# 14.actor-critic-agent

September 29, 2021

[1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

[2]:
```
df = pd.read_csv('../dataset/GOOG-year.csv')
df.head()
```

[2]:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 2016-11-02 | 778.200012 | 781.650024 | 763.450012 | 768.700012 | 768.700012 | 1872400 |
| 1 | 2016-11-03 | 767.250000 | 769.950012 | 759.030029 | 762.130005 | 762.130005 | 1943200 |
| 2 | 2016-11-04 | 750.659973 | 770.359985 | 750.560974 | 762.020020 | 762.020020 | 2134800 |
| 3 | 2016-11-07 | 774.500000 | 785.190002 | 772.549988 | 782.520020 | 782.520020 | 1585100 |
| 4 | 2016-11-08 | 783.400024 | 795.632996 | 780.190002 | 790.510010 | 790.510010 | 1350800 |

[3]:
```
from collections import deque
import random

class Actor:
    def __init__(self, name, input_size, output_size, size_layer):
        with tf.variable_scope(name):
            self.X = tf.placeholder(tf.float32, (None, input_size))
            feed_actor = tf.layers.dense(self.X, size_layer, activation = tf.nn.
 ↪relu)
            self.logits = tf.layers.dense(feed_actor, output_size)

class Critic:
```

```
    def __init__(self, name, input_size, output_size, size_layer,␣
↪learning_rate):
        with tf.variable_scope(name):
            self.X = tf.placeholder(tf.float32, (None, input_size))
            self.Y = tf.placeholder(tf.float32, (None, output_size))
            self.REWARD = tf.placeholder(tf.float32, (None, 1))
            feed_critic = tf.layers.dense(self.X, size_layer, activation = tf.
↪nn.relu)
            feed_critic = tf.layers.dense(feed_critic, output_size, activation␣
↪= tf.nn.relu) + self.Y
            feed_critic = tf.layers.dense(feed_critic, size_layer//2,␣
↪activation = tf.nn.relu)
            self.logits = tf.layers.dense(feed_critic, 1)
            self.cost = tf.reduce_mean(tf.square(self.REWARD - self.logits))
            self.optimizer = tf.train.AdamOptimizer(learning_rate).
↪minimize(self.cost)

class Agent:

    LEARNING_RATE = 0.001
    BATCH_SIZE = 32
    LAYER_SIZE = 256
    OUTPUT_SIZE = 3
    EPSILON = 0.5
    DECAY_RATE = 0.005
    MIN_EPSILON = 0.1
    GAMMA = 0.99
    MEMORIES = deque()
    MEMORY_SIZE = 300
    COPY = 1000
    T_COPY = 0

    def __init__(self, state_size, window_size, trend, skip):
        self.state_size = state_size
        self.window_size = window_size
        self.half_window = window_size // 2
        self.trend = trend
        self.skip = skip
        tf.reset_default_graph()
        self.actor = Actor('actor-original', self.state_size, self.OUTPUT_SIZE,␣
↪self.LAYER_SIZE)
        self.actor_target = Actor('actor-target', self.state_size, self.
↪OUTPUT_SIZE, self.LAYER_SIZE)
        self.critic = Critic('critic-original', self.state_size, self.
↪OUTPUT_SIZE, self.LAYER_SIZE, self.LEARNING_RATE)
```

```
        self.critic_target = Critic('critic-target', self.state_size, self.
↪OUTPUT_SIZE,
                                    self.LAYER_SIZE, self.LEARNING_RATE)
        self.grad_critic = tf.gradients(self.critic.logits, self.critic.Y)
        self.actor_critic_grad = tf.placeholder(tf.float32, [None, self.
↪OUTPUT_SIZE])
        weights_actor = tf.get_collection(tf.GraphKeys.TRAINABLE_VARIABLES,␣
↪scope='actor')
        self.grad_actor = tf.gradients(self.actor.logits, weights_actor, -self.
↪actor_critic_grad)
        grads = zip(self.grad_actor, weights_actor)
        self.optimizer = tf.train.AdamOptimizer(self.LEARNING_RATE).
↪apply_gradients(grads)
        self.sess = tf.InteractiveSession()
        self.sess.run(tf.global_variables_initializer())

    def _assign(self, from_name, to_name):
        from_w = tf.get_collection(tf.GraphKeys.TRAINABLE_VARIABLES,␣
↪scope=from_name)
        to_w = tf.get_collection(tf.GraphKeys.TRAINABLE_VARIABLES,␣
↪scope=to_name)
        for i in range(len(from_w)):
            assign_op = to_w[i].assign(from_w[i])
            self.sess.run(assign_op)

    def _memorize(self, state, action, reward, new_state, dead):
        self.MEMORIES.append((state, action, reward, new_state, dead))
        if len(self.MEMORIES) > self.MEMORY_SIZE:
            self.MEMORIES.popleft()

    def _select_action(self, state):
        if np.random.rand() < self.EPSILON:
            action = np.random.randint(self.OUTPUT_SIZE)
        else:
            prediction = self.sess.run(self.actor.logits, feed_dict={self.actor.
↪X:[state]})[0]
            action = np.argmax(prediction)
        return action

    def _construct_memories_and_train(self, replay):
        states = np.array([a[0] for a in replay])
        new_states = np.array([a[3] for a in replay])
        Q = self.sess.run(self.actor.logits, feed_dict={self.actor.X: states})
        Q_target = self.sess.run(self.actor_target.logits, feed_dict={self.
↪actor_target.X: states})
```

```
        grads = self.sess.run(self.grad_critic, feed_dict={self.critic.X:
↪states, self.critic.Y:Q})[0]
        self.sess.run(self.optimizer, feed_dict={self.actor.X:states, self.
↪actor_critic_grad:grads})

        rewards = np.array([a[2] for a in replay]).reshape((-1, 1))
        rewards_target = self.sess.run(self.critic_target.logits,
                                       feed_dict={self.critic_target.X:
↪new_states,self.critic_target.Y:Q_target})
        for i in range(len(replay)):
            if not replay[0][-1]:
                rewards[i] += self.GAMMA * rewards_target[i]
        cost, _ = self.sess.run([self.critic.cost, self.critic.optimizer],
                                feed_dict={self.critic.X:states, self.critic.Y:
↪Q, self.critic.REWARD:rewards})
        return cost

    def get_state(self, t):
        window_size = self.window_size + 1
        d = t - window_size + 1
        block = self.trend[d : t + 1] if d >= 0 else -d * [self.trend[0]] +␣
↪self.trend[0 : t + 1]
        res = []
        for i in range(window_size - 1):
            res.append(block[i + 1] - block[i])
        return np.array(res)

    def buy(self, initial_money):
        starting_money = initial_money
        states_sell = []
        states_buy = []
        inventory = []
        state = self.get_state(0)
        for t in range(0, len(self.trend) - 1, self.skip):
            action = self._select_action(state)
            next_state = self.get_state(t + 1)

            if action == 1 and initial_money >= self.trend[t]:
                inventory.append(self.trend[t])
                initial_money -= self.trend[t]
                states_buy.append(t)
                print('day %d: buy 1 unit at price %f, total balance %f'% (t,␣
↪self.trend[t], initial_money))

            elif action == 2 and len(inventory):
                bought_price = inventory.pop(0)
```

```
                initial_money += self.trend[t]
                states_sell.append(t)
                try:
                    invest = ((close[t] - bought_price) / bought_price) * 100
                except:
                    invest = 0
                print(
                    'day %d, sell 1 unit at price %f, investment %f %%, total␣
↪balance %f,'
                    % (t, close[t], invest, initial_money)
                )

            state = next_state
        invest = ((initial_money - starting_money) / starting_money) * 100
        total_gains = initial_money - starting_money
        return states_buy, states_sell, total_gains, invest

    def train(self, iterations, checkpoint, initial_money):
        for i in range(iterations):
            total_profit = 0
            inventory = []
            state = self.get_state(0)
            starting_money = initial_money
            for t in range(0, len(self.trend) - 1, self.skip):
                if (self.T_COPY + 1) % self.COPY == 0:
                    self._assign('actor-original', 'actor-target')
                    self._assign('critic-original', 'critic-target')

                action = self._select_action(state)
                next_state = self.get_state(t + 1)

                if action == 1 and starting_money >= self.trend[t]:
                    inventory.append(self.trend[t])
                    starting_money -= self.trend[t]

                elif action == 2 and len(inventory) > 0:
                    bought_price = inventory.pop(0)
                    total_profit += self.trend[t] - bought_price
                    starting_money += self.trend[t]

                invest = ((starting_money - initial_money) / initial_money)

                self._memorize(state, action, invest, next_state,␣
↪starting_money < initial_money)
                state = next_state
                batch_size = min(len(self.MEMORIES), self.BATCH_SIZE)
                replay = random.sample(self.MEMORIES, batch_size)
```

```
                cost = self._construct_memories_and_train(replay)
                self.T_COPY += 1
                self.EPSILON = self.MIN_EPSILON + (1.0 - self.MIN_EPSILON) * np.
↪exp(-self.DECAY_RATE * i)
            if (i+1) % checkpoint == 0:
                print('epoch: %d, total rewards: %f.3, cost: %f, total money:␣
↪%f'%(i + 1, total_profit, cost,
                                                                              ␣
↪  starting_money))
```

[4]:
```
close = df.Close.values.tolist()
initial_money = 10000
window_size = 30
skip = 1
batch_size = 32
agent = Agent(state_size = window_size,
              window_size = window_size,
              trend = close,
              skip = skip)
agent.train(iterations = 200, checkpoint = 10, initial_money = initial_money)
```

```
epoch: 10, total rewards: 1539.185237.3, cost: 2.181347, total money:
1684.395196
epoch: 20, total rewards: 1308.335026.3, cost: 658.992737, total money:
11308.335026
epoch: 30, total rewards: 810.315002.3, cost: 19406.357422, total money:
5871.594971
epoch: 40, total rewards: 380.889899.3, cost: 436790400.000000, total money:
7327.869879
epoch: 50, total rewards: 676.170224.3, cost: 27570524160.000000, total money:
10676.170224
epoch: 60, total rewards: 796.770199.3, cost: 935274741760.000000, total money:
10796.770199
epoch: 70, total rewards: 47.440366.3, cost: 8344191369216.000000, total money:
7043.150388
epoch: 80, total rewards: 450.169980.3, cost: 88121093914624.000000, total
money: 6472.479916
epoch: 90, total rewards: 443.664980.3, cost: 675454474256384.000000, total
money: 9427.024965
epoch: 100, total rewards: 350.460142.3, cost: 1153362061950976.000000, total
money: 10350.460142
epoch: 110, total rewards: 247.584961.3, cost: 6317238688677888.000000, total
money: 9230.944946
epoch: 120, total rewards: 138.510132.3, cost: 3956869119726321664.000000, total
money: 8102.600097
epoch: 130, total rewards: 410.025086.3, cost: 2205253088434978816.000000, total
money: 10410.025086
```

```
epoch: 140, total rewards: 513.814999.3, cost: 5849743807884558336.000000, total
money: 9497.174984
epoch: 150, total rewards: 876.734991.3, cost: 25442419893862400.000000, total
money: 9860.094976
epoch: 160, total rewards: 216.929627.3, cost: 73146239398445056.000000, total
money: 9244.369629
epoch: 170, total rewards: 26.000066.3, cost: 210379489706770432.000000, total
money: 7992.250066
epoch: 180, total rewards: 230.090269.3, cost: 378469838063927296.000000, total
money: 8194.180234
epoch: 190, total rewards: 31.099796.3, cost: 1333389845631860736.000000, total
money: 6978.079776
epoch: 200, total rewards: 158.599487.3, cost: 459357028892629008384.000000,
total money: 10158.599487
```

[5]:
```
states_buy, states_sell, total_gains, invest = agent.buy(initial_money =␣
↪initial_money)
```

```
day 3: buy 1 unit at price 782.520020, total balance 9217.479980
day 4: buy 1 unit at price 790.510010, total balance 8426.969970
day 5, sell 1 unit at price 785.309998, investment 0.356538 %, total balance
9212.279968,
day 6, sell 1 unit at price 762.559998, investment -3.535694 %, total balance
9974.839966,
day 16: buy 1 unit at price 761.679993, total balance 9213.159973
day 17: buy 1 unit at price 768.239990, total balance 8444.919983
day 18, sell 1 unit at price 770.840027, investment 1.202609 %, total balance
9215.760010,
day 19, sell 1 unit at price 758.039978, investment -1.327712 %, total balance
9973.799988,
day 20: buy 1 unit at price 747.919983, total balance 9225.880005
day 21: buy 1 unit at price 750.500000, total balance 8475.380005
day 22, sell 1 unit at price 762.520020, investment 1.952085 %, total balance
9237.900025,
day 24, sell 1 unit at price 771.190002, investment 2.756829 %, total balance
10009.090027,
day 25: buy 1 unit at price 776.419983, total balance 9232.670044
day 26, sell 1 unit at price 789.289978, investment 1.657607 %, total balance
10021.960022,
day 27: buy 1 unit at price 789.270020, total balance 9232.690002
day 28: buy 1 unit at price 796.099976, total balance 8436.590026
day 31, sell 1 unit at price 790.799988, investment 0.193846 %, total balance
9227.390014,
day 32, sell 1 unit at price 794.200012, investment -0.238659 %, total balance
10021.590026,
day 33: buy 1 unit at price 796.419983, total balance 9225.170043
day 34, sell 1 unit at price 794.559998, investment -0.233543 %, total balance
10019.730041,
```

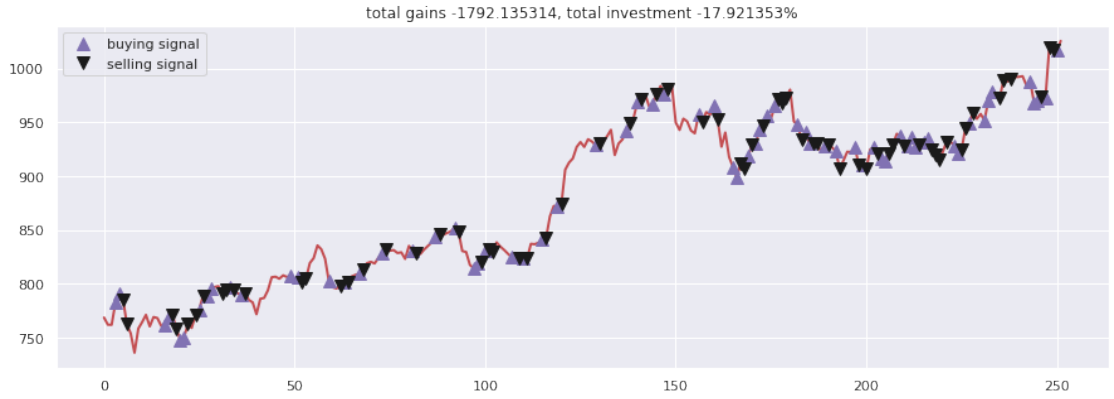
day 36: buy 1 unit at price 789.909973, total balance 9229.820068
day 37, sell 1 unit at price 791.549988, investment 0.207620 %, total balance 10021.370056,
day 49: buy 1 unit at price 807.880005, total balance 9213.490051
day 51: buy 1 unit at price 806.070007, total balance 8407.420044
day 52, sell 1 unit at price 802.174988, investment -0.706171 %, total balance 9209.595032,
day 53, sell 1 unit at price 805.020020, investment -0.130260 %, total balance 10014.615052,
day 59: buy 1 unit at price 802.320007, total balance 9212.295045
day 62, sell 1 unit at price 798.530029, investment -0.472377 %, total balance 10010.825074,
day 63: buy 1 unit at price 801.489990, total balance 9209.335084
day 64, sell 1 unit at price 801.340027, investment -0.018711 %, total balance 10010.675111,
day 67: buy 1 unit at price 809.559998, total balance 9201.115113
day 68, sell 1 unit at price 813.669983, investment 0.507681 %, total balance 10014.785096,
day 73: buy 1 unit at price 828.070007, total balance 9186.715089
day 74, sell 1 unit at price 831.659973, investment 0.433534 %, total balance 10018.375062,
day 81: buy 1 unit at price 830.630005, total balance 9187.745057
day 82, sell 1 unit at price 829.080017, investment -0.186604 %, total balance 10016.825074,
day 87: buy 1 unit at price 843.250000, total balance 9173.575074
day 88, sell 1 unit at price 845.539978, investment 0.271566 %, total balance 10019.115052,
day 92: buy 1 unit at price 852.119995, total balance 9166.995057
day 93, sell 1 unit at price 848.400024, investment -0.436555 %, total balance 10015.395081,
day 97: buy 1 unit at price 814.429993, total balance 9200.965088
day 98: buy 1 unit at price 819.510010, total balance 8381.455078
day 99, sell 1 unit at price 820.919983, investment 0.796875 %, total balance 9202.375061,
day 100: buy 1 unit at price 831.409973, total balance 8370.965088
day 101, sell 1 unit at price 831.500000, investment 1.463068 %, total balance 9202.465088,
day 102, sell 1 unit at price 829.559998, investment -0.222511 %, total balance 10032.025086,
day 107: buy 1 unit at price 824.669983, total balance 9207.355103
day 109, sell 1 unit at price 823.349976, investment -0.160065 %, total balance 10030.705079,
day 110: buy 1 unit at price 824.320007, total balance 9206.385072
day 111, sell 1 unit at price 823.559998, investment -0.092198 %, total balance 10029.945070,
day 115: buy 1 unit at price 841.650024, total balance 9188.295046
day 116, sell 1 unit at price 843.190002, investment 0.182971 %, total balance 10031.485048,

day 119: buy 1 unit at price 871.729980, total balance 9159.755068
day 120, sell 1 unit at price 874.250000, investment 0.289083 %, total balance 10034.005068,
day 129: buy 1 unit at price 928.780029, total balance 9105.225039
day 130, sell 1 unit at price 930.599976, investment 0.195950 %, total balance 10035.825015,
day 137: buy 1 unit at price 941.859985, total balance 9093.965030
day 138, sell 1 unit at price 948.820007, investment 0.738966 %, total balance 10042.785037,
day 140: buy 1 unit at price 969.539978, total balance 9073.245059
day 141, sell 1 unit at price 971.469971, investment 0.199063 %, total balance 10044.715030,
day 144: buy 1 unit at price 966.950012, total balance 9077.765018
day 145, sell 1 unit at price 975.599976, investment 0.894562 %, total balance 10053.364994,
day 147: buy 1 unit at price 976.570007, total balance 9076.794987
day 148, sell 1 unit at price 980.940002, investment 0.447484 %, total balance 10057.734989,
day 156: buy 1 unit at price 957.369995, total balance 9100.364994
day 157, sell 1 unit at price 950.630005, investment -0.704011 %, total balance 10050.994999,
day 160: buy 1 unit at price 965.590027, total balance 9085.404972
day 161, sell 1 unit at price 952.270020, investment -1.379468 %, total balance 10037.674992,
day 165: buy 1 unit at price 908.729980, total balance 9128.945012
day 166: buy 1 unit at price 898.700012, total balance 8230.245000
day 167, sell 1 unit at price 911.710022, investment 0.327935 %, total balance 9141.955022,
day 168, sell 1 unit at price 906.690002, investment 0.889061 %, total balance 10048.645024,
day 169: buy 1 unit at price 918.590027, total balance 9130.054997
day 170, sell 1 unit at price 928.799988, investment 1.111482 %, total balance 10058.854985,
day 171: buy 1 unit at price 930.090027, total balance 9128.764958
day 172: buy 1 unit at price 943.830017, total balance 8184.934941
day 173, sell 1 unit at price 947.159973, investment 1.835300 %, total balance 9132.094914,
day 174: buy 1 unit at price 955.989990, total balance 8176.104924
day 176: buy 1 unit at price 965.400024, total balance 7210.704900
day 177, sell 1 unit at price 970.890015, investment 2.867041 %, total balance 8181.594915,
day 178, sell 1 unit at price 968.150024, investment 1.271983 %, total balance 9149.744939,
day 179, sell 1 unit at price 972.919983, investment 0.778947 %, total balance 10122.664922,
day 182: buy 1 unit at price 947.799988, total balance 9174.864934
day 183, sell 1 unit at price 934.090027, investment -1.446504 %, total balance 10108.954961,

```
day 184: buy 1 unit at price 941.530029, total balance 9167.424932
day 185: buy 1 unit at price 930.500000, total balance 8236.924932
day 186, sell 1 unit at price 930.830017, investment -1.136449 %, total balance
9167.754949,
day 187, sell 1 unit at price 930.390015, investment -0.011820 %, total balance
10098.144964,
day 189: buy 1 unit at price 927.960022, total balance 9170.184942
day 190, sell 1 unit at price 929.359985, investment 0.150865 %, total balance
10099.544927,
day 192: buy 1 unit at price 922.900024, total balance 9176.644903
day 193, sell 1 unit at price 907.239990, investment -1.696829 %, total balance
10083.884893,
day 197: buy 1 unit at price 926.960022, total balance 9156.924871
day 198, sell 1 unit at price 910.979980, investment -1.723919 %, total balance
10067.904851,
day 199: buy 1 unit at price 910.669983, total balance 9157.234868
day 200, sell 1 unit at price 906.659973, investment -0.440336 %, total balance
10063.894841,
day 202: buy 1 unit at price 927.000000, total balance 9136.894841
day 203, sell 1 unit at price 921.280029, investment -0.617041 %, total balance
10058.174870,
day 204: buy 1 unit at price 915.890015, total balance 9142.284855
day 205: buy 1 unit at price 913.809998, total balance 8228.474857
day 206, sell 1 unit at price 921.289978, investment 0.589586 %, total balance
9149.764835,
day 207, sell 1 unit at price 929.570007, investment 1.724648 %, total balance
10079.334842,
day 209: buy 1 unit at price 937.340027, total balance 9141.994815
day 210, sell 1 unit at price 928.450012, investment -0.948430 %, total balance
10070.444827,
day 211: buy 1 unit at price 927.809998, total balance 9142.634829
day 212: buy 1 unit at price 935.950012, total balance 8206.684817
day 213: buy 1 unit at price 926.500000, total balance 7280.184817
day 214, sell 1 unit at price 929.080017, investment 0.136884 %, total balance
8209.264834,
day 215: buy 1 unit at price 932.070007, total balance 7277.194827
day 216: buy 1 unit at price 935.090027, total balance 6342.104800
day 217, sell 1 unit at price 925.109985, investment -1.158184 %, total balance
7267.214785,
day 218, sell 1 unit at price 920.289978, investment -0.670267 %, total balance
8187.504763,
day 219, sell 1 unit at price 915.000000, investment -1.831408 %, total balance
9102.504763,
day 221, sell 1 unit at price 931.580017, investment -0.375366 %, total balance
10034.084780,
day 223: buy 1 unit at price 928.530029, total balance 9105.554751
day 224: buy 1 unit at price 920.969971, total balance 8184.584780
day 225, sell 1 unit at price 924.859985, investment -0.395253 %, total balance
```

```
9109.444765,
day 226, sell 1 unit at price 944.489990, investment 2.553831 %, total balance
10053.934755,
day 227: buy 1 unit at price 949.500000, total balance 9104.434755
day 228, sell 1 unit at price 959.109985, investment 1.012110 %, total balance
10063.544740,
day 231: buy 1 unit at price 951.679993, total balance 9111.864747
day 232: buy 1 unit at price 969.960022, total balance 8141.904725
day 233: buy 1 unit at price 978.890015, total balance 7163.014710
day 235, sell 1 unit at price 972.599976, investment 2.198216 %, total balance
8135.614686,
day 236, sell 1 unit at price 989.250000, investment 1.988739 %, total balance
9124.864686,
day 238, sell 1 unit at price 989.679993, investment 1.102267 %, total balance
10114.544679,
day 243: buy 1 unit at price 988.200012, total balance 9126.344667
day 244: buy 1 unit at price 968.450012, total balance 8157.894655
day 245: buy 1 unit at price 970.539978, total balance 7187.354677
day 246, sell 1 unit at price 973.330017, investment -1.504756 %, total balance
8160.684694,
day 247: buy 1 unit at price 972.559998, total balance 7188.124696
day 248, sell 1 unit at price 1019.270020, investment 5.247561 %, total balance
8207.394716,
day 249, sell 1 unit at price 1017.109985, investment 4.798361 %, total balance
9224.504701,
day 250: buy 1 unit at price 1016.640015, total balance 8207.864686
```

[6]:
```
fig = plt.figure(figsize = (15,5))
plt.plot(close, color='r', lw=2.)
plt.plot(close, '^', markersize=10, color='m', label = 'buying signal',␣
 ↪markevery = states_buy)
plt.plot(close, 'v', markersize=10, color='k', label = 'selling signal',␣
 ↪markevery = states_sell)
plt.title('total gains %f, total investment %f%%'%(total_gains, invest))
plt.legend()
plt.show()
```


total gains -1792.135314, total investment -17.921353%
buying signal
selling signal
1000
950
900
850
800
750
0
50
100
150
200
250

[ ]: