Photonic power splitter design based on deep learning and gradient descend method

Anmin Kong*+, Landobasa YM Tobing*, Yao Zhang
Institute of Microelectronics (IME), Agency for Science, Technology and Research(A*STAR),
2 Fusionopolis Way, Innovis #08-02, Singapore 138634, Republic of Singapore
*) These authors contribute equally, +) kong_anmin@ime.a-star.edu.sg

ABSTRACT

Increasing demand for high density and broadband photonic integrated circuit (PIC) has motivated designs for photonic devices with high performance and compact footprint. However, the number of parameters in traditional photonic device designs is limited by the working principles for such device, which often results in a perceived trade-off among device performances, such as between bandwidth and footprint. Nonlinear optimization such as direct binary search (DBS) and genetic algorithms (GA) have been demonstrated in photonic designs, yet they have drawbacks such as slow convergence time in the range of 96-140 hours. By contrast, designs based on deep learning model relates device performances (output) and device parameters (inputs) via data-driven methodology, which enable arbitrarily large number of design parameter space that may overcome the perceived trade-off in traditional photonic designs.

In this work, we propose a design methodology based on a combination of deep learning model and gradient descent method for photonic power splitter with arbitrary splitting ratio. Using pixel-based device geometry, a deep learning model relating the geometric parameters and device spectral performance is first established. Afterwards, a figure of merit based on a target splitting ratio is optimized through gradient descent method to yield the corresponding pixel-based device geometry. We demonstrate this method in the design of photonic power splitter (based on 400-nm thick Si3N4) with 1:3 ratio, with transmission efficiency of >85% and device footprint not exceeding 16 um^2. Compared to genetic algorithm and direct binary search method, our proposed method also exhibits much faster convergence time.

Keywords: Photonic inverse design, deep learning, gradient descend method, photonic power splitter.

1. INTRODUCTION

Silicon photonics seamlessly integrates with the Complementary Metal Oxide Semiconductor (CMOS) process, offering the benefits of cost-effectiveness. Its versatility finds applications in various domains such as data communication, photonic computing, sensing, and ranging. The growing need for high-density and broadband photonic integrated circuits (PICs) propels the development of compact and high-performance photonic devices. However, conventional designs, constrained by inherent device principles, encounter challenges in the number of adjustable parameters, resulting in trade-offs between crucial aspects like bandwidth and size.

The design of integrated photonic device with complicated geometries is often non-intuitive. Iterative optimization algorithms can be used to optimize the device geometric parameter for targeted optical performance. Nevertheless, establishing explicit functions to define the connections between geometry parameters and optical performance proves to be highly challenging. Consequently, gradient-free algorithms like Direct Binary Search (DBS[1]) and Genetic Algorithm (GA[2]) are utilized, but they exhibit sluggish convergence rates. For instance, DBS might take up to 140 hours, and GA up to 96 hours to converge.

There are gradient-based algorithms like topology optimization[3] (TO) and the adjoint method[4], but they entail significant computational demands. For example, the adjoint method requires solving the original partial differential equations (PDEs) and their corresponding adjoint counterparts, leading to high computational costs, particularly in complex systems or large-scale simulations.

Deep Neural Networks (DNNs) [5] introduce a new paradigm in silicon photonic design due to their ability to uncover complex relationships between input and output data. Designs rooted in deep learning establish connections between device performance and parameters through a data-driven approach, enabling exploration of a vast design parameter space and overcoming perceived trade-offs in traditional photonic designs. Tahersima [6] introduce a method to building and training a Convolutional Neural Network (CNN) as a forward model to predict the output spectra of a power splitter. Tang [7] introduce a method to train a Conditional Variational AutoEncoder (CVAE) as a inverse model to design the component geometry according to the target optical spectrum.

In this study, we introduce a design approach that combines a deep learning model with the gradient descent method for creating a photonic power splitter capable of arbitrary splitting ratios. Initially, a deep learning model is established, correlating geometric parameters with the spectral performance of the device using a pixel-based representation. Subsequently, employing the gradient descent method, we optimize a figure of merit based on a specified splitting ratio to determine the corresponding pixel-based device geometry. We apply this methodology to design a photonic power splitter (utilizing 400-nm thick Si3N4) with a 1:3 ratio, achieving a transmission efficiency of over 85% and a compact device footprint not exceeding 16 um^2. Notably, our proposed method demonstrates significantly faster convergence times compared to genetic algorithms and the direct binary search method.

The paper is organized as follows: Section 2.1 introduces our deep neural network (DNN) model, while Section 2.2 offers insights into our inverse design approach, leveraging the DNN model's output in conjunction with the gradient descent method. Furthermore, Section 2.3 outlines the active learning strategy employed for performance enhancement. Section 3 provides a comprehensive discussion of the experimental results, and Section 4 presents the concluding remarks.

2. OUR METHOD

2.1 Forward deep-learning neural network

We implement a three-port configuration on a conventional fully etched Si3N4 platform. The setup involves a single input and two output ports, each with a width of 0.8 um, linked through adiabatic tapers to a square design region measuring 4 um in width. The silicon nitride platform has a thickness of 400 nm. TE-polarized light is guided through the single-mode input waveguide to reach the 4×4 um^2 design area. Subsequently, the light is tapered into two single-mode output waveguides, corresponding to port 1 and port 2, respectively. The design region is discretized into a grid of 20×20 pixels, each represented by a circle with a 100 nm radius. Each pixel possesses a binary state, where the value 1 signifies a state of non-etching, and the value 0 denotes an etched state.

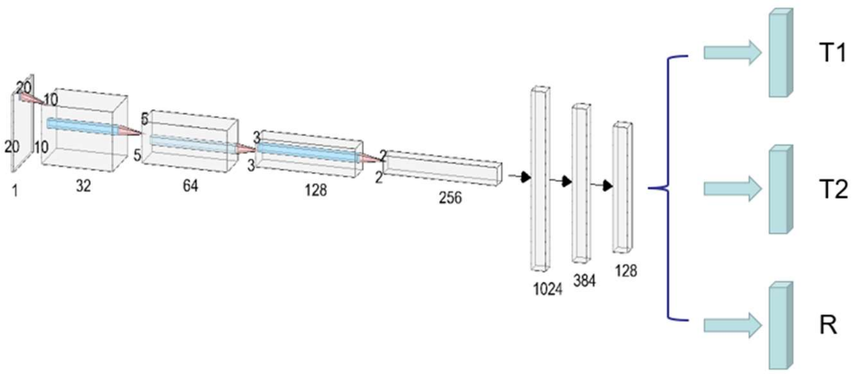


Figure 1. The architecture of the forward deep-learning neural network consists of four convolutional layers with a kernel size of 3 and a stride of 2. The number of output channels for these layers is 32, 64, 128, and 256, respectively. (Figure is plot using NN-SVG https://github.com/alexlenail/NN-SVG)

Concerning the forward deep-learning neural network, the neural network takes a 20x20 hole vectors as input data. The neural network's output comprises the spectral transmission responses at port 1 (T1), port 2 (T2), and the reflection from the input port (R). Each spectral response consists of 11 data points, serving as labels for distinct input hole vector patterns. The objective is to train a neural network capable of predicting the spectral response at each port based on the input hole vectors (HV). To establish the relationship between HV and the corresponding output spectra, we propose an 8-layer deep learning model. The structure of the suggested forward deep-learning neural network is depicted in Figure 1.

The proposed structure comprise four convolutional layers and four full connected (FC) layers. The number of output channels for the four convolutional layers are 32, 64, 128 and 256, respectively, with a kernel size of 3×3 and a stride of 2. Consequently, the output feature size of the convolutional layers are 10×10, 5×5, 3×3 and 2×2, respectively. The convolutional layers serve to extract the spatial features from the input hold vector (HV). After the feature extraction layer, the output feature map is flattened into a 1024×1 vector. We use 4 FC layers to map the feature into the corresponding spectra at output port 1 (T_1), output port 2 (T_2) and the reflection (R) at the input port, respectively. The process involves converting the 1024×1 vector into a 128×1 vector through three FC layers. Subsequently, three separate branches are employed to transform the 128×1 vector into the 11-point output corresponding to T_1, T_2, and R, respectively. The loss function of the DNN model is simply a mean square error (MSE) and can be expressed as

$$Loss = \sum_{i=1}^{n}\left(\hat{T}_{1i} - T_{1i}\right)^2 + \sum_{i=1}^{n}\left(\hat{T}_{2i} - T_{2i}\right)^2 + \sum_{i=1}^{n}\left(\hat{R}_i - R_i\right)^2 \tag{1}$$

Where $\hat{T}_1$, $\hat{T}_2$ and $\hat{R}$ are predict spectra, T_1, T_2, and R are the true spectra, respectively.

Batch normalization is applied to each convolutional layer to enhance training stability and facilitate the convergence of the DNN model. ReLU serves as the activation function to augment the model's non-linearity, enabling it to capture intricate patterns and relationships within the data. And Sigmoid is employed at the output of the fully connected (FC) layer due to its proficiency in fitting smooth curves, particularly for modeling the output spectra.

2.2 Inverse design based on gradient descend method

Our inverse design is based on the gradient descend method. Note that PyTorch and other machine learning framework like TensorFlow offer the autograd function [8]. They can compute the gradient of a function with respect to the inputs by using the automatic differentiation. Assume we have a complex non-linear system (which can be molded using DNN), the input of the system is the hole vector (HV) and the output of the system is a loss function based on some optimization criterion. We can calculate the gradient of the loss function with respect to the input hole vector directly using PyTorch.

Our objective is to find the optimal hole vector capable of achieving the desired power split ratio and maximizing the efficiency of the power splitter. To achieve this goal, we define a target Figure Of Merit (FOM) as the cost function. The FOM comprise two parts, the first part represents the loss incurred due to an inaccurate power split, and the second part represents the loss caused by the efficiency of the power splitter. The FOM can be expressed as follows:

$$FOM = \alpha \int_{\lambda_{min}}^{\lambda_{max}} (T_1 \times r - T_2)^2 d\lambda + (1-\alpha) \int_{\lambda_{min}}^{\lambda_{max}} (1 - T_2 - T_1)^2 d\lambda \tag{2}$$

Where $r = {}^{T_2}/_{T_1}$ is the target power split ratio, λ_{min} and λ_{max} are our target wavelength, respectively. α is the weighting factor to balance the power split error and the insertion loss. In our experiment, we observed that the efficiency of the power splitter contributes significantly more than the accuracy loss. Thus, we use $\alpha = 0.6$, our experiments show that this parameter can balance the splitting ratio accuracy and the efficiency very well. Ideally, FOM should be a very small value close to zero.

Our inverse design can be organized in seven steps: (1) Initialize the hold vector (with a few negative value which stand for etched point); (2) Apply sign function, convert the hole vector into +1 and -1 as the input of the DNN model; (3) Calculate the spectral response of the hole vector using the pre-trained DNN model; (4) Calculate the FOM using the spectral response predicted by the DNN model; (5) Calculate the gradient of the FOM function with respect to the input hole vector; (6) Update the hole vector along the negative gradient direction; (7) Repeat step (2) ~ (6) till the FOM is small enough.

Note that the desired hole vector contains only two type of values, -1 (for etched) and +1 (for not etched). When we update the hole vector, we use a sign function: if the value is smaller than 0, it is mapped as -1 (etched), otherwise it is mapped as +1 (not etched). However, sign function is not differentiable, we can’t get the gradient of a sign function. To solve this problem, we modify the backward function and skip the sign function when we calculate the gradient of FOM [9].

2.3 Active Learning

To further improve the performance of the inverse design, we incorporate the principles of active learning [10].

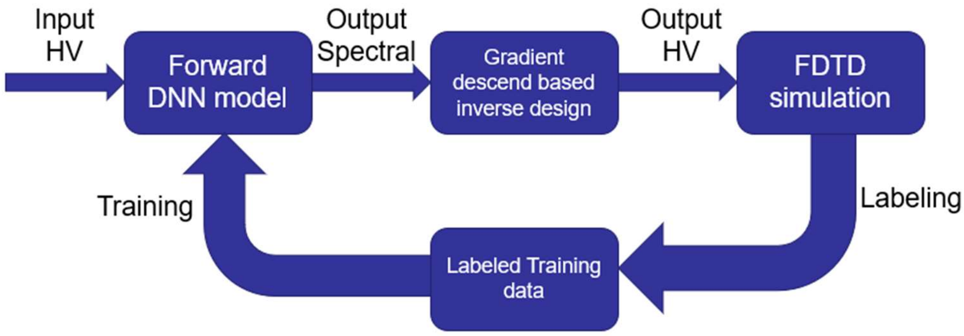


Figure 2. The structure of the active learning based method.

The forward DNN model is first trained using the original dataset. Then the trained DNN model is used to generate 2,000 HV patterns with different splitting ratios. In the third step, we use finite-difference time-domain (FDTD) method to get the true spectral of the 2,000 HV patterns. The FDTD results of the hole vector are added as labels, and the new training data are appended to the original training data for the second-round training. The detailed process is shown in Figure 2.

3. EXPERIMENT AND DISCUSSION

We utilize approximately 20,000 hole vectors (HV) paired with their respective spectral data for training. This dataset comprises randomly selected hole vectors with different percent of zeros (etched points) and carefully chosen semi-optimized hole vectors to collect a diverse set of labeled training data. An additional 2,000 hole vector (HV) patterns, featuring diverse splitting ratios, are generated using the proposed gradient descent method and labeled using FDTD simulation. These newly generated data, along with the original training dataset, are combined to form the updated training data, which is then used to fine-tune the forward DNN model.

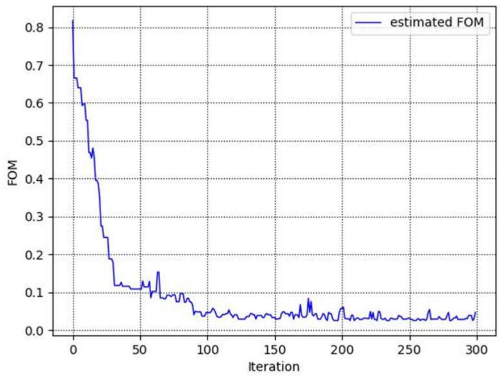
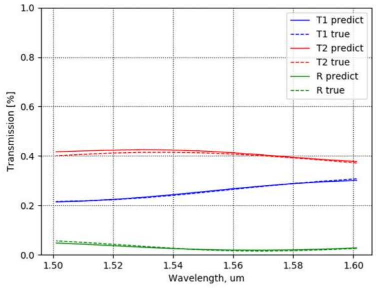
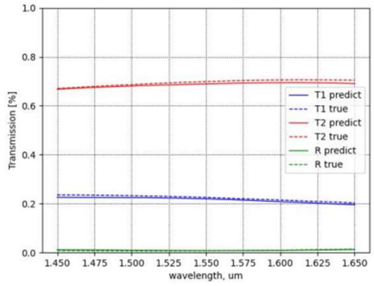


Figure 3. FOM vs the number of iteration.

Figure 3 shows the convergence of the FOM (target ratio is 3, $\alpha = 0.6$) vs iteration of the proposed gradient based method. The optimizer we used is Adam and the learning rate is 5e-3. The initial HV is a fixed pattern with a few negative value (etched points). It can be observed that the proposed method converge very fast, it almost converged after 100 iterations which takes only a few seconds. We also observe that the proposed method generates multiple results with similar Figures of Merit (FOM), demonstrating its inherent capability to address the issue of multiple-to-one solutions.



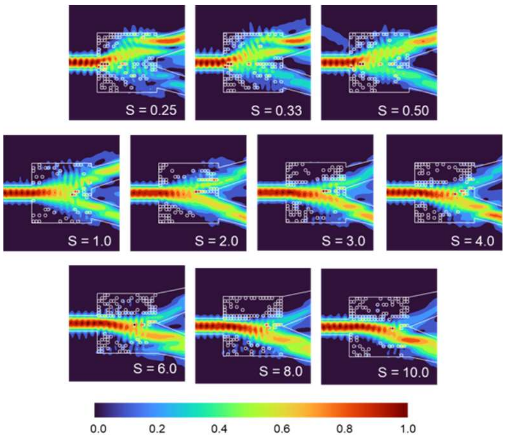
(a) Spectra plot after 5 iterations



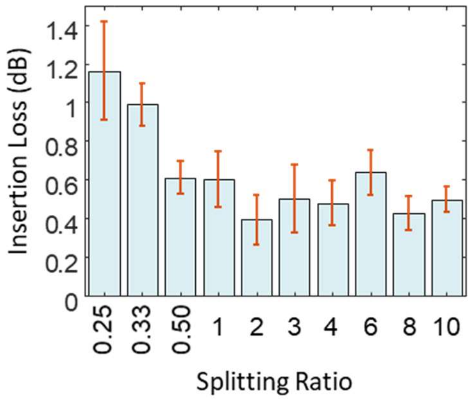
(b) Spectra plot after 100 iterations

Figure 4. The spectra after 5 and 100 iterations, respectively.

Figure 4a and 4b compare the spectra predict by the DNN model and the true spectra at 5 and 100 iterations (target ratio is 3, $\alpha = 0.6$), respectively. It can be observed that the disparity between the predicted spectra and the actual spectra is noticeably minimal, indicating the commendable performance of the forward DNN model. Furthermore, as the number of iterations increases, we observe a reduction in errors attributed to incorrect splitting ratios, leading to an enhancement in the power splitter's efficiency. By the 100th iteration, the device's efficiency exceeds 85%.



(a) Electromagnetic energy density plot



(b) Insertion loss at different split ratio

Figure 5. The performance at different splitting ratio.

Figure 5a and 5b show the Electromagnetic energy density plot and the insertion loss at different splitting ratio. It can be observed that the proposed method works well in different split ratio. And the average insertion loss is larger than 85%.

4. CONCLUSION

In this study, we introduce a design approach that combines deep learning with the gradient descent method to create a photonic power splitter capable of achieving arbitrary splitting ratios. Initially, a DNN model is established, mapping geometric parameters to the spectral performance of the device using a pixel-based representation. Subsequently, utilizing the gradient descent method, we optimize a figure of merit based on a specified splitting ratio to determine the corresponding pixel-based device geometry. Applying this methodology, we design a photonic power splitter (using 400-nm thick Si3N4) with different splitting ratio and achieving a transmission efficiency exceeding 85% and a device footprint not exceeding 16 um^2.

REFERENCES

[1] Seldowitz, M.A.; Allebach, J.P.; Sweeney, D.W. "Synthesis of digital holograms by direct binary search," Appl. Opt. 1987, 26, 2788-2798.

[2] Mao, S.; Cheng, L.; Mu, X.; Wu, S.; Fu, H. "Ultra-Broadband Compact Polarization Beam Splitter Based on Asymmetric Etched Directional Coupler," In Proceedings of the Conference on Lasers and Electro-Optics/Pacific Rim, Sydney, Australia, 3 August 2020; p. C12H-11.

[3] Jensen, J.S.; Sigmund, O., "Topology optimization for Nano-photonics," Proc. Laser photonics Rev. 2011, 5, 308-321.

[4] Lalau-Keraly, C.M.; Bhargava, S.; Miller, O.D.; Yablonovitch,E., "Adjoint shape optimization applied to electromagnetic design," Opt. Express 2013, 21, 21693-21701.

[5] LeCun, Y.; Bengio, Y.;Hinton, G. "Deep Learning," Nature 2015, 521,436-444.

[6] Tahersima, M.H.; Kojima, K.; et al., "Nanostructured photonic power splitter design via convolutional neural networks," In Proceedings of the 2019 Conference on Lasers and Electro-Optics (CLEO), Washington, DC, USA, pp. 1-2.

[7] Tang, Y.; Kojima, K. ; et al., "Generative Deep Learning Model for Inverse Design of Integrated Nanophotonic Devices," Proc. Laser photonics Rev. 2020, 14.

[8] Paszke, A.; Gross, S. ; et al., "Automatic Differentiation in PyTorch," Proc. NIPS 2017 Workshop on Autodiff, 2017.

[9] Courbariaux, M., et al. "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830 (2016).

[10] Aggarwal, Charu C., et al. "Active learning: A survey." Data Classification: Algorithms and Applications. CRC Press, 2014. 571-605.