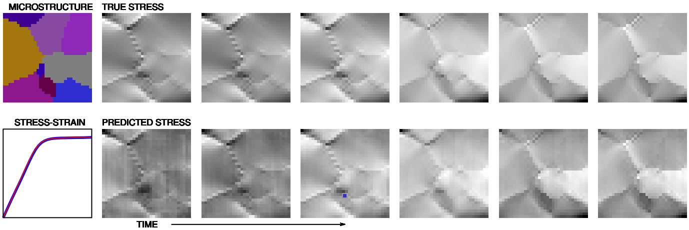
1716 Data-driven approaches in computational solid mechanics
15th World Congress on Computational Mechanics (WCCM-APCOM 2022)
31 July – 5 August 2022, Yokohama, Japan

Machine learning constitutive models of inelastic materials with microstructure

Reese E. Jones, A. Frankel, C. Safta, K.L. Johnson

Sandia National Laboratories, Livermore, CA 94551, USA



boratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LL
Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE

Overview

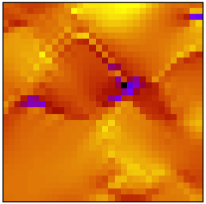
Motivation: constitutive models are the weakness of simulation

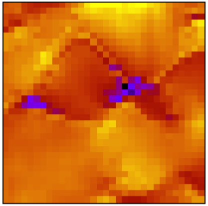
Goal: efficient accurate surrogate models of material processes

Everyone is doing machine learning, it is easy and sometimes useful.

- a paraphrase of George Box

Which one is the ML prediction?





one is a "deep fake"

Outline

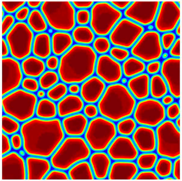
Problems of interest
Architectures
- A hybrid CNN-RNN
- Tensor basis NN
- Graph CNN-RNN
- ConvLSTM
- Neural ODE

Conclusion

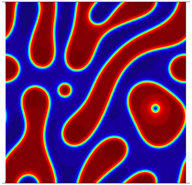
Please ask questions

Microstructural problems of interest: problem statement

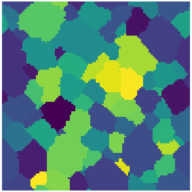
Premise: the state of each of these systems/processes can be encoded as an **image**/field with multiple **channels** $\phi(\mathbf{X})$.



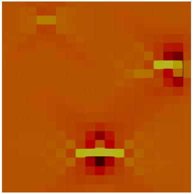
bubbles



multi-phase



polycrystal



pores/inclusions

Classes of closure problems:

- **property estimation**: map initial image $\phi(\mathbf{X})$ to a *static* quantity ε, e.g. diffusivity
- **homogenization**: map initial image $\phi(\mathbf{X})$ and forcing $\epsilon(t)$ to *evolving* scalar quantity $\Psi(t)$, e.g. energy
- **field prediction**: map initial image $\phi(\mathbf{X})$ and forcing $\epsilon(t)$ to an evolving *field* $\boldsymbol{\sigma}(\mathbf{X}, t)$, e.g. stress field

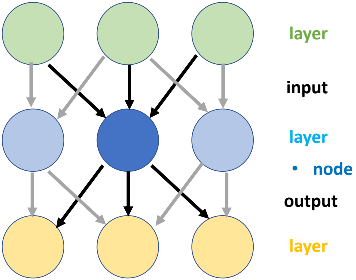
Applications: subgrid models, structure-property exploration /optimization, & material uncertainty quantification

Neural networks - basics/background

The simplest **neural network** (NN) is a *multilayer perceptron* (MLP), a directed graph of densely connected nodes organised in layers. Inputs are weighted, summed and transformed to outputs by *non-linear* ramp/switch-like activation functions.

$$y_j = f\underbrace{\left(\sum_i w_{ij} x_i + b_j\right)}_{\text{linear transform}}$$

The parameters w, b are trained via backpropagation and stochastic descent. NN are compounded trainable affine transforms with non-linear maps & can be compact universal approximators.



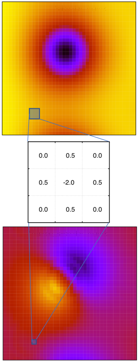
A NN is basically a functional form to be fit with chosen inputs, output, & information flow. Like LEGOS^{TM}, layers with particular characteristics can be linked to create architectures that follow physical principles & traditional modeling techniques.

Deep learning: convolutional neural networks

Direct application of a MLP to **image** data is impractical due to **size of weight matrix**. Application to a reduced set of features is problematic, as all informative **features may not be apparent**.

Convolution with a **kernel** is a standard technique in (time) signal and (spatial) image processing that has been adapted to ML.

Size of kernel $\ll$ size of image



For example, filters can:

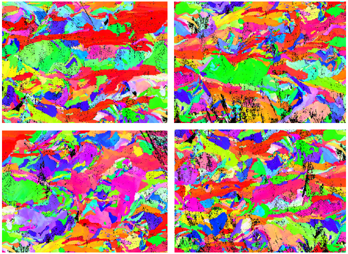
- **Smooth/filter noise:** convolving an image with a Gaussian kernel.
- **Average/coarsen:** multiplying with constant moving patch
- **Gradients and higher derivatives:** filter corresponding a finite difference stencil.
- **Features:** edge detection, clustering, segmentation, ...

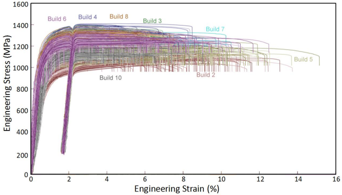
A convolutional NN trains the weights W and bias b for a (small) kernel and multiple filters/kernels can detect multiple hidden features.

Exemplar homogenization problems: images and responses

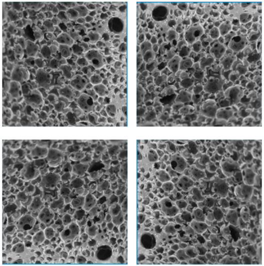
If we observe **initial** microstructures and mechanical tests:

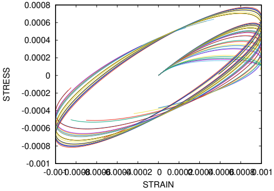
(A) polycrystal





(B) porosity

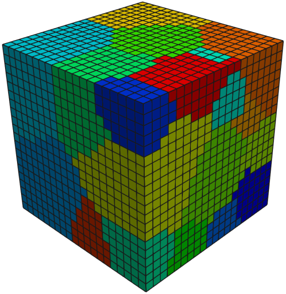


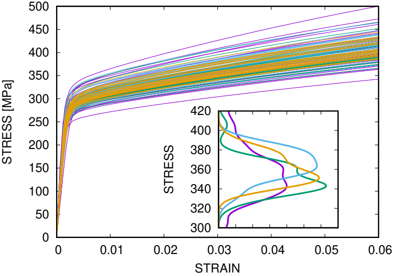


can we **predict** particular: (a) stress-strain averages or, (b) full field stress evolution

Challenge: training burden and sampling

Reasonably complex NNs need a lot of data to train well. Even with high-throughput tests we cannot currently generate more than $\approx 10^2$ tests, we need a dataset with $\approx 10^4$ samples for a NN. So, we resort to high-fidelity simulation data ☹





For example, we generate microstructural realizations of oligocrystals with different textures (crystal orientations) and run crystal plasticity simulations with a variety of loading modes.

Efficient, sufficient sampling for history dependent response is an **open question**.

A hybrid CNN-RNN network for time evolution

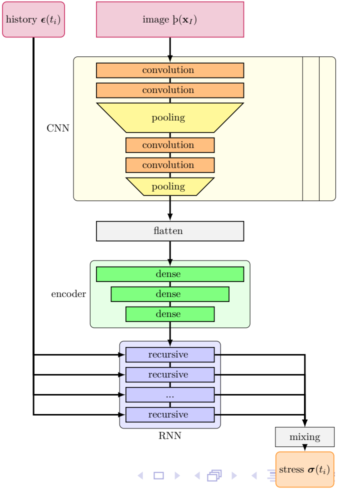
To predict a static property we can apply a **CNN** to (all channels of) a microstructure image. To predict the evolution of the average stress we augment the **CNN** with an **RNN** that models the **loading/time dependence** [Frankel *CompMatSci* 2019].

A recurrent NN (RNN) uses a causal time filter to process history/loading information (and the hidden image features). An RNN for time is an analog to the CNN for spatial data.

The (hidden) features distilled by the CNN feed into the RNN. The CNN output (post encoder) is only correlated with the observable stress through a RNN.

How many features should the CNN reduce the image to?

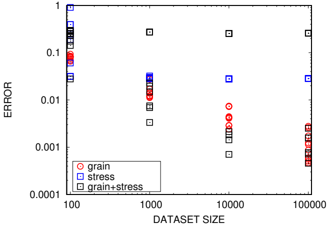
We are also exploring alternatives to RNN such as NODE based on traditional time integrators.



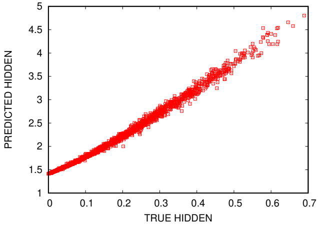
Predicting the response due to “hidden” features

Does the deep NN discover the hidden features?

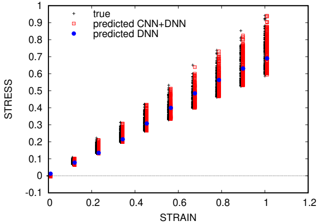
A **test problem** where we *know* what “hidden” microstructural features the observable stress depends on, e.g. average misorientation



Training NN on stress-strain alone stalls, but given initial **microstructures** continues to learn



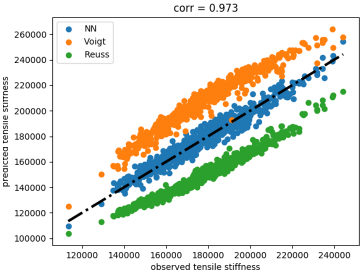
True and learned hidden feature are **highly correlated** - but not identical



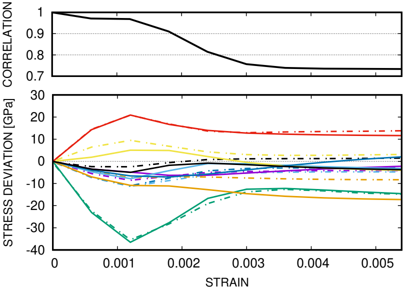
Observing the microstructure enables prediction of microstructure variations

Predicting the particular response to microstructure

Using data from the ensemble of polycrystals, we can make predictions of the crystal plastic mechanical response that are significantly better than traditional homogenization theory.



Correlation of elastic response (NN, Voigt and Reuss predictions), NN on par with Hill average.



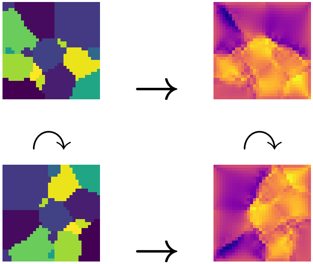
Trajectories of discrepancy from mean: solid lines data, dashed: NN prediction. Trajectories drift with accumulated error. Plastic response is better than Sachs or Taylor estimates.

Physical symmetries

Satisfaction of physical constraints and symmetries is expected in physical models and is necessary for conservation, stability, etc

How do we learn/impose physical constraints?

- **Augment** the dataset with many examples of what should happen, e.g. rotate the inputs and outputs (soft and inefficient)
- **Penalize** loss / training objective function (soft & introduces a meta parameter and can be hard to converge)
- **Embed** the symmetry in the NN architecture so that the response exactly preserves the symmetry (can be hard to formulate)



Objectivity and representation theory

We prefer to **embed symmetries** in the NN structure – so that they are exact/not learned. Let's go back to classical theory...

Material frame indifference for constitutive function $\mathbf{M}(\mathbf{A})$

$$\mathbf{G}\mathbf{M}(\mathbf{A})\mathbf{G}^T = \mathbf{M}(\mathbf{G}\mathbf{A}\mathbf{G}^T) \ ,$$

$\mathbf{M}$ model must commute with the symmetry op for every member $\mathbf{G}$ of the orthogonal group.

Based on the spectral $\mathbf{A} = \sum_{i=1}^{3} \lambda_i \mathbf{a}_i \otimes \mathbf{a}_i$, and Cayley-Hamilton theorems

$$\mathbf{A}^3 - \operatorname{tr}(\mathbf{A})\mathbf{A}^2 + \frac{1}{2}\left(\operatorname{tr}^2 \mathbf{A} - \operatorname{tr} \mathbf{A}^2\right) \mathbf{A} - \det(\mathbf{A})\mathbf{I} = \mathbf{0}$$

one can obtain a compact **general representation**/model form:

$$\boxed{\mathbf{M}(\mathbf{A}) = c_0(\mathcal{I})\mathbf{I} + c_1(\mathcal{I})\mathbf{A} + c_2(\mathcal{I})\mathbf{A}^2 = \sum_i c_i(\mathcal{I})\mathbf{A}^i}$$

in form of **unknown** coefficient functions of invariants and a **known** tensor basis. Inputs: scalar invariants $\mathcal{I}$ & tensor basis $\mathcal{B} = \{\mathbf{A}^0, \mathbf{A}^1, \mathbf{A}^2\}$.

A tensor basis neural network

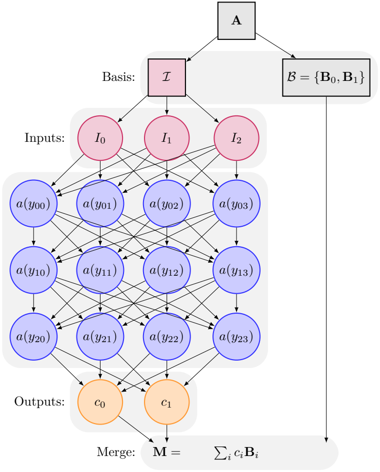
A **tensor basis neural network** is an NN implementation of this representation [Ling *JCP* 2016]: where the coefficients are unknown scalar functions of the invariants $\mathcal{I} = \{I_0, I_1, \ldots\}$

$$\mathbf{M} = \sum_i c_i(\mathcal{I})\mathbf{B}_i$$

and a final merge/sum layer associates c_i with the **tensor basis** $\mathcal{B} = \{\mathbf{A}^0, \mathbf{A}^1, \ldots\}$.

Effectively a MLP mapping invariants to coefficients + a sum with a known basis.

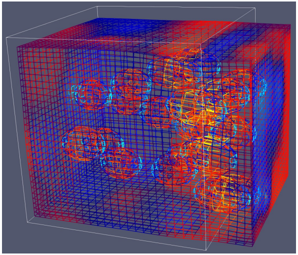
It is adept at representing the response with exact invariance / avoiding the need for data augmentation for symmetry.



A TBNN looks like a component based NN albeit with a basis constructed from the input.

Mesh data & graph-based convolutional neural networks

If we have microstructural images as inputs, CNNs work great for structured grid/rastered image but the **need interpolation** for mesh-based fields and do not inherently **satisfy invariance** $\mathbf{G}\boldsymbol{\sigma}(\epsilon, \phi)\mathbf{G}^T = \boldsymbol{\sigma}(\mathbf{G}\epsilon\mathbf{G}^T, \mathbf{G}\phi\mathbf{G}^T)$ where ϕ is the initial microstructure.

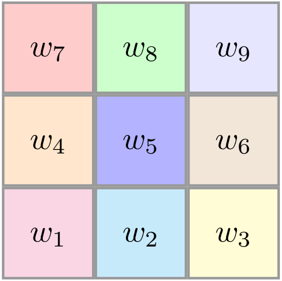


Reducing the grains to nodes and shared interfaces to edges has been shown effective [Vlassis 2020] However this approach loses information (eg the details of the grain and interface geometry) and hence **requires featurization**.

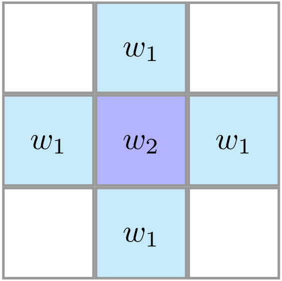
We have applied graph convolutions directly to the mesh topology [Frankel *JMLMC* 2021]. This approach does not require featurization but can benefit from it. It does not increase the number of parameters since the same kernels are being employed.

Graph-based convolutional neural networks

Graph based convolution layers/filters [Kipf & Welling 2016] can be applied directly to the graph based on the mesh topology : **elements** are graph **nodes** and shared **faces** are graph **edges**.



CNN filter

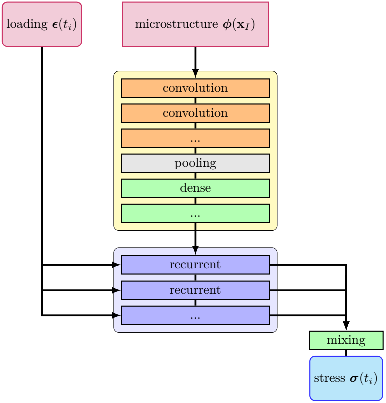


GCNN filter

The GCNN filter uses the same weights for all the neighbors (permutational invariance), hence it produces the same output when the image is rotated.

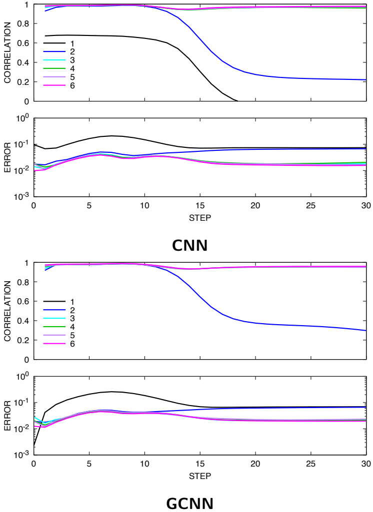
GCNNs vs. CNNs

GCNNs have similar performance to CNN with fewer parameters and inherent invariance.



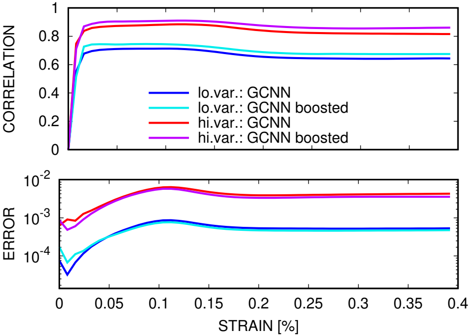
GCNN-RNN

Convergence with number of filters



Feature boosting

GCNNs (and CNNs) can be **boosted** by **embedding** obvious **features** into the image (or further down the CNN-RNN pipeline)



adding node volumes to image of orientation angles

The improvement is marginal but distinct for a NN that is already fairly accurate.

Full field predictions: convLSTM

An architecture similar to the CNN-RNN we used to predict system-level evolution can be used to predict **full-field** (element/pixel level) evolution.

Inputs: pairs of

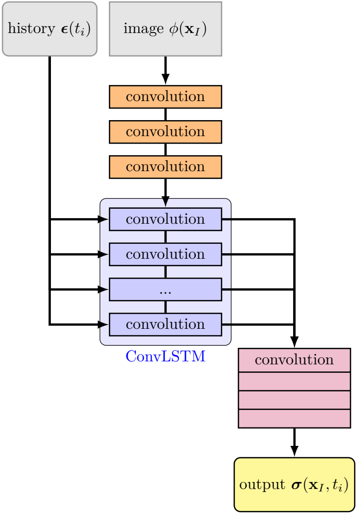
- $\phi(\mathbf{X})$: image of initial microstructure
- $\epsilon(t)$: system level strain history

The image is fed to a *convolutional* neural network to process its latent features but not reduce them to a list of scalars – **each layer/filter output is also an image** so that spatial relationships are preserved.

This initial condition-like input is combined with the strain history in a *recurrent-convolutional* neural network, a convLSTM [Shi *NIPS* 2017].

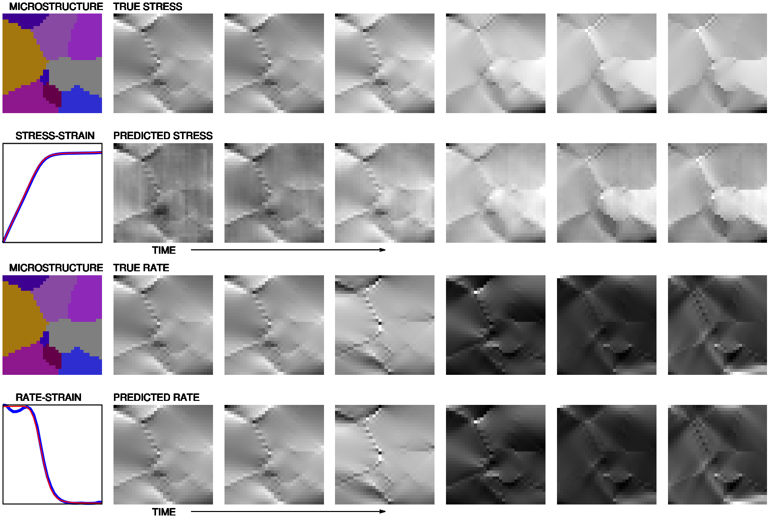
The output of the convLSTM is processed by another CNN unit to produce

Output: $\boldsymbol{\sigma}(\mathbf{X}, t)$ full field stress evolution



Full field predictions

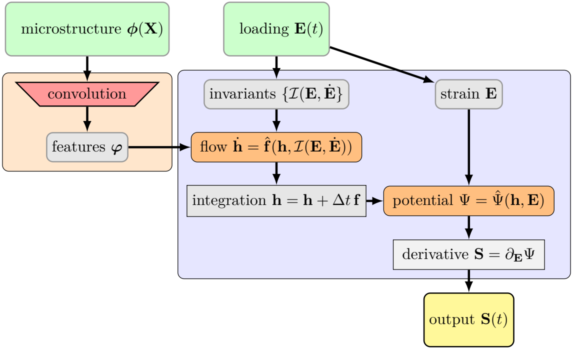
A convLSTM combines the RNN (time) and CNN (space) into PDE-like model



[FRANKEL *MLSciTech* 2020]

An internal state variable neural ODE model

Premise: it is better to infer state variables, like damage, than prescribe them. So we augment the observable state with hidden states that are learned.



Stress

$$\mathbf{S} = \mathsf{NN}_{\mathbf{S}}(\mathbf{h}, \mathbf{E})$$

Flow

$$\dot{\mathbf{h}} = \mathsf{NN}_{\mathbf{h}}(\mathbf{h}, \mathbf{E})$$

RNN are locked into a particular time step. NODE have the same sense of time scaling as the usual dynamical models and employ the same time integrators.

Model variants and accuracy

There are multiple ways of formulating a general stress response:

- potential, as in thermodynamics

$$\mathbf{S} = \partial_{\mathbf{E}} \mathsf{NN}(\mathbf{E}, \mathbf{h})$$

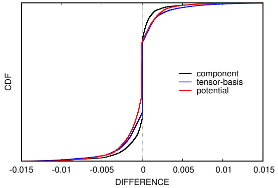
- equivariant tensor basis

$$\mathbf{S} = \sum_i \mathsf{NN}_i(\mathbf{E}, \mathbf{h}) \mathbf{B}_i(\mathbf{E})$$

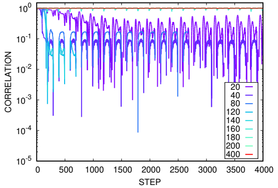
- components of a fixed basis

$$\mathbf{S} = \sum_i \mathsf{NN}_{(ij)}(\mathbf{E}, \mathbf{h}) \mathbf{B}_{(ij)}$$

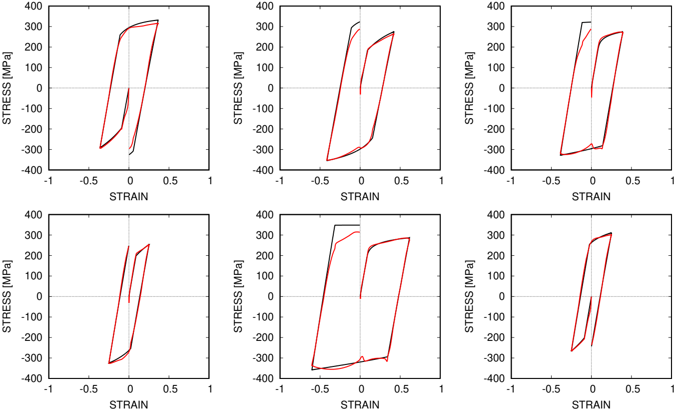
CDF of errors for TB, potential, component



Time extrapolation with sequential training



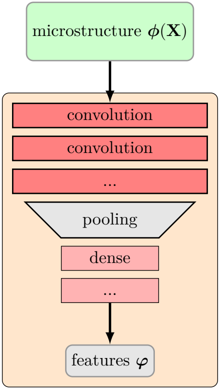
Elastoplasticity



Even without an explicit yield surface, the NODE seems to be learning the non-smooth flow field.

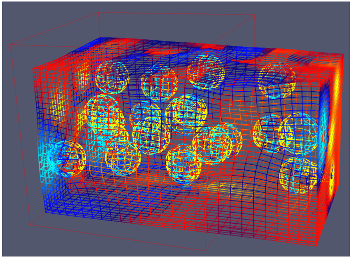
Graphs for microstructure + ODEs for evolution

We can combine a NODE & a Graph CNN to reduce the initial microstructures to latent features

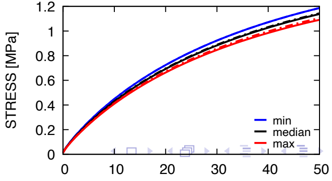


that become additional hidden state variables in the NODE flow evolution.

Microstructures with pores or hard inclusions



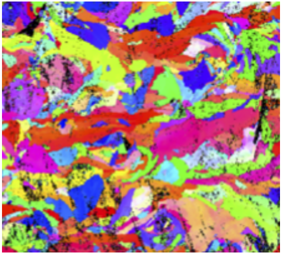
Predictions vs. truth for min, mean, max error



Conclusion

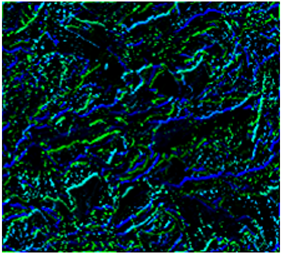
Applications:

- subgrid / multiscale surrogate models
- structure-property exploration / material optimization
- material uncertainty quantification



Open issues:

- architecture / meta parameter optimization
- interpretability (latent space / low dimensional manifold)
- training burden / multifidelity (experimental+simulation) data



rjones@sandia.gov

References: **rjones@sandia.gov**

- J Ling, RE Jones, JA Templeton. *Machine learning strategies for systems with invariance properties* JCompPhys (2016).
- RE Jones, JA Templeton, CM Sanders, JT Ostien. *Machine learning models of plastic flow based on representation theory* CompModEngSci (2018)
- AL Frankel, RE Jones, C Alleman, JA Templeton *Predicting the mechanical response of oligocrystals with deep learning* CompMatSci (2019)
- AL Frankel, K Tachida, RE Jones *Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model* MachLearn:SciTech, (2020)
- AL Frankel, RE Jones, L Swiler *Tensor Basis Gaussian Process Models of Hyperelastic Materials* JMachLearnModComp (2020)
- AL Frankel, C Safta, C Alleman, RE Jones, *Mesh-based graph convolutional neural network models of processes with complex initial states* JMachLearnModComp, (2021)
- RE Jones, AL Frankel, KL Johnson *A neural ordinary differential equation framework for modeling inelastic stress response via internal state variables*, JMachLearnModComp (2021)
- W Bridgman, X Zhang, G Teichert, M Khalil, K Garikipati, RE Jones, *A heteroencoder architecture for prediction of failure locations in porous metals using variational inference*, CMAME (2022)
- JN Fuhg, N Bouklas, RE Jones. Learning hyperelastic anisotropy from data via a tensor basis neural network. arXiv (2022).

... new work on graph reduction forthcoming.