WE4P19

Simulation Study of Orbit Correction by Neural Network in Taiwan Photon Source

M.S. Chiu[†], Caroline Flex, F.H. Tseng, Y.S. Cheng, H.J. Tsai, G.H. Luo

NSRRC, Hsinchu 300092, Taiwan

Abstract

Machine learning has been applied in many scientific and engineering fields in recent decades. Many research articles also presented remarkable achievements in applied to either operation or designing particle accelerator. This paper is focused on the simulated orbit correction by the algorithm of neural networks, a subset of machine learning, in Taiwan photon source. The training data for neural networks is generated by accelerator toolbox (AT).

Introduction

The Taiwan Photon Source (TPS) [1,2] is designed as a 3-GeV synchrotron light source, encompassing a 518.4m circumference. The lattice structure of the storage ring consists of 24 Double-Bend Achromat (DBA) cells, providing 18 short straight sections (7m) and 6 long straight sections (12 m). Three long straight sections, located at 3-fold symmetric position, adopt symmetrical double mini-β_y lattice in which a set of quadrupole triplet is installed in the middle of the long straight section to accommodate double undulators. Each of DBA cell is outfitted with 7 beam position monitors (BPMs). Two BPMs installed in the injection section are unused. There are additional six BPMs installed in three double mini-β_y sections. The TPS storage ring employs 72 horizontal and 96 vertical slow orbit corrector magnets to define the electron golden orbit, which is monitored by 172 BPMs.

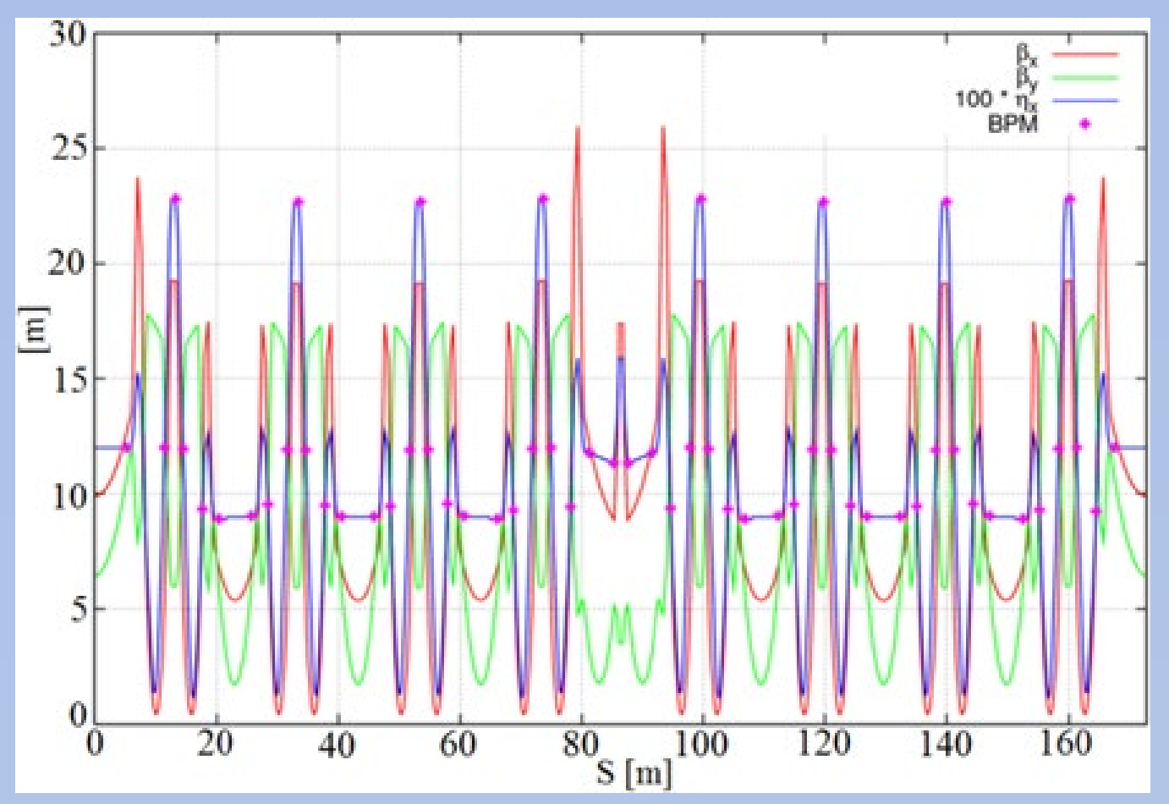


Figure 1 : Twiss functions for the double mini-β_y lattice in the 1/3 TPS storage ring.

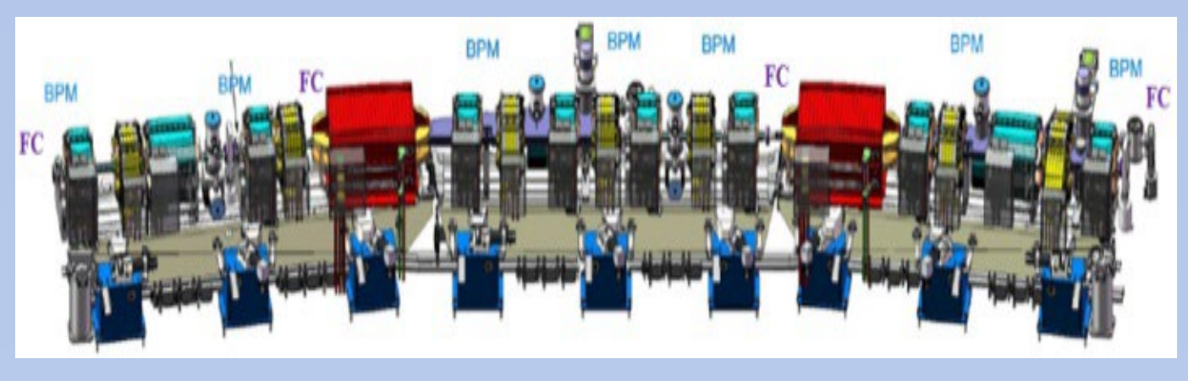


Figure 2: DBA cell.

Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI). It can enable computers to learn from huge data and make predictions or decisions without being explicitly programmed. ML can be roughly classified into three types: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model is trained on labeled dataset, meaning that the input data is paired with output or target values. The goal of supervised learning is to search a mapping from inputs to outputs. Once adeptly trained, the model can make precise predictions or classifications when encountered with new or unseen data.

Neural Network

Neural network, a subset of machine learning, is a computational model inspired by the human brain's structure. It can assist us to find out the rules or relationships between the input and output of a nonlinear or complex system. A typical architecture of the feedforward neural network is shown in Fig. 3. For simplicity, we will use one hidden layer as an example. The circles in each layer stand for neurons, called nodes. The linking arrows in between layers show the signal transduction pathways. The input layer is used to feed data. The output layer gives us the predictions by the neural network. The hidden layer is the main processing units in the neural network to process the data. Usually in each neuron, it executes two things: (a) sum the data passed from the previous layer multiplied by a weight matrix W and then add a bias value B, (b) pass the weighted sum of the data to an activation function f to make a transformation. After that, the transformed data is sent to next layer.

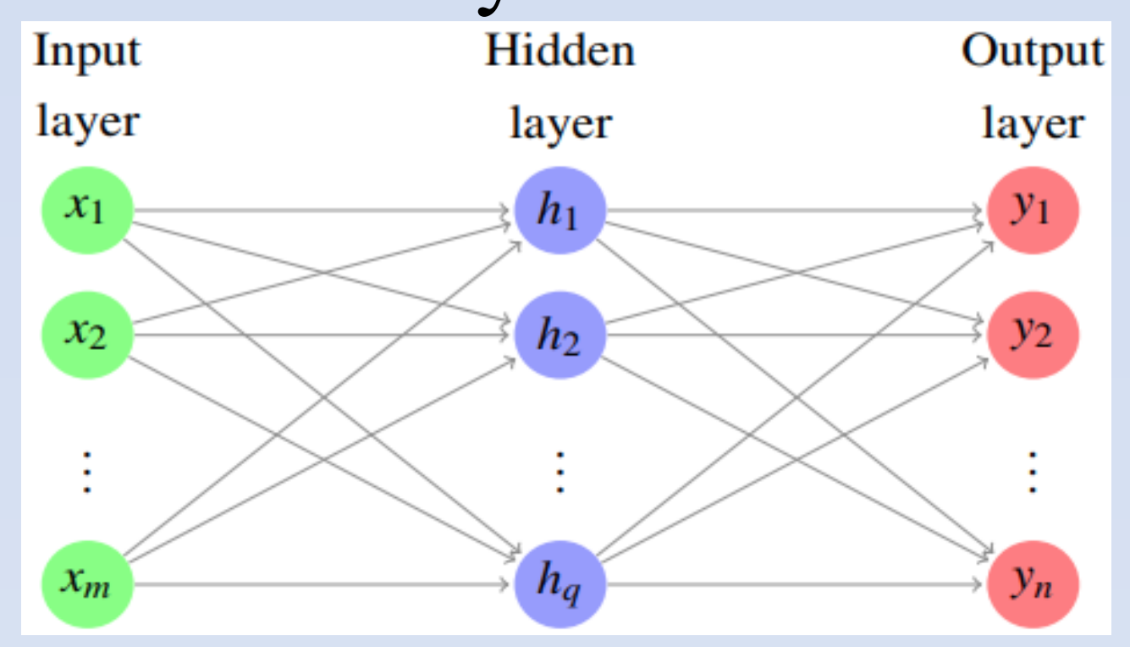


Figure 3: Feed-forward neural network

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_q \end{bmatrix} = f(W_1X+B_1) = f\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q1} & w_{q2} & \cdots & w_{qm} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \right)$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = W_2H+B_2 = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1q} \\ w_{21} & w_{22} & \cdots & w_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nq} \end{bmatrix} \times \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_q \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

, where X is the input data, Y is the outputs of the neural network. H stands for the outputs from hidden layer. W_1 and W_2 are weight matrix. B_1 and B_2 are bias vectors. The subscript m, q, and n are the neuron number in input layer, hidden layer, and output layer, respectively. f is an activation function. Here, assuming the activation in the output layer is linear.

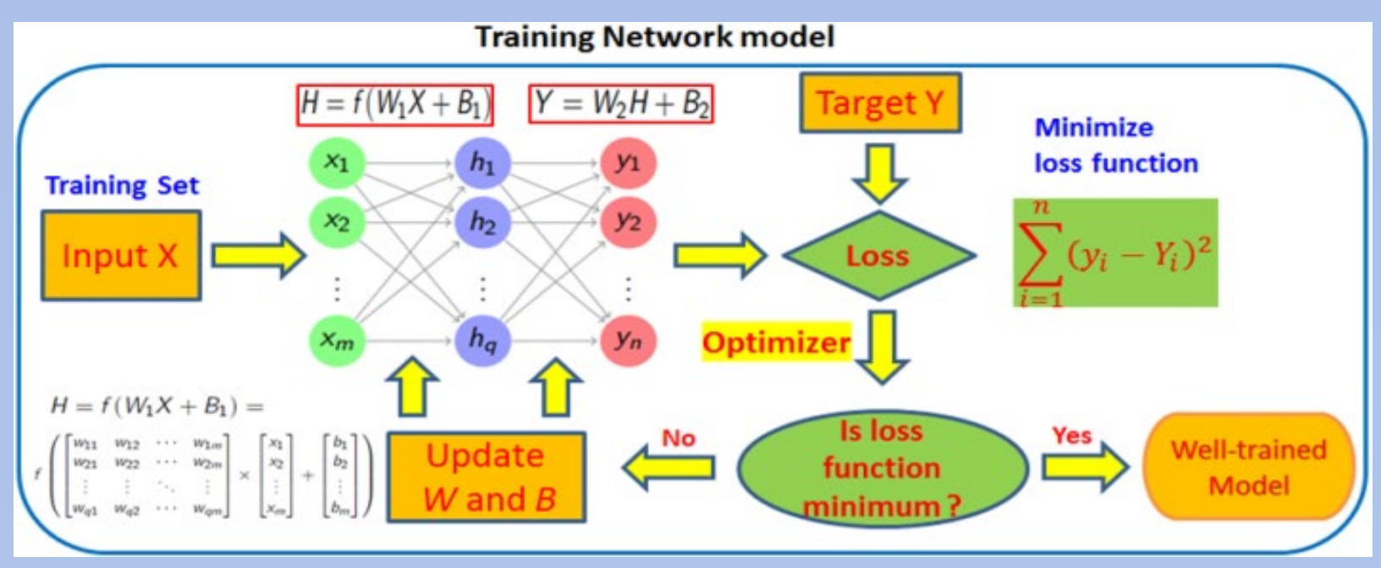
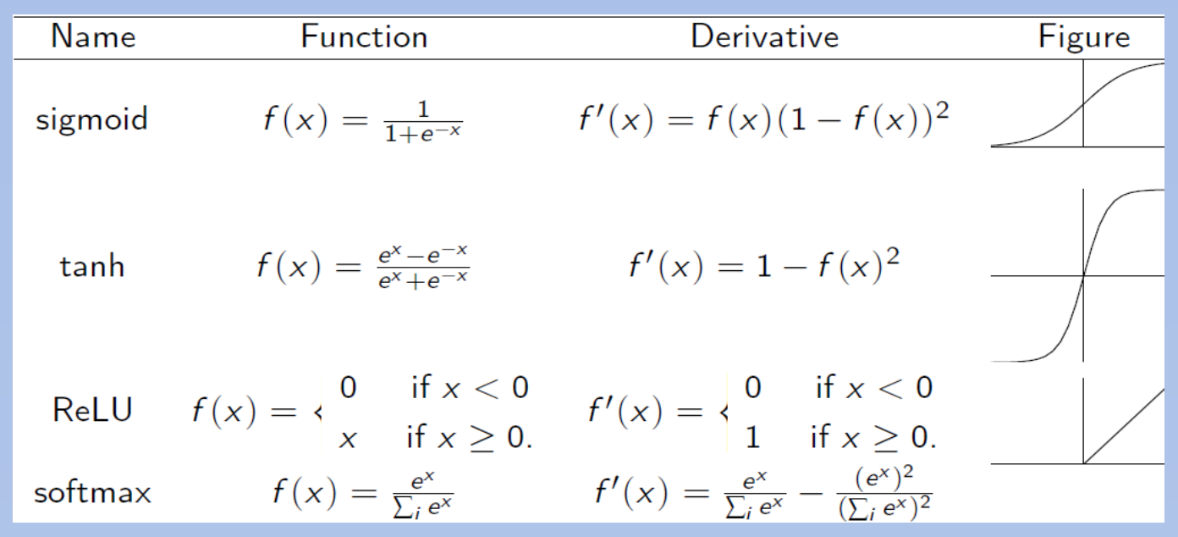


Figure 5: Flow chart for training neural network model.



Name	Function	Derivative	Figure
sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))^2$	
tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$	
softmax	$f(x) = \frac{e^x}{\sum_i e^x}$	$f'(x) = \frac{e^x}{\sum_i e^x} - \frac{(e^x)^2}{(\sum_i e^x)^2}$	

Figure 4: Commonly used activation function.

In the beginning of training process, the weight matrix elements are randomly assigned. During training process, the optimizer will update the weight matrix and bias values to minimize the loss function, which is defined as the square of the difference between the outputs of neural network and target values.

Simulation Detail

Accelerator Toolbox (AT) is used to generate training data. 3000 sets of 72 horizontal correctors strengths within +/- 2.5 μrad are randomly assigned with the matlab command 'rand'. Using a for-loop selects a set of 72 random number to assign the strengths of the 72 corrector magnets respectively, followed by using MML command getx to get the orbit. Eventually, we have 3000 different orbits associated with 3000 sets of different strengths of the 72 corrector magnets.

Python is used to develop machine learning application. Tensorflow [5] and keras [6], machine learning packages, are used to build the neural network model. Scikit-learn [7], data mining toolbox, is used to pre-process data, e.g. normalization and split data into training and validation sets of data. For TPS orbit correction in the horizontal plane, the number of input neurons is 172 BPMs, number of output neurons is 72, number of hidden neurons is 172. Figure 6 shows the loss function for training and validation sets of data. The loss function of the training and validation sets of data converge during Training process. That means no overfitting and underfitting phenomenon. After training, the trained model is saved by the keras package.

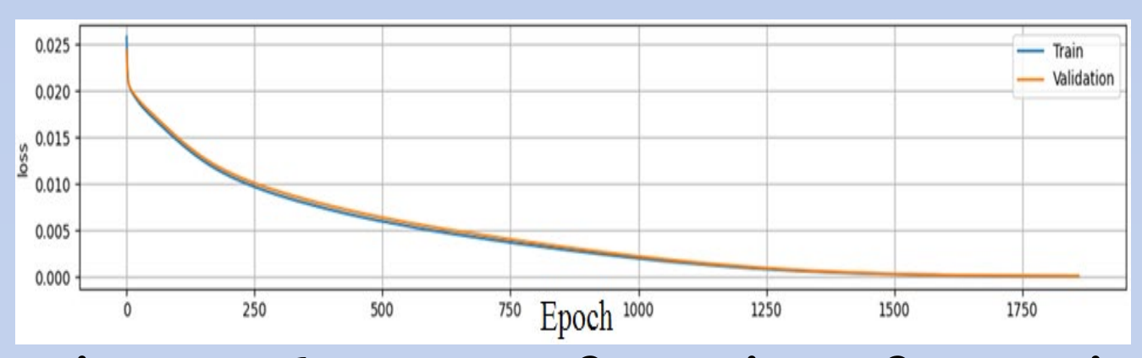


Figure 6: Loss function for training and validation sets of data

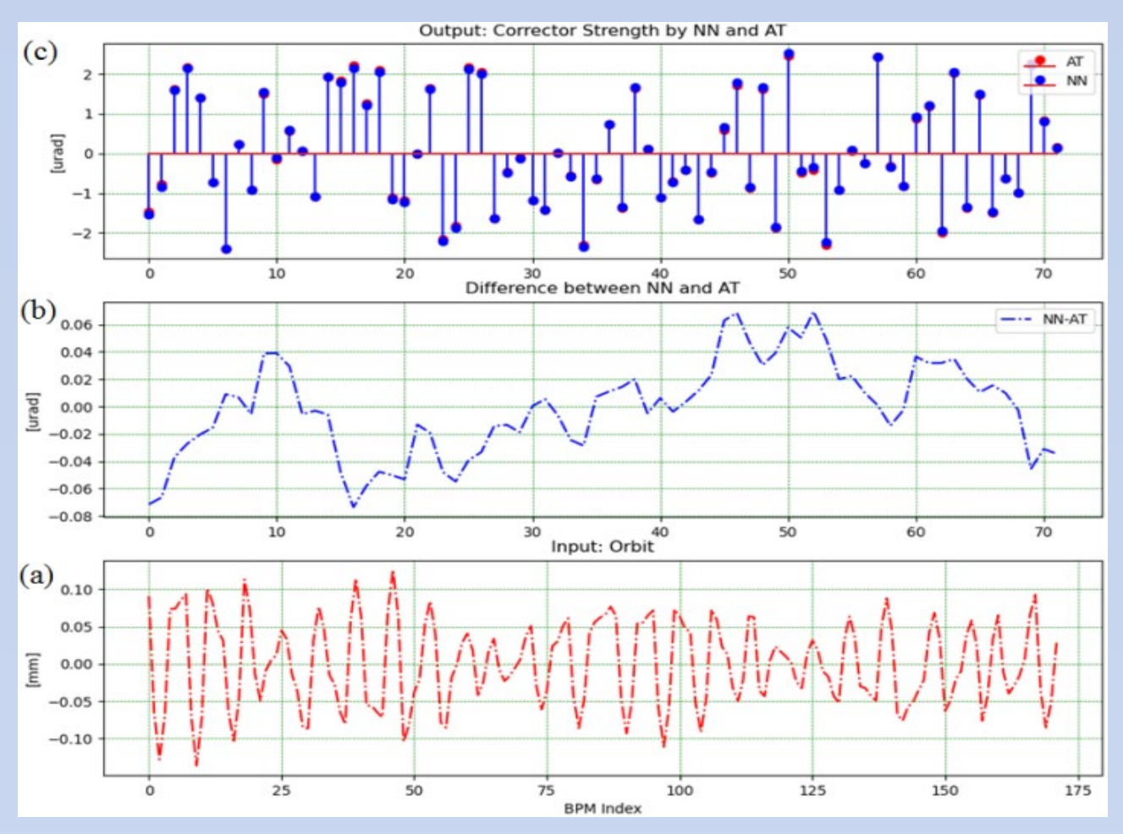


Figure 7: Accuracy of the trained neural network

AT simulator is used to verify the performance of the trained neural network on orbit correction. Figure 8 shows the implementation of the neural network on orbit correction. The unknown orbit distortion shown in figure 9 (a), generated by shifting 249 quadrupoles randomly within +/- 3 μm in horizontal plane with the AT command 'setshift', is feed into the trained neural network. The trained network will predict one set of 72 corrector strengths. Using the predicted corrector strengths to correct the orbit distortion in AT simulator and iterate three times. The corrected orbit by the trained neural network is shown in figure 9 (b).

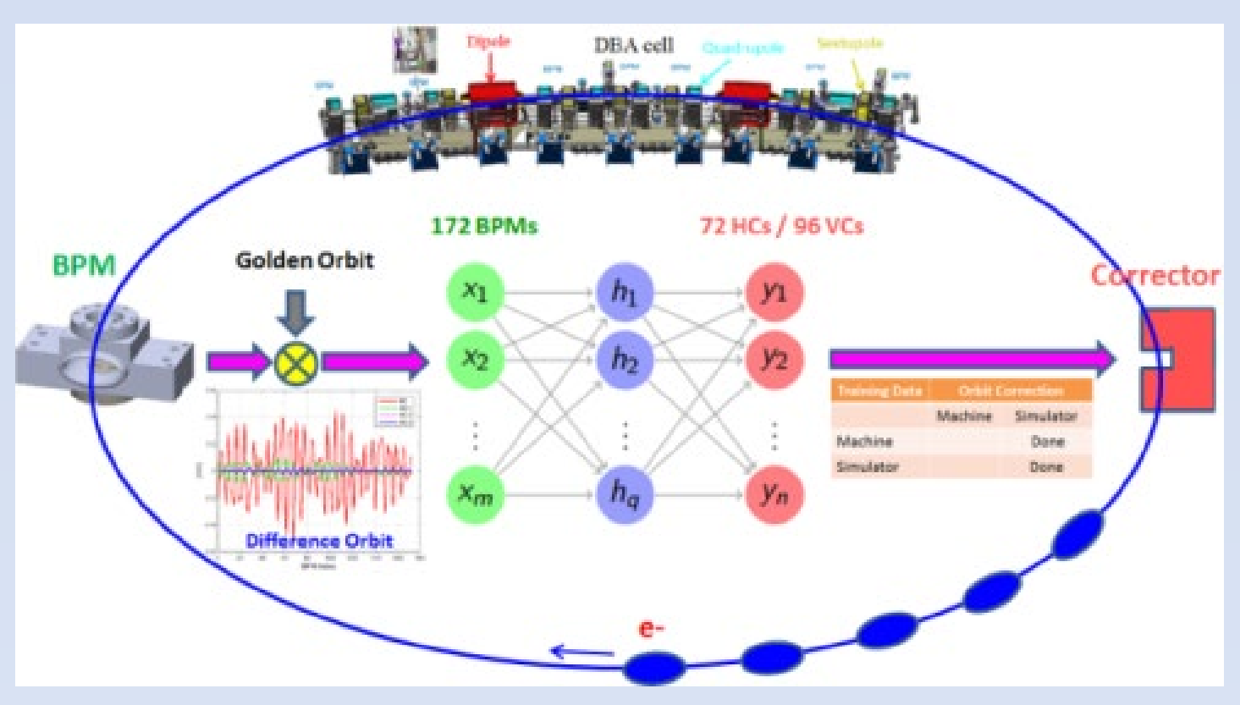


Figure 8: Implement neural network for orbit correction

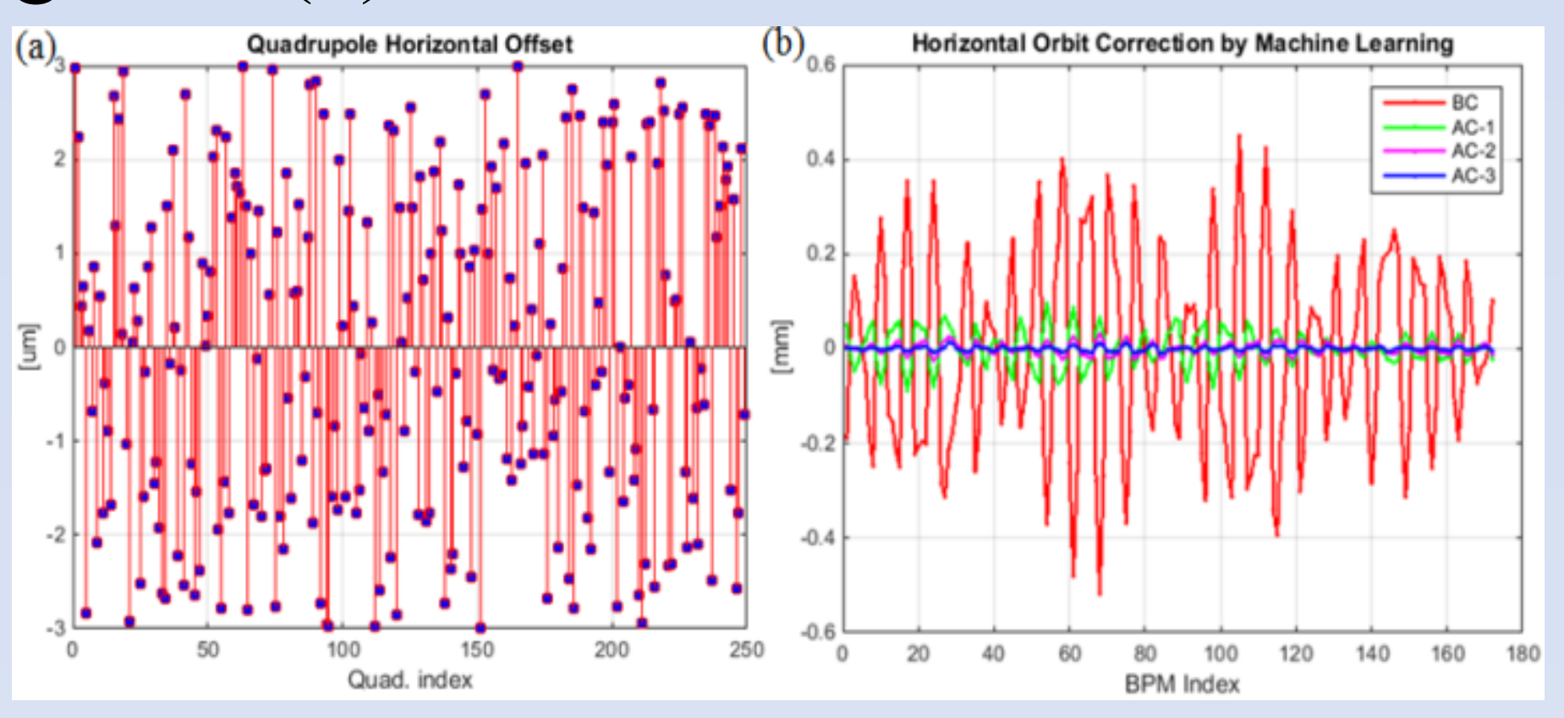


Figure 9: (a) Misalignment quantities of 249 quadrupole magnets (b) Orbit correction by neural network

Reference

[1] C.C. Kuo *et al.*, "Commissioning of the Taiwan Photon Source", in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, TUXC3, pp. 1314-1318..

[2] M.S. Chiu *et al.*, "Double mini-betay lattice of TPS storage ring", in *Proc. PAC'11*, San Sebastián, Spain, May 2011, WEPC035, pp. 2082-2084.

[3] D. Schirmer, "Orbit correction with machine learning techniques at the synchrotron light source DELTA", in Proc. ICALEPCS2019, paper WEPHA138, pp.1426-1430

[4] D. Schirmer, "A machine learning approach to electron orbit control at the 1.5 GeV synchrotron light source DELTA", in Proc. IPAC'22, Bangkok, Thailand, May 2022, paper TUPOPT058, pp. 1137-1140

[5] https://www.tensorflow.org/

[6] https://keras.io/

[7] https://scikit-learn.org/stable/