



Study of Orbit Correction by Neural Networks in Taiwan Photon Source

M.S. Chiu[†], Y.S. Cheng, 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 in Taiwan photon source. The training data for neural networks is generated by accelerator toolbox (AT).

TPS Lattice Function:

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).

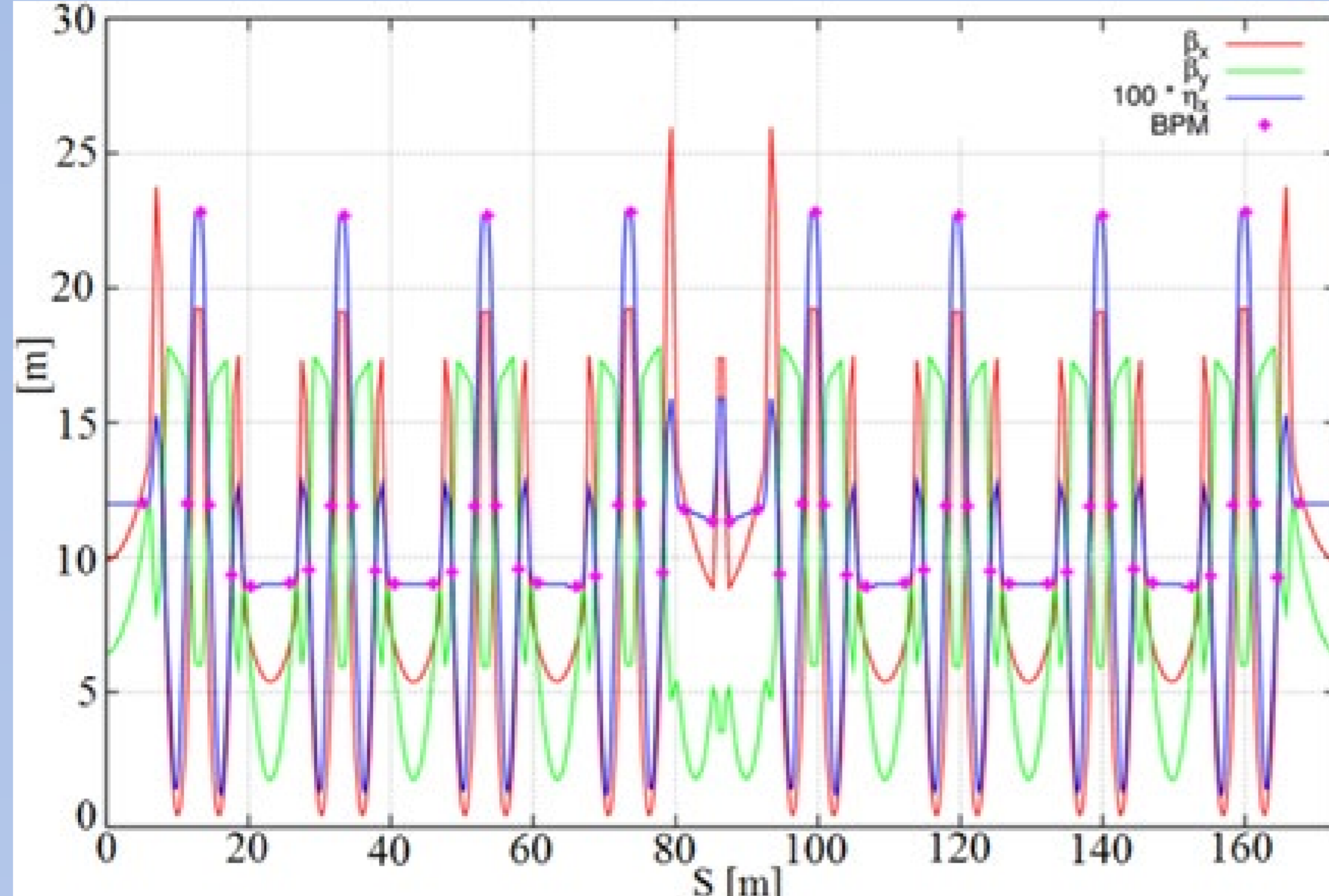


Figure 1 : Twiss functions for the double mini- β_y lattice in the 1/3 TPS storage ring.

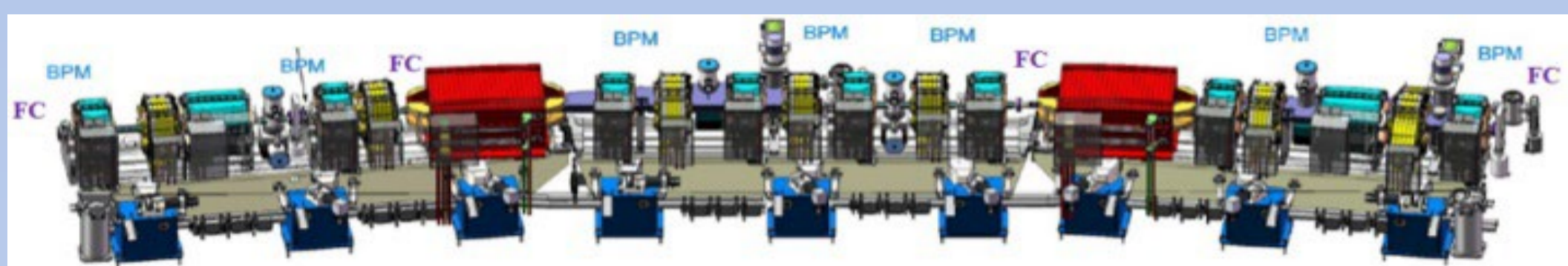


Figure 2: DBA cell.

BPM: 7
HC: 3
VC: 4

Neural Networks

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.

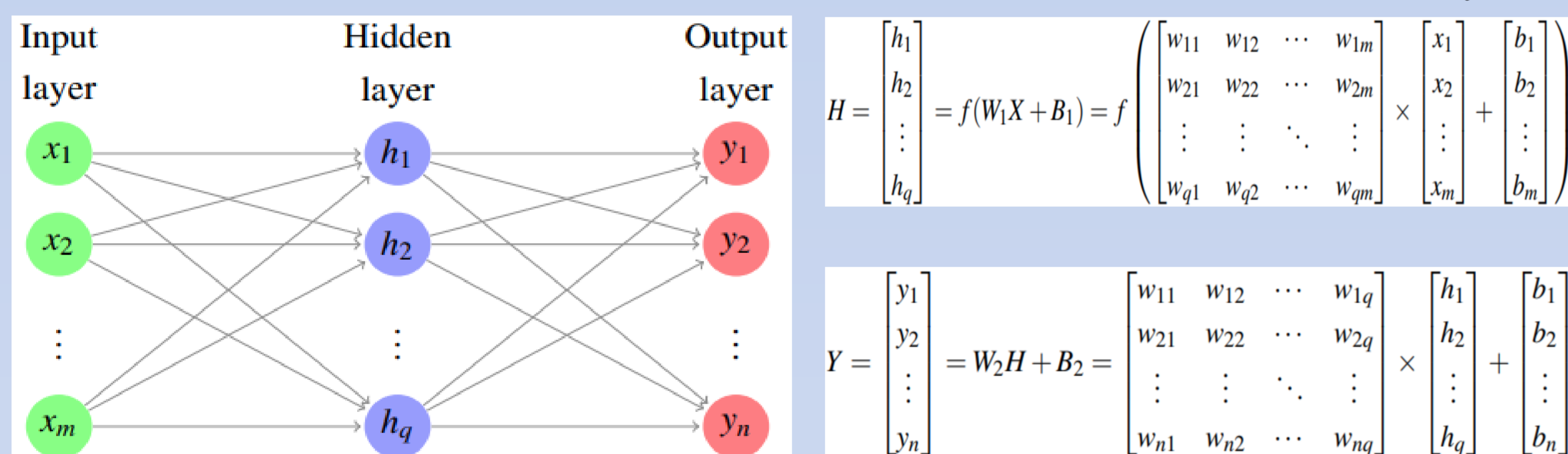


Figure 3: Feed-forward neural network

, 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.

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	$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	$f'(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} - \frac{e^{x_i} e^{x_j}}{(\sum_j e^{x_j})^2}$	

Figure 4: Commonly used activation functions.

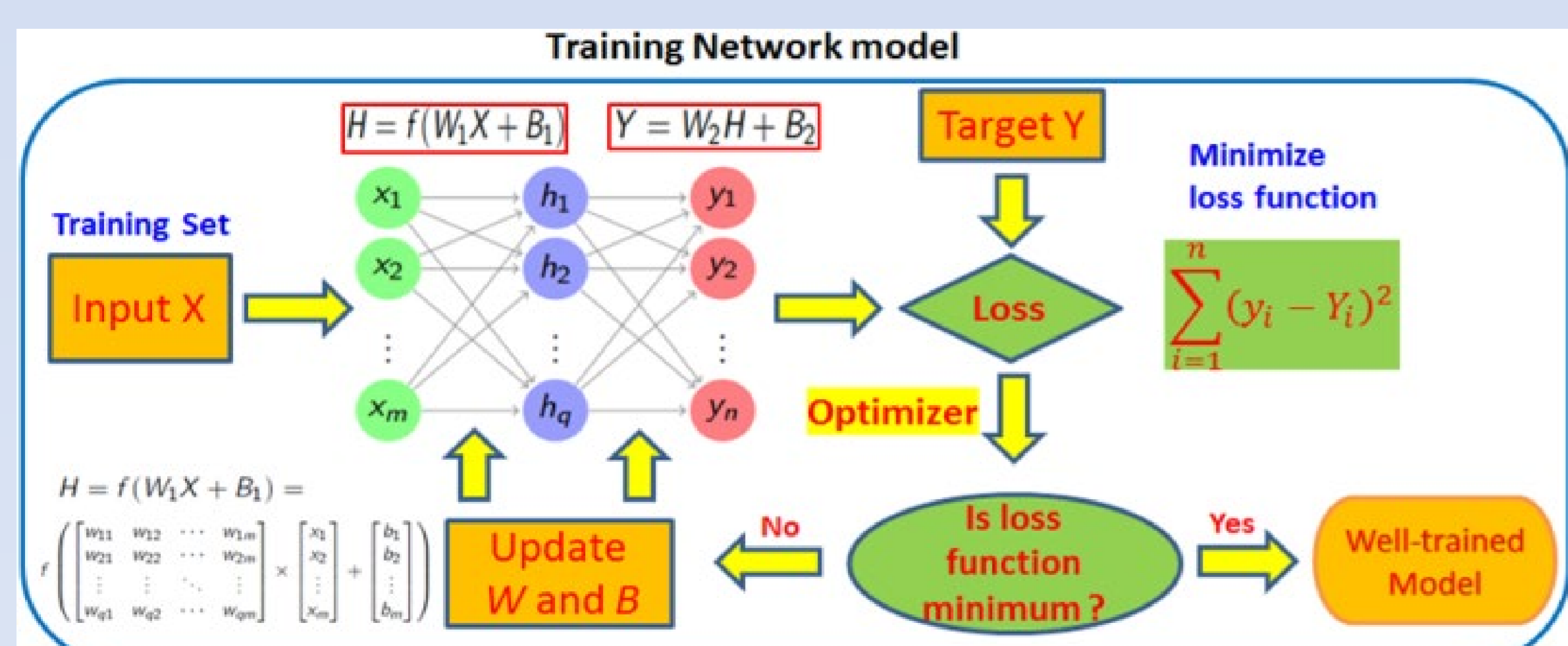


Figure 5: Flow chart for training neural network model.

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-beta lattice of TPS storage ring", in *Proc. PAC'11*, San Sebastián, Spain, May 2011, WEPC035, pp. 2082-2084.

Simulation Detail

Accelerator Toolbox (AT) is used to generate training data. 3000 sets of 72 horizontal correctors strengths within $\pm 2.5 \mu\text{rad}$ 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.

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.

The number of input neurons is 172 BPMs, number of output neurons is 72 correctors in horizontal plane, 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. After training, the trained model is saved by the keras package.

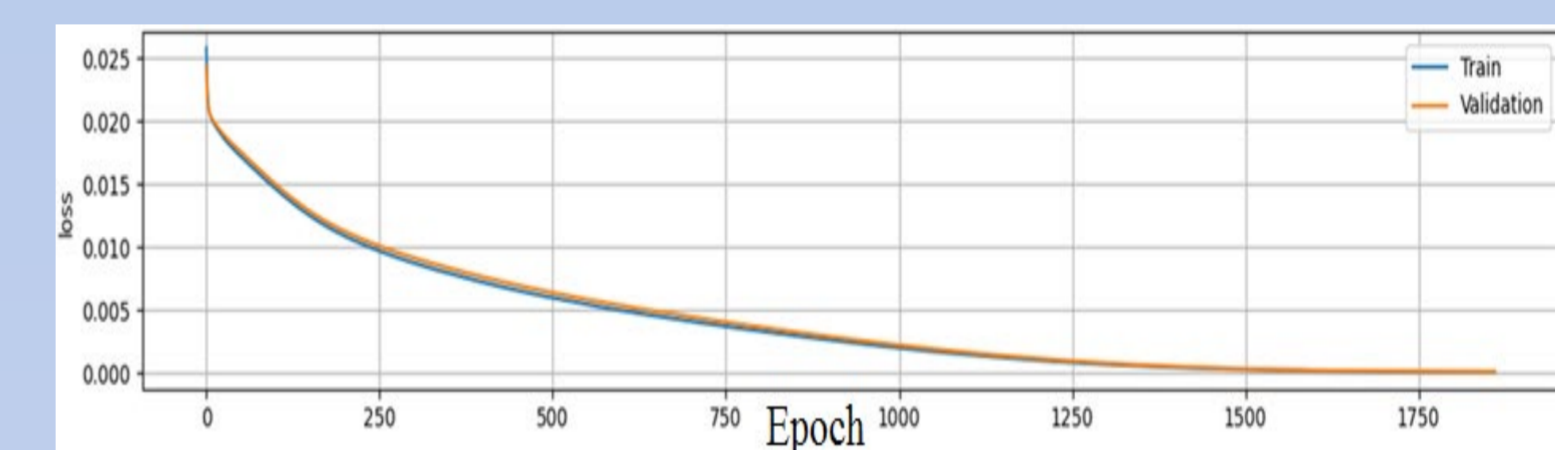


Figure 6: Loss function for training and validation sets of data

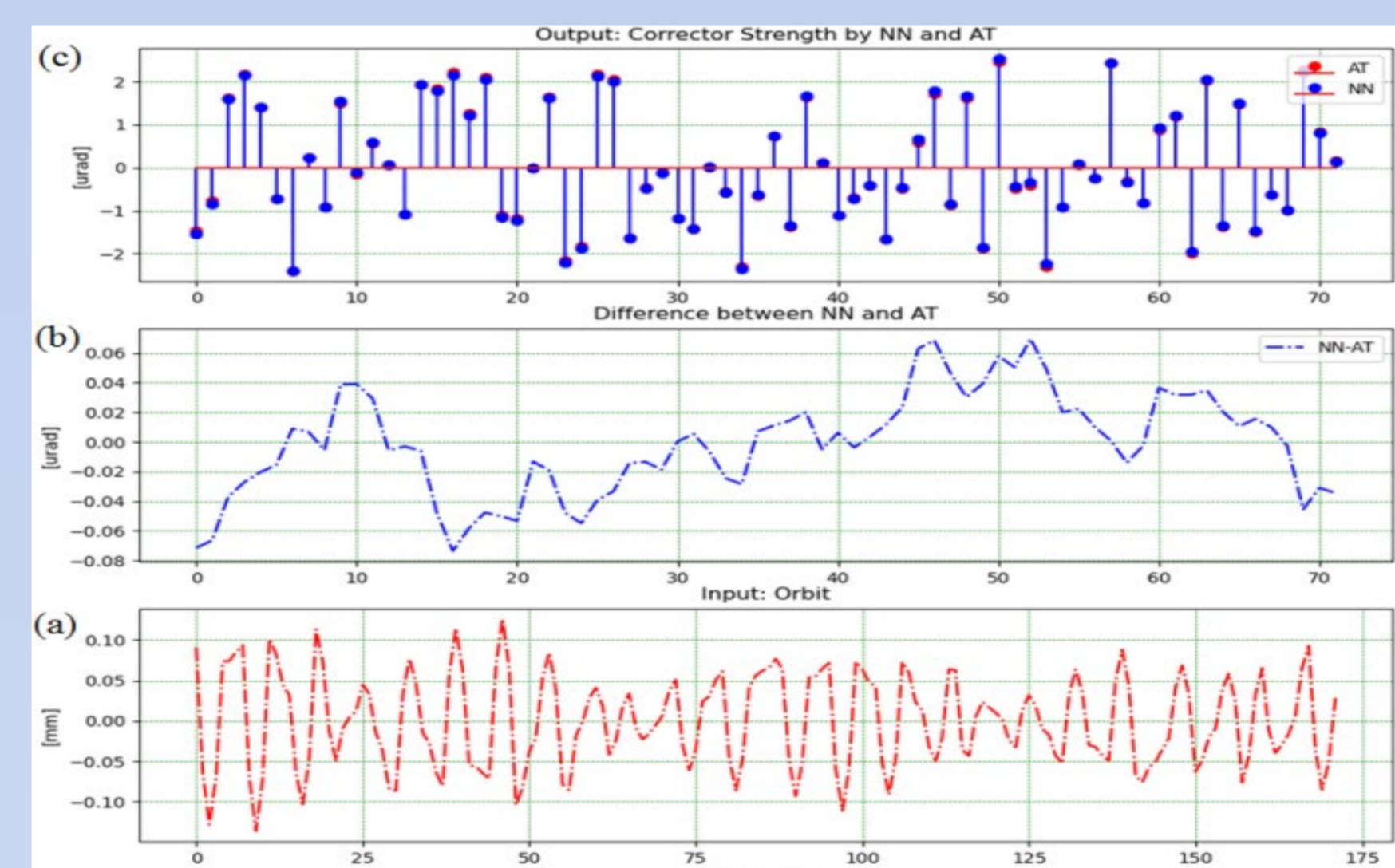


Figure 7: Accuracy of the trained neural network

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 $\pm 3 \mu\text{m}$ 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).

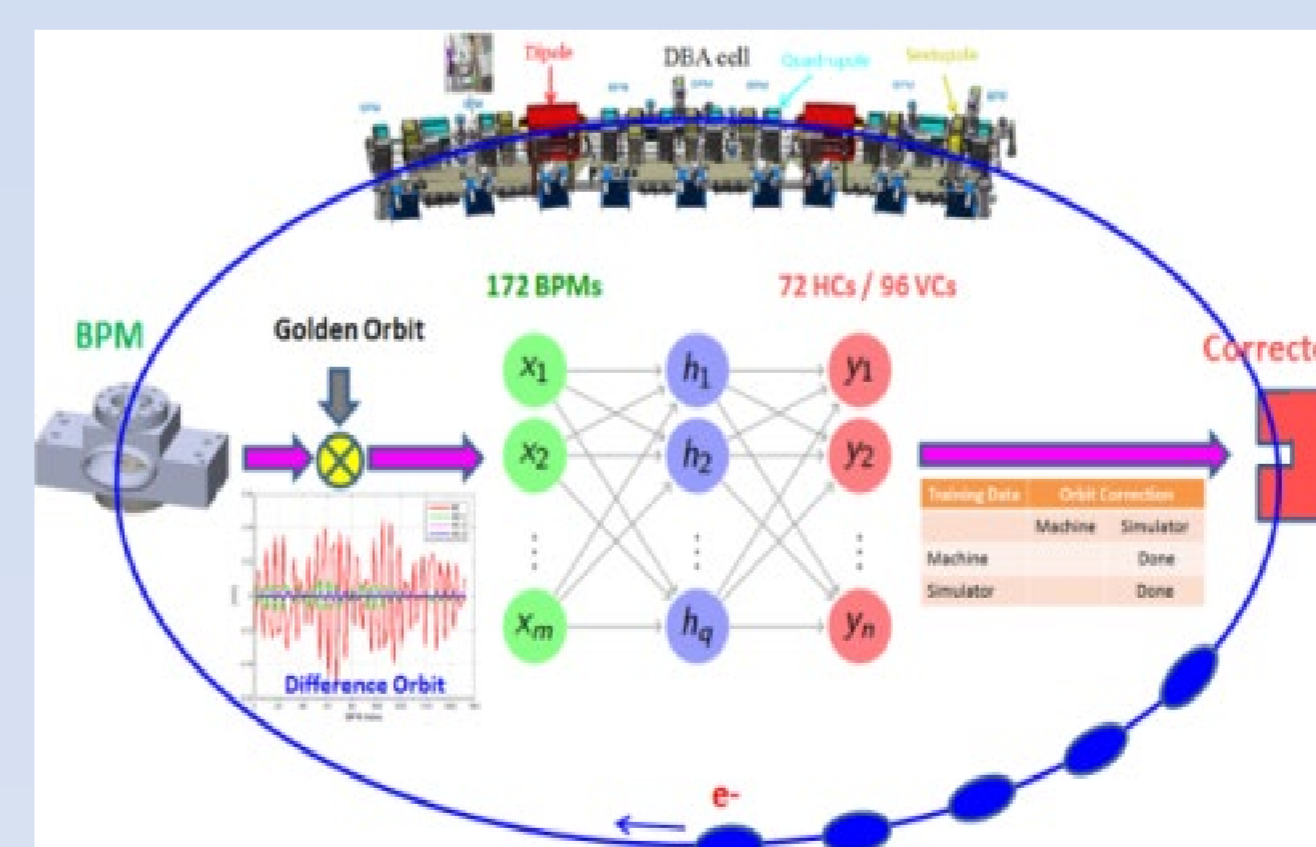


Figure 8: Implement neural network for orbit correction

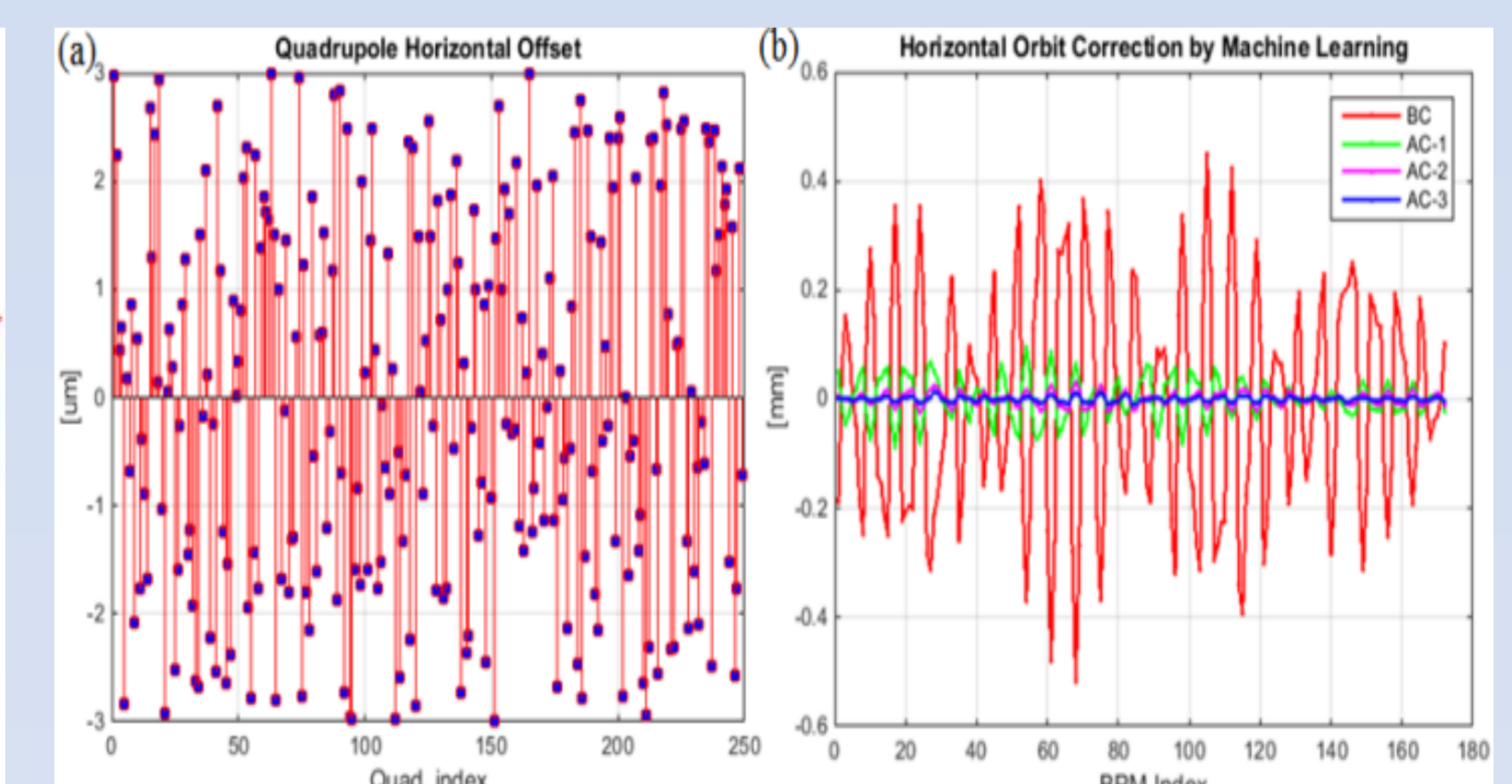


Figure 9: (a) Misalignment quantities of 249 quadrupole magnets (b) Orbit correction by neural network

[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/>