Tutorial: AutoML for NLP

Kevin Duh
Johns Hopkins University
Baltimore, USA
kevinduh@cs.jhu.edu

Xuan Zhang
Johns Hopkins University
Baltimore, USA
xuanzhang@jhu.edu

1 Brief Description

Automated Machine Learning (AutoML) is an emerging field that has potential to impact how we build models in NLP. As an umbrella term that includes topics like **hyperparameter optimization** and **neural architecture search**, AutoML has recently become mainstream at major conferences such as NeurIPS, ICML, and ICLR. The inaugural AutoML Conference[1] was started in 2022, and with this community effort, we expect that deep learning software frameworks will begin to include AutoML functionality in the near future.

What does this mean to NLP? Currently, models are often built in an ad hoc process: we might borrow default hyperparameters from previous work and try a few variant architectures, but it is never guaranteed that final trained model is optimal. Automation can introduce rigor in this model-building process. For example, hyperparameter optimization can help NLP *researchers* find reasonably accurate models under limited computation budget, leading to fairer comparison of proposed and baseline methods. Similarly, neural architecture search can help NLP *developers* discover models with the desired speed-accuracy tradeoffs for deployment.

This tutorial will summarize the main AutoML techniques and illustrate how to apply them to improve the NLP model-building process. The goal is to provide the audience with the necessary background to follow and use AutoML research in their own work.

Type of tutorial: Cutting-Edge[2]

2 Target Audience

The tutorial is aimed at NLP researchers and developers who have experience in building deep learning models and are interested in exploring the potential of AutoML in improving their system-building process. Recommended prerequisites are:

- NLP: Familiarity with common neural networks used in the field, especially the Transformer architecture.
- Machine Learning: Understanding of classical supervised learning. Knowledge of Bayesian and Evolutionary methods will be a plus, but not required.
- Programming: Basic experience with training models in deep learning frameworks like PyTorch or Tensorflow.

3 Tutorial Content

Outline: This is a 3-hour tutorial. It is divided into two parts:

1. Overview of major AutoML techniques
   (a) Hyperparameter optimization
   (b) Neural architecture search
2. Application of AutoML to NLP
   (a) Evaluation
   (b) Multiple objectives for deployment
   (c) Cost and carbon footprint
   (d) Software design best practices
   (e) Literature survey

In Part 1, we will focus on two major sub-areas within AutoML: Hyperparameter optimization is the problem of finding optimal hyperparameters, such as learning rate of gradient descent and embedding size of Transformers, based on past training experience. Neural architecture search is the problem of designing the optimal combination of neural network components in a fined-grained fashion. We will summarize these rapidly developing fields and

[1] https://2022.automl.cc
[2] Tutorial Website: https://www.cs.jhu.edu/~kevinduh/a/automl-tutorial-2023/

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics
Tutorial Abstracts, pages 25–26
May 2-4, 2023

explain several representative algorithms, including Bayesian Optimization, Evolutionary Strategies, Population-Based Training, Asynchronous Hyperband, and DARTS.

Part 2 will discuss the practical issues of applying AutoML research to NLP. Questions we will seek to answer include: (a) How do we evaluate AutoML methods on NLP tasks? (b) How can we extend AutoML methods to deployment situations that require multiple objectives, such as inference speed and test accuracy? (c) What is the cost (and carbon footprint) of these methods, and when will it be worthwhile? (d) How should we design our model-building software given a specific computing environment, and what existing tools are available?

Reading List: The tutorial will be self-contained, so there is no required reading list. For a preview of the techniques we will cover, the audience is welcomed to refer to survey papers such as (Feurer and Hutter, 2019; Elsken et al., 2019).

We gave a similar tutorial titled "AutoML for Machine Translation" at AMTA 2022, a machine translation conference. The tutorial slides are available[3]. For the EACL tutorial, we will add discussion on recent uses of AutoML in various NLP applications, ranging from text classification to large language models.

4 Presenters

Kevin Duh is a senior research scientist at the Johns Hopkins University Human Language Technology Center of Excellence (HLTCOE) and an assistant research professor in the Department of Computer Science. His research interests lie at the intersection of NLP and Machine Learning. He has given several conference tutorials on the topics of machine learning and machine translation at, e.g., AMTA 2022, SLTU 2018, IJCNN 2017, DL4MT Winter School 2015.

Xuan Zhang is a Ph.D. student in the Department of Computer Science at Johns Hopkins University (JHU). She performs research in Machine Translation, with specific interests in Sign Language Translation, Hyperparameter Optimization, Curriculum Learning, and Domain Adaptation. She co-presented the AMTA 2022 tutorial on AutoML.

[3] https://www.cs.jhu.edu/~kevinduh/notes/2209-AMTA-AutoMLtutorial.pdf

5 Ethics Statement

There are at least three concerns relevant for responsible use of AutoML technology.

- Energy: Improper use of AutoML may lead to wasted computation in the extreme, e.g. training of thousands of neural models that are eventually discarded. That is why we feel it is important in the tutorial to include a section on cost and carbon footprint.
- Jobs: Some worry that AutoML may reduce the need for data scientists. It is true that some of the black magic involved in hyperparameter and architecture tuning is taken away by AutoML, but we believe that AutoML tools will relieve data scientists to focus on more interesting problems regarding the underlying task, similar to how auto-differentiation tools revolutionized deep learning.
- Bias: If there are underlying biases in the training data, AutoML may output a "optimized" model that exacerbates the bias more so than a manual model-building process. It is thus even more important to check for fairness and bias in an AutoML setup.

References

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Matthias Feurer and Frank Hutter. 2019. *Hyperparameter Optimization*, chapter 1. Springer.