Hyperparameter_Multiprocessing

May 19, 2023

[1]:
```
import concurrent.futures
import threading
import import_ipynb
from Softmax_Regression import test_work
import time
```

```
importing Jupyter notebook from Softmax_Regression.ipynb
(70000,)
(70000, 1)
category count: 1
[array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=uint8)]
regularization_parameter:0
Initial Cost is: 2.3024850979937166

2.2786589684539185
0.5559547252262049
0.5605332823837571
0.4568691436948896
0.48187963852916055
0.3187135733160607
0.45564174043136263
0.26870769689196256
0.41237311579214025
0.17107500739802817
0.2791846995324109
0.34003142552398347
0.26151930746303453
0.35679754692635873
0.26362894427519346
0.2696832327482677
0.18652846617872168
0.3716692012582663
0.17058755321920455
0.19663436426285044
0.3848164482335992
0.27803315532301204
0.42005768316959197
0.21526408547623627
```

```
0.307634923817838
0.2751623359696283
0.16550367723358866
0.35202585230940564
0.21876935994394714
0.20885406030415216
0.1590029538653099
0.3484740777246235
0.23276467047855356
0.49611634810362126
0.25171775633104393
0.22233737846839402
0.3254907927870663
0.42472213301635325
0.24964224522339257
0.3046025463498409
0.16348291902021198
0.5472683603007049
0.29978640425594616
0.33483584657446236
0.24560767209109038
0.17594353668981424
0.22546290071874253
0.2345823345806975
0.3199104128184141
0.16630328926393562
0.9137
```

[2]:
```
start_proc_time = time.process_time()
start_perf_counter = time.perf_counter()

retval = test_work(1.0,0.001)

end_proc_time = time.process_time()
end_perf_counter = time.perf_counter()
print(f"      : {int(round((end_proc_time - start_proc_time) * 1000))}ms")
print(f"    : {int(round((end_perf_counter - start_perf_counter) * 1000))}ms")


print(retval[0].shape)
print(retval[1].shape)
```

```
     : 2531ms
   : 46750ms
(50000, 1)
(785, 10)
```

[3]:
```
#ThreadPoolExecutor   : https://docs.python.org/ko/3/library/concurrent.futures.
 ↪html#threadpoolexecutor-example

import concurrent.futures
import urllib.request

URLS = ['http://www.foxnews.com/',
        'http://www.cnn.com/',
        'http://europe.wsj.com/',
        'http://www.bbc.co.uk/',
        'http://nonexistant-subdomain.python.org/']

# Retrieve a single page and report the URL and contents
def load_url(url, timeout):
    with urllib.request.urlopen(url, timeout=timeout) as conn:
        return conn.read()

# We can use a with statement to ensure threads are cleaned up promptly
with concurrent.futures.ThreadPoolExecutor(max_workers=5) as executor:
    # Start the load operations and mark each future with its URL
    future_to_url = {executor.submit(load_url, url, 60): url for url in URLS}
    for future in concurrent.futures.as_completed(future_to_url):
        url = future_to_url[future]
        try:
            data = future.result()
        except Exception as exc:
            print('%r generated an exception: %s' % (url, exc))
        else:
            print('%r page is %d bytes' % (url, len(data)))
```

```
'http://nonexistant-subdomain.python.org/' generated an exception: <urlopen
error [Errno 11001] getaddrinfo failed>
'http://www.bbc.co.uk/' page is 571374 bytes
'http://www.cnn.com/' page is 2044782 bytes
'http://www.foxnews.com/' page is 551620 bytes
'http://europe.wsj.com/' generated an exception: HTTP Error 403: Forbidden
```

[4]:
```
def sub_work(args):
    func = args[2]

    learning_rate, regs = func(args[1], args[0])

    print(f"start:learning_rate:{learning_rate:.4f},regs:{regs}, {args}")
    result = test_work(learning_rate, regs)
    print(f"done:learning_rate:{learning_rate:.4f},{result[3]}")
    cost_history = result[0]
    result_arg = [learning_rate, regs, result[2],result[3],]
```

```
    return result_arg
```

[5]:
```
#with Pool() as p:result = p.map(test_work, [i * 0.01 for i in range(0,50)])


def converter_func(i,j):
    learning_rate = 0.001 + i*i* 0.001
    regs = j*0.2
    return (learning_rate, regs)


hlist=[[(i,j)for j in range(20)]for i in range(25)]
with concurrent.futures.ThreadPoolExecutor(max_workers=12) as executor:
    args = [[i,j,converter_func] for i,j in sum(hlist, [])]

    future_to_work = {executor.submit(sub_work, arg): arg for arg in args}
    for future in concurrent.futures.as_completed(future_to_work):
        arg = future_to_work[future]
        try:
            data = future.result()
        except Exception as exc:
            print(arg)
            print(exc)
            hlist[arg[0]][arg[1]]=exc
        else:
            hlist[arg[0]][arg[1]]=data
```

```
start:learning_rate:0.0010,regs:0.0, [0, 0, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0020,regs:0.0, [0, 1, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0050,regs:0.0, [0, 2, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0100,regs:0.0, [0, 3, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0170,regs:0.0, [0, 4, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0260,regs:0.0, [0, 5, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0370,regs:0.0, [0, 6, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0500,regs:0.0, [0, 7, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0650,regs:0.0, [0, 8, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.0820,regs:0.0, [0, 9, <function converter_func at
0x0000024D3DBF8A60>]
start:learning_rate:0.1010,regs:0.0, [0, 10, <function converter_func at
```

```
0x0000024D3DBF8A60>]
start:learning_rate:0.1220,regs:0.0, [0, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.098
start:learning_rate:0.1450,regs:0.0, [0, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.098
start:learning_rate:0.1700,regs:0.0, [0, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.098
start:learning_rate:0.1970,regs:0.0, [0, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.098
start:learning_rate:0.2260,regs:0.0, [0, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.098
start:learning_rate:0.2570,regs:0.0, [0, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.098
start:learning_rate:0.2900,regs:0.0, [0, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.098
start:learning_rate:0.3250,regs:0.0, [0, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.098
start:learning_rate:0.3620,regs:0.0, [0, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.098
start:learning_rate:0.0010,regs:0.2, [1, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.098
start:learning_rate:0.0020,regs:0.2, [1, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.098
start:learning_rate:0.0050,regs:0.2, [1, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.098
start:learning_rate:0.0100,regs:0.2, [1, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.098
start:learning_rate:0.0170,regs:0.2, [1, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.098
start:learning_rate:0.0260,regs:0.2, [1, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.098
start:learning_rate:0.0370,regs:0.2, [1, 6, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.098
start:learning_rate:0.0500,regs:0.2, [1, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.098
start:learning_rate:0.0650,regs:0.2, [1, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.098
start:learning_rate:0.0820,regs:0.2, [1, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.098
start:learning_rate:0.1010,regs:0.2, [1, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.098
start:learning_rate:0.1220,regs:0.2, [1, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.9024
start:learning_rate:0.1450,regs:0.2, [1, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8905
start:learning_rate:0.1700,regs:0.2, [1, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8864
start:learning_rate:0.1970,regs:0.2, [1, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8897
start:learning_rate:0.2260,regs:0.2, [1, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8825
start:learning_rate:0.2570,regs:0.2, [1, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.888
start:learning_rate:0.2900,regs:0.2, [1, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8678
start:learning_rate:0.3250,regs:0.2, [1, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8948
start:learning_rate:0.3620,regs:0.2, [1, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.897
start:learning_rate:0.0010,regs:0.4, [2, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8635
start:learning_rate:0.0020,regs:0.4, [2, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8931
start:learning_rate:0.0050,regs:0.4, [2, 2, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8591
start:learning_rate:0.0100,regs:0.4, [2, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8902
start:learning_rate:0.0170,regs:0.4, [2, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.881
start:learning_rate:0.0260,regs:0.4, [2, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8767
start:learning_rate:0.0370,regs:0.4, [2, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8928
start:learning_rate:0.0500,regs:0.4, [2, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8863
start:learning_rate:0.0650,regs:0.4, [2, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8971
start:learning_rate:0.0820,regs:0.4, [2, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8951
start:learning_rate:0.1010,regs:0.4, [2, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8884
start:learning_rate:0.1220,regs:0.4, [2, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8867
start:learning_rate:0.1450,regs:0.4, [2, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8875
start:learning_rate:0.1700,regs:0.4, [2, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8906
start:learning_rate:0.1970,regs:0.4, [2, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8786
start:learning_rate:0.2260,regs:0.4, [2, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8901
start:learning_rate:0.2570,regs:0.4, [2, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8748
start:learning_rate:0.2900,regs:0.4, [2, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8969
start:learning_rate:0.3250,regs:0.4, [2, 18, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8762
start:learning_rate:0.3620,regs:0.4, [2, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8834
start:learning_rate:0.0010,regs:0.6000000000000001, [3, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8823
start:learning_rate:0.0020,regs:0.6000000000000001, [3, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8888
start:learning_rate:0.0050,regs:0.6000000000000001, [3, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8778
start:learning_rate:0.0100,regs:0.6000000000000001, [3, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8622
start:learning_rate:0.0170,regs:0.6000000000000001, [3, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8863
start:learning_rate:0.0260,regs:0.6000000000000001, [3, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8875
start:learning_rate:0.0370,regs:0.6000000000000001, [3, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8802
start:learning_rate:0.0500,regs:0.6000000000000001, [3, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8936
start:learning_rate:0.0650,regs:0.6000000000000001, [3, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8832
start:learning_rate:0.0820,regs:0.6000000000000001, [3, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8871
start:learning_rate:0.1010,regs:0.6000000000000001, [3, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8724
start:learning_rate:0.1220,regs:0.6000000000000001, [3, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8862
start:learning_rate:0.1450,regs:0.6000000000000001, [3, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8751
start:learning_rate:0.1700,regs:0.6000000000000001, [3, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.874
start:learning_rate:0.1970,regs:0.6000000000000001, [3, 14, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8889
start:learning_rate:0.2260,regs:0.6000000000000001, [3, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8674
start:learning_rate:0.2570,regs:0.6000000000000001, [3, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8957
start:learning_rate:0.2900,regs:0.6000000000000001, [3, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8831
start:learning_rate:0.3250,regs:0.6000000000000001, [3, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.9
start:learning_rate:0.3620,regs:0.6000000000000001, [3, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8826
start:learning_rate:0.0010,regs:0.8, [4, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8979
start:learning_rate:0.0020,regs:0.8, [4, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8803
start:learning_rate:0.0050,regs:0.8, [4, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8652
start:learning_rate:0.0100,regs:0.8, [4, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.891
start:learning_rate:0.0170,regs:0.8, [4, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8788
start:learning_rate:0.0260,regs:0.8, [4, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8792
start:learning_rate:0.0370,regs:0.8, [4, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8779
start:learning_rate:0.0500,regs:0.8, [4, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8851
start:learning_rate:0.0650,regs:0.8, [4, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8713
start:learning_rate:0.0820,regs:0.8, [4, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.889
start:learning_rate:0.1010,regs:0.8, [4, 10, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8831
start:learning_rate:0.1220,regs:0.8, [4, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8669
start:learning_rate:0.1450,regs:0.8, [4, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8817
start:learning_rate:0.1700,regs:0.8, [4, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8887
start:learning_rate:0.1970,regs:0.8, [4, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8685
start:learning_rate:0.2260,regs:0.8, [4, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8744
start:learning_rate:0.2570,regs:0.8, [4, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8341
start:learning_rate:0.2900,regs:0.8, [4, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8912
start:learning_rate:0.3250,regs:0.8, [4, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.893
start:learning_rate:0.3620,regs:0.8, [4, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8759
start:learning_rate:0.0010,regs:1.0, [5, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8926
start:learning_rate:0.0020,regs:1.0, [5, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8635
start:learning_rate:0.0050,regs:1.0, [5, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8554
start:learning_rate:0.0100,regs:1.0, [5, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8768
start:learning_rate:0.0170,regs:1.0, [5, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8779
start:learning_rate:0.0260,regs:1.0, [5, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8891
start:learning_rate:0.0370,regs:1.0, [5, 6, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.886
start:learning_rate:0.0500,regs:1.0, [5, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8928
start:learning_rate:0.0650,regs:1.0, [5, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8897
start:learning_rate:0.0820,regs:1.0, [5, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.868
start:learning_rate:0.1010,regs:1.0, [5, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8851
start:learning_rate:0.1220,regs:1.0, [5, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8681
start:learning_rate:0.1450,regs:1.0, [5, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8772
start:learning_rate:0.1700,regs:1.0, [5, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8874
start:learning_rate:0.1970,regs:1.0, [5, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8849
start:learning_rate:0.2260,regs:1.0, [5, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8946
start:learning_rate:0.2570,regs:1.0, [5, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.888
start:learning_rate:0.2900,regs:1.0, [5, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8785
start:learning_rate:0.3250,regs:1.0, [5, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8746
start:learning_rate:0.3620,regs:1.0, [5, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.883
start:learning_rate:0.0010,regs:1.2000000000000002, [6, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8815
start:learning_rate:0.0020,regs:1.2000000000000002, [6, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8804
start:learning_rate:0.0050,regs:1.2000000000000002, [6, 2, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8905
start:learning_rate:0.0100,regs:1.2000000000000002, [6, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8785
start:learning_rate:0.0170,regs:1.2000000000000002, [6, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8588
start:learning_rate:0.0260,regs:1.2000000000000002, [6, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8975
start:learning_rate:0.0370,regs:1.2000000000000002, [6, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8834
start:learning_rate:0.0500,regs:1.2000000000000002, [6, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8777
start:learning_rate:0.0650,regs:1.2000000000000002, [6, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8551
start:learning_rate:0.0820,regs:1.2000000000000002, [6, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8764
start:learning_rate:0.1010,regs:1.2000000000000002, [6, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8782
start:learning_rate:0.1220,regs:1.2000000000000002, [6, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8802
start:learning_rate:0.1450,regs:1.2000000000000002, [6, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8641
start:learning_rate:0.1700,regs:1.2000000000000002, [6, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8735
start:learning_rate:0.1970,regs:1.2000000000000002, [6, 14, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8862
start:learning_rate:0.2260,regs:1.2000000000000002, [6, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8856
start:learning_rate:0.2570,regs:1.2000000000000002, [6, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.875
start:learning_rate:0.2900,regs:1.2000000000000002, [6, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8845
start:learning_rate:0.3250,regs:1.2000000000000002, [6, 18, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8546
start:learning_rate:0.3620,regs:1.2000000000000002, [6, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8807
start:learning_rate:0.0010,regs:1.4000000000000001, [7, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8845
start:learning_rate:0.0020,regs:1.4000000000000001, [7, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8716
start:learning_rate:0.0050,regs:1.4000000000000001, [7, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8513
start:learning_rate:0.0100,regs:1.4000000000000001, [7, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.887
start:learning_rate:0.0170,regs:1.4000000000000001, [7, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8971
start:learning_rate:0.0260,regs:1.4000000000000001, [7, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.876
start:learning_rate:0.0370,regs:1.4000000000000001, [7, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8836
start:learning_rate:0.0500,regs:1.4000000000000001, [7, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8756
start:learning_rate:0.0650,regs:1.4000000000000001, [7, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8722
start:learning_rate:0.0820,regs:1.4000000000000001, [7, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.885
start:learning_rate:0.1010,regs:1.4000000000000001, [7, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8673
start:learning_rate:0.1220,regs:1.4000000000000001, [7, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8796
start:learning_rate:0.1450,regs:1.4000000000000001, [7, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8805
start:learning_rate:0.1700,regs:1.4000000000000001, [7, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8868
start:learning_rate:0.1970,regs:1.4000000000000001, [7, 14, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8832
start:learning_rate:0.2260,regs:1.4000000000000001, [7, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8771
start:learning_rate:0.2570,regs:1.4000000000000001, [7, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8594
start:learning_rate:0.2900,regs:1.4000000000000001, [7, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8787
start:learning_rate:0.3250,regs:1.4000000000000001, [7, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8628
start:learning_rate:0.3620,regs:1.4000000000000001, [7, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.835
start:learning_rate:0.0010,regs:1.6, [8, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8786
start:learning_rate:0.0020,regs:1.6, [8, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8709
start:learning_rate:0.0050,regs:1.6, [8, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8754
start:learning_rate:0.0100,regs:1.6, [8, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8811
start:learning_rate:0.0170,regs:1.6, [8, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8695
start:learning_rate:0.0260,regs:1.6, [8, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8896
start:learning_rate:0.0370,regs:1.6, [8, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8897
start:learning_rate:0.0500,regs:1.6, [8, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8907
start:learning_rate:0.0650,regs:1.6, [8, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8842
start:learning_rate:0.0820,regs:1.6, [8, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8944
start:learning_rate:0.1010,regs:1.6, [8, 10, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8897
start:learning_rate:0.1220,regs:1.6, [8, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8884
start:learning_rate:0.1450,regs:1.6, [8, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.832
start:learning_rate:0.1700,regs:1.6, [8, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8849
start:learning_rate:0.1970,regs:1.6, [8, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.884
start:learning_rate:0.2260,regs:1.6, [8, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.874
start:learning_rate:0.2570,regs:1.6, [8, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8902
start:learning_rate:0.2900,regs:1.6, [8, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8718
start:learning_rate:0.3250,regs:1.6, [8, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8878
start:learning_rate:0.3620,regs:1.6, [8, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8792
start:learning_rate:0.0010,regs:1.8, [9, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8194
start:learning_rate:0.0020,regs:1.8, [9, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8839
start:learning_rate:0.0050,regs:1.8, [9, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8659
start:learning_rate:0.0100,regs:1.8, [9, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8809
start:learning_rate:0.0170,regs:1.8, [9, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8797
start:learning_rate:0.0260,regs:1.8, [9, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8752
start:learning_rate:0.0370,regs:1.8, [9, 6, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8852
start:learning_rate:0.0500,regs:1.8, [9, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8609
start:learning_rate:0.0650,regs:1.8, [9, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8645
start:learning_rate:0.0820,regs:1.8, [9, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8968
start:learning_rate:0.1010,regs:1.8, [9, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8868
start:learning_rate:0.1220,regs:1.8, [9, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8606
start:learning_rate:0.1450,regs:1.8, [9, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8708
start:learning_rate:0.1700,regs:1.8, [9, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8706
start:learning_rate:0.1970,regs:1.8, [9, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8533
start:learning_rate:0.2260,regs:1.8, [9, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8666
start:learning_rate:0.2570,regs:1.8, [9, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8738
start:learning_rate:0.2900,regs:1.8, [9, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8838
start:learning_rate:0.3250,regs:1.8, [9, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8794
start:learning_rate:0.3620,regs:1.8, [9, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8738
start:learning_rate:0.0010,regs:2.0, [10, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8681
start:learning_rate:0.0020,regs:2.0, [10, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8657
start:learning_rate:0.0050,regs:2.0, [10, 2, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8709
start:learning_rate:0.0100,regs:2.0, [10, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8752
start:learning_rate:0.0170,regs:2.0, [10, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8813
start:learning_rate:0.0260,regs:2.0, [10, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8824
start:learning_rate:0.0370,regs:2.0, [10, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8913
start:learning_rate:0.0500,regs:2.0, [10, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8825
start:learning_rate:0.0650,regs:2.0, [10, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8689
start:learning_rate:0.0820,regs:2.0, [10, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8722
start:learning_rate:0.1010,regs:2.0, [10, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8802
start:learning_rate:0.1220,regs:2.0, [10, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8844
start:learning_rate:0.1450,regs:2.0, [10, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8671
start:learning_rate:0.1700,regs:2.0, [10, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8719
start:learning_rate:0.1970,regs:2.0, [10, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8754
start:learning_rate:0.2260,regs:2.0, [10, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8855
start:learning_rate:0.2570,regs:2.0, [10, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8735
start:learning_rate:0.2900,regs:2.0, [10, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8651
start:learning_rate:0.3250,regs:2.0, [10, 18, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8806
start:learning_rate:0.3620,regs:2.0, [10, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8841
start:learning_rate:0.0010,regs:2.2, [11, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8909
start:learning_rate:0.0020,regs:2.2, [11, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8744
start:learning_rate:0.0050,regs:2.2, [11, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8629
start:learning_rate:0.0100,regs:2.2, [11, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8875
start:learning_rate:0.0170,regs:2.2, [11, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.882
start:learning_rate:0.0260,regs:2.2, [11, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8789
start:learning_rate:0.0370,regs:2.2, [11, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8852
start:learning_rate:0.0500,regs:2.2, [11, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8752
start:learning_rate:0.0650,regs:2.2, [11, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8816
start:learning_rate:0.0820,regs:2.2, [11, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8806
start:learning_rate:0.1010,regs:2.2, [11, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8699
start:learning_rate:0.1220,regs:2.2, [11, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8781
start:learning_rate:0.1450,regs:2.2, [11, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8854
start:learning_rate:0.1700,regs:2.2, [11, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8684
start:learning_rate:0.1970,regs:2.2, [11, 14, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8656
start:learning_rate:0.2260,regs:2.2, [11, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8743
start:learning_rate:0.2570,regs:2.2, [11, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8827
start:learning_rate:0.2900,regs:2.2, [11, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.86
start:learning_rate:0.3250,regs:2.2, [11, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8856
start:learning_rate:0.3620,regs:2.2, [11, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8818
start:learning_rate:0.0010,regs:2.4000000000000004, [12, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.864
start:learning_rate:0.0020,regs:2.4000000000000004, [12, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8759
start:learning_rate:0.0050,regs:2.4000000000000004, [12, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8772
start:learning_rate:0.0100,regs:2.4000000000000004, [12, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.88
start:learning_rate:0.0170,regs:2.4000000000000004, [12, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8907
start:learning_rate:0.0260,regs:2.4000000000000004, [12, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8893
start:learning_rate:0.0370,regs:2.4000000000000004, [12, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8672
start:learning_rate:0.0500,regs:2.4000000000000004, [12, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8782
start:learning_rate:0.0650,regs:2.4000000000000004, [12, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8821
start:learning_rate:0.0820,regs:2.4000000000000004, [12, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8596
start:learning_rate:0.1010,regs:2.4000000000000004, [12, 10, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.886
start:learning_rate:0.1220,regs:2.4000000000000004, [12, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8695
start:learning_rate:0.1450,regs:2.4000000000000004, [12, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8938
start:learning_rate:0.1700,regs:2.4000000000000004, [12, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8788
start:learning_rate:0.1970,regs:2.4000000000000004, [12, 14, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.869
start:learning_rate:0.2260,regs:2.4000000000000004, [12, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8826
start:learning_rate:0.2570,regs:2.4000000000000004, [12, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8912
start:learning_rate:0.2900,regs:2.4000000000000004, [12, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8716
start:learning_rate:0.3250,regs:2.4000000000000004, [12, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8774
start:learning_rate:0.3620,regs:2.4000000000000004, [12, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8617
start:learning_rate:0.0010,regs:2.6, [13, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8781
start:learning_rate:0.0020,regs:2.6, [13, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8654
start:learning_rate:0.0050,regs:2.6, [13, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8726
start:learning_rate:0.0100,regs:2.6, [13, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8743
start:learning_rate:0.0170,regs:2.6, [13, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8834
start:learning_rate:0.0260,regs:2.6, [13, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.893
start:learning_rate:0.0370,regs:2.6, [13, 6, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8691
start:learning_rate:0.0500,regs:2.6, [13, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8798
start:learning_rate:0.0650,regs:2.6, [13, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8461
start:learning_rate:0.0820,regs:2.6, [13, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8699
start:learning_rate:0.1010,regs:2.6, [13, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8595
start:learning_rate:0.1220,regs:2.6, [13, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8538
start:learning_rate:0.1450,regs:2.6, [13, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8683
start:learning_rate:0.1700,regs:2.6, [13, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8796
start:learning_rate:0.1970,regs:2.6, [13, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8697
start:learning_rate:0.2260,regs:2.6, [13, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8715
start:learning_rate:0.2570,regs:2.6, [13, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8829
start:learning_rate:0.2900,regs:2.6, [13, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.872
start:learning_rate:0.3250,regs:2.6, [13, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8893
start:learning_rate:0.3620,regs:2.6, [13, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8591
start:learning_rate:0.0010,regs:2.8000000000000003, [14, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8761
start:learning_rate:0.0020,regs:2.8000000000000003, [14, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8699
start:learning_rate:0.0050,regs:2.8000000000000003, [14, 2, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8641
start:learning_rate:0.0100,regs:2.8000000000000003, [14, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8809
start:learning_rate:0.0170,regs:2.8000000000000003, [14, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8687
start:learning_rate:0.0260,regs:2.8000000000000003, [14, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8789
start:learning_rate:0.0370,regs:2.8000000000000003, [14, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.883
start:learning_rate:0.0500,regs:2.8000000000000003, [14, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8727
start:learning_rate:0.0650,regs:2.8000000000000003, [14, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8876
start:learning_rate:0.0820,regs:2.8000000000000003, [14, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8775
start:learning_rate:0.1010,regs:2.8000000000000003, [14, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8784
start:learning_rate:0.1220,regs:2.8000000000000003, [14, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.851
start:learning_rate:0.1450,regs:2.8000000000000003, [14, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8967
start:learning_rate:0.1700,regs:2.8000000000000003, [14, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8664
start:learning_rate:0.1970,regs:2.8000000000000003, [14, 14, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8586
start:learning_rate:0.2260,regs:2.8000000000000003, [14, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8834
start:learning_rate:0.2570,regs:2.8000000000000003, [14, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8702
start:learning_rate:0.2900,regs:2.8000000000000003, [14, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8634
start:learning_rate:0.3250,regs:2.8000000000000003, [14, 18, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.883
start:learning_rate:0.3620,regs:2.8000000000000003, [14, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8815
start:learning_rate:0.0010,regs:3.0, [15, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8746
start:learning_rate:0.0020,regs:3.0, [15, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8728
start:learning_rate:0.0050,regs:3.0, [15, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8726
start:learning_rate:0.0100,regs:3.0, [15, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.878
start:learning_rate:0.0170,regs:3.0, [15, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8632
start:learning_rate:0.0260,regs:3.0, [15, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8704
start:learning_rate:0.0370,regs:3.0, [15, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8812
start:learning_rate:0.0500,regs:3.0, [15, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8822
start:learning_rate:0.0650,regs:3.0, [15, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8814
start:learning_rate:0.0820,regs:3.0, [15, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8716
start:learning_rate:0.1010,regs:3.0, [15, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8712
start:learning_rate:0.1220,regs:3.0, [15, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8574
start:learning_rate:0.1450,regs:3.0, [15, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8586
start:learning_rate:0.1700,regs:3.0, [15, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8793
start:learning_rate:0.1970,regs:3.0, [15, 14, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8877
start:learning_rate:0.2260,regs:3.0, [15, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.886
start:learning_rate:0.2570,regs:3.0, [15, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8759
start:learning_rate:0.2900,regs:3.0, [15, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8724
start:learning_rate:0.3250,regs:3.0, [15, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8804
start:learning_rate:0.3620,regs:3.0, [15, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8751
start:learning_rate:0.0010,regs:3.2, [16, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8688
start:learning_rate:0.0020,regs:3.2, [16, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.867
start:learning_rate:0.0050,regs:3.2, [16, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8848
start:learning_rate:0.0100,regs:3.2, [16, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8873
start:learning_rate:0.0170,regs:3.2, [16, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8873
start:learning_rate:0.0260,regs:3.2, [16, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8748
start:learning_rate:0.0370,regs:3.2, [16, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8851
start:learning_rate:0.0500,regs:3.2, [16, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8688
start:learning_rate:0.0650,regs:3.2, [16, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8772
start:learning_rate:0.0820,regs:3.2, [16, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8877
start:learning_rate:0.1010,regs:3.2, [16, 10, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8853
start:learning_rate:0.1220,regs:3.2, [16, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8721
start:learning_rate:0.1450,regs:3.2, [16, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8795
start:learning_rate:0.1700,regs:3.2, [16, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8697
start:learning_rate:0.1970,regs:3.2, [16, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8544
start:learning_rate:0.2260,regs:3.2, [16, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8903
start:learning_rate:0.2570,regs:3.2, [16, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8789
start:learning_rate:0.2900,regs:3.2, [16, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8754
start:learning_rate:0.3250,regs:3.2, [16, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8749
start:learning_rate:0.3620,regs:3.2, [16, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8757
start:learning_rate:0.0010,regs:3.4000000000000004, [17, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8633
start:learning_rate:0.0020,regs:3.4000000000000004, [17, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8815
start:learning_rate:0.0050,regs:3.4000000000000004, [17, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8944
start:learning_rate:0.0100,regs:3.4000000000000004, [17, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8598
start:learning_rate:0.0170,regs:3.4000000000000004, [17, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8935
start:learning_rate:0.0260,regs:3.4000000000000004, [17, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8745
start:learning_rate:0.0370,regs:3.4000000000000004, [17, 6, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8825
start:learning_rate:0.0500,regs:3.4000000000000004, [17, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8916
start:learning_rate:0.0650,regs:3.4000000000000004, [17, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8769
start:learning_rate:0.0820,regs:3.4000000000000004, [17, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8807
start:learning_rate:0.1010,regs:3.4000000000000004, [17, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8679
start:learning_rate:0.1220,regs:3.4000000000000004, [17, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8903
start:learning_rate:0.1450,regs:3.4000000000000004, [17, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8604
start:learning_rate:0.1700,regs:3.4000000000000004, [17, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8817
start:learning_rate:0.1970,regs:3.4000000000000004, [17, 14, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8777
start:learning_rate:0.2260,regs:3.4000000000000004, [17, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8665
start:learning_rate:0.2570,regs:3.4000000000000004, [17, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8692
start:learning_rate:0.2900,regs:3.4000000000000004, [17, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8792
start:learning_rate:0.3250,regs:3.4000000000000004, [17, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8774
start:learning_rate:0.3620,regs:3.4000000000000004, [17, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8876
start:learning_rate:0.0010,regs:3.6, [18, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8851
start:learning_rate:0.0020,regs:3.6, [18, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8717
start:learning_rate:0.0050,regs:3.6, [18, 2, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8893
start:learning_rate:0.0100,regs:3.6, [18, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8727
start:learning_rate:0.0170,regs:3.6, [18, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.884
start:learning_rate:0.0260,regs:3.6, [18, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8836
start:learning_rate:0.0370,regs:3.6, [18, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8872
start:learning_rate:0.0500,regs:3.6, [18, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8752
start:learning_rate:0.0650,regs:3.6, [18, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8766
start:learning_rate:0.0820,regs:3.6, [18, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8889
start:learning_rate:0.1010,regs:3.6, [18, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8938
start:learning_rate:0.1220,regs:3.6, [18, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8813
start:learning_rate:0.1450,regs:3.6, [18, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8891
start:learning_rate:0.1700,regs:3.6, [18, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8798
start:learning_rate:0.1970,regs:3.6, [18, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8835
start:learning_rate:0.2260,regs:3.6, [18, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.857
start:learning_rate:0.2570,regs:3.6, [18, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8814
start:learning_rate:0.2900,regs:3.6, [18, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8777
start:learning_rate:0.3250,regs:3.6, [18, 18, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8783
start:learning_rate:0.3620,regs:3.6, [18, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8657
start:learning_rate:0.0010,regs:3.8000000000000003, [19, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8942
start:learning_rate:0.0020,regs:3.8000000000000003, [19, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8836
start:learning_rate:0.0050,regs:3.8000000000000003, [19, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8593
start:learning_rate:0.0100,regs:3.8000000000000003, [19, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8669
start:learning_rate:0.0170,regs:3.8000000000000003, [19, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8818
start:learning_rate:0.0260,regs:3.8000000000000003, [19, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8703
start:learning_rate:0.0370,regs:3.8000000000000003, [19, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8598
start:learning_rate:0.0500,regs:3.8000000000000003, [19, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8704
start:learning_rate:0.0650,regs:3.8000000000000003, [19, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8847
start:learning_rate:0.0820,regs:3.8000000000000003, [19, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.873
start:learning_rate:0.1010,regs:3.8000000000000003, [19, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8822
start:learning_rate:0.1220,regs:3.8000000000000003, [19, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8681
start:learning_rate:0.1450,regs:3.8000000000000003, [19, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8592
start:learning_rate:0.1700,regs:3.8000000000000003, [19, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.861
start:learning_rate:0.1970,regs:3.8000000000000003, [19, 14, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8734
start:learning_rate:0.2260,regs:3.8000000000000003, [19, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8677
start:learning_rate:0.2570,regs:3.8000000000000003, [19, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.861
start:learning_rate:0.2900,regs:3.8000000000000003, [19, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8769
start:learning_rate:0.3250,regs:3.8000000000000003, [19, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8733
start:learning_rate:0.3620,regs:3.8000000000000003, [19, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8631
start:learning_rate:0.0010,regs:4.0, [20, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8586
start:learning_rate:0.0020,regs:4.0, [20, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8811
start:learning_rate:0.0050,regs:4.0, [20, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8783
start:learning_rate:0.0100,regs:4.0, [20, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8814
start:learning_rate:0.0170,regs:4.0, [20, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8664
start:learning_rate:0.0260,regs:4.0, [20, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.881
start:learning_rate:0.0370,regs:4.0, [20, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8868
start:learning_rate:0.0500,regs:4.0, [20, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8764
start:learning_rate:0.0650,regs:4.0, [20, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8893
start:learning_rate:0.0820,regs:4.0, [20, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8648
start:learning_rate:0.1010,regs:4.0, [20, 10, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8797
start:learning_rate:0.1220,regs:4.0, [20, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8699
start:learning_rate:0.1450,regs:4.0, [20, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8859
start:learning_rate:0.1700,regs:4.0, [20, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8848
start:learning_rate:0.1970,regs:4.0, [20, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8682
start:learning_rate:0.2260,regs:4.0, [20, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8813
start:learning_rate:0.2570,regs:4.0, [20, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.866
start:learning_rate:0.2900,regs:4.0, [20, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.883
start:learning_rate:0.3250,regs:4.0, [20, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.89
start:learning_rate:0.3620,regs:4.0, [20, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.878
start:learning_rate:0.0010,regs:4.2, [21, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.894
start:learning_rate:0.0020,regs:4.2, [21, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8465
start:learning_rate:0.0050,regs:4.2, [21, 2, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8712
start:learning_rate:0.0100,regs:4.2, [21, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8738
start:learning_rate:0.0170,regs:4.2, [21, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8812
start:learning_rate:0.0260,regs:4.2, [21, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8724
start:learning_rate:0.0370,regs:4.2, [21, 6, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8576
start:learning_rate:0.0500,regs:4.2, [21, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8564
start:learning_rate:0.0650,regs:4.2, [21, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8744
start:learning_rate:0.0820,regs:4.2, [21, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8725
start:learning_rate:0.1010,regs:4.2, [21, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8711
start:learning_rate:0.1220,regs:4.2, [21, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8816
start:learning_rate:0.1450,regs:4.2, [21, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8654
start:learning_rate:0.1700,regs:4.2, [21, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8669
start:learning_rate:0.1970,regs:4.2, [21, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8328
start:learning_rate:0.2260,regs:4.2, [21, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8611
start:learning_rate:0.2570,regs:4.2, [21, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8679
start:learning_rate:0.2900,regs:4.2, [21, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8866
start:learning_rate:0.3250,regs:4.2, [21, 18, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8716
start:learning_rate:0.3620,regs:4.2, [21, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8957
start:learning_rate:0.0010,regs:4.4, [22, 0, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8829
start:learning_rate:0.0020,regs:4.4, [22, 1, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8744
start:learning_rate:0.0050,regs:4.4, [22, 2, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8843
start:learning_rate:0.0100,regs:4.4, [22, 3, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8888
start:learning_rate:0.0170,regs:4.4, [22, 4, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8764
start:learning_rate:0.0260,regs:4.4, [22, 5, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.8697
start:learning_rate:0.0370,regs:4.4, [22, 6, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8876
start:learning_rate:0.0500,regs:4.4, [22, 7, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8793
start:learning_rate:0.0650,regs:4.4, [22, 8, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8871
start:learning_rate:0.0820,regs:4.4, [22, 9, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8693
start:learning_rate:0.1010,regs:4.4, [22, 10, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8733
start:learning_rate:0.1220,regs:4.4, [22, 11, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8774
start:learning_rate:0.1450,regs:4.4, [22, 12, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8751
start:learning_rate:0.1700,regs:4.4, [22, 13, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8775
start:learning_rate:0.1970,regs:4.4, [22, 14, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8627
start:learning_rate:0.2260,regs:4.4, [22, 15, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8343
start:learning_rate:0.2570,regs:4.4, [22, 16, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8805
start:learning_rate:0.2900,regs:4.4, [22, 17, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8783
start:learning_rate:0.3250,regs:4.4, [22, 18, <function converter_func at
0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8606
start:learning_rate:0.3620,regs:4.4, [22, 19, <function converter_func at
0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8696
start:learning_rate:0.0010,regs:4.6000000000000005, [23, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8599
start:learning_rate:0.0020,regs:4.6000000000000005, [23, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.873
start:learning_rate:0.0050,regs:4.6000000000000005, [23, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8658
start:learning_rate:0.0100,regs:4.6000000000000005, [23, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8722
start:learning_rate:0.0170,regs:4.6000000000000005, [23, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.856
start:learning_rate:0.0260,regs:4.6000000000000005, [23, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8483
start:learning_rate:0.0370,regs:4.6000000000000005, [23, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.889
start:learning_rate:0.0500,regs:4.6000000000000005, [23, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8668
start:learning_rate:0.0650,regs:4.6000000000000005, [23, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.869
start:learning_rate:0.0820,regs:4.6000000000000005, [23, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8543
start:learning_rate:0.1010,regs:4.6000000000000005, [23, 10, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8724
start:learning_rate:0.1220,regs:4.6000000000000005, [23, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8651
start:learning_rate:0.1450,regs:4.6000000000000005, [23, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8741
start:learning_rate:0.1700,regs:4.6000000000000005, [23, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.869
start:learning_rate:0.1970,regs:4.6000000000000005, [23, 14, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.8867
start:learning_rate:0.2260,regs:4.6000000000000005, [23, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8808
start:learning_rate:0.2570,regs:4.6000000000000005, [23, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8678
start:learning_rate:0.2900,regs:4.6000000000000005, [23, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.8727
start:learning_rate:0.3250,regs:4.6000000000000005, [23, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8691
start:learning_rate:0.3620,regs:4.6000000000000005, [23, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8903
start:learning_rate:0.0010,regs:4.800000000000001, [24, 0, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8876
start:learning_rate:0.0020,regs:4.800000000000001, [24, 1, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.851
start:learning_rate:0.0050,regs:4.800000000000001, [24, 2, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8899
start:learning_rate:0.0100,regs:4.800000000000001, [24, 3, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.8852
start:learning_rate:0.0170,regs:4.800000000000001, [24, 4, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8777
start:learning_rate:0.0260,regs:4.800000000000001, [24, 5, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.878
start:learning_rate:0.0370,regs:4.800000000000001, [24, 6, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8816
start:learning_rate:0.0500,regs:4.800000000000001, [24, 7, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8913
start:learning_rate:0.0650,regs:4.800000000000001, [24, 8, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8779
start:learning_rate:0.0820,regs:4.800000000000001, [24, 9, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8766
start:learning_rate:0.1010,regs:4.800000000000001, [24, 10, <function
converter_func at 0x0000024D3DBF8A60>]
```

```
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.851
start:learning_rate:0.1220,regs:4.800000000000001, [24, 11, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0010,regularization_parameter:0.0010,score:0.8728
start:learning_rate:0.1450,regs:4.800000000000001, [24, 12, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0020,regularization_parameter:0.0020,score:0.8553
start:learning_rate:0.1700,regs:4.800000000000001, [24, 13, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0050,regularization_parameter:0.0050,score:0.8791
start:learning_rate:0.1970,regs:4.800000000000001, [24, 14, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0100,regularization_parameter:0.0100,score:0.876
start:learning_rate:0.2260,regs:4.800000000000001, [24, 15, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0170,regularization_parameter:0.0170,score:0.8471
start:learning_rate:0.2570,regs:4.800000000000001, [24, 16, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0260,regularization_parameter:0.0260,score:0.8787
start:learning_rate:0.2900,regs:4.800000000000001, [24, 17, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0370,regularization_parameter:0.0370,score:0.856
start:learning_rate:0.3250,regs:4.800000000000001, [24, 18, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0500,regularization_parameter:0.0500,score:0.8765
start:learning_rate:0.3620,regs:4.800000000000001, [24, 19, <function
converter_func at 0x0000024D3DBF8A60>]
done:learning_rate:0.0650,regularization_parameter:0.0650,score:0.8297
done:learning_rate:0.0820,regularization_parameter:0.0820,score:0.8522
done:learning_rate:0.1010,regularization_parameter:0.1010,score:0.8584
done:learning_rate:0.1220,regularization_parameter:0.1220,score:0.8713
done:learning_rate:0.1450,regularization_parameter:0.1450,score:0.834
done:learning_rate:0.1700,regularization_parameter:0.1700,score:0.8655
done:learning_rate:0.1970,regularization_parameter:0.1970,score:0.87
done:learning_rate:0.2260,regularization_parameter:0.2260,score:0.8653
done:learning_rate:0.2570,regularization_parameter:0.2570,score:0.8753
done:learning_rate:0.2900,regularization_parameter:0.2900,score:0.8696
done:learning_rate:0.3250,regularization_parameter:0.3250,score:0.8596
done:learning_rate:0.3620,regularization_parameter:0.3620,score:0.8784
```

[6]:
```
import json
with open('result_storege.json','w') as w:
    json.dump(hlist, w, indent = '\t')
    w.close()
```

[ ]:
```

```