NN-based reduced order modeling of PDEs

Andrea Boselli
Carlo Ghiglione
Leonardo Perelli



From a Classical Approach...

Model: $\forall \mu, \underline{\beta}$ $\begin{cases} -\mu \Delta u + \underline{\beta} \cdot \nabla u = f(\underline{x}) & \underline{x} \in \Omega \\ BCs & \underline{x} \in \partial\Omega \end{cases}$

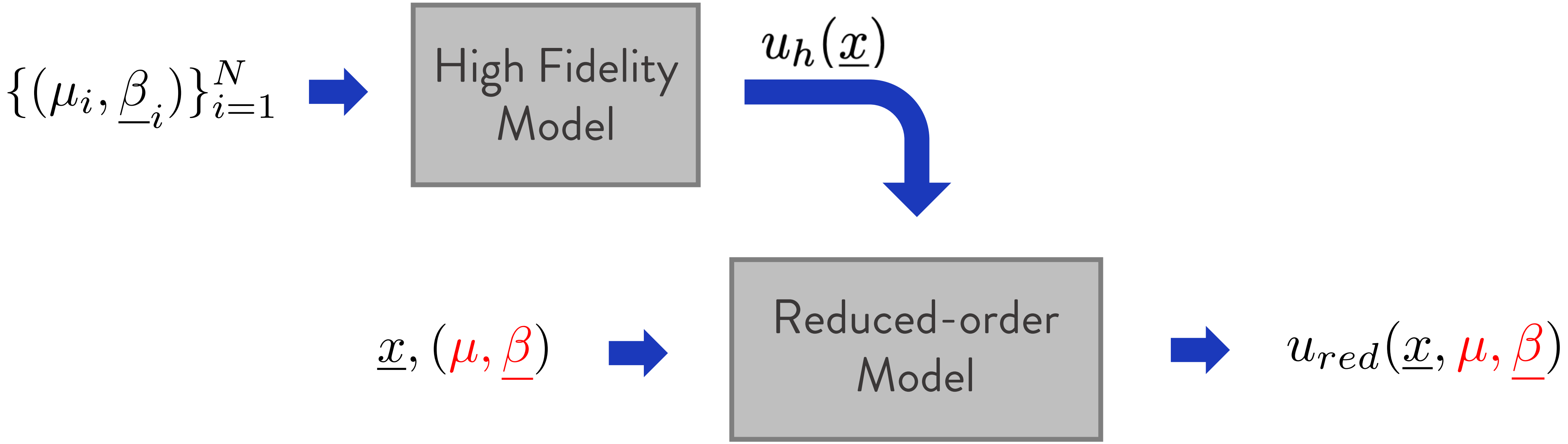
Numerical Method Solution: $u_h(\underline{x}) \approx u(\underline{x})$

Pros: High accuracy

Cons: Time duration
Computational resources

➡ Very expensive to solve for many sets of parameters!

… to Reduced Order Modeling

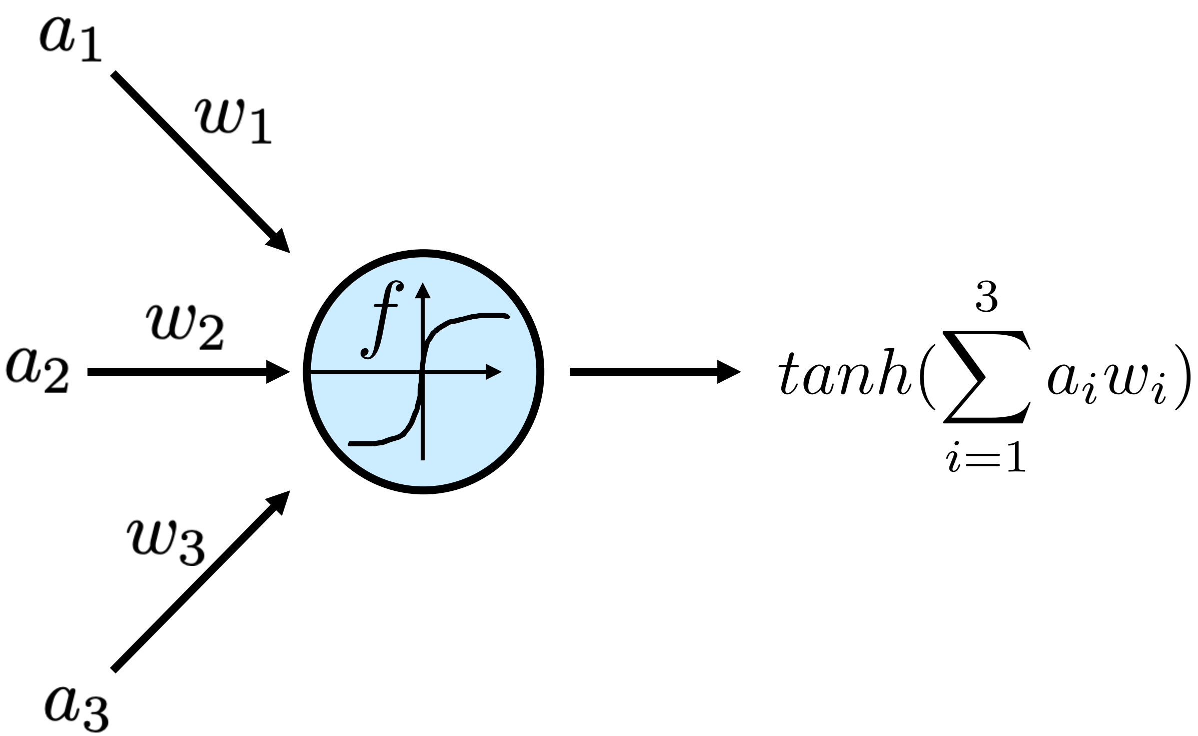


- Long training time
- Model design

→

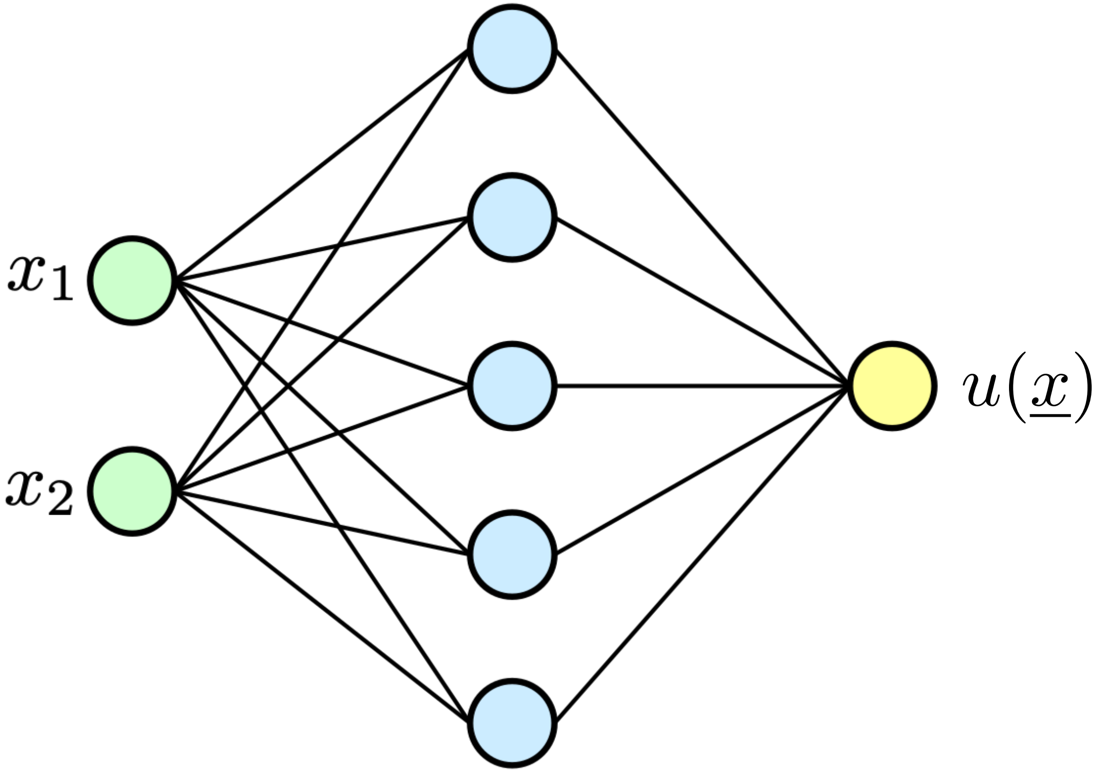
- Fast execution
- Reusable

Neurons



- Inputs: a_1, a_2, a_3
- Weights: w_1, w_2, w_3
- Activation function: $tanh$
- Output: $tanh(\sum_{i=1}^{3} a_i w_i)$

Neural Networks



- Stack of layers of neurons
- Very complex and nested functions
- Can approximate any function
  (Universal Approximation Theorem, Hornik,1991)
- Weights determine the output of the network

How to find the best weights?

Loss Minimization

The loss measures the distance between the output of the neural network function and the target function

Loss $L(\underline{w})$ ⬇ Quality of the NN approximation ⬆

➡ Minimize $L(\underline{w})$: $\underline{w}_{opt} = \underset{\underline{w}}{argmin}\, L(\underline{w})$

➡ Gradient descent: $\underline{w}_{n+1} = \underline{w}_n - \eta \nabla L(\underline{w}_n)$

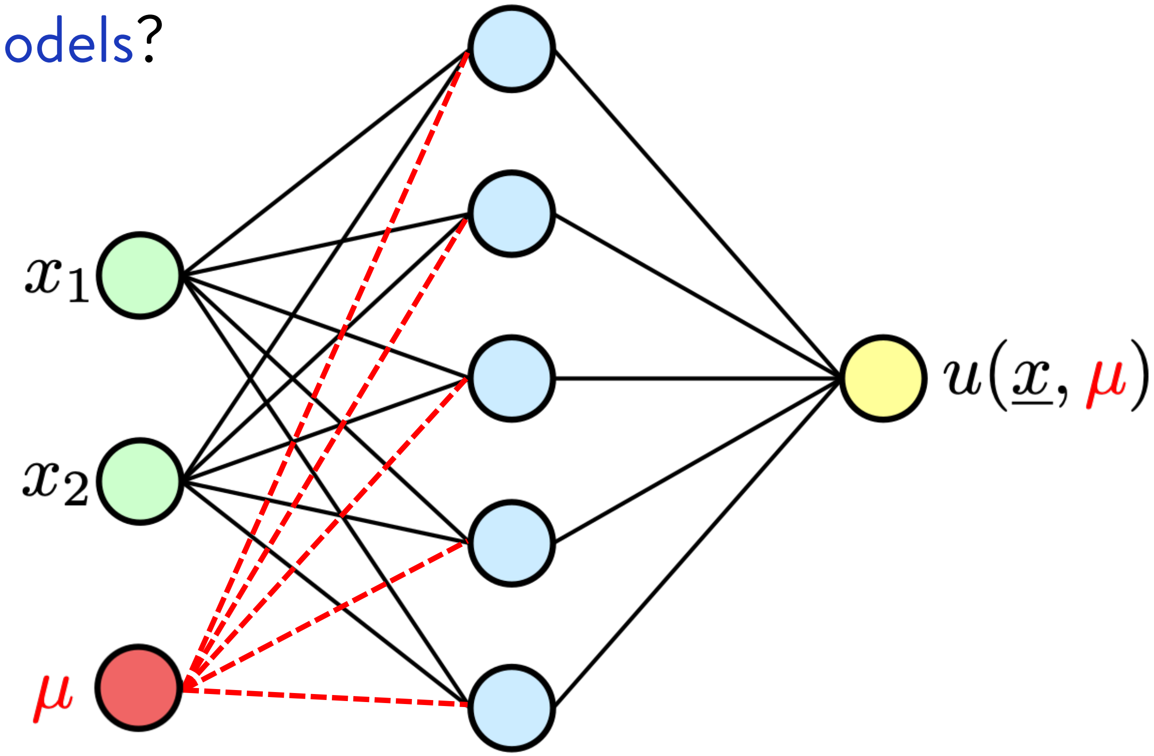
NNs meet PDEs

How to use Neural Networks as Reduced Order Models?

➡ Include PDE's parameters in the input of NN

How to include knowdedge of the physics?

➡ PINNs!



PINNs

To enforce the physics of the problem, we introduce in the loss function the residual of the Neural Network solution with respect to the PDE

$$L(\underline{w}) \;=\; \alpha_1 L_{Fit}(\underline{w}) \;+\; \alpha_2 L_{PDE}(\underline{w}) \;+\; \alpha_3 L_{BC}(\underline{w})$$

- $L_{Fit}(\underline{w})$: Approximation error
- $L_{PDE}(\underline{w})$: PDE residual of Neural Network solution
- $L_{BC}(\underline{w})$: BC residual of Neural Network solution

Goals

Does the PDE loss term help?

How much should the different loss terms be weighted?

How accurate are the solutions?