Inverse design for integrated photonics using deep neural network

Kojima, Keisuke; Koike-Akino, Toshiaki; Tang, Yingheng; Wang, Ye

TR2021-061 June 04, 2021

Abstract

Focusing on nanophotonic power splitters, we show that a generative neural network can design a series of devices that achieve nearly arbitrary target performance, with an excellent capability to generalize training data produced by the adjoint method.

Integrated Photonics Research, Silicon and Nanophotonics (IPR)

Inverse design for integrated photonics using deep neural network

Keisuke Kojima[1,*], Toshiaki Koike-Akino[1], Yingheng Tang[1,2], and Ye Wang[1]

[1] *Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.*
[2] *School of Electrical and Computer Engineering and Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47907, USA.*

[*] *kojima@merl.com*

Abstract: Focusing on nanophotonic power splitters, we show that a generative neural network can design a series of devices that achieve nearly arbitrary target performance, with an excellent capability to generalize training data produced by the adjoint method.

1. Introduction

As the functionality requirement for photonic integrated circuits increases, there is a desire to shrink the device size significantly. There has been significant interest in integrated nanophotonic devices, and the adjoint method [1, 2] has been applied with great success. A potentially complementary technology to the adjoint method is deep neural networks (DNNs), which may be categorized into three models [3]. The first model is a forward regression model, wherein the trained DNN is used within an optimization loop. The second is an inverse regression model, in which the trained DNN takes the target performance as input and generates the desired structure. The third model is a generative network, which can randomly generate a series of optimized design candidates for a given target performance. Generative modeling is a fast growing category, wherein a DNN is trained from a variety of device samples and can generate a series of device structures that can achieve a given target characteristic. In this paper, we review how the generative modeling can be applied to a design of nanophotonic power splitters, and complement the adjoint method [4].

2. Device Structure and Simulation

We consider a nanostructured power splitter with an arbitrary and fixed splitting ratio towards two output ports, targeting flat response with low insertion loss, based on a silicon-on-insulator (SOI) structure with one input and two output ports. It consists of the optimization regions of $3\mu\text{m} \times 3.6\mu\text{m}$, having 151×181 pixels of 20 nm square size as shown in Fig. 1 [4].

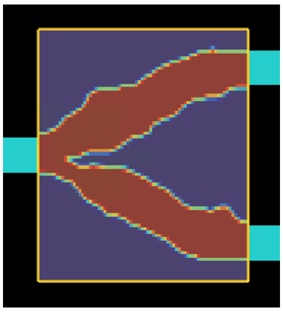

Fig. 1: Schematic of the SOI-based nanophotonic power splitter. The rectangular region of $3\mu\text{m} \times 3.6\mu\text{m}$ shows the area for optimization with 151×181 pixels.

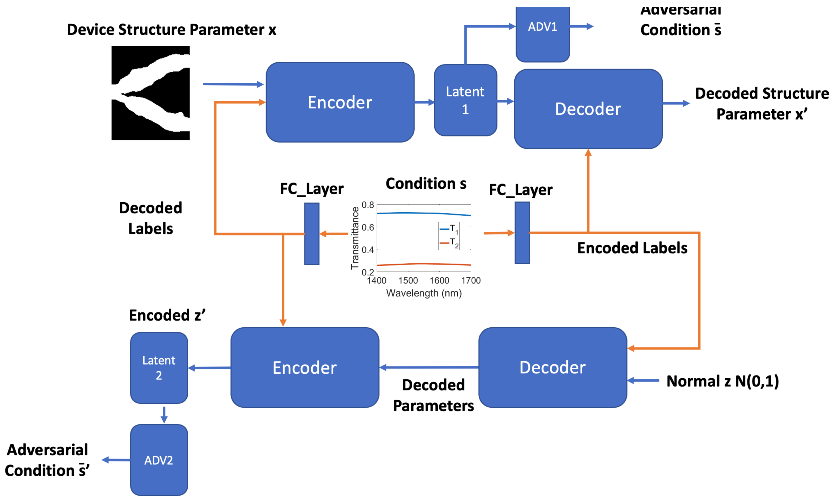

Fig. 2: Adversarial CVAE network with cycle consistency [6]. Both encoders share the same weight, and both decodes share the same weight.

We adopt the adjoint method, to generate the training data, using the LumOpt library [7] to interface with Lumerical two-dimensional (2D) finite-difference time-domain (FDTD). The adjoint method starts with an initial condition and then updates the pixels by calculating the gradients at each pixel. Then, the pixel values are gradually binarized, and also the manufacturing constraints are incorporated. The target splitting power ratios for the two output ports were chosen as, $0.9 : 0.1$, $0.85 : 0.15$, $0.825 : 0.175$, $0.8 : 0.2$, $\ldots$, $0.525 : 0.475$, $0.5 : 0.5$, excluding $0.875 : 0.125$ and $0.725 : 275$ for the testing purpose. We conducted a total of 15 optimization runs, each between 146 and 233 iterations, and accumulated 1,729 data examples for DNN training.

3. Network Architecture and Training

We use a variant of the conditional variational autoencoder (CVAE) for the inverse design of nanophotonic power splitters [5, 6]. For device generation, the trained decoder of the CVAE model is used with the desired condition along with latent variables sampled from the normal distribution, by which a series of device topology candidates are generated. Then, we introduced an adversarial CVAE (A-CVAE) [5], where a separate branch to the adversary block is used for isolating the latent variable z from the conditional variations s (the target spectra). Further, we implement cycle consistency (CC) to further improve the training of the network [6]. The benefit of CC along with the A-CVAE model is that in addition to training the encoder-decoder pair for forward reconstruction, the reversed-order decoder-encoder pipeline is also trained for latent space consistency as shown in Fig. 2 (respective DNN modules share the same weights). Correspondingly, we are comparing not only the decoded device topologies but also the latent variables.

For the training of the whole A-CVAE model with CC, we use an adversarial loss formulation, that incorporates the mean-square error (MSE) loss between the decoded topology x' and actual topology x, a Kullback–Leibler (KL) divergence term to ensure that the latent variables follow the normal distribution, the MSE between the two latent variables z and z' (to enforce reconstruction of the latent variables by the reversed decoder-encoder pair), and the two adversarial blocks that aim to make both $\bar{s}$ and $\bar{s}'$ orthogonal to s as much as possible.

4. Device Generation Performance

In order to verify the effectiveness of the generative model, we consider a series of target from 0.46 : 0.54 to 0.95 : 0.05, with 0.01 increments, to fill the gap of the training data. Fig. 3 shows the plot of mean T_1 and T_2 over the range of 1,450 nm and 1,650 nm of the devices used for the 1,729 training (blue) and the generated (red) devices by our generative model. For example, targeting 0.725 : 0.275, in the middle of the gap, the total transmittance is 98.9%, which compares well to the those of the training data; specifically, that for 0.75 : 0.25 is 98.9%, and that for 0.7 : 0.3 is 99.2%. The number of devices evaluated for each target is 5, and thus a total of 250 devices is generated and validated. The red markers in Fig. 3 show the generated devices, demonstrating a capability that the achieved mean T_1 can arbitrarily cover from 0.48 to 0.90 with an excellent total transmittance grater than 98%.

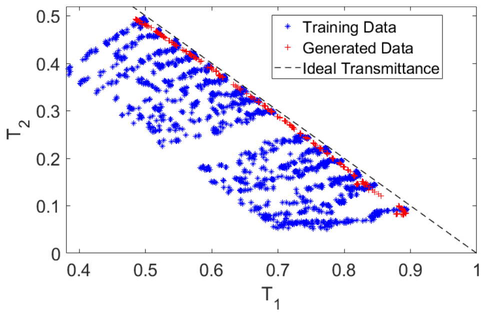

Fig. 3: Mean T_1 and T_2 over the wavelength range of 1,450 nm and 1,650 nm of the devices used for the training and the generated devices by the generative model when the target T_1 is swept from 0.45 to 0.95. The dashed line shows the ideal 100 % transmittance [4].

As the above results show, generative deep learning models can produce a series of good devices based on the statistical characteristics of the training data and target conditions. For our example of 2D FDTD simulations, the adjoint method for each device takes about 2.5 hours. The A-CVAE-CC network training takes about 40 minutes on a computer using a graphic processor unit (GPU) board, which is a one time investment. When we need to generate a series of devices covering a wide range of target performances, as in Fig. 3, it takes 20 seconds to generate these 250 devices, and 21 minutes to validate the performance. Therefore, once a solid DNN model is established, it can generate devices to fill the gaps of the training data much faster than individual adjoint method runs. This DNN method can easily be extended to 3D FDTD simulations [6].

5. Conclusion

To design complicated nanophotonic devices with hundreds of thousands of parameters, a sophisticated design algorithm is necessary, and deep learning offers a promising solution. A series of adjoint method optimization runs are used to generate training data with different target performances. Once trained, nano-optic broadband power splitter design via cycle-consistent adversarial DNN can generate a series of improved device structures given target spectral data. Deep learning shows a good promise for generating structures not covered by the training data, with much shorter time.

References

1. C. M. Lalau-Keraly, *et al.* "Adjoint shape optimization applied to electromagnetic design, " Opt. Exp., **21**, 21693 (2013).
2. A. Y. Piggott, *et al.* "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," Nature Photonics, **9**, 374 (2015).
3. K. Kojima, *et al.* "Deep neural networks for inverse design of nanophotonic devices," JLT, **39**, 1010 (2021).
4. K. Kojima, *et al.* "Inverse design of nanophotonic devices using deep learning," (unpublished).
5. Y. Tang, *et al.* "Generative deep learning model for inverse design of integrated nanophotonic devices," Laser & Photonics Reviews, DOI: 10.1002/lpor.202000287 (2020).
6. Y. Tang, *et al.* "Nano-optic broadband power splitter design via cycle-consistent adversarial deep learning," CLEO, SW4E.1 (2021).
7. `https://lumopt.readthedocs.io`.