

## 15.actor-critic-duel-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/G00G-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 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)

            tensor_action, tensor_validation = tf.split(feed_actor,2,1)
            feed_action = tf.layers.dense(tensor_action, output_size)
            feed_validation = tf.layers.dense(tensor_validation, 1)
            self.logits = feed_validation + tf.subtract(feed_action,
```

```

tf.
↪reduce_mean(feed_action,axis=1,keep_dims=True))

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)
            tensor_action, tensor_validation = tf.split(feed_critic,2,1)
            feed_action = tf.layers.dense(tensor_action, output_size)
            feed_validation = tf.layers.dense(tensor_validation, 1)
            feed_critic = feed_validation + tf.subtract(feed_action,tf.
↪reduce_mean(feed_action,axis=1,keep_dims=True))
            feed_critic = tf.nn.relu(feed_critic) + 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)
        batch_size = min(len(self.MEMORIES), self.BATCH_SIZE)
        state = next_state
        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)

```

WARNING:tensorflow:From <ipython-input-3-a50a3d0b4e36>:13: calling reduce\_mean (from tensorflow.python.ops.math\_ops) with keep\_dims is deprecated and will be removed in a future version.

Instructions for updating:

keep\_dims is deprecated, use keepdims instead

epoch: 10, total rewards: 707.200200.3, cost: 0.405626, total money: 9715.020207

epoch: 20, total rewards: 1598.640143.3, cost: 30.734631, total money:  
10581.530158

epoch: 30, total rewards: 1271.279733.3, cost: 465.966644, total money:  
10254.169748

epoch: 40, total rewards: 611.054993.3, cost: 38.079464, total money:  
2818.014953

epoch: 50, total rewards: 1098.115172.3, cost: 71481.406250, total money:  
1453.295102

epoch: 60, total rewards: 575.370237.3, cost: 45955692.000000, total money:  
9558.260252

epoch: 70, total rewards: 1020.545110.3, cost: 244974075904.000000, total money:  
10003.435125

epoch: 80, total rewards: 824.555359.3, cost: 62751015698432.000000, total  
money: 4025.125366

epoch: 90, total rewards: 182.215205.3, cost: 3949580517376.000000, total money: 10182.215205  
epoch: 100, total rewards: 861.215276.3, cost: 7310792458240.000000, total money: 7918.025274  
epoch: 110, total rewards: 68.690005.3, cost: 3184271573385216.000000, total money: 10068.690005  
epoch: 120, total rewards: 205.980352.3, cost: 224217291292672.000000, total money: 10205.980352  
epoch: 130, total rewards: 256.794983.3, cost: 363017178972160.000000, total money: 8275.784973  
epoch: 140, total rewards: 1586.720156.3, cost: 530019768074240.000000, total money: 11586.720156  
epoch: 150, total rewards: 824.849978.3, cost: 3151772092727296.000000, total money: 8881.750002  
epoch: 160, total rewards: 222.490291.3, cost: 6080023886823424.000000, total money: 9205.850276  
epoch: 170, total rewards: 37.630069.3, cost: 9586346603577344.000000, total money: 9020.990054  
epoch: 180, total rewards: 510.125126.3, cost: 22490134536519680.000000, total money: 5604.765140  
epoch: 190, total rewards: 639.559874.3, cost: 106721235701858304.000000, total money: 9669.019896  
epoch: 200, total rewards: 945.395079.3, cost: 31826508674760704.000000, total money: 384.445006

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

day 0: buy 1 unit at price 768.700012, total balance 9231.299988  
day 2: buy 1 unit at price 762.020020, total balance 8469.279968  
day 3, sell 1 unit at price 782.520020, investment 1.797842 %, total balance 9251.799988,  
day 4, sell 1 unit at price 790.510010, investment 3.738746 %, total balance 10042.309998,  
day 5: buy 1 unit at price 785.309998, total balance 9257.000000  
day 8: buy 1 unit at price 736.080017, total balance 8520.919983  
day 11, sell 1 unit at price 771.229980, investment -1.792925 %, total balance 9292.149963,  
day 12: buy 1 unit at price 760.539978, total balance 8531.609985  
day 14: buy 1 unit at price 768.270020, total balance 7763.339965  
day 15: buy 1 unit at price 760.989990, total balance 7002.349975  
day 17, sell 1 unit at price 768.239990, investment 4.369087 %, total balance 7770.589965,  
day 20, sell 1 unit at price 747.919983, investment -1.659347 %, total balance 8518.509948,  
day 21, sell 1 unit at price 750.500000, investment -2.312991 %, total balance 9269.009948,  
day 22, sell 1 unit at price 762.520020, investment 0.201058 %, total balance

10031.529968,  
 day 27: buy 1 unit at price 789.270020, total balance 9242.259948  
 day 33: buy 1 unit at price 796.419983, total balance 8445.839965  
 day 34, sell 1 unit at price 794.559998, investment 0.670237 %, total balance  
 9240.399963,  
 day 35, sell 1 unit at price 791.260010, investment -0.647896 %, total balance  
 10031.659973,  
 day 36: buy 1 unit at price 789.909973, total balance 9241.750000  
 day 37, sell 1 unit at price 791.549988, investment 0.207620 %, total balance  
 10033.299988,  
 day 41: buy 1 unit at price 786.140015, total balance 9247.159973  
 day 43: buy 1 unit at price 794.020020, total balance 8453.139953  
 day 44: buy 1 unit at price 806.150024, total balance 7646.989929  
 day 45, sell 1 unit at price 806.650024, investment 2.608951 %, total balance  
 8453.639953,  
 day 47: buy 1 unit at price 807.909973, total balance 7645.729980  
 day 48: buy 1 unit at price 806.359985, total balance 6839.369995  
 day 49, sell 1 unit at price 807.880005, investment 1.745546 %, total balance  
 7647.250000,  
 day 55: buy 1 unit at price 823.869995, total balance 6823.380005  
 day 56, sell 1 unit at price 835.669983, investment 3.661844 %, total balance  
 7659.049988,  
 day 57, sell 1 unit at price 832.150024, investment 3.000341 %, total balance  
 8491.200012,  
 day 58, sell 1 unit at price 823.309998, investment 2.102040 %, total balance  
 9314.510010,  
 day 59: buy 1 unit at price 802.320007, total balance 8512.190003  
 day 60, sell 1 unit at price 796.789978, investment -3.286928 %, total balance  
 9308.979981,  
 day 62: buy 1 unit at price 798.530029, total balance 8510.449952  
 day 63, sell 1 unit at price 801.489990, investment -0.103452 %, total balance  
 9311.939942,  
 day 69, sell 1 unit at price 819.239990, investment 2.593511 %, total balance  
 10131.179932,  
 day 75: buy 1 unit at price 830.760010, total balance 9300.419922  
 day 76, sell 1 unit at price 831.330017, investment 0.068613 %, total balance  
 10131.749939,  
 day 77: buy 1 unit at price 828.640015, total balance 9303.109924  
 day 78: buy 1 unit at price 829.280029, total balance 8473.829895  
 day 80, sell 1 unit at price 835.239990, investment 0.796483 %, total balance  
 9309.069885,  
 day 83: buy 1 unit at price 827.780029, total balance 8481.289856  
 day 84, sell 1 unit at price 831.909973, investment 0.317136 %, total balance  
 9313.199829,  
 day 86: buy 1 unit at price 838.679993, total balance 8474.519836  
 day 87: buy 1 unit at price 843.250000, total balance 7631.269836  
 day 88: buy 1 unit at price 845.539978, total balance 6785.729858  
 day 89, sell 1 unit at price 845.619995, investment 2.155158 %, total balance



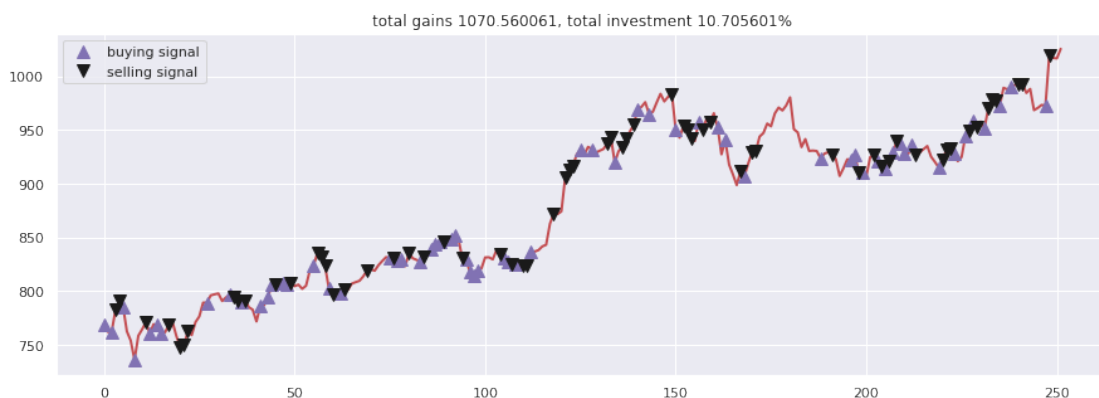
7631.349853,  
 day 91: buy 1 unit at price 848.780029, total balance 6782.569824  
 day 92: buy 1 unit at price 852.119995, total balance 5930.449829  
 day 94, sell 1 unit at price 830.460022, investment -0.980108 %, total balance  
 6760.909851,  
 day 95: buy 1 unit at price 829.590027, total balance 5931.319824  
 day 96: buy 1 unit at price 817.580017, total balance 5113.739807  
 day 97: buy 1 unit at price 814.429993, total balance 4299.309814  
 day 98: buy 1 unit at price 819.510010, total balance 3479.799804  
 day 104, sell 1 unit at price 834.570007, investment -1.029350 %, total balance  
 4314.369811,  
 day 105: buy 1 unit at price 831.409973, total balance 3482.959838  
 day 106: buy 1 unit at price 827.880005, total balance 2655.079833  
 day 107, sell 1 unit at price 824.669983, investment -2.468245 %, total balance  
 3479.749816,  
 day 108: buy 1 unit at price 824.729980, total balance 2655.019836  
 day 110, sell 1 unit at price 824.320007, investment -2.881786 %, total balance  
 3479.339843,  
 day 111, sell 1 unit at price 823.559998, investment -3.351640 %, total balance  
 4302.899841,  
 day 112: buy 1 unit at price 837.169983, total balance 3465.729858  
 day 118, sell 1 unit at price 872.299988, investment 5.148321 %, total balance  
 4338.029846,  
 day 121, sell 1 unit at price 905.960022, investment 10.809952 %, total balance  
 5243.989868,  
 day 122, sell 1 unit at price 912.570007, investment 12.050147 %, total balance  
 6156.559875,  
 day 123, sell 1 unit at price 916.440002, investment 11.827798 %, total balance  
 7072.999877,  
 day 125: buy 1 unit at price 931.659973, total balance 6141.339904  
 day 128: buy 1 unit at price 932.169983, total balance 5209.169921  
 day 132, sell 1 unit at price 937.080017, investment 12.709740 %, total balance  
 6146.249938,  
 day 133, sell 1 unit at price 943.000000, investment 13.905396 %, total balance  
 7089.249938,  
 day 134: buy 1 unit at price 919.619995, total balance 6169.629943  
 day 136, sell 1 unit at price 934.010010, investment 13.250401 %, total balance  
 7103.639953,  
 day 137, sell 1 unit at price 941.859985, investment 12.505226 %, total balance  
 8045.499938,  
 day 139, sell 1 unit at price 954.960022, investment 2.500918 %, total balance  
 9000.459960,  
 day 140: buy 1 unit at price 969.539978, total balance 8030.919982  
 day 143: buy 1 unit at price 964.859985, total balance 7066.059997  
 day 149, sell 1 unit at price 983.409973, investment 5.496850 %, total balance  
 8049.469970,  
 day 150: buy 1 unit at price 949.830017, total balance 7099.639953  
 day 152, sell 1 unit at price 953.400024, investment 3.673260 %, total balance

8053.039977,  
 day 153, sell 1 unit at price 950.760010, investment -1.936998 %, total balance 9003.799987,  
 day 154, sell 1 unit at price 942.309998, investment -2.337125 %, total balance 9946.109985,  
 day 156: buy 1 unit at price 957.369995, total balance 8988.739990  
 day 157, sell 1 unit at price 950.630005, investment 0.084224 %, total balance 9939.369995,  
 day 159, sell 1 unit at price 957.090027, investment -0.029243 %, total balance 10896.460022,  
 day 161: buy 1 unit at price 952.270020, total balance 9944.190002  
 day 163: buy 1 unit at price 940.489990, total balance 9003.700012  
 day 167, sell 1 unit at price 911.710022, investment -4.259296 %, total balance 9915.410034,  
 day 168: buy 1 unit at price 906.690002, total balance 9008.720032  
 day 170, sell 1 unit at price 928.799988, investment -1.242969 %, total balance 9937.520020,  
 day 171, sell 1 unit at price 930.090027, investment 2.580819 %, total balance 10867.610047,  
 day 188: buy 1 unit at price 923.650024, total balance 9943.960023  
 day 191, sell 1 unit at price 926.789978, investment 0.339951 %, total balance 10870.750001,  
 day 196: buy 1 unit at price 922.219971, total balance 9948.530030  
 day 197: buy 1 unit at price 926.960022, total balance 9021.570008  
 day 198, sell 1 unit at price 910.979980, investment -1.218797 %, total balance 9932.549988,  
 day 199: buy 1 unit at price 910.669983, total balance 9021.880005  
 day 202, sell 1 unit at price 927.000000, investment 0.004313 %, total balance 9948.880005,  
 day 203: buy 1 unit at price 921.280029, total balance 9027.599976  
 day 204, sell 1 unit at price 915.890015, investment 0.573208 %, total balance 9943.489991,  
 day 205: buy 1 unit at price 913.809998, total balance 9029.679993  
 day 206, sell 1 unit at price 921.289978, investment 0.001080 %, total balance 9950.969971,  
 day 207: buy 1 unit at price 929.570007, total balance 9021.399964  
 day 208, sell 1 unit at price 939.330017, investment 2.792705 %, total balance 9960.729981,  
 day 209: buy 1 unit at price 937.340027, total balance 9023.389954  
 day 210: buy 1 unit at price 928.450012, total balance 8094.939942  
 day 212: buy 1 unit at price 935.950012, total balance 7158.989930  
 day 213, sell 1 unit at price 926.500000, investment -0.330261 %, total balance 8085.489930,  
 day 219: buy 1 unit at price 915.000000, total balance 7170.489930  
 day 220, sell 1 unit at price 921.809998, investment -1.656819 %, total balance 8092.299928,  
 day 221, sell 1 unit at price 931.580017, investment 0.337122 %, total balance 9023.879945,

day 222, sell 1 unit at price 932.450012, investment -0.373952 %, total balance 9956.329957,  
 day 223: buy 1 unit at price 928.530029, total balance 9027.799928  
 day 226: buy 1 unit at price 944.489990, total balance 8083.309938  
 day 227, sell 1 unit at price 949.500000, investment 3.770492 %, total balance 9032.809938,  
 day 228: buy 1 unit at price 959.109985, total balance 8073.699953  
 day 229, sell 1 unit at price 953.270020, investment 2.664426 %, total balance 9026.969973,  
 day 231: buy 1 unit at price 951.679993, total balance 8075.289980  
 day 232, sell 1 unit at price 969.960022, investment 2.696697 %, total balance 9045.250002,  
 day 233, sell 1 unit at price 978.890015, investment 2.062332 %, total balance 10024.140017,  
 day 234, sell 1 unit at price 977.000000, investment 2.660559 %, total balance 11001.140017,  
 day 235: buy 1 unit at price 972.599976, total balance 10028.540041  
 day 238: buy 1 unit at price 989.679993, total balance 9038.860048  
 day 240, sell 1 unit at price 992.179993, investment 2.013162 %, total balance 10031.040041,  
 day 241, sell 1 unit at price 992.809998, investment 0.316264 %, total balance 11023.850039,  
 day 247: buy 1 unit at price 972.559998, total balance 10051.290041  
 day 248, sell 1 unit at price 1019.270020, investment 4.802791 %, total balance 11070.560061,

```

[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()
  
```



[ ]: