Bulletin of Electrical Engineering and Informatics
Vol. 8, No. 1, March 2019, pp. 276~282
ISSN: 2302-9285, DOI: 10.11591/eei.v8i1.1405

Progress in neural network based techniques for signal integrity analysis–a survey

Chan Hong Goay, Azniza Abd Aziz, Nur Syazreen Ahmad, Patrick Goh
School of Electrical & Electronic Engineering, Universiti Sains Malaysia,
14300 Nibong Tebal, Pulau Pinang, Malaysia

Article Info

Article history:

Received Oct 15, 2018
Revised Nov 19, 2018
Accepted Dec 19, 2018

Keywords:

Artificial Neural Networks
High speed interconnects
Signal integrity
S-parameters

ABSTRACT

With the increase in data rates, signal integrity analysis has become more time and memory intensive. Simulation tools such as 3D electromagnetic field solvers can be accurate but slow, whereas faster models such as design equations and equivalent circuit models lack accuracy. Artificial neural networks (ANNs) have recently gained popularity in the RF and microwave circuit modeling community as a new modeling tool. This has in turn spurred progress towards applications of neural networks in signal integrity. A neural network can learn from a set of data generated during the design process. It can then be used as a fast and accurate modeling tool to replace conventional approaches. This paper reviews the recent advancement of neural networks in the area of signal integrity modeling. Key advancements are considered, particularly those that assist the ability of the neural network to cope with an increasing number of inputs and handle large amounts of data.

Corresponding Author:

Patrick Goh,
School of Electrical & Electronic Engineering,
Universiti Sains Malaysia,
14300 Nibong Tebal, Pulau Pinang, Malaysia.
Email: eepatrick@usm.my

1. INTRODUCTION

As signal rates increase to the multi-gigabit per second range, signal integrity (SI) becomes a very significant factor in circuit design. At low data rates, a simple conductor can be used to transmit the signal over short distances without causing severe signal degradation issues. However, it becomes more difficult to maintain the characteristics of the transmitted signal waveform as the signal speed increases. This is because effects such as ringing, crosstalk, reflections, and ground bounce start to become significant at high data rates. Consequently, the design engineers of high-speed circuits need to take these effects into consideration, resulting in even more complex electronic designs. In addition, the engineers will also need to consider for variations of design parameters due to manufacturing process limitations. This can also affect the electrical properties of high-speed circuits and cause further unwanted problems.

An example high-speed interconnect topology of a PCI Express 2.0 system is shown in Figure 1. Usually, SI analysis involves two simulation tools, electromagnetic (EM) field solvers and circuit simulators [1]. An EM field solver is used to extract the frequency response of the high-speed interconnect structure. The circuit simulator is then used to carry out time domain simulations to obtain the corresponding output waveforms. EM field solvers are accurate but they are also slow. Even through circuit simulators are generally fast, when the time domain responses involves very long bit sequences, it can still take up a considerable amount of time. Traditionally, engineers need to perform multiple EM and time domain simulations during the circuit design stage which are costly in terms of both computational power and time. Thus, there is an ever increasing demand of faster and more efficient strategies for high-speed circuit modeling and analysis.

Artificial neural networks (ANNs) have been widely applied in RF and microwave circuit modeling problems [2-5]. ANN is an information processing system, with its design inspired by the neuronal structure of mammalian brains. Neural networks can learn the relationship between design parameters and electrical properties of electronic designs. Then, the well trained neural network can be used in the design process, thereby partially or completely replacing the traditional simulation tools. This will speed up the design process because a well-trained ANN is many times faster than both the EM field solver and the circuit simulator. For example, in [6], it is presented that a full wave EM simulation of a side-coupled circular waveguide dual-mode filter takes about 6 minutes when using a mode-matching-based EM simulator, and 45 minutes when using a finite-element-based EM simulator. On the other hand, the ANN method only requires 0.006 second for each evaluation.

Recent works show that ANNs have also been used in fast SI modeling applications [7-11]. Most of the works focus on eye diagram modeling or prediction of SI metrics such as crosstalk or jitter. The eye diagram is a graphical metric that is commonly used to evaluate the performance of high-speed systems. Figure 2 shows an eye diagram with its height, width and timing jitter labelled. The eye diagram is constructed by slicing the time-domain signal waveform into sections that are a small number of symbols in length, and overlaying them on top of each other. Ideally, the eye opening should be as wide as possible so that the design will have enough margins for voltage and timing requirements at the receiver. An eye diagram can also be used to estimate the bit error rate (BER) of a system, which is the rate at which error occurs in digital data transmissions.

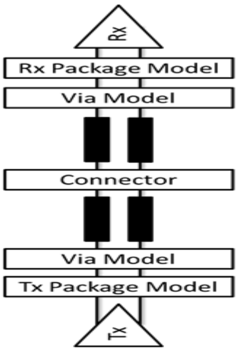


Figure 1. PCI express 2.0 topology from transmitter to receiver

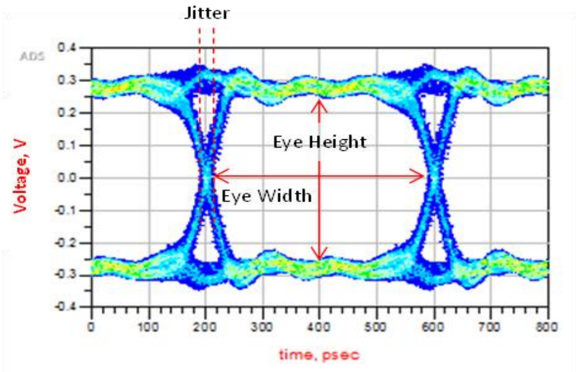


Figure 2. Example of a received eye diagram of a 2.5 Gbps signal

This paper reviews the recent advancement in fast SI analysis using ANN, with a special attention on eye diagrams and output waveform modeling. The remainder of the paper is organized as follows. Section II discusses the background of ANN for circuit modeling. Section III presents an in-depth discussion about recent work in SI analysis using ANN. Finally, Section IV presents a short conclusion and proffers some future work on the subject.

2. ARTIFICIAL NEURAL NETWORKS

ANN is a simplified mathematical model of a biological neural network that consists of interconnected neurons. The multi-layered perceptron (MLP) is one of the most widely used artificial neural network. Its neurons are arranged in L layers, where layer 1 is the input layer, layer L is the output layer, and layers 2 to L-1 are hidden layers. A neural network with one hidden layer is often considered a shallow neural network while those with multiple hidden layers are considered deep neural networks. The general structure of an L-layer perceptron with n input neurons and m output neurons is shown in Figure 3. During training, ANN learns by adjusting its weights as to minimize the training errors, which are defined as the differences between the desired outputs from the training samples and the modeled outputs by the ANN. While back-propagation remains as one of the popular training algorithm, it has been reported that the quasi-Newton, Levenberg-Marquardt and conjugate gradient methods can outperform primitive back-propagation in terms of speed and accuracy in microwave modeling problems [12]. The training goal is to achieve generalization. A neural model with good generalization can provide accurate answers even when it is tested with inputs that it has never encountered before in the training process. Early stopping and regularization are often used to improve the generalization capability of a neural network and avoid over-fitting. The readers are referred to reference [13] for more background about ANN.

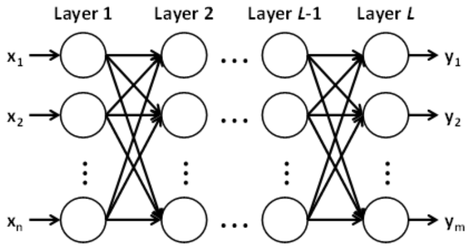


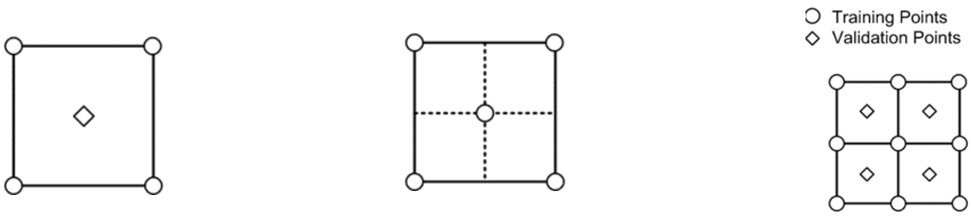
Figure 3. Generic MLP structure

3. SIGNAL INTEGRITY MODELING WITH NEURAL NETWORKS

3.1. Sampling techniques

Neural networks can only be used for circuit modeling after it is trained. Thus, the first step in neural modeling is to generate input-target pairs of the problem to be learned. Often, the generated input-target pairs are divided into three groups called the training set, validation set, and testing set. The training set is used for weights and biases adjustment of the neural model, while the validation set is used to determine the stopping criteria of the training process. Finally, the testing set is used for an unbiased performance estimation of the neural model. Design of experiments (DoE) is a very popular sampling method in various neural modeling applications. One of its variants called the Taguchi design of experiments, which use orthogonal arrays for efficient design space exploration, has been used for the eye height and timing jitter neural modeling of high-speed interconnects [14]. Although DoE has been used in many applications and proven to be an effective sampling strategy, DoE sampling is a one-shot approach which means that the sampling process and training process are carried out separately. If the engineers do not have the full understanding of the input-target mapping function, it is very difficult to decide the sample size to be generated. Also, the degree of linearity may not be consistent throughout the whole design space. These problems can cause under-sampling or over-sampling to occur.

In order to solve this problem, adaptive sampling is proposed [15]. Instead of generating all of the samples at once, adaptive sampling only generates a small amount of samples at first, and then adds further samples iteratively. The idea is to add more samples in a highly non-linear region of the design space compared to the linear regions. At the start of the algorithm, the whole design space is divided into 2n equal volume regions where n is the number of input variables of the problem. Then, training and validation samples are generated for each region and a neural model is created using the newly generated samples. All regions are given performance scores based on the validation errors, which are the errors between the neural network predictions and the targets of the validation samples. The region with the worst performance is then further split into 2n equal volume regions. The process continues until the neural network reaches the minimum user defined accuracy. Figure 4 illustrates the splitting of the worst performing region during the adaptive sampling process for a 2-dimensional modeling problem. In this case, training points are generated at the corners of the regions whereas validation points are generated at the center of the regions. Once the splitting is completed, the validation point of the worst performing region becomes a training point at the next iteration. The pseudocode of the adaptive sampling algorithm is presented in Algorithm 1.



(a) (b) (c)

Figure 4. 2D visualization of the splitting process. (a) The worst performing region. (b) The region is split into 2n new regions, four new regions in this case; also the validation point at the center of original region is changed to a training point. (c) Generate new training and validation points for the worst performing region

Adaptive sampling also allows easier integration between data generation and neural model creation. Several papers on automated model generation (AMG) use adaptive sampling as part of the AMG algorithm [16, 17]. Recent research has also proposed the use of adaptive sampling as a sampling method in the modeling of eye diagrams [18].

Algorithm 1 Adaptive Sampling

```
Initialize: design space as region R.
while (netPerformance < desiredPerformance)
    Generate non-existing learning samples, L and validation samples, V for region R.
    Create and train an intermediate neural model using L.
    Compute netPerformance using V.
    if (netPerformance < desiredPerformance)
        Identify worst performing region as R*.
        Split R* into equal volume regions R1, R2, …, Rk, where k=2^n.
        Delete R* from R.
        Add regions R1, R2, …, Rk, into R.
    end if
end while
```

3.2. Input data selection/preprocessing

It is a common practice to describe a complex design with its port responses instead of its actual design parameters in order to protect any proprietary information about its structure. Thus, some researchers use the frequency responses of the circuits, such as the S-parameters, in place of the design parameters as inputs of their neural models [6-8]. However, this technique has a few weaknesses compared to the conventional approach of using design parameters. Firstly, the neural model generated is probably not suitable for use in optimization routines since it is very difficult to know the structures of the designs just from the S-parameters alone. Secondly, and perhaps more importantly, the neural model in this case does not replace the EM field solver, which is usually the most time consuming part of the design simulation. Third and finally, the neural model will have a large number of inputs, which may slow down the training process. This is because S-parameters are frequency dependent and are normally generated for at least a few hundred points across the frequency range of interest. In addition, for an N-port network, the S-parameter matrix has N2 elements. For example, the S-parameter matrix of a 2-port network has four elements, S11, S12, S21, and S22. Also, the total amount of inputs will be doubled due to the fact that S-parameters are complex numbers, with real and imaginary parts. Despite this, continuous researches are being carried out in this area to improve the accuracy and decrease the model development cost.

In order to reduce the number of simulations, a method called reduced training set (RTS) has been proposed [8]. Initially, the frequency responses of all designs are generated. Then, a certain number of frequency points, M are selected. After that, three designs that contribute to the maximum, median, and minimum values of the frequencies responses are selected as the training samples for each of the selected frequency. This is visualized in Figure 5. The sample sizes of Sd from DoE, and Sr from RTS are given by:

$$S_d = 2^{n-1} + 2n + 1 \tag{1}$$

$$3K \leq S_r \leq 3MK \tag{2}$$

where K is the number of S-parameters used as inputs. Unlike S_d, S_r does not glow exponentially with n. Thus, this technique can reduce the amount of time domain simulations.

Another popular input data preprocessing step is by using feature selection techniques to extract a select few relevant frequencies from large amounts of uniformly sampled frequencies, which can be up to hundreds or thousands of points. In [9], a feature selection method centered around a fast correlation based filter (FCBF) is used to identify the relevant S-parameters metrics. This allows the engineers to train the neural model with only relevant inputs, resulting in a more compact and accurate neural model. A similar idea is used in [19] where the S-parameters are first approximated by a rational function in a form of:

$$f(s) = \sum_{q=1}^{Q} \frac{r_q}{s - p_q} + d + sh \tag{3}$$

where Q is the order of approximation, r_q are the residues, p_q are the poles, d and h are the constant and proportional terms respectively. If $f(s)$ does not have an asymptotic value, the terms d and h can be set to zero, reducing (3) into:

$$f(s)=\sum_{q=1}^{Q}\frac{r_q}{s-p_q} \tag{4}$$

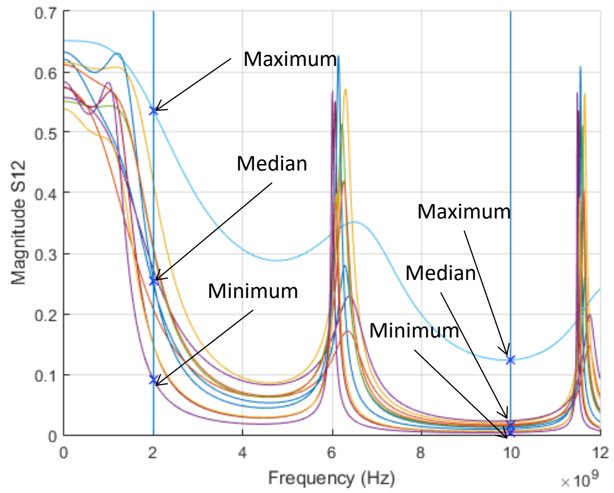


Figure 5. Generation of the input data for a reduced training set

Vector fitting [20], [21] is used to extract the poles and residues from the S-parameters, which are then used in the ANN training process. Since the order of the rational function, Q is much lower than the number of frequency points in the S-parameter, this speeds up the training process significantly. The comparison between neural networks with S-parameters and poles/residues as inputs in terms of the training time is shown in Table 1. The results display significant speed-up when vector fitting is used, the speed-up factor ranges from about 22× (1 hidden neuron) to 1642× (10 hidden neurons). This is because vector fitting reduces the total number of inputs to the neural networks from 1002 to just 108, which reduces the training time and memory requirement for the learning process of the neural networks. Finally it is worth nothing that higher order of approximations, Q can lead to better fitting accuracy during the vector fitting process, but it can also increase the number of inputs to the neural networks. Thus, it is important to keep Q as small as possible without compromising the fitting accuracy.

Table 1. Comparison between training time with s-parameters vs. poles and residues as inputs

No. of hidden neurons	Average training time of neural models with S-parameters as inputs (s)	Average training time of neural models with poles and residues as inputs (s)
1	5.16	0.23
2	31.15	0.24
3	77.31	0.29
4	199.86	0.35
5	398.91	0.50
6	455.62	0.63
7	661.28	0.84
8	1069.50	1.04
9	6261.60	1.18
10	6418.70	3.91

3.3. Output parameter modeling

Eye diagram modeling is one of the most commonly seen applications of neural networks in the field of signal integrity. Specifically, most of the works focus on prediction of the eye-height and eye-width. This is because the minimum height and width of signals at the receiver are important metrics for the performance evaluation of a high-speed channel. Sometimes, a neural network is also used to model the timing jitter of the eye. The eye diagram modeling problem is described as follows. Suppose there is a function that maps design parameters (input) to eye metrics (output), and that a neural network is to be used to learn that function. Conventionally, two simulations are carried out to get the eye diagram. First, frequency responses such as S-parameters are extracted from the design. Then the S-parameters are used as a representation of the design in a transient simulation to get the output waveform and the eye diagram is constructed. By using a trained neural network as a replacement for this mapping, both simulations can be omitted, speeding up the design process significantly. This is especially useful in cases where the eye diagram needs to be generated repeatedly as the design parameters are tweaked, for example during an optimization process. Recently, deep neural networks (DNNs) have also been used in signal integrity applications [10, 11]. DNN refers to a neural network with multiple layers of hidden neurons. A comparative study is carried out and the results show that DNN regression outperform linear regression and support vector regression in terms of eye height and eye width modeling.

4. CONCLUSION

Artificial neural networks have gained attention as popular alternatives to traditional modeling techniques. They can be used to provide fast, accurate, and flexible solutions to many modeling problems. This paper has reviewed the applications of neural networks in the area of signal integrity and some of the challenges in this area. As the complexity of the electronic designs increases, it has become increasingly difficult to model such design. Large number of input samples for neural model creation increases model development costs and reduces time to market. The sampling methods that are reviewed in this paper can be used to reduce the development cost of neural models. Moreover, the reviewed sampling method can also be used in an AMG algorithm which will greatly reduce reliance on human supervision and control. Future work on this area should focus on further advancing input data reduction techniques. A shift towards deep learning could also prove beneficial in potentially allowing complex designs to be modeled accurately without any input data preprocessing.

ACKNOWLEDGEMENTS

This work is supported by the Universiti Sains Malaysia Bridging Grant (304.PELECT.6316113).

REFERENCES

[1] S. H. Hall and H. L. Heck, Advanced Signal Integrity for High-Speed Digital Designs. Hoboken, NJ, USA: Wiley, 2011.
[2] W. Liu, W. Na, L. Zhu and Q. Zhang, "A review of neural network based techniques for nonlinear microwave device modeling," *2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, Beijing, 2016, pp. 1-2.
[3] Qi-Jun Zhang, K. C. Gupta and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design - from theory to practice," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1339-1350, April 2003.
[4] F. Feng and Q. J. Zhang, "*Neural space mapping optimization for EM design*," in Asia-Pacific Microwave Conference (APMC), Nanjing, China, 2015, pp. 1-3.
[5] V. M. R. Gongal-Reddy, F. Feng and Q. J. Zhang, "*Parametric modeling of millimeter-wave passive components using combined neural networks and transfer functions*," in Global Symposium on Millimeter-Waves (GSMM), Montreal, QC, 2015, pp. 1-3.
[6] H. Kabir, Y. Wang, M. Yu and Q. Zhang, "High-Dimensional Neural-Network Technique and Applications to Microwave Filter Modeling," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 1, pp. 145-156, Jan. 2010.
[7] N. Ambasana, D. Gope, B. Mutnury and G. Anand, "Application of artificial neural networks for eye-height/width prediction from s-parameters," *2014 IEEE 23rd Conference on Electrical Performance of Electronic Packaging and Systems*, Portland, OR, 2014, pp. 99-102.
[8] N. Ambasana, G. Anand, B. Mutnury and D. Gope, "Eye Height/Width Prediction From$S$-Parameters Using Learning-Based Models," in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 6, pp. 873-885, June 2016.
[9] N. Ambasana, G. Anand, D. Gope and B. Mutnury, "S-Parameter and Frequency Identification Method for ANN-Based Eye-Height/Width Prediction," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 5, pp. 698-709, May 2017.
[10] T. Lu, K. Wu, Z. Yang and J. Sun, "High-speed channel modeling with deep neural network for signal integrity analysis", *in IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, San Jose, CA, 2017, pp. 1-3.
[11] T. Lu, J. Sun, K. Wu and Z. Yang, "High-Speed Channel Modeling With Machine Learning Methods for Signal Integrity Analysis," in *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 6, pp. 1957-1964, Dec. 2018.
[12] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood and Q. J. Zhang, "Smart Modeling of Microwave Devices," *IEEE Microwave Magazine*, vol. 11, no. 3, pp. 105-118, May 2010.

[13] S. Haykin, Neural networks: a comprehensive foundation. London: Prentice-Hall International, 1999.
[14] W. T. Beyene, "Application of Artificial Neural Networks to Statistical Analysis and Nonlinear Modeling of High-Speed Interconnect Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 166-176, Jan. 2007.
[15] V. K. Devabhaktuni and Q. J. Zhang, "*Neural Network Training-Driven Adaptive Sampling Algorithm for Microwave Modeling*," in 30th European Microwave Conference, Paris, France, 2000, pp. 1-4.
[16] V. Devabhaktuni, M. C. E. Yagoub and Qi-Jun Zhang, "A robust algorithm for automatic development of neural network models for microwave applications," in *IEEE MTT-S International Microwave Symposium*, Phoenix, AZ, 2001, pp. 2087-2090.
[17] W. C. Na and Q. J. Zhang, "Automated Knowledge-Based Neural Network Modeling for Microwave Applications," *IEEE Microwave and Wireless Components Letters*, vol. 24, no. 7, pp. 499-501, July 2014.
[18] C. H. Goay and P. Goh, "Neural Networks for Eye Height and Eye Width Prediction with an Improved Adaptive Sampling Algorithm," *Communications in Computer and Information Science*, vol. 751, pp. 189-201, Aug. 2017.
[19] C. H. Goay, P. Goh, N. S. Ahmad and M. F. Ain, "Eye-Height/Width Prediction Using Artificial Neural Networks from S-Parameters with Vector Fitting", *Journal of Engineering Science and Technology*, vol. 13, no. 3, pp. 625-639, 2018.
[20] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting", *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052-1061, Jul. 1999.
[21] B. Gustavsen, "Improving the Pole Relocating Properties of Vector Fitting", *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1587-1592, Jul. 2006.