# Simulation, Analysis, and Validation of Computational Models

## — Bonus I: PINN

Lecturer: Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

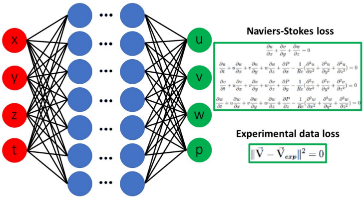
# Overview

- Neural Networks
- Physics
- PINN

# Last time: Physics-informed neural networks

Physics-informed ML

- Efficient machine learning
- Physics-informed reinforcement learning, active learning
- Physics-informed regularisation



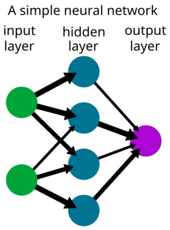

Understanding system physics

- Qualitative modelling by identification of underlying regularities
- Learning to simulate complex phenomena from sparse data based on physics priors (PDE).
- Closed loop systems to perform process optimization

see e.g. https://www.youtube.com/watch?v=ISp-hq6AH3Q

# Neural networks

- Hardware:
  1. Generation of NN (1958): Electronic computers
  2. Gen. NN (1986): VLSI
  3. Gen. NN (2012): GPUs
- Function approximation, data generation, novelty detection

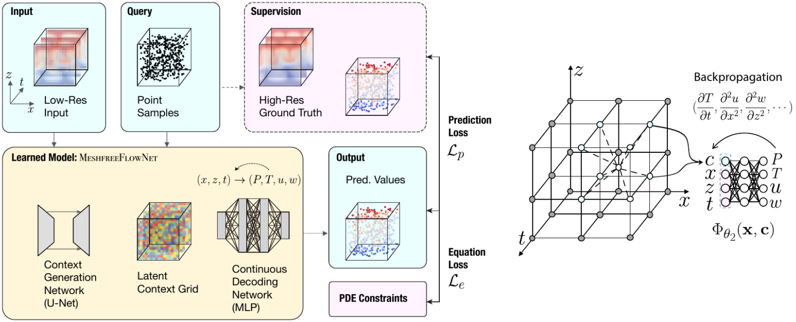



- Hyperparameters: Batch size, learning rate, regularisation, unit type, architecture, cost function
- Problems:
  - High sample complexity and long training time
  - Efficiency, complexity, theory, verifiability
  - Explainabilty, unlearnability, robustness to adversarial attacks
  - Insight, understanding, intelligence

# Physics

- Physics is the study motion and behavior of matter in space and time.
- Reduction of the RW complexity to essential and repeatable aspects.
- Regularity of continuous trajectories (or probabilities or wave functions) can be described by differential equations which are usually derived from the principle of least action.
- In addition to dynamics, also symmetry and conservation laws can be incorporated.
- Extrapolation of known regularities can be used to make testable predictions, which may lead to an insight in more complex regularities.

# Weather and climate modelling




Esmaeilzadeh e.a. (2020) MeshFreeFlowNet: A physics-constrained deep continuous space-time super-resolution framework. In SC20: Int. Conf. HPCNSA, p. 1-15.

see also Kashinath e.a. (2021) APhysics-informed machine learning: case studies for weather and climate modelling. Phil. Trans. R. Soc. A**379**, 202000093.

# What is a physics-informed neural network?

- Machine learning can solve a scientific problem using data alone.
- Do these algorithms "understand" the scientific problems they are trying to solve?
- Minimising MSE of NN output and true values

$$\min \frac{1}{N} \sum_{i=1}^{N} \left( u_{\mathrm{NN}}\left(x_i, \theta\right) - u_{\mathrm{true}}\left(x_i\right)\right)^2$$

  does not mean that the neural network can generalise well.
- The question is not how to improve generalisation, but to describe sets of possible data, e.g. for temporal data by a relation

$$\dot{x} = F(x)$$

Ben Moseley (https://benmoseley.blog)

# PINN

- *Big data* (by definition): all relevant structure can be inferred from the data
- *Small data* can be as useful, if underlying principles ("physics") are known (e.g. as initial values)
- What can be achieved for "some data" with "some physics"?

- Add the known differential equations directly into the loss function when training the neural network.

$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0 \quad \text{and} \quad \frac{1}{N}\sum_i \left(u_{\mathrm{NN}}\left(x_i\right) - u_i\right)^2$$

- PINN started 2017, but there is earlier work of similar flavour.

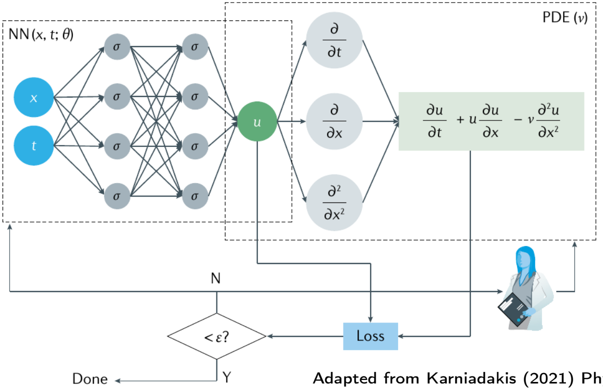
Lagaris (1998) Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Transact. Neural Networks* **9**, 987

# PINN

Given: PDE $\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + v\frac{\partial^2 u}{\partial x^2}$ and data $(x_i, t_i, u_i)$, $i = 1, \ldots, N_{\text{data}}$

Loss: $\mathcal{L} = w_{\text{data}}\mathcal{L}_{\text{data}} + w_{\text{PDE}}\mathcal{L}_{\text{PDE}}$ where $w_{\text{data}}$ and $w_{\text{PDE}}$ are weights and

$$\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}}(u(x_i, t_i) - u_i)^2; \quad \mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}}\sum_{j=1}^{N_{\text{PDE}}}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2}\right)^2\Bigg|_{x=x_j(t)}$$




Adapted from Karniadakis (2021) Physics-informed machine learning. *Nat. Rev. Phys.* 3, 422

# PINN: Remarks

Given: PDE $\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + v\frac{\partial^2 u}{\partial x^2}$ and data $(x_i, t_i, u_i)$, $i = 1, \ldots, N_{\text{data}}$

Loss: $\mathcal{L} = w_{\text{data}}\mathcal{L}_{\text{data}} + w_{\text{PDE}}\mathcal{L}_{\text{PDE}}$ where $w_{\text{data}}$ and $w_{\text{PDE}}$ are weights and

$$\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}}\left(u(x_i, t_i) - u_i\right)^2, \;\; \mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}}\sum_{j=1}^{N_{\text{PDE}}}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2}\right)^2\Bigg|_{x=x_j(t)}$$

- PDE data $(x_j, t_j)$ can be different from training data $(x_i, t_i, u_i)$. E.g. trivial case: $N_{\text{data}} = 1$ just check whether initial value is met.
- Trust in PDE and in data can be in different (weights!).
- Error components can be in different ranges (weights!).
- PDE and data can be spatially heterogeneous.
- PDE and data have different stiffness (Edit PDE?)

Karniadakis (2021) Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422

# Automatic differentiation

- **Numerical differentiation**: Calculate differences between data points. As this tends to amplify errors, it is usually combined with a smoothing scheme, e.g. "five-point stencil":

$$\frac{df(x)}{dx} \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}$$

- **Symbolic differentiation**: Manipulation of expressions by rewriting rules e.g. $h(g(x))' = h'(g(x)) \cdot g'(x)$.
- **Automatic differentiation**[1]: Computational form of symbolic differentiation that emphasises computability. Classic example:

$$\frac{d \tanh x}{dx} = 1 - x^2$$

# Automatic differentiation

- A function $y = f[x]$ (in any programming language) receives argument $x$ and returns the value $y$.
- In addition, another function calculates

$$\frac{dy}{dx} = \left.\frac{f[z]}{dz}\right|_{z=x}$$

- Represent $f$ as computational graph and calculate derivative by

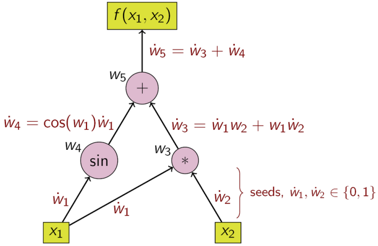
$$\delta w_i = \sum_{j \in \{\text{predecessors of i}\}} \frac{\partial w_i}{\partial w_j}\, \delta w_j$$

- Focus on "computational" functions $(\cos, \exp, \tanh \text{ etc.})$

# Automatic differentiation: Example

$$
\begin{aligned}
y = f(x_1, x_2) &= x_1 x_2 + \sin x_1 \\
&= w_1 w_2 + \sin w_1 \\
&= w_3 + w_4 \\
&= w_5
\end{aligned}
$$

$$\frac{dy}{dx_1} = x_2 + \cos x_1$$



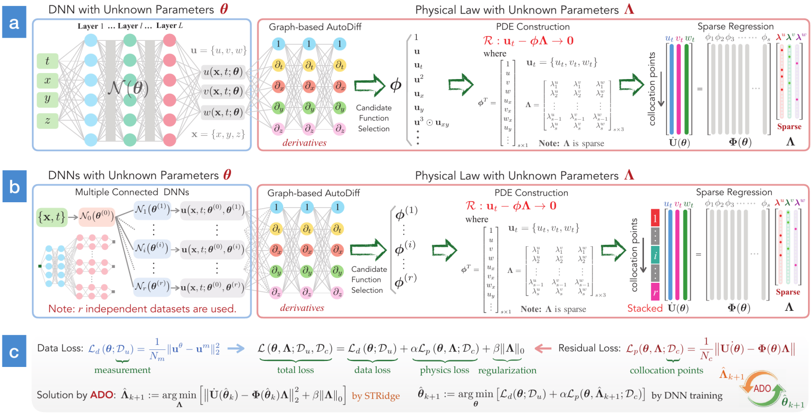

Construct a corresponding structure for a computational derivative.
Seed determines what derivative is taken (here $x_1$)

| value | derivative |
|---|---|
| $w_1 = x_1$ | $\delta w_1 = 1$ (seed) |
| $w_2 = x_2$ | $\delta w_2 = 0$ (seed) |
| $w_3 = w_1 \cdot w_2$ | $\delta w_3 = w_2 \cdot \delta w_1 + w_1 \cdot \delta w_2$ |
| $w_4 = \sin w_1$ | $\delta w_4 = \cos w_1 \cdot \delta w_1$ |
| $w_5 = w_3 + w_4$ | $\delta w_5 = \delta w_3 + \delta w_4$ |

# Automatic differentiation: Pseudocode

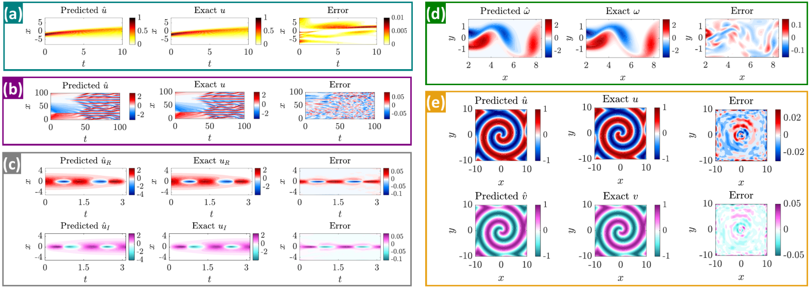
```
<float,float> evaluate_derive(expr Z, var V) {
        if is_var(Z)
                if (Z = V) return {value_of(Z), 1};
                else return {value_of(Z), 0};
        else if (Z = A + B)
                {a, da} = evaluate_derive(A, V);
                {b, db} = evaluate_derive(B, V);
                return {a + b, da + db};
        else if (Z = A - B)
                {a, da} = evaluate_derive(A, V);
                {b, db} = evaluate_derive(B, V);
                return {a - b, da - db};
        else if (Z = A * B)
                {a, da} = evaluate_derive(A, V);
                {b, db} = evaluate_derive(B, V);
                return {a * b, b * da + a * db};
}
```

# PINN for learning equations from scarce data




Chen e.a. (2021) Physics-informed learning of governing equations from scarce data. *Nature comm.* 12, 6136.

# PINN for learning equations from scarce data



(a) Burgers equation, (b) Kuramoto-Sivashinsky equation, (c) nonlinear Schrödinger equation, (d) Navier-Stokes equation, and (e) $\lambda - \omega$ reaction-diffusion equations. Sparsely sampled measurement data has 10% noise.

Chen e.a. (2021) Physics-informed learning of governing equations from scarce data. *Nature comm*. **12**, 6136.

# PINNs: Further benefits

- Ill-posed and inverse problems.
- Scalability when combined with domain decomposition
- Search for new intrinsic variables and representations
- Incorporate conservation laws

Karniadakis (2021) Physics-informed machine learning. Nat. Rev. Phys. 3, 422

# PINNs: Limitations (and amendments)

- Discontinuous behavior: piecewise PINNs
- Translation and advective dominance ("wind") require special tuning, as all systems with strongly different time scales: Distributed PINNs
- Soft constraints may require re-weighing the loss terms.
- Chaotic systems remain chaotic and their prediction is limited (high precision simulations of deterministic systems can be impressively predictable by PINNs).
- PINNs need to be informed: More general differential equations can be used or by Genetic Programming.
- Can get stuck in local optima like any other optimisation method.
- Grid-based numerical solvers are quicker in forward problems.
- Limited to physics: CINNs, BINNs, LINNs have been proposed and tested.

Rout (2021) Numerical approximation in CFD problems using physics informed machine learning. (arxiv)

# Open problems

- Domain decomposition
- Phase transitions
- Combination of physics with other information (e.g. boundary conditions)
- Incorporation of non-physics laws (such as Black Scholes)
- Causality
- Efficiency
- Theory of PINN: Validation

Cuomo e.a. (2022) Scientific machine learning through physics–informed neural networks: Where we are and what's next. *J. Scient. Comput.* **92**, 88.

# Conclusion

- PINNs can provide reasonable extrapolations of data and in this respect perform better than standard neural networks.
- It could seem as if PINN have an understanding of the underlying physical principles which is no surprise as this information was made available to the PINN in the first place.
- Including existing physical principles into machine learning leads to more versatile models, nontrivial predictions, and thus can help to improve scientific understanding.

# Next topics

Next week

- Testing
- Validation
- Verification
- Confidence

Next bonus lecture

- Connections to kernel methods