AutoML Tweakathon

DL-Lab 2018

NeurIPS 2018 AutoML Challenge

The AutoML for Lifelong Machine Learning uses real-world dataset to compare state-of-the-art AutoML systems.

Compared to previous challenges this year they focus on drifting concepts, getting away from the simpler i.i.d. cases.

The challenge consisted of 2 phases:

The **Feedback phase:** uses 5 similar datasets with labels in order to develop and test your code submission

The **AutoML phase** is the blind test phase with on 5 new datasets

The DL-Lab AutoML Tweakathon

- The goal is to achieve the **highest test performance** on a **private** test set
- You **do not** need to develop a full AutoML system since we will not run your code on new data
- We will provide a **training / public and a private test split**.
- For the private test split we will not give you the labels
- At the end we will evaluate your **final test predictions** on the private test split
- The final test performance **will not** determine your grade instead we will evaluate your approach and how you executed it
- We will have a leaderboard (google doc) where you can put your results on the **public test set** in order to compare yourself with other groups

The Data

The challenge provided 4 datasets but we will use only dataset D (but feel free to also exploit the other datasets, e g. for meta-learning)

The dataset contains ~1.5 million data points and 76 features:

- **Categorical Feature**: an integer describing which category the instance belongs to.
- **Numerical Feature**: a real value.
- **Multi-value Categorical Feature**: a set of integers, split by the comma. The size of the set is not fixed and can be different for each instance. For example, topics of an article, words in a title, items bought by a user and so on.
- **Time Feature**: an integer describing time information.

Milestones

1. split up into **groups of 3** people and send us an email with your names and a group name until **8th of January**
2. get familiar with the dataset and split your training data set into a training / validation set
3. fit a simple feed forward neural network as a baseline
4. optimize the hyperparameters and architecture choices
5. try to crack the high score by for example:
    - feature engineering / data preprocessing
    - cross-validation
    - trying our different neural network architectures
    - improving the hyperparameter optimization process
    - ensembles
    - meta-learning
    - ...