Automated Hyperparameter Tuned DNN for Predicting Uncertain Oscillations

Qasim Khadim, Grzegorz Orzechowski, Emil Kurvinen,
Aki Mikkola and Johannes Gerstmayr

May 31, 2024

The 7th International Conference on Multibody System Dynamics
June 9-13, 2023, Madison, WI, USA

Automated Hyperparameter Tuned DNN for Predicting Uncertain Oscillations

Qasim Khadim*, **Grzegorz Orzechowski#**, **Emil Kurvinen***, **Aki Mikkola#**, **Johannes Gerstmayr†**

* Machine and Vehicle Design
University of Oulu
90570 Oulu, Finland
[qasim.khadim, emil.kurvinen]@oulu.fi

\# Mechanical Engineering
LUT University
53850 Lappeenranta, Finland
[grzegorz.orzechowski, aki.mikkola]@lut.fi

†Department of Mechatronics
Leopold-Franzens-Universität Innsbruck
6020 Innsbruck, Austria
johannes.gerstmayr@uibk.ac.at

1 INTRODUCTION

The structural flexibility and oscillations present significant challenges for the precise and accurate control of mobile machines and heavy equipment, such as excavators, flexible robotic manipulators, and spacecrafts [1]. These challenges arise from the uncertain and nonlinear nature of the oscillations that occur in these structures. This problem is magnified by strain sensors installation in such applications, which may contribute to the implementation and economic difficulties. To solve this, this study introduces an automated hyperparameter-tuned deep neural network (DNN) approach [2] to predict uncertain oscillations in a system. The proposed DNN is applied to predict the uncertain oscillations of a mass in a nonlinear oscillator.

2 METHODS

Figure 1 demonstrates the architecture of DNN with automated hyperparameter tuning. This DNN, designed for regression tasks, learns the behavior of a system $\boldsymbol{\mathcal{Y}}$ by utilizing the input vector $\boldsymbol{\mathcal{X}}$ within a history window of i.

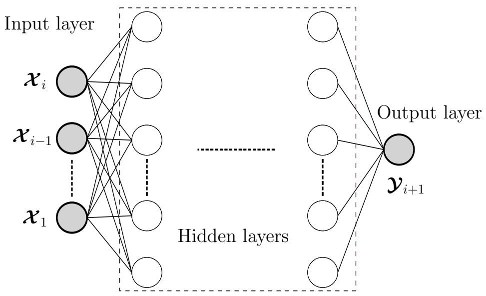

Figure 1: Architecture of a deep neural network with automated hyperparameter.

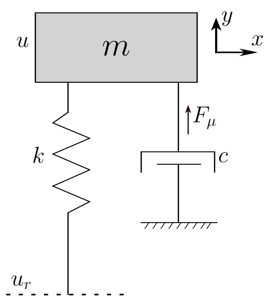

Figure 2: Uncertain mass oscillations in a one-dimensional nonlinear oscillator.

The regression relationship between $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ can be explained as [3],

$$\boldsymbol{\mathcal{Y}}_{i+1} = \boldsymbol{\mathcal{N}}(\boldsymbol{\mathcal{X}}_{1:i}; \boldsymbol{\Psi}), \tag{1}$$

where $\boldsymbol{\mathcal{N}}$ is the feedforward neural network (FFN) and $\boldsymbol{\Psi}$ is the vector of trainable parameters. The hyperparameters in the DNN architecture are searched using random optimization algorithm [2]. These hyperparameters include learning rate, number of layers, dropout, and activation function. The history window i depends upon the number of the steps observed for damped oscillations in a system.

Figure 2 shows a nonlinear oscillator. This system results in uncertain oscillations u in mass $m = 1$ kg due to the linear spring coefficient $k = 1.6 \times 10^3$ N/m, damping coefficient $c = 8$ Ns/m and the random base movement u_r. The force F in the nonlinear oscillator is described as,

$$F = k(u - u_r) + kk_0(u - u_r)^3 + c\dot{u} + F_\mu, \tag{2}$$

where $k_0 = 2\ \mathrm{m}^{-2}$ is coefficient of cubic stiffness and F_μ is the Stribeck frictional force [4] in the damper. The proposed DNN is trained, using Eq. (1), $\boldsymbol{\mathcal{X}} = u_{r_{1:i}}$ in a history window $i = 200$ for the damped oscillations and $\boldsymbol{\mathcal{Y}} = u_{i+1}$.

3 RESULTS AND CONCLUSIONS

Simulation data was obtained from the open source multibody Exudyn software [4] on the Windows Linux Subsystem, taking approximately 45 s. The training data set comprises 1024 simulations, while the validation set includes 256 simulations. This simulation data is used in a standard Keras Python environment to search hyperparameters with the random optimization algorithm. The hyperparameters search resulted in 3 hidden layers with $\tanh$ activation function, where each hidden layer has 64 nodes, dropout 0.1 and learning rate of 0.003. The DNN is trained with the tuned hyperparameter, and its training performance on the training and validation data is described in Figure 3. The proposed DNN took 3059 epochs in the training process, which took approximately 114 seconds. After verification, the trained DNN model is used

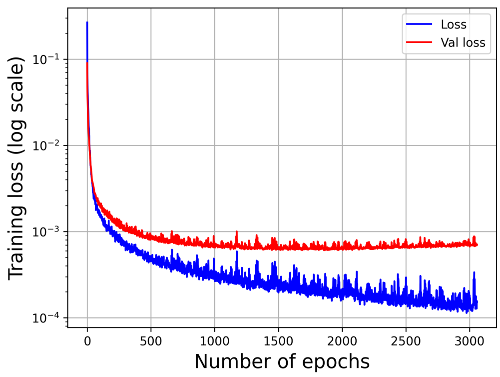

Figure 3: Training performance of DNN.

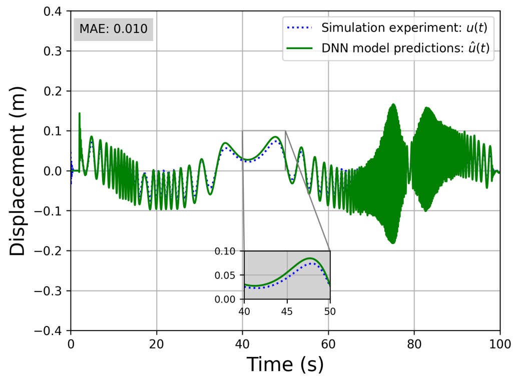

Figure 4: Predicting uncertain mass oscillations.

to estimate the uncertain oscillations of the mass, $\hat{u}$. During this simulation experiment, the base of the nonlinear oscillator is excited via a base motion $u_r = 0.2\left(\sin\left(\frac{h}{100-80\sin\left(\frac{h}{1000}\right)}\right) + \cos\left(\frac{h}{700}\right)\right)$, where h is the simulation step size. The input data is arranged in layers within a continuous frame, utilizing the history window i for faster computations. As depicted in Figure 4, the proposed DNN can predict $\hat{u}$ with a relative mean absolute error of 0.010, achieving faster than real-time simulations. The proposed approach offers promising prospects for enhancing the control and structural health monitoring of machines and equipment. Further research, however, is needed to explain this study in details, implications and broaden its applications to a multibody system, control algorithms and structural health monitoring in these domains.

REFERENCES

[1] Sarkhel, P.; Mithilesh, M. D.; et al.: Robust deflection control and analysis of a fishing rod-type flexible robotic manipulator for collaborative robotics. Journal of Machine Learning Research, Vol. 159, pp. 104293, 2023.

[2] Bergstra, J.; Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research, Vol. 13, No. 1, pp. 281–305, 2012.

[3] Pan, Y.; Ventura, C. E.; Li, T.: Sensor placement and seismic response reconstruction for structural health monitoring using a deep neural network. Bulletin of Earthquake Engineering, Vol. 20, No. 9, pp. 4513–4532, Springer, 2022.

[4] Gerstmayr, J.: Exudyn–A C++ based Python package for flexible multibody systems. Multibody System Dynamics, 2023.