Project Proposal

Title

AutoML-based Pipeline for Deep Learning Tasks

Team Members

- Astha Gupta (ag7982@nyu.edu)
- Gabe Cemaj (gc2728@nyu.edu)

Introduction

In recent years, the idea of AutoML has become more prevalent and gained significant popularity. The concept of automating the process of picking and designing a ML pipeline is very attractive as this is often a very time consuming step. With the growth of ML and DL more and more decisions must be made to pick the correct combination of parameters to build a ML pipeline.

The space from which to pick extends well beyond tradition hyperparameters. Not only must we now pick an architecture, depth, width, activation function, and optimizations, but also, there is a large array of options to pick from with regards to the preprocessing and normalization steps. This large search space would benefit from an automated way of exploring it to find a combination that results in the best performance.

Previous Work

This project aims to combine many ideas from previous work. This paper CFG AutoML [1], proposed a methodology to explore AutoML piepliens in an end to end manner. We hope to take that and expand it with DARTS [2] to explore the architecture space. By combining these two papers we hope to achieve a differentiable space that consists not only of the neural space but the entire pipeline. We also hope to combine this with a standard hyperparameter tuning algorithm to further tune the entire pipeline.

Goal/ Objective

This project aims to string together many of the aspects of AutoML to build a grand unified solution to generate end to end ML pipelines. The idea is to combine neural architecture search, along with hyperparameter tuning and grander exploration of preprocessing combinations to automatically generate a full end to end pipeline for performing deep learning tasks.

Challenges

- Setting a limit to the number of parameters, models and hyperparameters in the search space.
- Trade-off between complexity and number of choices for building the neural architectures.
- Trade-off between computational performance and model performance.
- Trade-off between cost optimization and hyper-parameter tuning.
- Choice of Architecture Optimization algorithm.
- Limited work done in terms of automation of the complete pipeline.

Approach and Implementation

The various steps that are involved in the AutoML pipeline include Dataset Preparation, Feature Engineering, Model Generation and Model Evaluation. The first step would involve defining the search space for the model and specification of the algorithmic and architectural choices to be included at each step of the pipeline.

To evaluate this project we will attempt to generate models to perform over a variety of standard benchmarking tasks in a few different domains. Domain we hope to explore: Computer Vision (ImageNet, COCO, BraTS), NLP (SQuAD), and other domains (Click Logs). We will compare our auto generated models to the SOTA in for each dataset and evaluate the cost benefit analysis of the extra computer needed to generate these automatically. We hope to answer the question of how one might best spend their resources when trying to build a full ML pipeline in a production system (comparing performance vs compute cost vs manual tuning costs).

- Hardware and Software Requirements

Given the computationally heavy nature of most deep learning projects, one or more GPUs may be required during the development of this project. Software requirements include (but are not limited to) Tensorflow, Keras, Pytorch and Numpy libraries.

Planned Demo

We aim to show off a few auto generated models and their performance in comparison to prevalent SOTA models. An important goal of this project is to be able to demonstrate high performance across domains, and to show that we can generate automated end-to-end pipelines for a variety of data types.

References

1. Suilan Estevez-Velarde, Yoan Gutierrez, Andres Montoyo, 'Yudivian Almeida-Cruz, *AutoML strategy based on grammatical evolution: A case study about knowledge discovery from text'* , Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.
2. Hanxiao Liu and Karen Simonyan and Yiming Yang, *DARTS: Differentiable Architecture Search*, arXiv preprint arXiv:1806.09055, 2018.