

CS 446 Fall 2019 Project

Performance Predictors for Meta-Learning and AutoML

75 teams4 years ago

[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#)

[Submissions](#) [Late Submissions](#)

Overview

Start

Nov 5, 2019

Close

Dec 13, 2019

Description

keyboard_arrow_up

Overview

As you have learned in class, selecting model architectures and hyperparameters is often a difficult task in applied machine learning. Automated machine learning (AutoML) is a framework of tools designed to automate this model selection process. There is rising evidence in [1,2,3] that performance predictors have the potential to be useful in this process. Performance predictors take a dataset and hyperparameters as input and predict the performance of the resulting trained model. Thus, an effective performance predictor can be used to select good model architectures and hyperparameters, e.g., since the learned predictor is a function, one could create approaches for optimizing it. In this project, your goal will be to train a performance predictor.

Each example in the training set is a network architecture, associated meta-data (details below), and the train and test performance of the associated model when it was trained to convergence. Your task is to develop a supervised machine learning solution that takes a model architecture and hyperparameters and predicts the training and testing performance. Your solution will be evaluated based on the quality of the performance prediction compared to the ground truth.

We encourage you to ensure your methods generalize well e.g. using train-test splits or cross-validation. We will use a blind test set strategy to evaluate how well your methods generalize (details in evaluation). You are free to use any model or strategy for this project (whether covered in class or not).

References

- [1] Neural Architecture Optimization, <https://arxiv.org/abs/1808.07233>
- [2] Progressive Neural Architecture Search, <https://arxiv.org/abs/1712.00559>
- [3] Accelerating Neural Architecture Search using Performance Prediction, <https://arxiv.org/abs/1705.10823>
- [4] A Surprising Linear Relationship Predicts Test Performance in Deep Networks, <https://arxiv.org/abs/1807.09659>
- [5] Predicting the Generalization Gap in Deep Networks with Margin Distributions, <https://arxiv.org/abs/1810.00113>

Learning Goals

- Gain real-world experience with ML, subject to blind test evaluation
- Explore methods for learning with non-vector data e.g. embeddings
- Gain real-world experience with data preprocessing/cleaning, if necessary

Kaggle competition and grading

Your performance will be evaluated via a Kaggle competition. You may sign up for the Kaggle competition here: <https://kaggle.com/c/cs446-fa19>

We will identify you using your NetID. An account created using your NetID is preferable for easy matching during the grading time; however, you may use your existing Kaggle account. Please make sure to fill out the signup form below which includes username details.

You will be restricted to a maximum of 4 submissions per day. The leaderboard shows scores on the validation set. Final grades are based on scores evaluated on a hidden test set with similar distributions as the training and the validation set.

Grading: Your project grade will be determined by thresholds, i.e., you achieve a given grade when your performance is better than the appropriate threshold on the hidden test set. Thresholds will be announced near the end of the project competition. Selected grade thresholds will roughly depend on class performance and our internal baseline tests.

Due date: Competition Ends Dec 12, 11:59 PM

Data

The dataset includes a collection of textual descriptions of neural network model architectures trained on Cifar-10, i.e., inputs "X". The labels "Y" are the final training and testing performance scores of these models. Do not train your own models on cifar-10, you are provided with training and test examples. Your goal will be to build a performance predictor for this data set.

Provided Features

The performance of a neural network often depends on a few factors [4,5], some of which we have selected as features:

- 1.) Architecture description as a string and its hyperparameters
- 2.) The first X epochs of training and validation error history
- 3.) Initialization statistics. Specifically mean, std and L2 norm of the network for each layer before training starts

While your goal is to predict the final train error and final test error, we have interleaved (and identified) examples corresponding to training and testing error. You may consider training a separate performance prediction for the train and test error.

Prediction:

Test labels are submitted as a csv file to the Kaggle site in the format

[id, predicted score]

Note that the training error and test error scores are interleaved and evaluated together.

Computational Resources:

You are free to use any compute resources you like. Microsoft Azure has graciously donated cloud computing credits for each student to use. If you choose to use these, please see the instructions below.

Another great source of compute for the project is **Google Colab**. It is free to use and gives you access to a Tesla K80 GPU. Set up for this is a bit easier than Azure but in order to run your code you must be in a Colab notebook (same as a Jupyter notebook).

After filling up the Google form (informing us you will use Azure), please signup for a free trial of azure: <https://azure.microsoft.com/en-us/free/>. We will add you to the course compute group to access additional computing resources.

Kaggle username declaration and computation signup form [NOTE: FORM CLOSING NOV 28].

Please sign up for computation (if desired) in this form as well: <https://forms.gle/KhkpMAvmJ2c3sVAy7>.

Citation

keyboard_arrow_up

Han Heul Jo, Linling Miao, Patrick Cole, Sanmi, Sriram Sundararaman, Vishnu Jaganathan. (2019). CS 446 Fall 2019 Project. Kaggle. <https://kaggle.com/competitions/cs446-fa19>

