Discrete Fracture Network insights by eXplainable AI

NIPS 2020 Workshop Machine Learning and the Physical Sciences - 11^{st} December 2020

Berrone S.[1], Della Santa F.[1], Mastropietro A.[1,2], Pieraccini S.[1], Vaccarino F.[1,3].

[1]Politecnico di Torino, [2]Addfor S.p.A., Torino, [3]ISI Foundation, Torino

POLITECNICO DI TORINO

SmartData@PoliTO

Introduction

Underground analysis of fractured media requires flow simulations (e.g. applications geothermal applications, oil & gas production).

Discrete Fracture Networks (DFN) are discrete models composed by a network of $2D$ polygonal fractures in a $3D$ domain, that can accurately simulate the flow of a fracture medium.

The reformulation by [Berrone, Pieraccini, Scialo' 2013, 2014, 2016] guarantees advantages but numerical solutions of DFN are still prohibitive for the large number of simulations required by Uncertainty Quantification (UQ) analyses.

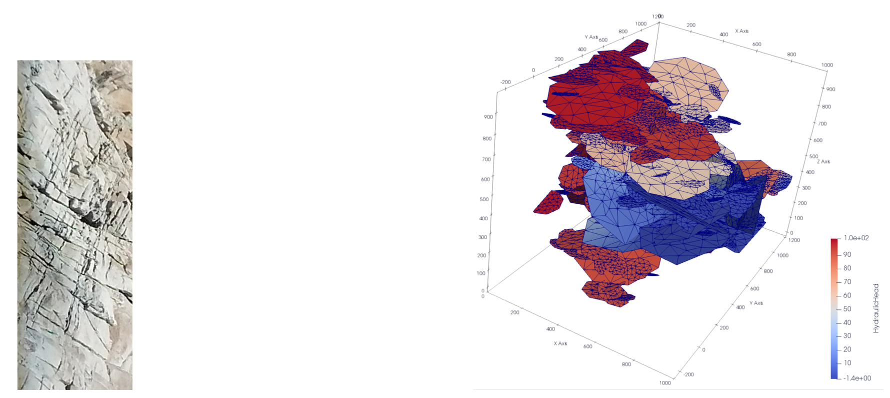


Figure 1: surface of a natural fractured medium (left) and a DFN (right)

Major Issues

Fracture media cannot be fully described, then:

- **Generation**: lacks of full deterministic data
  $\Rightarrow$ **DFNs stochastically generated**.

> Quantify the uncertainty of stochastic generation
>
> $\Longrightarrow$ **Uncertainty Quantification** (UQ)

- **Simulation**: **complex** computational domain & **expensive** computations

> **Reduce the DFN complexity**
>
> $\Longrightarrow$ Backbone Identification

Main Target: Backbone Identification

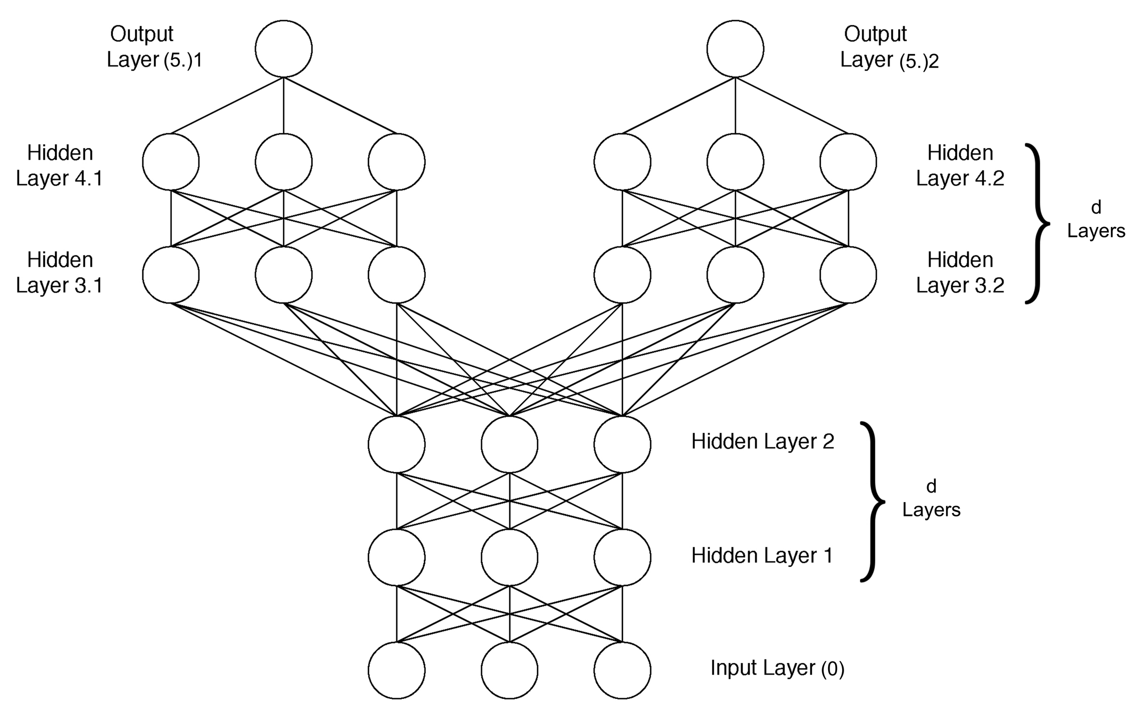
> Backbone B: **sub-network** of fractures with **transport characteristics** approximating the original DFN

- **DFN158**: Fix the DFN geometry with $n = 158$ fractures randomly generated from geological distributions (7 outflow fractures).
  Assume varying fracture trasmissivities $\log_{10} \kappa_i \sim \mathcal{N}(-5, 1/3)$.
- **Flow Simulation**: fixed boundary Dirichlet conditions of fixed pressure ΔH between influx and outflux fractures.
- **Backbone validation**: run flow simulations of fractures sub-network and compare ϕ, ϕ_B.

> ϕ, ϕ_B exiting flux distributions of full DFN and Backbone:
>
> $\Longrightarrow \phi \approx \phi_B$

NN for Flux Regression in DFN

> Use a **Neural Network (NN)** for regression of **exiting fluxes** $\varphi \in \mathbb{R}^m$ from DFN158



- Neural Network Fully connected multi-headed, tree-shaped architecture, trunk and branches depth 3, 158 units $\times$ layer, softplus activation, Adam optimizer, early stopping with patience 150.

	$\mathcal{F}_8$	$\mathcal{F}_{12}$	$\mathcal{F}_{14}$	$\mathcal{F}_{78}$	$\mathcal{F}_{90}$	$\mathcal{F}_{98}$	$\mathcal{F}_{107}$
$D_{\mathrm{KL}}/\mathcal{E}$	0.0009	0.0003	0.0010	0.0002	0.0033	0.0379	0.0010

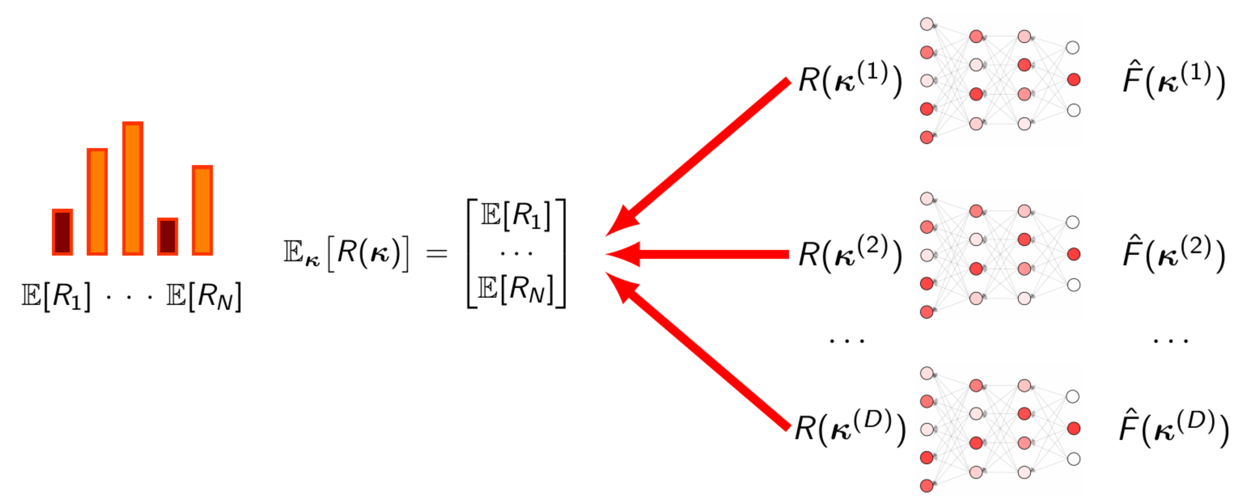
Table 1: Dissimilarity between ϕ, $\hat{\phi}$, actual and predicted outflux distributions; D_{KL}: KL divergence between ϕ, $\hat{\phi}$; $\mathcal{E}$: entropy of ϕ

LRP for Backbone Identification

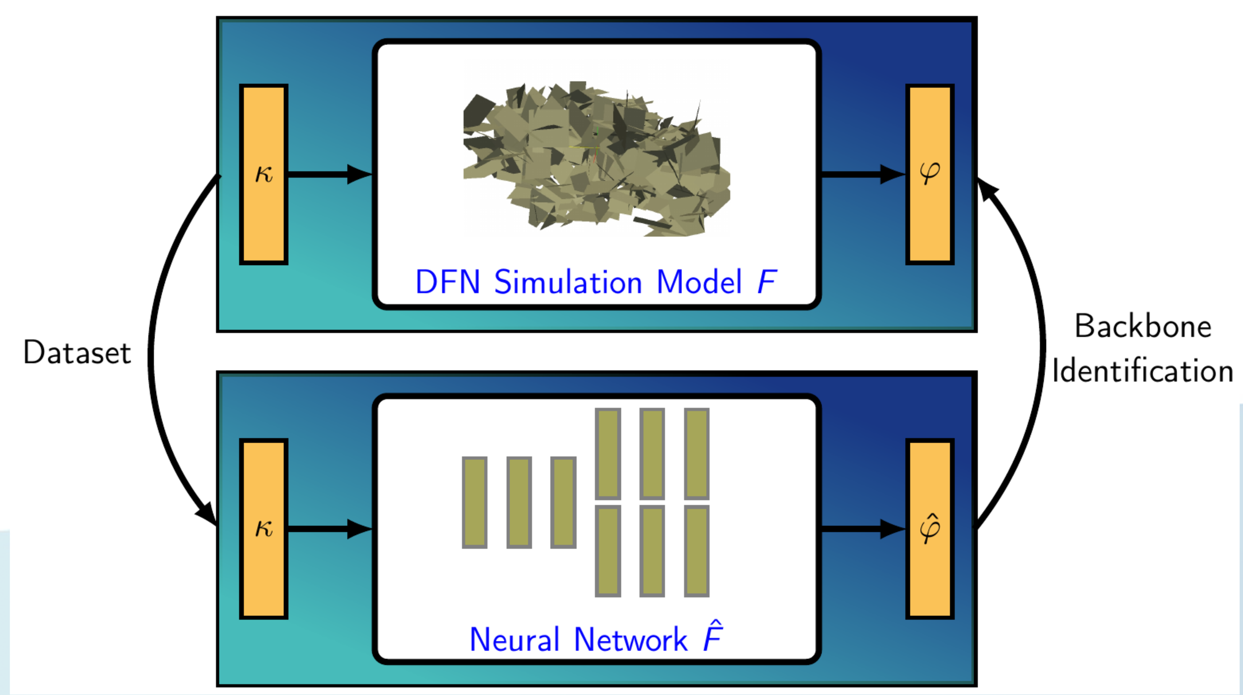
- **Local algorithm** of **eXplainable AI**
  Layer-wise Relevance Propagation (LRP) [Bach, 2015]:

$$R_i^{(\ell)} = \sum_{j \in (\ell+1)} R_{i \leftarrow j}^{(\ell, \ell+1)}, \qquad \text{neuron } i \in \ell \text{ layer.}$$

- Here extend to **global explanation**:
  Expected Relevance as a **feature selection** algorithm:



- Overall pipeline for **Backbone Identification**:



Results

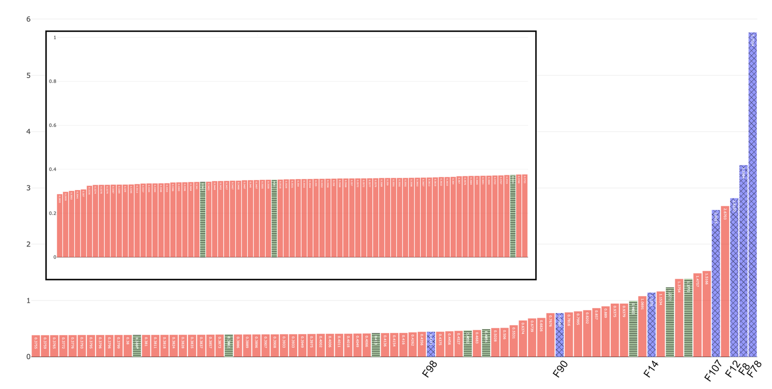


Figure 2: Fractures ordered by ascending value of r (top-left corner: lowest 60%); (blue) labelled outflow fractures; (green) inflow fractures.
Outflow fractures are in the top-25% of expected relevance
$\Rightarrow$ **NN approximates fluxes coherently with the DFN topology**.

Figure 3: Graphs of DFN158 (top-left) and Backbones with top expected relevance: 75% (top-right), 50% (bottom-left), 25% (bottom-right).
NN seems understanding that some bottleneck nodes are fundamental: a source-sink path is kept for the backbone top 25% expected relevance.