Building an AutoML System for Fun and Non-profit

Niels Bantilan

Background

OBERLIN
COLLEGE & CONSERVATORY

Columbia University
MAILMAN SCHOOL
OF PUBLIC HEALTH

ML Engineer @

Open Source Projects

Yet Another AutoML Project

Six Levels of AutoML

Level 0: No automation. You code your own ML algorithms. From scratch. In C++.	
Level 1: Use of high-level algorithm APIs. Sklearn, Keras, Pandas, H2O, XGBoost, etc.	
Level 2: Automatic hyperparameter tuning and ensembling. Basic model selection.	
Level 3: Automatic (technical) feature engineering and feature selection, technical data augmentation, GUI.	
Level 4: Automatic domain & problem specific feature engineering, data augmentation, and data integration.	
Level 5: Full ML Automation. Ability to come up with super-human strategies for solving hard ML problems without any input or guidance. Fully conversational interaction with the human user.	

JARVIS

Source: https://medium.com/@tunguz/six-levels-of-auto-ml-a277aa1f0f38

Why Another AutoML Project?

auto-sklearn

GLUON

auto-weka

Google Cloud
AutoML Tables BETA

AutoKeras

Why Another AutoML Project?

For fun 🙃

What goes into building AutoML systems? 🤔

✨✨ Combine some techniques I wanted to learn about ✨✨

MetaRL-Based AutoML

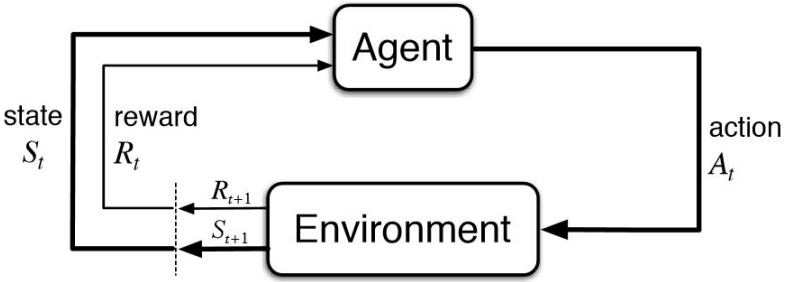
Objective

Proof of concept applying MetaRL algorithms to the **C**ombined **A**lgorithm **S**election and **H**yperparameters optimization (CASH) problem

Metalearning (learning to learn) **+** **Deep Reinforcement Learning** **+** **Automated Supervised Learning**

MetaRL-Based AutoML

Reinforcement Learning API

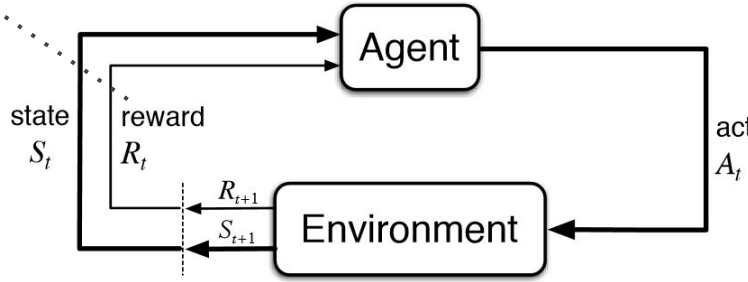


MetaRL-Based AutoML

Validation Performance

e.g. accuracy, AUC, F1-score, R-squared, etc.

PyTorch



scikit learn

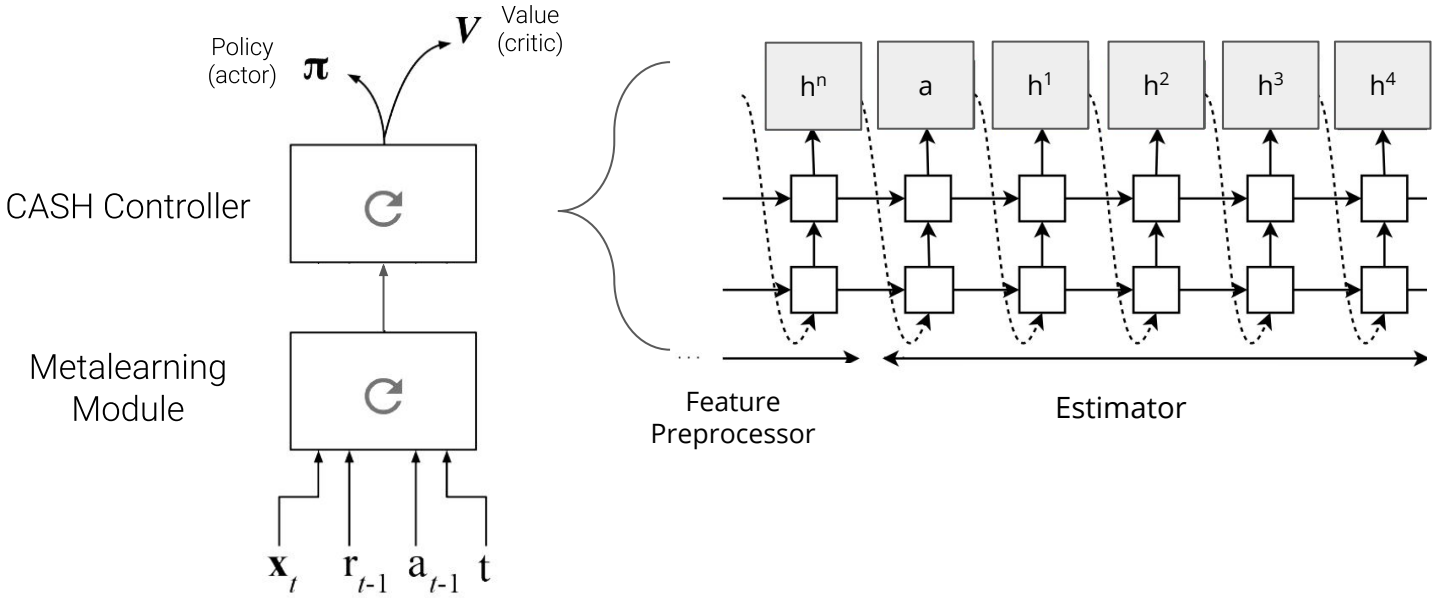
Meta Features

n examples
n features
n categorical
n continuous
n classes
class entropy
...

Task Distribution

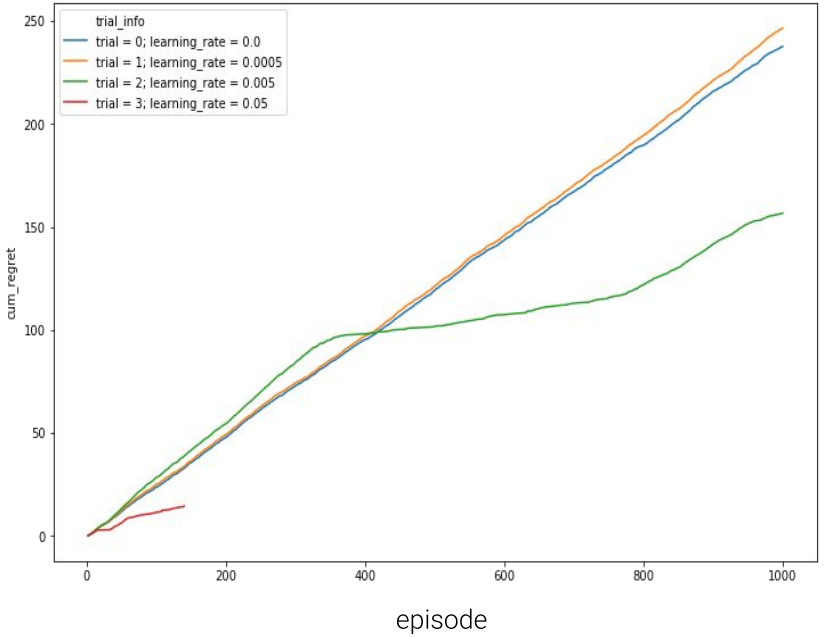
MetaRL-Based AutoML

GRU Advantage Actor Critic

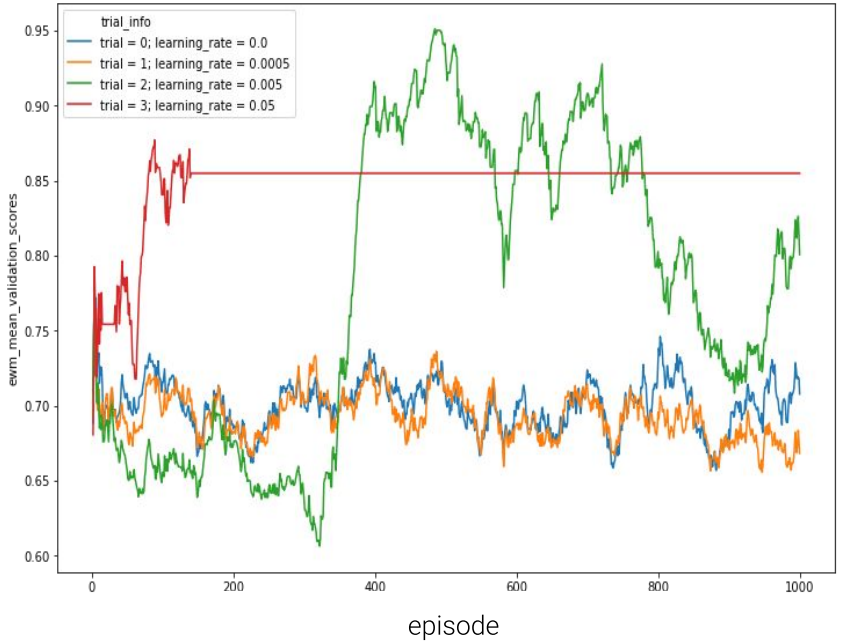


Does it Learn?

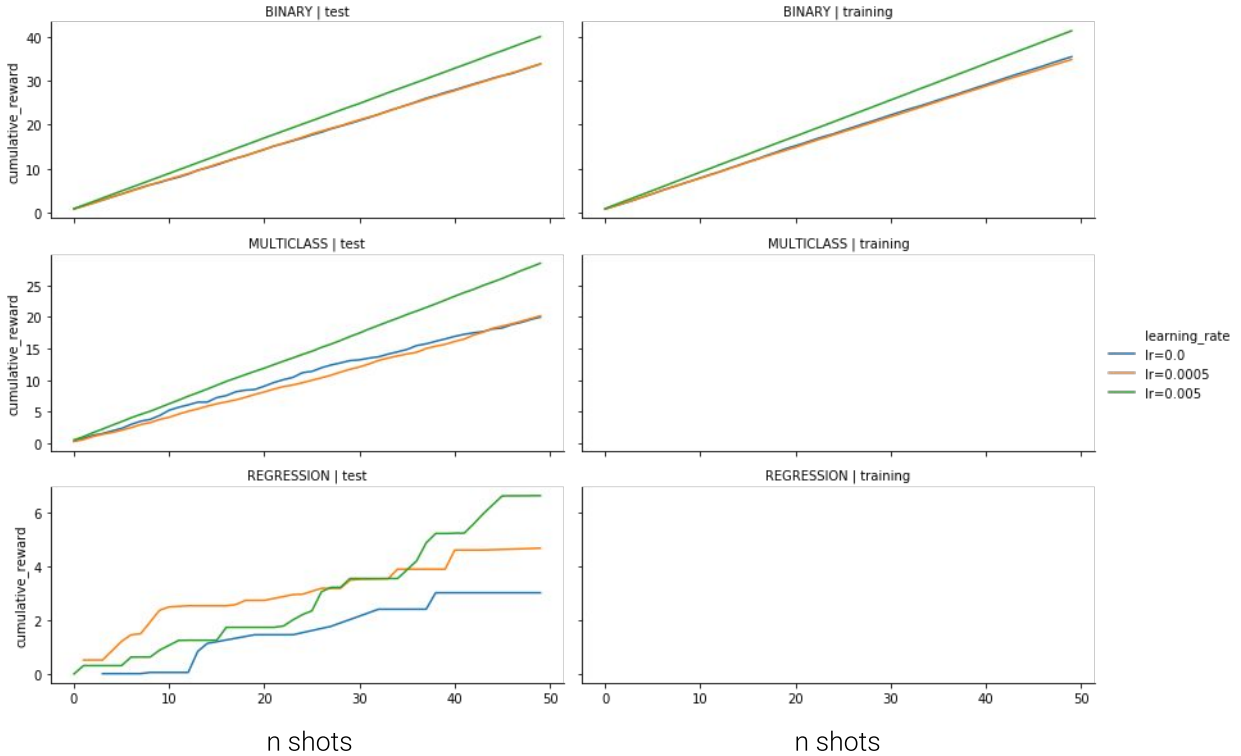
Cumulative Regret



EWM Mean Validation Score



Does it Meta-Learn?



Future Work

Controller-managed model ensembling

Update policy interpreter to use *sklearn.compose.ColumnTransformer*

Richer meta-feature set

Implement benchmark for different MetaRL/AutoML algorithms

Add more task-types (survival models, event model, multi-output models, time-series etc.)

Packaging up `metalearn`, add front-facing API, add documentation

Generalize API to non-sklearn ML algorithms (skorch)

Multi-metric reward functions

References

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., … Botvinick, M. (n.d.). LEARNING TO REINFORCEMENT LEARN.

Zoph, B., & Le, Q. V. (2016). Neural Architecture Search with Reinforcement Learning. Retrieved from http://arxiv.org/abs/1611.01578

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (n.d.). Efficient and Robust Automated Machine Learning. Retrieved from https://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-25566-3_40

Warning: meta-ml is in alpha... no docs or examples

Github: https://github.com/cosmicBboy/ml-research/tree/master/metalearn
Twitter: @cosmicbboy