Deep Interval Neural Network in Computational Solid Mechanics

By: David Betancourt, PhD Student CSE-CEE
and Rafi Muhanna
Reliable Engineering Computing - Georgia Tech

Georgia Tech

Georgia Tech Computational Science and Engineering
College of Computing

Georgia Tech School of Civil and Environmental Engineering
College of Engineering

Presentation Outline

- Problem Background
  - Computational Solid Mechanics in Civil Engineering
  - FEM
  - Uncertainty in FEM inputs
  - Interval Uncertainty
- Interval Fields
- Deep Interval Neural Network

Georgia Tech

Background

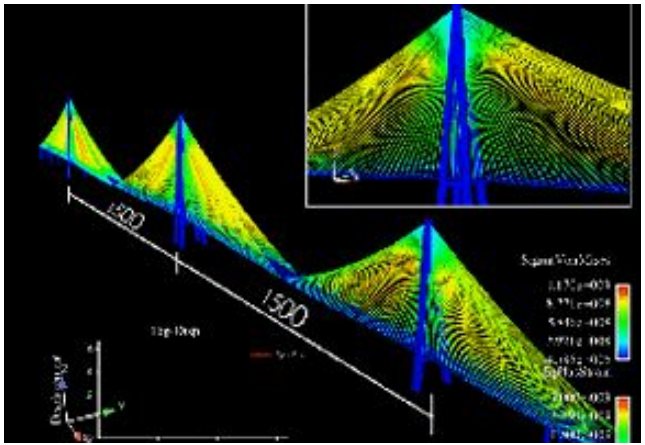
Computational Solid Mechanics in Civil Engineering

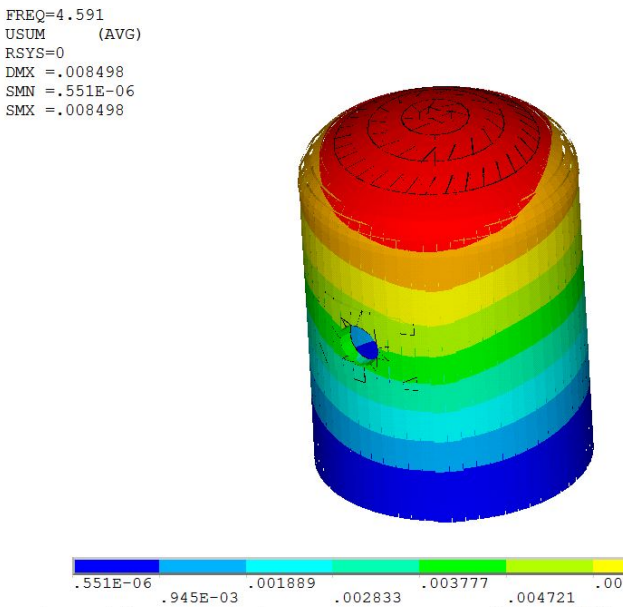
- The *system* consists of a structure or domain that undergoes some applied stress or vibration.

Photo credit:
Archinect

Computational Solid Mechanics in Civil Engineering

- The finite element method (FEM) is typically used in computational solid mechanics to find the system's response.



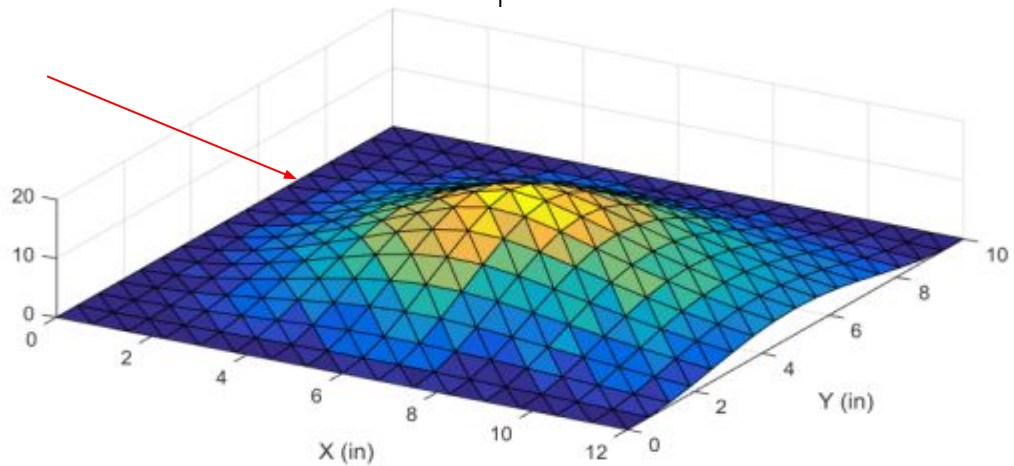


FEM input and outputs

- Inputs:
  - Material properties
  - Support conditions
  - Applied pressure

- Outputs:
  - Displacements
  - Stresses
  - Vibration frequencies

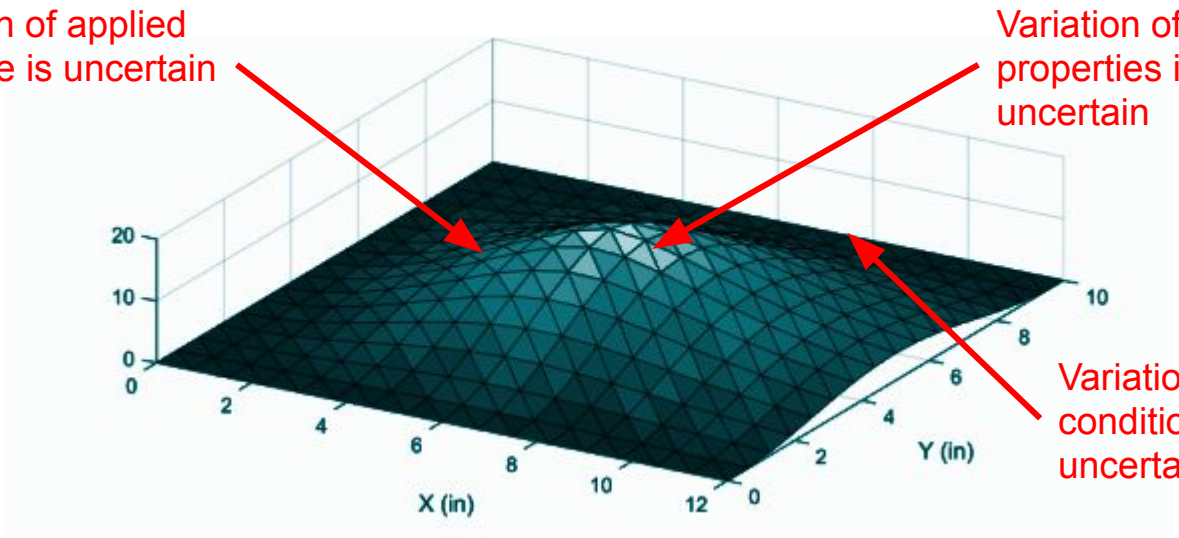
FE Mesh



Georgia Tech

The research problem: *Quantifying spatiotemporal uncertainty in systems inputs*

- Many times in civil engineering we lack enough knowledge about the system inputs.



Georgia Tech

Uncertainty in civil engineering systems

Not modeling uncertainty properly can lead to:

- Catastrophic events.
- Inaccurate expectations of systems performance.
- Increased engineering schedules and costs.

In the context of computational mechanics:

- Uncertainty can be spatially-varying and/or temporally varying.
  - Loadings, materials, boundary conditions.

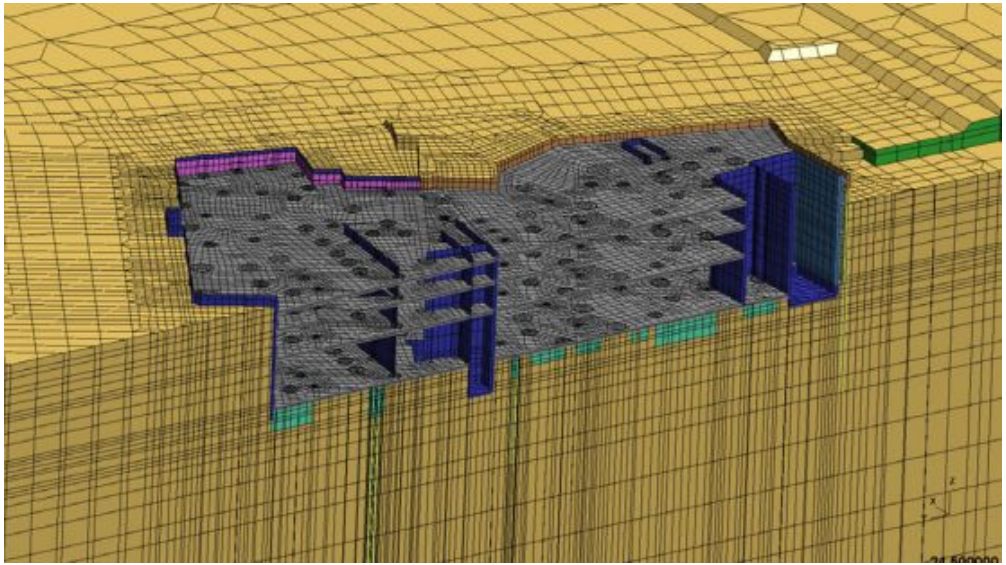
Georgia Tech

Uncertainty in civil engineering systems

Uncertainty in civil engineering systems

Uncertainty in civil engineering systems (SFEM)

- In computational mechanics:
  - Uncertainty is commonly modeled with with random field theory within the Stochastic Finite Element method (SFEM).
  - SFEM makes assumptions about the probability distribution of the uncertainty, **usually Gaussian.**



Georgia Tech

Uncertainty in engineering systems (SFEM vs IFEM)

- In computational mechanics:
  - When we don't have enough data to select a probability distribution → SFEM is inadequate. Is the world Gaussian?
  - Interval Finite Element method (IFEM) is an alternative.
  - Efficient algorithms by Muhanna & Mullen.
- The IFEM:
  - Does not make assumptions about probability distributions (only UB and LB)
  - Can model epistemic and aleatory uncertainty.
- BUT the IFEM:
  - Does not have an "interval field" that can model spatiotemporal-varying uncertainty as is the case with SFEM.

Georgia Tech

IFEM Process

Obtain the Response

Obtain primary responses: displacements, stresses, modes of vibrations.

Obtain secondary responses: moments, failure modes.

IFEM

Discretize the structure, perform interval finite element analysis to find the structural responses with interval enclosure.

Quantify Input Uncertainty

Model the spatiotemporal variation of the system input.

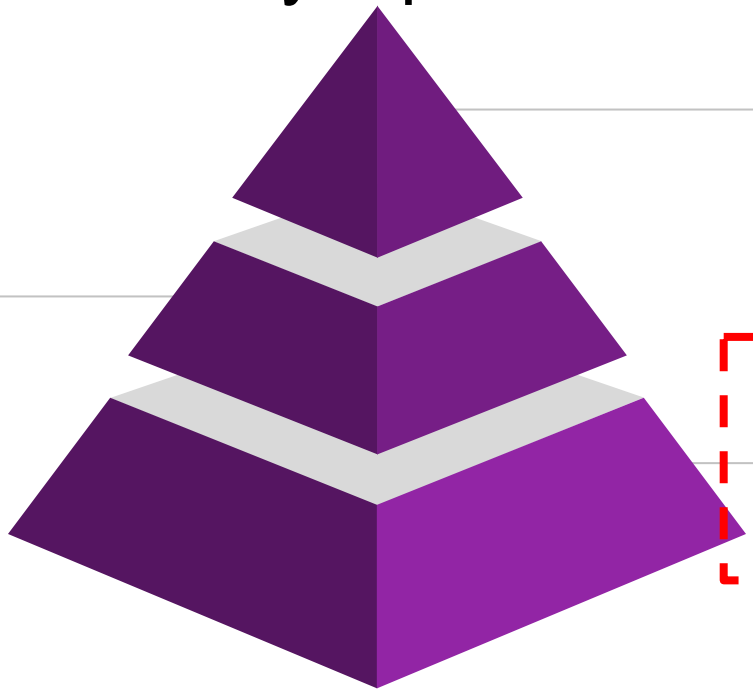
Interval Field

Supervised Interval Field

- IFEM:
  - Does not have a unified interval field framework that can model spatiotemporal-varying uncertainty in the model input as in the case with SFEM.
- Solution:
  - Use deep neural network to infer the uncertain properties in the field from indirect measurements:
    - ***Called: "Supervised Interval Field"***
  - Works for any domain dimension (1D, 2D, 3D).
  - Independent of IFEM mesh discretization.

Georgia Tech

Prior Research: Quantify Input Variability



Obtain the Response

3

Obtain primary responses: displacements, stresses, modes of vibrations.

Obtain secondary responses: moments, failure modes.

IFEM

Discretize the structure, perform interval finite element analysis to find the structural responses with interval enclosure.

2

Quantify Input Uncertainty

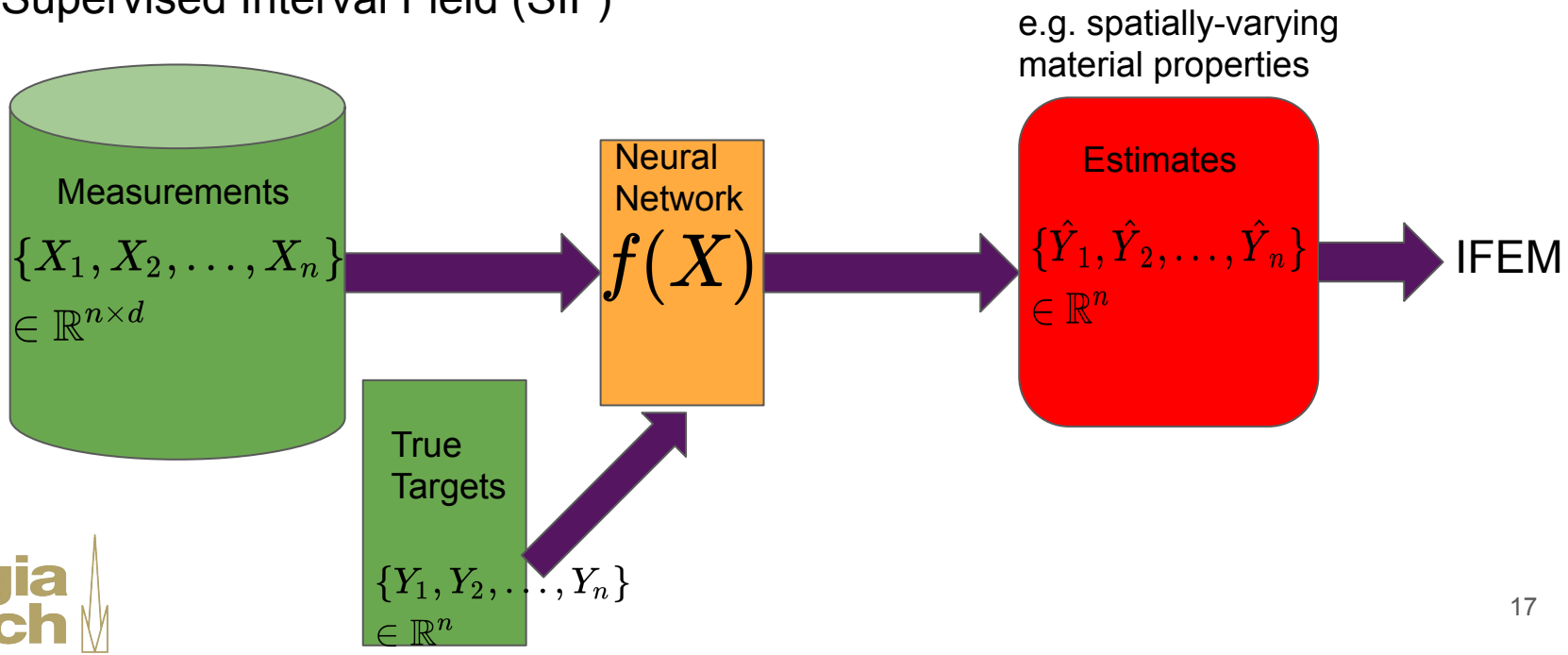
1

Model the spatiotemporal variation of the system input.

Georgia Tech

Prior Research: Quantify Input Variability

- Quantifying the variability of a property Y using observations X.
- Supervised Interval Field (SIF)

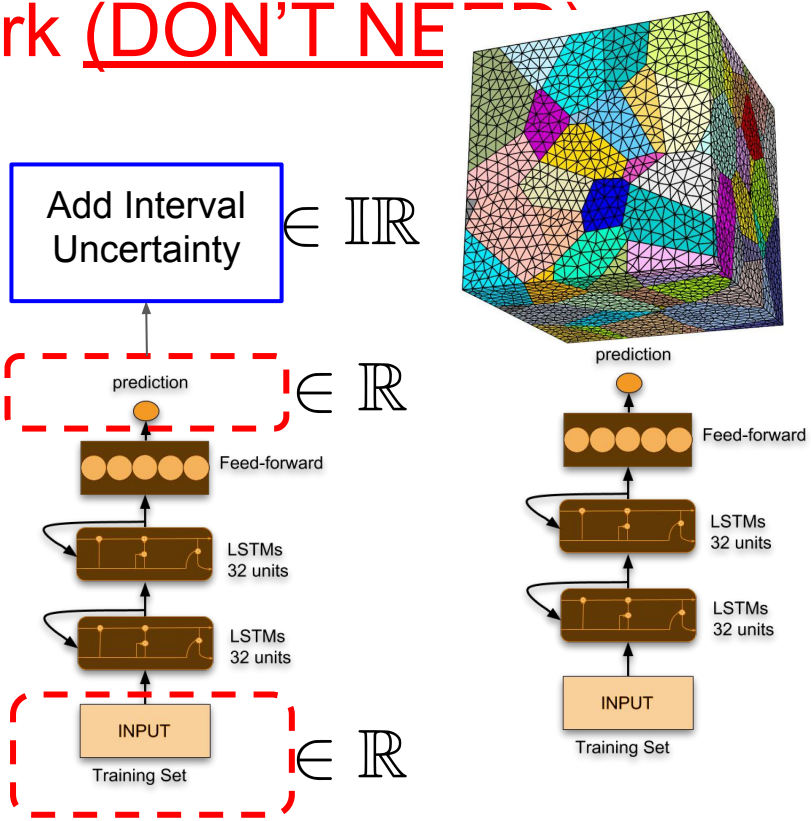


Supervised Learning

- Supervised Learning:
  - Train/Fit model $f : X \rightarrow Y$.
  - After model is trained:
    - Make predictions on unseen data.
- Goal of Supervised Learning:
  - Not overfit.
  - Generalize to unseen data.

SIF-IFEM a unified framework (DON'T NE

- SIF provides to the IFEM the uncertain property with respect to the model's mesh.
- SIF is independent of mesh, therefore, must discretize and map into IFEM mesh.
- Mid-point interval value from SIF prediction assigned to each FE in the mesh.
- Average of the prediction taken over the size of FE.



Deep Interval Neural Network

Deep Interval Neural Network

Problem with real-valued neural network:

- Not able to propagate interval-valued inputs nor make interval-valued predictions.

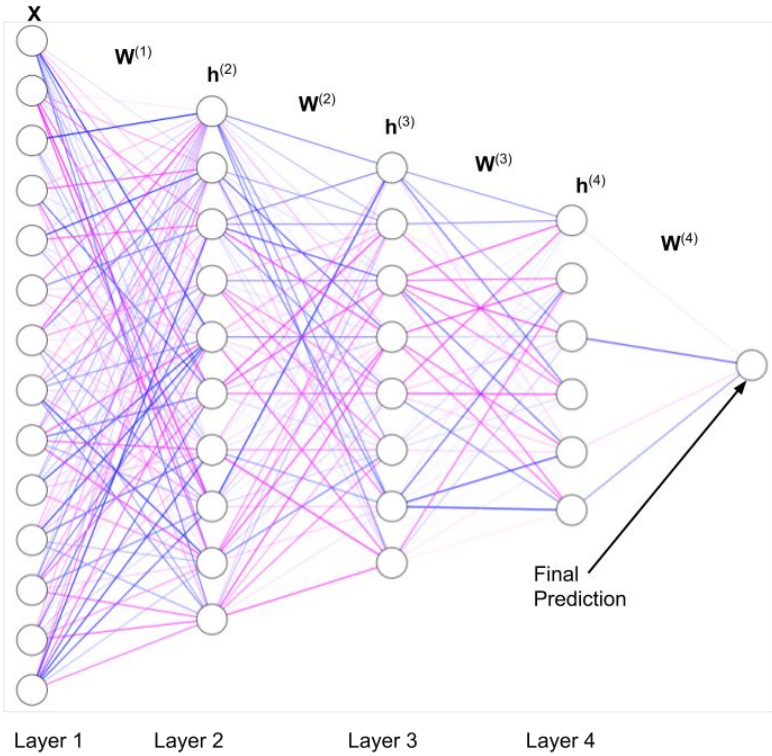
Solution:

- Deep Interval Neural Network *using Gradient-Based Optimization*
    - Developed by David Betancourt and Dr. Rafi Muhanna.

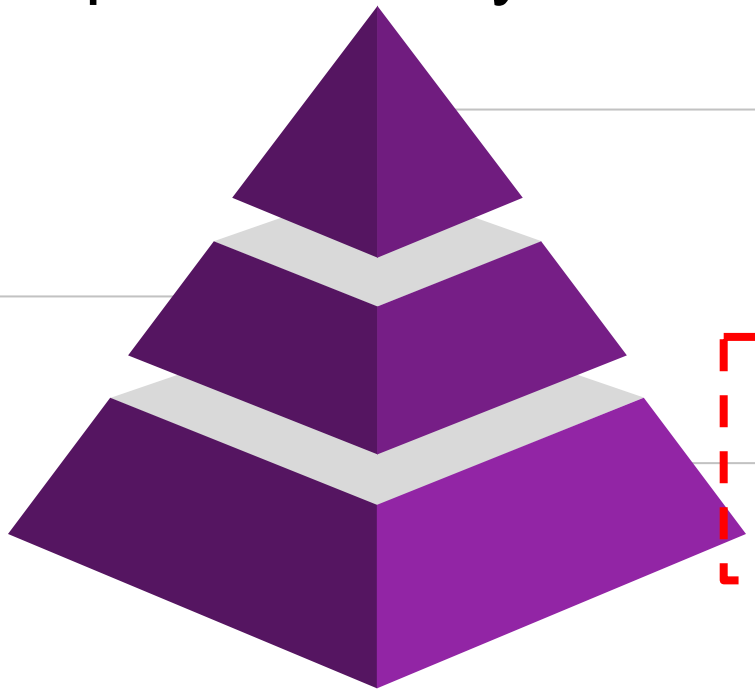
Georgia Tech

Deep Interval Neural Network (DINN)

- The problem (computational mechanics):
  - Need to model input uncertainty to IFEM.
    - e.g. space-varying material properties.
- Solution:
  - DINN takes interval matrices as input and outputs interval predictions in a regression setting to form the *interval field*.



DINN: Quantify Input Variability and Uncertainty



Obtain the Response

3 Obtain primary responses: displacements, stresses, modes of vibrations.

Obtain secondary responses: moments, failure modes.

IFEM

Discretize the structure, perform interval finite element analysis to find the structural responses with interval enclosure. 2

Quantify Input Uncertainty

1 Model the spatiotemporal variation of the system input.

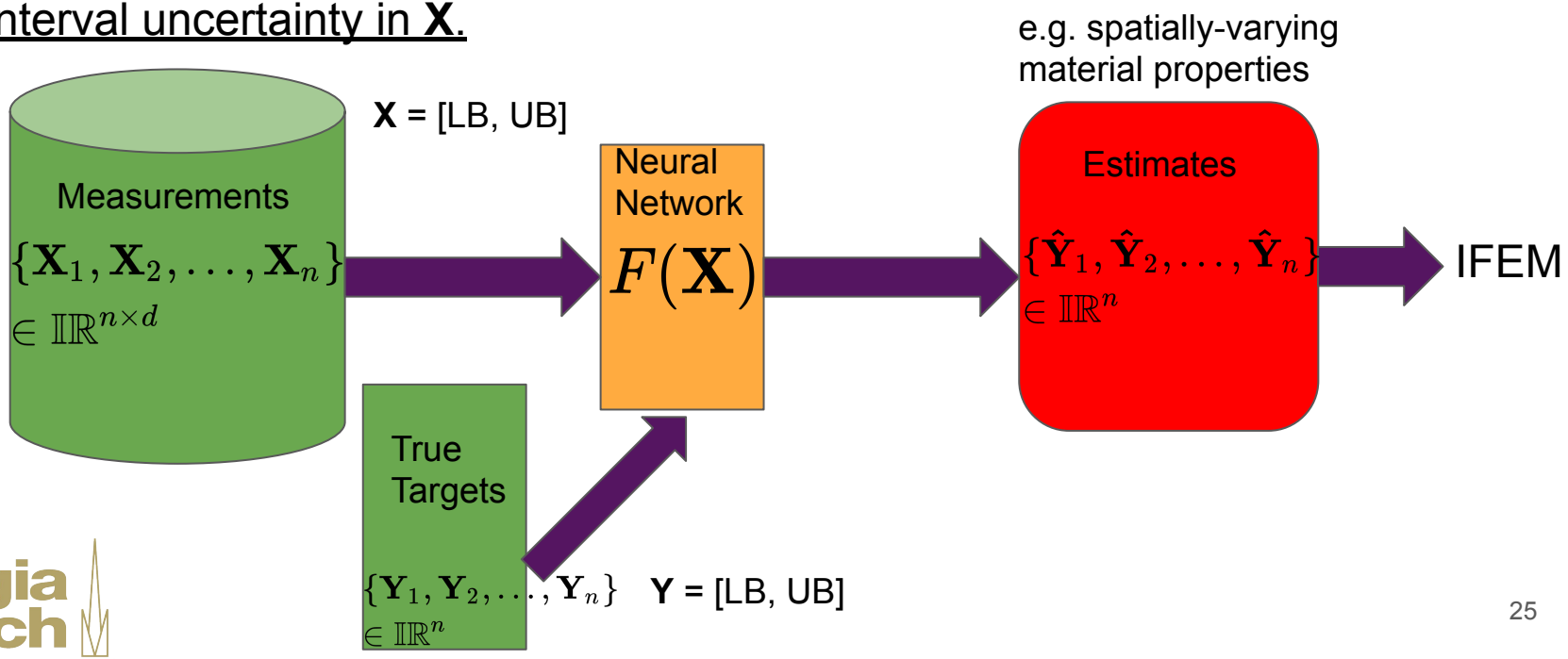
Intervals Definition

Let $\mathbb{IR}$ be the set of interval real numbers. Interval real numbers are a closed and bounded set of real numbers, s.t. if $\mathbf{X} \in \mathbb{IR}$ is an interval, its endpoints are

$$\underline{X} = \inf(\mathbf{X}) \in \mathbb{R} \text{ and}$$
$$\overline{X} = \sup(\mathbf{X}) \in \mathbb{R}.$$

Current Research: Quantify Input Variability and Uncertainty

- Quantifying the variability of a property **Y** using observations **X**, <u>considering interval uncertainty in **X**.</u>



Georgia Tech

Computational Aspects

Part II

What is the challenge with interval matrices in neural networks?

- For NN → Need a well-behaved loss function (Lipschitz continuous).
- For Intervals → Need to reduce interval dependency.
    - Interval enclosure gets too wide when we have repeated expressions.
        - E.g. $\mathbf{X}\,\mathbf{X} \mathrel{!}= \mathbf{X}^2$

Interval Dependency

- Interval dependency is the overestimation of the computation of a bound.
- Occurs when interval variables appear more than once in an interval expression.
- Major consideration when designing algorithms with intervals.
- Cannot simply replace floating point computations with intervals.

- Interval multiplication:

Case	$\underline{X \cdot Y}$	$\overline{X \cdot Y}$
$0 \le \underline{X}$ and $0 \le \underline{Y}$	$\underline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ and $0 \le \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \le 0$ and $0 \le \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \underline{Y}$
$0 \le \underline{X}$ and $\underline{Y} < 0 < \overline{Y}$	$\overline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \le 0$ and $\underline{Y} < 0 < \overline{Y}$	$\underline{X} \cdot \overline{Y}$	$\underline{X} \cdot \underline{Y}$
$0 \le \underline{X}$ and $\overline{Y} \le 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ and $\overline{Y} \le 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \underline{Y}$
$\overline{X} \le 0$ and $\overline{Y} \le 0$	$\overline{X} \cdot \overline{Y}$	$\underline{X} \cdot \underline{Y}$
$\underline{X} < 0 < \overline{X}$ and $\underline{Y} < 0 < \overline{Y}$	$\min\{\underline{X}\overline{Y}, \overline{X}\underline{Y}\}$	$\max\{\underline{X}\underline{Y}, \overline{X}\overline{Y}\}$

Georgia Tech

Interval Dependency - Monotonic Functions

- Choose functions that are monotonic so that we can compute tight bounds.

$$\mathbf{X}^2 = [\underline{X}^2, \overline{X}^2]$$

$$\exp(\mathbf{X}) = [\exp(\underline{X}), \exp(\overline{X})]$$

$$\mathbf{X}^{\frac{1}{2}} = [\underline{X}^{\frac{1}{2}}, \overline{X}^{\frac{1}{2}}] \text{ for } 0 < X_0$$

- Interval multiplication:

Case	$\underline{X \cdot Y}$	$\overline{X \cdot Y}$
$0 \le \underline{X}$ and $0 \le \underline{Y}$	$\underline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ and $0 \le \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \le 0$ and $0 \le \underline{Y}$	$\underline{X} \cdot \overline{Y}$	$\overline{X} \cdot \underline{Y}$
$0 \le \underline{X}$ and $\underline{Y} < 0 < \overline{Y}$	$\overline{X} \cdot \underline{Y}$	$\overline{X} \cdot \overline{Y}$
$\overline{X} \le 0$ and $\underline{Y} < 0 < \overline{Y}$	$\underline{X} \cdot \overline{Y}$	$\underline{X} \cdot \underline{Y}$
$0 \le \underline{X}$ and $\overline{Y} \le 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \overline{Y}$
$\underline{X} < 0 < \overline{X}$ and $\overline{Y} \le 0$	$\overline{X} \cdot \underline{Y}$	$\underline{X} \cdot \underline{Y}$
$\overline{X} \le 0$ and $\overline{Y} \le 0$	$\overline{X} \cdot \overline{Y}$	$\underline{X} \cdot \underline{Y}$
$\underline{X} < 0 < \overline{X}$ and $\underline{Y} < 0 < \overline{Y}$	$\min\{\underline{X}\overline{Y}, \overline{X}\underline{Y}\}$	$\max\{\underline{X}\underline{Y}, \overline{X}\overline{Y}\}$

Georgia Tech

Interval Lipschitzness

- Interval functions are Lipschitz continuous if they are *united extensions* of a real-valued function.
- Examples:

$$\mathbf{X}^2 = [\underline{X}^2, \overline{X}^2]$$

$$\exp(\mathbf{X}) = [\exp(\underline{X}), \exp(\overline{X})]$$

$$\mathbf{X}^{\frac{1}{2}} = [\underline{X}^{\frac{1}{2}}, \overline{X}^{\frac{1}{2}}] \text{ for } 0 < X_0$$

- Choosing a real-valued Lipschitz function guarantees a Lipchitz interval function!

Georgia Tech

Important Property: Isotonic Inclusion of Interval Functions

- Isotonic inclusion guarantees that an interval extension function *F* contains the exact upper and lower bounds of a real-valued function *f.*
- Guaranteed bounds but still need to reduce interval dependency.

DINN

- DINN is a predictive model *F* in regression setting:

$$F : [\underline{X}, \overline{X}] \rightarrow [\underline{Y}, \overline{Y}] \in \mathbb{IR}$$

- NP-hard for intervals.
- To fit (train) the DINN:
  - Need a training dataset with n examples.
  - Each example has *d* dimensionality

$$\mathcal{T} = \{(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \ldots, (\mathbf{X}_n, \mathbf{Y}_n)\}$$

- Learning of function **Y** by minimizing a loss (error) function

$$\mathcal{L}(\hat{F}(\mathbf{X}_i), \mathbf{Y}_i; \mathbf{W})$$

  - Mean Square Error for regression (monotonic!)

Georgia Tech

DINN: training set

$$\mathcal{T} = \{(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \ldots, (\mathbf{X}_n, \mathbf{Y}_n)\}$$

Features

Targets

	$\mathbf{X}_{i,1}$	$\mathbf{X}_{i,2}$	...	$\mathbf{X}_{i,d}$	$\mathbf{Y}_i$
example 1	[100.2, 110.3]	[250.1, 268.8]	...	[83.4, 85.7]	[31,33.1]
example 2	[93.7, 95.1]	[260.1, 271.1]	...	[72.1, 73.7]	[25.1, 26.7]
...	...	...	...	...	...
example n	[85.3, 88.1]	[240.4, 255.7]		[78.8, 79.6]	[35.1, 36.1]

$$\mathbf{X} \in \mathbb{R}^{n \times d} \qquad \mathbf{Y} \in \mathbb{R}^{n}$$

Georgia Tech

DINN Training

- Steps for training the network:
  a. Compute the loss at each training epoch via forward propagation.
  b. Compute the gradient of the loss w.r.t. The model parameters **W** via Backpropagation.
  c. Minimize the loss function via mini batch stochastic gradient descent.

Forward Propagation Algorithm

Algorithm 1: Foward Propagation Algorithm for Each Mini-batch of Data

Obtain the input raw features X

Embed X into interval input features $\mathbf{X}$ considering lower $\underline{X}$ and upper bounds $\overline{X}$ for each sample

Choose number of layers L

Require model parameters $\mathbf{W}^l$, for each layer l

Obtain the ground-truth interval output targets $\mathbf{Y}$

$\mathbf{h}^{(0)} = \mathbf{X}$

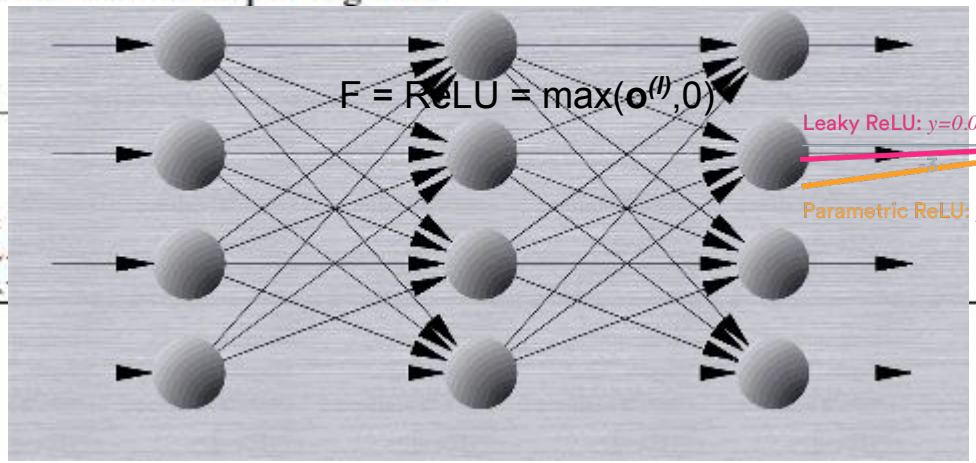
for $l = 1 : L$ **do**

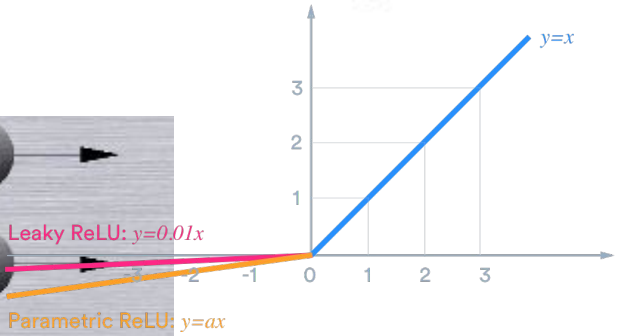
  $\mathbf{o}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-}$

  $\mathbf{h}^{(l)} = F(\mathbf{o}^{(l)})$

Prediction $\hat{F}(\mathbf{X}) =$

Compute loss $\mathcal{L}(\hat{F}($





Backpropagation Algorithm

Algorithm 2: Backpropagation Algorithm for Interval Loss for each mini-batch of data

Run forward computation algorithm

$\hat{F}(X) = \hat{Y}$

$\mathcal{G} \leftarrow \boldsymbol{\delta}_{out}^{(L)}$

for $l = L, L-1, \ldots, 1$ *layers* **do**

$\mathcal{G} \leftarrow \boldsymbol{\delta}_{a}^{(l)} = \mathcal{G} \odot \nabla_{\mathbf{h}} F(\mathbf{o}^{(l)})$

$\nabla_{\mathbf{W}^{(l)}} \mathcal{L} = \mathcal{G}\mathbf{h}^{(l-1)} + \lambda \nabla_{\mathbf{W}^{(l)}} \Omega(\mathbf{W})$

$\nabla_{\mathbf{b}^{(l)}} \mathcal{L} = \mathcal{G} + \lambda \nabla_{\mathbf{b}^{(l)}} \Omega(\mathbf{W})$

$\mathcal{G} \leftarrow \boldsymbol{\delta}_{h}^{(l-1)} = \mathbf{W}^{(l)T} \mathcal{G}$

Georgia Tech

Mini-batch SGD

- Optimization is done via stochastic gradient descent
  - Update rule:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \nabla_{\mathbf{W}} \mathcal{L}(\hat{F}(\mathbf{X}_i), \mathbf{Y}_i)$$

Regularization

- Prevents overfitting
- Stabilizes solution
  - Very important for intervals
- Currently using L1 and L2 regularization

Experiments: UCI Concrete Dataset

Solver	R2 Coefficient	Training time
GD	0.85	4.3 hrs
SGD	0.83	20 min
GD + Momentum	0.95	3.2 hrs
SGD + Momentum	0.95	16 min

- Network Hyperparameters:
  - Hidden Dimensions: H1 = 128, H2 = 64, H3 = 16.
  - Mini-batch size = 10
  - Learning Rate = 0.005
  - L2 regularization.
  - Features: 8 chemical/physical measurements.
  - Targets: Concrete Strength.

Georgia Tech

Experiments: UCI Concrete Dataset

- Input uncertainty: input x 10e-3 radius
- Predicted uncertainty:
  - Mean upper bound: 20% of target
  - Mean lower bound: 15% of target

Georgia Tech

Future work

- Structural Health Monitoring data
  - Forward problem
  - Inverse problem
- Deep Interval Recurrent Neural Network with LSTM units
  - For sequences of data involving long-term dependencies.
- Different regularization methods
  - Dropout
- Decision-making under uncertainty for autonomous systems:
  - Reinforcement Learning

Georgia Tech

Thank you!

Twitter: @davidbtncourt

Georgia Tech