updated-NES-google

September 29, 2021

[2]:

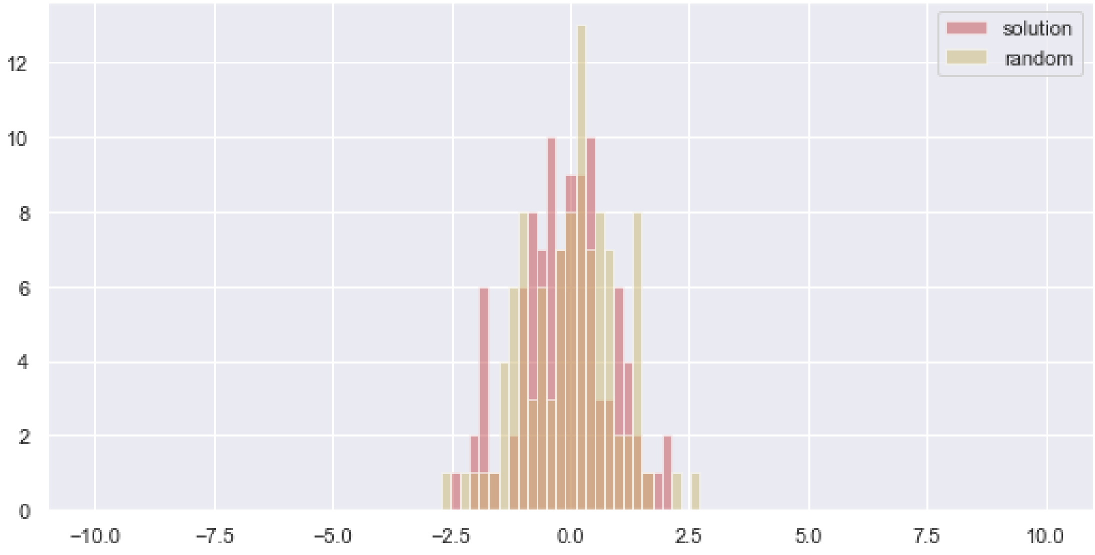
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

[21]:

```
plt.figure(figsize = (10, 5))
bins = np.linspace(-10, 10, 100)

solution = np.random.randn(100)
w = np.random.randn(100)

plt.hist(solution, bins, alpha = 0.5, label = 'solution', color = 'r')
plt.hist(w, bins, alpha = 0.5, label = 'random', color = 'y')
plt.legend()
plt.show()
```



[22]:

```
def f(w):
    return -np.sum(np.square(solution - w))
```

```
npop = 50
sigma = 0.1
alpha = 0.001

for i in range(5000):

    if (i + 1) % 1000 == 0:
        print(
            'iter %d. w: %s, solution: %s, reward: %f'
            % (i + 1, str(w[-1]), str(solution[-1]), f(w))
        )
    N = np.random.randn(npop, 100)
    R = np.zeros(npop)
    for j in range(npop):
        w_try = w + sigma * N[j]
        R[j] = f(w_try)

    A = (R - np.mean(R)) / np.std(R)
    w = w + alpha / (npop * sigma) * np.dot(N.T, A)
```

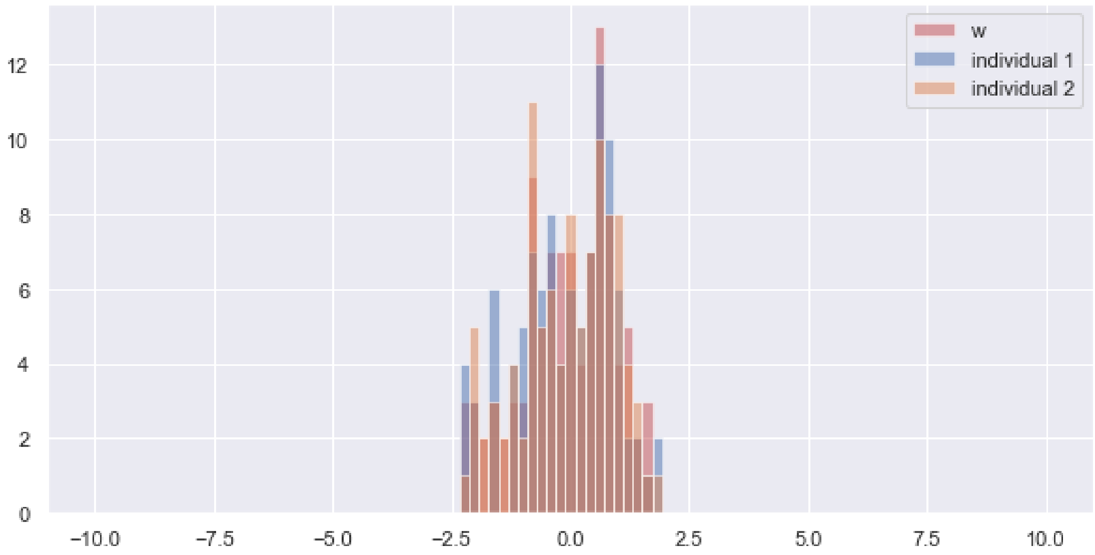
```
iter 1000. w: 0.0952791586701015, solution: 0.5720518054873052, reward:
-20.148099
iter 2000. w: 0.5750455468679501, solution: 0.5720518054873052, reward:
-0.008058
iter 3000. w: 0.5751585748688035, solution: 0.5720518054873052, reward:
-0.008793
iter 4000. w: 0.5665604300033952, solution: 0.5720518054873052, reward:
-0.007711
iter 5000. w: 0.5619489293298067, solution: 0.5720518054873052, reward:
-0.005604
```

[12]:

```
'''
I want to compare my first two individuals with my real w
'''
plt.figure(figsize=(10,5))

sigma = 0.1
N = np.random.randn(npop, 100)
individuals = []
for j in range(2):
    individuals.append(w + sigma * N[j])


plt.hist(w, bins, alpha=0.5, label='w',color='r')
plt.hist(individuals[0], bins, alpha=0.5, label='individual 1')
plt.hist(individuals[1], bins, alpha=0.5, label='individual 2')
plt.legend()
plt.show()
```



[29]:

```
import pandas as pd
google = pd.read_csv('/Users/huseinzolkepli/Desktop/GOOG.csv')
google.head()
```

[29]:

```
         Date        Open        High         Low       Close   Adj Close  \
0  2017-10-16  992.099976  993.906982  984.000000  992.000000  992.000000
1  2017-10-17  990.289978  996.440002  988.590027  992.179993  992.179993
2  2017-10-18  991.770020  996.719971  986.974976  992.809998  992.809998
3  2017-10-19  986.000000  988.880005  978.390015  984.450012  984.450012
4  2017-10-20  989.440002  991.000000  984.580017  988.200012  988.200012

    Volume
0   910500
1  1290200
2  1057600
3  1313600
4  1183200
```

[58]:

```
def get_state(data, t, n):
    d = t - n + 1
    block = data[d : t + 1] if d >= 0 else -d * [data[0]] + data[: t + 1]
    res = []
    for i in range(n - 1):
        res.append(block[i + 1] - block[i])
    return np.array([res])
```

[60]:

```
close = google.Close.values.tolist()
get_state(close, 0, 10)
```

[60]:

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

[61]:

```
get_state(close, 1, 10)
```

[61]:

```
array([[0.      , 0.      , 0.      , 0.      , 0.      , 0.      ,
        0.      , 0.      , 0.179993]])
```

[62]:

```
get_state(close, 2, 10)
```

[62]:

```
array([[0.      , 0.      , 0.      , 0.      , 0.      , 0.      ,
        0.      , 0.179993, 0.630005]])
```

[63]:

```
class Deep_Evolution_Strategy:
    def __init__(
        self, weights, reward_function, population_size, sigma, learning_rate
    ):
        self.weights = weights
        self.reward_function = reward_function
        self.population_size = population_size
        self.sigma = sigma
        self.learning_rate = learning_rate

    def _get_weight_from_population(self, weights, population):
        weights_population = []
        for index, i in enumerate(population):
            jittered = self.sigma * i
            weights_population.append(weights[index] + jittered)
        return weights_population

    def get_weights(self):
        return self.weights

    def train(self, epoch = 100, print_every = 1):
        lasttime = time.time()
        for i in range(epoch):
            population = []
            rewards = np.zeros(self.population_size)
            for k in range(self.population_size):
                x = []
                for w in self.weights:
                    x.append(np.random.randn(*w.shape))
                population.append(x)
            for k in range(self.population_size):
                weights_population = self._get_weight_from_population(
                    self.weights, population[k]
                )
                rewards[k] = self.reward_function(weights_population)
            rewards = (rewards - np.mean(rewards)) / np.std(rewards)
            for index, w in enumerate(self.weights):
                A = np.array([p[index] for p in population])
                self.weights[index] = (
                    w
                    + self.learning_rate
                    / (self.population_size * self.sigma)
                    * np.dot(A.T, rewards).T
                )
            if (i + 1) % print_every == 0:
                print(
                    'iter %d. reward: %f'
                    % (i + 1, self.reward_function(self.weights))
                )
        print('time taken to train:', time.time() - lasttime, 'seconds')
```

[64]:

```
class Model:
    def __init__(self, input_size, layer_size, output_size):
        self.weights = [
            np.random.randn(input_size, layer_size),
            np.random.randn(layer_size, output_size),
            np.random.randn(layer_size, 1),
            np.random.randn(1, layer_size),
        ]

    def predict(self, inputs):
        feed = np.dot(inputs, self.weights[0]) + self.weights[-1]
        decision = np.dot(feed, self.weights[1])
        buy = np.dot(feed, self.weights[2])
        return decision, buy

    def get_weights(self):
        return self.weights

    def set_weights(self, weights):
        self.weights = weights
```

[65]:

```
window_size = 30
model = Model(window_size, 500, 3)
```

[67]:

```
initial_money = 10000
starting_money = initial_money
```

```
len_close = len(close) - 1
weight = model
skip = 1

state = get_state(close, 0, window_size + 1)
inventory = []
quantity = 0

max_buy = 5
max_sell = 5


def act(model, sequence):
    decision, buy = model.predict(np.array(sequence))
    return np.argmax(decision[0]), int(buy[0])


for t in range(0, len_close, skip):
    action, buy = act(weight, state)
    next_state = get_state(close, t + 1, window_size + 1)
    if action == 1 and initial_money >= close[t]:
        if buy < 0:
            buy = 1
        if buy > max_buy:
            buy_units = max_buy
        else:
            buy_units = buy
        total_buy = buy_units * close[t]
        initial_money -= total_buy
        inventory.append(total_buy)
        quantity += buy_units
    elif action == 2 and len(inventory) > 0:
        if quantity > max_sell:
            sell_units = max_sell
        else:
            sell_units = quantity
        quantity -= sell_units
        total_sell = sell_units * close[t]
        initial_money += total_sell

    state = next_state
((initial_money - starting_money) / starting_money) * 100
```

[67]:

```
-89.2658852200001
```

[77]:

```
import time
```

```python
class Agent:

    POPULATION_SIZE = 15
    SIGMA = 0.1
    LEARNING_RATE = 0.03

    def __init__(
        self, model, money, max_buy, max_sell, close, window_size, skip
    ):
        self.window_size = window_size
        self.skip = skip
        self.close = close
        self.model = model
        self.initial_money = money
        self.max_buy = max_buy
        self.max_sell = max_sell
        self.es = Deep_Evolution_Strategy(
            self.model.get_weights(),
            self.get_reward,
            self.POPULATION_SIZE,
            self.SIGMA,
            self.LEARNING_RATE,
        )

    def act(self, sequence):
        decision, buy = self.model.predict(np.array(sequence))
        return np.argmax(decision[0]), int(buy[0])

    def get_reward(self, weights):
        initial_money = self.initial_money
        starting_money = initial_money
        len_close = len(self.close) - 1

        self.model.weights = weights
        state = get_state(self.close, 0, self.window_size + 1)
        inventory = []
        quantity = 0
        for t in range(0, len_close, self.skip):
            action, buy = self.act(state)
            next_state = get_state(self.close, t + 1, self.window_size + 1)
            if action == 1 and initial_money >= self.close[t]:
                if buy < 0:
                    buy = 1
                if buy > self.max_buy:
                    buy_units = self.max_buy
                else:
                    buy_units = buy
                total_buy = buy_units * self.close[t]
                initial_money -= total_buy
                inventory.append(total_buy)
                quantity += buy_units
            elif action == 2 and len(inventory) > 0:
                if quantity > self.max_sell:
                    sell_units = self.max_sell
                else:
                    sell_units = quantity
                quantity -= sell_units
                total_sell = sell_units * self.close[t]
                initial_money += total_sell

            state = next_state
        return ((initial_money - starting_money) / starting_money) * 100

    def fit(self, iterations, checkpoint):
        self.es.train(iterations, print_every = checkpoint)

    def buy(self):
        initial_money = self.initial_money
        len_close = len(self.close) - 1
        state = get_state(self.close, 0, self.window_size + 1)
        starting_money = initial_money
        states_sell = []
        states_buy = []
        inventory = []
        quantity = 0
        for t in range(0, len_close, self.skip):
            action, buy = self.act(state)
            next_state = get_state(self.close, t + 1, self.window_size + 1)
            if action == 1 and initial_money >= self.close[t]:
                if buy < 0:
                    buy = 1
                if buy > self.max_buy:
                    buy_units = self.max_buy
                else:
                    buy_units = buy
                total_buy = buy_units * self.close[t]
                initial_money -= total_buy
                inventory.append(total_buy)
                quantity += buy_units
                states_buy.append(t)
                print(
                    'day %d: buy %d units at price %f, total balance %f'
                    % (t, buy_units, total_buy, initial_money)
                )
            elif action == 2 and len(inventory) > 0:
                bought_price = inventory.pop(0)
                if quantity > self.max_sell:
                    sell_units = self.max_sell
                else:
                    sell_units = quantity
                if sell_units < 1:
                    continue
                quantity -= sell_units
                total_sell = sell_units * self.close[t]
                initial_money += total_sell
                states_sell.append(t)
                try:
                    invest = ((total_sell - bought_price) / bought_price) * 100
                except:
                    invest = 0
                print(
                    'day %d, sell %d units at price %f, investment %f %%, total balance %f,'
                    % (t, sell_units, total_sell, invest, initial_money)
                )
            state = next_state

        invest = ((initial_money - starting_money) / starting_money) * 100
        print(
            '\ntotal gained %f, total investment %f %%'
            % (initial_money - starting_money, invest)
        )
        plt.figure(figsize = (20, 10))
        plt.plot(close, label = 'true close', c = 'g')
        plt.plot(
            close, 'X', label = 'predict buy', markevery = states_buy, c = 'b'
        )
        plt.plot(
            close, 'o', label = 'predict sell', markevery = states_sell, c = 'r'
        )
        plt.legend()
        plt.show()
```

[78]:

```
model = Model(input_size = window_size, layer_size = 500, output_size = 3)
agent = Agent(
    model = model,
    money = 10000,
    max_buy = 5,
    max_sell = 5,
    close = close,
    window_size = window_size,
    skip = 1,
)
```

[79]:

```
agent.fit(iterations = 500, checkpoint = 10)
```

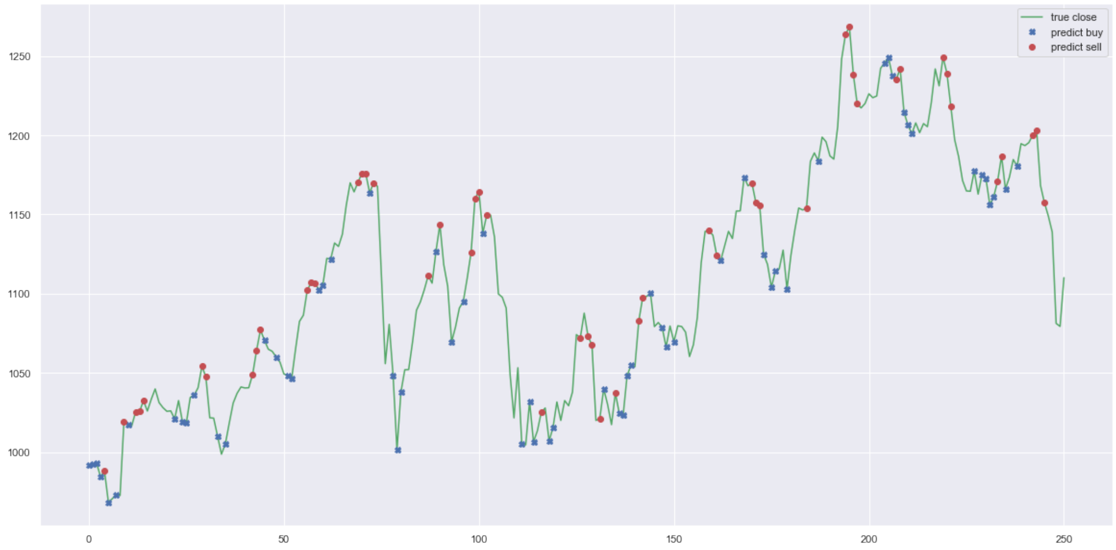
```
iter 10. reward: 36.181611
iter 20. reward: 50.767101
iter 30. reward: 65.467698
iter 40. reward: 71.316103
iter 50. reward: 82.881994
iter 60. reward: 84.293704
iter 70. reward: 78.501997
iter 80. reward: 94.488579
iter 90. reward: 86.526799
iter 100. reward: 85.882890
iter 110. reward: 86.063284
iter 120. reward: 90.334301
iter 130. reward: 85.850098
iter 140. reward: 91.399606
iter 150. reward: 87.862805
iter 160. reward: 97.226486
iter 170. reward: 86.767297
iter 180. reward: 97.016782
iter 190. reward: 97.843791
iter 200. reward: 89.146606
iter 210. reward: 96.508885
iter 220. reward: 97.765979
iter 230. reward: 98.256375
iter 240. reward: 99.942482
iter 250. reward: 94.536183
iter 260. reward: 96.916185
iter 270. reward: 93.193185
iter 280. reward: 100.844085
iter 290. reward: 100.994682
iter 300. reward: 101.523774
iter 310. reward: 102.090896
iter 320. reward: 102.176091
iter 330. reward: 92.306981
iter 340. reward: 105.409190
iter 350. reward: 103.159886
iter 360. reward: 99.091287
iter 370. reward: 108.475085
iter 380. reward: 102.349682
iter 390. reward: 110.289382
iter 400. reward: 103.371389
iter 410. reward: 110.951287
iter 420. reward: 111.561078
iter 430. reward: 112.275285
iter 440. reward: 113.112587
iter 450. reward: 110.838887
iter 460. reward: 111.243782
iter 470. reward: 112.924874
iter 480. reward: 111.705677
iter 490. reward: 110.903074
iter 500. reward: 112.986871
time taken to train: 60.56475520133972 seconds
```

[80]:

```
agent.buy()
```

```
day 0: buy 1 units at price 992.000000, total balance 9008.000000
day 1: buy 1 units at price 992.179993, total balance 8015.820007
day 2: buy 1 units at price 992.809998, total balance 7023.010009
day 3: buy 5 units at price 4922.250060, total balance 2100.759949
day 4, sell 5 units at price 4941.000060, investment 398.084683 %, total balance
7041.760009,
day 5: buy 5 units at price 4842.250060, total balance 2199.509949
day 7: buy 5 units at price 4866.650085, total balance -2667.140136
day 9, sell 5 units at price 5096.350100, investment 413.651770 %, total balance
2429.209964,
day 10: buy 5 units at price 5085.549925, total balance -2656.339961
day 12, sell 5 units at price 5127.500000, investment 416.463373 %, total
balance 2471.160039,
day 13, sell 5 units at price 5127.899780, investment 4.177962 %, total balance
7599.059819,
day 14, sell 3 units at price 3097.439940, investment -36.033045 %, total
balance 10696.499759,
day 22: buy 1 units at price 1020.909973, total balance 9675.589786
day 24: buy 1 units at price 1019.090027, total balance 8656.499759
day 25: buy 5 units at price 5091.900025, total balance 3564.599734
day 27: buy 5 units at price 5179.799805, total balance -1615.200071
day 29, sell 5 units at price 5271.049805, investment 416.308974 %, total
balance 3655.849734,
day 30, sell 5 units at price 5237.050170, investment 413.894752 %, total
balance 8892.899904,
day 33: buy 5 units at price 5050.849915, total balance 3842.049989
day 35: buy 5 units at price 5025.750120, total balance -1183.700131
day 42, sell 5 units at price 5245.750120, investment 3.021467 %, total balance
4062.049989,
day 43, sell 5 units at price 5320.949705, investment 2.725007 %, total balance
9382.999694,
day 44, sell 2 units at price 2154.280030, investment -57.348168 %, total
balance 11537.279724,
day 45: buy 1 units at price 1070.680054, total balance 10466.599670
day 48: buy 1 units at price 1060.119995, total balance 9406.479675
day 51: buy 5 units at price 5240.700075, total balance 4165.779600
day 52: buy 5 units at price 5232.000120, total balance -1066.220520
day 56, sell 5 units at price 5511.149900, investment 9.658255 %, total balance
4444.929380,
day 57, sell 5 units at price 5534.699705, investment 416.933110 %, total
balance 9979.629085,
day 58, sell 2 units at price 2212.520020, investment 108.704678 %, total
balance 12192.149105,
day 59: buy 5 units at price 5513.049925, total balance 6679.099180
day 60: buy 5 units at price 5527.600100, total balance 1151.499080
day 62: buy 5 units at price 5608.800050, total balance -4457.300970
day 69, sell 5 units at price 5851.849975, investment 11.661608 %, total balance
1394.549005,
day 70, sell 5 units at price 5879.199830, investment 12.370025 %, total balance
7273.748835,
day 71, sell 5 units at price 5877.899780, investment 6.617931 %, total balance
13151.648615,
day 72: buy 5 units at price 5818.449705, total balance 7333.198910
day 73, sell 5 units at price 5849.699705, investment 5.827115 %, total balance
13182.898615,
day 78: buy 5 units at price 5242.899780, total balance 7939.998835
day 79: buy 5 units at price 5007.600100, total balance 2932.398735
day 80: buy 5 units at price 5188.900145, total balance -2256.501410
day 87, sell 5 units at price 5556.699830, investment -0.928901 %, total balance
3300.198420,
day 89: buy 1 units at price 1126.790039, total balance 2173.408381
day 90, sell 5 units at price 5718.750000, investment -1.713510 %, total balance
7892.158381,
day 93: buy 5 units at price 5347.600100, total balance 2544.558281
day 96: buy 5 units at price 5475.300295, total balance -2930.742014
day 98, sell 5 units at price 5630.000000, investment 7.383323 %, total balance
2699.257986,
day 99, sell 5 units at price 5800.200195, investment 15.827943 %, total balance
8499.458181,
day 100, sell 5 units at price 5822.500000, investment 12.210677 %, total
balance 14321.958181,
day 101: buy 1 units at price 1138.170044, total balance 13183.788137
day 102, sell 2 units at price 2298.979980, investment 104.029136 %, total
balance 15482.768117,
day 111: buy 5 units at price 5025.499880, total balance 10457.268237
day 113: buy 5 units at price 5158.950195, total balance 5298.318042
day 114: buy 5 units at price 5032.349855, total balance 265.968187
day 116, sell 5 units at price 5125.700075, investment 1.993835 %, total balance
5391.668262,
day 118: buy 1 units at price 1007.039978, total balance 4384.628284
day 119: buy 5 units at price 5077.250060, total balance -692.621776
day 126, sell 5 units at price 5360.399780, investment 3.904856 %, total balance
4667.778004,
day 128, sell 5 units at price 5364.799805, investment 6.606257 %, total balance
10032.577809,
day 129, sell 5 units at price 5337.249755, investment 429.993831 %, total
balance 15369.827564,
day 131, sell 1 units at price 1021.179993, investment -79.887144 %, total
balance 16391.007557,
day 132: buy 1 units at price 1040.040039, total balance 15350.967518
day 135, sell 1 units at price 1037.310059, investment -0.262488 %, total
balance 16388.277577,
day 136: buy 5 units at price 5121.900025, total balance 11266.377552
day 137: buy 1 units at price 1023.719971, total balance 10242.657581
day 138: buy 5 units at price 5241.049805, total balance 5001.607776
day 139: buy 5 units at price 5273.950195, total balance -272.342419
day 141, sell 5 units at price 5413.800050, investment 5.699057 %, total balance
5141.457631,
day 142, sell 5 units at price 5487.849730, investment 436.069422 %, total
balance 10629.307361,
day 144: buy 1 units at price 1100.199951, total balance 9529.107410
day 147: buy 1 units at price 1078.589966, total balance 8450.517444
day 148: buy 5 units at price 5331.799925, total balance 3118.717519
day 150: buy 5 units at price 5348.649900, total balance -2229.932381
day 159, sell 5 units at price 5698.300170, investment 8.724404 %, total balance
3468.367789,
day 161, sell 5 units at price 5619.299925, investment 6.548218 %, total balance
9087.667714,
day 162: buy 5 units at price 5604.349975, total balance 3483.317739
day 168: buy 1 units at price 1173.459961, total balance 2309.857778
day 170, sell 5 units at price 5849.199830, investment 431.648799 %, total
balance 8159.057608,
day 171, sell 5 units at price 5788.300170, investment 436.654368 %, total
balance 13947.357778,
day 172, sell 4 units at price 4621.919920, investment -13.314078 %, total
balance 18569.277698,
day 173: buy 5 units at price 5624.050295, total balance 12945.227403
day 175: buy 5 units at price 5519.899900, total balance 7425.327503
day 176: buy 5 units at price 5571.099855, total balance 1854.227648
day 179: buy 5 units at price 5514.450075, total balance -3660.222427
day 184, sell 5 units at price 5769.500120, investment 7.868345 %, total balance
2109.277693,
day 187: buy 5 units at price 5919.299925, total balance -3810.022232
day 194, sell 5 units at price 6318.499755, investment 12.742776 %, total
balance 2508.477523,
day 195, sell 5 units at price 6341.649780, investment 440.423192 %, total
balance 8850.127303,
day 196, sell 5 units at price 6192.500000, investment 10.107479 %, total
balance 15042.627303,
day 197, sell 5 units at price 6098.699950, investment 10.485698 %, total
balance 21141.327253,
day 204: buy 5 units at price 6228.049925, total balance 14913.277328
day 205: buy 5 units at price 6245.499880, total balance 8667.777448
day 206: buy 5 units at price 6188.049925, total balance 2479.727523
day 207, sell 5 units at price 6175.050050, investment -0.850987 %, total
balance 8654.777573,
day 208, sell 5 units at price 6210.499880, investment -0.560404 %, total
balance 14865.277453,
day 209: buy 5 units at price 6071.900025, total balance 8793.377428
day 210: buy 5 units at price 6032.449950, total balance 2760.927478
day 211: buy 5 units at price 6004.799805, total balance -3243.872327
day 219, sell 5 units at price 6246.500245, investment 0.944568 %, total balance
3002.627918,
day 220, sell 5 units at price 6195.599975, investment 2.037253 %, total balance
9198.227893,
day 221, sell 5 units at price 6090.949705, investment 0.969751 %, total balance
15289.177598,
day 227: buy 1 units at price 1177.359985, total balance 14111.817613
day 229: buy 5 units at price 5876.649780, total balance 8235.167833
day 230: buy 5 units at price 5862.650145, total balance 2372.517688
day 231: buy 1 units at price 1156.050049, total balance 1216.467639
day 232: buy 1 units at price 1161.219971, total balance 55.247668
day 233, sell 5 units at price 5855.449830, investment -2.487177 %, total
balance 5910.697498,
day 234, sell 5 units at price 5934.349975, investment 404.038701 %, total
balance 11845.047473,
day 235: buy 5 units at price 5830.449830, total balance 6014.597643
day 238: buy 1 units at price 1180.489990, total balance 4834.107653
day 242, sell 5 units at price 6000.549925, investment 2.108347 %, total balance
10834.657578,
day 243, sell 5 units at price 6014.749755, investment 2.594383 %, total balance
16849.407333,
day 245, sell 4 units at price 4629.399904, investment 300.449782 %, total
balance 21478.807237,

total gained 11478.807237, total investment 114.788072 %
```


true close
predict buy
predict sell
1250
1200
1150
1100
1050
1000
0
50
100
150
200
250

```
[ ]:
```