dropt-cli

Chiayu Sun and Haopin Wu

Aug 03, 2020

CONTENTS

Dr.Opt is an ML model optimization platform consisting of

- Hyper-parameter optimization service
- Client Python package for service connoction & project control
- Project visualization & analysis via WubUI

Please read *Quickstart* to begin.

CHAPTER

ONE

QUICKSTART

Here we illustrate how one can create and run a Dr.Opt project based on our trial examples.

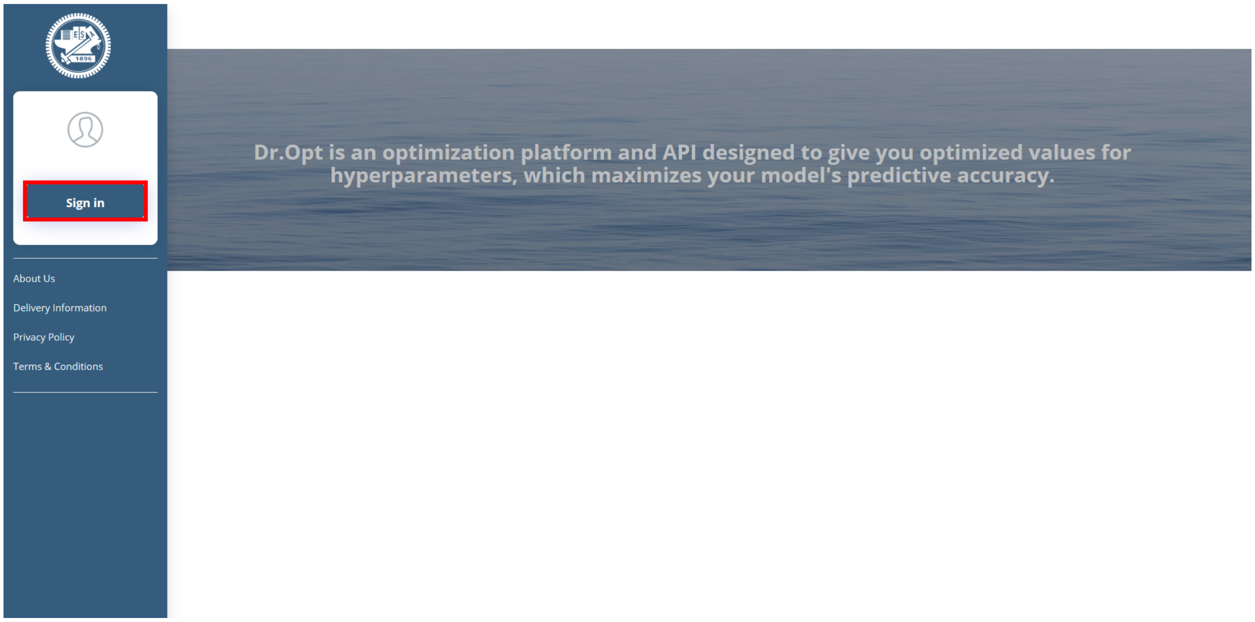
1.1 Prerequisites

Before starting, make sure the following things are installed in your system:

- Python 3.6 or newer
- pip

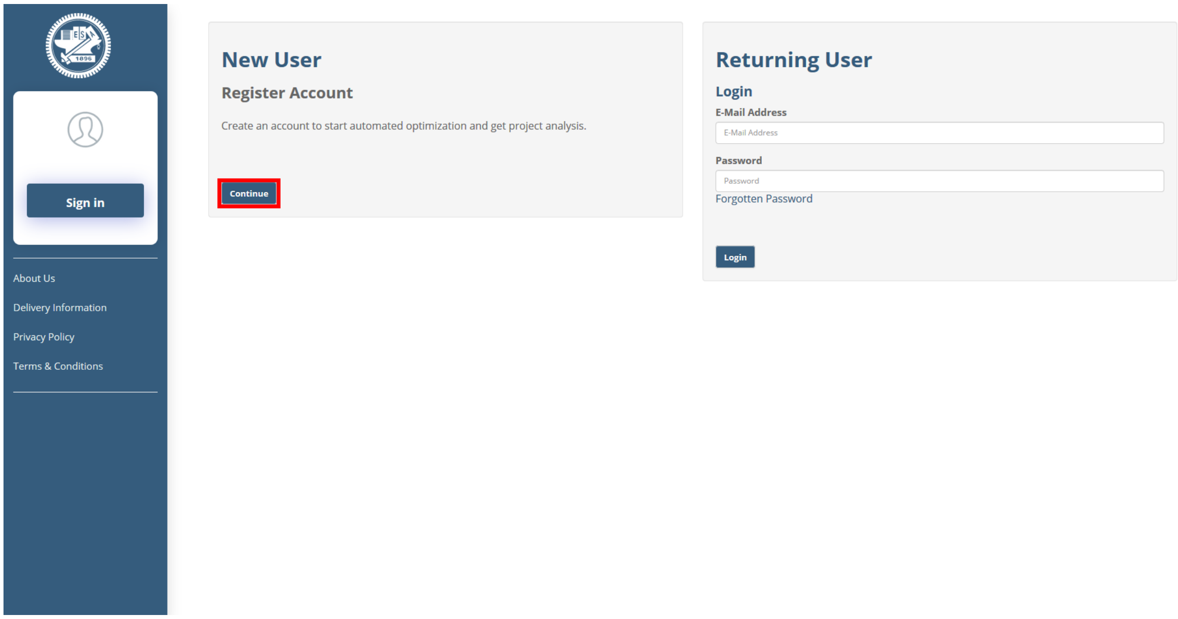
1.2 Registration/Get Access Token

1. Go to Dr.Opt server webpage and click `Sign in`.



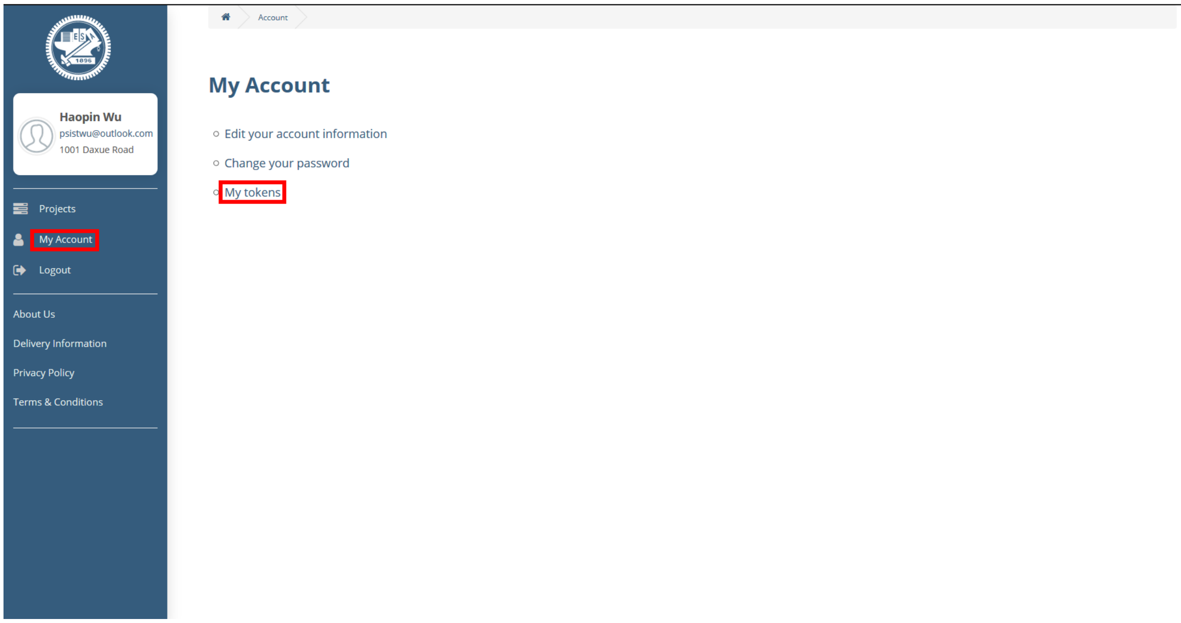
Dr.Opt is an optimization platform and API designed to give you optimized values for hyperparameters, which maximizes your model's predictive accuracy.

2. Click `Continue` and finish the registration. User will be notified via email once the registration is approved.



Create an account to start automated optimization and get project analysis.

3. In `My Account` page click `My tokens`



4. Copy the `api token` for later use.

1.3 Run our examples

1. Download our examples from GitHub:

```
$ git clone https://github.com/GoEdge-ai/dropt-example.git
```

2. Move to the directory of a trial example:

```
$ cd dropt-example/trials/func-eggholder
```

3. Install required Python package:

```
$ pip install -r requirements.txt
```

4. Create and run a new Dr.Opt project with our control script, in which `TEKON` is the access token of your account:

```
$ droptctl -t TOKEN create
```

5. Inspect the result on the Dr.Opt server webpage.

1.4 Your Turn!

We just showed you how to run a our example project. To run your own project, please refer to the following sections:

- *Prepare A Project*
- *Project Control*
- *Project Inspection*

CHAPTER

TWO

PREPARE A PROJECT

In Dr.Opt, a parameter optimization task is called a **project**. Before starting a new project, user needs to prepare a project folder containing all necessary files. Here we will brief how it is done.

2.1 Folder Structure

The following diagram depicts the minimal structure of a project folder:

```
MyProject
├── config.json
└── mymodel.py
```

- `mymodel.py`: A Python file for the model to be tuned
- `config.json`: A JSON file describing the configuration of the project

Note that the names of both files are customizable.

2.2 mymodel.py

The Python file should contain the following function:

```python
def run(params):
    ...

    return metric
```

- The input `params` represents the hyper-parameter configuration for the model.
- The output `metric` measures the performance, such as accuracy or latency, of the model.

Here is an example how `run` should work:

```python
>>> from mymodel import run
>>> params = {
...     "max_depth": 10,
...     "gamma": 0.25,
...     "alpha": 0.5,
...     "learning_rate": 0.001,
...     "subsample": 0.75,
...     "colsample": 0.75
... }
```

(continues on next page)

(continued from previous page)

```
>>> run(params)
0.732
```

2.3 config.json

We consider an example config file:

```
{
    "config": {
        "experimentName": "titanic-xgboost",
        "maxExecDuration": "1h",
        "maxTrialNum": 10,
        "parentProject": "None",
        "model": "model",
        "updatePeriod": 60,
        "tuner": {
            "builtinTunerName": "TPE",
            "classArgs": {"optimize_mode": "maximize"}
        }
    },

    "params": {
        "booster": "gbtree",
        "verbosity": 0,
        "base_score": 0.5,
        "colsample_bylevel": 1,
        "n_estimators": 50,
        "objective": "binary:logistic",
        "max_depth": 5,
        "gamma": 0.2,
        "subsample": 0.8,
        "colsample-bytree": 0.8,
        "lambda": 1,
        "alpha": 0.25,
        "eta": 0.01,
        "min_child_weight": 1.0
    },

    "search_space": {
        "max_depth": {"_type": "randint", "_value": [1, 5]},
        "gamma": {"_type": "uniform", "_value": [0.1, 1.0]},
        "subsample": {"_type": "uniform", "_value": [0.1, 1.0]},
        "colsample_bytree": {"_type": "uniform", "_value": [0.1, 1.0]},
        "alpha": {"_type": "uniform", "_value": [0.1, 1.0]},
        "eta": {"_type": "uniform", "_value": [0.1, 1.0]}
    }
}
```

Three main sections should be included in the JSON file:

2.3.1 config

This section contains config options of the project, which includes:

experimentName (string)

- Name of the project, which will be shown on the Dr.Opt webpage

maxTrialNum (number/integer)

- The maximum number of trials of this project

maxExecDuration (number/integer)

- The expected maximal execution time of the project in *hour* (default: 12 hrs)
- If the experiment time exceeds the **maxExecDuration**, the project state will change to "finish" and inpcomplete suggestions will be discarded.

parentProject (string)

- The parent project of the current one
- (Coming in the future) The newly created project can inherent the properties of the **parentProject**
- Just set to `None` for now

model (string)

- The Python file of the model to be tuned (without file extension)

mode (string)

- The optimization mode of the project. It can be `max` (default) or `min`.

updatePeriod (number/integer)

- The update period of the webpage (in *second*)

tuner (object)

- parameter search algorithm
- See page *Advisor* for detail

2.3.2 params

This section consists of default hyper-parameter of the model. Its format should conincide with that of input of `run` (see section *mymodel.py*).

2.3.3 search_space

This section describes the search space. Please read page *Search Space* for detail.

CHAPTER THREE

PROJECT CONTROL

Once a project folder is prepared, user can manage the project via the project control droptctl, which is included in the Dr.Opt client package dropt-cli. To install it, simply run

```
$ pip install dropt-cli
```

3.1 Basic Syntax

Here is the basic syntax of droptctl:

```
$ droptctl -s ADDRESS -p PORT -t TOKEN CMD
```

- ADDRESS and PORT indicate which Dr.Opt server droptctl will connect to. If not given, default Dr.Opt server will be used.
- TOKEN is the unique identification of each user. It can be found on one's own **My account** page.
- CMD is the command to be sent. Currently, two commands are supported:
  - create
  - resume

3.2 Create

```
$ droptctl -t TOKEN create -c CONFIG_FILE
```

Create and run a new project based on config file CONFIG_FILE. The default config file is config.json.

3.3 Resume

User may resume a project if interrupted.

```
$ droptctl -t TOKEN resume
```

A prompt will show all ongoing projects and user selects one to resume.

```
? Which project would you like to resume?  (Use arrow keys)
  [project 120: dummy] progress: 2/100 (created at 2020-05-08T15:46:54.059234+00:00)
  [project 119: dummy] progress: 4/100 (created at 2020-05-08T15:46:26.824813+00:00)
» [Project 75: func-eggholder] progress: 3/1000 (created at 2020-06-29T01:03:45.
↪065417+00:00)
  [Project 76: func-eggholder] progress: 2/1000 (created at 2020-06-29T01:03:55.
↪605235+00:00)
```

CHAPTER

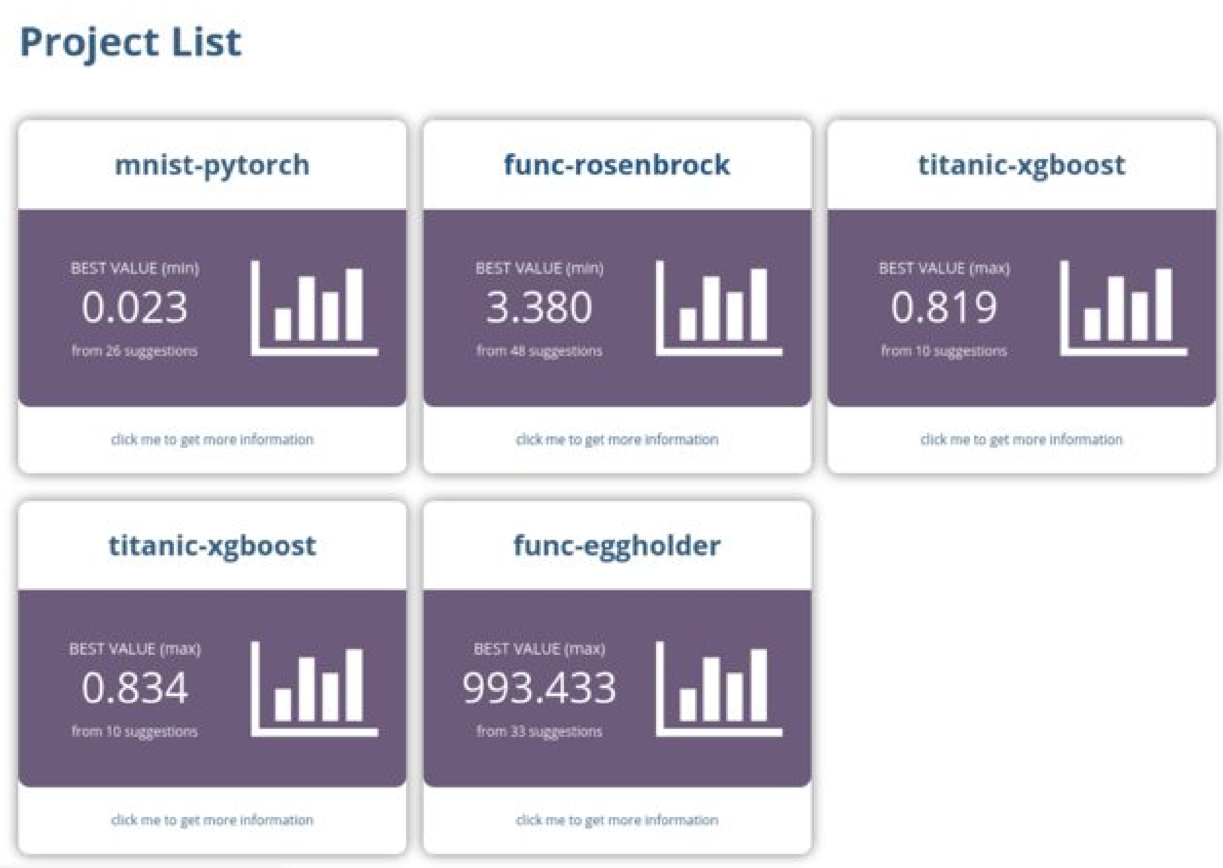
FOUR

PROJECT INSPECTION

Dr.Opt provides multiple visualization tools for analyzing projects. This page aims to introduce how to inspect Dr.Opt projects.

4.1 Project List

After login to the user account, the project list will first show up. It lists each project by the project name, the number of suggestions, and the best result.



Project List

mnist-pytorch

BEST VALUE (min)
0.023
from 26 suggestions

click me to get more information

func-rosenbrock

BEST VALUE (min)
3.380
from 48 suggestions

click me to get more information

titanic-xgboost

BEST VALUE (max)
0.819
from 10 suggestions

click me to get more information

titanic-xgboost

BEST VALUE (max)
0.834
from 10 suggestions

click me to get more information

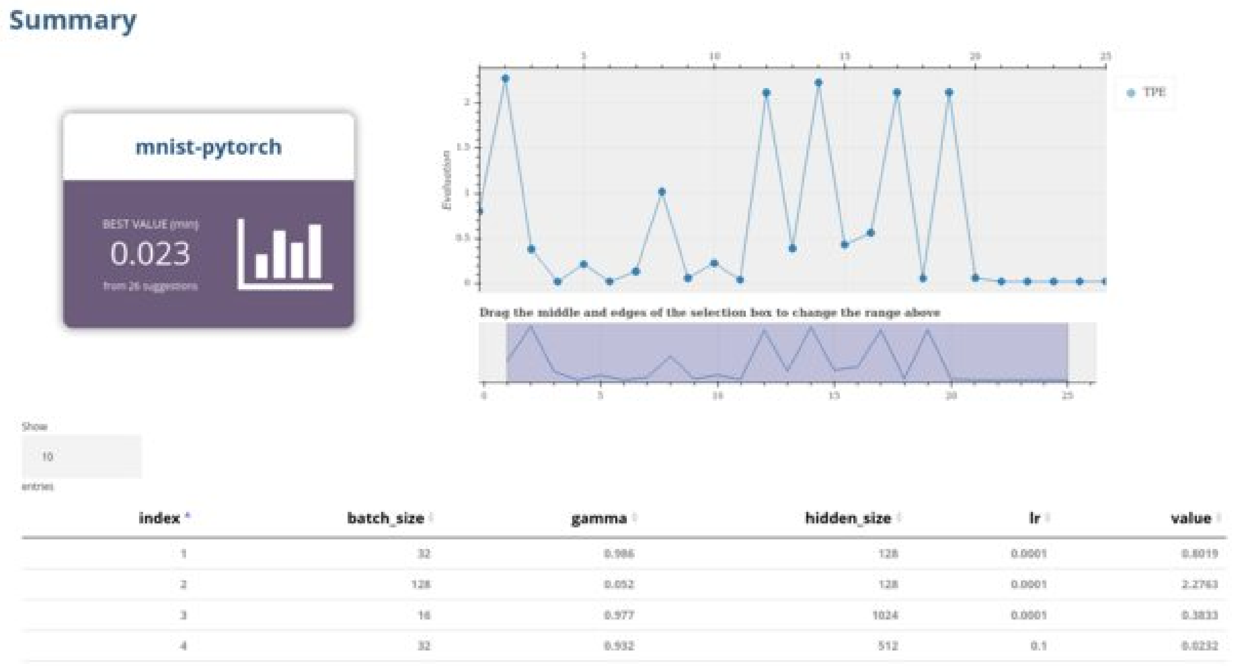
func-eggholder

BEST VALUE (max)
993.433
from 33 suggestions

click me to get more information

The project page can be opened by clicking the project block.

4.2 Summary



Summary

mnist-pytorch

BEST VALUE (min)

0.023

from 26 suggestions

5
10
15
20
25
2
1.5
1
0.5
0
Evaluation
TPE
Drag the middle and edges of the selection box to change the range above
0
5
10
15
20
25

Show

10

entries

index	batch_size	gamma	hidden_size	lr	value
1	32	0.986	128	0.0001	0.8019
2	128	0.052	128	0.0001	2.2763
3	16	0.977	1024	0.0001	0.3833
4	32	0.932	512	0.1	0.0232

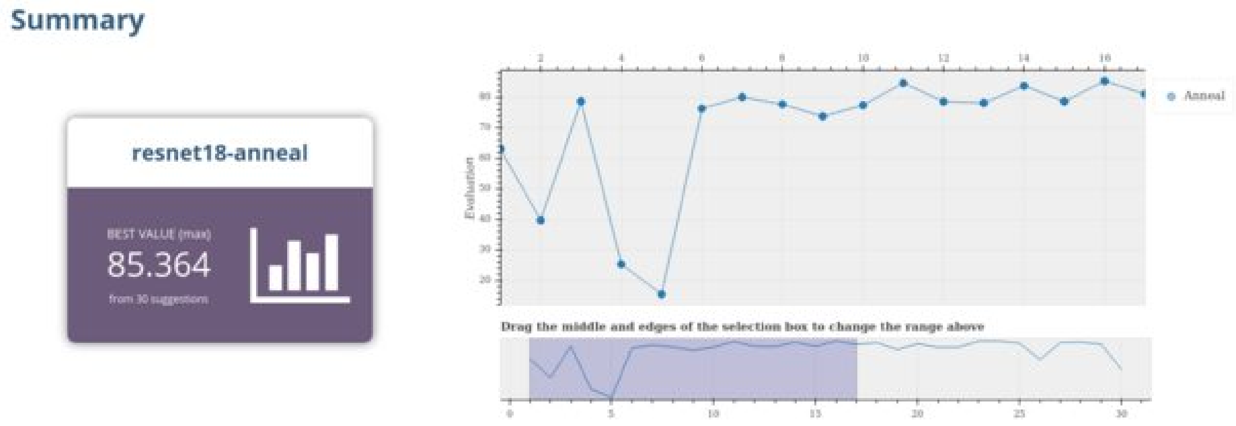
The summary page composes of three main parts:

- project card (project name & the best result)
- optimizing progress chart
- suggestion table

4.2.1 Project Card

Similar to the blocks of the project list, the project card indicates the name of the project and the best optimization result. The best value depends on the mode (minimize or maximize) set in the project config.

4.2.2 Optimizing Progress Chart



Summary

resnet18-anneal

BEST VALUE (max)

85.364

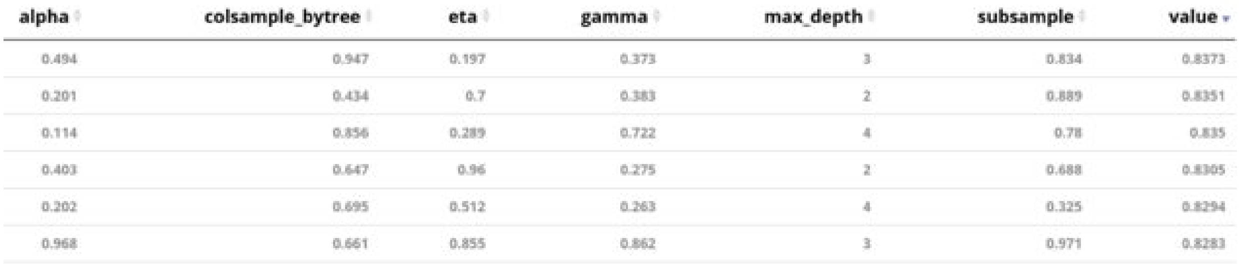
from 30 suggestions

2
4
6
8
10
12
14
16
80
70
60
50
40
30
20
Evaluation
Anneal
Drag the middle and edges of the selection box to change the range above
0
5
10
15
20
25
30

The chart in the upper-right corner plots the objective value of each trial. Users can evaluate the optimization progress by the line chart (e.g., Does it keep improving? Do the objective values fall in a certain range?)

4.2.3 Suggestions Table

The table below collects the suggestion values of a project. Each row represents the suggestion of a trial.



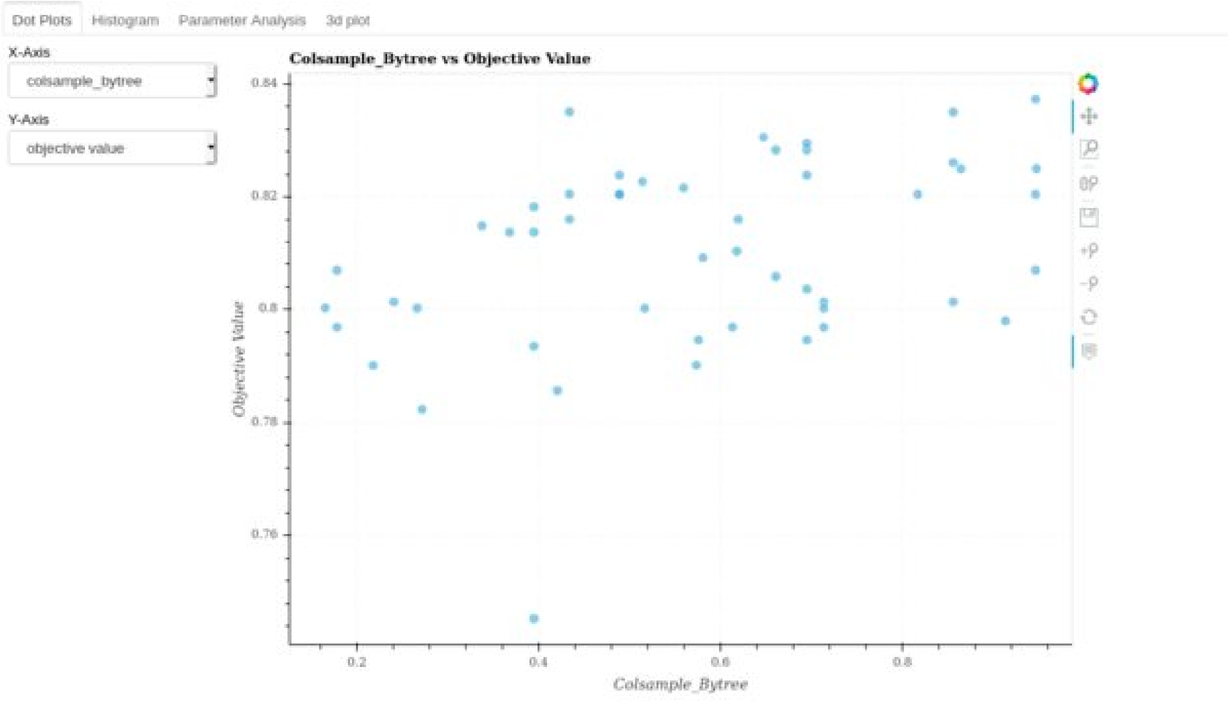
alpha	colsample_bytree	eta	gamma	max_depth	subsample	value
0.494	0.947	0.197	0.373	3	0.834	0.8373
0.201	0.434	0.7	0.383	2	0.889	0.8351
0.114	0.856	0.289	0.722	4	0.78	0.835
0.403	0.647	0.96	0.275	2	0.688	0.8305
0.202	0.695	0.512	0.263	4	0.325	0.8294
0.968	0.661	0.855	0.862	3	0.971	0.8283

The table can be sorted by clicking the column name. Take the following picture as an example, the user can find out the suggestions that obtain the best results by sorting by the object value.

4.3 Analysis

The analysis page contains multiple charts to visualize the suggestion results. It comprises four tabs:

4.3.1 Dot Plots

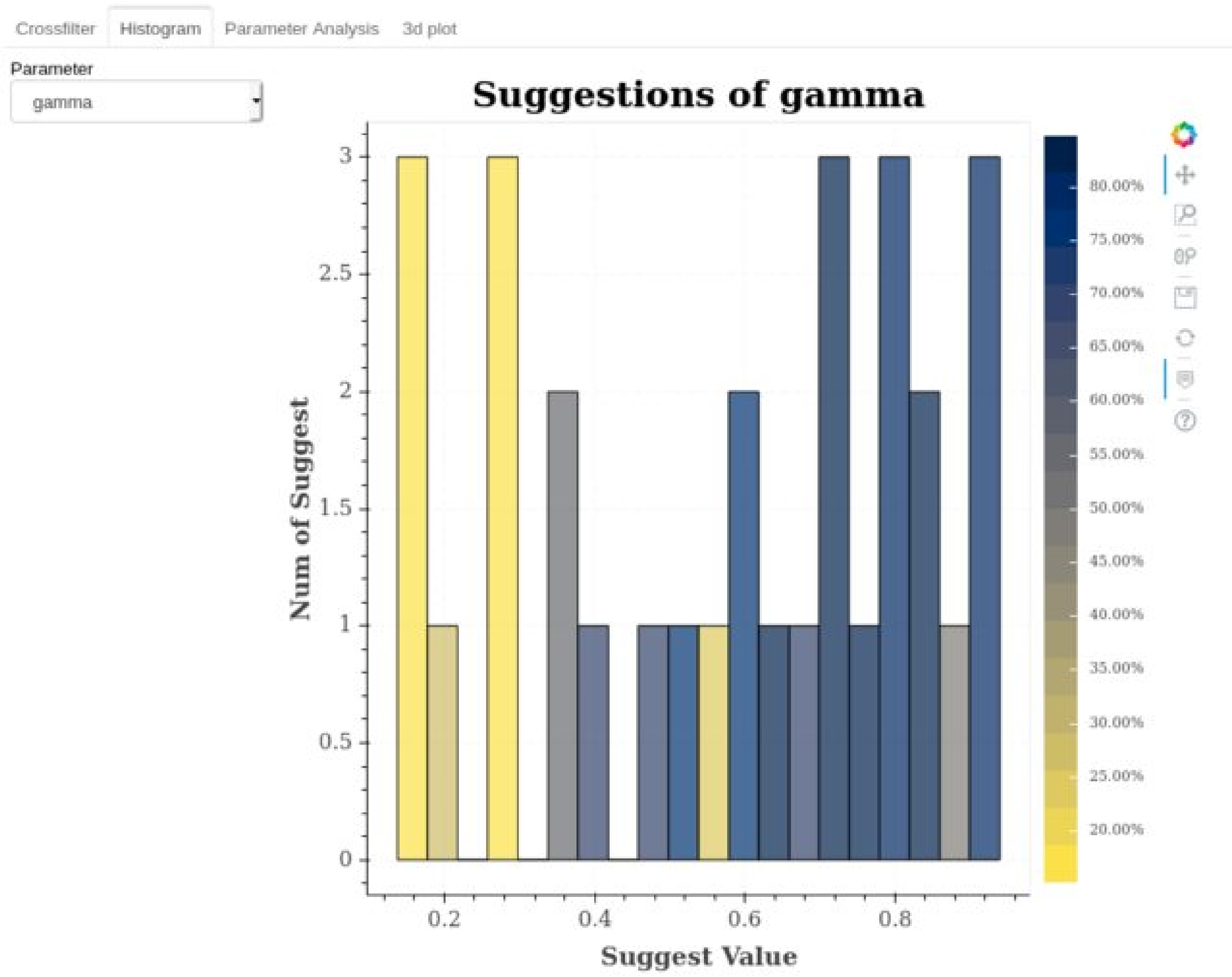


X-Axis
colsample_bytree

Y-Axis
objective value

Dot plots demonstrate the distribution & performance of each hyper-parameter. The plotted data of both the x-axis and the y-axis can be switched by the drop-down list.

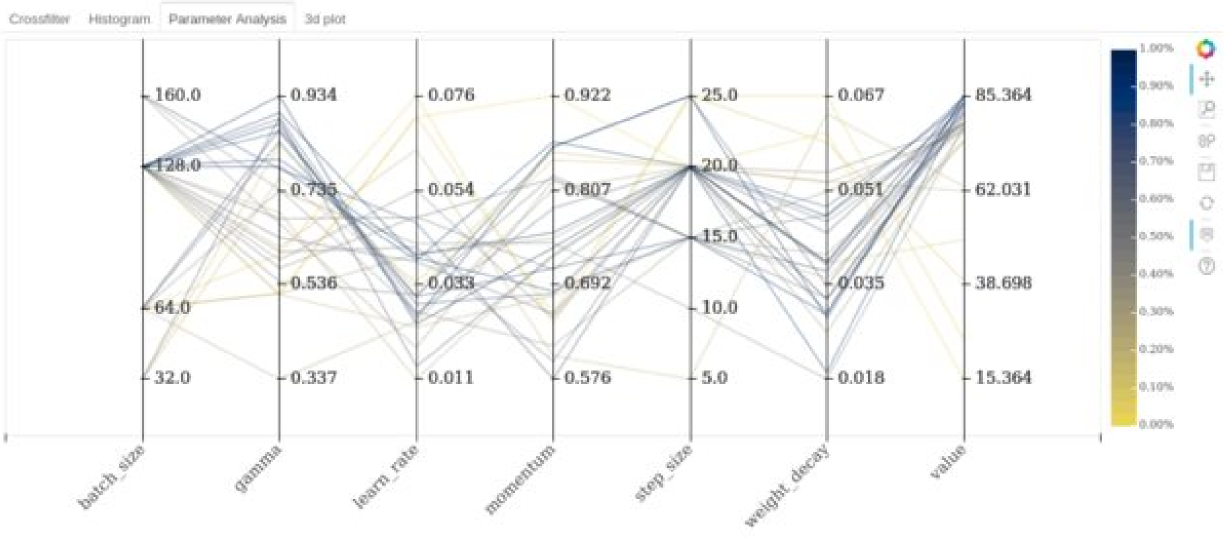
4.3.2 Histograms



Histograms illustrate the distribution of the suggested hyper-parameter values. That is, the range that the tuning algorithm suggests the most. Besides the number of suggestions, the performance is presented by the color-mapping.

4.3.3 Parameter Analysis

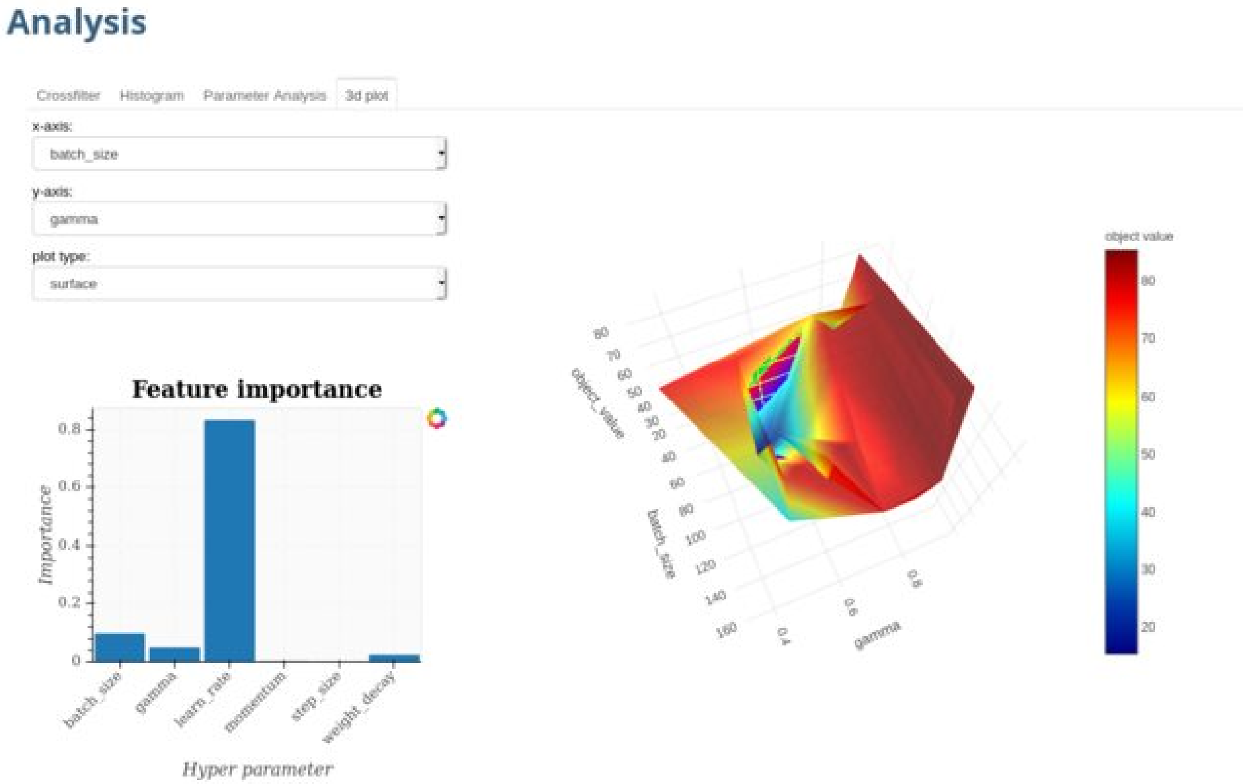
The parameter analysis page illustrates a parallel coordinates plot. It is commonly used for visualizing & comparing many variables together.



As for parameter optimization, it is practical for analyzing the well-performed range and the relationship between each parameter.

4.3.4 3D Plot & Feature Importance

The tab “3d plot” contains a 3-dimension surface plotter and a feature importance chart.

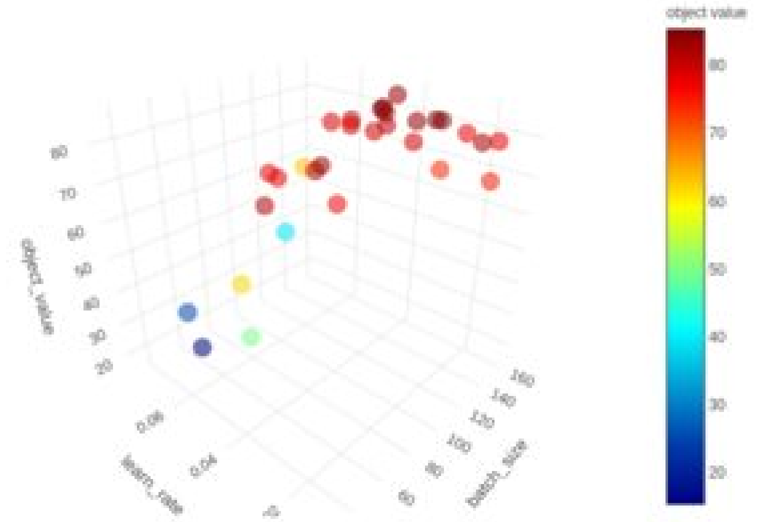

Analysis

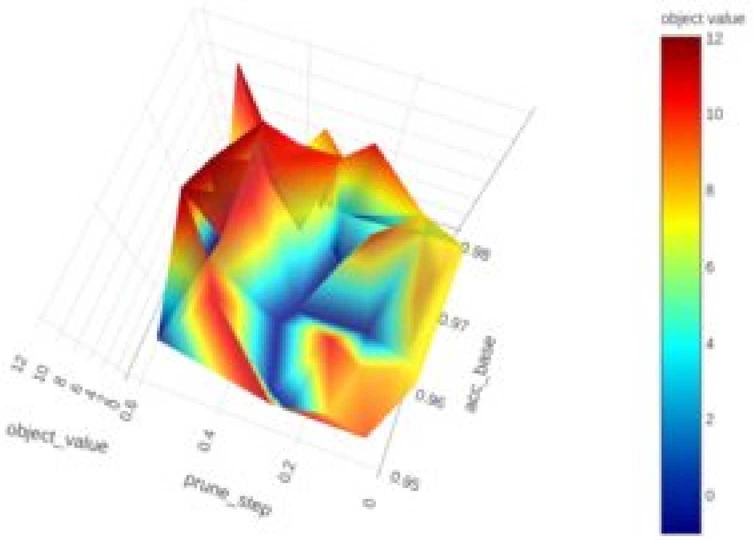
x-axis: batch_size

y-axis: gamma

plot type: surface

The 3D plotter can visualize the data by a surface or a scatter plot.

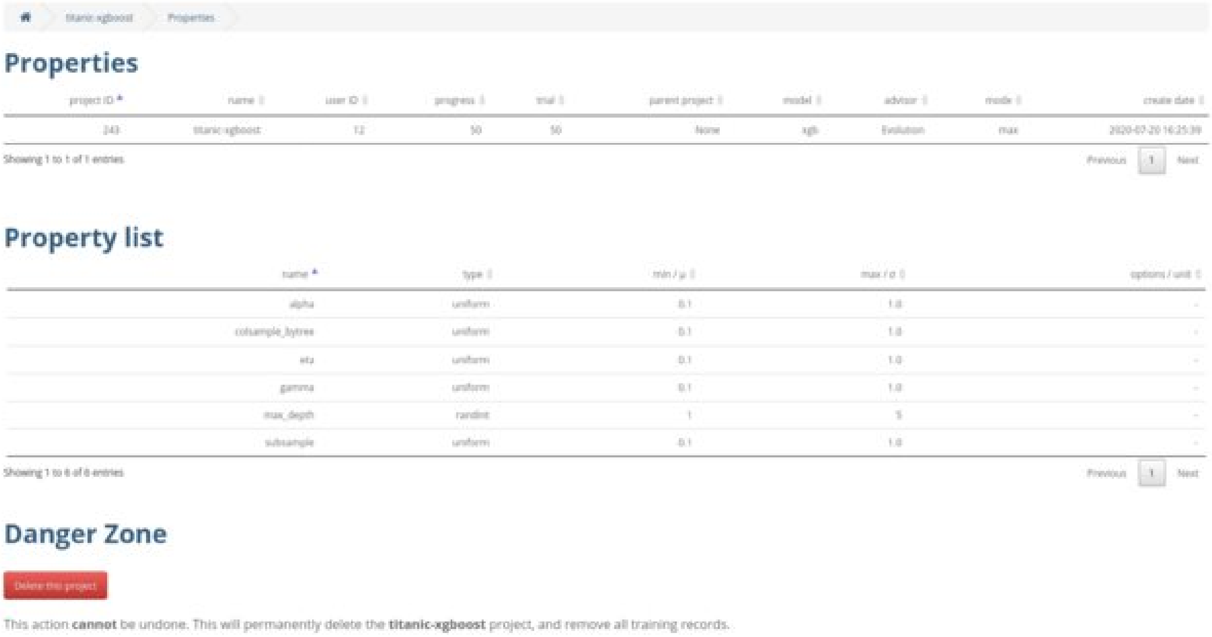




As for the tasks that have a constraint search space (e.g., some hyper-parameter combinations may directly fail), this kind of visualization may be useful to find out the valid range of value.

4.4 Properties

The detailed configuration of a project can be found on the properties page, which can be accessed by the sidebar.



Properties

project ID	name	user ID	progress	trial	parent project	model	advisor	mode	create date
243	titanic-xgboost	12	50	50	None	xgb	Evolution	max	2020-07-20 16:25:39

Showing 1 to 1 of 1 entries

Property list

name	type	min / μ	max / σ	options / unit
alpha	uniform	0.1	1.0	-
colsample_bytree	uniform	0.1	1.0	-
eta	uniform	0.1	1.0	-
gamma	uniform	0.1	1.0	-
max_depth	randint	1	5	-
subsample	uniform	0.1	1.0	-

Showing 1 to 6 of 6 entries

Danger Zone

This action **cannot** be undone. This will permanently delete the **titanic-xgboost** project, and remove all training records.

Through the properties page, the user can confirm the project settings, view the parameter search space, and delete the project.

4.5 Suggestions

The suggestions page lists the full suggestion history. It is similar to the table of the summary page, but the main difference that the suggestions page does not conclude the progress plot. We plan to add more features for this page and please look forward to our update!

CHAPTER

FIVE

ADVISOR & SEARCH SPACE

5.1 Advisor

Dr.Opt supports the following advisors:

Advisor	Description
Anneal	Begin with sampling from the prior and tends over time to sample from points closer and closer to the best ones observed.
Evolution	Randomly initialize a population based on the search space. It chooses better ones and does some mutation on them to get the next generation. Evolution may require many trials to work (ref).
Gaussian Process (coming soon)	A sequential model-based optimization (SMBO) approach that uses Gaussian Process as the surrogate (ref).
Grid Search	Perform an exhaustive search in a specified subset of the parameter space.
Random	Randomly select hyper-parameter combinations in the search space. Researches show that it might be surprisingly effective (ref).
TPE	A sequential model-based optimization (SMBO) approach that uses Tree-structured Parzen Estimators as the surrogate (ref).

5.1.1 Anneal

Arguments

optimize_mode `maximize` (default) or `minimize`

Example

```
{
    "builtinTunerName": "Anneal",
    "classArgs": {"optimize_mode": "maximize"}
}
```

5.1.2 Evolution

Arguments

optimize_mode `maximize` (default) or `minimize`

population_size The initial size of the population

Example

```
{
    "builtinTunerName": "Evolution",
    "classArgs": {
        "optimize_mode": "maximize",
        "population_size": 100
    }
}
```

5.1.3 Grid Search

Arguments

None

Config example

```
{
    "builtinTunerName": "GridSearch"
}
```

5.1.4 Random

Arguments

None

Config example

```
{
    "builtinTunerName": "Random"
}
```

5.1.5 TPE

Arguments

optimize_mode `maximize` (default) or `minimize`

Example

```
{
    "tuner": {
        "builtinTunerName": "TPE",
        "classArgs": {"optimize_mode": "maximize"}
    }
}
```

5.2 Search Space

Each parameter to search is assigned with certain space type. Dr.Opt currently supports the following search space types:

5.2.1 choice

Choose from a list of available options.

Format A list of of numbers or strings, e.g., [0.1, 0.01, 0.001, 0.0001] or [“Adam”, “SGD”, “Adadelta”]

Example

```
{
    "learning_rate": {
        "_type": "choice",
        "_value": [0.1, 0.01, 0.001, 0.0001]
    }
}
```

5.2.2 randint

Choose a random integer within an interval.

Format [*lower_bound* (inclusive), *upper_bound* (exclusive)]

Example

```
{
    "batch_size": {
        "_type": "randint",
        "_value": [8, 65]
    }
}
```

5.2.3 uniform

Choose a number randomly from a uniform distribution on an interval.

Format [*lower_bound* (inclusive), *upper_bound* (exclusive)]

Example

```
{
    "dropout_rate": {
        "_type": "uniform",
        "_value": [0.1, 0.5]
    }
}
```

5.2.4 quniform

Choose a number randomly from an interval descretized by a fixed step size.

Format [*lower_bound* (inclusive), *upper_bound* (exclusive), *step*]

Example

```
{
    "input_size": {
        "_type": "quniform",
        "_value": [224, 417, 32]
    }
}
```

Note: In this example, the possible values are: 224, 256, 288, 320, . . . , 384, 416.

5.2.5 normal

Choose a number randomly from a normal discribution with prescribed mean (μ) and standard deviation (σ).

Format [μ, σ]

Example

```
{
    "dropout_rate": {
        "_type": "normal",
        "_value": [0.5, 0.1]
    }
}
```

5.3 Support of Tuners/Search Space Types

	choice	randint	uniform	quniform	normal
Anneal	v	v	v	v	v
Evolution	v	v	v	v	v
Grid Search	v	v		v	
Random	v	v	v	v	v
TPE	v	v	v	v	v

CHAPTER

SIX

AUTHORS

6.1 Core Development Team

- Chiayu Sun <chiayusun.cs08g@nctu.edu.tw>
- Haopin Wu <psistwu@outlook.com>

CHAPTER SEVEN

LICENSE

```
MIT License

Copyright (c) 2020 Haopin Wu
Copyright (c) 2019-2020 Chiayu Sun
Copyright (c) 2014-2015 SigOpt Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

The modules in namespace `dropt.client` (`endpoint.py`, `exception.py`, `interface.py`, `objects.py`, `requestor.py`, `resource.py`) are modified from sigopt-python, which is under the MIT license:

```
The MIT License (MIT)

Copyright (c) 2014-2015 SigOpt Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
```

(continues on next page)

(continued from previous page)

```
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

The implementation of tuner algorithms are modified from NNI (Neural Network Intelligence), which is under the MIT license:

```
Copyright (c) Microsoft Corporation.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED *AS IS*, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

CHAPTER

EIGHT

CHANGELOG

dropt-cli uses Semantic Versioning.

CHAPTER NINE

LINKS

Dr.Opt service cloud https://dropt.goedge.ai

dropt-cli repo https://github.com/GoEdge-ai/dropt-cli

dropt-example repo https://github.com/GoEdge-ai/dropt-example