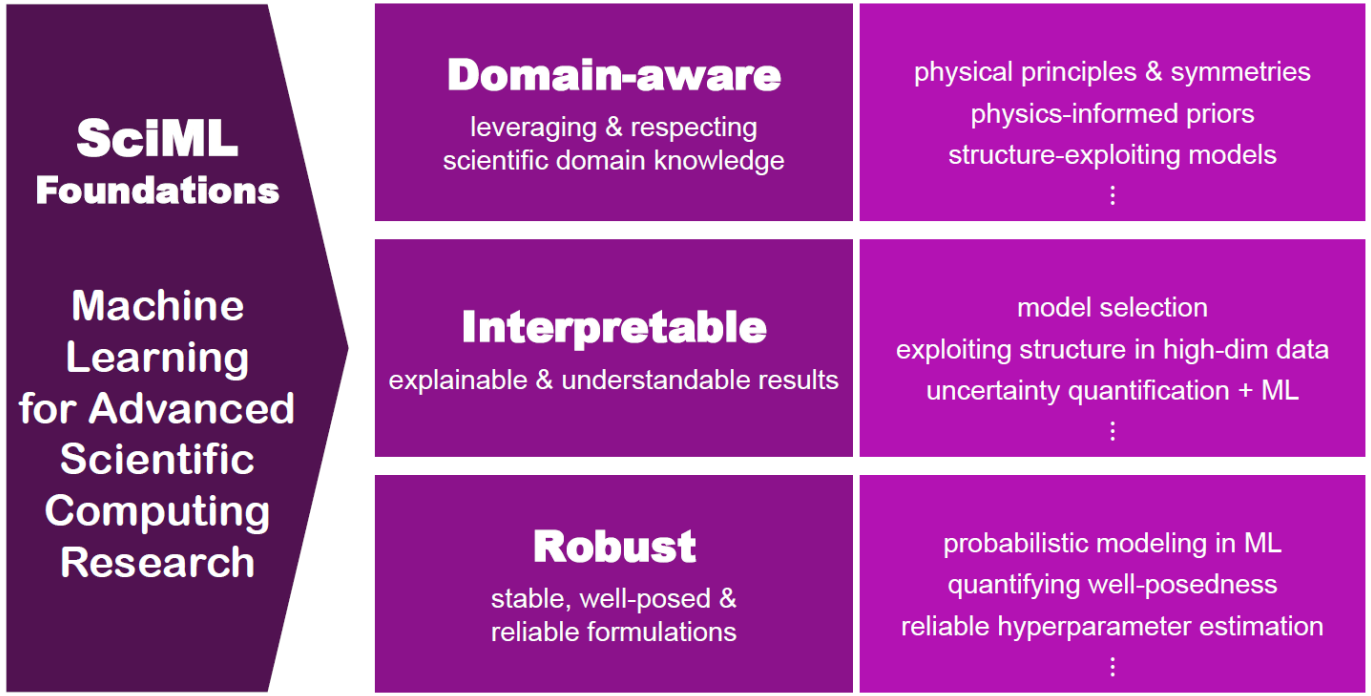
https://www.import.io/post/history-of-deep-learning/

- Artificial intelligence (AI) > Machine Learning (ML) > Deep Learning > Scientific Machine Learning (SciML).
- The expression "Deep Learning" was (probably) first used by Igor Aizenberg and colleagues around 2000.
- 1960s: Shallow Neural Networks.
- 1982: Hopfield Network – A Recurrent NN.
- 1988-89: Learning by backpropagation, Rumelhart, Hinton & Williams; hand-written text, LeCun.
- 1993 NVIDIA was founded; GeForce is the first GPU.
- 1990s Unsupervised Deep Learning.
- 1993: A Recurrent NN with 1,000 layers (Jürgen Schmidhuber)
- 1994: NN for solving PDEs, Dissanayake & Phan-Thien
- 1998: Gradient-based learning, LeCun.
- 1998: ANN for solving ODEs&PDEs, Lagaris, Likas & Fotiadis
- 1990-2000: Supervised Deep Learning.
- 2006: A fast learning algorithm for deep belief nets, Hinton.
- 2006-present: Modern Deep Learning.
- 2009: ImageNet: A large-scale hierarchical image database (Fei-Fei).
- 2010: GPUs are only up to 14 times faster than CPUs (Intel).
- 2010: Tackling the vanishing/exploding gradients: Glorot & Bengio.
- 2011: AlexNet – Convolutional NN (CNN) - Alex Krizhevsky.
- 2014: Generative Adversarial Networks (GANs) – Ian Goodfellow.
- 2015: Batch normalization, Ioffe & Szegedy.
- 2017: PINNs: Physics-Informed Neural Networks (Raissi, Perdikaris, Karniadakis).
- 2019: Scientific Machine Learning (ICERM workshop Jan. 2019; DOE report, Feb 2019).
- 2019: DeepONet – Operator regression (Lu, Jin, Karniadakis).



Basic Research Needs Workshop for Scientific Machine Learning
Core Technologies for Artificial Intelligence, DOE ASCR Report, Feb 2019

❑ Scientific machine learning (**SciML**) is a core component of artificial intelligence (AI) and a computational technology that can be trained, with scientific data, to augment or automate human skills.



❑ SciML must achieve the same level of scientific rigor expected of established methods deployed in science and applied mathematics. Basic requirements include validation and limits on inputs and context implicit in such validations, as well as verification of the basic algorithms to ensure they are capable of delivering known prototypical solutions.

❑ Can SciML achieve Robustness?

- NSF/ICERM Workshop on SciML, January 28-30, 2019 (Organizers: J. Hesthaven & G.E. Karniadakis)
- https://icerm.brown.edu/events/ht19-1-sml/)

1958 @ Cornell – Rosenblatt's Perceptron

➢ **Professor Frank Rosenblatt's perceptron paved the way for AI – 60 years too soon!**

❑ An IBM 704 – a 5-ton computer the size of a room – was fed a series of punch cards. After 50 trials, the computer taught itself to distinguish cards marked on the left from cards marked on the right.

- An image of the perceptron from Rosenblatt's "The Design of an Intelligent Automaton," Summer 1958.

FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

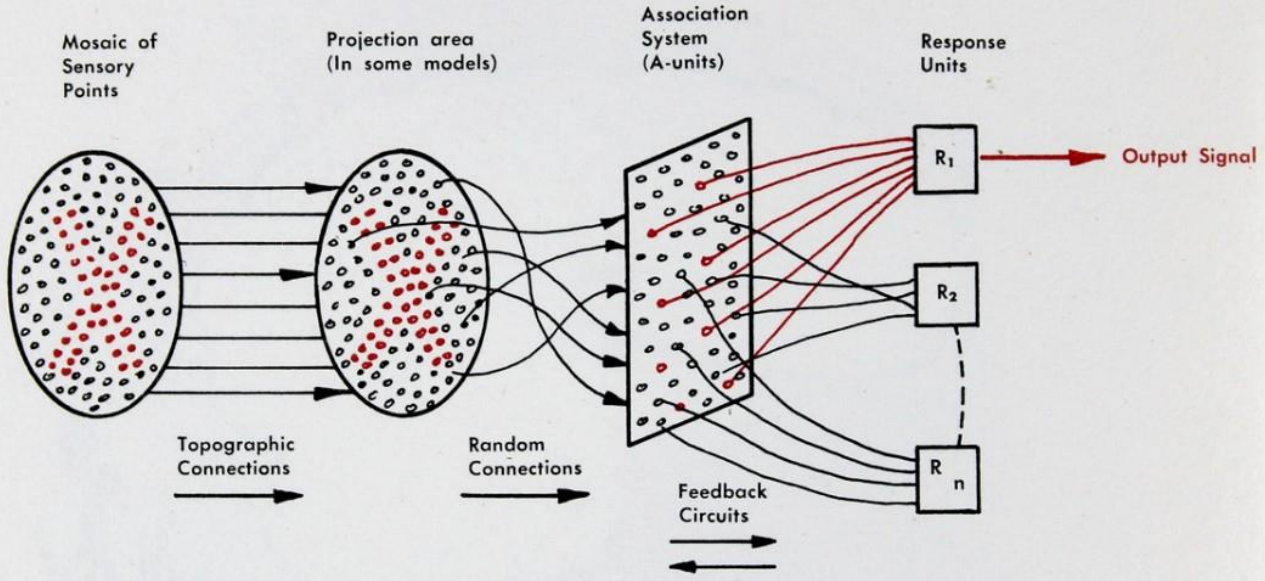
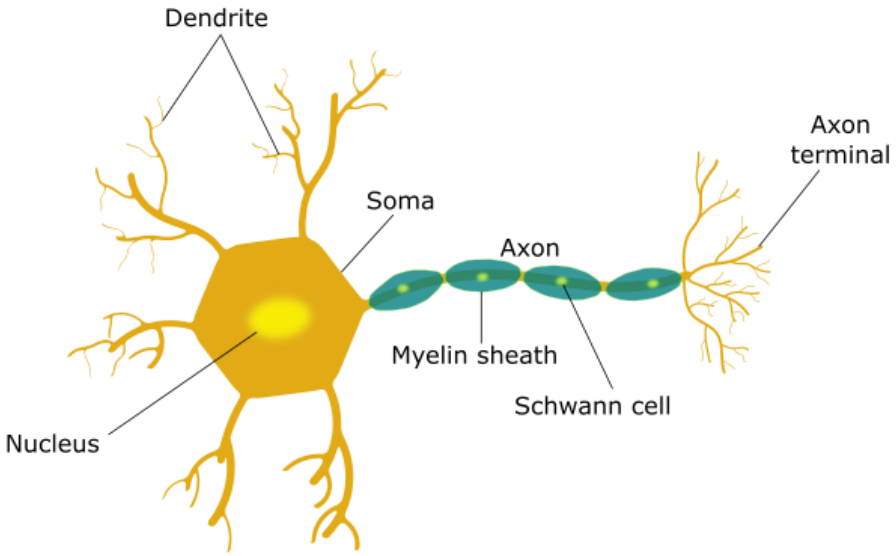
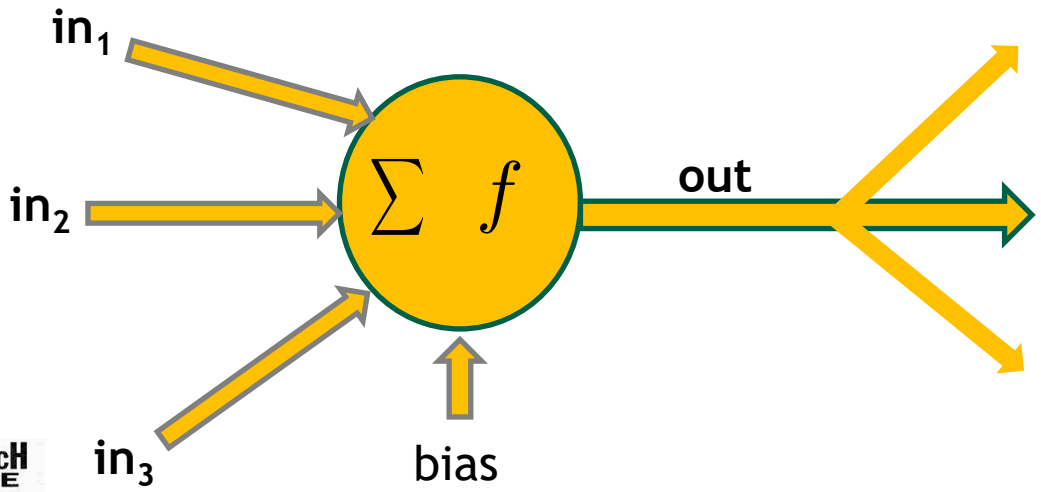


FIG. 2 — Organization of a perceptron.

https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Differences between Artificial and Biological Neural Networks





- **Size**: our brain contains about 86 billion neurons and more than 100-1000 trillion synapses. GPT3 has 175 billion parameters, and the Chinese Wu Dao has 1.75 trillion parameters.

- **Major Differences (in addition to size)**
  - Topology
  - Speed
  - Fault-tolerance
  - Power consumption
  - Signals
  - Learning

https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

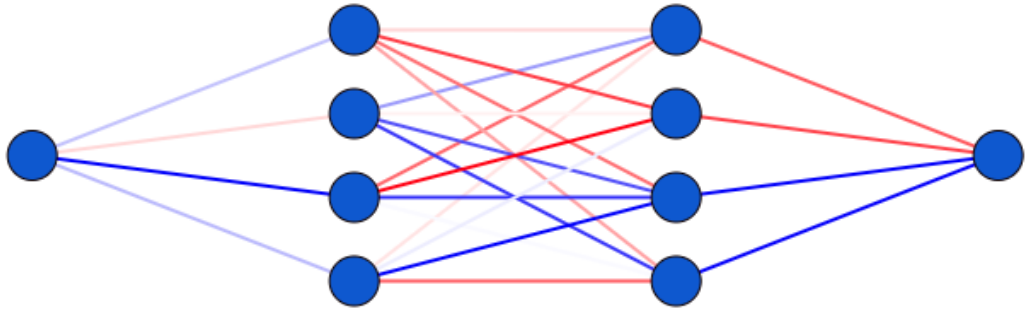
A PyTorch script to create a feed-forward fully-connected NN

```
def feed_forward(input_dim, output_dim, num_neurons, num_hidden_layers):
    import torch.nn as nn
    net=nn.Sequential()
    net.add_module(nn.Linear(input_dim, num_neurons))
    net.add_module(nn.Tanh())
    for num in range(num_hidden_layers):
        net.add_module(nn.Linear(num_neurons , num_neurons ))
        net.add_module(nn.Tanh())
    net.add_module(nn.Linear(num_neurons , output_dim))
    return net
```

- Change the number of layers
- Change the number of neurons
- Change the number of inputs
- Change the number of outputs

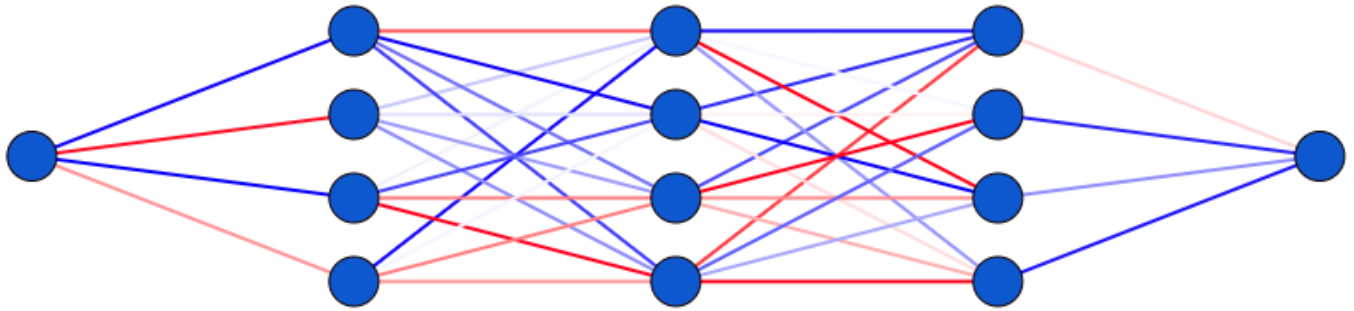
- To plot use: http://alexlenail.me/NN-SVG/index.html

```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=4, num_hidden_layers=2)
```



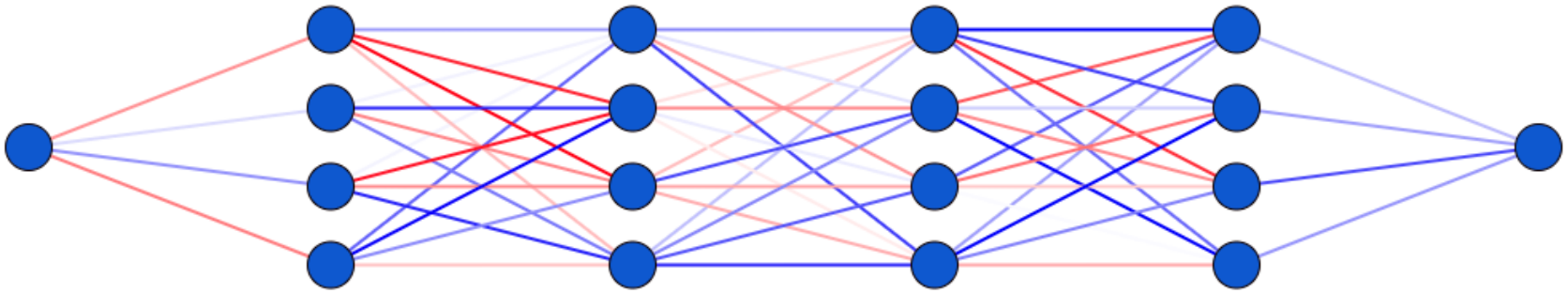
Depth=2

```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=4, num_hidden_layers=3)
```



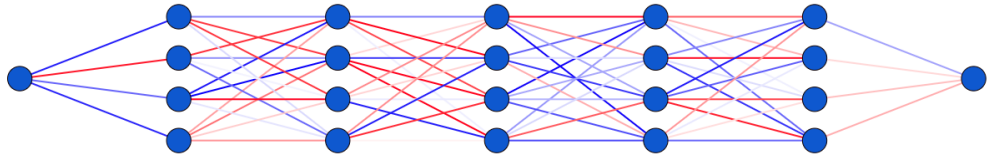
Depth=3

```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=4, num_hidden_layers=4)
```



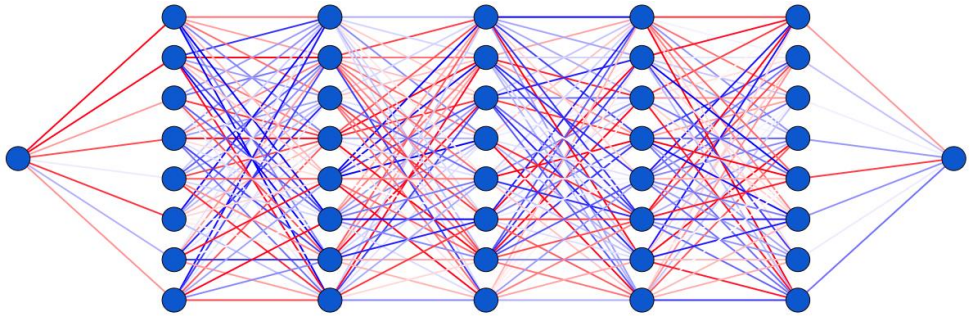
Depth=4

```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=4, num_hidden_layers=5)
```



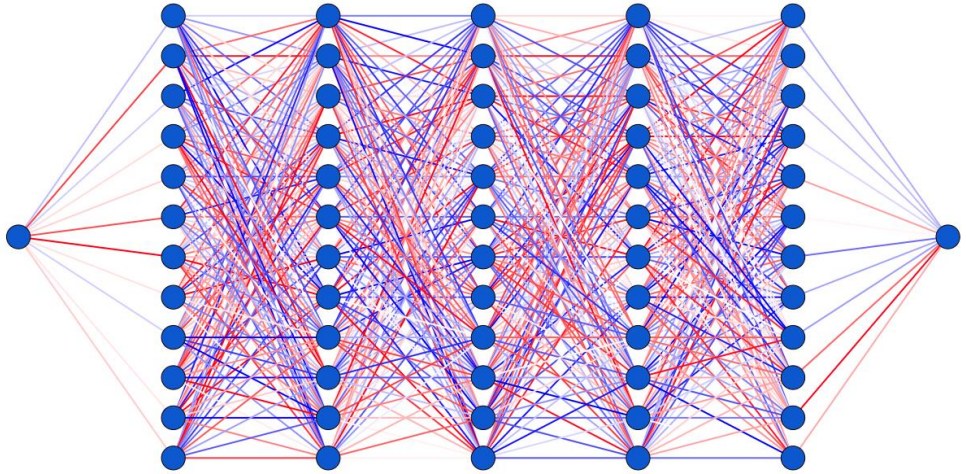
Width=4

```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=8, num_hidden_layers=5)
```



Width=8

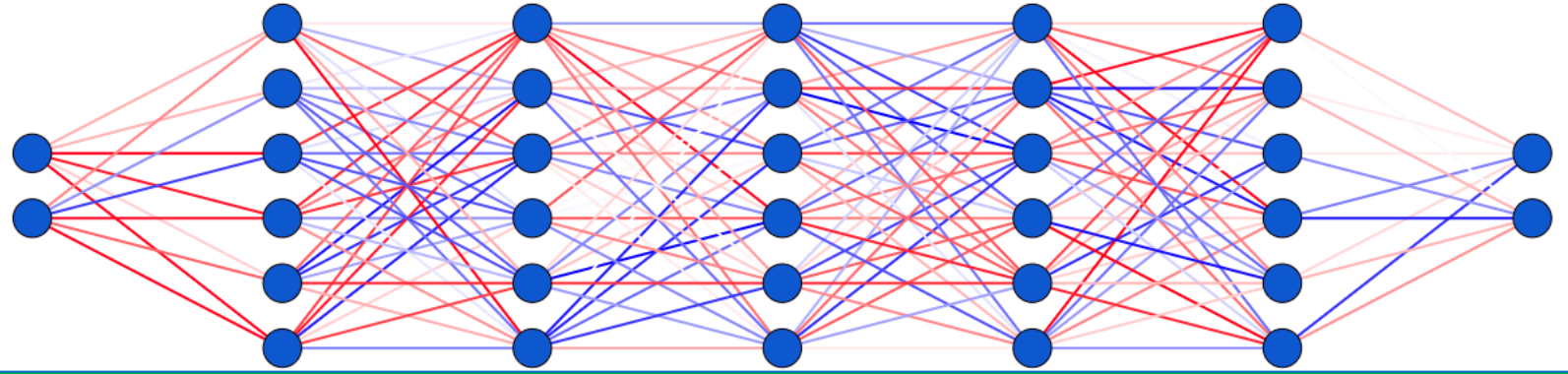
```
net = feed_forward(input_dim=1, output_dim=1, num_neurons=12, num_hidden_layers=5)
```



Width=12

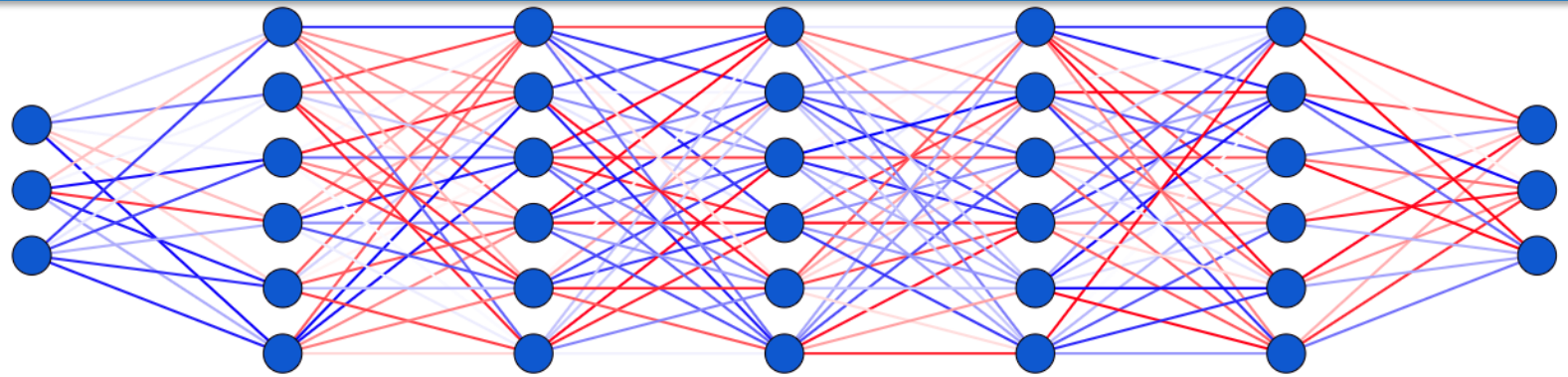
```
net = feed_forward(input_dim=2, output_dim=2, num_neurons=6, num_hidden_layers=5)
```

$[d_{in}, d_{op}]=[2,2]$



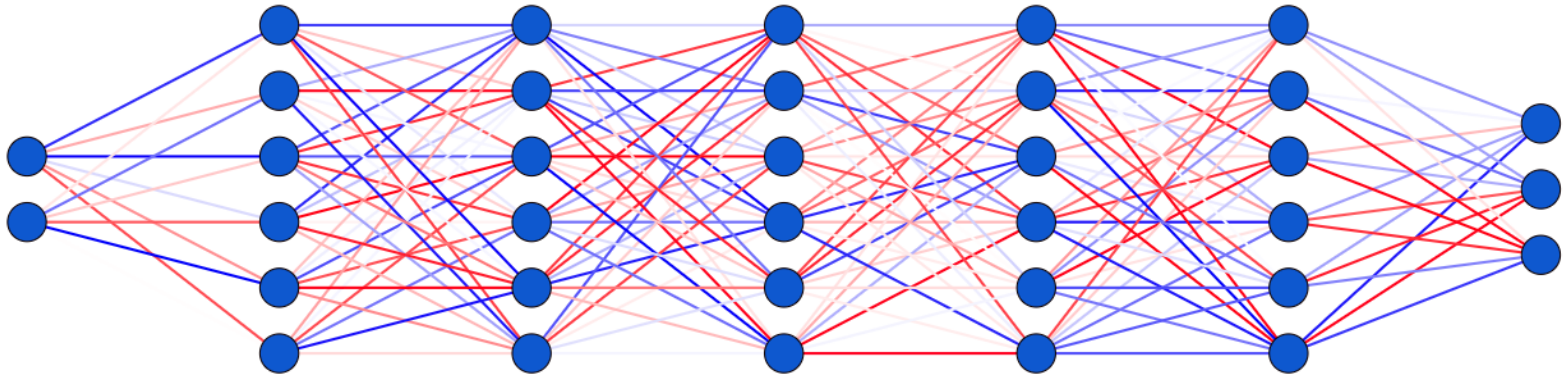
```
net = feed_forward(input_dim=3, output_dim=3, num_neurons=6, num_hidden_layers=5)
```

$[d_{in}, d_{op}]=[3,3]$



```
net = feed_forward(input_dim=2, output_dim=3, num_neurons=6, num_hidden_layers=5)
```

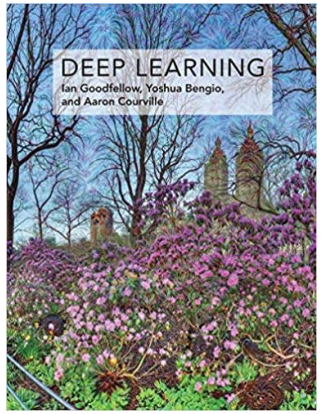
$[d_{in}, d_{op}]=[2,3]$

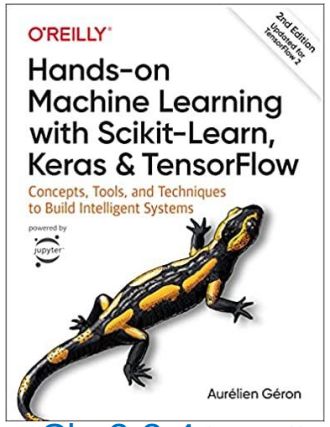


Course Objectives

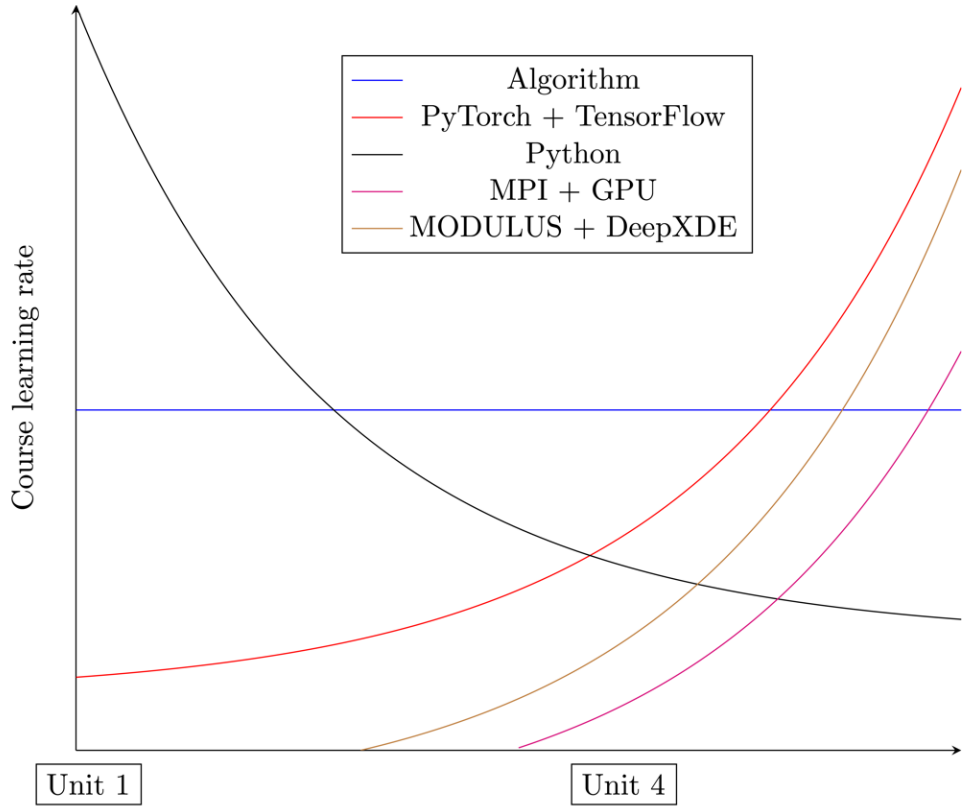
At the end of the course participants will be able to:

- ❑ Understand the underlying theory and mathematics of deep learning.
- ❑ Analyze and synthesize data in order to model physical, chemical, biological, and engineering systems.
- ❑ Apply physics-informed neural networks (PINNs) and neural operators to model and simulate multiphysics systems in science and engineering.
- ❑ Be fluent in python, Tensorflow, and PyTorch.
- ❑ Be fluent in multi-GPU computing.





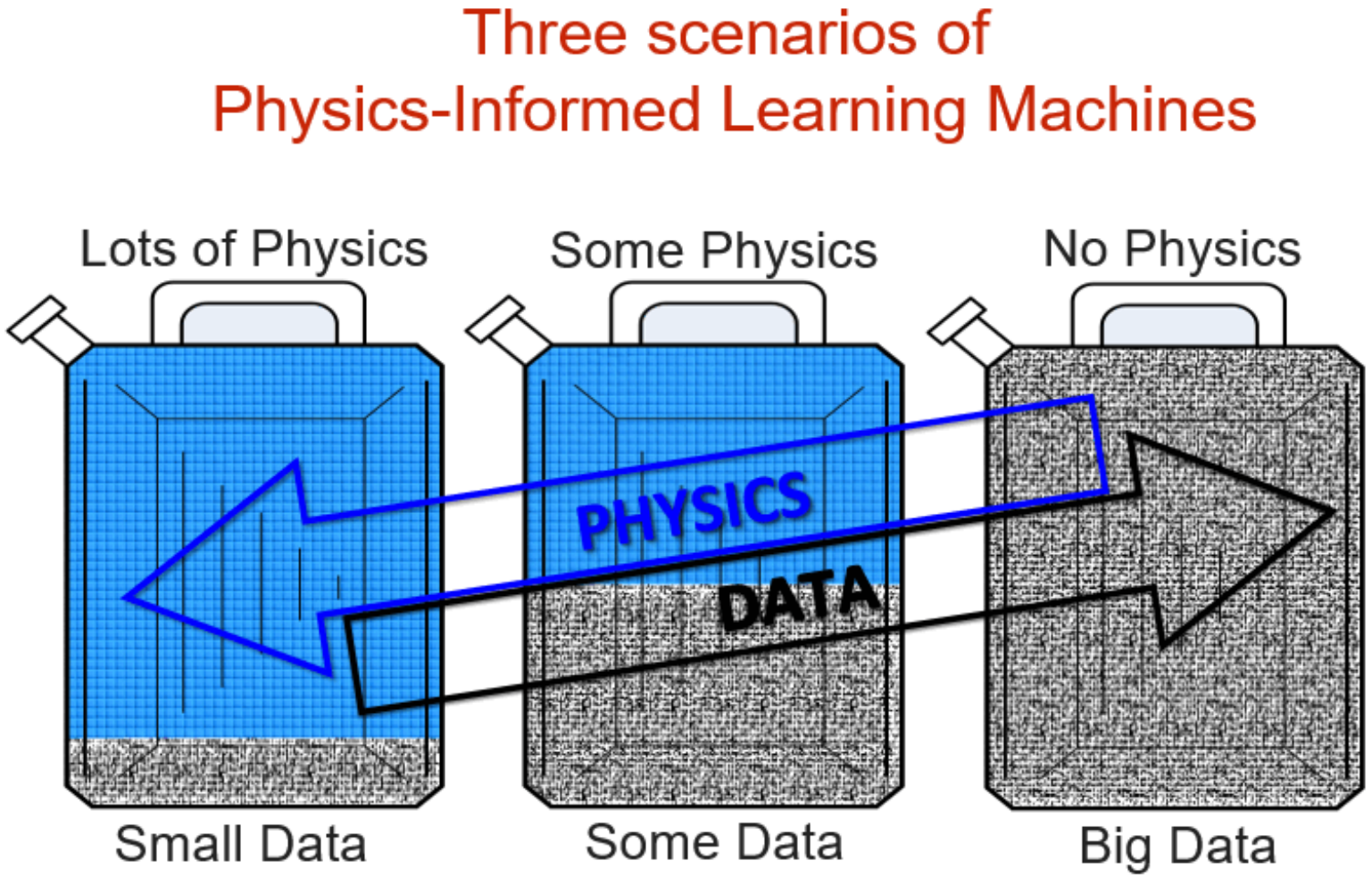
Course Teaching/Learning Rates



1. Goodfellow et al: Prerequisites: Ch. 2-3-4 https://www.deeplearningbook.org/

2. A. Geron: Ch. 4 & 11, O;Reily (2019).

Data + Physical Laws



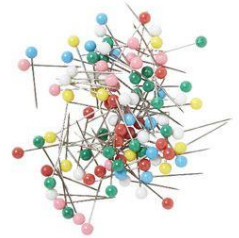
❑ **The 5D Law:**
Dinky, Dirty, Dynamic, Deceptive Data

PINN
- **DeepXDE**
- **Modulus**

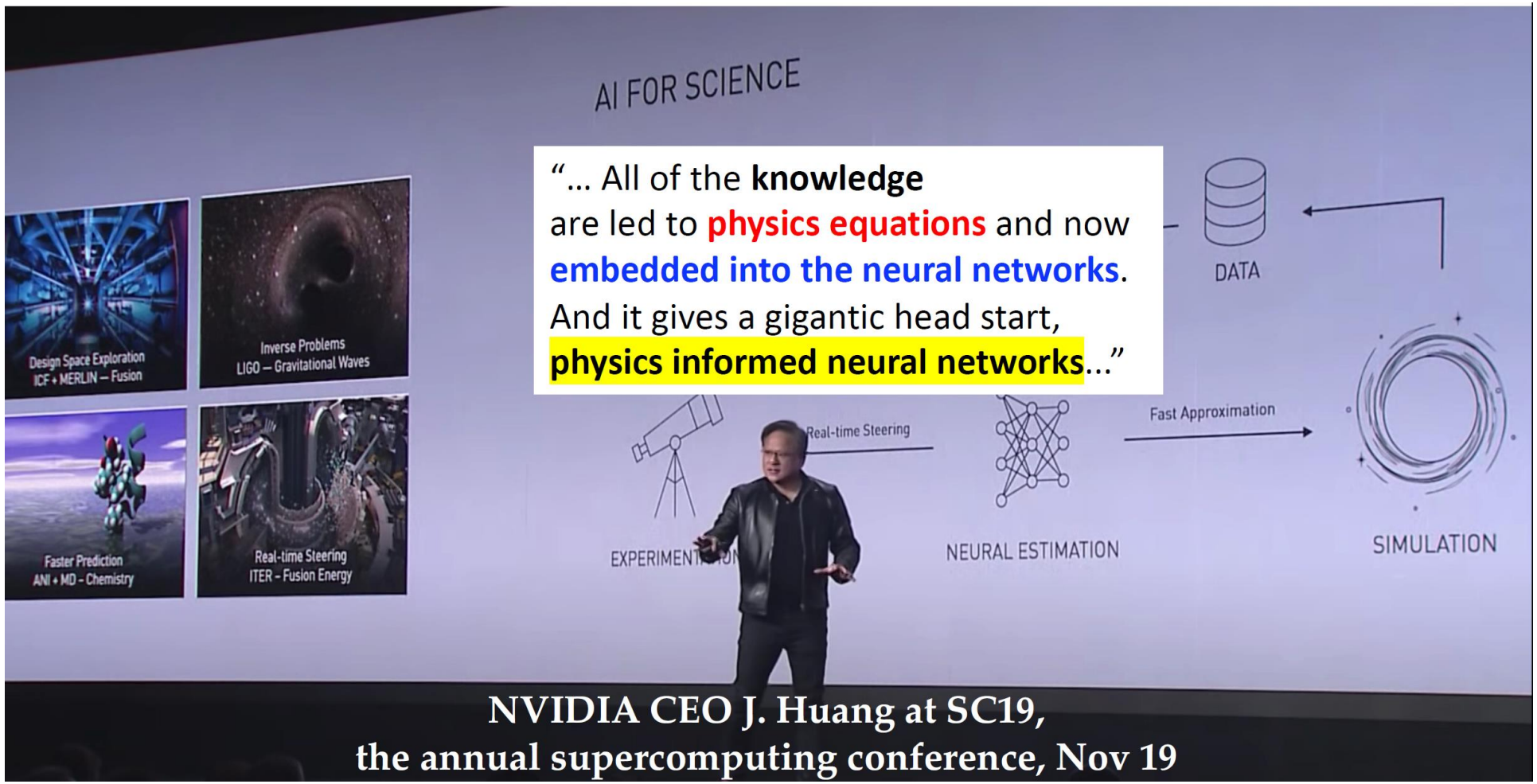
DeepONet
- **DeepXDE**

FNO
- **Modulus**

PINNs: Physics-Informed Neural Networks



https://www.youtube.com/watch?v=69nEEpdEJzU

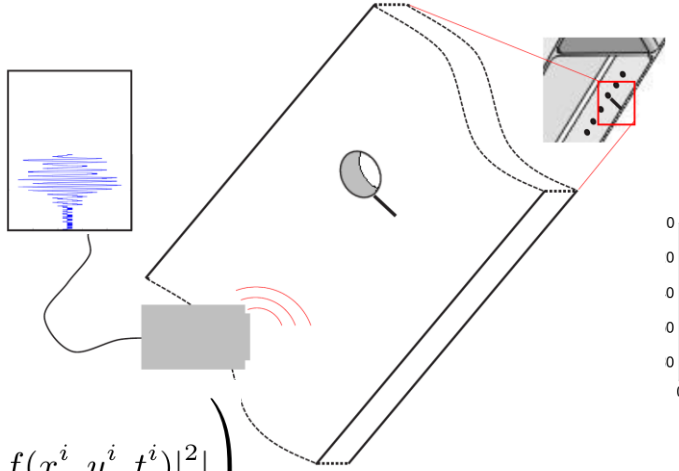


Reference: Raissi, Perdikaris & Karniadakis, arXiv 2017; patent 2020

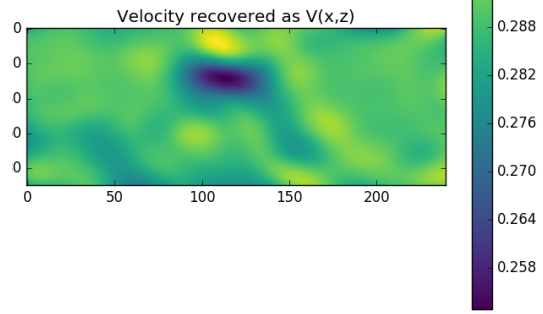
Ultra-Sound Testing of Materials (WPAFB Real Data)

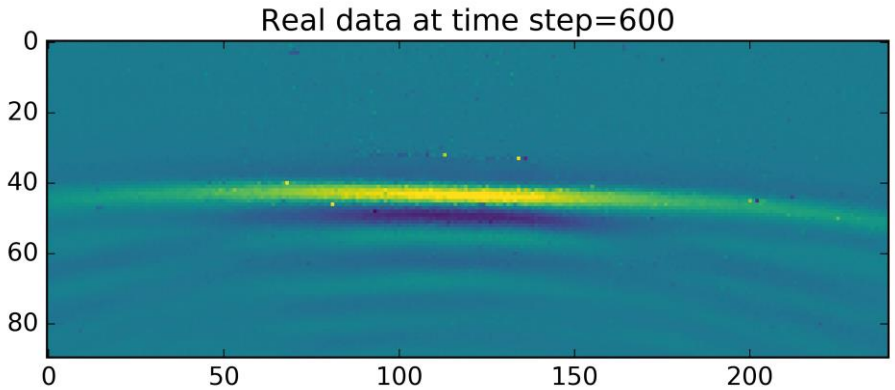
$$f := \frac{\partial^2 \Psi_{\mathrm{NN}}}{\partial t^2} - c^2 \frac{\partial^2 \Psi_{\mathrm{NN}}}{\partial x^2}$$

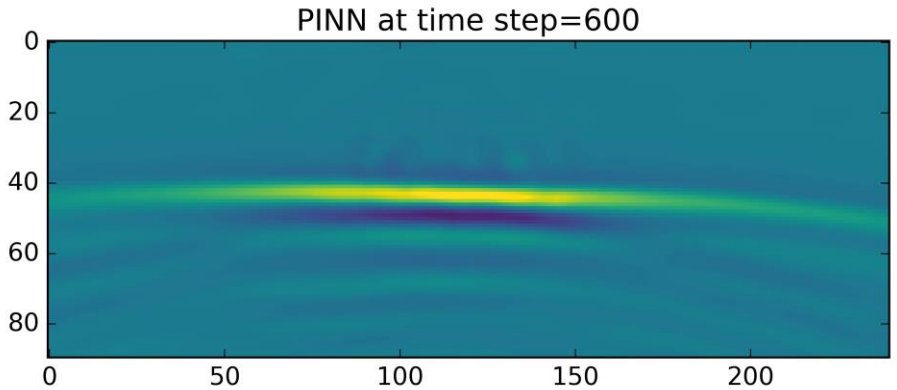
$$L = \sum_{i=1}^{N} \left(\underbrace{|\Psi_{\mathrm{NN}}(x^i, y^i, t^i) - \Psi^i_{\mathrm{data}}|^2}_{\text{data}} + \lambda \underbrace{|f(x^i, y^i, t^i)|^2|}_{\text{eq. enforcing}} \right)$$



Crack zone with space dependent c(x,y)



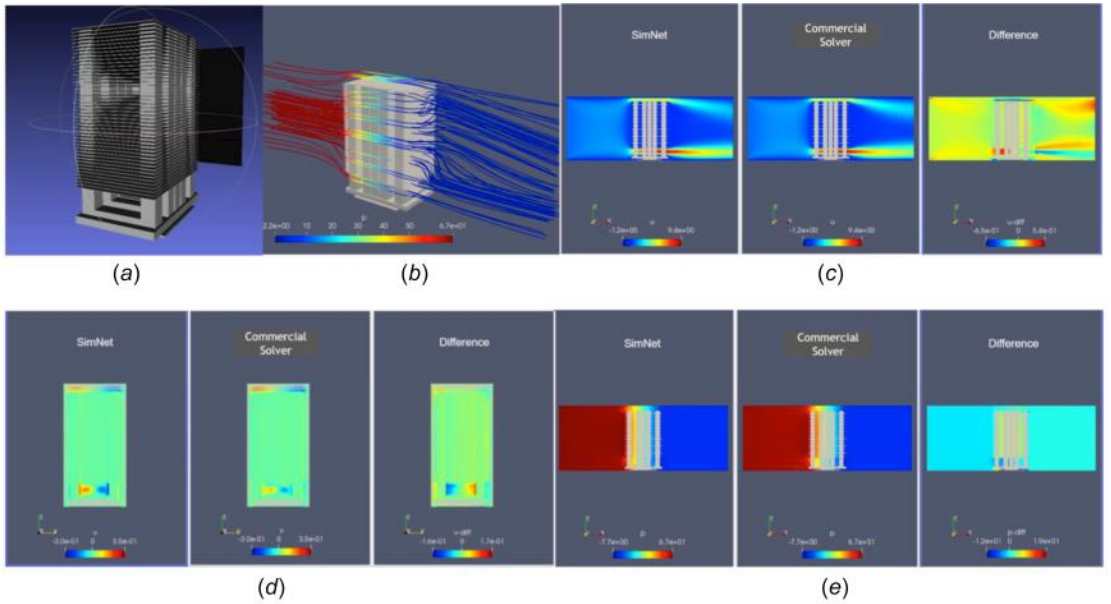




❑ Shukla K, Di Leoni PC, Blackshire J, Sparkman D, Karniadakis GE. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. Journal of Nondestructive Evaluation. 2020 Sep;39(3):1-20.

Physics-Informed Neural Networks for Heat Transfer Problems

J. Heat Transfer. 2021;143(6). doi:10.1115/1.4050542



Turbulent Flow (Re=20,687)		
	Temperature	Pressure Drop
SimNet - Fourier Network	43.1 °C	3.56
Commercial Solver	43.5 °C	3.6
OpenFOAM	41.6 °C	4.58

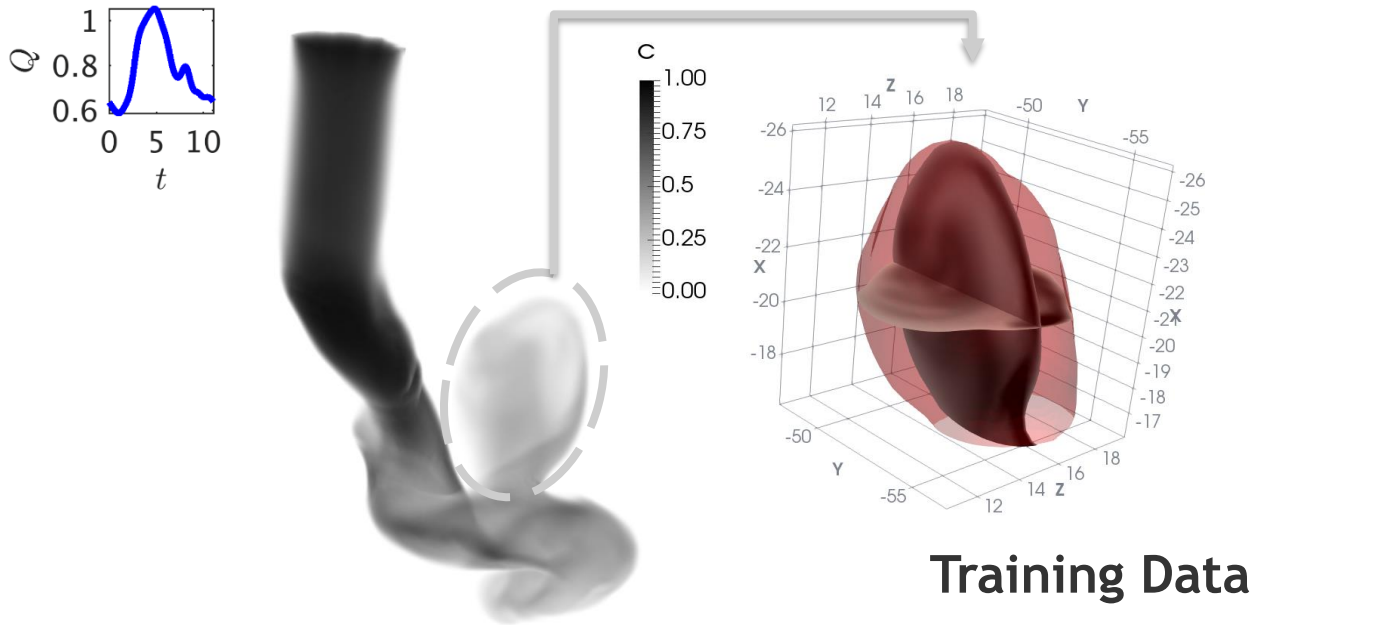
Computational Times (10 parameters, 3 values per parameter)	
SimNet	1000 V100 GPU hrs.
Traditional Solver (OpenFOAM) 59,049 separate runs (26 wall hours on 12 CPU cores)	18.4M CPU core hrs.

❑ **Solution of conjugate heat transfer problem on a complex parameterized geometry of a heat sink for NVSwitch in DGX-A100 (SimNet Simulation; now known as Modulus).**

(a) Geometry for a heat sink with fins and heat pipes to dissipate heat from GPU.
(b) Pressure color-coded flow streamlines and result comparisons between SimNet and CFD commercial code for (c) U-velocity, (d) V-velocity, and (e) pressure. (Courtesy of the Nvidia team).

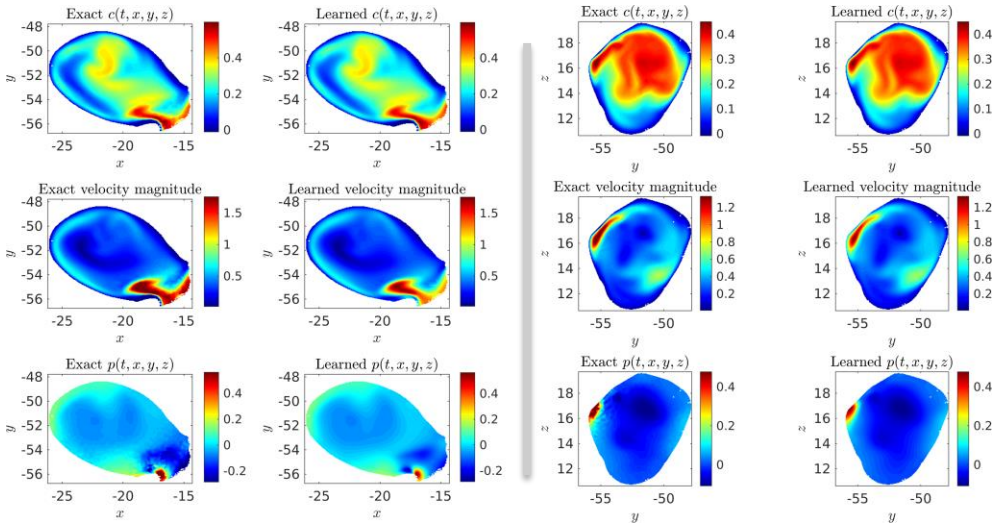
Hidden Fluid Mechanics: Flow dynamics in an intracranial aneurysm

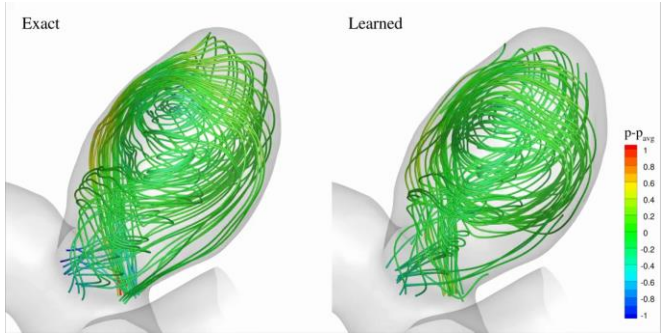
Science. 2020 Feb 28;367(6481):1026-30.



Training Data

Inferred Hidden Physics





Consumer Product Design Challenge: Rheology of a multicomponent complex fluid

***P&G* product (head & shoulders shampoo) design:**

"Data-driven physics-informed constitutive metamodeling of complex fluids: a multifidelity neural network (MFNN) framework." *Journal of Rheology* 65.2 (2021): 179-198.

Components:

- A background fluid/solvent (often water)
- Different colloidal particles
- Aromatic additives
- Different oils
- Surfactants for stabilization
- Different polymers
- Salt for pH level / salinity adjustment

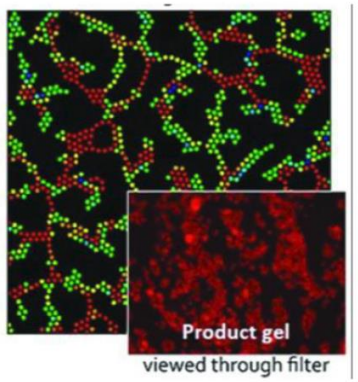
Challenge:

- No consistent constitutive model available
- No predictions are possible for aging behavior
- New formulation requires extensive testing
- Characterization
- Actual aging of samples

Expectations:

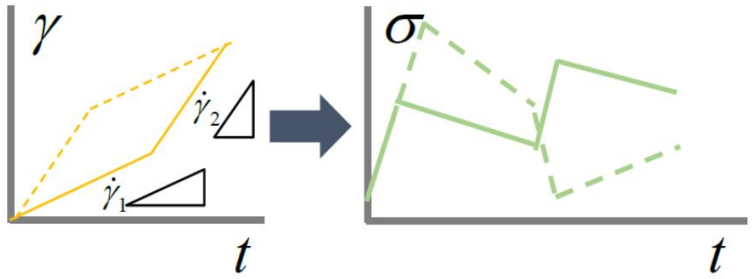
- A range for yield stress
  - To resist phase separation of larger particles
  - To easily flow out of the bottle for use
- Long shelf-life
- A given viscosity upon application
  - Rate-dependent viscosity
- Ease of processing

Typical microstructure



Mechanical response

Thixotropic Elasto-Visco Plastic (TEVP)

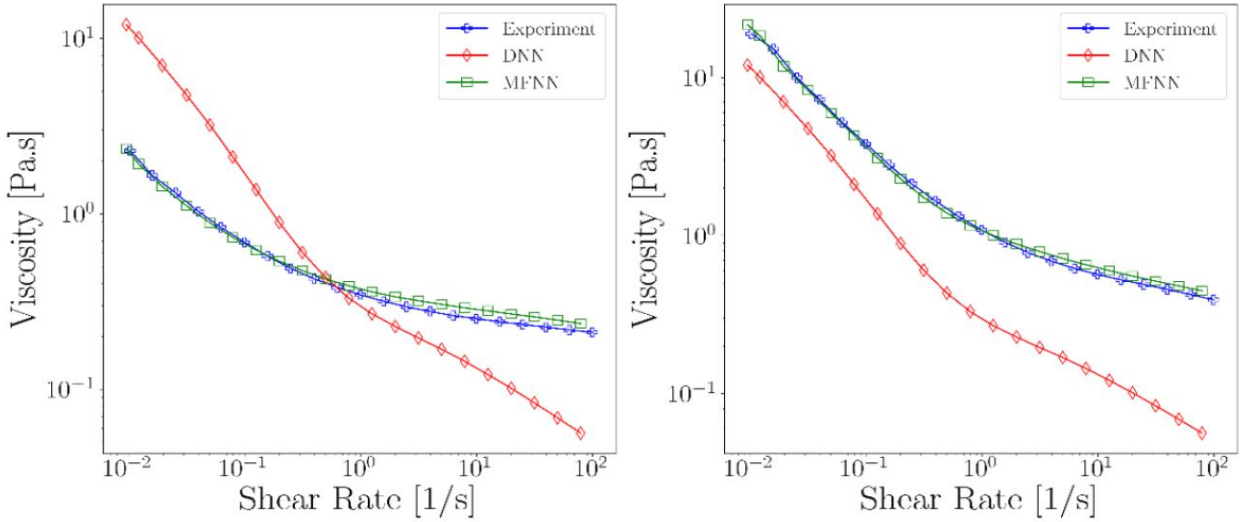


Rheology of a multicomponent complex fluid: Combining data and physics

"Data-driven physics-informed constitutive metamodeling of complex fluids: a multifidelity neural network (MFNN) framework." *Journal of Rheology* 65.2 (2021): 179-198.

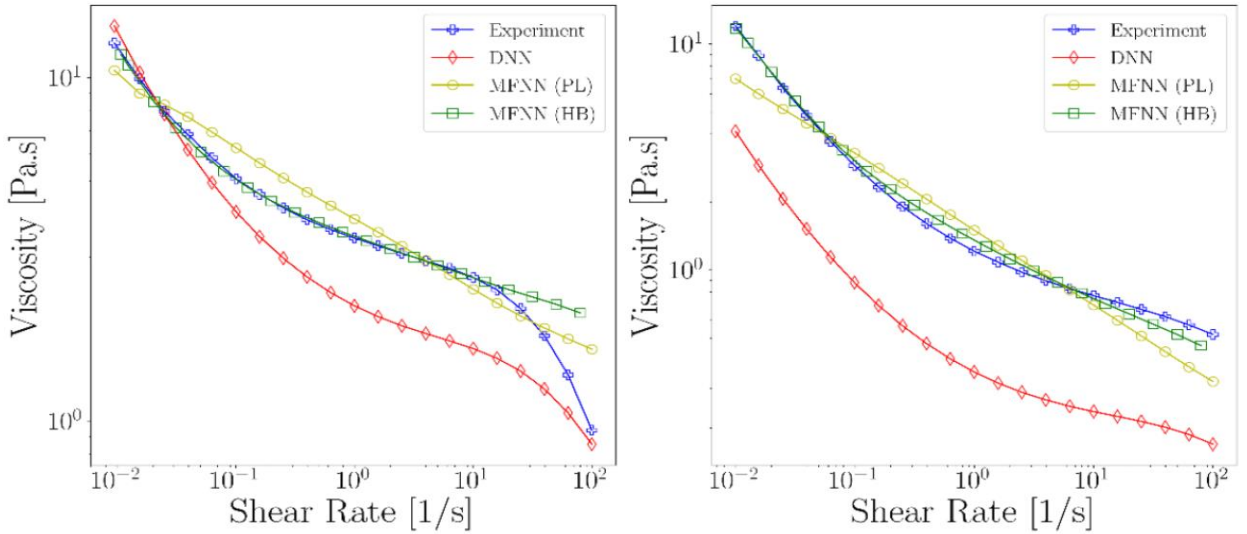
Capturing the rate dependent rheology:

- Two step shear-thinning
- Yield stress
- Different plateau values



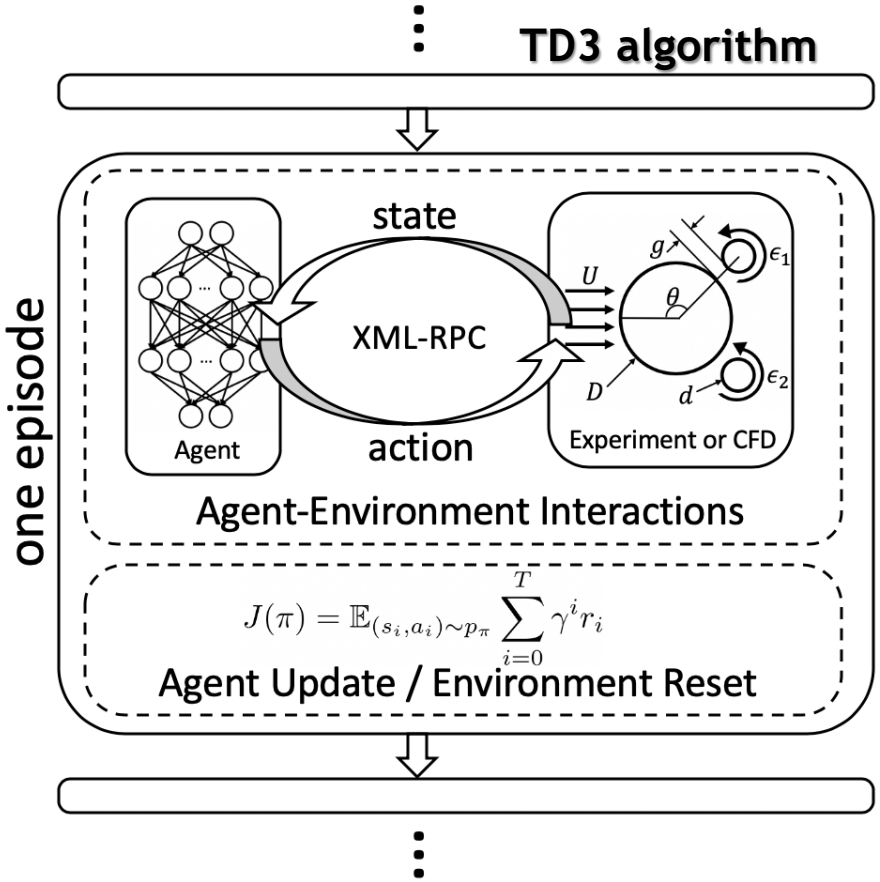
Role of physics intuition:

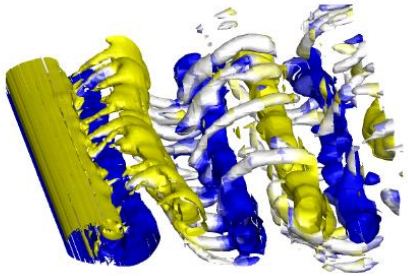
- Two parameter "power-law"
- Three parameter "Herschel-Bulkly"
  - Simplest physical model improves the predictions over DNN significantly
  - Commonly known general rheological models now provide predictive ability

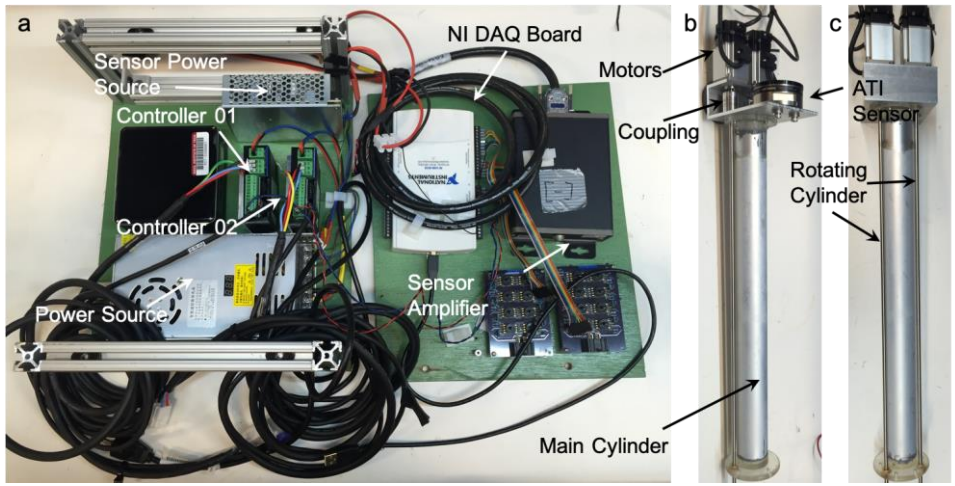


Accelerate Scientific Discovery: Reinforcement Learning for Flow Control in Experiments

Fan D, Yang L, Wang Z, Triantafyllou MS, Karniadakis GE. Reinforcement learning for bluff body active flow control in experiments and simulations. Proceedings of the National Academy of Sciences. 2020 Oct 20;117(42):26091-8.



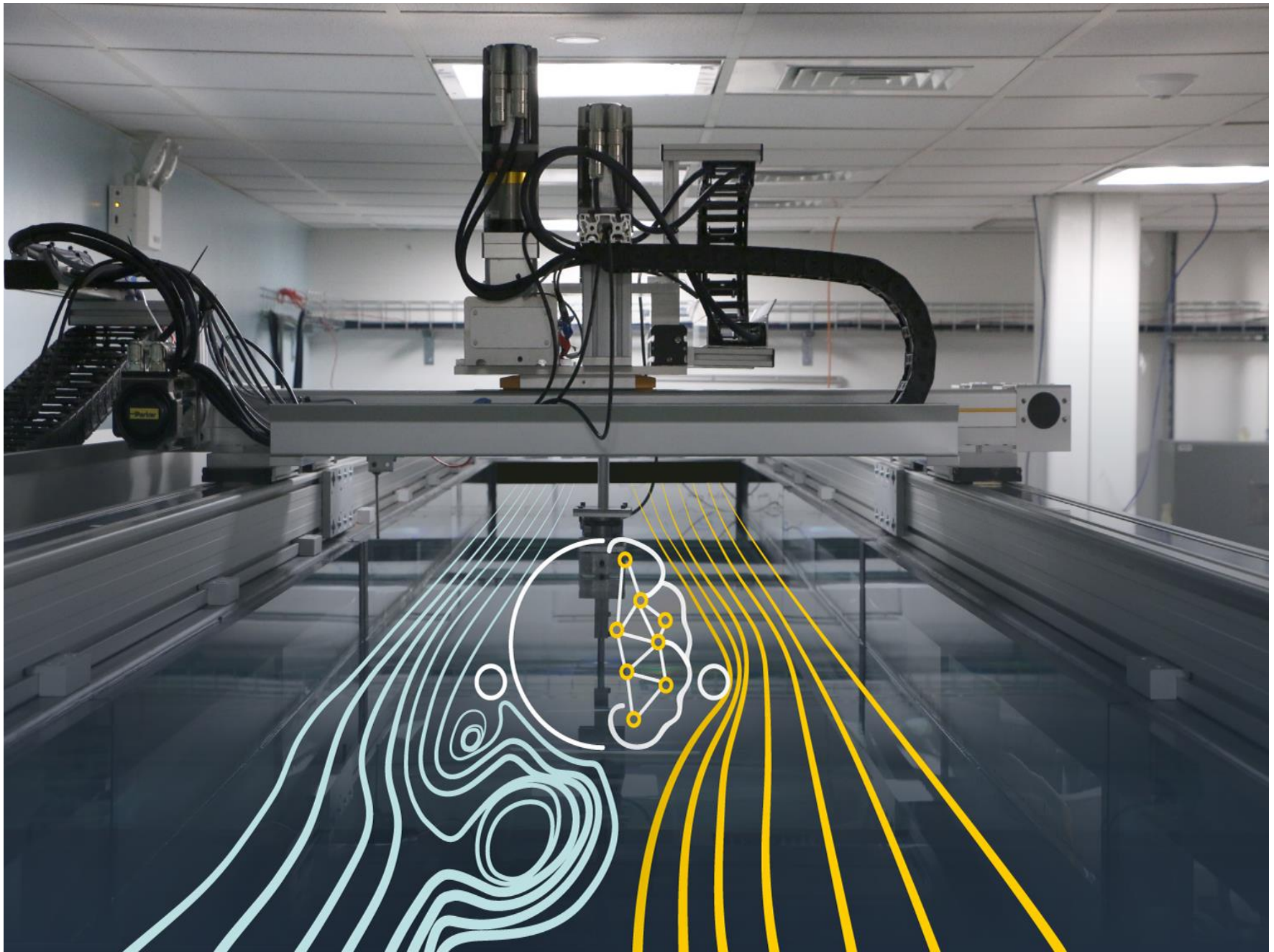




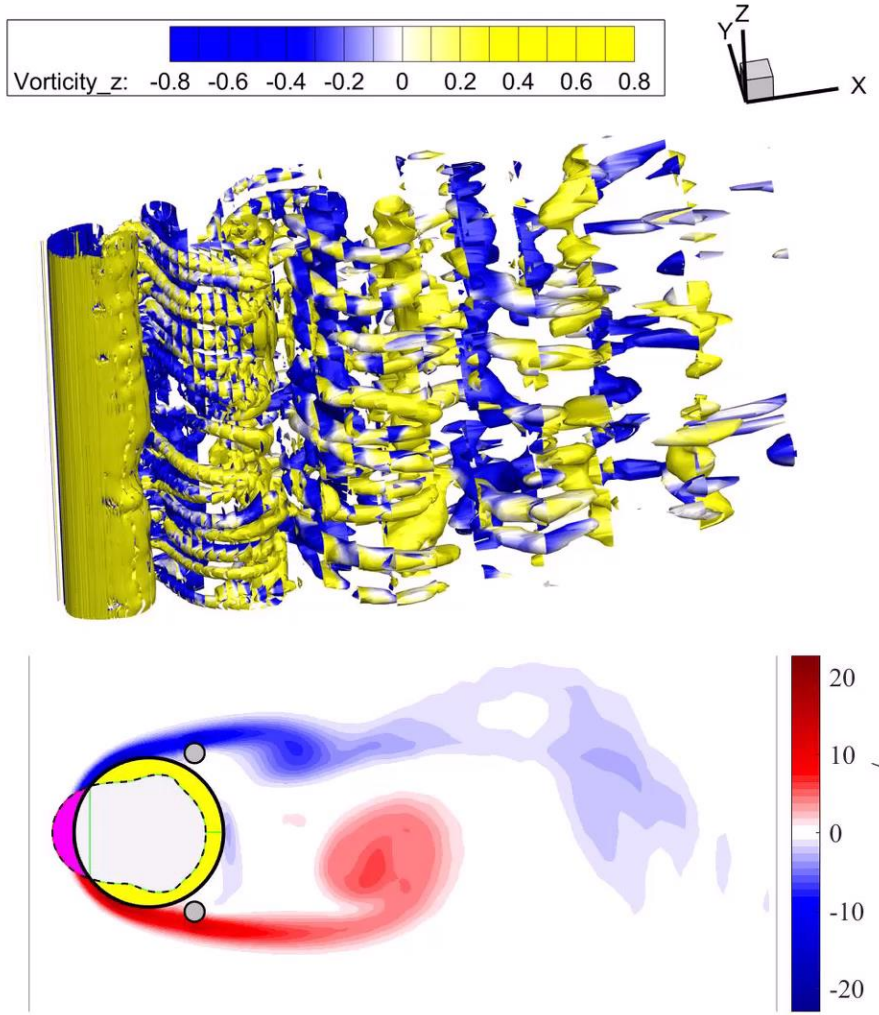
- We demonstrate, for first time, the effectiveness of RL in experimental fluid mechanics by applying it to reduce the drag of circular cylinders in turbulent flow. Although physics-agnostic, RL managed to reduce the drag by 30% very quickly.

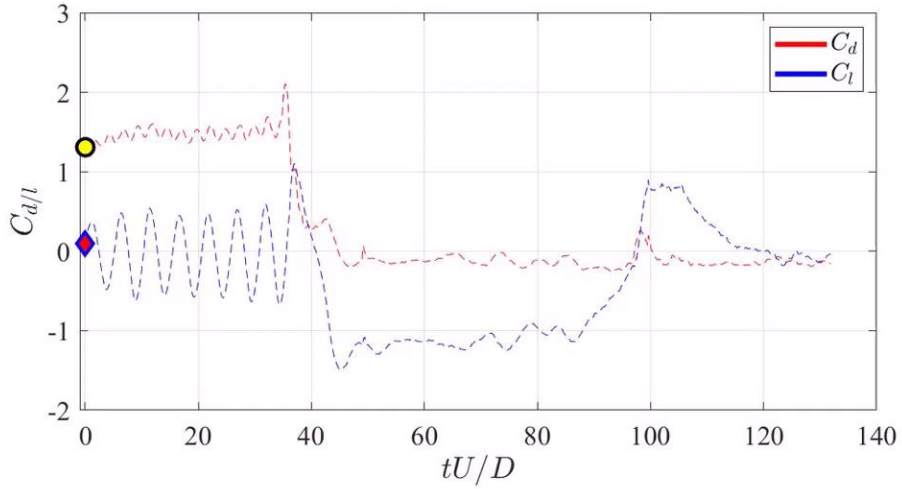
MIT Intelligent Towing Tank

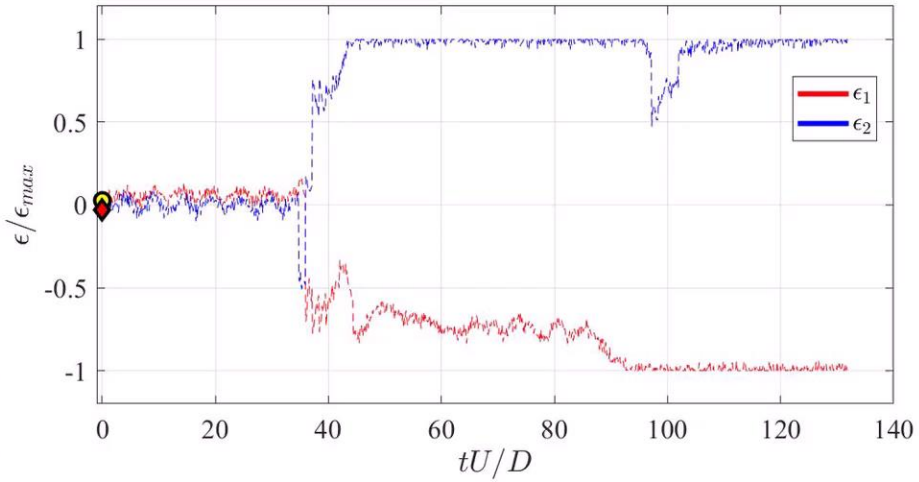
- Fan, Dixia, et al. "A robotic intelligent towing tank for learning complex fluid-structure dynamics." *Science Robotics* 4.36 (2019).



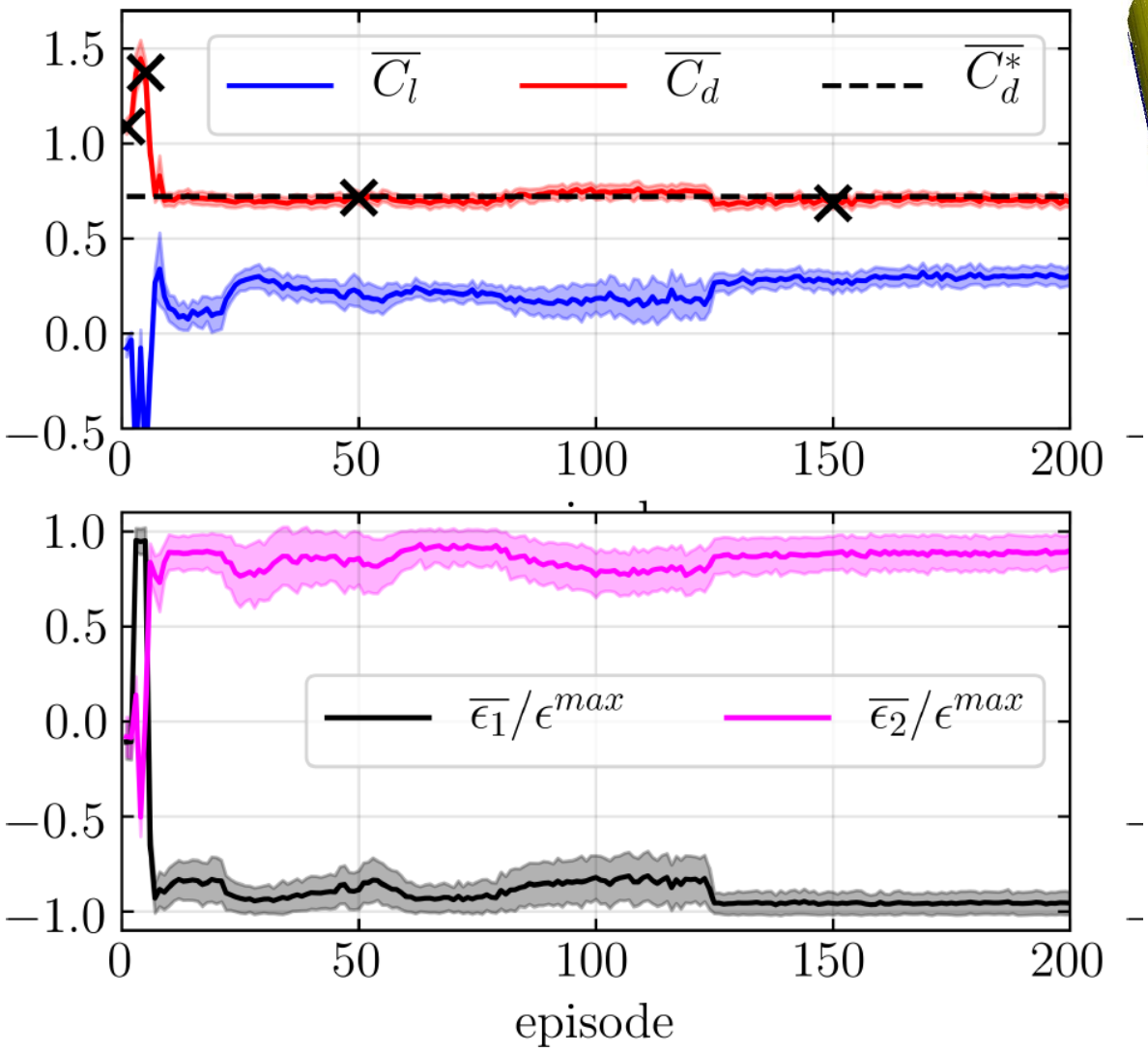
Reinforcement Learning for Cylinder Flow Control

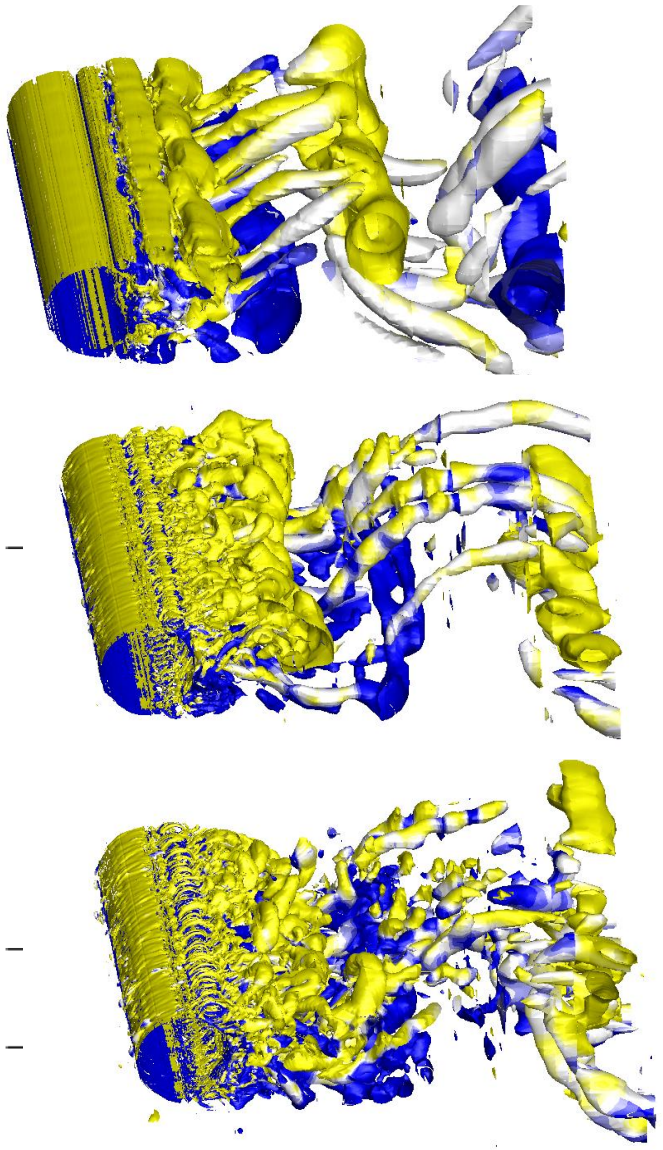






Case I: $r = -sgn(C_d)C_d^2 - 0.1{C_l}^2$ with KF; **PNAS, (2020)**





Supervised Learning Needs Big Data

- Supervised learning is the process of training a model by feeding it input data as well as corresponding output data. This input/output set is referred to as **labeled data**. It models the relationship between a set of descriptive features and target outcome(s). The model is a neural network that approximates implicitly the function that maps the input to the output in order to solve either classification or regression problems.

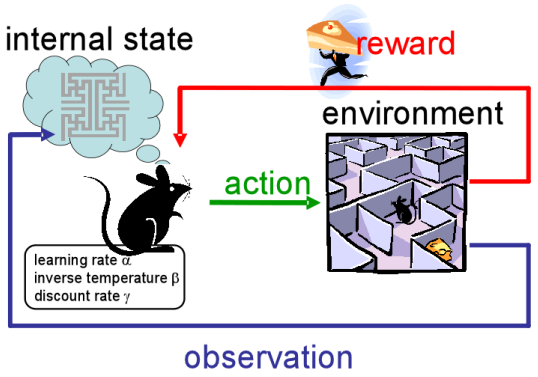
- In **Operator or Functional regression** the inputs are functions and the outputs are corresponding known functions for training, and unknown output functions for prediction.

- **Training - Where is the Big Data?**

  - Example: UCI Repository: https://archive.ics.uci.edu/ml/index.
  - Heart Disease (4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach): https://archive-beta.ics.uci.edu/ml/datasets/45
  - Airfoil Self-Noise: https://archive-beta.ics.uci.edu/ml/datasets/291: NASA wind tunnel data set based on aerodynamic and acoustic tests of 2D and 3D airfoil blade sections.
  - Challenger USA Space Shuttle O-Ring: https://archive-beta.ics.uci.edu/ml/datasets/92: Predict the number of O-rings that experience thermal distress on a flight at 31 degrees F given data on the previous 23 shuttle flights.

Unsupervised Learning: There are no predefined labels, i.e., no known set of outcomes. For example, solving ODEs based on initial conditions is unsupervised learning. Clustering, density estimation or looking for hidden patterns in the data are all examples of unsupervised learning.

Reinforcement Learning: Learning through delayed feedback by interacting with the environment. Reinforcement Learning deals with exploitation or exploration, Markov's decision processes, policy learning, and value learning. It is useful for flow control problems (e.g., drag force control in flow past a cylinder; navigation/autonomy, etc.



Summary

- Supervised Learning predicts based on a class type; it maps labelled data to known output; immediate feedback.
- Unsupervised Learning discovers underlying patterns; it explores patterns to predict the output; no feedback.
- Reinforcement Learning follows a trial-and-error approach; the learning agent works as a reward and action system; delayed feedback.

14 Different Types of Learning in Machine Learning

by **Jason Brownlee** on November 11, 2019 in **Start Machine Learning**

Self-Supervised Learning: It looks like unsupervised learning, where some of the data provides the supervision. Supervised learning algorithms are used to solve an alternate or pretext task, the result of which is a model or representation that can be used in the solution of the original modeling problem.

- **Autoencoders** is a general example of self-supervised learning algorithms. These are neural networks that are used to create a compressed representation of an input sample. They achieve this via a model that has an encoder and a decoder element separated by a bottleneck that represents the internal compact representation of the input. They are trained using a supervised learning method, but they solve an unsupervised learning problem; they are a projection method for reducing the dimensionality of input data.

- **Generative Adversarial Networks** (GANs) is another example of self-supervised learning. These are generative models that are inspired by game-theoretic approaches and were originally used for creating synthetic photographs using only a collection of unlabeled examples from the target domain. We will use GANs to obtain informative priors and solve stochastic differential equations in high dimensions.

https://machinelearningmastery.com/types-of-learning-in-machine-learning/

Course Roadmap

Module-1 (Basics)

- Lecture 1: Introduction
- Lecture 2: A primer on Python, NumPy, SciPy and *jupyter* notebooks
- Lecture 3: Deep Learning Networks
- Lecture 4: A primer on TensorFlow and PyTorch
- Lecture 5: Training and Optimization
- Lecture 6: Neural Network Architectures

Module-2 (PDEs and Operators)

- Lecture 7: Machine Learning using Multi-Fidelity Data
- Lecture 8: Data-Driven Dynamical Systems
- Lecture 9: Physics-Informed Neural Networks (PINNs)
- Lecture 10: PINN Extensions
- Lecture 11: Neural Operators
- Lecture 12: Uncertainty Quantification(UQ) in Scientific Machine Learning

Module-3 (Codes & Scalability)

- Lecture 13: DeepXDE Library
- Lecture 14: MODULUS Library
- Lecture 15: Multi-GPU Scientific Machine Learning

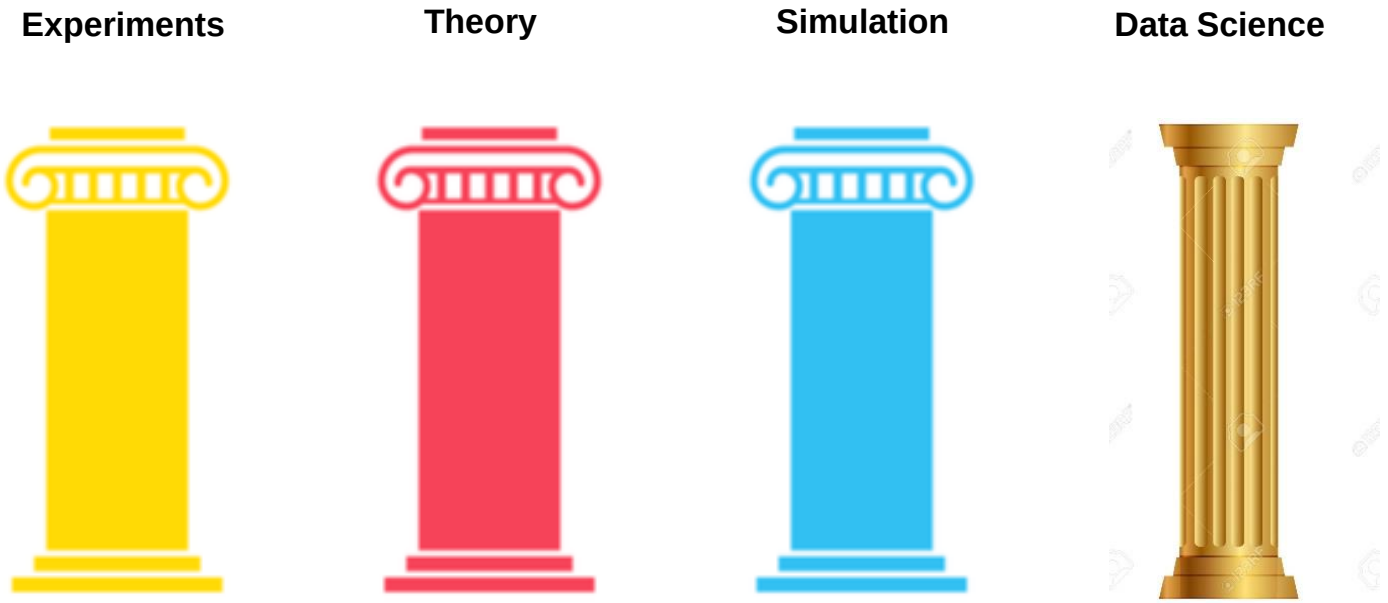
Course Projects

- Biomedicine
- Dynamical Systems
- Engine Dynamics
- Fluid Mechanics
- Geophysics
- Heat Transfer
- Materials

The Four Pillars of Scientific Methods

“Data science (AI+HPC) is the fastest growing field of computer science today. Because of several different factors, it has become the fourth pillar of the scientific method.”

Jensen Huang, NVIDIA



References

- Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE. Physics-informed neural networks for heat transfer problems. Journal of Heat Transfer. 2021 Jun 1;143(6).
- Fan D, Yang L, Wang Z, Triantafyllou MS, Karniadakis GE. Reinforcement learning for bluff body active flow control in experiments and simulations. Proceedings of the National Academy of Sciences. 2020 Oct 20;117(42):26091-8.
- Fan D, Jodin G, Consi TR, Bonfiglio L, Ma Y, Keyes LR, Karniadakis GE, Triantafyllou MS. A robotic intelligent towing tank for learning complex fluid-structure dynamics. Science Robotics. 2019 Nov 27;4(36):eaay5063.
- Géron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc.; 2019.
- Goodfellow I, Bengio Y, Courville A. Deep learning, Deep Learmning, MIT press; 2016; also, https://www.deeplearningbook.org/
- Mahmoudabadbozchelou M, Caggioni M, Shahsavari S, Hartt WH, Em Karniadakis G, Jamali S. Data-driven physics-informed constitutive metamodeling of complex fluids: A multifidelity neural network (mfnn) framework. Journal of Rheology. 2021 Mar 11;65(2):179-98.
- Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics. 2019 Feb 1;378:686-707 (published in arxiv in 2017).
- Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science. 2020 Feb 28;367(6481):1026-30.
- Shukla K, Di Leoni PC, Blackshire J, Sparkman D, Karniadakis GE. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. Journal of Nondestructive Evaluation. 2020 Sep;39(3):1-20.

- Artificial versus biological neurons: https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon
- DOE Report on Scinetific Machine Learning: https://www.osti.gov/servlets/purl/1478744
- History of Deep Learning: https://www.import.io/post/history-of-deep-learning/
- Types of machine learning: https://machinelearningmastery.com/types-of-learning-in-machine-learning/
- NSF/ICERM Workshop on SciML, January 28-30, 2019 (Organizers: J. Hesthaven & G.E. Karniadakis): https://icerm.brown.edu/events/ht19-1-sml/)
- Rosenblatt's perceptron: https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Deep Learning for Science and Engineering Teaching Kit

Thank You