

9.double-recurrent-q-learning-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	\
0	2016-11-02	778.200012	781.650024	763.450012	768.700012	768.700012	
1	2016-11-03	767.250000	769.950012	759.030029	762.130005	762.130005	
2	2016-11-04	750.659973	770.359985	750.560974	762.020020	762.020020	
3	2016-11-07	774.500000	785.190002	772.549988	782.520020	782.520020	
4	2016-11-08	783.400024	795.632996	780.190002	790.510010	790.510010	

	Volume
0	1872400
1	1943200
2	2134800
3	1585100
4	1350800

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

class Model:
    def __init__(self, input_size, output_size, layer_size, learning_rate,
        ↪name):
        with tf.variable_scope(name):
            self.X = tf.placeholder(tf.float32, (None, None, input_size))
            self.Y = tf.placeholder(tf.float32, (None, output_size))
            cell = tf.nn.rnn_cell.LSTMCell(layer_size, state_is_tuple = False)
            self.hidden_layer = tf.placeholder(tf.float32, (None, 2 *
        ↪layer_size))
```

```

        self.rnn, self.last_state = tf.nn.dynamic_rnn(inputs=self.
↪X, cell=cell,
                                                    dtype=tf.float32,
                                                    initial_state=self.
↪hidden_layer)
        self.logits = tf.layers.dense(self.rnn[:, -1], output_size)
        self.cost = tf.reduce_sum(tf.square(self.Y - self.logits))
        self.optimizer = tf.train.AdamOptimizer(learning_rate = ↪
↪learning_rate).minimize(self.cost)

class Agent:

    LEARNING_RATE = 0.003
    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()
    COPY = 1000
    T_COPY = 0
    MEMORY_SIZE = 300

    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.INITIAL_FEATURES = np.zeros((4, self.state_size))
        self.model = Model(self.state_size, self.OUTPUT_SIZE, self.LAYER_SIZE, ↪
↪self.LEARNING_RATE,
                                'real_model')
        self.model_negative = Model(self.state_size, self.OUTPUT_SIZE, self.
↪LAYER_SIZE, self.LEARNING_RATE,
                                'negative_model')
        self.sess = tf.InteractiveSession()
        self.sess.run(tf.global_variables_initializer())
        self.trainable = tf.trainable_variables()

    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, rnn_state):
        self.MEMORIES.append((state, action, reward, new_state, dead,
↪rnn_state))
        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:
            action = self.get_predicted_action([state])
        return action

    def _construct_memories(self, replay):
        states = np.array([a[0] for a in replay])
        new_states = np.array([a[3] for a in replay])
        init_values = np.array([a[-1] for a in replay])
        Q = self.sess.run(self.model.logits, feed_dict={self.model.X:states,
                                                         self.model.hidden_layer:
↪init_values})
        Q_new = self.sess.run(self.model.logits, feed_dict={self.model.X:
↪new_states,
                                                         self.model.hidden_layer:
↪init_values})
        Q_new_negative = self.sess.run(self.model_negative.logits,
                                       feed_dict={self.model_negative.X:new_states,
                                                  self.model_negative.hidden_layer:
↪init_values})
        replay_size = len(replay)
        X = np.empty((replay_size, 4, self.state_size))
        Y = np.empty((replay_size, self.OUTPUT_SIZE))
        INIT_VAL = np.empty((replay_size, 2 * self.LAYER_SIZE))
        for i in range(replay_size):
            state_r, action_r, reward_r, new_state_r, dead_r, rnn_memory =
↪replay[i]
            target = Q[i]
            target[action_r] = reward_r
            if not dead_r:
                target[action_r] += self.GAMMA * Q_new_negative[i, np.
↪argmax(Q_new[i])]

```

```

        X[i] = state_r
        Y[i] = target
        INIT_VAL[i] = rnn_memory
    return X, Y, INIT_VAL

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)
    init_value = np.zeros((1, 2 * self.LAYER_SIZE))
    for k in range(self.INITIAL_FEATURES.shape[0]):
        self.INITIAL_FEATURES[k,:] = state
    for t in range(0, len(self.trend) - 1, self.skip):
        action, last_state = self.sess.run([self.model.logits, self.model.
↪last_state],
                                           feed_dict={self.model.X:[self.
↪INITIAL_FEATURES],
                                           self.model.
↪hidden_layer:init_value})
        action, init_value = np.argmax(action[0]), last_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)
        )

        new_state = np.append([self.get_state(t + 1)], self.
↪INITIAL_FEATURES[:3, :], axis = 0)
        self.INITIAL_FEATURES = new_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
        init_value = np.zeros((1, 2 * self.LAYER_SIZE))
        for k in range(self.INITIAL_FEATURES.shape[0]):
            self.INITIAL_FEATURES[k,:] = state
        for t in range(0, len(self.trend) - 1, self.skip):
            if (self.T_COPY + 1) % self.COPY == 0:
                self._assign('real_model', 'negative_model')

            if np.random.rand() < self.EPSILON:
                action = np.random.randint(self.OUTPUT_SIZE)
            else:
                action, last_state = self.sess.run([self.model.logits,
                                                    self.model.last_state],
                                                    feed_dict={self.model.X:[self.
↪INITIAL_FEATURES],
                                                            self.model.
↪hidden_layer:init_value})
                action, init_value = np.argmax(action[0]), last_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)
        new_state = np.append([self.get_state(t + 1)], self.
→ INITIAL_FEATURES[:3, :], axis = 0)

        self._memorize(self.INITIAL_FEATURES, action, invest, new_state,
                        starting_money < initial_money, init_value[0])
        self.INITIAL_FEATURES = new_state
        batch_size = min(len(self.MEMORIES), self.BATCH_SIZE)
        replay = random.sample(self.MEMORIES, batch_size)
        X, Y, INIT_VAL = self._construct_memories(replay)

        cost, _ = self.sess.run([self.model.cost, self.model.optimizer],
                                feed_dict={self.model.X: X, self.model.
→ Y:Y,

                                                self.model.hidden_layer:
→ INIT_VAL})

        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)

```

WARNING:tensorflow:<tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x7fb85fd10940>: Using a concatenated state is slower and will soon be deprecated. Use state_is_tuple=True.

WARNING:tensorflow:<tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x7fb85f9de7b8>: Using a concatenated state is slower and will soon be deprecated. Use state_is_tuple=True.

epoch: 10, total rewards: 1305.274912.3, cost: 0.402263, total money: 777.284860

epoch: 20, total rewards: 582.070375.3, cost: 0.782595, total money: 804.650331
 epoch: 30, total rewards: 420.380369.3, cost: 1.481925, total money: 80.210326
 epoch: 40, total rewards: 1502.554748.3, cost: 0.343374, total money:
 2823.564757
 epoch: 50, total rewards: 589.170222.3, cost: 0.370314, total money: 6597.640193
 epoch: 60, total rewards: 1069.864985.3, cost: 0.733583, total money:
 10052.755000
 epoch: 70, total rewards: 900.360168.3, cost: 0.154633, total money: 8866.610168
 epoch: 80, total rewards: 625.559509.3, cost: 0.573019, total money: 9652.999511
 epoch: 90, total rewards: 966.905028.3, cost: 0.080430, total money: 6971.785033
 epoch: 100, total rewards: 784.169802.3, cost: 0.568819, total money:
 10784.169802
 epoch: 110, total rewards: 658.149963.3, cost: 0.052230, total money:
 9641.509948
 epoch: 120, total rewards: 615.210201.3, cost: 0.802322, total money:
 9595.940181
 epoch: 130, total rewards: 623.289978.3, cost: 0.278659, total money:
 10623.289978
 epoch: 140, total rewards: 595.960078.3, cost: 0.094435, total money:
 10595.960078
 epoch: 150, total rewards: 594.979550.3, cost: 0.360762, total money:
 1819.289547
 epoch: 160, total rewards: 794.614687.3, cost: 1.058314, total money:
 3118.034730
 epoch: 170, total rewards: 1225.854981.3, cost: 0.226553, total money:
 5322.584961
 epoch: 180, total rewards: 1099.610169.3, cost: 0.275357, total money:
 6189.200135
 epoch: 190, total rewards: 857.554813.3, cost: 0.417154, total money:
 7946.004825
 epoch: 200, total rewards: 1049.100096.3, cost: 0.839669, total money:
 3317.970090

```
[5]: states_buy, states_sell, total_gains, invest = agent.buy(initial_money = 10000,
    ↪ initial_money)
```

day 0: buy 1 unit at price 768.700012, total balance 9231.299988
 day 1, sell 1 unit at price 762.130005, investment -0.854691 %, total balance
 9993.429993,
 day 3: buy 1 unit at price 782.520020, total balance 9210.909973
 day 4, sell 1 unit at price 790.510010, investment 1.021059 %, total balance
 10001.419983,
 day 5: buy 1 unit at price 785.309998, total balance 9216.109985
 day 6, sell 1 unit at price 762.559998, investment -2.896945 %, total balance
 9978.669983,
 day 7: buy 1 unit at price 754.020020, total balance 9224.649963
 day 8, sell 1 unit at price 736.080017, investment -2.379248 %, total balance
 9960.729980,

day 13: buy 1 unit at price 769.200012, total balance 9191.529968
 day 16, sell 1 unit at price 761.679993, investment -0.977642 %, total balance 9953.209961,
 day 19: buy 1 unit at price 758.039978, total balance 9195.169983
 day 20, sell 1 unit at price 747.919983, investment -1.335021 %, total balance 9943.089966,
 day 24: buy 1 unit at price 771.190002, total balance 9171.899964
 day 28: buy 1 unit at price 796.099976, total balance 8375.799988
 day 29: buy 1 unit at price 797.070007, total balance 7578.729981
 day 31: buy 1 unit at price 790.799988, total balance 6787.929993
 day 32, sell 1 unit at price 794.200012, investment 2.983702 %, total balance 7582.130005,
 day 33, sell 1 unit at price 796.419983, investment 0.040197 %, total balance 8378.549988,
 day 35, sell 1 unit at price 791.260010, investment -0.728919 %, total balance 9169.809998,
 day 36, sell 1 unit at price 789.909973, investment -0.112546 %, total balance 9959.719971,
 day 38: buy 1 unit at price 785.049988, total balance 9174.669983
 day 41: buy 1 unit at price 786.140015, total balance 8388.529968
 day 42, sell 1 unit at price 786.900024, investment 0.235658 %, total balance 9175.429992,
 day 43, sell 1 unit at price 794.020020, investment 1.002367 %, total balance 9969.450012,
 day 44: buy 1 unit at price 806.150024, total balance 9163.299988
 day 46, sell 1 unit at price 804.789978, investment -0.168709 %, total balance 9968.089966,
 day 47: buy 1 unit at price 807.909973, total balance 9160.179993
 day 49, sell 1 unit at price 807.880005, investment -0.003709 %, total balance 9968.059998,
 day 51: buy 1 unit at price 806.070007, total balance 9161.989991
 day 52: buy 1 unit at price 802.174988, total balance 8359.815003
 day 54: buy 1 unit at price 819.309998, total balance 7540.505005
 day 55, sell 1 unit at price 823.869995, investment 2.208243 %, total balance 8364.375000,
 day 56: buy 1 unit at price 835.669983, total balance 7528.705017
 day 57: buy 1 unit at price 832.150024, total balance 6696.554993
 day 58, sell 1 unit at price 823.309998, investment 2.634713 %, total balance 7519.864991,
 day 59, sell 1 unit at price 802.320007, investment -2.073695 %, total balance 8322.184998,
 day 61, sell 1 unit at price 795.695007, investment -4.783584 %, total balance 9117.880005,
 day 62, sell 1 unit at price 798.530029, investment -4.040136 %, total balance 9916.410034,
 day 68: buy 1 unit at price 813.669983, total balance 9102.740051
 day 69, sell 1 unit at price 819.239990, investment 0.684554 %, total balance 9921.980041,

day 76: buy 1 unit at price 831.330017, total balance 9090.650024
 day 77: buy 1 unit at price 828.640015, total balance 8262.010009
 day 79, sell 1 unit at price 823.210022, investment -0.976747 %, total balance 9085.220031,
 day 81, sell 1 unit at price 830.630005, investment 0.240151 %, total balance 9915.850036,
 day 86: buy 1 unit at price 838.679993, total balance 9077.170043
 day 88: buy 1 unit at price 845.539978, total balance 8231.630065
 day 89, sell 1 unit at price 845.619995, investment 0.827491 %, total balance 9077.250060,
 day 91, sell 1 unit at price 848.780029, investment 0.383193 %, total balance 9926.030089,
 day 95: buy 1 unit at price 829.590027, total balance 9096.440062
 day 96, sell 1 unit at price 817.580017, investment -1.447704 %, total balance 9914.020079,
 day 97: buy 1 unit at price 814.429993, total balance 9099.590086
 day 101: buy 1 unit at price 831.500000, total balance 8268.090086
 day 102: buy 1 unit at price 829.559998, total balance 7438.530088
 day 104, sell 1 unit at price 834.570007, investment 2.472897 %, total balance 8273.100095,
 day 105, sell 1 unit at price 831.409973, investment -0.010827 %, total balance 9104.510068,
 day 106, sell 1 unit at price 827.880005, investment -0.202516 %, total balance 9932.390073,
 day 108: buy 1 unit at price 824.729980, total balance 9107.660093
 day 109, sell 1 unit at price 823.349976, investment -0.167328 %, total balance 9931.010069,
 day 114: buy 1 unit at price 838.210022, total balance 9092.800047
 day 117, sell 1 unit at price 862.760010, investment 2.928859 %, total balance 9955.560057,
 day 121: buy 1 unit at price 905.960022, total balance 9049.600035
 day 122: buy 1 unit at price 912.570007, total balance 8137.030028
 day 124: buy 1 unit at price 927.039978, total balance 7209.990050
 day 125: buy 1 unit at price 931.659973, total balance 6278.330077
 day 130, sell 1 unit at price 930.599976, investment 2.719762 %, total balance 7208.930053,
 day 131, sell 1 unit at price 932.219971, investment 2.153256 %, total balance 8141.150024,
 day 132, sell 1 unit at price 937.080017, investment 1.083021 %, total balance 9078.230041,
 day 133, sell 1 unit at price 943.000000, investment 1.217185 %, total balance 10021.230041,
 day 137: buy 1 unit at price 941.859985, total balance 9079.370056
 day 138, sell 1 unit at price 948.820007, investment 0.738966 %, total balance 10028.190063,
 day 142: buy 1 unit at price 975.880005, total balance 9052.310058
 day 143: buy 1 unit at price 964.859985, total balance 8087.450073
 day 144: buy 1 unit at price 966.950012, total balance 7120.500061

day 145, sell 1 unit at price 975.599976, investment -0.028695 %, total balance 8096.100037,
 day 146, sell 1 unit at price 983.679993, investment 1.950543 %, total balance 9079.780030,
 day 147: buy 1 unit at price 976.570007, total balance 8103.210023
 day 148: buy 1 unit at price 980.940002, total balance 7122.270021
 day 150, sell 1 unit at price 949.830017, investment -1.770515 %, total balance 8072.100038,
 day 151: buy 1 unit at price 942.900024, total balance 7129.200014
 day 152, sell 1 unit at price 953.400024, investment -2.372588 %, total balance 8082.600038,
 day 153: buy 1 unit at price 950.760010, total balance 7131.840028
 day 154, sell 1 unit at price 942.309998, investment -3.938060 %, total balance 8074.150026,
 day 155, sell 1 unit at price 939.780029, investment -0.330894 %, total balance 9013.930055,
 day 156, sell 1 unit at price 957.369995, investment 0.695232 %, total balance 9971.300050,
 day 159: buy 1 unit at price 957.090027, total balance 9014.210023
 day 160: buy 1 unit at price 965.590027, total balance 8048.619996
 day 161, sell 1 unit at price 952.270020, investment -0.503611 %, total balance 9000.890016,
 day 162: buy 1 unit at price 927.330017, total balance 8073.559999
 day 163: buy 1 unit at price 940.489990, total balance 7133.070009
 day 165: buy 1 unit at price 908.729980, total balance 6224.340029
 day 167: buy 1 unit at price 911.710022, total balance 5312.630007
 day 169, sell 1 unit at price 918.590027, investment -4.867490 %, total balance 6231.220034,
 day 170, sell 1 unit at price 928.799988, investment 0.158516 %, total balance 7160.020022,
 day 173, sell 1 unit at price 947.159973, investment 0.709203 %, total balance 8107.179995,
 day 174: buy 1 unit at price 955.989990, total balance 7151.190005
 day 175: buy 1 unit at price 953.419983, total balance 6197.770022
 day 176: buy 1 unit at price 965.400024, total balance 5232.369998
 day 177, sell 1 unit at price 970.890015, investment 6.840320 %, total balance 6203.260013,
 day 178: buy 1 unit at price 968.150024, total balance 5235.109989
 day 179: buy 1 unit at price 972.919983, total balance 4262.190006
 day 180: buy 1 unit at price 980.340027, total balance 3281.849979
 day 181, sell 1 unit at price 950.700012, investment 4.276578 %, total balance 4232.549991,
 day 182, sell 1 unit at price 947.799988, investment -0.856704 %, total balance 5180.349979,
 day 184, sell 1 unit at price 941.530029, investment -1.247085 %, total balance 6121.880008,
 day 185, sell 1 unit at price 930.500000, investment -3.615084 %, total balance 7052.380008,

day 186: buy 1 unit at price 930.830017, total balance 6121.549991
 day 190: buy 1 unit at price 929.359985, total balance 5192.190006
 day 191, sell 1 unit at price 926.789978, investment -4.272070 %, total balance 6118.979984,
 day 192, sell 1 unit at price 922.900024, investment -5.141220 %, total balance 7041.880008,
 day 193, sell 1 unit at price 907.239990, investment -7.456600 %, total balance 7949.119998,
 day 196: buy 1 unit at price 922.219971, total balance 7026.900027
 day 198: buy 1 unit at price 910.979980, total balance 6115.920047
 day 199, sell 1 unit at price 910.669983, investment -2.165813 %, total balance 7026.590030,
 day 200, sell 1 unit at price 906.659973, investment -2.442542 %, total balance 7933.250003,
 day 201, sell 1 unit at price 924.690002, investment 0.267835 %, total balance 8857.940005,
 day 204, sell 1 unit at price 915.890015, investment 0.538984 %, total balance 9773.830020,
 day 205: buy 1 unit at price 913.809998, total balance 8860.020022
 day 206, sell 1 unit at price 921.289978, investment 0.818549 %, total balance 9781.310000,
 day 209: buy 1 unit at price 937.340027, total balance 8843.969973
 day 210: buy 1 unit at price 928.450012, total balance 7915.519961
 day 211, sell 1 unit at price 927.809998, investment -1.016710 %, total balance 8843.329959,
 day 212: buy 1 unit at price 935.950012, total balance 7907.379947
 day 214, sell 1 unit at price 929.080017, investment 0.067856 %, total balance 8836.459964,
 day 216, sell 1 unit at price 935.090027, investment -0.091884 %, total balance 9771.549991,
 day 217: buy 1 unit at price 925.109985, total balance 8846.440006
 day 219, sell 1 unit at price 915.000000, investment -1.092841 %, total balance 9761.440006,
 day 220: buy 1 unit at price 921.809998, total balance 8839.630008
 day 221: buy 1 unit at price 931.580017, total balance 7908.049991
 day 222: buy 1 unit at price 932.450012, total balance 6975.599979
 day 226, sell 1 unit at price 944.489990, investment 2.460376 %, total balance 7920.089969,
 day 227: buy 1 unit at price 949.500000, total balance 6970.589969
 day 229, sell 1 unit at price 953.270020, investment 2.328303 %, total balance 7923.859989,
 day 230: buy 1 unit at price 957.789978, total balance 6966.070011
 day 231: buy 1 unit at price 951.679993, total balance 6014.390018
 day 232: buy 1 unit at price 969.960022, total balance 5044.429996
 day 233, sell 1 unit at price 978.890015, investment 4.980428 %, total balance 6023.320011,
 day 234: buy 1 unit at price 977.000000, total balance 5046.320011
 day 237, sell 1 unit at price 987.830017, investment 4.036863 %, total balance

```

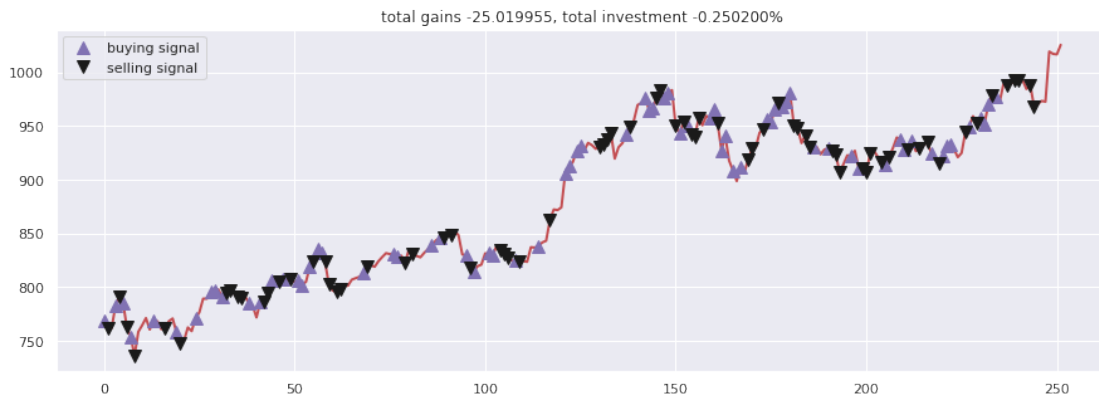
6034.150028,
day 239, sell 1 unit at price 992.000000, investment 3.571767 %, total balance
7026.150028,
day 240, sell 1 unit at price 992.179993, investment 4.255632 %, total balance
8018.330021,
day 243, sell 1 unit at price 988.200012, investment 1.880489 %, total balance
9006.530033,
day 244, sell 1 unit at price 968.450012, investment -0.875127 %, total balance
9974.980045,

```

```

[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()

```



```

[ ]:

```