Machine Learning Based MoM (ML-MoM) For Parasitic Capacitance Extractions

He Ming Yao[1], Yu Wei Qin[1,2], Li Jun Jiang[1,*]
[1] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
[2] Department of Electrical and Computer Engineering, Carnegie Mellon University
yaohmhk@hku.hk, yuweiq@andrew.cmu.edu, jianglj@hku.hk[*]

***Abstract*—This paper is a rethinking of the conventional method of moments (MoM) using the modern machine learning (ML) technology. By repositioning the MoM matrix and unknowns in an artificial neural network (ANN), the conventional linear algebra MoM solving is changed into a machine learning training process. The trained result is the solution. As an application, the parasitic capacitance extraction broadly needed by VLSI modeling is solved through the proposed new machine learning based method of moments (ML-MoM). The multiple linear regression (MLR) is employed to train the model. The computations are done on Amazon Web Service (AWS). Benchmarks demonstrated the interesting feasibility and efficiency of the proposed approach. According to our knowledge, this is the first MoM truly powered by machine learning methods. It opens enormous software and hardware resources for MoM and related algorithms that can be applied to signal integrity and power integrity simulations.**

Keywords—Method of Moments; Machine Learning; Capacitance Extraction; Artificial Neural Network.

I. Introduction

Computational electromagnetics provides fundamental physical layer models for today's electronic devices ranging from IC, packaging, board to connectors. Numerous methods have been developed. The method of moments (MoM) [1] is a popular approach applied to parasitic extractions, noise analysis, signal integrity, power integrity, etc [2-4].

Machine learning (ML) leads today's IT technology. It develops algorithms to learn from and make predictions on data [5]. It provides best predictions instead of accurate solutions. Hence, its applications to scientific computations are still very limited today. However, due to huge consumer applications, machine learning technologies have received unprecedented attentions, which have created enormous computing resources ranging from software algorithms to hardware platforms. Hence, it is our motivation to build the direct connection between the machine learning and MoM so that we could use ML resources to serve scientific computing for signal integrity and power integrity modeling applications.

In this paper, for the first time according to our knowledge, we reinterpreted MoM into a machine learning process using the artificial neural network (ANN). Based on this new interpretation, the conventional MoM solving is turned into a machine learning training process. Consequently, popular machine learning algorithms such as the multiple linear regression (MLR) are conveniently employed to train the model and obtain the intended solutions [6].

The newly proposed algorithm is applied to parasitic capacitance extractions. The computation employs the popular cloud computing platform – Amazon Web Service (AWS). Numerical tests demonstrate interesting performance advantages over the conventional MoM.

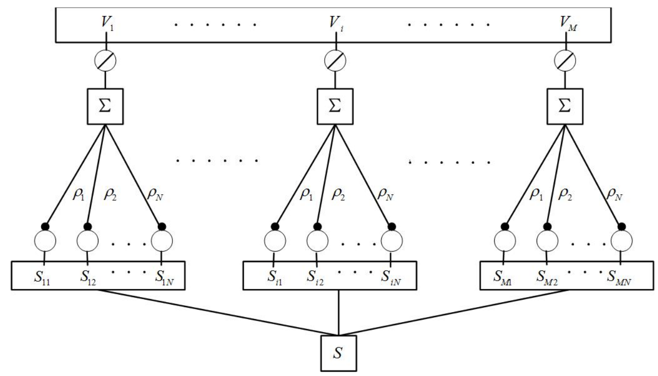


Fig. 1. Artificial nerual network representation of the machine learning based MoM (ML-MoM). ρ_n is the unknown (charge density), V_m, is the excitation (potential) of the MoM equation, and S_{mn} is the matrix coefficient of MoM.

II. From MoM to Machine Learning Process

For the capacitance extraction, the MoM matrix equation can be established based on the integral equation [7]. If the surface charge density ρ is discretized using the pulse basis function and the corresponding conductor potential vector is represented by V, the result matrix equation is

$$\begin{bmatrix} S_{11} & \cdots & S_{1N} \\ \vdots & \ddots & \vdots \\ S_{M1} & \cdots & S_{MN} \end{bmatrix} \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_N \end{bmatrix} = \begin{bmatrix} V_1 \\ \vdots \\ V_M \end{bmatrix} \quad \text{or} \quad \mathbf{S} \cdot \boldsymbol{\rho} = \mathbf{V} \qquad (1)$$

where the elastance matrix **S** has the dimension M by N. M is the number of field point, N is the unknown number for charge density vector $\boldsymbol{\rho}$, and the potential vector **V** has N terms. The elastance element S_{mn} could be weighted using basis at testing field positions. For convenience, the point correlation is employed in this work. Green's function is used as the integration kernel for computing S_{mn}.

When M=N, Eqn (1) is the conventional MoM matrix system. Direct or iterative methods are available to solve (1) based on different scales. After obtaining the charge density, Maxwell capacitances can be obtained using summed charges on each conductor [7].

The starting point of our new algorithm is to reinterpret MoM into a training model, as shown in Fig. 1. It is a single layer perceptron (neural network structure). The matrix element S_{mn} is used as the input training data. The charge density unknown ρ_n is used as the weighting coefficient. The potential V_m is the output of the neural network. Hence, solving MoM becomes a new process: feed S_{mn} to train the model. After training, we can get the accurate estimation of the weighting coefficient ρ_n. It is obvious that more training data will produce better accuracy in the final solution. Hence, when the new algorithm is used, more field points than source points are needed, which means M>N. Hence, the MoM solving process becomes a machine learning training process.

III. Machine Learning For Solving MoM Capacitance Extraction

There are enormous highly developed software resources for machine learning algorithms and hardware resources for machine learning computations.

Among many popular machine learning algorithms, the multiple linear regression (MLR) is employed for parasitic capacitance extractions [8][9]. MLR intrinsically reduces the least square error by using the following cost function

$$\boldsymbol{\varepsilon}_{LS} = \arg\min_{\rho} \sum_{m=1}^{M} \frac{1}{2} (\boldsymbol{S}_m \cdot \boldsymbol{\rho} - V_m)^2 \qquad (2)$$

where $\boldsymbol{S}_m$ is m^{th} row of $\mathbf{S}$ matrix. Regularization can be introduced to further improve the performance of MLR. The regulation by nature represents the extra penalty determined by the prior knowledge

$$\boldsymbol{\varepsilon}_{LS-reg} = \arg\min_{\rho} \sum_{m=1}^{M} \frac{1}{2} (\boldsymbol{S}_m \cdot \boldsymbol{\rho} - V_m)^2 + \frac{\lambda}{2} \boldsymbol{\rho}^T \cdot \boldsymbol{\rho} \qquad (3)$$

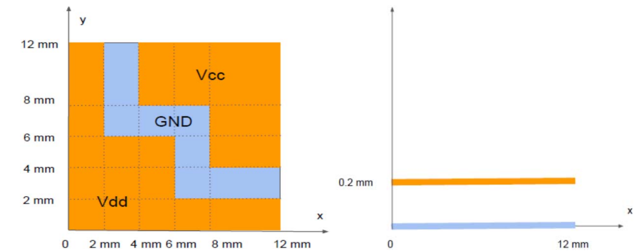
where λ is the regularization coefficient. For example, if it is known beforehand that the linear model is highly likely to over-fit the model, calculated $\boldsymbol{\rho}$ is likely to go higher in its magnitude than the true value. λ will then be assigned with some positive value to increase the cost for achieving big values of $\boldsymbol{\rho}$. Instead, for under-fitting prior knowledge, λ is usually chosen some negative value.

The software used for ML-MoM capacitance extraction in this paper is Python [10]. The preference over other candidates is due to its adaptability towards various systems and well-developed machine learning packages and libraries. Most numerical operations are completed using Numpy and Scipy Libraries [11], and parts involving machine learning are realized with Scikit-Learn – a most popular machine learning library in Python [12].

Among many popular hardware platforms for machine learning, we use Amazon Web Service (AWS) – a cloud-computing service suite providing a broad range of machines with a wide range of configurations. The machine used is m4 instance, the latest generation of General Purpose instance offered by AWS [13]. It is configured with 16 vCPUs and 64 GB memories. To test hypothesis over a wide range of matrix sizes, another 200 GB volume is attached to the machine.

IV. Numerical Examples

The first example is to extract capacitances of power planes as shown in Fig. 2. Three coupling plates (Vdd, Vcc, and GND) are located at two layers. The ground plate has a dimension of 12×12 mm^2, and the distance between two layers is 0.2mm. Pulse basis is used for the charge density discretization. First the conventional MoM and the newly proposed ML-MoM based on MLR training are compared in the capacitance extraction. The comparison result and relative error analysis are shown in Fig. 3. Interesting advantages of the proposed new algorithms over the traditional MoM are observed in terms of accuracy and memory usage. Detail discussions will be presented at the conference. It does not always say ML-MoM is superior over MoM at this moment. But it clearly states the efficiency and feasibility of ML-MoM.



(a) (b)

Fig. 2. Power-ground plane structure for capacitance extraction. (a) vertical view, (b) plane view

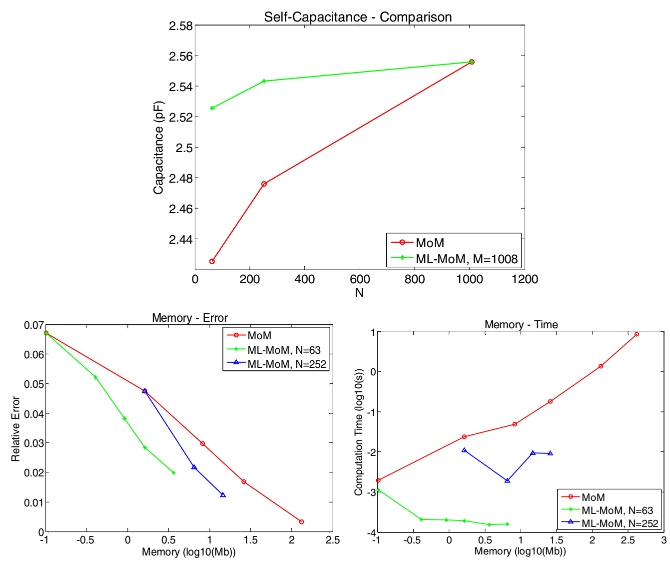


Fig. 3. Capacitance comparison between MoM and ML-MoM. N is the number of charge discretization, and M is the field testing point number.

The second example demonstrates the effectiveness of regularization and increasing number of the training data. Figure 4 shows two scenarios: over-fitting and under-fitting. Over-fitting is more likely to occur when the model itself is less complicated (low rank), whereas under-fitting occurs when the model is oversimplified.

In Fig. 4 (a), two plates are placed perpendicular to each other at a distance of 2 μm. The upper one carries 1V voltage and the lower one is grounded. Quantity of interests is the charge distribution over the upper plate. To make fair comparison, we set the converging point at N= 9216 for three lines as the reference value of total charge on the upper plate.

At a first glance, it is obvious that ML-MoM (blue starred) has little change as N (number of unknowns) increases. This is because over-fitting occurs for relatively simple model which doesn't need too many variables to capture useful features. Compared with ML-MoM, traditional MoM (red circle) produces results larger than the reference value. This phenomenon agrees with the fact that since MoM has number of data equal number of variables, thus requiring larger trained weights to bend the trend of curve so that it could perfectly fit all information contained in the square matrix, including irrelevant noises or minor inaccuracies. The regularized ML-MoM, on the other hand, is much closer to the true value comparing traditional MoM. Given a relatively high regularization coefficient, getting a large value training coefficient becomes costly. Regularized ML-MoM forces training process to ignore certain minor deviation in the data from the general trend of the curve.

In Fig. 4 (b), a same structure to the previous example is used. The converging point (N= 4032) of three planes is set as reference value. This time, the under-fitting is the source of error in final results. The key to increase accuracy given under-fitting is to include more features from the model. As a result, despite ML-MoM has more observation, enhancing N (the source discretization) still increases accuracy effectively. Another difference between over-fitting and under-fitting in ML-MoM is the sign of regularization coefficient. In over-fitting, λ is chosen to be a positive number. It represents the punishment for capturing minor features as prior knowledge determines them likely to be noise. Whereas in under-fitting, λ is chosen to be negative. It means that the training process will be rewarded whenever it amplifies the minor features given the prior knowledge.

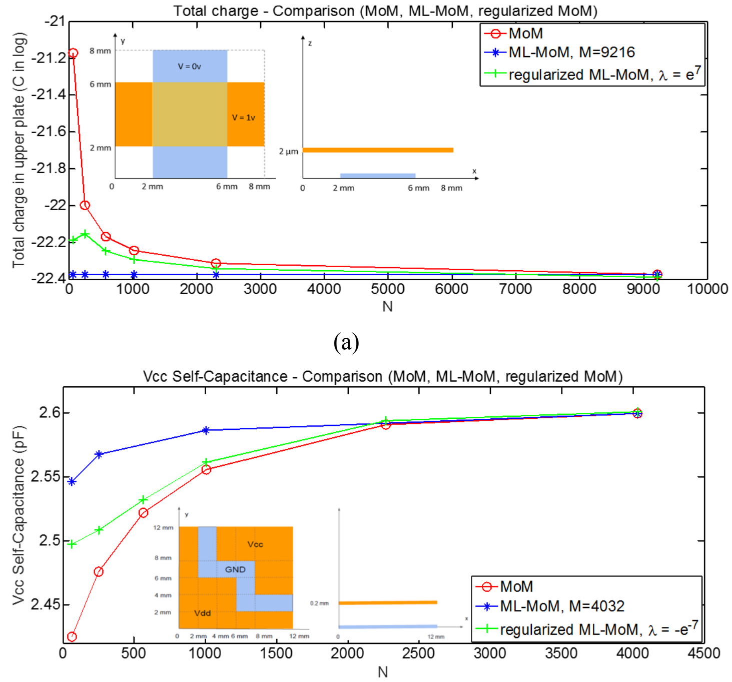
Even though the regularized ML-MoM does improve the accuracy, it requires prior knowledge about the model and a proper choice of λ. Both are not trivial to obtain in reality, making regularized MoM less applicable. On the other hand, ML-MoM avoids such concerns and offers a more viable solution.

V. Conclusion

In this paper, a novel machine learning based method of moment (ML-MoM) is introduced for the parasitic capacitance extraction. By reinterpret MoM using artificial neural network, the conventional MoM solving is changed into a standard machine learning training process. Further taking advantages of machine learning algorithms such as MLR, software resources such as Python, and hardware resources such as Amazon Web Service, parasitic extractions can be executed effectively through modern machine learning technologies. The proposed method opens a new gateway for modern computational electromagnetics (CEM) toward electromagnetic compatibility (EMC). Many new opportunities and possibilities could start from this point.

Acknowledgment

This work was supported in part by the Research Grants Council of Hong Kong (GRF 17207114 and GRF 17210815), NSFC 61271158, and Hong Kong UGC AoE/P–04/08.



(a)

(b)

Fig. 4. Comparison among MoM, ML-MoM and ML-MoM with regulation. (a) Over-fitting case: ML-MoM has 9216 observation point; N varies from 64 to 9216; regularzied ML-MoM has a regularization coefficient $\lambda = e^7$. (b) Under-fitting case: ML-MoM has 4032 observation point; N varies from 63 to 4032; regularzied ML-MoM has a regularization coefficient $\lambda = -e^{-7}$.

References

[1] R. F. Harrington, *Field Computation by Method of Moment*, Macmillan, New York, 1968.

[2] B. Karthikeyan and V. K. Hariharan, "Computation of free space capacitance of cylinder and truncated cone for spacecraft circuit modeling," *Int.J.Space Science and Engineering*, vol. 1, no. 2, pp. I77-189, Jan. 2013.

[3] P. D. Mehta and S. B. Chakrabarty, "Electrical capacitance of dielectric coated metallic parallel piped and closed cylinder isolated infree space," *J Electrostat*, vol. 71, no. 71, pp. 756-762, Aug. 2013.

[4] S. Ghosh and A. Chakrabarty, "Estimation of capacitance of different conducting bodies using rectangular subareas," *J Electrostat*, vol. 66, no.3-4, pp. 142-146, Mar. 2008.

[5] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, Aug. 2006.

[7] W. C. Chew and L. Jiang, "A Complete Variational Method for Capacitance Extractions," *Prog Electromagn Res*, vol. 56, pp. 19-32, Jan. 2006.

[8] Y. Zhang, R. Li, and C. L. Tsai, "Regularization parameter selections via generalized information criterion," *J Ame Stat Assoc*, vol. 105, no. 489, pp. 312-323, 2010.

[9] L. Breiman, "Heuristics of instability and stabilization in model selection," *Ann Stat*, vol. 24, no. 6, pp. 2350-2383, Dec. 1996.

[10] *Python Software Foundation. Python Language Reference*, version 2.7. Available at http://www.python.org

[11] J. Hugunin, "The Python Matrix Object: Extending Python for Numerical Computation," *Proceedings of the Third Python Workshop*, Reston, Va., Dec. 1995.

[12] F. Pedregosa, et al., "Machine Learning in Python", *J Mach Learn Res*, vol. 12, no. 10, pp. 2825-2830, Jan. 2011.

[13] http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/gsg-aws-intro.html