Optimizing ANN Training Performance for Chaotic Time Series Prediction Using Small Data Size

Lei Zhang

Abstract—**In this paper, the training performance of artificial neural network (ANN) is investigated based on three aspects of the training data: the data size, the distribution of the subsets for training, validation and testing, and the data segments of the training data, which are generated using Lorenz chaotic system equations and the forward Euler method to represent the chaotic features commonly found in real world applications, specifically for the Electroencephalogram (EEG) signals captured from brain activities. This research investigates the potentiality and feasibility of using small data set for training ANN to generate chaotic time series, which can be used for the simulation and analysis of chaotic features in EEG signals. Contradictory to the popular belief that better performance can be achieved by larger number of training samples, the training results show that the same level of training performance can be achieved by a relative small number of training samples for the generation and prediction of the chaotic system time series.**

Index Terms—**Electroencephalogram (EEG), time series, artificial neural network (ANN), small data, chaotic patterns.**

I. Introduction

Following the extraordinary accuracy achieved in image classification by using big data [1], enormous amount of effort has been dedicated to data acquisition in machine learning applications of all sorts, especially for pattern recognition and classification. The triumph of big data has been well celebrated by employing this training method to improve training performance and increase classification accuracy, which however comes at a high cost of excessive training time and power consumption. The complexity of convolutional neural network (CNN) required for abstracting multi-dimensional features from big training data in the deep learning process [2] inevitably results in the increasing of both training time and power consumption. For most machine learning applications in brain research using EEG signals, big data may be a luxury neither essential nor practical to be had.

EEG biomarkers have been developed to identify various mental disorders. Previous research has shown that machine learning techniques can be successfully applied to EEG signals to identify mental disorders such as schizophrenia [3], bipolar disorder [4] and depression [5], as well as to improve classification accuracy of different brain activities in brain-computer interface (BCI) [6]. EEG has better temporal resolution over other neuroimaging methods and can be used to capture fast changes of brain activities. However, EEG signals have some intrinsic limitations such as low signal to noise ratio due to noise and artifacts, as well as relatively low spatial resolution in that the electrodes can only detect collective regional brain activity. Besides, EEG time series patterns are highly individually dependent and can change over time for each individual subject. It is impractical and unfeasible to acquire big EEG data for each individual over a long period of time. In contrary, it can be extremely beneficial to use small data set that is sufficient to meet the requirement of machine learning and serve the purpose of classification and prediction of EEG time series signals.

As Artificial Neural Network (ANN) is originally inspired by biological neural network, from an evolutionary point of view, the ultimate goal should be to preserve energy in problem solving, instead of exhausting all energy in order to achieve perfect accuracy. For examples, it takes human brain little effort to recognize a face with a few key features, which is highly efficient without having to recall all specific details with 100% accuracy, unless one obtains the skills of a artist to attend to details. It therefore inspired the initial idea in this research to investigate the ANN training performance using small data set.

Section II explains background of the research project for the design of ANN-based chaotic system generators to simulate and predict EEG signals in brain research; Section III describe the generation of training samples; Section IV provides the ANN training results using different data sizes; Section V concludes the research work and offers some future research perspectives.

II. A Brief Background of the Brain Research

This research is a subproject of a multidisciplinary brain research program aiming to understand the brain functionality and enhance cognitive ability via brain stimulation. The design of ANN-based chaotic system models is motivated by the lack of theoretical foundation and practical individual data in brain research. EEG signals captured from brain demonstrate chaotic features such as bifurcation. A hypothesis is drawn from the explanations presented in [7] that a stable cortical state of the brain neural network may be manifested by an aperiodic signal of a chaotic attractor in the captured EEG data. The previous related research work published recently [8]-[10] were focus on the optimization of ANN architecture in order to efficiently generate and predict chaotic time series outputs which can be used for the simulation and analysis of EEGs signals for brain research. This research examines the possibility of using small training data set for ANN training without jeopardizing the training performance of the ANN-based chaotic system. This is of great importance for the study of brain activities with limited EEG signals for

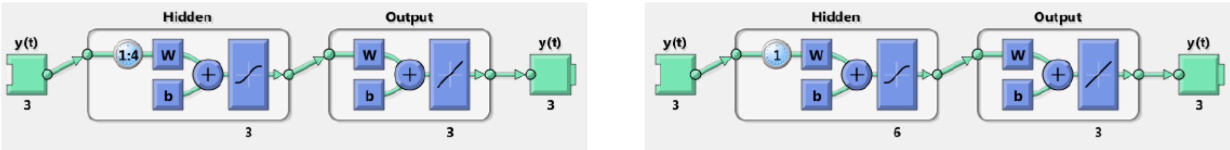
Manuscript received August 23, 2018; revised October 19, 2018.

L. Zhang is with the Faculty of Engineering and Applied Science, University of Regina, Regina, SK, Canada, S4S 6T2 (e-mail: lei.zhang@uregina.ca).

doi: 10.18178/ijmlc.2018.8.6.753

individuals.

The design and optimization of ANN architecture has been discussed in previous related research [8]. An ANN architecture with one hidden layer for Nonlinear Autoregressive (NAR) model can be defined by the number of hidden neurons (n) and the number of input delays (d) [10]. In this paper, only two ANN architectures ANN(n=3, d=4) and ANN(n=6, d=1) are selected for comparing training results using different training data sets. It is shown that when big training data set is used, these two architecture can achieve good training performance at low computational cost.



(a) ANN(n=3, d=4) (b) ANN(n=6, d=1)

Fig. 1. ANN architectures used for training.

The MATLAB Neural Network Toolbox and the nonlinear autoregressive (NAR) model in the nonlinear time series function *ntstool* are used for ANN training. The training performance is measured by the average mean square errors (MSEs) between the ANN outputs and the target outputs. Due to the random initial values assigned to all weights and bias in the ANN by the MATLAB *ntstool*, the MSE value varies for each training iteration, even with the same ANN architecture, training data set and training parameters. Therefore each training is carried out for 3 iterations. The smallest MSE is used for evaluating training performance among various training data in order to eliminate the abnormal big MSE value caused by a local minimum gradient [11]. It is reported in previous related research that the Levenberg-Marquardt (*trainlm*) training algorithm [12] has the best training efficiency in terms of well balanced training performance and training time, compared to the other training algorithms available with the *ntstool*. The default MATLAB training parameters are used for all training.

III. Generation of the Training Samples

The Lorenz system differential equations used for generating the training data are listed in (1).

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = \rho x - y - xz \qquad (1)$$
$$\frac{dz}{dt} = -\beta z + xy$$

The system has three outputs x, y and z, which are the target outputs of the ANN training, hence there are three neurons in both the input and output layers of the NAR ANN model. σ, ρ and β are three system parameters. Chaotic systems can be in chaotic, periodic or stable states depending on the setup of initial values and system parameters. In this study, all training data sets are generated with the same initial values $x_0 = y_0 = z_0 = 10$, and system parameters σ=10, ρ=28, β=8/3, which will set the Lorenz attractor in chaotic state with chaotic time series outputs.

Up to 16,000 training samples of Lorenz chaotic time series are generated using the three Lorenz system differential equations and the forward Euler method at two step sizes dt=0.01 and dt=0.005 [10]. The length of the time series segment depends on both the number of training samples and the step size. For example, 1,000 (1k) samples generated with the same initial conditions with dt=0.01 is considered to represent the equivalent length of time series segment as 2,000 (2k) samples with dt=0.005. The two time series segments with the same length are deemed to be equivalent but are in fact very different in nature due to the divergence of the two chaotic time series caused by the different step sizes. The training samples with smaller step size have better precision.

Different training data sets are formed by using different blocks of the generated data samples. Two different ANN architectures are selected to investigate the training performances based on the selection of training data. Training data set with limited number of samples can only represent small segments of the Lorenz attractor, which is intended to examine the training performance and determine if these partial representations are sufficient for the ANN training to generate the target outputs of the chaotic system. The training for each ANN architecture and training data set is carried out for 3 iterations, and the training time is measured in seconds.

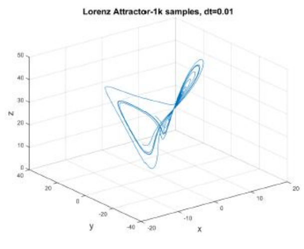
The generated training samples are divided into 3 subsets for training, validation and testing. The training subset is used in the training process for computing the gradient and updating the weights and biases to gradually reduce the MSE. The validation set is used to measure network generalization and stop the training if the ANN is overfitting on the training data. The network weights and biases are saved at the minimum validation MSE. The testing subset is used independently to compare the results of different network models. It can also detect poor division of training samples among the three data subsets when the minimum MSEs of the testing data and validation data occur at a significantly different training epoch. There are two optional functions for dividing the training samples into three subsets. The default 'dividerand' function divides the training data randomly. The 'divideblock' function divides the training data into three contiguous blocks, which can be used alternatively to evaluate the predictability of the ANN. The default divide ratios for the training, validation and test subsets are 70%, 15% and 15%.

IV. ANN Training Results

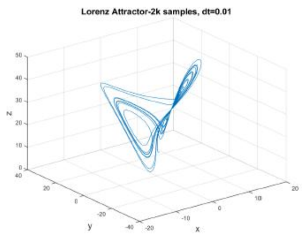
The ANN training is carried out for two selected ANN architectures: ANN(n=3, d=4) and ANN(n=6, d=1), as shown in Fig.1(a) and (b) respectively. The computational cost of ANN architecture is measured by the number of multiplications between neurons and weights. The ANN (n=3, d=4) has three neurons in the input, hidden and output layers. The actual number of neurons in the input layer after adding the delay is 12 (4×3). Therefore the computational cost is 45 (3×4×3+3×3). The ANN(n=6, d=1) has 3 neurons in both input and output layers and 6 neurons in the hidden layer, and a computation cost of 36 (3×6+6×3).

A. Training Results for ANN(n=3, d=4)

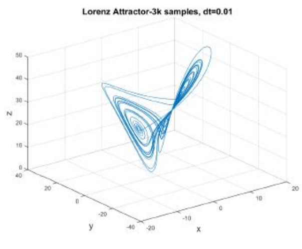
The training for the ANN(n=3, d=4) architecture is carried out using 12 training data sets with the number of samples incrementing by 1k samples from 1k up to 12k samples. The Lorenz system time series segments represented by the data sets of 1k to 8k with step size dt=0.01 are plotted in Fig. 2.



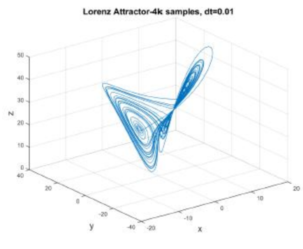
(a) 1k, dt=0.01



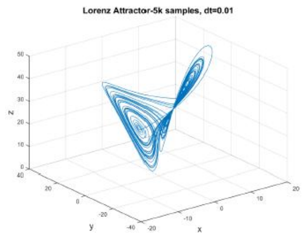
(b) 2k, dt=0.01



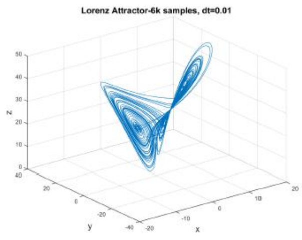
(c) 3k, dt=0.01



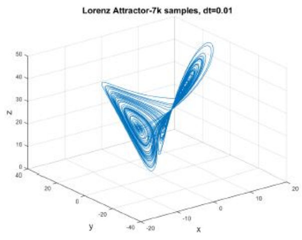
(d) 4k, dt=0.01



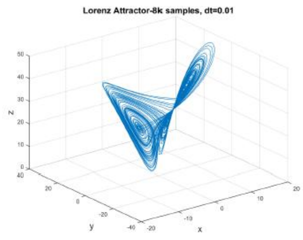
(e) 5k, dt=0.01



(f) 6k, dt=0.01

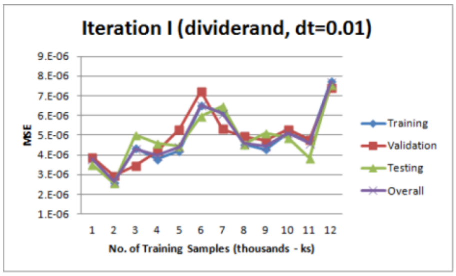


(g) 7k, dt=0.01

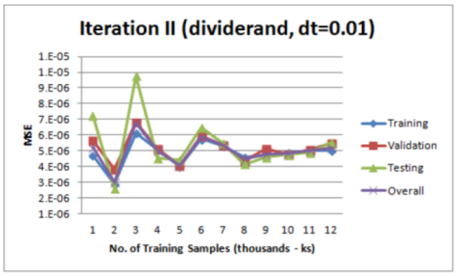


(h) 8k, dt=0.01

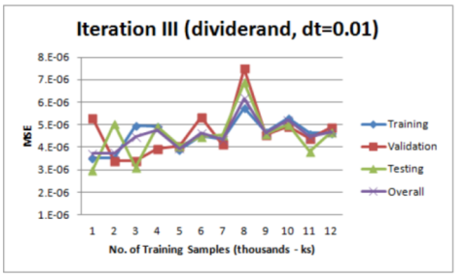
Fig. 2. Lorenz system time series segments of 1k to 8k training samples (dt=0.01).



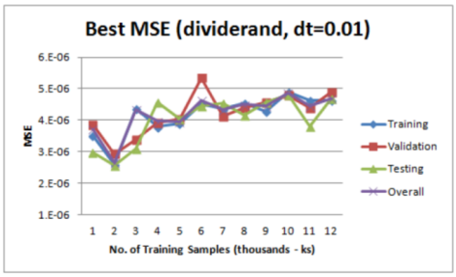
(a) Iteration I



(b) Iteration II



(c) Iteration III



(d) Best Performances of Three Iterations

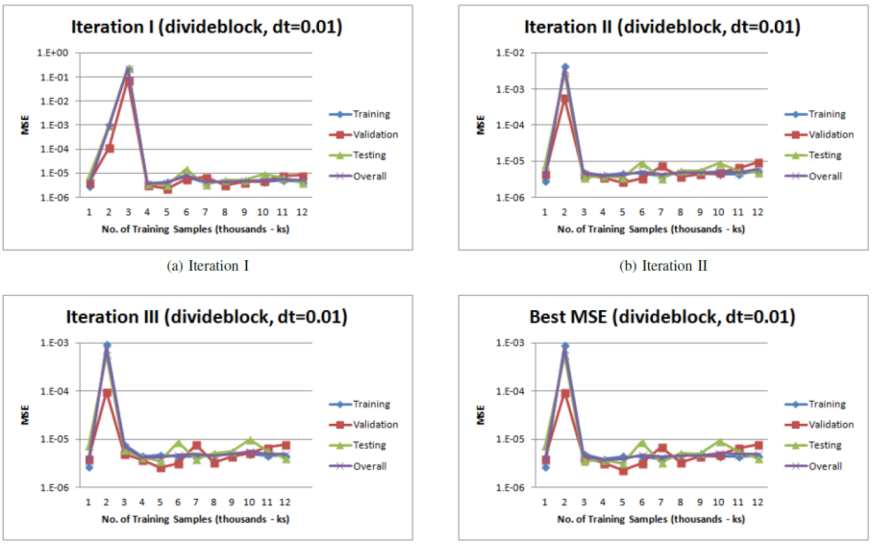
Fig. 3. Training results for ANN(n=3, d=4) using 1k to 12k training samples (dt=0.01, dividerand).

1) Training data (dt=0.01, 'dividerand')

First, the 'dividerand' option is used for randomly divide the training samples of each data set into three subsets (70% for training, 15% for validation and 15% for testing). The overall performance and the performances of three data subsets are plotted for each iteration in Fig. 3(a), (b) and (c) respectively. The best overall and subset training performances of three iterations are plotted in Fig. 3(d). As indicated by all three iterations, the training performance cannot be improved by increasing the number of training samples. Contradictory to the common belief that big data would inevitably lead to better training results, in this case of ANN training for chaotic system generation, the best performances degrades slightly yet consistently as the number of training samples increment from 1k to 12k.

2) Training data (dt=0.01, 'divideblock')

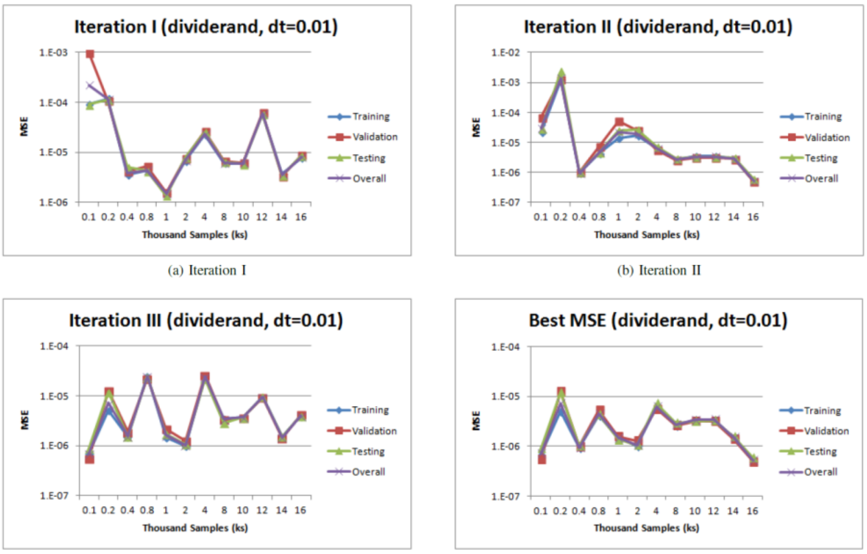
Second, the 'divideblock' option is used for dividing the training samples into three contiguous blocks. The training results are plotted in Fig. 4. The performance of validation subset can be used to indicate the predictability of the trained ANN. Similarly to the training results using the 'dividerand' option, the overall training performance is not improved by increasing the number of training samples, i.e., the training data size. Differently from the 'dividerand' results, an interesting degradation occurs unanimously for all three training iterations using 2k training samples, whereat the validation performance is better than the training performance. This indicates poor distribution of training samples in three subsets. In other words, the validation subset contains information that is not presented in the training subset. It can be observed from the 3D time series plots in Fig. 2 that the changing rates of the three dimensions can be different for the training and validation subsets if the validation subset happens to fall into the two focus points of the Lorenz attractor. The Lorenz time series segment of 2k training samples in Fig. 4(b) also indicates that the segment can only partially represent the Lorenz attractor; and it contains both fast and slow changing subsegments which could result in poor training performance caused by dividing the training samples using the 'divideblock' option.



(a) Iteration I (b) Iteration II

(c) Iteration III (d) Best Performances of Three Iterations

Fig. 4. Training results for ANN(n=3, d=4) using 1k to 12k training samples (dt=0.01, divideblock).



(a) Iteration I (b) Iteration II

(c) Iteration III (d) Best Performances of Three Iterations

Fig. 5. Training results for ANN(n=6, d=1) using 12 training data sets (dt=0.01, dividerand).

B. Training Results for ANN(n=6, d=1)

A different training approach is taken for the ANN(n=6, d=1). Both `dividerand' and 'divideblock' options are used to divide the training samples. The number of samples for 12 different data sets are: 100, 200, 400, 800, 1k, 2k, 4k, 8k, 10k, 12k, 14k and 16k. These data sizes are selected as such to compare the training results of time series segments with equivalent length but generated using different step sizes (dt=0.01 vs. dt=0.005).

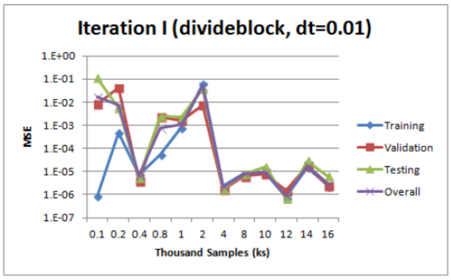
1) Training data (dt=0.01, 'dividerand')

The training results using training samples generated with step size dt=0.01 and divided with the 'dividerand' function for 12 different data sizes are plotted in Fig. 5. As the training samples are randomly divided, there is normally no significant difference among the MSEs of three training data subsets. The best MSEs zigzag between 1E-5 and 1E-6 as the number of training samples increases from 1k to 16k, without demonstrating any consistent improvement in performance.

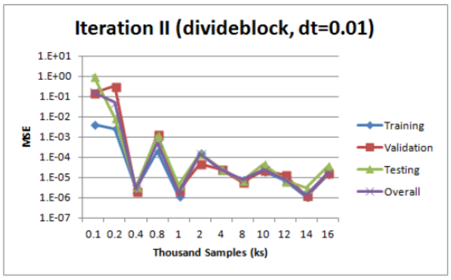
2) Training data (dt=0.01, 'divideblock')

The training results using the same data sets divided with the 'divideblock' function are plotted in Fig. 6. It is shown that when the data size is smaller than 1k, the validation MSE is much bigger than the training MSE. This indicates that the training performance is overfitting on the training data and the small number of training samples is not enough for training the ANN to learn the chaotic features of the Lorenz attractor. The differences of MSE for training, validation and testing subsets are relatively small when data sizes bigger than 1k (1k included) are used. The best MSEs zigzag between 1E-5 and 1E-6 as the number of training samples increases from 4k to 16k. As the training samples are divided into three contiguous blocks, the variation between the training and validation MSEs can be used to measure the prediction capability of the trained ANN. In the case of chaotic system, the size of validation and testing blocks should be considerably smaller than the training block, as they may represent time series segments of different changing rate compared to the training block.

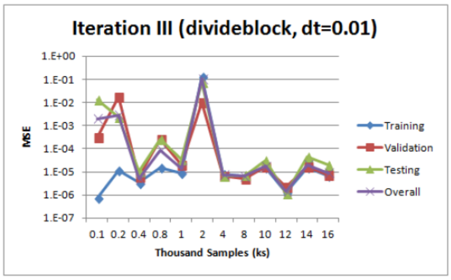
All MSEs are calculated for open-loop NAR model, where the MSE is calculated using the ANN outputs at each time step sequentially compared to the target outputs and the average MSE is calculated for all training samples. It is however infeasible to directly measure the MSE of close loop NAR model, as it is intrinsic for chaotic time series to demonstrate chaotic behavior and an infinitesimal difference can cause the outputs of the same chaotic system to diverge significantly from the original path after a short period of time. Nevertheless, the close-loop NAR is used to generate the prediction based on the defined system initial values and can be inspected visually.



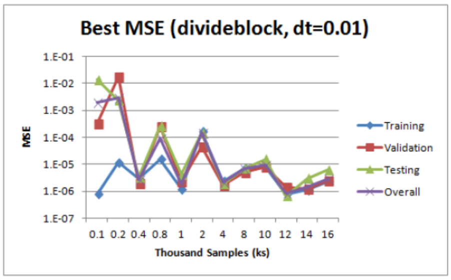
(a) Iteration I



(b) Iteration II

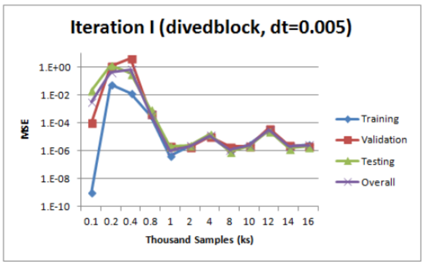


(c) Iteration III

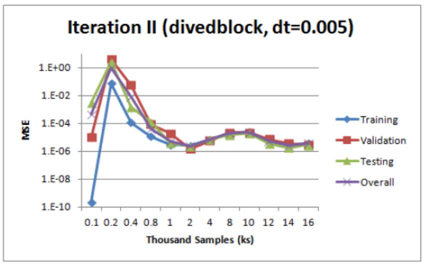


(d) Best Performances of Three Iterations

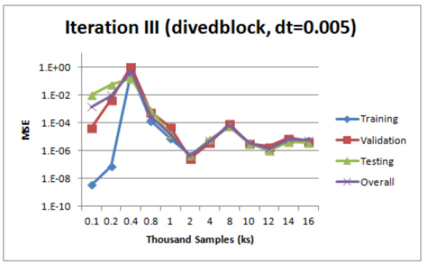
Fig. 6. Training results for ANN(n=6, d=1) using 12 training data sets (dt=0.01, divideblock).



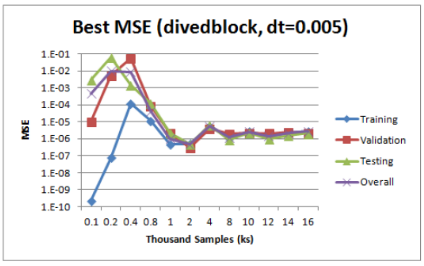
(a) Iteration I



(b) Iteration II



(c) Iteration III



(d) Best Performances of Three Iterations

Fig. 7. Training Results for ANN(n=6, d=1) using 12 training data sets (dt=0.005, divideblock).

3) *Training data (dt=0.005, 'divideblock')*

In order to investigate whether the ANN training performance and predictability can be improved by increasing the data precision, that is, to use more training samples to represent the same length of time series segments. The training samples are generated with step size dt=0.005 and the 'divideblock' option is used for dividing the training samples into three contiguous blocks. The training results are plotted in Fig. 7. It can be observed that as the number of training samples increases above 1k, the MSEs variation is more converging and less zigzag compared to the previous training with *dt*=0.01. This may be due to the doubled precision and halved training segment, a.k.a, doubled data precision.

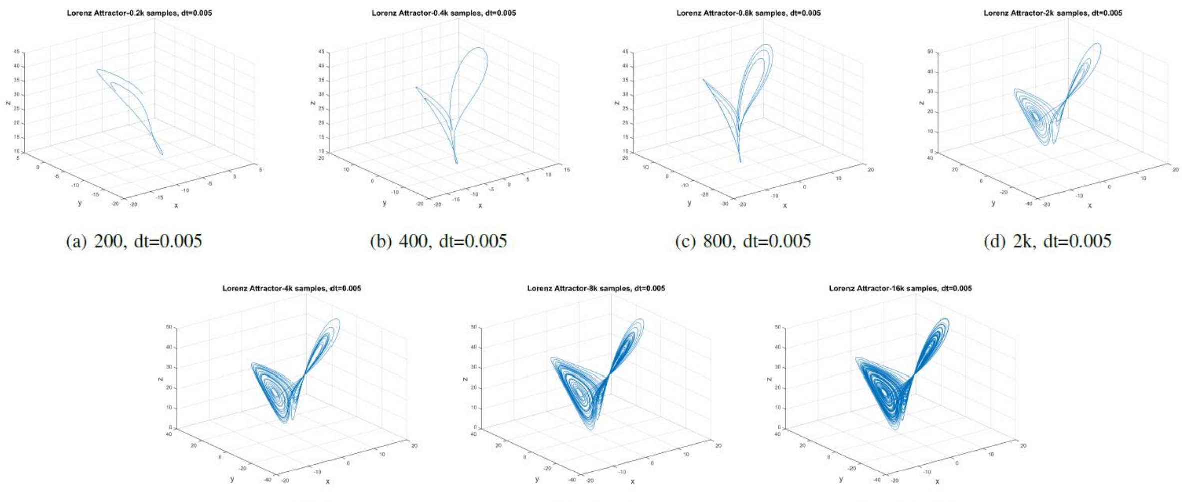
A number of data sizes of training data are carefully selected for comparing the equivalent time series segments generated with two step sizes (*dt*=0.01 and *dt*=0.005) and divided by the 'divideblock' function, as listed in Table I.

The Lorenz system time series equivalent segments represented by the training samples generated with step size *dt*=0.005 are plotted in Fig. 8.

TABLE I: TRAINING DATA WITH EQUIVALENT LENGTH

Step Size	**Number of Training Samples (ks)**						
dt=0.01	0.1	0.2	0.4	1	2	4	8
dt=0.005	0.2	0.4	0.8	2	4	8	16
Equivalent Length	1	2	4	10	20	40	80

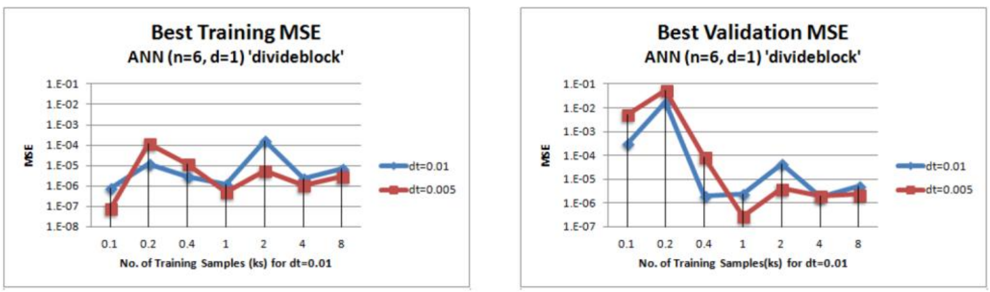
The best training and validation MSEs of two time series with equivalent length generated with two different step sizes are plotted in Fig. 9. These time series segments are considered equivalent in that they can represent a segment of the chaotic system outputs with the same length. The length of the segment is measured by the product of the number of training samples and the step size. For a segment with a given length, the training samples generated with a smaller step size will result in a larger number of training samples. For example, the length of 1k samples data set with *dt*=0.01 is 1k $\times$ 0.01=10; and the length of 2k samples data set with *dt*=0.005 is also 2k$\times$0.001=10. It needs to be clarified that the equivalent time series segments are not identical because they are generated using Euler method with different step sizes, which cause the time series to diverge. Nevertheless, the comparison results show that the training performance can be improved by increasing the precision of the training data with increased number of training samples when the training samples are more than 1k. The predictability indicated by the validation performance can also be improved using training samples generated with better representation precision by increasing the number of data samples to represent the time series segment with the same length. Moreover, it is insufficient to use less than 1k training samples to effectively train the ANN.



(a) 200, dt=0.005 (b) 400, dt=0.005 (c) 800, dt=0.005 (d) 2k, dt=0.005

(e) 4k, dt=0.005 (f) 8k, dt=0.005 (g) 16k, dt=0.005

Fig. 8. Lorenz system equivalent time series segments (*dt*=0.005).



(a) Best Training MSEs (b) Best Validation MSEs

Fig. 9. Training results for equivalent time series segments (ANN(*n*=6, *d*=1), divideblock).

V. CONCLUSION AND FUTURE WORK

Three conclusions can be drawn from the training results in the application of Lorenz chaotic system design based on ANN. First, the size of the training data does not have significant impact on the training performance. At the same precision, a small training data set can be trained to generate the desired chaotic time series with equivalent performance as a big data set but much reduced training time. Second, the training performance can be better evaluated by using the `divideblock' option in MATLAB to use three separate consequent time series segments for training, validation and testing, without any penalty of degrading the training performance compared to the default `dividerand' option, which randomly divide the training samples into three subsets. In fact, the three continuant segments divided by the 'divideblock' option can be used to test the predictability of the trained ANN by measuring the difference between the training and validation performances, as well as the difference between the training and testing performances. Third and most important, the performance and predictability of the trained ANN can be improved by using training samples with better precision, i.e. smaller time step (dt). It is the quality, rather than the quantity of the training data that primarily determines the ANN training results. It is not guaranteed that big data can create big value. One key reason the human brain is highly energy efficient is that it can ruthlessly screen out massive noncritical information (noise) in order to focus on processing a limited amount of most important information. The insight gained from this research lies in the understanding that big data alone is not the panacea for all machine learning applications. More data can only be beneficial for ANN training if it contains useful information. Excessive training data with low data quality may also bring in more noise and reduce training efficiency. The successive research will be carried out for the hardware implementation of the designed ANN with fixed-point data representation, which will introduce quantization error to both the training data and the ANN architecture, including the values of weights and biases, as well as the activation function.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[3] M. Shim, H.-J. Hwang, D.-W. Kim, S.-H. Lee, and C.-H. Im, "Machine-learning- based diagnosis of schizophrenia using combined sensor-level and source-level EEG features," *Schizophrenia Research*, vol. 176, no. 2-3, pp. 314–319, Oct. 2016.

[4] H. G. Schnack, M. Nieuwenhuis, N. E. van Haren, L. Abramovic, T. W. Scheewe, R. M. Brouwer, H. E. H. Pol, and R. S. Kahn, "Can structural MRI aid in clinical classification? A machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects," *Neuro Image*, vol. 84, pp. 299–306, Jan. 2014.

[5] J. Kambeitz, C. Cabral, M. D. Sacchet, I. H. Gotlib, R. Zahn, M. H. Serpa, M. Walter, P. Falkai, and N. Koutsouleris, "Detecting neuroimaging biomarkers for depression: A meta-analysis of multivariate pattern recognition studies," *Biological Psychiatry*, vol. 82, no. 5, pp. 330–338, Sep. 2017.

[6] M. Ahn, J. H. Hong, and S. C. Jun, "Feasibility of approaches combining sensor and source features in brain–computer interface," *Journal of Neuroscience Methods*, vol. 204, no. 1, pp. 168–178, Feb. 2012.

[7] W. J. Freeman, *Societies of Brains: A Study in the Neuroscience of Love and Hate (INNS Series of Texts, Monographs, and Proceedings Series)*, Psychology Press, 1995.

[8] L. Zhang, "Artificial neural network model design and topology analysis for fpga implementation of Lorenz chaotic generator," in *Proc. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Apr. 2017, pp. 216–219.

[9] L. Zhang, "Design and implementation of neural network based chaotic system model for the dynamical control of brain stimulation," in *Proc. the Second International Conference on Neuroscience and Cognitive Brain Information (BRAININFO 2017)*, Nice, France, July 2017, pp. 14–21.

[10] L. Zhang, "Artificial neural networks model design of Lorenz chaotic system for eeg pattern recognition and prediction," in *Proc. The first IEEE Life Sciences Conference (LSC)*. Sydney, Australia: IEEE, Dec. 2017, pp. 39–42.

[11] A. Atakulreka and D. Sutivong, *Avoiding Local Minima in Feedforward Neural Networks by Simultaneous Learning*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 100–109.

[12] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

Lei Zhang is an Assisstant Professor in the Faculty of Engineering and Applied Science at the University of Regina in Canada. She received her Ph.D degree in electronic engineering at Brunel University West London in the UK in 2012.

Prior to her academic career, Lei Has worked as a Project Leader at the TWI Ltd in the UK for six years. Her main work was dedicated to the design and development of a new generation multi-channel ultrasound testing system Teletest® Focus for oil and gas pipeline inspection. She also worked as a Senior Electronic Engineer and Consultant at Cambridge Technology Group for one year. She has expertise and industrial experiences in commercial product design and testing. Her current research interests include artificial intelligence, machine learning, FPGA hardware acceleration for ANN implementation, dynamic patterns recognition using ANN, non-invasive brain stimulation, close loop EEG monitoring of brain functionality, and identification and prediction of mental disorders using EEG-based machine learning approach.

Dr. Zhang is a registered Professional Engineer (PEng) with the Association of Professional Engineers and Geoscientists in Canada. She is also a Chartered Engineer (CEng) with the British Engineering Council, and an European Engineer (Eur Ing) with the European Federation of National Engineering Associations. She has published over 30 peer-reviewed articles.