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.

Introduction
- Deep neural networks (DNN) have been widely used for photonic device design [1]–[10]
- The application has been exponentially growing by 240% every year
- A generative model based on adversarial conditional variational autoencoder (ACVAE) can efficiently design nanophotonic device [4]
- However, hyperparameter selection requires great amount of manual trial-and-error efforts
- We used automated machine learning (AutoML) to optimize model hyperparameters [11]

Learning for Photonic Device
- Application of DNN to optical devices has attracted much attention in the community
- The number of articles follows the Moore’s law
- The annual growth rate is 240%

Nanophotonic Power Splitter
- We consider to design compact nanophotonic power splitter
- Silicon-on-insulator (SOI) platform
- Wideband wavelengths: 1450nm–1650nm

Generative Adversarial CVAE
- Generative CVAE model [4] with cycle consistency [10] is used to optimize the splitter

DNN Hyperparameters
- Selection of hyperparameters for DNN model is cumbersome
- Huge search area: number of hidden layers/nodes, activation functions, convolutional kernel sizes/strides/dilations/channels, latent size, adversarial coefficient, learning rates, etc.
- A large amount of manual efforts in trial-and-error exploration is required to find best hyperparameters

FANOVA Importance Score
- Hyperparameter importance score
- Major factors: Learning rate and activation

AutoML Tuning
- We use AutoML framework called Optuna [11]
- AutoML automatically searches for good hyperparameters
- Bayesian optimization (BO)-based efficient search

Hyperparameter Exploration
- Explored over 10 hours on 6 GPUs
- Manual tuning takes just 20 seconds to generate 250 good devices

Conclusions
- We demonstrated that hyperparameter tuning of a generative DNN model can be efficiently automated and accelerated via AutoML

References