Prediction of Deformed Shape in Incremental Sheet Forming Processing Feedforward Neural Network

Hai Son Le, Quoc Tuan Pham, Anh Tuan Nguyen, Hoang Son Tran, and Xuan Van Tran

Abstract Numerical simulations of the incremental sheet forming (ISF) process using the finite element method (FEM) provide essential information for designing parts in automotive industries. However, solving numerous high-complexity FEM models during the designing phase requires many resources, leading to an increase in the final product's cost. This study presents a feedforward neural network (FFNN) to predict the deformed shape of an AA1050 sheet subjected to an ISF process. FEM solutions obtained from various vertical step size (Δz) of the forming tool are used to train and validate the FFNN. The model is then used to predict the deformed shape demonstrating by the displacement in the forming depth direction. The norm of the relative errors between the FFNN solution and FEM solution at the last forming step is about 2%. The predictive results illustrate the feasibility and potential of using FFNN as an efficient surrogate model to replace the time-consuming FEM-based ISF process simulation.

Keywords Incremental sheet forming · Finite element method · Aluminum alloys sheet · Plasticity deformation · Feedforward neural network

1 Introduction

In recent decades, incremental sheet forming (ISF), a die-less forming process, has been actively developed to manufacture parts requiring small-batch productions such as prototypes and complicated components [1]. In this process, a rigid tool is programmed to move following designated tool paths and plastically deforms a blank sheet into pre-designed shapes. Several process parameters influence the

H. S. Le · A. T. Nguyen · H. S. Tran · X. V. Tran (✉)
Institute of Strategies Development, Thu Dau Mot University, Binh Duong, Vietnam
e-mail: xuantv@tdmu.edu.vn

Q. T. Pham
Division of Computational Mathematics and Engineering, Faculty of Civil Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
e-mail: phamquoctuan@tdtu.edu.vn

A. Kumar et al. (eds.), *Proceedings of the International Conference on Cognitive and Intelligent Computing*, Cognitive Science and Technology,
https://doi.org/10.1007/978-981-19-2350-0_4

formability and qualities of the final products manufactured by ISF, for example, tool diameter, tool moving speed, and vertical step size [2]. Notably, an optimization procedure commonly for ISF requires a large number of finite element method (FEM) simulations to achieve the optimized process parameters [3]. That raises the demand for the development of surrogate models, which are able to provide numerical approximations for the complex input–output relationship within an efficient time.

In machine learning (ML)-related publications, for example, Ehsan et al. [4] proposed a multi-network of physics-informed neural network (PINN) to achieve more accurate predictions on various solution fields in linear elastic and nonlinear plastic problems. In addition, Pham et al. [5] adopted a back-propagation neural network (BPNN) to search for Pareto optimal solutions of formability and thickness of AA5052-H32 sheets subjected to an ISF process.

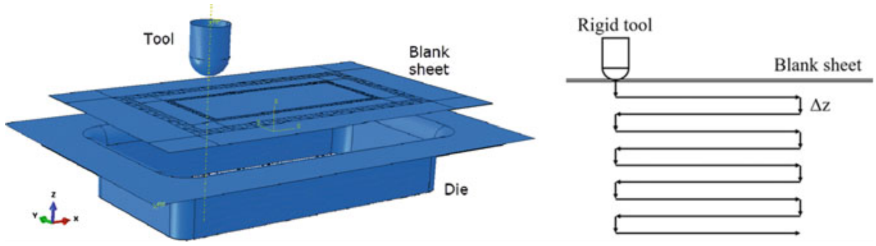
This study aims to develop a surrogate model to predict the displacement fields obtained from FEM solutions of an ISF process. Several FE models are developed to simulate the ISF groove tests performed for AA1050 sheets, which vary the vertical step size (Δz). Based on the simulation results, a feedforward neural network (FFNN) is trained to address the relationship between the imposed Δz and the displacement fields obtained from FE simulations of the ISF process. After validation, a comparison between the FFNN prediction and FEM solutions for the displacement fields of a test performed with an unseen value of Δz is presented. Furthermore, the usefulness of the trained FFNN model is discussed.

2 Data Acquisition and Neural Networks

This section describes the detailed setup configuration of the FEM model used to simulate the ISF groove test. Later, a strategy to achieve the useful data is introduced based on simulated results. Finally, the most suitable architecture of FFNN, as well as the data structure, is discussed.

2.1 Data Acquisition via a Virtual Groove Test

Figure 1a shows the assembly of the groove tests [6]which are developed in Abaqus/Explicit for the aluminum sheet AA1050 with an initial thickness of 0.3 mm [7]. In this test, a rigid tool with a diameter of 10 mm is programmed to move following the tool path presented in Fig. 1b. The blank sheet is modeled by 4 node shell elements with reducing integration (S4R) with a size 1×1 mm^2 of the smallest element. Totally, the sheet is modeled with 2573 nodes and 2577 elements. Whereas, tool and die are modeled by discrete rigid quadrilateral elements (R3D4).



(a) FEM model (b) Tool path

Fig. 1 Finite element model to simulate the groove test

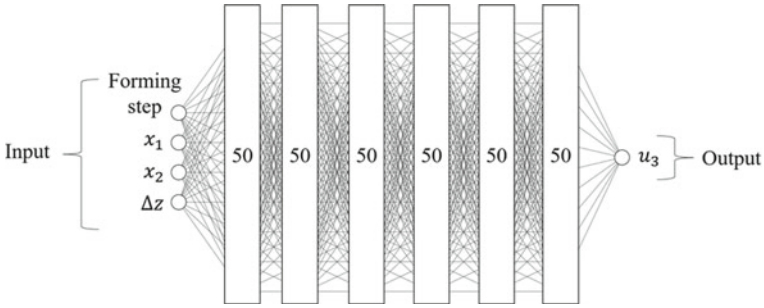


Fig. 2 Proposed FFNN architecture

2.2 Feedforward Neural Network (FFNN) Architecture

After a trial–error procedure, the most suitable FFNN architecture for re-building the deformed shape is presented in Fig. 2.There are four essential input features of the developed FFNN, including (i) the forming step, (ii) and (iii) $x - y$ coordinates of the point of interest on the initial blank sheet (labeled as $x_1 - x_2$), and (iv) Δz value. Consequently, the output is the displacement of the considering point in the Z-direction. There are six hidden layers with 50 neurons each. The linear activation is used on the output layer to produce the desired output values.

To generate data used in the FFNN model,2573-dimensional vectors of displacements in the thickness direction of all nodes are recorded according to various values of the forming step Δz (i.e., Δz = 0.5, 0.6, 0.7, 0.8, 0.9, 1.0). Six data samples of different Δz values are split into the training, cross valid, and test sets. In the following work, the developed neural network structure is used to train two models, named FFNN 1 and FFNN 2. The former is trained with three data samples, while the latter is trained with four data samples. After each training epoch, both FFNN models perform the prediction on a cross valid set (i.e., $\Delta z = 0.6$) to record the overfitting phenomena. When the training phase ends, the performance of trained

Table 1 Two dataset structures for training and testing the proposed FFNN

Training	FFNN 1	FFNN2
	$\Delta z = [0.5, 0.8, 1.0]$	$\Delta z = [0.5, 0.8, 0.9, 1.0]$
Cross-validation	$\Delta z = [0.6]$	$\Delta z = [0.6]$
Testing	$\Delta z = [0.7, 0.9]$	$\Delta z = [0.7]$

models for the remaining test set is used to evaluate the goodness of these models. The detailed dataset is reported in Table 1.

3 Results and Discussion

The FFNN models were implemented in Python using TensorFlow [8]. An NVIDIA GeForce GTX 1030 graphics processing unit (GPU) with a 3.6 GHz quad-core CPU and 8 GB RAM are used for data generation and FFNN training. The adaptive moment estimation algorithm [9] is used to update the network weights with a learning rate of 0.001. The batch size is set up at 12,800. The loss function is the mean squared error (MSE) metric defined as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(u_i^{\mathrm{FFNN}} - u_i^{\mathrm{FEM}}\right)^2, \tag{1}$$

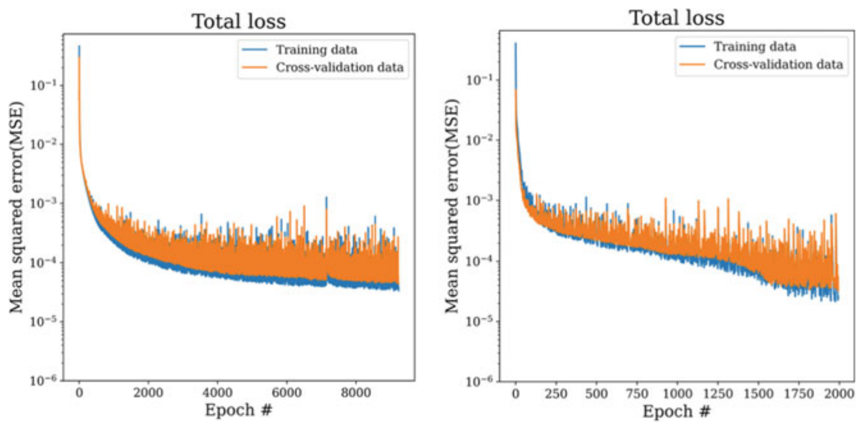
where u_i^{FFNN} and u_i^{FEM} are the displacements at the node i obtained from the FFNN and FEM model, respectively, and N is the number of nodes containing in the FEM simulation.

Figure 3 illustrates the training and validation errors of the proposed FFNN architecture for the two training models. Generally, by adding one more sample for the training data, the FFNN2 convergence rate is much faster (i.e., epoch = 1993) than that of FFNN 2 (i.e., epoch = 9231).

Table 2 shows the comparison between the calculation time of the FEM and FFNN models. It is seen that both two FFNN models are well-trained within few minutes; meanwhile, the prediction time of the trained models is less than the second. In contrast, the FEM model requires almost 50 min to finalize the results. Although the

Table 2 Computation time of FEM and FFNN models

Model	Computing time (s)	
	Training	Prediction
FFNN 1	513.0	0.373
FFNN 2	198.0	
FEM		2983.0



(a) FFNN 1

(b) FFNN2

Fig. 3 Convergence of the proposed **a** FFNN 1 trained with three data samples and **b** FFNN 2 trained with four data samples

efficiency of the trained FFNN models outperforms the conventional FEM model, collaborating the two algorithms may be an efficient way to modeling and analysis the ISF process, particularly.

To evaluate the reliability of the FFNN solutions, the relative error norm metric, which estimates the difference between the FFNN predicted vector solution and those of the FEM model, is calculated as follows at each simulation step:

$$\eta = \frac{\left\| \mathbf{u}_k^{\text{FEM}} - \mathbf{u}_k^{\text{FFNN}} \right\|_2}{\left\| \mathbf{u}_k^{\text{FEM}} \right\|_2} \times 100\%, \qquad (2)$$

where $\| \, \|$ denotes the L^2 norm; $\mathbf{u}_k^{\text{FEM}}$ and $\mathbf{u}_k^{\text{FFNN}}$ are the solution vectors at the forming process step k of the FEM and FFNN models, respectively. Figure 4 shows the norm's evolution based on the cross-validation and test samples. It is indicated that the performance of the FFNN model trained with four samples (i.e., dash lines) is generally better than that of the one trained with three data samples (i.e., solid lines).In practice, the simulation results achieved at the final forming step are the most important during an ISF process. As shown in Fig. 4, in the last forming step, the norm of the FFNN model trained with four data samples is even lower than 2%, which indicates a good approximation.

To illustrate a comprehensive picture regarding the two FFNNs' performance on the test data sample at the final forming step, the absolute error between the FEM solutions and FFNNs' predictions is point-wised in Fig. 5. Comparison between the error distribution exhibited in Fig. 5b, c clarifies that adding more samples to the training dataset increases the accuracy of the FFNN predictions. After all, it is reasonable to conclude that the FFNN expresses great potential and efficiency to

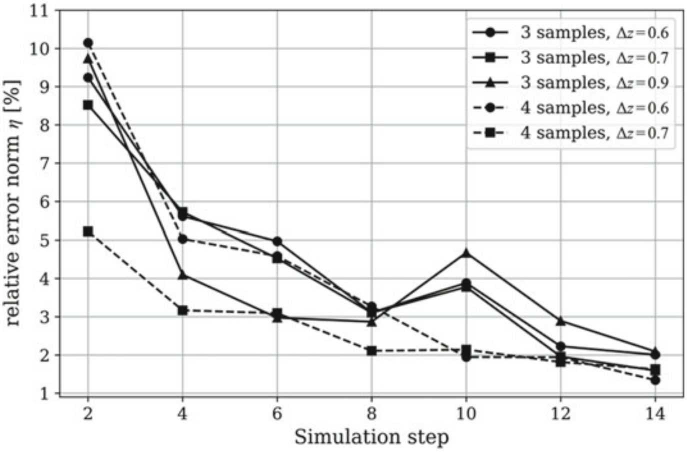


Fig. 4 Relative error norm between the solution vector of FFNN model and FEM model

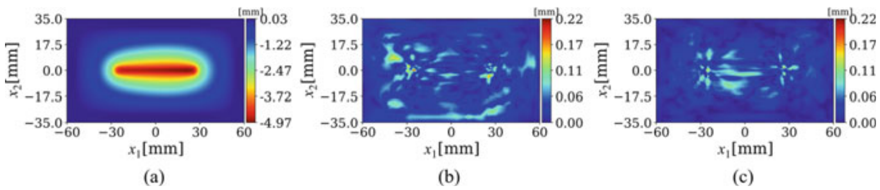


Fig. 5 Forming step number 14: **a** the displacement distribution of the FEM model, **b** error distribution of the FFNN 1 model trained with three samples, **c** error distribution of the FFNN 2 model trained with four samples

reproduce a reliable approximation for the FEM simulations of the considering ISF process.

4 Conclusion

In this work, two simple FFNN model strained with three and four samples for predicting the displacement field in the ISF process are well developed. The validation stage indicates that the model can predict the displacement field of the ISF process at the last forming steps with a relative error norm below 2%. The proposed framework shows the developed FFNN's potential to be a useful surrogate model that can save considerable computing time. From perspective, considering more parameters for the FFNN model, such as the material properties to increase the complexity of high-fidelity solutions, which could be a challenge and the motivation in the further work.

Acknowledgements This work was funded by Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF. 2020.DA15.

References

1. Nguyen D-T, Park J-G, Lee H-J, Kim Y-S (2010) Finite element method study of incremental sheet forming for complex shape and its improvement. Proc Inst Mech Eng Part B J Eng Manuf 224(6):913–924. https://doi.org/10.1243/09544054JEM1825
2. Li Y, Chen X, Liu Z, Sun J, Li F, Li J, Zhao G (2017) A review on the recent development of incremental sheet-forming process. Int J Adv Manuf Technol 92(5–8):2439–2462. https://doi.org/10.1007/s00170-017-0251-z
3. Xiao X, Kim J-J, Hong M-P, Yang S, Kim Y-S (2020) RSM and BPNN modeling in incremental sheet forming process for AA5052 sheet: multi-objective optimization using genetic algorithm. Metals 10(8):1003. https://doi.org/10.3390/met10081003
4. Gaddam DKR, Ansari MD, Vuppala S, Gunjan VK, Sati MM (2022) A performance comparison of optimization algorithms on a generated dataset. In: ICDSMLA 2020. Springer, Singapore, pp 1407–1415
5. Pham QT, Nguyen HQ, Tran VX, Xiao X, Kim JJ, Kim YS (2021) Process parameter optimization for incremental forming of aluminum alloy 5052-H32 sheets using back-propagation neural network. In: Research in intelligent and computing in engineering. Springer Singapore, pp 585–594
6. Narayana GS, Ansari MD, Gunjan VK (2022) Instantaneous approach for evaluating the initial centers in the agricultural databases using K-means clustering algorithm. J Mob Multimedia, pp 43–60
7. Szachogluchowicz I, Sniezek L, Slezak T, Kluczyński J, Grzelak K, Torzewski J, Fras T (2020) Mechanical properties analysis of the AA2519-AA1050-Ti6Al4V explosive welded laminate. Materials 13(19):4348. https://doi.org/10.3390/ma13194348
8. Rashid E, Prakash M, Ansari MD, Gunjan VK (2021) Formalizing open source software quality assurance model by identifying common features from open source software projects. In: ICCCE 2020. Springer, Singapore, pp 1375–1384
9. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. In: International conference on learning representations