Inverse Design of Nanophotonic Devices using Deep Neural Networks

Kojima, Keisuke; Tang, Yingheng; Koike-Akino, Toshiaki; Wang, Ye; Jha, Devesh; Parsons, Kieran; TaherSima, Mohammad; Sang, Fengqiao; Klamkin, Jonathan; Qi, Minghao

TR2020-130 September 30, 2020

Abstract

We present two different approaches to apply deep learning to inverse design for nanophotonic devices. First, we use a regression model, with device parameters as inputs and device responses as outputs, or vice versa. Second, we use a novel generative model to create a series of improved designs. We demonstrate them to design nanophotonic power splitters with multiple splitting ratios.

Asia Communications and Photonics Conference (ACP)
Inverse Design of Nanophotonic Devices using Deep Neural Networks

Keisuke Kojima1,2,* Yingheng Tang1,3 Toshiaki Koike-Akino1, Ye Wang1, Devesh Jha1, Kieran Parsons1, Mohammad H. Tahersima1, Fengqiao Sang2, Jonathan Klamkin2, and Minghao Qi3

1Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.
2Electrical and Computer Engineering Dept., University of California, Santa Barbara, CA, USA
3Electrical and Computer Engineering Dept., Purdue University, West Lafayette, IN, USA

*kojima @ merl.com

OCIS codes: 250.5300, 160.3918, 100.4996.

Abstract: We present two different approaches to apply deep learning to inverse design for nanophotonic devices. First, we use a regression model, with device parameters as inputs and device responses as outputs, or vice versa. Second, we use a novel generative model to create a series of improved designs. We demonstrate them to design nanophotonic power splitters with multiple splitting ratios. © 2020 The Author(s)

1. Introduction

Application of neural networks (NNs) to improve the automation of photonic device design has recently drawn increased attention. In this paper, we show two major categories. First, we show examples of a deep neural network (DNN) used as a regression model. A NN based forward modeling in conjunction with an optimization method has been used for designing nanophotonic power splitters [1], nanophotonic mode converters [2], and optical attenuators [3]. We describe examples of using DNNs for designing nanophotonic power splitters in forward design and inverse design [4]. Second, we show an example of a DNN used as a generative model, which produces a set of new designs which can satisfy the target performance better. Generative models have been recently used for designing materials [5] and plasmonic devices [6, 7]. Here, we introduce a novel conditional variational autoencoder (CVAE) [8] combined with an adversarial network [9]. This is applied to a multi-level nanostructured device design, which can be a more complex optimization problem compared to a binary nanostructured device design, and can benefit from more sophisticated design algorithms.

2. Regression Model

The target device for our design/optimization is shown in Fig. 1, where 220nm-thick silicon-on-insulator (SOI) is used as the waveguide, and 45 nm-radius holes with 130 nm spacing are arranged to split and guide the light. Fig. 2 shows the concept of the regression model, where the network takes device topology design as input and spectral response of the metadevice as label or vice versa. In the first example, we use a DNN in the forward design model. Since the nanophotonic and photonic crystal design are analogous to image processing and recognition problems, we use convolutional neural networks (CNN) to improve the forward prediction accuracy [10, 11].
Fig. 3 shows a comparison of a standard direct binary search (DBS) and DNN-assisted DBS. The latter compares 400 possible positions to flip using DNN before conducting each 3D finite-difference time-domain (FDTD) simulation. Here, we plotted a metric defined as
\[
\text{Metric} = |T_1 - T_1^*|^2 + |T_2 - T_2^*|^2 + \alpha \times R^2,
\]
where \(T_1\) and \(T_2\) are the lowest transmitted power within the spectral range of 1450 nm and 1650 nm. \(T_1^* = 0.3\) and \(T_2^* = 0.7\) are used as an example target. We chose \(\alpha = 10\) as a weighting factor to suppress reflection. The training data do not include any devices with the splitting ratio between 0.27 and 0.33, where the optimal conditions are used.

Alternatively, a DNN can be used in the inverse design mode as shown in the bottom of Fig. 2. Here, we train the DNN using spectral responses as inputs, and device topology as outputs. In this case, we were able to design a device in a single run with a splitting ratio not in the training data [4].

3. Generative Deep Learning Model

In order to generate a series of improved designs from existing sub-optimal designs, we constructed a new generative deep learning model based on a CVAE [6, 8] and an adversarial block [7, 9] as shown in Fig. 4 [12, 13]. The device structure is similar to the one described in Section 2, but with a footprint of 2 \times 25 \mu m^2 with 111.8 nm spacing of etched holes with maximum radius of 38.7 nm. We first use a variational autoencoder [14] and convolutional layers [11] to encode the original 20 \times 20 data (HV: hole vector) to 60 latent variables which are the mean (\(\mu\)) and co-variance (\(\sigma\)) of the normal distribution. Then we concatenate the transmission information to the latent variables and pass into the decoder. The purpose of the adversarial block is to make the encoder only extract the pattern information rather than the performance information. The generated pattern gives the Bernoulli’s distribution at each location point and different hole sizes are used to represent such probability of the appearance of etched holes at certain locations.

To measure the performance of the generated devices, we use the following figure of merit (FOM):
\[
\text{FOM} = 1 - 10 \times \left[ \int_a^b |T_1(\lambda) - T_1^*(\lambda)|^2 d\lambda + \int_a^b |T_2(\lambda) - T_2^*(\lambda)|^2 d\lambda + \int_a^b \alpha \times R^2(\lambda) d\lambda \right],
\]
where \(T_1(\lambda), T_2(\lambda), R(\lambda)\), and \([\cdot]^*\) denote transmissions of output ports 1 and 2, reflection at input port at certain wavelengths, and corresponding target values, respectively. Here, we used a wavelength range of \(a = 1250\) nm and \(b = 1800\) nm. Figure 5 shows the FOM of the generated devices as a function of the splitting ratio. Each star indicates generated data using the CVAE model, while open circles show data using the Adversarial-VAE model. Each color represents different target splitting ratio. The results show that for each targeting splitting ratio, the A-CVAE model generates improved FOM. The generated patterns, beam propagation, \(T_1\), \(T_2\), and \(T_1 + T_2\) spectra are shown in Fig. 6.

References