ISBN: 978-3-95450-223-3 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2019-WEPLE06

LINEAR AND SECOND ORDER MAP TRACKING WITH ARTIFICIAL NEURAL NETWORK*

Y. Sun†, ANL, Lemont, IL, USA

Abstract

In particle accelerators, the tracking simulation is usually performed with symplectic integration, or linear/nonlinear transfer maps. In this paper, it is shown that the linear/nonlinear transfer maps may be represented by an artificial neural network. To solve this multivariate regression problem, both random datasets and structured datasets are explored to train the neural networks. The achieved accuracy will be discussed.

INTRODUCTION

In particle accelerators, the numerical tracking simulation is usually performed with either symplectic integration [1], or linear/nonlinear transfer maps [1]. For large storage rings, the particle accelerator is composed of thousands (or tens of thousands) of components. Tracking simulation through these components for hundreds of turns may take a long computing time. In this paper, it is preliminarily explored on representing the linear/nonlinear transfer maps with an artificial neural network.

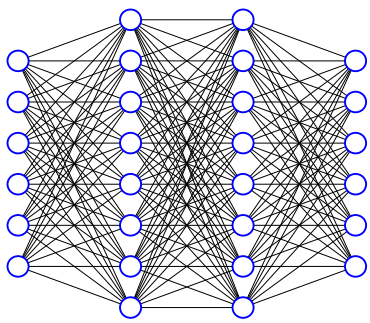

Figure 1: A fully connected feedforward network with two hidden layers and eight neurons on each hidden layer.

Using a basic neural network framework that has been developed in Python [2], it is possible to generate a fully connected feedforward neural network model. The details of this neural network framework are discussed in another paper of this proceedings [3]. Figure 1 shows a fully connected feedforward network with two hidden layers and eight neurons on each hidden layer, with input and output as particle's 3D coordinates. The default optimization algorithm *adam* and activations *Relu* are employed [3].

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

† yisun@anl.gov

LINEAR MAP

With particle's 3D coordinates of $\mathbf{X} = (x, x', y, y', z, \delta)$, it is possible to perform tracking with a linear map. The linear map can be expressed in the form of a six by six matrix $\mathbf{R}$. The particle's final 3D coordinates $\mathbf{Y}$ are then calculated using the initial 3D coordinates of $\mathbf{X}$ and the transfer matrix of $\mathbf{R}$, $\mathbf{Y} = \mathbf{R} \cdot \mathbf{X}$. Here, two different linear transfer matrix are employed to evaluate the robustness of the neural network model, as shown in Fig. 2.

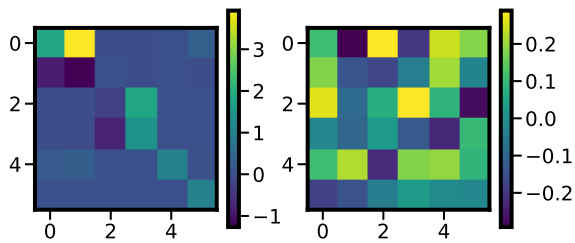

Figure 2: First order transfer matrix $\mathbf{R}$. Left: from a FODO cell; right: generated using random numbers.

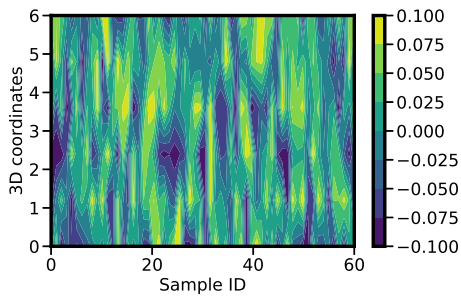

Figure 3: Input training data, size of 6 by 1000 (showing first 60).

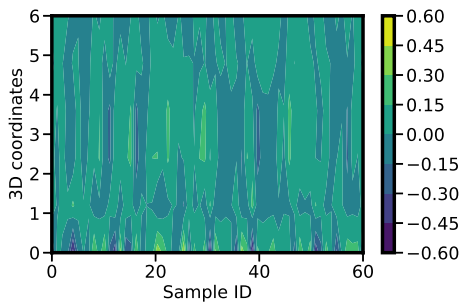

Figure 4: Output training data, size of 6 by 1000 (showing first 60).

The normalised input training data are generated randomly which comprises of 1000 different samples. The dimention

ISBN: 978-3-95450-223-3 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2019-WEPLE06

of the input training data is six. The output training data is calculated using the linear transfer matrix and the input training data. The training datasets are shown in Fig. 3 and Fig. 4.

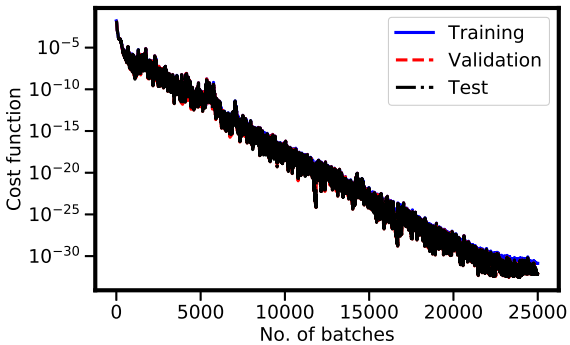

Figure 5: Cost (linear map) as a function of number of batches, for training/validation/test datasets.

In the case that there is no hidden layers, the neural network will simply find the linear map as shown in Fig. 2. Employing a four layer neural network as shown in Fig. 1, the training cost function of mean square error is plotted against the number of batches for training/validation/test datasets, as shown in Fig. 5. It is observed that very high accuracy is achieved for all three datasets. The differences between the artificial neural network predicted output $\hat{\mathbf{Y}}$, and the real output $\mathbf{Y}$, is then negligible, as shown in Fig. 6. This artificial neural network model can well represent the one turn linear map as expected.

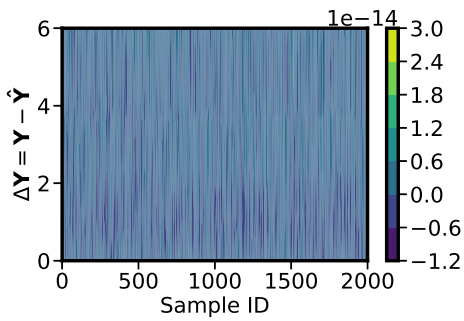

Figure 6: Differences between the artificial neural network predicted output $\hat{\mathbf{Y}}$, and the real output $\mathbf{Y}$.

SECOND ORDER MAP

In the case that the second order transfer map is also considered for particle tracking simulations, the particle's final 3D coordinates $\mathbf{Y}$ are then calculated using the initial 3D coordinates of $\mathbf{X}$ and the linear transfer matrix of $\mathbf{R}$ plus the second order transfer matrix of $\mathbf{T}$, $\mathbf{Y} = \mathbf{R} \cdot \mathbf{X} + \mathbf{T} \cdot \mathbf{XX}$. The combined linear transfer matrix of $\mathbf{R}$ plus the second order transfer matrix of $\mathbf{T}$ are shown in Fig. 7, which is generated using random numbers.

As mentioned in the above sections, for particle tracking simulations with second order map, both random input data

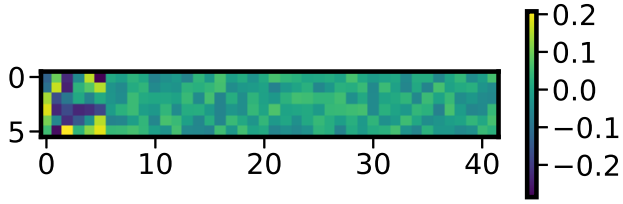

Figure 7: Combined linear transfer matrix of $\mathbf{R}$ plus the second order transfer matrix of $\mathbf{T}$, generated using random numbers.

and structured input data are employed. The structured input data is generated on equally spaced grids of normalized input spaces. An example of the input datasets is shown in Fig. 8.

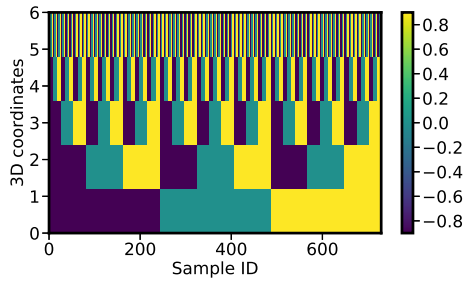

Figure 8: Structured input training data $\mathbf{X}$, size of 6 by 729.

A five layer feedforward neural network model is adopted for training. For this case, the neural network model is harder to train than for the linear map case. As shown in Fig. 9, the cost functions are reduced along number of batches for training/validation/test datasets. This figure seems to point out that a larger/denser neural network model may be needed to further improve the performance.

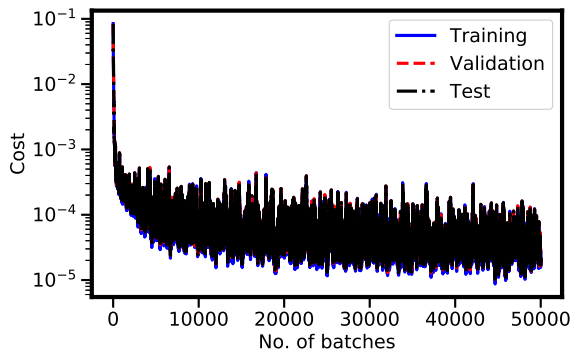

Figure 9: Cost (second order map) as a function of number of batches, for training/validation/test datasets.

The artificial neural network predicted output $\hat{\mathbf{Y}}$ and the real output $\mathbf{Y}$ are shown side by side in Fig. 10, for the test datasets. They seem to be identical, however the rms relative difference is roughly 2×10^{-3}.

A one dimentional scan on training data size shows that data size of 30k seems to be enough to achieve high accuracies for all three datasets (training, validation and test data).

ISBN: 978-3-95450-223-3 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2019-WEPLE06

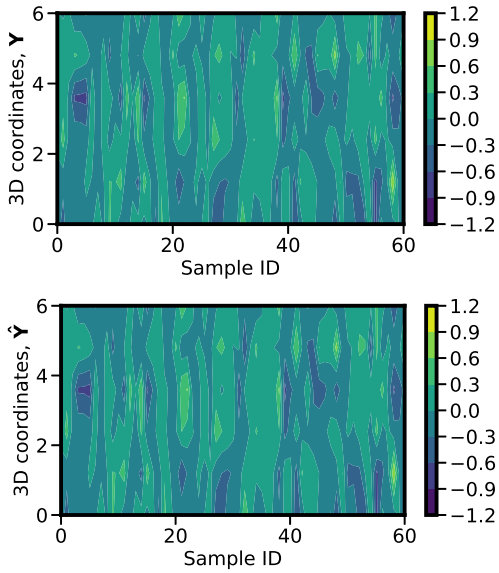

Figure 10: The artificial neural network predicted output $\hat{\mathbf{Y}}$ (bottom) and the real output $\mathbf{Y}$ (top) for the test datasets.

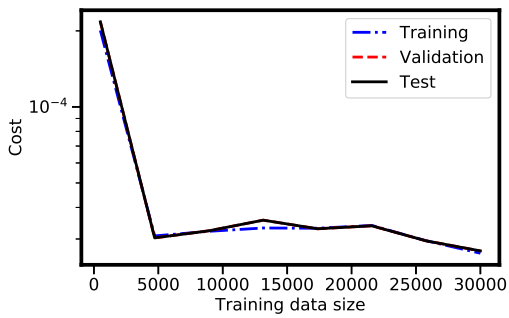

Figure 11: One dimentional scan on training data size, showing the cost for training, validation and test datasets.

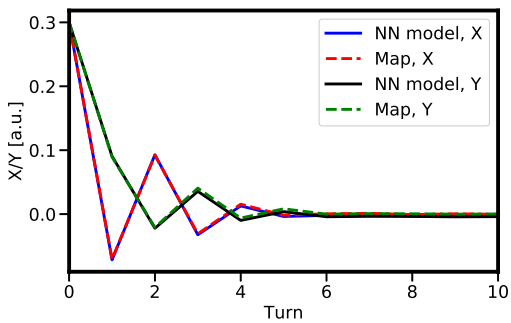

Figure 12: Comparison of particle tracking results (horizontal coordinate X and vertical coordinate Y), between neural network model and second order map.

The lowest cost found is around 9×10^{-6}. It is observed that the test dataset has the largest cost, as it is not involved in the training process. The results are shown in Fig. 11.

Particle tracking simulations are performed either with the second order map as discussed above, or with the trained neural network model. A comparison of particle tracking results of horizontal coordinate X and vertical coordinate Y is shown in Fig. 12. It is observed that the difference is relatively small and the amplitudes are damped to zero.

A two dimentional scan is performed on grids of learning rates and initialization random seed number. The initialization random seed number is from 0 to 20, while the learning rate is from 10^{-3} to 10^{-2}. The figure of merit here is the average cost. As shown in Fig. 13, for this specific neural network training problem, the best learning rate is around 0.003, where initialization random seed number introduces small difference.

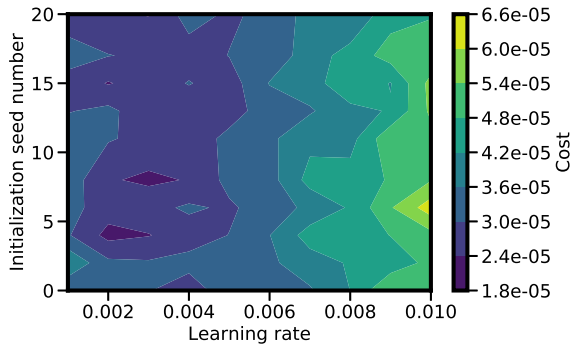

Figure 13: Two dimentional scan on learning rates and initialization random seed number.

CONCLUSIONS

Fully connected feedforward neural network models are generated, using a basic neural network framework that has been developed in Python. For particle tracking with linear map, it is possible to achieve very high accuracy even with a multi layer neural network. For particle tracking with linear and second order maps, an accuracy of 2×10^{-3} has been achieved with a relatively small neural network. Some hyperparameters are scanned for better performance. Comparison of particle tracking results demonstrate reasonable agreement between neural network model and second order map.

ACKNOWLEDGMENT

The author thanks Coursera [4] and Udemy [5] for providing free online courses on neural networks.

REFERENCES

[1] A. Chao and M. Tigner, *Handbook of Accelerator Physics and Engineering*, World Scientific, 1999.

[2] Python Software Foundation, "Python Language Reference." http://www.python.org

[3] Y. P. Sun, "Transfer Matrix Classification With Artificial Neural Network", presented at the North American Particle Accelerator Conf. (NAPAC'19), Lansing, MI, USA, Sep. 2019, paper WEPLE07.

[4] Coursera, https://www.coursera.org

[5] Udemy, https://www.udemy.com