MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

AutoML Hyperparameter Tuning of Generative DNN Architecture for Nanophotonic Device Design

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

TR2022-046 May 20, 2022

Abstract

We introduce an automated machine learning (AutoML) framework to construct a deep neural network model relevant for inverse design of nanophotonic devices without relying on manual trial-and-error hyperparameter tuning.

Conference on Lasers and Electro-Optics (CLEO) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

AutoML Hyperparameter Tuning of Generative DNN Architecture for Nanophotonic Device Design

Toshiaki Koike-Akino, Keisuke Kojima, Ye Wang

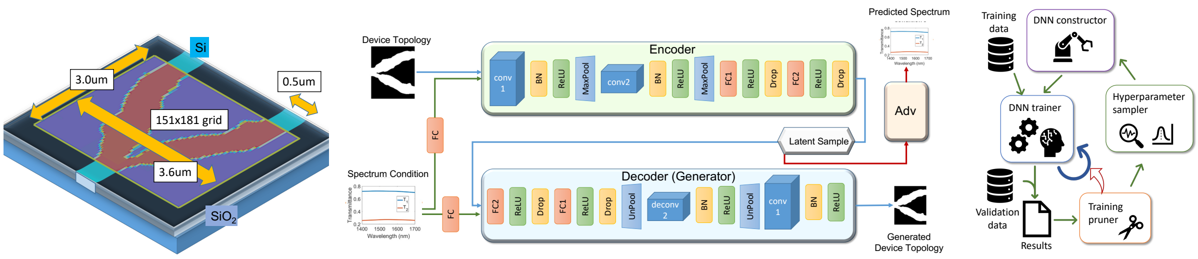
Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.

{*koike, kojima, yewang*}@merl.com

Abstract: We introduce an automated machine learning (AutoML) framework to construct a deep neural network model relevant for inverse design of nanophotonic devices without relying on manual trial-and-error hyperparameter tuning.

1. Introduction: Beyond the most successful framework called the adjoint method [1], a new framework based on deep neural networks (DNN) has drawn increased attention for the design of photonic devices more recently [2–4]. While a number of different DNN architectures has been proposed in literature, generative models have shown state-of-the-art performance [3,4]. However, most DNN models require careful tuning of hyperparameters such as the number of neurons, the number of hidden layers, the type of activation functions, etc. Finding suitable hyperparameters generally involves a considerable amount of manual trial-and-error. The automated machine learning (AutoML) [5] framework was proposed to tackle this issue by an efficient exploration of DNN architectures and hyperparameter tuning. In this paper, we demonstrate that AutoML can efficiently construct a generative DNN model to design a compact broadband photonic beam splitter. We specifically consider a hybrid use of the adjoint method and an adversarial conditional variational autoencoder with cycle consistency (ACVAE-CC) [4] to seamlessly generate good nanophotonic topologies for a given target power splitting ratio.

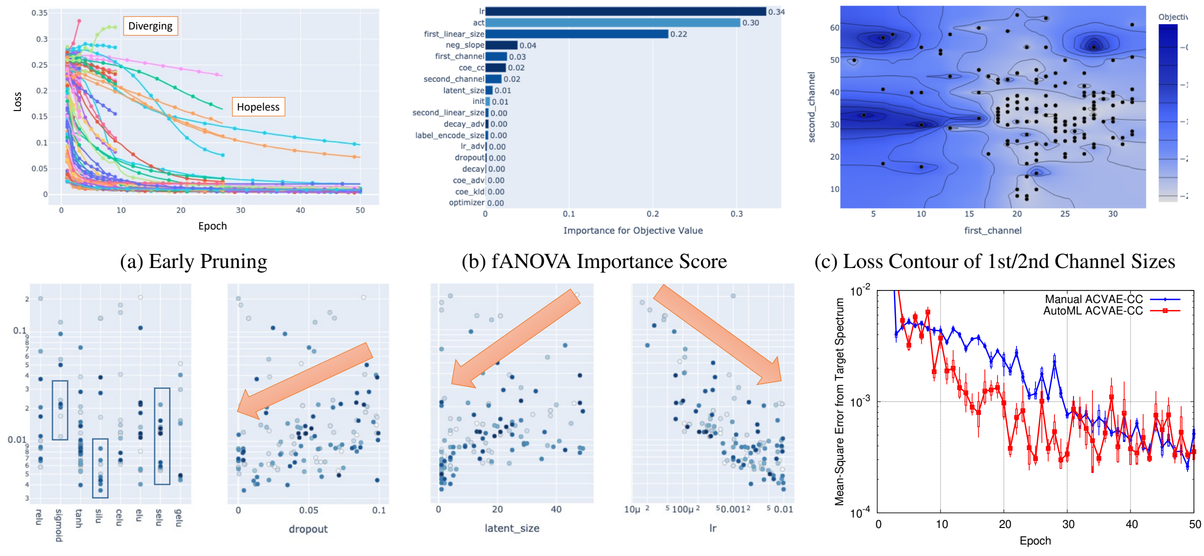
2. Device Structure: We consider a nanostructured power splitter, based on a silicon-on-insulator (SOI) structure, having a target splitting ratio towards two output ports, with a low insertion loss and a flat spectrum over a wide range of wavelengths. It consists of an optimization region of $3\mu\mathrm{m} \times 3.6\mu\mathrm{m}$, having 151×181 nano-pixels of 20 nm square size as shown in Fig. 1(a). The splitter has one input port and two output ports, each of which is a straight waveguide having a width of $0.5\mu\mathrm{m}$. We adopt the adjoint method to generate training data using Lumerical two-dimensional (2D) finite-difference time-domain (FDTD). It iteratively updates the pixels given an initial condition by calculating the gradients. Then, the pixel values are binarized under a fabrication constraint of minimal deposition size. Although the adjoint method is effective at finding better device parameters, it requires a considerable amount of computing resources due to FDTD runs for each target spectrum. In order to complement the adjoint method, we use an ACVAE model to seamlessly generate good device topologies given an on-demand spectrum response. We carried out a total of 15 optimization runs at distinct target splitting ratios from $0.9:0.1$ to $0.5:0.5$, each around 200 gradient steps, and accumulated 1,729 data samples for DNN training.



(a) SOI nanophotonic power splitter (b) Generative ACVAE Model (c) AutoML Tuning

Fig. 1: Nanophotonic power splitter design with AutoML hyperparameter tuning for generative DNN model.

3. Network Architecture: Fig. 1(b) shows an ACVAE model for the device design [4]. For device generation, a trained decoder (bottom) is used with a desired condition along with latent variables sampled from the normal distribution, by which a series of device topology candidates are generated. For the training of the whole ACVAE model with CC, we use an adversarial loss formulation, that incorporates the mean-square error (MSE) loss between the decoded topology and actual topology, a Kullback–Leibler divergence (to enforce that the latent variables follow the normal distribution), the MSE between the two latent variables (to enforce reconstruction of the latent variables by the reversed decoder-encoder pair), and adversarial blocks (to make latent variables orthogonal to the spectrum as much as possible). The ACVAE is constructed with many hyperparameters for convolution, activation, pooling, linear layer, and dropout blocks. Fig. 1(c) depicts AutoML to optimize the DNN hyperparameters. We use Bayesian optimization based on tree-Parzen estimator and hyperband pruning [5]. It automatically



(a) Early Pruning (b) fANOVA Importance Score (c) Loss Contour of 1st/2nd Channel Sizes

(d) Activation Function (e) Dropout Rate (f) Latent Size (g) Learning Rate (h) Device Design Performance

Fig. 2: AutoML analysis of ACVAE-CC hyperparameter tuning.

explores the latent size, activation functions, dropout rate, convolutional channel sizes, hidden node sizes, regularization factors, and learning rate, without the need of human effort in hyperparameter tuning.

4. Results: We partition the dataset into a $9:1$ training/validation split. AutoML explores 1,000 trials of hyperparameter tuning, where each trains with adaptive momentum gradient over a maximum of 50 epochs. Fig. 2(a) shows learning trajectories, depicting that some trials are efficiently pruned by the hyperband strategy due to hopeless or diverging trends. Fig. 2(b) presents the functional analysis of variance (fANOVA) score [5] to assess the importance of hyperparameters, finding that learning rate, activation functions, and hidden node size are the most influential hyperparameters. Convolutional channel sizes are moderately important, with the loss contour plotted in Fig. 2(c) showing a non-trivial landscape for the combination of channel sizes in the first and second convolution layers, indicating difficulty for manual tuning. Figs. 2(d)–(g) show the relation between training loss and hyperparameters. Some activation functions, such as sigmoid, perform relatively poor compared to the sigmoid linear unit (SiLU), and lower dropout rate, smaller latent size, and higher learning rate are generally beneficial.

Fig. 2(h) validates the benefit of AutoML. Here, the optimized hyperparameters via the AutoML trials are used to construct the most relevant DNN model. We use the optimized model to generate 20 new device topologies at an unseen target of splitting ratio $0.725:0.275$. The spectral response of those generated devices are evaluated via FDTD, and compared with the desired flat spectrum. This figure shows the MSE between achieved spectrum and desired spectrum over the range of wavelengths 1,450 nm and 1,650 nm. We also plot the case with manually optimized hyperparameters used in [4], where more than 10 hours of hand-crafted tuning was performed. It was verified that AutoML tuning outperforms manual tuning, achieving MSE lower than 10^{-3} at early epochs. The achieved MSE corresponds to the insertion loss of 0.106 dB for the designed nanophotonic beam splitter.

For our case, the adjoint method takes about 2.5 hours for a single run. The ACVAE-CC network training takes up to 40 minutes with a graphic processor whereas one thousand AutoML trials takes about 10 hours. When we need to generate a series of devices covering a wide range of on-demand spectra, it takes 20 seconds to generate 250 devices, and 21 minutes to validate the spectrum via FDTD. Therefore, once a DNN model is established, it can generate good devices much faster than individual adjoint method runs.

5. Summary: We showed that hyperparameter tuning of a generative DNN model can be efficiently automated and accelerated to achieve a good performance of 0.1 dB insertion loss with fewer training epochs.

References

1. C. M. Lalau-Keraly *et al.*, "Adjoint shape optimization applied to electromagnetic design," OpEx **21**, 21693–701 (2013).
2. M. H. Tahersima *et al.*, "Deep neural network inverse design of integrated photonic power splitters," Sci. Reports **9**, 1368 (2019).
3. Z. Liu *et al.*, "Generative model for the inverse design of metasurfaces," Nano letters **18**, 6570–6576 (2018).
4. Y. Tang *et al.*, "Nano-optic broadband power splitter design via cycle-consistent adversarial deep learning," in *CLEO,* (2021), pp. SW4E–1.
5. T. Akiba *et al.*, "Optuna: A next-generation hyperparameter optimization framework," in *ACM SIGKDD,* (2019).