Extreme Mechanics Letters 36 (2020) 100659

Contents lists available at ScienceDirect

Extreme Mechanics Letters

journal homepage: www.elsevier.com/locate/eml

Mechanical MNIST: A benchmark dataset for mechanical metamodels

Emma Lejeune

Department of Mechanical Engineering, Boston University, Boston, MA 02215, United States of America

ARTICLE INFO

Article history:
Received 11 December 2019
Received in revised form 20 February 2020
Accepted 2 March 2020
Available online 9 March 2020

Keywords:
Finite element analysis
Machine learning
Metamodel
Dataset

ABSTRACT

Metamodels, or models of models, map defined model inputs to defined model outputs. Typically, metamodels are constructed by generating a dataset through sampling a direct model and training a machine learning algorithm to predict a limited number of model outputs from varying model inputs. When metamodels are constructed to be computationally cheap, they are an invaluable tool for applications ranging from topology optimization, to uncertainty quantification, to multi-scale simulation. By nature, a given metamodel will be tailored to a specific dataset. However, the most pragmatic metamodel type and structure will often be general to larger classes of problems. At present, the most pragmatic metamodel selection for dealing with mechanical data has not been thoroughly explored. Drawing inspiration from the benchmark datasets available to the computer vision research community, we introduce a benchmark data set (Mechanical MNIST) for constructing metamodels of heterogeneous material undergoing large deformation. We then show examples of how our benchmark dataset can be used, and establish baseline metamodel performance. Because our dataset is readily available, it will enable the direct quantitative comparison between different metamodeling approaches in a pragmatic manner. We anticipate that it will enable the broader community of researchers to develop improved metamodeling techniques for mechanical data that will surpass the baseline performance that we show here.

1. Introduction

Predictive mechanical models are foundational to engineering design and analysis [1]. Therefore, method development towards improving the predictive ability of mechanical simulation has and continues to be a highly active area of research [2]. High-fidelity computational models are used to make predictions that enable innovation in fields ranging from structural [3], to biomechanical engineering [4,5]. Because these models are often quite computationally expensive, researchers frequently construct metamodels as a part of their broader simulation framework [6]. Metamodels (also referred to as surrogate models) are computationally cheap models of the original model that map defined model inputs to defined model outputs, often referred to as quantities of interest (QoI) [7]. Though metamodels typically only predict a small portion of the total model output with some associated error, they typically take orders of magnitude less time to evaluate than the direct models that they are approximating [8]. These computationally cheap metamodels have enabled research in fields such as optimization [9,10], uncertainty quantification [11,12], and multi-scale simulation [13–15], where an intractable number of direct model evaluations would otherwise be required [16]. For approximating mechanical simulations, researchers have used metamodeling techniques such as standard Gaussian Process Regression [17], and neural networks [18]. Furthermore, there has been substantial recent research in developing physics informed machine learning methods that synthesize available physical knowledge and available data to make predictions [19,20].

Though these techniques have been reasonably successful, the most pragmatic metamodel type and structure for dealing with mechanical data is far from thoroughly explored. The motivation for developing improved strategies for constructing metamodels relevant to mechanical data is twofold. First, metamodels enable unprecedented exploration of the model parameter space [21,22] and enhanced metamodel performance will lead to improvements in the computational methods that rely on them [23]. Second, similar to synthetic datasets in computer vision [24], the synthetic datasets generated by mechanical models are a proxy for real world data. For many problems in mechanics, acquiring curated datasets of experimental data is much more resource intensive than generating curated datasets from simulation [25,26]. Therefore, working with synthetic datasets, and subsequently metamodels, is an initial step in method development and a motivation for the future strategic collection of experimental data.

Through emerging metamodeling techniques have enormous potential to enhance simulation, strategies for constructing metamodels remain ad hoc. Namely, metamodels are typically trained

E-mail address: elejeune@bu.edu.

https://doi.org/10.1016/j.eml.2020.100659
2352-4316/© 2020 Elsevier Ltd. All rights reserved.

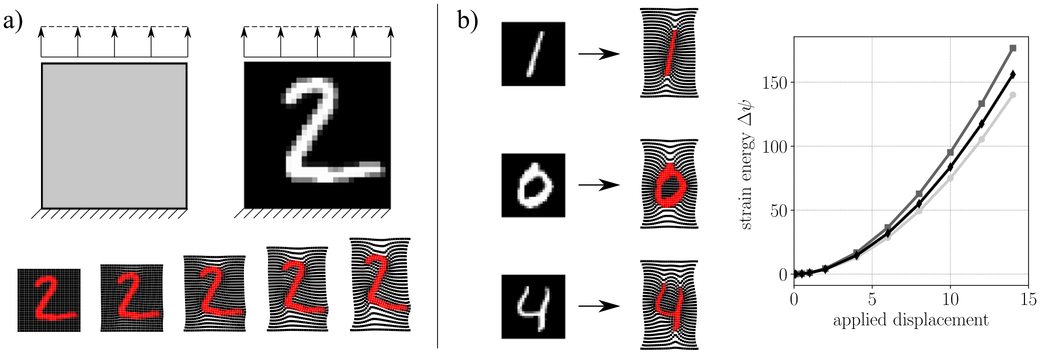

Fig. 1. Inspired directly by the MNIST dataset of handwritten digits used by the computer vision community, Mechanical MNIST is a dataset relevant to heterogeneous materials undergoing large deformation: (a) Mechanical MNIST uses the MNIST grayscale bitmap images to dictate material properties in a finite element simulation of a two-dimensional block being stretched; (b) From these simulations, multiple different forms of data (ex: full field displacement, total change in strain energy) are generated and contribute to the Mechanical MNIST dataset. Additional load cases are presented in Appendix A.1.

to capture meaning from privately held and/or difficult to interpret datasets generated with in-house software [27]. Though this approach is often necessary, it makes it difficult to compare between different metamodeling strategies and thus develop best practices for pragmatic metamodel selection specific to mechanical data. The objective of this paper is to introduce a benchmark data set that will enable improved metamodeling techniques for problems involving heterogeneous material undergoing large deformation. In addition to introducing this dataset, we also show examples of metamodels trained to this data that subsequently dictate a baseline for metamodel performance. We draw inspiration directly from the MNIST dataset used by the computer vision research community [28], and construct a mechanically relevant equivalent — Mechanical MNIST. Looking forward, we hope that this work will be one of many benchmark toy problems available to researchers interested in metamodeling of mechanical simulations.

The remainder of this paper is organized as follows. First, in Section 2.1, we describe our method for running the simulations to generate Mechanical MNIST. Then, in Section 2.2, we describe how this data can be used to train a metamodel. Briefly, in Section 2.3 we also discuss alternative approaches to metamodeling beyond the scope of this paper. In Section 3, we present the results of training different metamodels to Mechanical MNIST data. In Section 3.1, we predict final change in strain energy from material properties, and in Section 3.2 we predict material properties from final displacement. Critically, Mechanical MNIST is freely available online and access details are described in Supplementary materials.

2. Methods

In this Section, we describe how Mechanical MNIST is generated and the metamodeling strategies that we tested in this work. We note that the code used to generate Mechanical MNIST and the code used to create the metamodels from the Mechanical MNIST dataset are both freely available. Access information is available in Section 5.

2.1. Data generation

As illustrated in Fig. 2, Mechanical MNIST is generated by first converting the MNIST bitmap images to heterogeneous blocks of material. For simplicity, we use a compressible Neo-Hookean material model:

$$\psi = \frac{1}{2}\mu\left[\mathbf{F}:\mathbf{F} - 3 - 2\ln(\det\mathbf{F})\right] + \frac{1}{2}\lambda\left[\frac{1}{2}((\det\mathbf{F})^2 - 1) - \ln(\det\mathbf{F})\right] \tag{1}$$

where ψ is strain energy, $\mathbf{F}$ is the deformation gradient, and μ and λ are Lamé parameters equivalent to Young's modulus E and Poisson's ratio ν:

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \qquad \nu = \frac{\lambda}{2(\lambda + \mu)}\,. \tag{2}$$

To convert the MNIST bitmap images to material properties, we divide the material domain such that it corresponds with the grayscale bitmap and then specify E as

$$E = \frac{b}{255.0}(100.0 - 1.0) + 1.0 \tag{3}$$

where b is the corresponding value of the grayscale bitmap that can range from 0–255. Poisson's ratio is kept fixed at $\nu = 0.3$ throughout the domain. This strategy means that the Mechanical MNIST material domains contain a soft background material with "digits" that are two orders of magnitude stiffer.

Then, we run a finite element simulation where the bottom of the domain is fixed (Dirichlet boundary condition), the left and right edges of the domain are free, and the top of the domain is moved to a set of given fixed displacements. This is illustrated in Fig. 1a. In keeping with the size of the MNIST bitmap (28 × 28 pixels), the domain is a 28 × 28 unit square. We prescribe displacement at the top of the domain up to 50% of the initial domain size. The applied displacements $\mathbf{d}$ are:

$$\mathbf{d} = [\,0.0,\ 0.001,\ 0.01,\ 0.1,\ 0.5,\ 1.0,\ 2.0,\ 4.0,\ 6.0,\ 8.0,\ 10.0,\ 12.0,\ 14.0\,] \tag{4}$$

and data is generated at each displacement step. We run all finite element simulations using the FEniCS computing platform [29,30]. Mesh refinement studies were conducted, and we determined that a mesh with 39,200 quadratic triangular elements is sufficient to capture the converged solution while not needlessly using computational resources. This mesh corresponds to 50 quadratic triangular elements per bitmap pixel. In addition to the uniaxial extension load case discussed here, we have shear, equibiaxial extension, and confined compression load cases in the Mechanical MNSIT collection. Additional information on these load cases is presented in Appendix A.1.

From these simulations we generate data on the total change in strain energy $\Delta\psi$, total reaction force, and full field domain displacement. We store this data at each level of applied displacement. In summary, Mechanical MNIST contains the following data separated into test and training groups:

- original MNIST grayscale bitmaps stored in text files — each row of the file contains one image reshaped to a 1d array

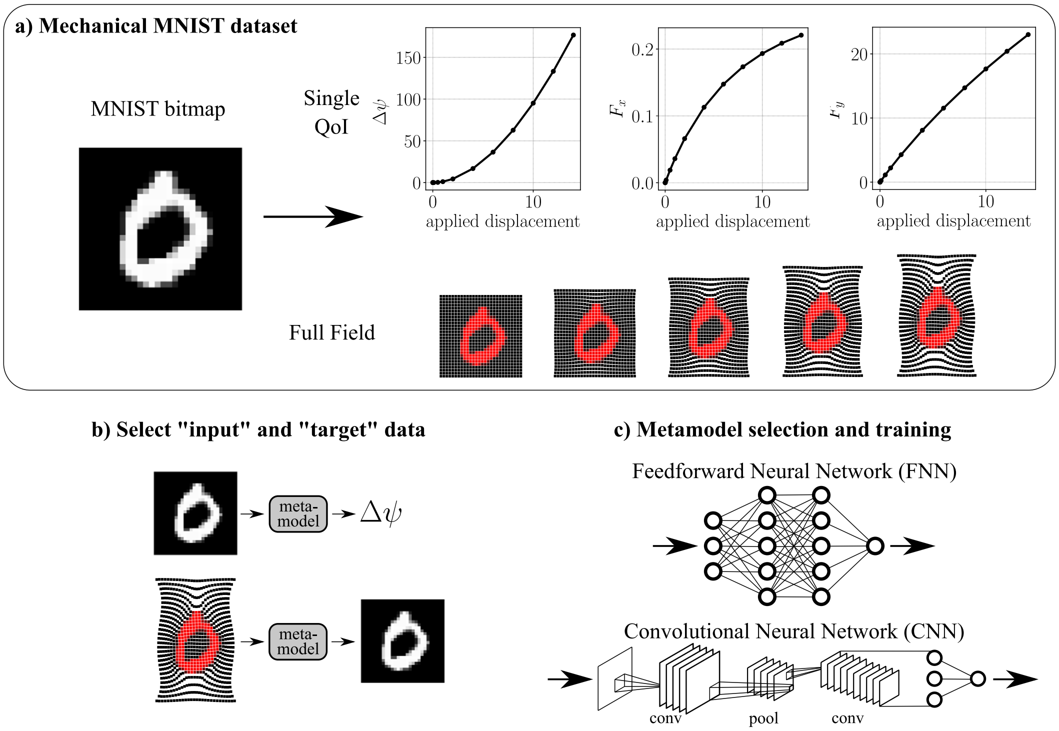

Fig. 2. (a) Illustration of the data that makes up the Mechanical MNIST dataset; (b) examples of different input and output data pairs that can be chosen from the Mechanical MNIST dataset; (c) Schematic illustration of a feedforward neural network (FNN) and a convolutional neural network (CNN), two metamodeling techniques explored in this paper.

- change in strain energy at every step of applied displacement — each row of the file contains a 1d array of $\Delta\psi$ at each step
- change in x reaction force at every step of applied displacement — each row of the file contains a 1d array of the total x reaction force computed on the top surface of the block
- change in y reaction force at every step of applied displacement — each row of the file contains a 1d array of the total y reaction force computed on the top surface of the block
- change in x position at the center of each initial "pixel" at every step of applied displacement — each row of the file contains x displacement values reshaped to a 1d array (there is one text file per applied displacement step)
- change in y position at the center of each initial "pixel" at every step of applied displacement — each row of the file contains y displacement values reshaped to a 1d array (there is one text file per applied displacement step)

Information on accessing the supporting code to load and analyze this data is provided in Section 5.

2.2. Baseline metamodels

With the Mechanical MNIST dataset, we consider two main formats of input data mapped to output data. In both cases, the input data will be full-field information i.e. information on a 28 × 28 grid (initial bitmap, x displacement map, y displacement map). The output data will then either be a single variable ($\Delta\psi$, reaction force) or full-field information (initial bitmap, x displacement map, y displacement map). It is also possible to predict multiple variables (ex: $\Delta\psi$ at every load step) or multiple pieces of full-field information (ex: both x and y displacement maps). We keep the same split between test and training data as the original MNIST dataset. In the remainder of this section we briefly introduce the metamodeling techniques that we use for the baseline results shown in Section 3.

2.2.1. Feedforward neural network

There are multiple examples in the literature of feedforward neural networks (FNNs) used to generate metamodels of physical simulation [31] and mechanical data [32]. The basic architecture of a feed-forward neural network is illustrated in Fig. 2c. Briefly, the FNN maps input feature vectors $\mathbf{x}$ to output values y. Fundamentally, a neural network is composed of "neurons," or basic units that take a weighted sum of inputs, and apply an activation function to the sum. In a FNN, information flows from a flattened input vector through hidden layers of connected neurons to a final output layer [33]. The FNN has several parameters, $\boldsymbol{\theta}$, and process of "training" the FNN involves using a software library to determine what $\boldsymbol{\theta}$ should be. In this work, we construct FNNs with both the PyTorch library [34] and the scikit-learn library [35].

2.2.2. Convolutional neural network

Recently, convolution neural networks (CNNs) have gained substantial popularity for generating metamodels of physical simulation [36]. The basic architecture of a convolutional neural network is illustrated in Fig. 2c. Similar to FNNs, CNNs start with an input layer, contain hidden layers, and finish with an output layer. However, rather than the flat input that goes into a FNN, the input to a CNN is an image (in this case a 28 × 28 image), and the hidden layers typically include both convolution layers and pooling layers. Convolution layers map a single array (the image) to multiple arrays by filtering the image with a set of independent kernels. Pooling layers progressively reduce the size of the arrays by applying a filter that reduces dimensionality [37]. Both convolutional and pooling layers are illustrate in Fig. 2c. This architecture, where the structure of the image is preserved, means that CNNs are potentially better suited to capture spatial relationships than FNNs [38]. In this work, we construct a CNN with the PyTorch library [34].

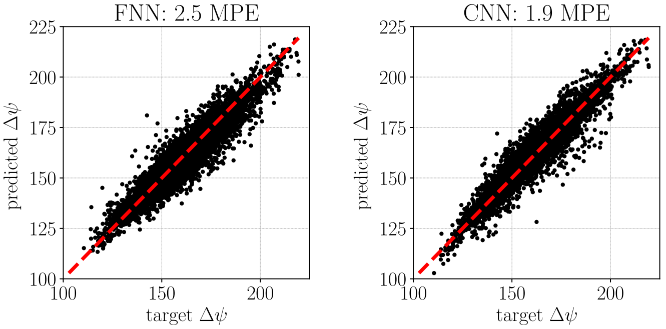

Fig. 3. Illustration of test error when predicting total change in strain energy $\Delta\psi$ at the final level of applied displacement (50% of initial length) from initial pixel bitmaps. Left: Feedforward neural network (FNN) with 2.5% mean error; Right: Convolutional neural network (CNN) with 1.9% mean error. Similar plots for the alternative load cases are presented in Appendix A.1.

2.3. Note on alternative approaches

Although implementing a physics informed machine learning method is beyond the scope of this paper, we note that this area of research is highly relevant to the development of metamodels for mechanical data [39–41]. In particular, recent work focused on constraining neural networks based on physical laws shows substantial promise for making predictions particularly in systems where there is limited available data [12,42]. Furthermore, model order reduction, a technique for decreasing computational complexity, is an alternative approach beyond the scope of this paper [43–45]. One major motivation for sharing the full Mechanical MNIST dataset and the finite element scripts used to generate the data is that it is a strategy for pragmatic comparison between entirely data based machine learning approaches, physics informed machine learning approaches, and even potentially model order reduction strategies. The efficacy of machine learning techniques is judged by their predictive power. Therefore, comparisons made with benchmark datasets are a powerful way to demonstrate the utility of novel methods.

3. Results and discussion

In this Section, we present baseline results of predictions for the Mechanical MNIST data. We note that while the performance of our metamodels is good, it is likely far from optimal. The motivation for sharing this dataset is that we hope Mechanical MNIST will serve as a benchmark for alternate metamodeling strategies to capture heterogeneous material undergoing large deformation. From an entirely pragmatic perspective, alternative metamodeling strategies are noteworthy if and when they outperform the baseline results shown here.

3.1. Predicting a single QoI

The original MNIST dataset takes 28 × 28 pixel images of handwritten digits and classifies each image as a digit from 0–9. Our analogy to the original MNIST dataset is a regression problem, where the input is the same 28 × 28 pixel images and the output is a single value $\Delta\psi$ which reflects the total change in strain energy when the block is stretched to 50% of its original length (uniaxial extension load case). Inspired by models used to classify the original MNIST dataset, we reformulate a FNN that achieved approximately 97% accuracy on the original MNIST dataset and a CNN that achieved approximately 99% accuracy on the original MNIST dataset to address our regression problem. Then, we evaluate the performance of the FNN and CNN on our Mechanical MNIST dataset.

The performance of the FNN and the CNN are shown in Fig. 3. Here we report the mean percent error (MPE). The test error for the FNN was 2.5% and the test error for the CNN was 1.9%. The training error for the FNN was 2.4% and the training error for the CNN was 1.9%. The plots in Fig. 3 show the predicted value of $\Delta\psi$ with respect to the target value of $\Delta\psi$. These results, in particular the results of the CNN, represent the baseline performance of a metamodel fit to the Mechanical MNIST data. We also note that we tried similar NN architectures to the final one chosen for both the FNN and CNN but did not notice substantial improvement in test error. We anticipate that an alternative neural network architecture, or potentially an ensemble method will lead to improved performance in the future.

3.2. Full field prediction

Here we show an example of a full field prediction. We predict the initial material properties from the final displacement field (a common goal of inverse problems). The results of this comparison are shown in Fig. 4. Across the entire test set, the mean absolute test error for the initial material property prediction was 13 bitmap intensity units. For context, the standard deviation of bitmap pixel values is 79 with range 0–255. For the metamodel, we use a simple FNN constructed with the MLPRegressor function in Python scikit-learn [35]. Similar to the results presented in Section 3.1, we view these results as an example of baseline performance. Further exploration of alternative neural network architectures and/or alternative metamodeling methods will likely yield lower test error.

The Mechanical MNIST dataset presents a toy problem that allows for the exploration of hypothetical scenarios where different types of data are available. For example, the prediction of initial material properties from displacement assumes that the initial material properties are unknown. The metamodeling techniques explored in this Section treat the finite element model used to generate the Mechanical MNIST data as a black box. We note briefly that alternative techniques such as inverse modeling would likely be able to outperform these black box approaches if the problem could be formulated with substantial domain knowledge [46]. We include additional discussion on the use of the Mechanical MNIST dataset in Appendix A.2.

4. Conclusion

Inspired by benchmark datasets available to the computer vision research community [28,52], we introduce a dataset, Mechanical MNIST, relevant to metamodeling of heterogeneous materials undergoing large deformation. We first describe how this dataset was generated, and then describe our methods for creating baseline metamodels with the dataset. Notably, we are able to predict change in strain energy on the test data from the initial bitmap with a mean percent error of 1.9%. Because our dataset is readily available, we anticipate that future metamodeling endeavors potentially put forward by other research groups will exceed this performance. Though Mechanical MNIST represents a toy problem, we anticipate that metamodeling techniques (not the trained models themselves) that have high predictive power on Mechanical MNIST will be well suited for tackling related problems in creating metamodels of heterogeneous materials undergoing large deformation.

There are many interesting and highly related research questions beyond the scope of this initial work, many of which are directly inspired by ongoing research in computer vision. For example, MNIST classification can be improved via multiple data

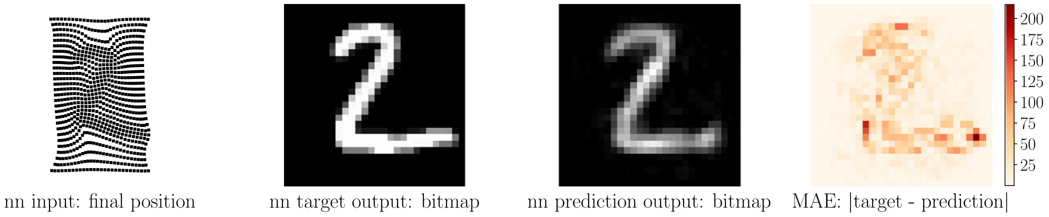

Fig. 4. Illustration of the performance of a FNN on a single example from the test set of predicting initial material properties from final displacement. In this example, Mean Absolute Error (MAE) is 16 bitmap intensity units. On average across the entire test dataset, MAE is 13 bitmap intensity units. This can be compared to the range 0–255 and the standard deviation of 79 for bitmap intensity units.

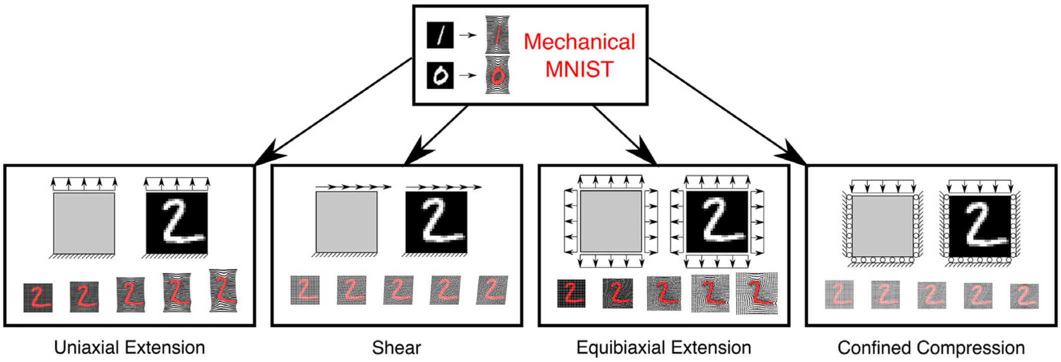

Fig. 5. The four loading cases present in the Mechanical MNIST collection [47] are: Uniaxial Extension [48], Shear [49], Equibiaxial Extension [50], and Confined Compression [51].

augmentation strategies [53–55]. For the original MNIST, an augmented bitmap typically maps to the same output class. For Mechanical MNIST, an augmented bitmap will map to different quantitative outputs. Broadly speaking, data augmentation and sensitivity to noise "attacks" specific to regression problems is an ongoing area of research [56]. Likewise, the identification of algorithms that can work with as little data as possible is particularly relevant to mechanical data [57]. As pointed out in Section 2.3, these methods will potentially incorporate domain knowledge.

In addition to addressing these compelling problems, future work will involve generating and curating additional datasets of both computational and experimental data to address mechanically relevant problems beyond the limited scope of Mechanical MNIST in its present form. Currently, the Mechanical MNIST collection contains four load cases with a Neo-Hookean constitutive model and spatially heterogeneous material. Future contributions to the collection will tackle alternative constitutive laws, alternative constitutive parameter ranges, material anisotropy, and simulations of varying fidelity and dimension. We note that the code used to generate Mechanical MNIST is freely available (see Supplementary materials) and the Mechanical MNIST collection is shared with a CC BY-SA 4.0 license which allows other researchers to freely share and adapt the dataset. As more curated datasets become available it will be interesting to see, from an entirely pragmatic perspective, if specific neural network architectures or alternate metamodeling techniques tend to perform best on mechanical data.

5. Supplementary materials

Mechanical MNIST is available through the OpenBU Institutional Repository (Collection: https://open.bu.edu/handle/2144/39371, Uniaxial Extension: https://open.bu.edu/handle/2144/38693, Shear: https://open.bu.edu/handle/2144/39429, Equibiaxial Extension: https://open.bu.edu/handle/2144/39428, and Confined Compression: https://open.bu.edu/handle/2144/39427). The code used to generate these datasets and the baseline metamodel examples is available on the Mechanical MNIST GitHub Repository (https://github.com/elejeune11/Mechanical-MNIST).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the staff of the Boston University Research Computing Services and the OpenBU Institutional Repository (in particular Eleni Castro) for their invaluable assistance with generating and disseminating Mechanical MNIST. This work was made possible though start up funds from the Boston University Department of Mechanical Engineering.

Appendix

A.1. Mechanical MNIST: additional load cases

In the main body of the text, we introduced the Mechanical MNIST dataset with an emphasis on the uniaxial extension load case. In addition to uniaxial extension, the Mechanical MNIST dataset contains (at the time of this publication) three additional load cases, illustrated in Fig. 5. We anticipate that the Mechanical MNIST collection will continue to grow to cover additional areas of interest such as alternative constitutive laws and simulations of varying fidelity.

Following the procedure described in Section 2.2.2, we show baseline metamodel performance on each dataset. Specifically, we train multiple CNNs with the same architecture separately on each Mechanical MNIST load case. The baseline performance of the CNN on each dataset is summarized in Fig. 6. Briefly,

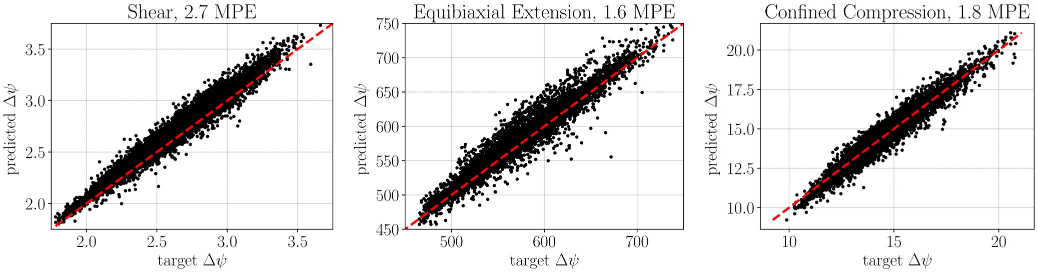

Fig. 6. Left: baseline CNN performance on Mechanical MNIST Shear; Center: baseline CNN performance on Mechanical MNIST Equibiaxial Extension; Right: baseline CNN performance on Mechanical MNIST Confined Compression. The mean percent error (MPE) is stated within the title of each figure.

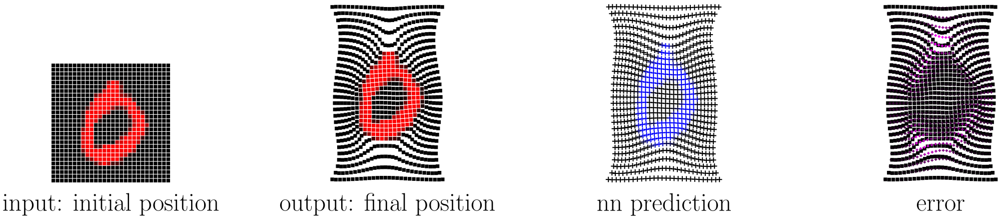

Fig. 7. Illustration of the performance of a FNN on a single example from the test set for predicting final displacement from initial displacement. Mean Absolute Error on the test data is 0.44 length units, which we can compare to initial block side length of 28 length units and the standard deviation of pixel-center displacement which is 4.3 length units.

baseline performance of the CNN on the Shear dataset is 2.60% training error, and 2.69% test error. Baseline performance of the CNN on the Equibiaxial Extension dataset is 1.59% training error, and 1.61% test error. Baseline performance of the CNN on the Confined Compression dataset is 1.76% training error, and 1.83% test error. Notably, the performance of the CNN is fairly consistent across datasets.

A.2. Mechanical MNIST: beyond metamodels and inverse problems

In addition to the examples highlighted in the main body of the text, the Mechanical MNIST dataset can be set up to test multiple different types of predictive model. An example of this is shown in Fig. 7. In this example, the full field displacement at the final load step is predicted from an initial perturbation of the same loading type. Again, we use a simple FNN constructed with the MLPRegressor function in Python scikit-learn [35]. Though Mechanical MNIST lacks much of the complexity of real world data, it is a starting point for developing machine learning based tools to predict deformation under large loading from small perturbations that could ultimately be relevant to applications such as non destructive testing [58,59].

Beyond this example, another option for reformulating the input and output of the metamodel would be to use the full Mechanical MNIST dataset with multiple loadings described in Appendix A.1 to set up a classification problem where full field displacement is used to predict the loading class (uniaxial extension, shear, equibiaxial extension, or confined compression). It would be interesting to investigate how (and if) data from one load case could be leveraged to make predictions about other load cases. Researchers can also add different amounts and types of noise to the Mechanical MNIST dataset to better capture real world data acquisition. Looking forward, we hope that beyond metamodels, Mechanical MNIST will serve as a synthetic dataset that is a placeholder for real world data during method development.

References

[1] T.J. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.

[2] M. Alber, A.B. Tepole, W.R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W.W. Lytton, P. Perdikaris, L. Petzold, et al., Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences, npj Digit. Med. 2 (1) (2019) 1–11.

[3] T.A. Helgedagsrud, Y. Bazilevs, K.M. Mathisen, O.A. Øiseth, Computational and experimental investigation of free vibration and flutter of bridge decks, Comput. Mech. 63 (1) (2019) 121–136.

[4] E. Lejeune, B. Dortdivanlioglu, E. Kuhl, C. Linder, Understanding the mechanical link between oriented cell division and cerebellar morphogenesis, Soft Matter 10 (2019) 2204–2215.

[5] M.K. Rausch, A.M. Zöllner, M. Genet, B. Baillargeon, W. Bothe, E. Kuhl, A virtual sizing tool for mitral valve annuloplasty, Int. J. Numer. Methods Biomed. Eng. 33 (2) (2017) e02788.

[6] A.I. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (1–3) (2009) 50–79.

[7] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P.K. Tucker, Surrogate-based analysis and optimization, Prog. Aerosp. Sci. 41 (1) (2005) 1–28.

[8] P.Z. Hanakata, E.D. Cubuk, D.K. Campbell, H.S. Park, Accelerated search and design of stretchable graphene kirigami using machine learning, Phys. Rev. Lett. 121 (25) (2018) 255304.

[9] H.S. Kim, M. Koc, J. Ni, A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network, Int. J. Mach. Tools Manuf. 47 (2) (2007) 211–222.

[10] K.K. Vu, C. d'Ambrosio, Y. Hamadi, L. Liberti, Surrogate-based methods for black-box optimization, Int. Trans. Oper. Res. 24 (3) (2017) 393–424.

[11] T. Lee, S.Y. Turin, A.K. Gosain, I. Bilionis, A.B. Tepole, Propagation of material behavior uncertainty in a nonlinear finite element model of reconstructive surgery, Biomech. Model. Mechanobiol. 17 (6) (2018) 1857–1873.

[12] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 394 (2019) 136–152.

[13] M. Peirlinck, F.S. Costabal, K. Sack, J. Choy, G. Kassab, J. Guccione, M. De Beule, P. Segers, E. Kuhl, Using machine learning to characterize heart failure across the scales, Biomech. Model. Mechanobiol. 18 (6) (2019) 1987–2001.

[14] F. Sahli Costabal, J. Yao, E. Kuhl, Predicting drug-induced arrhythmias by multiscale modeling, Int. J. Numer. Methods Biomed. Eng. 34 (5) (2018) e2964.

[15] K. Wang, W. Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, Comput. Methods Appl. Mech. Engrg. 334 (2018) 337–380.

[16] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Rev. 60 (3) (2018) 550–591.

[17] T. Lee, A.K. Gosain, I. Bilionis, A.B. Tepole, Predicting the effect of aging and defect size on the stress profiles of skin from advancement, rotation and transposition flap surgeries, J. Mech. Phys. Solids 125 (2019) 572–590.

[18] K. Wang, W. Sun, Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning, Comput. Methods Appl. Mech. Engrg. 346 (2019) 216–241.

[19] M. Raissi, G.E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141.

[20] D. Zhang, L. Lu, L. Guo, G.E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, J. Comput. Phys. 397 (2019) 108850.

[21] L. Bonfiglio, P. Perdikaris, S. Brizzolara, G. Karniadakis, Multi-fidelity optimization of super-cavitating hydrofoils, Comput. Methods Appl. Mech. Engrg. 332 (2018) 63–85.

[22] G.H. Teichert, K. Garikipati, Machine learning materials physics: Surrogate optimization and multi-fidelity algorithms predict precipitate morphology in an alternative to phase field dynamics, Comput. Methods Appl. Mech. Engrg. 344 (2019) 666–693.

[23] K. Wang, W. Sun, Q. Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation, Comput. Mech. (2019) 1–33.

[24] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, Flownet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2758–2766.

[25] R. Gupta, S. Salager, K. Wang, W. Sun, Open-source support toward validating and falsifying discrete mechanics models using synthetic granular materials—Part I: Experimental tests with particles manufactured by a 3D printer, Acta Geotech. 14 (4) (2019) 923–937.

[26] R. Gupta, S. Salager, K. Wang, W. Sun, Open-source support toward validating and falsifying discrete mechanics models using synthetic granular materials—Part I: Experimental tests with particles manufactured by a 3D printer, Acta Geotech. 14 (4) (2019) 923–937.

[27] E. Lejeune, C. Linder, Interpreting stochastic agent-based models of cell death, Comput. Methods Appl. Mech. Engrg. (2019) 112700.

[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[29] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, G.N. Wells, The FEniCS project version 1.5, Arch. Numer. Softw. 3 (100) (2015).

[30] A. Logg, K.-A. Mardal, G. Wells, Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, Vol. 84, Springer Science & Business Media, 2012.

[31] G. Teichert, A. Natarajan, A. Van der Ven, K. Garikipati, Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions, Comput. Methods Appl. Mech. Engrg. 353 (2019) 201–216.

[32] S. Jung, J. Ghaboussi, Neural network constitutive model for rate-dependent materials, Comput. Struct. 84 (15–16) (2006) 955–963.

[33] K. Gurney, An Introduction to Neural Networks, CRC press, 2014.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, 2017.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[36] M. Schwarzer, B. Rogan, Y. Ruan, Z. Song, D.Y. Lee, A.G. Percus, V.T. Chau, B.A. Moore, E. Rougier, H.S. Viswanathan, et al., Learning to fail: Predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks, Comput. Mater. Sci. 162 (2019) 322–332.

[37] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[39] G. Pang, L. Yang, G.E. Karniadakis, Neural-net-induced Gaussian process regression for function approximation and PDE solution, J. Comput. Phys. 384 (2019) 270–288.

[40] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, J. Comput. Phys. 348 (2017) 683–693.

[41] A.M. Tartakovsky, C.O. Marrero, D. Tartakovsky, D. Barajas-Solano, Learning parameters and constitutive relationships with physics informed deep neural networks, 2018, arXiv preprint arXiv:1808.03398.

[42] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.

[43] D. Amsallem, M. Zahr, Y. Choi, C. Farhat, Design optimization using hyper-reduced-order models, Struct. Multidiscip. Optim. 51 (4) (2015) 919–940.

[44] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction: Formulations for physics-based machine learning, Comput. & Fluids 179 (2019) 704–717.

[45] M.J. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1111–1135.

[46] N.H. Gokhale, P.E. Barbone, A.A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: the compressible case, Inverse Problems 24 (4) (2008) 045010.

[47] E. Lejeune, Mechanical MNIST, 2020, URL https://open.bu.edu/handle/2144/39371.

[48] E. Lejeune, Mechanical MNIST – Uniaxial extension, 2019, URL https://open.bu.edu/handle/2144/38693.

[49] E. Lejeune, Mechanical MNIST – Shear, 2020, URL https://open.bu.edu/handle/2144/39429.

[50] E. Lejeune, Mechanical MNIST – Equibiaxial extension, 2020, URL https://open.bu.edu/handle/2144/39428.

[51] E. Lejeune, Mechanical MNIST – Confined compression, 2020, URL https://open.bu.edu/handle/2144/39427.

[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[53] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, 2017, arXiv preprint arXiv:1712.04621.

[54] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.

[55] S.C. Wong, A. Gatt, V. Stamatescu, M.D. McDonnell, Understanding data augmentation for classification: when to warp? in: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2016, pp. 1–6.

[56] E. Balda, A. Behboodi, R. Mathar, Perturbation analysis of learning algorithms: generation of adversarial examples from classification to regression, IEEE Trans. Signal Process. (2019).

[57] C. White, D. Ushizima, C. Farhat, Neural networks predict fluid dynamics solutions from tiny datasets, 2019, arXiv preprint arXiv:1902.00091.

[58] D.E. Martin, A.E. Severns, J.M. Kabo, Determination of mechanical stiffness of bone by pQCT measurements: correlation with non-destructive mechanical four-point bending test data, J. Biomech. 37 (8) (2004) 1289–1293.

[59] R. Shah, M.C. Pierce, F.H. Silver, A method for nondestructive mechanical testing of tissues and implants, J. Biomed. Mater. Res. A 105 (1) (2017) 15–22.