Electrostatic Potential Profile Generator

for Two-Dimensional Semiconductor Devices

Seung-Cheol Han, Jonghyun Choi, and Sung-Min Hong,

School of EECS, Gwangju Institute of Science and Technology, Gwangju, Republic of Korea, email: [smhong@gist.ac.kr](mailto:smhong@gist.ac.kr)

*Abstract*—In this work, deep neural networks are employed to generate two-dimensional electrostatic potential profiles for an efficient device simulation. By using simulation results for various BJT devices, the deep neural networks are trained in supervised manner. To use the generated electrostatic potential profile as an initial guess for a non-equilibrium condition, carrier densities are estimated by the frozen field simulation. With the generated potential profiles, the number of the Newton iterations can be reduced significantly.

1. INTRODUCTION

In these days, deep neural networks are applied to various fields. In the semiconductor research field, there are two major directions related to deep neural networks. The first direction is to develop a new hardware system to implement deep neural networks efficiently [1, 2]. The other direction is to use the deep neural networks as an optimization tool in the technology development cycle [3, 4]. Nonetheless, it is difficult to find an effort to improve the device simulation directly.

In the device simulation, the CPU time depends on the number of bias points. To improve the computational efficiency, initial solutions are provided by the deep neural networks. If the generated profiles are close enough to the numerical solutions, the bias ramping process can be skipped. In our previous work [5], the deep neural networks could generate good initial profiles for the one-dimensional structures. In this work, to be more realistic than one-dimensional structures, we propose deep neural networks for two-dimensional structures such as BJTs.

1. NEURAL NETWORKS

As discussed in our previous work [5], the electrostatic potential is a key quantity for deep neural networks. Under a fixed electrostatic potential profile, the electron and hole continuity equations become linear or at most locally nonlinear, depending on the adopted physical models. For this reason, the deep neural network is designed to generate electrostatic potential profiles. Fig. 1 shows a conceptual diagram for the designed neural network. Bias conditions are used as input parameters. The output parameters are generated electrostatic profiles. Differences between generated and simulated electrostatic profiles are used to calculate the mean square error loss for a linear regression model in the training phase [6]. Since the device simulators need additional quantities such as electron densities and hole densities, the carrier densities are estimated by fixed-potential simulations. To construct the neural network structure, the code written in Python with PyTorch library is considered. For the device simulation, our in-house code written in C++ is used.

1. BIPOLAR JUNCTION TRANSISTOR

In our previous work [5], a deep neural network was applied only to the one-dimensional case. As its extension to the two-dimensional case, the convolutional neural network structures are considered. In this work, the modified generative neural network structure of DCGAN is used [7]. Fig. 2 shows the neural network structure employed in this work. There are two BJT structures as shown in Fig. 3. The first example is a simple BJT structure. It is a quasi-1D structure. The training data set contains 10,000 instances of 64-by-64 electrostatic potential profiles. The base voltage varies from 0.0V to 0.5V. The collector voltage is changed from 0.0V to 0.4V. The training and validation errors are measured as functions of the learning epoch as shown in Fig. 4. After the training phase, the trained convolutional network can generate an approximate potential profile in the inference phase. Fig. 5 shows an example of the electrostatic potential profile generated by the trained convolutional neural network. Its error is shown in Fig. 6. The maximum error of all test cases (not included in the training set) is lower than 21mV. Fig. 7 shows the convergence behavior.

The second example is a more realistic BJT structure. The base contact is located on the top surface. The training and validation error are shown in Fig. 8. Fig. 9 and Fig. 10 show an example of the generated electrostatic potential profile and its error, respectively. The maximum error of all test cases (not included in the training set) is lower than 38mV. Also its convergence behavior is shown Fig. 11.

1. CONCLUSION

In conclusion, the trained convolutional neural networks can generate the electrostatic potential profiles which are close to the solutions. These profiles are used as initial solutions. For all cases tested in this work, the number of the Newton iterations is 4 with the generated potential. On the other hand, the bias ramping with 9 bias points takes 36 iterations. Currently, we are working on a single neural network which can consider multiple structures. Progress on this direction will be reported elsewhere.

REFERENCES

[1] W. Haensch, “Analog computing for deep learning: Algorithms, materials & architectures,” IEDM, 2019.

[2] J. Wesler, J. W. Pitera, and C. Goldberg, “Future computing hardware for AI,” IEDM, 2019.

[3] Bokyeom Kim and Mincheol Shin, “Machine-Learning-Based Device Optimization with TCAD,” KCS, 2020.

[4] R. Orihara et al., “Approximation of time-consuming simulation based on generative adversarial network,” ISCSA, 2018.

[5] S.-C. Han and S.-M. Hong, “Deep neural network for generation of the initial electrostatic potential profile,” SISPAD, 2019.

[6] S. J. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach,” Third Edition, Prentice Hall, 2010.

[7] A. Radford et al., “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv:1511.06434.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fig. 1. Conceptual diagram for the proposed deep neural network. Two-dimensional structures are considered in this work. | | Fig. 2. Layer structure of the CNN structure adopted in the two-dimensional problem. The output layer generates a 64-by-64 matrix corresponding to the two-dimensional simulation domain. It is a modified version of the generator in the deep convolutional generative adversarial network in [6]. | | | |
| Fig. 3. Structures under consideration. A simple BJT (Left) and a more realistic BJT (Right). | | Fig. 4. Training and validation errors of a convolutional neural network which is trained for the simple BJT structure. | | | Fig. 5. Numerical solution (Left) and a generated potential profile by the convolutional neural network with the simple BJT structure (Right). when and 0.2586. |
| Fig. 6. Difference between two potential profiles in Fig. 5. The maximum absolute error is lower than 21mV. | Fig. 7. Comparison of convergence behavior. The simulation with the generated initial profiles takes only four iterations for the converged solution. The target bias condition is and . | | | | Fig. 8. Training and validation errors of a convolutional neural network, which is trained for the realistic BJT structure. |
| Fig. 9. Numerical solution (Left) and a generated potential profile by the convolutional neural network with the more realistic BJT structure.(Right) when and 0.3475. | | | Fig. 10. Difference between two potential profiles in Fig. 9. The maximum absolute error is lower than 38mV. | Fig. 11. Comparison of convergence behavior. The target bias condition is and . | |