advisor Documentation

tobe

Nov 11, 2019

Contents:

CHAPTER 1

Advisor

1.1

Advisor is the hyper parameters tuning system for black box optimization.

It is the open-source implementation of Google Vizier with these features.

- Easy to use with API, SDK, WEB and CLI
- Support abstractions of Study and Trial
- Included search and early stop algorithms
- Recommend parameters with trained model
- Same programming interfaces as Google Vizier
- Command-line tool just like Microsoft NNI.

1.2

- [x] Grid Search
- [x] Random Search
- [x] Bayesian Optimization
- [x] TPE(Hyperopt)
- [x] Random Search(Hyperopt)
- [x] Simulate Anneal(Hyperopt)
- [x] Quasi Random(Chocolate)

- [x] Grid Search(Chocolate)
- [x] Random Search(Chocolate)
- [x] Bayes(Chocolate)
- [x] CMAES(Chocolate)
- [x] MOCMAES(Chocolate)
- [ ] SMAC Algorithm
- [x] Early Stop First Trial Algorithm
- [x] Early Stop Descending Algorithm
- [ ] Performance Curve Stop Algorithm

CHAPTER 2

2.1 Pip

```
pip install advisor
```

2.2 From Source

```
git clone git@github.com:tobegit3hub/advisor.git

cd ./advisor/advisor_client/

python ./setup.py install
```

2.3 Docker

```
docker run -d -p 8000:8000 tobegit3hub/advisor
```

2.4 Docker Compose

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/docker-compose.yml

docker-compose up -d
```

2.5 Kubernetes

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/kubernetes_advisor.
↪yaml

kubectl create -f ./kubernetes_advisor.yaml
```

CHAPTER 3

pip

```
pip install advisor
```

```
advisor_admin server start
```

http://127.0.0.1:8000

```
git clone --depth 1 https://github.com/tobegit3hub/advisor.git && cd ./advisor/

advisor run -f ./advisor_client/examples/python_function/config.json
```

```
advisor study describe -s demo
```

CHAPTER 4

4.1

```
advisor_admin server start
```

4.2 Docker

```
docker run -d -p 8000:8000 tobegit3hub/advisor
```

4.3 Docker Compose

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/docker-compose.yml

docker-compose up -d
```

4.4 Kubernetes

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/kubernetes_advisor.
↪yaml

kubectl create -f ./kubernetes_advisor.yaml
```

4.5

```
git clone --depth 1 https://github.com/tobegit3hub/advisor.git && cd ./advisor/

pip install -r ./requirements.txt

./manage.py migrate

./manage.py runserver 0.0.0.0:8000
```

CHAPTER 5

5.1

```
advisor_admin server start
```

5.2

```
advisor_admin server stop
```

5.3

```
advisor run -f ./advisor_client/examples/python_function/config.json
```

5.4 Study

```
advisor study list
```

5.5 Study

```
advisor study describe -s demo
```

5.6 Trial

```
advisor trials list
```

CHAPTER 6

SDK

6.1

```
client = AdvisorClient()
```

6.2 Study

```
study_configuration = {
  "goal":
  "MINIMIZE",
  "randomInitTrials":
  1,
  "maxTrials":
  5,
  "maxParallelTrials":
  1,
  "params": [
      {
          "parameterName": "gamma",
          "type": "DOUBLE",
          "minValue": 0.001,
          "maxValue": 0.01,
          "feasiblePoints": "",
          "scalingType": "LINEAR"
      },
      {
          "parameterName": "C",
          "type": "DOUBLE",
          "minValue": 0.5,
          "maxValue": 1.0,
          "feasiblePoints": "",
```

(continues on next page)

(continued from previous page)

```
            "scalingType": "LINEAR"
        },
        {
            "parameterName": "kernel",
            "type": "CATEGORICAL",
            "minValue": 0,
            "maxValue": 0,
            "feasiblePoints": "linear, poly, rbf, sigmoid, precomputed",
            "scalingType": "LINEAR"
        },
        {
            "parameterName": "coef0",
            "type": "DOUBLE",
            "minValue": 0.0,
            "maxValue": 0.5,
            "feasiblePoints": "",
            "scalingType": "LINEAR"
        },
    ]
}
study = client.create_study("Study", study_configuration,
                            "BayesianOptimization")
```

6.3 Study

```
study = client.get_study_by_id(6)
```

6.4 Trial

```
trials = client.get_suggestions(study.id, 3)
```

6.5

```
parameter_value_dicts = []
for trial in trials:
  parameter_value_dict = json.loads(trial.parameter_values)
  print("The suggested parameters: {}".format(parameter_value_dict))
  parameter_value_dicts.append(parameter_value_dict)
```

6.6

```
metrics = []
for i in range(len(trials)):
  metric = train_function(**parameter_value_dicts[i])
  metrics.append(metric)
```

6.7 Trial

```
for i in range(len(trials)):
  trial = trials[i]
  client.complete_trial_with_one_metric(trial, metrics[i])
is_done = client.is_study_done(study.id)
best_trial = client.get_best_trial(study.id)
print("The study: {}, best trial: {}".format(study, best_trial))
```

CHAPTER 7

7.1 YAML

```yaml
name: "demo"
algorithm: "BayesianOptimization"
trialNumber: 10
path: "./advisor_client/examples/python_function/"
command: "./min_function.py"
search_space:
  goal: "MINIMIZE"
  randomInitTrials: 3
  params:
    - parameterName: "x"
      type: "DOUBLE"
      minValue: -10.0
      maxValue: 10.0
```

7.2 JSON

```json
{
  "name": "demo",
  "algorithm": "BayesianOptimization",
  "trialNumber": 10,
  "concurrency": 1,
  "path": "./advisor_client/examples/python_function/",
  "command": "./min_function.py",
  "search_space": {
      "goal": "MINIMIZE",
      "randomInitTrials": 3,
      "params": [
          {
              "parameterName": "x",
              "type": "DOUBLE",
```

(continues on next page)

(continued from previous page)

```
                "minValue": -10.0,
                "maxValue": 10.0,
                "scalingType": "LINEAR"
            }
        ]
    }
}
```