The algorithm for generating the training set for the problem of elastoplastic deformation of the metal rod

Mykhailo Seleznov
Moscow Institute of Physics and Technology (National Research University), 9, Institutsky Per., Dolgoprudny, Moscow Region 141701, Russian Federation
mihailselezniov@yandex.ru

Abstract. The paper proposes an algorithm for forming a small training set, which will provide a reasonable quality of a surrogate ML-model for the problem of elastoplastic deformation of a metal rod under the action of a longitudinal load pulse. This dynamic physical problem is computationally simple and convenient for testing various approaches, but at the same time it is physically quite complex, because it contains a significant range of effects. So, the methods tested on this problem can be further applied to other areas. This work demonstrates the possibility of a surrogate ML-model to provide a reasonable prediction quality for a dynamic physical problem with a small training set size.

1. Introduction
Designing engineering structures and protective shells often requires solving the optimization problems for their parameters. For example, the problem may be minimizing the mass of a structure at a given static bearing capacity, or minimizing the mass of the protective shell while maintaining the ability to withstand a given range of shock loads.

Solving optimization problems typically uses the methods of mathematical modeling. The algorithm for selecting the optimal set of parameters can be different and depends on the nature of the problem. The parameters to optimize also vary depending on the nature of the problem. For example, optimizing the protective shell against shock loads involves the variation of elastic and strength properties of the material, thickness and geometry of the structure. The consequences of impacts by particles of different mass and shape, acting at different speeds and at different angles, should be calculated for each set of parameters. Although these calculations are possible, they are extremely resource-intensive. The computation time for one dynamic three-dimensional problem statement can take many hours or even days [1].

The solution of optimization problems can be sped up using surrogate models built using machine learning methods [2]. In this approach, a traditional numerical solver is used to generate the data on which the ML model is trained. The resulting ML-model gives answers in a fraction of a second in comparison with the hours for numerical solver. After that the ML-model is used to solve the multiparameter optimization problem. “Instant” ML prediction is used to obtain the result for the specific set of parameters. It should be noted that the answer of the ML-model may be incorrect in some cases, which is due to the very nature of machine learning. If the ML model gives a reasonable
answer in most cases, its application allows one to estimate promising ranges of parameters faster than using the numerical solver. These ranges of parameters can be checked and refined with the numerical solver, if required.

This approach raises a rather important question of the size of the training set. What is the minimum training set size that is sufficient to build a surrogate model of reasonable quality to describe a dynamic physical problem? It is desirable for real life problems to use smaller sets, since obtaining each sample for training still requires the numerical calculation of the problem. Each numerical calculation can take hours of computational time, so reducing the number of train samples is important.

This work proposes the algorithm to prepare a small train set and demonstrates the results of applying this algorithm for building surrogate ML model for dynamic elastoplastic deformation of a steel rod.

2.  Modeling techniques

The physical problem, considered in this work, is longitudinal pulse loading of the metal rod. The rod may experience either pure elastic or irreversible elastoplastic deformations, depending on the parameters of the load [3, 4].

The formulation of the problem is described using four parameters: Young’s modulus, material density, yield strength (elastic limit), impact velocity. The cross-sectional area and the length of the rod are taken constant for all sets. A third-party solver is used to perform calculations of the direct problem. The mathematical model returns a binary answer - whether there are any irreversible plastic deformations in the rod or not.

The properties of the rod and the load together with the results of direct calculation are used as the input to train the surrogate machine learning model. The surrogate model should attribute a set of four input parameters to one of two possible outcomes of the simulation.

The following three-layer neural networks are used as ML-models:

- Multilayer Perceptron using 3 hidden layers of 50 neurons each with ReLu activation function [5],
- Multilayer Perceptron using 3 hidden layers of 70 neurons each with ReLu activation function,
- Multilayer Perceptron using 3 hidden layers of 90 neurons each with hyperbolic tangent activation function.

The Scikit-Learn library is used for the implementation of the models.

These ML models are quite simple, but they are expected to provide reasonable demonstration of possible neural networks results. This work concentrates on the algorithm of the small training set formation, so the building of the optimal neural network is beyond the scope of this work.

2.1.  Datasets

Two datasets are used in this work:

- Set # 1 - the large dataset to assess the quality of the model;
- Set # 2 - the small training set for the ML-model.

The set #1 consists of 100 values for each parameter. In total, 100 million problem statements are calculated using the direct numerical solver.

Each statement of the direct problem has 4 parameters. The parameter values vary in the following ranges:

- Young’s modulus: 0.5e11 - 4e11 Pa,
- density: 1500 - 9000 kg / m³,
- elastic limit: 0.5e8 - 4e8 Pa,
- impact velocity: 1 - 10 m/s.
The values for each parameter are evenly distributed over the specified range. For example, the values for density are 1500, 1575.7, 1651.5, 1727.3,…, 9000.

2.2. The algorithm for forming the small training set #2 consists of the following stages

1) Take any 2 samples with different simulation outcomes: the rod in the first sample received irreversible plastic deformations, the rod in the second sample was deformed purely elastically. For the convenience of the algorithm, one can take samples that are as different as possible. For example, one can take the sample with the maximum possible values of the parameters of the rod as well as its opposite - the sample with minimum values of these parameters.

2) Train simple Multi-Layer Perceptron on the available number of samples. This work uses at this stage default neural network parameters from the sklearn package.

3) Submit the large test set of hypothetical samples to the network input. The answers for these hypothetical samples are not known at this moment. This work uses 96 millions of hypothetical samples. These samples use 99 values for each of the 4 problem statement parameters. The values of the parameters are between their values in the set #1. Thus, the values of the parameters in the resulting small training set and in the test set will not coincide. These 99 values are evenly spaced from the lower to the upper limit of each value, so we predict the outcome of the simulations on a grid of possible combinations. It’s worth emphasizing that calculations by the numerical solver for these samples are not performed, there are only prediction results.

4) Calculate the number of the results with pure elastic deformations between all predicted samples, divide it by 100 million and subtract from one. For example, 1 - 20 million / 100 million = 0.8. This coefficient is referred to as “optimal confidence level” in the further text. Numerical experiments showed that extreme values of this coefficient are undesirable, so in this work the following limiting logic was applied:

   If the coefficient is greater than 0.9, it is fixed at 0.9.
   If the coefficient is less than 0.1, it is fixed at 0.1.
   If the training set from step 1 contains less than 10 samples, the coefficient is set to 0.5.

5) Each sample from the grid of hypothetical values has a probability of getting one or another outcome according to the ML model. These probabilities can be obtained using the predict_proba method on the trained Multi-layer Perceptron model from step 2. The samples are selected for further processing if their probability is in the range of +/- 0.05 from the optimal confidence level. For example, with the optimal confidence level of 0.8, the probability range of 0.75 - 0.85 is used.

6) The Euclidean distance is calculated from each sample selected at step 5 to each sample from step 1 with pure elastic deformations. The sample with the maximum distance is selected. If there are several such samples, the Euclidean distance is calculated to the samples of another class from step 1, and the sample with the maximum distance is selected. This logic selects the sample with parameters that are most unlike the existing samples in the training set.

7) The parameters of the sample from step 6 are used to obtain the real result using the numerical solver.

8) The sample is included into the train set from step 1. All operations are repeated until the train set contains 100 samples.

Thus, a small set of 100 samples is formed, which is supposed to be sufficiently representative for training the surrogate ML-model. The resource-intensive numerical solver was used only for these 100 samples.

3. Results

Let us estimate the prediction quality if we use the train set #2 obtained using the algorithm described above.
The figure 1 shows the quality of trained neural networks when training using the set #2. The size of the training set on the figure varies from 20 to 100 samples. The large dataset #1 was used to assess the quality.

The figure 2 shows for comparison the results of the ML model that was trained using random training samples (from 100 to 10,000,000 samples with the parameters selected at random from the same ranges of parameters, that were used for the set #1).

The F-score for the class of pure elastically deformed rods was chosen as the quality indicator, since it severely punishes errors of types 1 and 2. Other metrics either follow the trend observed in the F-score, or are more optimistic.

![Figure 1. F-score using train set #2 with different number of samples and test set #1.](image-url)
4. Discussion

It can be seen that the first neural network (Multilayer Perceptron, 3 hidden layers of 50 neurons, ReLu activation function) showed a quality of 0.854 with 100 samples in the formed small training set. A similar quality of 0.853 was achieved when training using 10,000 random samples, and the quality did not increase further when adding new samples to the training set. Similar results were obtained with the other two neural networks. Even with 10 million random samples in the training set, the prediction quality of neural networks is only slightly higher than the result obtained with 100 samples from the set #2.

This work demonstrates that the surrogate ML-models can give a reasonable prediction quality for dynamic physical problems even with train sets of small sizes. The proposed algorithm for forming a small training set does not use any information about the features of the physical problem under consideration, therefore it can be quite simply generalized to other areas.

References

[1] Beklemysheva, K.A., Petrov, I.B.: Damage Modeling in Hybrid Composites Subject to Low-Speed Impact. Mathematical Models and Computer Simulations, vol. 11, pp. 469-478 (2019).
[2] Kim, S.H., Boukouvala, F.: Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. Optimization Letters, vol. 14, pp. 989-1010 (2020).
[3] Ivlev D.D., Theory of ideal plasticity. - M. : Nauka, 1966. -- 232 p.
[4] Rabotnov Yu.N., Mechanics of a Deformable Solid. - M. : Nauka, 1979. -- 744 p.
[5] Rumelhart D.E., Hinton G.E., Williams R.J.: Learning internal representations by error propagation. In: Parallel distributed processing: Explorations in the microstructure of cognition, MIT Press, pp. 318-362 (1986).