

12

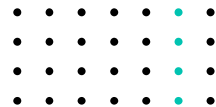
# 베이지안 최적화

2. 패키지 소개



# 패키지 소개

## 소개



<https://github.com/fmfn/BayesianOptimization>

- 페르난도 노게이라(Fernando Nogueira)에 의해 2014년에 개발된 파이썬 패키지
- 베이지안 확률에 대한 개념 없이도 쉽게 사용할 수 있다.

## Bayesian Optimization

Travis CI failing codecov 98% pypi v1.2.0

Pure Python implementation of bayesian global optimization with gaussian processes.

- PyPI (pip):

```
$ pip install bayesian-optimization
```



# 패키지 소개

## 프로그램 기본 구조



```
...
@Misc{
  author = {Fernando Nogueira},
  title = {{Bayesian Optimization}: Open source constrained
          global optimization tool for {Python}},
  year = {2014--},
  url = " https://github.com/fmfn/BayesianOptimization"
}
...

from bayes_opt import BayesianOptimization

def black_box_function(x, y):
    return -x ** 2 - (y - 1) ** 2 + 1

pbounds = {'x': (2, 4), 'y': (-3, 3)}

optimizer = BayesianOptimization(
    f=black_box_function,
    pbounds=pbounds,
    random_state=1,
)

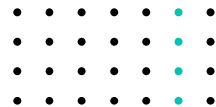
optimizer.maximize(
    init_points=2,
    n_iter=3,
)
```

iter	target	x	y
1	-7.135	2.834	1.322
2	-7.78	2.0	-1.186
3	-7.11	2.218	-0.7867
4	-12.4	3.66	0.9608
5	-6.999	2.23	-0.7392



# 패키지 소개

## 클래스 변수



### max 클래스 변수

- 가장 성능이 좋은 목표함수의 반환값과 파라미터를 기록하고 있는 클래스 변수

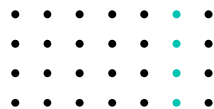
```
...
@Misc{
  author = {Fernando Nogueira},
  title = {{Bayesian Optimization}: Open source constrained
          global optimization tool for {Python}},
  year = {2014--},
  url = " https://github.com/fmfn/BayesianOptimization"
}
...
print(optimizer.max)

{'target': -6.999472814518675, 'params': {'x': 2.2303920156083024, 'y': -0.7392021938893159}}
```



# 패키지 소개

## 클래스 변수



### res 클래스 변수

```
...
@Misc{
    author = {Fernando Nogueira},
    title = {{Bayesian Optimization}: Open source constrained
            global optimization tool for {Python}},
    year = {2014--},
    url = " https://github.com/fmfn/BayesianOptimization"
}
...

for i, res in enumerate(optimizer.res):
    print("Iteration {}: {}".format(i, res))
```

```
Iteration 0:
    {'target': -7.135455292718879, 'params': {'x': 2.8340440094051482, 'y': 1.3219469606529488}}
Iteration 1:
    {'target': -7.779531005607566, 'params': {'x': 2.0002287496346898, 'y': -1.1860045642089614}}
Iteration 2:
    {'target': -7.109925819441113, 'params': {'x': 2.2175526295255183, 'y': -0.7867249801593896}}
Iteration 3:
    {'target': -12.397162416009818, 'params': {'x': 3.660003815774634, 'y': 0.9608275029525108}}
Iteration 4:
    {'target': -6.999472814518675, 'params': {'x': 2.2303920156083024, 'y': -0.7392021938893159}}
```

• 최적화 실행 결과를  
단계별로 리스트 (list)  
형태로 저장하고 있는  
클래스 변수



# 패키지 소개

## 클래스 함수



```
...
@Misc(
    author = {Fernando Nogueira},
    title = {{Bayesian Optimization}: Open source constrained
              global optimization tool for {Python}},
    year = {2014--},
    url = " https://github.com/fmfn/BayesianOptimization"
)
...
optimizer.set_bounds(new_bounds={"x": (-2, 3)})

optimizer.maximize(
    init_points=0,
    n_iter=5,
)
```

iter	target	x	y
6	-2.942	1.98	0.8567
7	-0.4597	1.096	1.508
8	0.5304	-0.6807	1.079
9	-5.33	-1.526	3.0
10	-5.419	-2.0	-0.5552

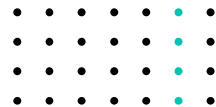
### set\_bounds

- set\_bounds 함수를  
사용해서 학습 과정에서  
파라미터의 범위를 변경할  
수 있다



# 패키지 소개

## 클래스 함수



```
...
@Misc{
    author = {Fernando Nogueira},
    title = {{Bayesian Optimization}: Open source constrained
            global optimization tool for {Python}},
    year = {2014--},
    url = " https://github.com/fmfn/BayesianOptimization"
}
...

optimizer.probe(
    params={"x": 0.5, "y": 0.7},
    lazy=True,
)

optimizer.probe(
    params=[-0.3, 0.1],
    lazy=True,
)

optimizer.maximize(init_points=0, n_iter=0)
```

iter	target	x	y
11	0.66	0.5	0.7
12	0.1	-0.3	0.1

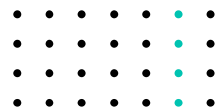
probe

- probe 함수를 사용하면  
파라미터 값을 지정해서  
최적화를 수행할 수 있다



# 패키지 소개

## 파일 저장



```
...
@Misc{
    author = {Fernando Nogueira},
    title = {{Bayesian Optimization}: Open source constrained
            global optimization tool for {Python}},
    year = {2014--},
    url = " https://github.com/fmfn/BayesianOptimization"
}
...

from bayes_opt.logger import JSONLogger
from bayes_opt.event import Events

logger = JSONLogger(path="./bayesian/logs.json")
optimizer.subscribe(Events.OPTIMIZATION_STEP, logger)

optimizer.maximize(
    init_points=2,
    n_iter=3,
)
```

iter	target	x	y
13	-12.48	-1.266	-2.446
14	-3.854	-1.069	-0.9266
15	-3.594	0.7709	3.0
16	0.8237	0.03431	1.419
17	0.9721	-0.1051	0.8701

### JSON 파일 저장

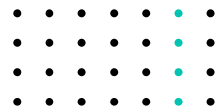
- maximize 함수를 호출할 때 이벤트를 파일에 기록한다





# 패키지 소개

## 파일 저장



### JSON 파일 재사용

- 기존에 학습된 내역을 재 활용해서 보다 효율적인 학습을 진행할 수 있다

```
...
@Misc{
  author = {Fernando Nogueira},
  title = {{Bayesian Optimization}: Open source constrained
          global optimization tool for {Python}},
  year = {2014--},
  url = " https://github.com/fmfn/BayesianOptimization"
}
...
```

```
from bayes_opt.util import load_logs
```

```
new_optimizer = BayesianOptimization(
    f=black_box_function,
    pbounds={"x": (-2, 2), "y": (-2, 2)},
    verbose=2,
    random_state=7,
)
```

```
load_logs(new_optimizer, logs=["./bayesian/logs.json"])
```

```
new_optimizer.maximize(
    init_points=2,
    n_iter=3,
)
```

	iter	target	x	y
	1	-1.887	-1.695	1.12
	2	0.928	-0.2464	0.8939
	3	-4.825	2.0	-0.351
	4	-3.456	2.0	1.675
	5	-0.5305	-0.7284	2.0

