Symbolic-Based Recurrent Neural Networks for Metamodeling of Nonlinear Structural Models

Yiming Jia
Department of Civil and Environmental Engineering
Northeastern University
Boston, United States
jia.yim@northeastern.edu

Mehrdad Sasani
Department of Civil and Environmental Engineering
Northeastern University
Boston, United States
sasani@neu.edu

***Abstract*—Nonlinear time history analyses of structural models, widely used in civil engineering, can be time-consuming. For an urban scale, modeling a large number of structures and carrying out nonlinear time history analyses under different levels of hazard intensity is not practical. This can be resolved by the use of metamodeling, which can effectively reduce the computational cost while maintaining necessary engineering accuracy. Neural networks have been proven to be a powerful tool for metamodel development. However, training a reliable deep learning model requires a large training dataset that contains representative input-output relationships, which typically cannot be satisfied in practical applications. Practically, a trained deep learning model is a "black box", which has limited (or no) generalizability. In this paper, a symbolic-based neural network that does not require an immense amount of data is proposed to capture nonlinear structural behavior. In symbolic-based neural networks, symbolic activation functions are capable of finding mathematical expressions to describe the mechanism of input-output relationship, and the hidden state can store the sequence information of nonlinearity. Case studies are carried out for reinforced concrete structures subjected to a series of selected pulse-type ground motions. The results show that the symbolic-based neural network is a promising approach for estimating the nonlinear building responses without a large training dataset.**

Keywords—neural network, metamodeling, nonlinear structural response, numerical integration, ordinary differential equation.

I. Introduction

Nonlinear time history analysis of structural models can provide reliable information regarding building behaviors under hazard. It is, therefore, used extensively by civil engineers and practitioners for multiple objectives, such as building performance estimation and community resilience evaluation. For an urban scale, there is a need to carry out reliable nonlinear time history analyses under different levels of hazard intensity. This requires a large number of repeated analyses, which can be time-consuming and impractical. One way to address this issue effectively is metamodeling, which can enable computationally efficient analysis of complex structures. Recent studies have shown that deep learning is a promising approach to metamodeling for predicting nonlinear structural responses [1-5]. However, training a reliable deep learning model requires a large amount of data that contains representative input-output relationships, which typically cannot be satisfied in most engineering practices. A deep learning model is a "black box" model, which highly depends on the quality of training data, leading to low accuracy and generalizability outside available data (e.g., training datasets). Additionally, a trained deep learning model is not user-friendly for engineers, who may not have background knowledge of deep learning.

A symbolic-based recurrent neural network (SRNN), which can effectively overcome the aforementioned limitations of deep learning, is proposed for the metamodeling of nonlinear building responses. An SRNN is a recurrent neural network that uses symbolic activation functions and is designed by leveraging domain-specific knowledge. The symbolic activation functions in SRNNs (e.g., sign, absolute value, sine, cosine, square, and multiplication) can describe the functional relationships between inputs and outputs, and in turn discover underlying closed-form formulations, thereby making the SRNN generalizable. The hidden state of SRNN, which links two adjacent time steps, can store sequence information and be useful for learning the nonlinearity of time series data. Domain-specific knowledge and fundamental principles of existing metamodels are embedded in SRNN via variations of input data and selections of mathematical operators. These knowledge and principles can provide constraints to the parameters of SRNN, alleviate overfitting issues, and in turn reduce the need for large training datasets (e.g., > 100 sets of building responses in [5]). Unlike other deep learning models (e.g., CNN, LSTM, transformers), the outcomes of SRNN can be simplified as an ordinary differential equation. The application of SRNN can be easily achieved by any numerical integration method, which is readily accessible for engineers.

In this paper, SRNNs are used to estimate the nonlinear responses of reinforced concrete structures under seismic excitation.

II. Symbolic-Based Recurrent Neural Networks

The SRNN for nonlinear building response estimation is analogous to finding the closed-form formulation of a governing equation of motion. Using a single degree of freedom (SDOF) system for illustration, the governing equation of motion at time i (t_i) can be expressed as

$$\ddot{u}_i + f[u_i, \dot{u}_i, t_i, h(u_i, \dot{u}_i, t_i, h_{i-1})] = -\gamma a_{g_i} \qquad (1)$$

where u_i, $\dot{u}_i$, and $\ddot{u}_i$ = displacement, velocity, and acceleration relative to ground at t_i; h = hidden state that stores the information from the previous time step; f = mass-normalized internal restoring force; a_{g_i} = ground motion (GM) acceleration

CMMI of the National Science Foundation.

at t_i; γ= influence scalar, which is set equal to one here. As shown in Fig. 1, the SRNN includes two parts – functions h and f – which are both modeled as symbolic neural networks [6-8], denoted as SNN_H and SNN_F.

The inputs of SRNN include u_i, $\dot{u}_i$, and the signs of u_i and $\dot{u}_i$ [6, 7]. The sine and cosine functions of u_i and $\dot{u}_i$, which can increase the model accuracy [7-9], are also included as the inputs of SRNN. As observed in parametric studies, including the relative time (τ_i), which varies from 0 to the building's fundamental period at the given time interval, as an input of SRNN can also increase the model accuracy. Additionally, u_i / ($|u_i|$+1) is included to model post-yielding softening behavior [10]. In order to learn the potentially complex input-output relationship, the proposed SNN_H includes two layers of commonly used symbolic activation functions, which can produce a polynomial function up to the fourth order (see Fig. 2 (a)). Using the concept of residual neural network [11] and analogous to current symbolic neural networks [6-8, 12], SNN_H allows inputs (and outputs of hidden layers) to pass directly to the following layers, which can significantly reduce the number of parameters while maintaining required accuracy. The proposed SNN_F is a fully connected network that includes only one layer, which is the linear combination of inputs (see Fig. 2 (b)).

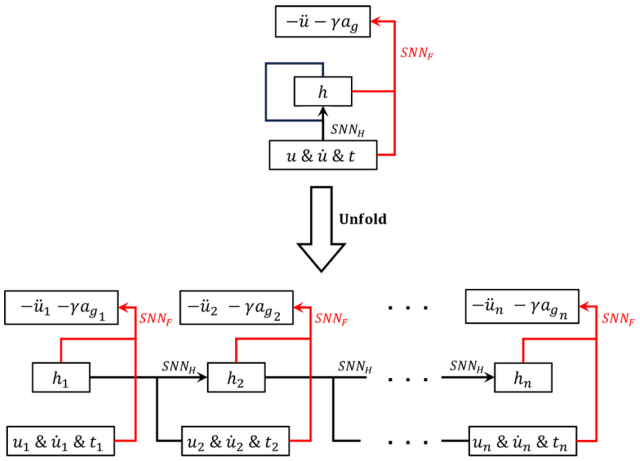

Fig. 1. The SRNN designed for building response prediction from time step 1 to n.

III. Training Algorithm

As shown in Fig. 1, except for the hidden state at the first time step (h_1, which is initialized as 0), there is no information known about the hidden states of the SRNN prior to training. More specifically, from the second time step, one input used in the SNN_F calculation is unknown. This implies that the SRNN cannot be simply trained by minimizing the difference between recorded and estimated acceleration time series. One way to resolve this issue is by embedding SRNN with numerical integration. Numerical integration can link building responses between two adjacent time steps via a time-integration scheme and provide inputs for the hidden state. Using SRNN embedded with numerical integration, the time series of displacement, velocity, and acceleration can be estimated together; training can be conducted by minimizing the differences between these time series. The corresponding loss function is

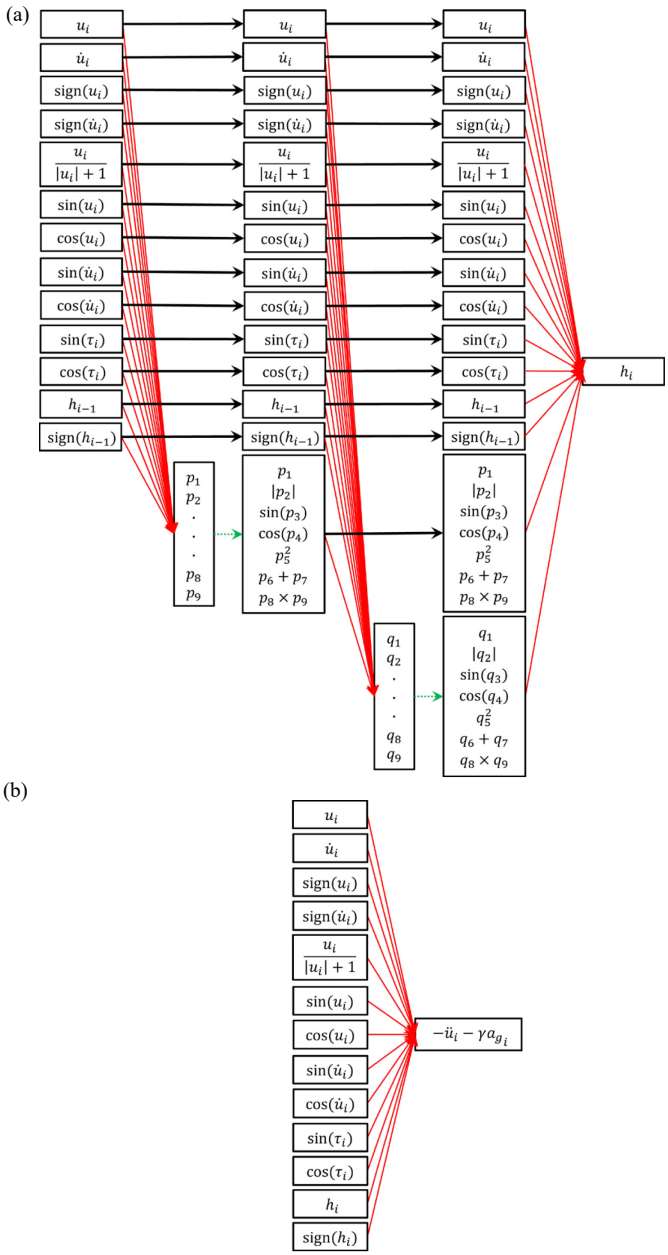

Fig. 2. Architectures of two parts of the SRNN (a) SNN_H and (b) SNN_F (solid thick black arrow = identity, solid thin red arrow = fully connected with weights and bias, dotted green arrow = mathematical operations).

$$\text{Loss} = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_{ij} - \hat{\mathbf{Y}}_{ij})^2 \qquad (2)$$

where $\mathbf{Y}_{ij}$ = $[\alpha \cdot u_{ij}, \beta \cdot \dot{u}_{ij}, \ddot{u}_{ij}]$, a vector including the recorded displacement, velocity, and acceleration of the jth training time series at the ith time step; α and β = hyperparameters to increase the contributions of displacement and velocity in loss, respectively, which can be estimated based on the ratios between the magnitudes of displacement and velocity to acceleration; $\hat{\mathbf{Y}}_{ij}$ = vector including the displacement, velocity, and acceleration estimated by the SRNN; n = number of time steps of the jth training time series; m = number of training time series.

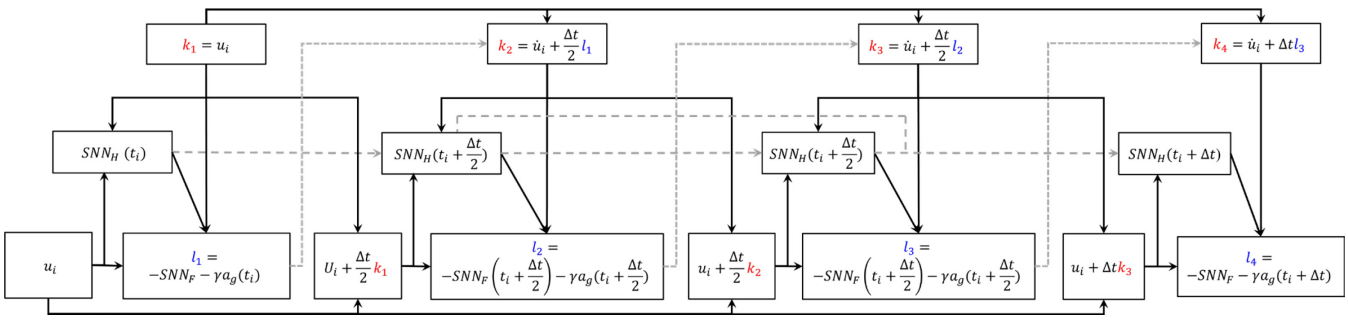

Fig. 3. Slopes of displacements and velocities used in the RK embedding with SRNN ($a_g\left(t_i + \frac{\Delta t}{2}\right) \approx \frac{a_g(t_i) + a_g(t_i + \Delta t)}{2}$, $SNN_H(t_i + \Delta t)$ is esitmated using the average value of two $SNN_H(t_i + \frac{\Delta t}{2})$).

A. Fourth Order Runge-Kutta Integration

For illustrative purposes, the fourth order Runge-Kutta integration method (RK4) [13] is used to embed with SRNN. Other numerical integration methods can also be used. In RK4, with displacement and velocity at t_i, the states at t_{i+1} can be estimated as

$$u_{i+1} = u_i + \frac{1}{6}\Delta t(k_1 + 2k_2 + 2k_3 + k_4) \quad (3)$$

$$\dot{u}_{i+1} = \dot{u}_i + \frac{1}{6}\Delta t(l_1 + 2l_2 + 2l_3 + l_4) \quad (4)$$

where k_1, k_2, k_3, and k_4 = slopes of displacement (see Fig. 3); l_1, l_2, l_3, and l_4 = slopes of velocity (see Fig. 3); and Δt = time interval between two adjacent time steps.

B. Parameter Pruning

As shown in Fig. 2, the SRNN includes several variations of displacement and velocity as inputs and candidate symbolic activation functions. Even for an SDOF system, the SRNN has a large number of parameters (i.e., 357). To improve the generalizability of SRNN, the trivial and redundant parameters are pruned using the following steps:

1) For the first 50% of training epochs, train the SRNN using the loss function as expressed in (2).

2) For the 50% to 75% of training epochs, train the SRNN using the following loss function, which is

$$\text{Loss} = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_{ij} - \widehat{\mathbf{Y}}_{ij})^2 + \delta(\mathbf{Y}_{top_j} - \widehat{\mathbf{Y}}_{top_j})^2\right] \quad (5)$$

where top_j = top few positive and negative values of displacements, velocities, and accelerations of the jth training time series (e.g., top 10); δ = user-defined hyperparameter larger than 1 that emphasizes the importance of peak building responses (e.g., 10).

3) For the 75% to 85% of training epochs, include the L1 norm of the SRNN's parameters in the loss function to shrink these parameters towards zero [14]. The loss function is

$$\text{Loss} = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_{ij} - \widehat{\mathbf{Y}}_{ij})^2 + \delta(\mathbf{Y}_{top_j} - \widehat{\mathbf{Y}}_{top_j})^2\right] + \lambda\|\boldsymbol{\Theta}\|_1 \quad (6)$$

where λ = user-defined hyperparameter to control the contribution of L1 norm to loss; $\|\boldsymbol{\Theta}\|_1$ = L1 norm of SRNN's parameters.

4) Set the SRNN's parameters with absolute values smaller than a user-defined threshold as zero [14]. This serves to prune out these trivial and redundant parameters.

5) For the 85% to 100% of training epochs, train the SRNN using the loss function as expressed in (5). This allows the remaining parameters to migrate to their appropriate values.

Note that the boundaries of training epochs (e.g., 50%, 75%, 85%) are user-defined. The aforementioned boundaries are the ones used in Section IV. Applications, Part C. Nonlinear Response Estimation for Single Degree of Freedom System.

IV. Applications

The nonlinear time history analysis results of simplified and detailed models of a 3-story reinforced concrete building under series of selected pulse-type GMs are used to evaluate the proposed SRNNs.

A. Building and Structural Models

A 3-story representative reinforced concrete building located in Financial District, San Francisco, California, is designed as a reinforced concrete special moment frame structure, according to ASCE 7 [15] Risk Category II at design hazard level. The typical floor plan of the designed building is shown in Fig. 4. The story heights are 4.27 m and 3.66 m for the first story and all other stories, respectively. The fundamental period of the designed building is about 0.6s.

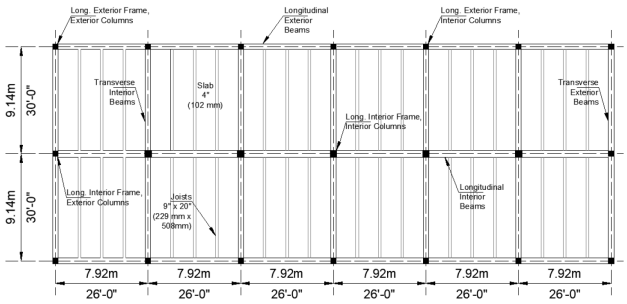

Fig. 4. Plan view of the designed 3-story representative reinforced concrete building.

A detailed 3D nonlinear model of the designed building is developed using OpenSees [16]. The distributed plasticity is included using nonlinear beam-column elements with sections discretized into concrete core, concrete cover, and steel fibers. The modified Kent and Park model is used to account for the confinement effects in the concrete core [17]. Buckling and bar-slip effects of reinforcing bars are included [18-21]. Additional information about the building design and the development of its detailed 3D nonlinear model is available [22]. The responses at the center node of each story are used to assemble a multiple degrees of freedom (MDOF) system for SRNN evaluation. A simplified 2D nonlinear model of the designed building is also developed based on [23], using a truss element with uniaxial hysteretic material in OpenSees [16]. The model's responses are used to assemble an SDOF system for SRNN evaluation.

B. Building Response Data

The nonlinear time history analyses of the simplified and detailed structural models of the designed building are conducted under a series of pulse-type GMs using OpenSees [16]. Note that only pulse-type GMs are selected because they typically have larger damage potential than ordinary GMs [28]. Using spectral acceleration as the GM intensity measure, the pulse-type GMs are selected by matching the median and median +/– logarithmic standard deviation of the conditional spectrum at the building's fundamental period [29-31]. Additional information about the GM selection is available [22]. The nonlinear responses of structural models obtained from these selected pulse-type GMs are randomly separated into two datasets – training and test (see Table 1). The training dataset is used to train SRNN to learn the governing equation(s) of motion. The test dataset is used to evaluate the generalizability of learned SRNN, more specifically, to test the performance of learned SRNN for unseen GMs. Note that for the detailed structural model, which is much more complex than the simplified one, the nonlinear time history analysis is performed at a small time interval (0.01s) to alleviate the convergence issue of finite element approximation in OpenSees [16].

TABLE I. DATASETS FOR SRNN EVALUATION

Structural model type	Number of			GM Time (s)	
	Degree(s) of freedom	*Training GMs*	*Test GMs*	*Interval*	*Duration*
Simplified	1	20	4	0.1	15
Detailed	3	6	1	0.01	12

C. Nonlinear Response Estimation for Single Degree of Freedom System

Since the simplified structural model is an SDOF system, the SRNN shown in Fig. 1 and Fig. 2 can be used without modification. The SRNN is trained using the Adam optimizer [32] for 1000 epochs with a batch size of 150. Note that the total number of time steps of one GM is 150, which indicates one GM's responses is used as a batch. The initial learning rate is 0.005 and decays 10% whenever the training loss stops decreasing for 4 epochs. The hyperparameter for the l_1 norm is set as 0.001. A threshold as 0.001 is used to prune the parameters of SRNN. Note that the aforementioned hyperparameters are obtained by a parametric study as multi-objective optimization for minimizing training time while maintaining reasonable prediction accuracy.

The number of parameters after pruning is 337, which is about 94% of the original 357 parameters. It is worth noting that, for SNN_H, $|p_2|$ is pruned out in the second layer. It means that the absolute value of linear combination of inputs is not needed. For SNN_F, cosine function of u_i is pruned out. The pruned out parameters can provide some guidance to the users when applying SRNN to similar engineering problems.

The generalizability of learned SRNN is evaluated as the prediction accuracy of nonlinear response estimation of simplified structural model under 4 unseen (test) GMs. The comparisons between the recorded nonlinear responses and the ones estimated by SRNN near the main pulse of one representative test GM are shown in Fig. 5. The Pearson correlation coefficient (ρ) between these two sets of building responses is used as a numerical measurement of prediction accuracy. Using displacement for illustration, ρ is calculated as

$$\rho = \frac{\sum_{i=1}^{n}[(u_i - \bar{u}) \cdot (\hat{u}_i - \bar{\hat{u}})]}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2 \cdot \sum_{i=1}^{n}(\hat{u}_i - \bar{\hat{u}})^2}} \tag{7}$$

where $\bar{u}$ and $\bar{\hat{u}}$ are the mean values of recorded and estimated displacements. The closer the ρ to 1, the higher the accuracy is. As shown in Fig. 5, all ρ's are larger than 0.95, which means a very high prediction accuracy (generalizability). Both the graphical (Fig. 5) and numerical (ρ) results indicate that the SRNN can accurately capture the nonlinear responses of simplified structural model. Also, the structural analysis time can be reduced from a few minutes (using OpenSees [16]) to a few seconds (using SRNN).

only the second story is adjacent. Three SRNNs are trained as a system of second order ordinary differential equations embedded with RK4. The training setups are similar to the ones for the simplified structural model. It is worth noting that the small time interval (0.01s) leads to a large number of time steps for one GM (1200). Based on the parametric studies, using the building responses of one GM as a training batch can be time-consuming and/or cause numerical instability. To resolve the aforementioned issues, a batch size of 120 is used, which corresponds to 1.2s – equal to two fundamental periods of the designed building. The hidden state is initialized as zero for each batch. It implies that the hidden state can only store the sequence information of nonlinearity for a portion of GM. This is a trade-off between training efficiency and model accuracy. As a result of this trade-off, the detailed structural model's SRNNs cannot reach the same level of accuracy as the simplified structural model's SRNN does, which uses one GM's responses as a batch. Consequently, the parameter pruning, which may further reduce prediction accuracy, is not performed for this MDOF case study.

The comparisons between the recorded and estimated nonlinear responses near the main pulse of the test GM are shown in Fig. 6. The SRNNs can finish the nonlinear response prediction in a few seconds, which is significantly faster than the 24 hours structural analysis using OpenSees [16]. However, near the peak displacements, the ρ for displacement varies from 0.83 to 0.88 (see Fig. 6 (c)), which is not as high as the one for simplified structural model (i.e., 0.96). It means that the generalizability of detailed structural model's SRNNs is lower than the one of simplified structural model, which may be due to the following reasons:

1) The error between the closed-form formulations estimated by SRNNs and the governing equations of motion is drastically accumulated throughout integrations, especially for this test GM with a large number of time steps (1200).

2) The hidden state initialization for each batch leads to that the hidden state can only store a portion of (not all) the sequence information of nonlinearity.

3) The training dataset is comparably small (6 pulse-type GMs compared with 20 pulse-type GMs for SDOF system). The limited training data may not provide sufficient information for the detailed structural model's SRNNs to comprehensively learn the complex nonlinear behaviors.

If the user requires the model accuracy to be further improved, it can be achieved by 1) including a layer of symbolic activation functions in SNN_F to produce a polynomial function and 2) increasing batch size and using a different numerical integration method (e.g., Euler's method).

V. Conclusions

SRNN designed by leveraging domain-specific knowledge and fundamental principles of existing metamodels can approximately capture nonlinear building responses with low computational cost while maintaining necessary engineering accuracy. Training SRNN does not require a large dataset, and the learned SRNN can have high generalizability (i.e., the prediction accuracy of structural nonlinear response under an unseen GM). The results of SRNN are two sets of equations obtained from SNN_H and SNN_F, which can construct an ordinary differential equation. It means that, when applying SRNN, only a numerical integration is needed, no deep learning knowledge is required. This makes SRNN to be a user-friendly approach that can be easily implemented by engineers. The following conclusions are drawn based on the applications:

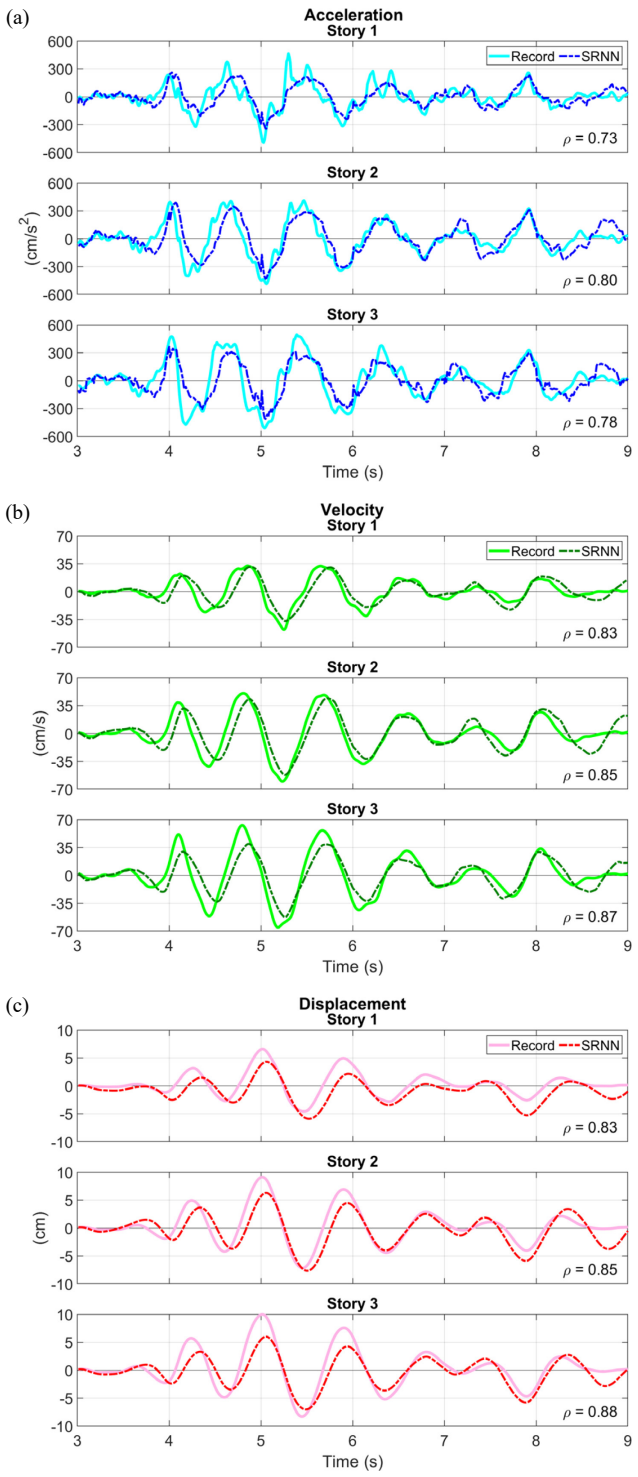

Fig. 6. Comparison between the nonlinear responses of the detailed structural model under the test GM estimated by the SRNNs and recorded by nonlinear time history analysis at each story for (a) accelerations, (b) velocities, and (c) displacements.

- Including the variations of displacement, velocity, time as inputs, and commonly used symbolic activation functions in SRNNs can improve prediction accuracy.
- For an SDOF system, SRNN has high generalizability and is presented as accurately predicting nonlinear responses with a small training dataset (e.g., 20 pulse-type GMs).
- For an MDOF system, by leveraging the concept of shear-beam [23-27], SRNNs can capture an approximation of nonlinear responses. Due to the comparatively small training dataset (e.g., 6 pulse-type GMs) and hidden state initialization for each batch, the prediction accuracy is not as high as the one for the SDOF system.
- When the training time series has a large number of time steps (e.g., > 1000), hidden state initialization for each batch can be a potential bottleneck for model accuracy improvement.

The SRNNs presented in this paper can also be applied to other engineering problems in which the governing equations are ordinary differential ones.

ACKNOWLEDGMENT

This paper is based upon research supported by the National Science Foundation under Grant No. CMMI-2053741. The authors greatly appreciate this support.

REFERENCES

[1] R-T. Wu and M. R. Jahanshahi, "Deep convolutional neural network for structural dynamic response estimation and system identification," Journal of Engineering Mechanics, ASCE, vol. 145, no. 1, pp. 04018125: 1–25, 2019.

[2] B. K. Oh, Y. Park, and H. S. Park, "Seismic response prediction method for building structures using convolutional neural networks." Structural Control and Health Monitoring, vol. 27, no. 5, pp. e2519: 1–17, 2020.

[3] M. Stoffel, F. Bamer, and B. Markert, "Deep convolutional neural networks in structural dynamics under consideration of viscoplastic material behaviour," Mechanics Research Communications, vol. 108, pp. 103565: 1–6, 2020.

[4] R. Zhang, Y. Liu, and H. Sun, "Physicial-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling," Engineering Structures, vol. 215, pp. 110704: 1–13, 2020.

[5] R. Zhang, Y. Liu, and H. Sun, "Physics-informed multi-LSTM networks for metamodeling of nonlinear structures," Computer Methods in Applied Mechanics and Engineering, vol. 369, pp. 113226: 1–16, 2020.

[6] Z. Chen, Y. Liu, and H. Sun, "Forecasting of nonlinear dynamics based on symbolic invariance," Computer Physics Communications, vol. 277, pp. 108382: 1–16, 2022.

[7] Y, Jia., M. Sasani., and H, Sun, "Symbolic neural networks for surrogate modeling of structures," presented at the 14th International Conference on Application of Statistics and Probability in Civil Engineering, Dublin, Ireland, July, 2023.

[8] Z. Chen, Y. Liu, and H. Sun, "Symbolic deep learning for structural system identification," Journal of Structural Engineering, ASCE, vol. 148, no. 9, pp. 04022116: 1–14, 2022.

[9] A. T. Saric, A. A. Saric, M. K. Transtrum, and A. M. Stakovic, "Symbolic regression for data-driven dynamic model refinement in power systems," IEEE Transactions on Power Systems, vol. 36, no. 3, pp. 2390–2402, 2021.

[10] Y. Feng, Z-X. Guo, and Y-C. Gao, "An unconditionally stable explicit algorithm for nonlinear structural dynamics," Journal of Engineering Mechanics, ASCE, vol. 144, no. 6, pp. 04018034: 1–8, 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, June, 2016.

[12] Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network." Journal of Computational Physics, vol. 399, pp. 108925: 1–17, 2019.

[13] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae." Journal of Computational and Applied Mathematics, vol. 6, no. 1, pp. 19–26, 1980.

[14] S. S. Sahoo, C. H. Lampert, and G. Martius, "Learning equations for extrapolation and control," arXiv: 1806.07259, 2017.

[15] ASCE, "Minimum design loads for buildings and other structures," ASCE/SEI 7-22, Reston, Virginia, USA, 2022.

[16] F. McKenna, M. H. Scott, and G. L. Fenves, "Nonlinear finite-element analysis software architecture using object composition," Journal of Computing in Civil Engineering, ASCE, vol. 24, no. 1, pp. 95–107, 2010.

[17] B. Scott, R. Park, and M. Priestley, "Stress-strain behavior of concrete confined by overlapping hoops at low and high strain rates," Journal of the American Concrete Institute, vol. 79, no. 2, pp. 13–27, 1982.

[18] C. R. Urmson and J. B. Mander, "Local buckling analysis of longitudinal reinforcing bars," Journal of Structural Engineering, ASCE, vol. 138, no. 1, pp. 62–71, 2012.

[19] S. Sagiroglu and M. Sasani, "Progressive collapse-resisting mechanisms of reinforced concrete structures and effects of initial damage locations," Journal of Structural Engineering, ASCE, vol. 140, no. 3, pp. 04013073: 1–12, 2014.

[20] J. A. Muarry, E. Hecht, and M. Sasani, "Modeling bar slip in nonductile reinforced concrete columns," Journal of Structural Engineering, ASCE, vol. 142, no. 10, pp. 04016085: 1–12, 2016.

[21] NIST, "Guidelines for nonlinear structural analysis for design of buildings Part IIb – Reinforced concrete moment frames," NIST, Gaithersburg, Maryland, USA, 2017.

[22] S. Chan Esquivel, Y. Jia, and M. Sasani, "Earthquake resilience of spatially distributed building clusters: Methodology and application," Submitted to Journal of Structural Engineering, ASCE, 2023, unpublished.

[23] E. Miranda and S. Taghavi, "Approximate floor acceleration demands in multistory buildings I: formulation," Journal of Structural Engineering, ASCE, vol. 131, no. 2, pp. 203–211, 2005.

[24] F. Khoshnoudian and E. Ahmadi, "Effects of pulse period of near-field ground motions on the seismic demands of soil–MDOF structure systems using mathematical pulse model," Earthquake Engineering Structural Dynamics, vol . 42, no. 11, pp. 1565–1582, 2013.

[25] B. Ganjavi, I. Hajirasouliha, and A. Bolourchi, "Optimum lateral load distribution for seismic design of nonlinear shear-buildings considering soil-structure interaction." Soil Dynamics and Earthquake Engineering, vol. 88, pp. 356–368, 2016.

[26] O. Escalona and J. Wong, "Rigid body response & performance based design of seismically isolated structures," Proceedings of 11th U.S. National Conference on Earthquake Engineering, Los Angeles, California, June, 2018.

[27] M. D. Joyner and M. Sasani, "Building performance for earthquake resilience," Engineering Structures, vol. 210, pp. 110371: 1–14, 2020.

[28] NIST, "Selecting and scaling earthquake ground motions for performing response-history analyses," NIST, Gaithersburg, Maryland, USA, 2011.

[29] J. W. Baker, "Conditional mean spectrum: Tool for ground motion selection," Journal of Structural Engineering, ASCE, vol. 137, no. 3, pp. 322–331, 2011.

[30] T. Lin, S. C. Harmsen, J. W. Baker, and N. Luco, "Conditional spectrum computation incorporating multiple causal earthquakes and ground motion prediction models," Bulletin of the Seismological Society of America, vol. 103, no. 2A, pp. 1103–1116, 2013.

[31] J. W. Baker and C. Lee, "An improved algorithm for selecting ground motions to match a conditional spectrum," Journal of Earthquake Engineering, vol. 2, no. 4, pp. 708–723, 2018.

[32] D. R. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv: 1412.6980, 2014.