DEEP LEARNING FOR MANUFACTURING

Predicting and Preventing Manufacturing Defects

Sumit Sinha
Digital Lifecycle Management (DLM)
WMG, University of Warwick

WMG
THE UNIVERSITY OF WARWICK

TABLE OF CONTENTS

FRAMEWORK

Goal

Intelligent & Automated Root Cause Analysis (RCA) for multi-stage production/assembly systems

Analysis of a system: Forward Propagation

CAE Simulation

$$X \longrightarrow \boxed{\begin{array}{c}\textbf{System}\\ Y = F(X)\end{array}} \longrightarrow Y$$

X: Takes as input a set of control parameters & incoming material

Y: Gives as output a system response or component

T: Constraints such as specification limits

Estimating $\boldsymbol{F}$ enables:

- Process Parameter Optimization
- Fixture Optimization
- Tolerance Analysis

Synthesis of a system: Backward Propagation

Artificial Intelligence: Deep Learning

$$X \longleftarrow \boxed{\begin{array}{c}\textbf{System}\\ X = F^{-1}(Y)\end{array}} \longleftarrow Y$$

X: Input control parameters have to be estimated

Y: Given the output is known

T

Estimating $\boldsymbol{F^{-1}}$ enables:

- Root Cause Analysis
- Process Control and Correction
- Tolerance Synthesis

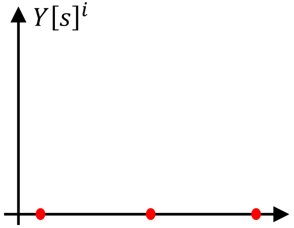
FRAMEWORK

Method

How is Root Cause Analysis done using Computer Aided Engineering (CAE) and Deep Learning?

Step 1: Intelligent Sampling of X

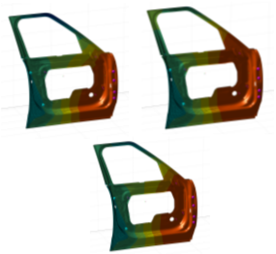
$X[s]^i = \textbf{Sampling}\ (\textbf{X})$



$\boldsymbol{X[s]^i}$ →

Step 2: Response Evaluation Y using CAE simulation

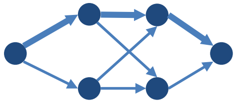
$Y[s]^i = \widehat{F}\ (X[s]^i)$



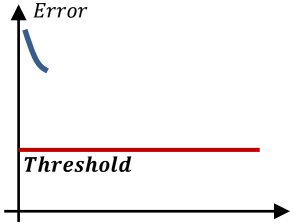
$\boldsymbol{X[s]^i, Y[s]^i}$ →

Step 3: Deep Learning Model Building

$\widehat{X}[s]^i = \widehat{F}_i^{\,-1}\ (Y[s]^i)$



$\boldsymbol{\widehat{X}[s]^i, X[s]^i}$ →



$\boldsymbol{Error(\widehat{X}[s]^i; X[s]^i) \le \epsilon}$

Model Performance Evaluation

YES →

Step 4: Trained Model Deployment

$\widehat{X} = \widehat{F}^{-1}\ (Y[s]^i)$

NO

$\widehat{X}[s]^i, \sigma(\widehat{Y}[s]^i)\ for\ i = 1,2,3 \ldots, N$

$\widehat{\boldsymbol{F}}, \widehat{\boldsymbol{X}}$ *are estimates for* $\boldsymbol{F}$ *and* $\boldsymbol{X}$ *while* $\boldsymbol{i}$ *reprsents the interation that goes from* $\boldsymbol{i = 1, 2, 3\ to\ N}$ *until the model error is below the required threshold* $\boldsymbol{\epsilon}$

$\boldsymbol{s}$ *represents the sample from the population*

$\boldsymbol{\sigma(\widehat{Y}[s]^i)}$ *represents the model uncertainity distribution for the prediction* $Y[s]^i$

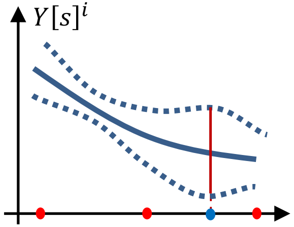
FRAMEWORK

Method

How is Root Cause Analysis done using Computer Aided Engineering (CAE) and Deep Learning?

Step I: Intelligent Sampling of X

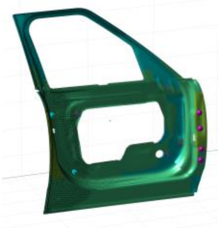
$X[s]^i = \textbf{Sampling}\ (\mathbf{X})$



$\boldsymbol{X[s]^i} \longrightarrow$

Step 2: Response Evaluation Y using CAE simulation

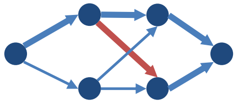
$Y[s]^i = \widehat{\boldsymbol{F}}\ (X[s]^i)$



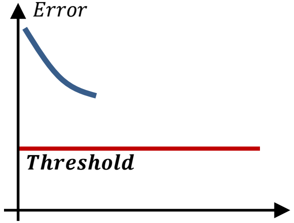
$\boldsymbol{X[s]^i, Y[s]^i} \longrightarrow$

Step 3: Deep Learning Model Building

$\widehat{X}[s]^i = \widehat{\boldsymbol{F}}_i^{-1}\ (Y[s]^i)$



$\widehat{\boldsymbol{X}}[s]^i, \boldsymbol{X}[s]^i \longrightarrow$



$\boldsymbol{Error}\big(\widehat{\boldsymbol{X}}[\boldsymbol{s}]^i; \boldsymbol{X}[\boldsymbol{s}]^i\big) \leq \boldsymbol{\epsilon}$

Model Performance Evaluation

NO

$\widehat{X}[s]^i, \sigma\big(\widehat{Y}[s]^i\big)\ for\ i = 1,2,3 \ldots, N$

$\widehat{\boldsymbol{F}}, \widehat{\boldsymbol{X}}$ are estimates for $\boldsymbol{F}$ and $\boldsymbol{X}$ while $\boldsymbol{i}$ reprsents the interation that goes from $\boldsymbol{i = 1, 2, 3\ to\ N}$ until the model error is below the required threshold $\boldsymbol{\epsilon}$

$\boldsymbol{s}$ represents the sample from the population

$\boldsymbol{\sigma}\big(\widehat{\boldsymbol{Y}}[\boldsymbol{s}]^i\big)$ represents the model uncertainity distribution for the prediction $Y[s]^i$

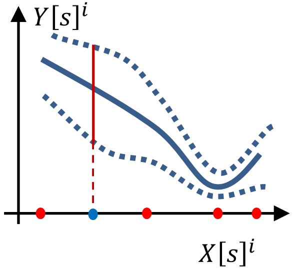
FRAMEWORK

Method

How is Root Cause Analysis done using Computer Aided Engineering (CAE) and Deep Learning?

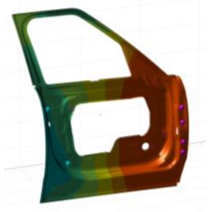
Step 1: Intelligent Sampling of X

$X[s]^i = \text{Sampling (X)}$



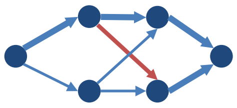
Step 2: Response Evaluation Y using CAE simulation

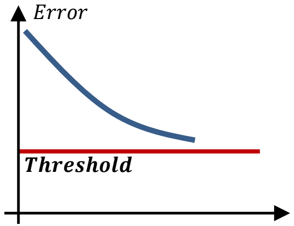
$Y[s]^i = \hat{F}\,(X[s]^i)$



Step 3: Deep Learning Model Building

$\hat{X}[s]^i = \hat{F}_i^{-1}\,(Y[s]^i)$





$Error(\hat{X}[s]^i; X[s]^i) \leq \epsilon$

Model Performance Evaluation

$X[s]^i$ →

$X[s]^i, Y[s]^i$ →

$\hat{X}[s]^i, X[s]^i$ →

no

$\hat{X}[s]^i, \sigma(\hat{Y}[s]^i)\ for\ i = 1,2,3 \ldots, N$

$\hat{F}, \hat{X}$ *are estimates for* $\boldsymbol{F}$ *and* $\boldsymbol{X}$ *while* $\boldsymbol{i}$ *reprsents the interation that goes from* $\boldsymbol{i = 1, 2, 3}$ ***to*** $\boldsymbol{N}$ *until the model error is below the required threshold* $\boldsymbol{\epsilon}$

$\boldsymbol{s}$ *represents the sample from the population*

$\boldsymbol{\sigma}(\hat{Y}[s]^i)$ *represents the model uncertainity distribution for the prediction* $Y[s]^i$

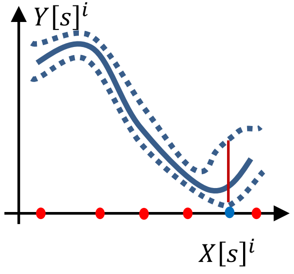
FRAMEWORK

Method

How is Root Cause Analysis done using Computer Aided Engineering (CAE) and Deep Learning?

Step 1: Intelligent Sampling of X

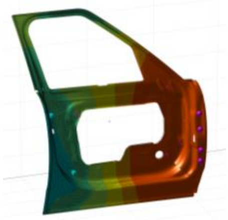
$X[s]^i = \textbf{Sampling}\ (\mathbf{X})$



$\boldsymbol{X[s]^i}$ →

Step 2: Response Evaluation Y using CAE simulation

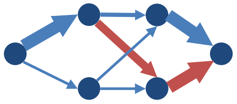
$Y[s]^i = \widehat{\boldsymbol{F}}\ (X[s]^i)$



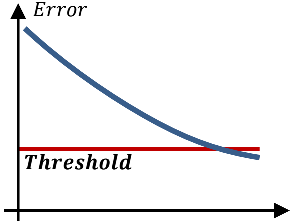
$\boldsymbol{X[s]^i, Y[s]^i}$ →

Step 3: Deep Learning Model Building

$\widehat{X}[s]^i = \widehat{\boldsymbol{F}}_{\boldsymbol{i}}^{\ -1}\ (Y[s]^i)$



$\boldsymbol{\widehat{X}[s]^i, X[s]^i}$ →



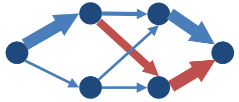
$\boldsymbol{Error}(\widehat{\boldsymbol{X}}[s]^i; \boldsymbol{X}[s]^i) \leq \boldsymbol{\epsilon}$

Model Performance Evaluation

YES →

Step 4: Trained Model Deployment

$\widehat{X} = \widehat{\boldsymbol{F}}^{-1}\ (Y[s]^i)$



NO

$\widehat{X}[s]^i, \sigma\left(\widehat{Y}[s]^i\right)\ for\ i = 1,2,3 \ldots, N$

$\widehat{\boldsymbol{F}}, \widehat{\boldsymbol{X}}$ *are estimates for* $\boldsymbol{F}$ *and* $\boldsymbol{X}$ *while* $\boldsymbol{i}$ *reprsents the interation that goes from* $\boldsymbol{i = 1, 2, 3\ to\ N}$ *until the model error is below the required threshold* $\boldsymbol{\epsilon}$

$\boldsymbol{s}$ *represents the sample from the population*

$\boldsymbol{\sigma}(\widehat{\boldsymbol{Y}}[\boldsymbol{s}]^{\boldsymbol{i}})$ *represents the model uncertainity distribution for the prediction* $Y[s]^i$

SOFTWARE – Overview

- Python Based
- TensorFlow Backend
- Platform Independent
- Object-Oriented
- GPU Enabled
- GitHub Project*
- Fully-Documented**
- Open-Source

In the pipeline:

- Distributed Computing
- Advance Multi-Stage Case Studies

GitHub Project *: https://github.com/sumitsinha/Deep_Learning_for_Manufacturing

*Documentation***: https://sumitsinha.github.io/Deep_Learning_for_Manufacturing/html/index.html

SOFTWARE – NOVELTY

Uncertainty Quantification

System Dimensionality

Architecture Selection

System Collinearity

Probabilistic Model using Bayesian Inference

Used to quantify the uncertainty of predictions using approximate Bayesian inference

Active Learning

Adaptive sampling strategies to handle the dimensionality of the system

3D CNN Architecture*

Optimized 3D Convolutional neural network model architectures

Gaussian Mixture based Multi-Mode Output Model

Used to handle the collinearities present within a system

The library includes implementation of novel research and development to solve the stated challenges

Sinha, S., Glorieux, E., Franciosa, P., & Ceglarek, D. (2019, June). 3D convolutional neural networks to estimate assembly process parameters using 3D point-clouds. In Multimodal Sensing: Technologies and Applications (Vol. 11059, p. 110590B). International Society for Optics and Photonics.

SOFTWARE – SOLUTIONS

Transfer Learning

Consists of routines and guidelines for transferring knowledge between different cases and quantifying the similarity

Pre-Trained Models

Consists of trained neural network models that can be directly leveraged for application using marginal transfer learning

Datasets

Supervised learning datasets for various assembly case studies

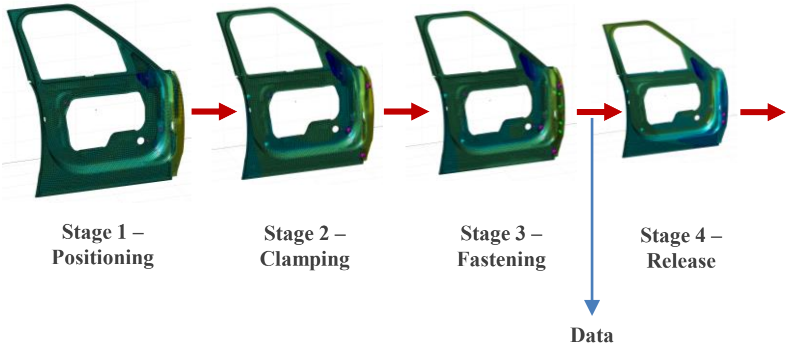
Data Convergence Study

Used to quantify the amount of data required for model training

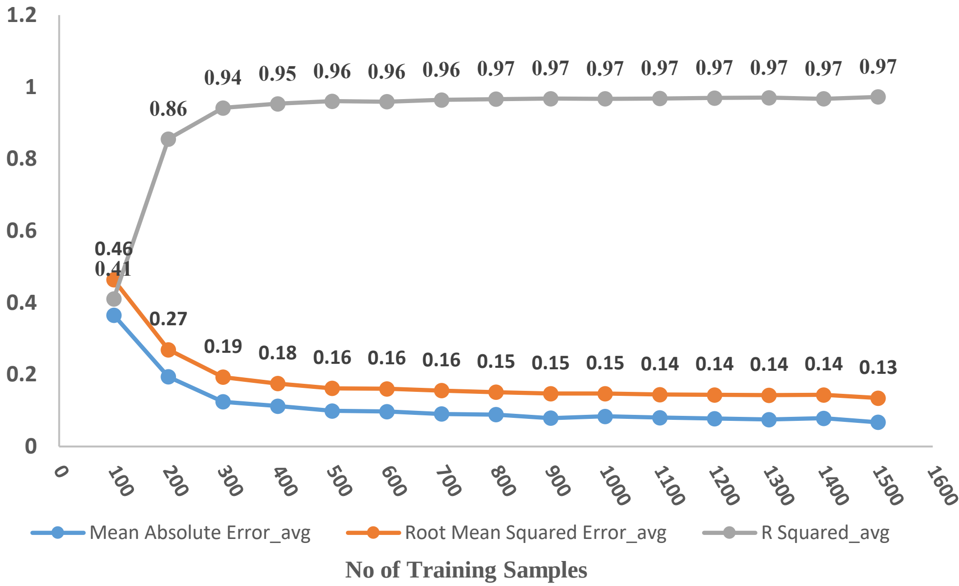
The library also includes datasets and key modules for various tasks not included in standard deep learning libraries, that help solve the model transferability challenges

SOFTWARE – CASE STUDY

Target: Intelligent Root Cause Analysis for two-part assembly



If data is collected at stage 3, the model converges with 1000 training samples with a root mean square error (RMSE) of 0.14 and R-squared value of 0.97



Testing is done up to double of the specification limit used for training

COMPETITIVE ADVANTAGE

Result

Application and integration of CAE Simulation with deep learning ensures early estimation and prediction of process parameter variations hence they can be prevented from manifesting into product defects

- Improvement in quality due to increased RCA capabilities
- Improvement in productivity due to reduction in scrap
- Reduction of machine downtime due to rapid automated RCA
- Reduction of manual expertise for RCA of defects

COMPETITIVE ADVANTAGE

Current Scenario

Only Product Data

Only having product data can only be used for monitoring and not RCA

- Limited RCA capabilities/only monitoring
- Requirement of manual expertise
- No/limited data for AI model training

1

Process data sensors for all parameters

Directly obtain process data for each parameter without the need for any learning

High RCA Capabilities

- High costs
- Difficult to setup within an online system

2

Product Data + CAE Simulation + Artificial Intelligence

Use AI model trained on real and simulated data to model relationship and suggest minimum additional process sensors in various sub-stages of the system

- Low costs
- High RCA Capabilities
- Strategic sensor placement

THANK-YOU

Does anyone have any questions?

CONTACT

sumit.sinha.1@warwick.ac.uk
+44 (0) 7570200029

IMC, WMG, University of Warwick
in linkedin.com/in/sumit-sinha-32891956