Training Better CNN Models for 3-D Capacitance Extraction with Neural Architecture Search

Haoyuan Li*, Dingcheng Yang*, and Wenjian Yu
Dept. Computer Science & Tech., BNRist, Tsinghua University, Beijing 100084, China

***Abstract*—More accurate capacitance extraction is demanded for IC design nowadays. The pattern matching approach and the field solver for capacitance extraction have the drawbacks of inaccuracy and large computational cost, respectively. Recent work [1] proposes a grid-based data representation and a convolutional neural network based capacitance models (called CNN-Cap) for 3-D capacitance extraction. In this work, the techniques of neural architecture search (NAS) is proposed to train better models for 3-D capacitance extraction. Experimental results show that the obtained NAS-Cap model achieves higher accuracy than [1].**

I. Introduction

Accurately modeling the interconnect parasitics (resistance and capacitance) is crucial for guaranteeing the performance of ICs [2], [3]. As billions of transistors are placed within a chip, it is very challenging to perform full-chip capacitance extraction. A solution of this is the pattern matching approach widely used in commercial RCX tools. It divides a large structure into small substructures, and then computes the capacitances of each substructure with pre-built empirical formulas or look-up tables. For a given process technology, a pattern library is pre-characterized by enumerating millions of sample structures and solving the capacitances for them with accurate field solver. Then, the empirical formulas or tables can be obtained for the pattern structures, so that the capacitances can be quickly computed for a specific design. However, this approach loses accuracy for nowadays IC, due to limited coverage of interconnect typologies or the error of empirical formulas.

Another approach is based on field solver [3], [4], which has the highest accuracy. However, due to excessive computational cost, it can only handle a few of small pattern structures. Recently, the deep learning opens the third way for capacitance extraction [1], [5]. A convolutional neural network based approach (called CNN-Cap) for building models for computing capacitances of 2-D and 3-D structures was proposed [1]. It employs a novel grid-based data representation and leverages the CNN's ability of capturing spatial information to deliver more accurate models than using MLP neural network. The CNN-Cap model based on ResNet architecture [6] exhibits good accuracy on predicting total capacitance of 3-D structures, i.e. with less than 5% error in about 99% probability, and runs 191X faster than the fast random walk based capacitance solver [4], [7]. However, its performance on coupling capacitance is not good enough; there are 4.1% of computed coupling capacitances with more than 10% relative error [1].

*Equal contribution. This work is partially supported by the National Key R&D Program of China (No. 2021ZD0114703) and NSFC (No. 62090025).

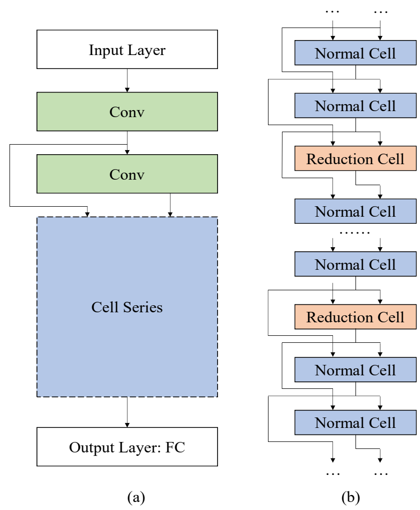

Fig. 1. The architecture of the proposed model. (a) Overview of the whole network model, (b) Connection among cells in the cell series.

Neural architectures assume a pivotal role in the realm of deep learning. Several techniques of neural architecture search (NAS) was proposed to automatically and efficiently design the neural architecture that achieves high performance, such as DARTS [8]. Inspired by the fact that human-designed networks are often constructed from a repetitive stacking of a basic module, DARTS only searches for a basic module (called cell), where a continuous differentiable optimization problem is solved by gradient descent techniques. In this work, we leverage DrNAS [9], an improved version of DARTS, to obtain more accurate CNN models for 3-D capacitance extraction. We find out a better CNN structure than ResNet used in CNN-Cap. Experimental results on 3-D interconnect structures show that the proposed techniques largely reduce the model's prediction error on coupling capacitance. Compared to CNN-Cap [1], the average error of coupling capacitance is reduced from 3.1% to 2.5% and the percentage for coupling capacitances with error larger than 10% is reduced from 4.1% to 1.7%.

II. Proposed Method

In order to find a more suitable network for capacitance extraction, we use the NAS technique to do automatic search. The search space of NAS should be large enough but does not consume huge computational resource for searching. We consider the network architecture in Fig. 1(a), with the connection among cells in the cell series in Fig. 1(b). As in ResNet and many other convolutional network, feature map shrinks its size and increases the number of channels while passing through some layers, we employ two kinds of cells: normal cell and

979-8-3503-4859-0/DATE24/© 2024 EDAA

reduction cell. The reduction cells are the cells that halve the size of input data and double the number of channels.

Either normal cell or reduction cell can be represented as a directed acyclic graph (DAG) consisting of N_d nodes. Each node $x^{(i)}$ is an intermediate representation and each directed edge (i,j) is associated with an operation $o^{(i,j)}$ that transforms $x^{(i)}$. The operation is selected from a candidate set O. In order to use gradient-based optimization method for searching the operations, the continuous relaxation is applied. An edge is now associated with a weighted sum of operations

$$\hat{o}^{(i,j)}(x) = \sum_{o \in O} \theta_o^{(i,j)} o(x), \quad \text{s.t.} \sum_{o \in O} \theta_o^{(i,j)} = 1 \ , \qquad (1)$$

where $\theta_o^{(i,j)}$ is the probability weight. Now, the search process can be defined as an optimization problem:

$$\begin{aligned} &\min_{\theta} \mathcal{L}(w^*, \theta), \\ \text{s.t. } & w^* = \arg\min_{w} \mathcal{L}(w, \theta), \\ & \sum_{o \in O} \theta_o^{(i,j)} = 1, \ \forall (i,j), i < j, \end{aligned} \qquad (2)$$

where w are the parameters of the operation function. Directly optimizing θ with gradient descent will probably break the constraints. To maintain the constraints of θ during optimization, we can substitute θ with $\theta = \text{Softmax}(\alpha)$. So, we can optimize α instead and will not break the constraints.

In this work, the operation set O contains common building blocks of convolutional networks, such as convolutional layers of different size, max pooling and average pooling. Besides, there are two special operations, identity operation and zero operation. The zero operation indicates that there is no connection between two node. As for the convolutional layers, a RELU layer is inserted before each of them. Specifically, the operation set contains 3×3, 5×5 and 7×7 normal convolutions, 3×3 and 5×5 separable convolutions, 3×3 and 5×5 dilated separable convolutions, 3×3 max pooling, 3×3 average pooling, identity, and zero operations (11 in total).

The loss function is mean square relative error (MSRE):

$$\mathcal{L}(w, \theta) = \frac{1}{N} \sum_{i=1}^{N} (1 - \frac{f(\boldsymbol{x}^{(i)}; w, \theta)}{\boldsymbol{y}^{(i)}})^2 \ , \qquad (3)$$

where N is the number of training data, $f(\cdot; w, \theta)$ is the neural network, $\boldsymbol{x}^{(i)}$ indicates the i-th input data, and $\boldsymbol{y}^{(i)}$ is the label. This MSRE function includes the relative error, which can attain same accuracy on capacitance of different orders of magnitude. Once the NAS is completed, the architecture, i.e. parameters θ, is fixed. Then, the model represented by w is retained with the loss function (3) with fixed θ.

With some experiments of this NAS process, we find out that the resulted structure does not have batch normalization (BN) layer in most cells. It is different from that the original ResNet or CNN-Cap. We further find out that, the training loss and the validation loss have an inconsistent tendency during training when using the BN, which usually harms the performance. Therefore, we remove all BN layers and perform NAS again. The resulted network model, named NAS-Cap, has the normal cell and reduction cell shown in Fig. 2.

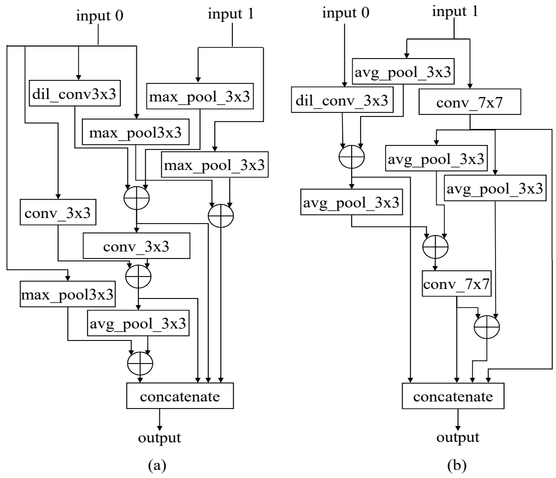

Fig. 2. Architecture of a cell in NAS-Cap. (a) Normal cell. (b) Reduction cell.

III. Experimental Results

We carry out the experiments with the dataset in [10] including 8685 3-D interconnect structures from an SRAM design. Each corresponds to a $5\mu m \times 5\mu m$ window. Setting the metal-layer combination (1, 2, 3) and different master conductors, it further results in 13579 sample structures. We split them to a training subset (90%) and a testing subset (10%).

We compare the performance of proposed NAS-Cap and CNN-Cap [1], [10]. During the model training, the number of epochs is 250, and cosine annealing strategy is applied to learning rate. The prediction errors on coupling capacitance are listed in Table I. We see that with the proposed NAS-Cap model most of coupling capacitances have error within 10%, and only for 1.7% of them the error is be larger than 10%. And, the average error is just 2.5%. As for the total capactiance, NAS-Cap performs with similar accuracy to CNN-Cap.

TABLE I
Prediction Errors on Coupling Capacitance

Method	Err_{avg}	Err_{max}	Ratio (Err>10%)
CNN-Cap [1]	3.1%	44%	4.1%
NAS-Cap	2.5%	40%	1.7%

References

[1] D. Yang, H. Li, W. Yu, Y. Guo, and W. Liang, "CNN-Cap: Effective convolutional neural network-based capacitance models for interconnect capacitance extraction," *ACM TODAES*, vol. 28, no. 4, pp. 1–22, 2023.

[2] L. Lavagno, L. Scheffer, and G. Martin, *EDA for IC Implementation, Circuit Design, and Process Technology*. CRC press, 2006.

[3] W. Yu, M. Song, and M. Yang, "Advancements and challenges on parasitic extraction for advanced process technologies," in *Proc. ASPDAC*, 2021.

[4] W. Yu, H. Zhuang, C. Zhang, G. Hu, and Z. Liu, "RWCap: A floating random walk solver for 3-D capacitance extraction of very-large-scale integration interconnects," *IEEE TCAD*, vol. 32, no. 3, pp. 353–366, 2013.

[5] M. S. Abouelyazid, S. Hammouda, and Y. Ismail, "Accuracy-based hybrid parasitic capacitance extraction using rule-based, neural-networks, and field-solver methods," *IEEE TCAD*, vol. 41, no. 12, pp. 5681–5694, 2022.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[7] M. Yang and W. Yu, "Floating random walk capacitance solver tackling conformal dielectric with on-the-fly sampling on eight-octant transition cubes," *IEEE TCAD*, vol. 39, no. 12, pp. 4935–4943, 2020.

[8] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. ICLR*, 2019.

[9] X. Chen, R. Wang, M. Cheng, X. Tang, and C.-J. Hsieh, "DrNAS: Dirichlet neural architecture search," in *Proc. ICLR*, 2020.

[10] CNNCap, 2023, [Online]. Available: https://github.com/ydc123/CNNCap.