```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
```

ZAD 1

```
def y_(sum):
    if sum > 0:
        return 1
    else:
        return -1

def suma(xi, xi2, w1, w2, x0, Theta):
    sum = xi * w1 + xi2 * w2 + Theta * x0
    return sum

x1 = [2, 2, 0, -2, -2, 0, 4]
x2 = [1, 2, 6, 10, 0, 0, -20]
#x2 = [1, 2, -16, 10, 0, 0, -20]
d = [1, 1, 1, -1, -1, -1, -1]
w = [0, 0, 0]
Theta = 1 # x0 bias
iterator = 0

plt.scatter(x1,x2)
while True:
    z = True
    for i in range(7):
        y = y_(suma(x1[i], x2[i], w[0], w[1], Theta, w[2]))
        if y != d[i]:
            w[0] = w[0] + d[i] * x1[i]
            w[1] = w[1] + d[i] * x2[i]
            w[2] = w[2] + d[i] * Theta
            z = False
        print(y , d[i])
    if w[1] != 0:
      a= -w[0]/w[1]
      b = -w[2]/w[1]
    print(f"a: {a}, b: {b}")
    fx = [a*x+b for x in x1]
    _ = plt.plot(x1[:3],x2[:3],'o', color = "green")
    _ = plt.plot(x1[3:],x2[3:],'o')
    _ = plt.plot(x1,fx,'r')
    iterator += 1
    print()
    print(iterator)
    print()
    if z:
        break

print("w 0 =", w[2] , "w 1 =", w[0] , "w 2 =", w[1])
```

```
-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: 0.0, b: 0.09090909090909091

1

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 2.0

2

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: 0.0, b: 0.36363636363636365

3

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

4

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.16666666666666666, b: 0.5

5

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 3.5

6

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.16666666666666666, b: 0.75

7

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

8

-1 1
1 1
```

```
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.3076923076923077, b: 0.8461538461538461

9

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.0

10

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.3076923076923077, b: 1.0769230769230769

11

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

12

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.42857142857142855, b: 1.1428571428571428

13

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.25

14

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.42857142857142855, b: 1.3571428571428572

15

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

16

-1 1
1 1
1 1
1 -1
-1 -1
```

```
 1  1
-1 -1
1 -1
a: -0.5333333333333333, b: 1.4

17

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.4

18

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.5333333333333333, b: 1.6

19

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

20

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.625, b: 1.625

21

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.5

22

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.625, b: 1.8125

23

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

24

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
```

```
a: -0.7058823529411765, b: 1.8235294117647058

25

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.571428571428571

26

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.7058823529411765, b: 2.0

27

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

28

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.7777777777777778, b: 2.0

29

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.625

30

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.7777777777777778, b: 2.1666666666666665

31

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

32

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.8421052631578947, b: 2.1578947368421053

33
```

```
1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.666666666666667

34

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.8421052631578947, b: 2.3157894736842106

35

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

36

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.9, b: 2.3

37

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.7

38

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.9, b: 2.45

39

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

40

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.9523809523809523, b: 2.4285714285714284

41

1 1
```

```
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.7272727272727275

42

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -0.9523809523809523, b: 2.5714285714285716

43

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

44

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -1.0, b: 2.5454545454545454

45

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 4.75

46

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
1 -1
a: -1.0, b: 2.6818181818181817

47

1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -2.0, b: 5.0

48

-1 1
1 1
1 1
1 -1
-1 -1
-1 -1
-1 -1
a: -9.333333333333334, b: 20.0

49

-1 1
1 1
-1 1
-1 -1
```

```
-1 -1
-1 -1
-1 -1
a: -3.0, b: 5.8

50

1 1
1 1
1 1
-1 -1
-1 -1
-1 -1
-1 -1
a: -3.0, b: 5.8
```

```
#a= -w[0]/w[1]
#b = -w[2]/w[1]
#print(f"a: {a}, b: {b}")
```

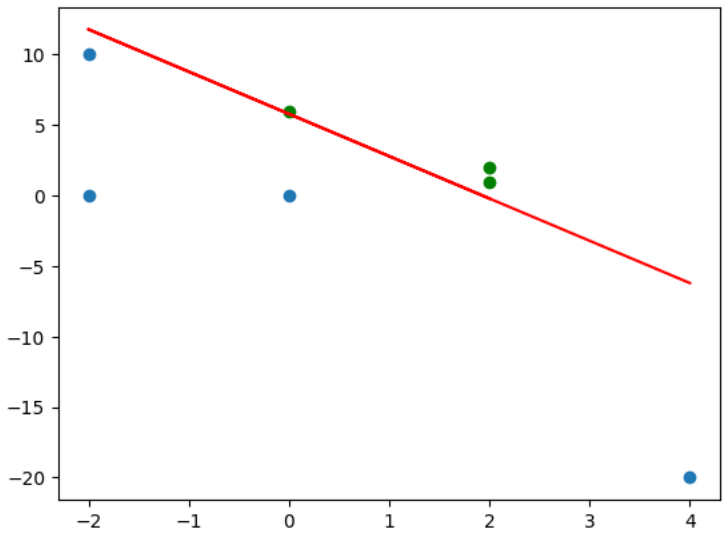
```
a: -3.0, b: 5.8
```

```
#y = [a*x+b for x in x1]
```

```
#y
```

```
[-0.20000000000000018, -0.20000000000000018, 5.8, 11.8, 11.8, 5.8, -6.2]
```

```
_ = plt.plot(x1[:3],x2[:3],'o', color = "green")
_ = plt.plot(x1[3:],x2[3:],'o')
_ = plt.plot(x1,fx,'r')
```



ZAD 2

```
import math

SIZE_TAB = 13
beta = 0.5
ni = 0.35

def f(suma):
    y = (1 - math.exp(-1*suma*beta))/(1+math.exp(-1*suma*beta))
    return y

one = [-1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1]
four = [1, -1, 1, 1, 1, 1, -1, -1, 1, -1, -1, 1, 1]
d = [-1, 1]
w = [0]*SIZE_TAB
w2 = [0]*SIZE_TAB
iterator = 0
e_tab = []
while True:
    E = 0
    for i in range(12):
        s = one[i] * w[i] + w[12] * one[12]
        y = f(s)
        w[i] += d[0] * one[i] * ni
```

```
            w[12] += d[0] * one[12] * ni
            print(y,d[0])
            s2 = four[i] * w2[i] + w2[12] * four[12]
            y2 = f(s2)
            w2[i] += d[1] * four[i] * ni
            w2[12] += d[1] * four[12] * ni
            print(y2,d[1])
        E += 0.5*(d[0] - y)*(d[0] - y)+0.5*(d[1] - y2)*(d[1] - y2)
        print(y,"<-- y koncowy dla 1",d[0],"<--d oczekiwana")
        print(y2,"<-- y koncowy dla 4",d[1],"<--d oczekiwana")
        iterator += 1
        print("\n",iterator,"<-- przebieg funkcji\n")
        print(E,"blad sumaryczny")
        e_tab.append(E)
        if E <= 0.0000000005:
            break

    plt.plot(e_tab)
```

```
0.0 -1
0.0 1
-0.08727737447415772 -1
0.08727737447415773 1
-0.17323515783466006 -1
0.17323515783466012 1
-0.25663237659073246 -1
0.25663237659073246 1
-0.3363755443363322 -1
0.3363755443363322 1
-0.4115700556740225 -1
0.4115700556740224 1
-0.481549798364308 -1
0.48154979836430806 1
-0.545884518793477 -1
0.5458845187934771 1
-0.6043677771171636 -1
0.6043677771171636 1
-0.6569904600491983 -1
0.6569904600491984 1
-0.7039056039366212 -1
0.7039056039366212 1
-0.7453899238492567 -1
0.7453899238492567 1
-0.7453899238492567 <-- y koncowy dla 1 -1 <--d oczekiwana
0.7453899238492567 <-- y koncowy dla 4 1 <--d oczekiwana

 1 <-- przebieg funkcji

0.06482629087748729 blad sumaryczny
-0.8135705538363528 -1
0.8135705538363527 1
-0.8411229016320432 -1
0.8411229016320432 1
-0.8649066177207417 -1
0.8649066177207417 1
-0.8853516482022624 -1
0.8853516482022624 1
-0.9028636912994877 -1
0.9028636912994876 1
-0.9178174435990699 -1
0.9178174435990698 1
-0.9305531409332848 -1
0.9305531409332849 1
-0.9413755384972873 -1
0.9413755384972872 1
-0.9505546004392487 -1
0.9505546004392487 1
-0.9583273109626391 -1
0.958327310962639 1
-0.9649001551103512 -1
0.9649001551103512 1
-0.9704519366134539 -1
0.9704519366134539 1
-0.9704519366134539 <-- y koncowy dla 1 -1 <--d oczekiwana
0.9704519366134539 <-- y koncowy dla 4 1 <--d oczekiwana

 2 <-- przebieg funkcji

0.0008730880498953486 blad sumaryczny
-0.9790865875361638 -1
0.9790865875361637 1
-0.9824145675206502 -1
0.9824145675206502 1
-0.985216917311436 -1
0.9852169173114361 1
-0.9875754937830745 -1
0.9875754937830745 1
-0.9895597486128832 -1
0.9895597486128832 1
-0.9912285066847746 -1
0.9912285066847745 1
-0.992631520201128 -1
0.9926315202011281 1
-0.9938108171296763 -1
0.9938108171296764 1
-0.9948018644753532 -1
0.9948018644753533 1
-0.9956345671093096 -1
0.9956345671093096 1
-0.9963341221150144 -1
0.9963341221150144 1
-0.9969217472486472 -1
0.9969217472486471 1
-0.9969217472486472 <-- y koncowy dla 1 -1 <--d oczekiwana
0.9969217472486471 <-- y koncowy dla 4 1 <--d oczekiwana

 3 <-- przebieg funkcji

9.475640001211187e-06 blad sumaryczny
```

```
-0.9978298055470948 -1
0.9978298055470949 1
-0.9981778976111987 -1
0.9981778976111988 1
-0.9984701996041594 -1
0.9984701996041594 1
-0.998715640598828 -1
0.9987156405988281 1
-0.9989217243160301 -1
0.9989217243160301 1
-0.9990947555535189 -1
0.999094755553519 1
-0.9992400309704963 -1
0.9992400309704962 1
-0.9993619997421465 -1
0.9993619997421463 1
-0.999464398774423 -1
0.999464398774423 1
-0.9995503664595334 -1
0.9995503664595333 1
-0.9996225383440512 -1
0.9996225383440513 1
-0.9996831275617949 -1
0.9996831275617949 1
-0.9996831275617949 <-- y koncowy dla 1 -1 <--d oczekiwana
0.9996831275617949 <-- y koncowy dla 4 1 <--d oczekiwana

 4 <-- przebieg funkcji

1.0040814209404929e-07 blad sumaryczny
-0.9997766933187409 -1
0.999776693318741 1
-0.9998125402783868 -1
0.999812540278387 1
-0.9998426332525774 -1
0.9998426332525774 1
-0.9998678957102907 -1
0.9998678957102908 1
-0.9998891029505544 -1
0.9998891029505543 1
-0.9999069058645483 -1
0.9999069058645483 1
-0.9999218508904095 -1
0.9999218508904095 1
-0.9999343967697488 -1
0.9999343967697489 1
```

Zadanie 3

```
0.9999537697371759 1
```

```
import math

WIELKOSC_TAB = 2
WIELKOSC_TAB2 = 13
beta = 1
ni = 1

def f(suma):
    return 1 / (1 + math.exp(-beta * suma))

def fprim(x):
    y = beta * f(x)* (1-f(x))
    return y

w = [0]*13
one = [1,-1,-1,1,-1,-1,1,-1,1,1,-1,-1,1]
four = [1,1,1,1,1,1,1,-1,-1,1,-1,-1,1]
like_one = [1,-1,-1,1,-1,-1,1,-1,1,1,-1,-1,1]
like_four = [1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, -1, 1]
d = [-1, 1]
iterator = 0
d4 = 0
d2 = 0
d1 = 1
s2=0
e_tab = []
while True:
    E=0
    s=0
    s2=0
    for i in range(13):
      s = s + one[i] * w[i]
      s2 = s2 + four[i] * w[i]
    y2 = f(s2)
    y = f(s)
    for i in range(13):
        w[i] = w[i] + ni * beta*(1-y)*y*(d1-y)*one[i]
```

```
            w[i] = w[i] + ni * beta*(1-y2)*y2*(d4-y2)*four[i]

        iterator += 1
        E +=  0.5*(d1 - y)*(d1 - y) +0.5*(d4 - y2)*(d4 - y2)
        print(E,"blad sumaryczny")
        e_tab.append(E)
        if E <= 0.0005:
            break


s1,s2,s3,s4= 0,0,0,0
for i in range(13):
  s1+=one[i] * w[i]
  s2+=four[i] * w[i]
  s3+=like_one[i] * w[i]
  s4+=like_four[i] * w[i]


print("wyjscie dla 1",f(s1),"\nwyjscie dla  4",f(s2),"\n wyjscie dla niby 1",f(s3),"\nwyjscie dla niby 4",f(s4))

print(e_tab)
plt.plot(e_tab)
```

```
0.25 blad sumaryczny
0.04959535183281184 blad sumaryczny
0.026594329976262845 blad sumaryczny
0.01820534323195647 blad sumaryczny
0.013824598180667213 blad sumaryczny
0.011130144480494819 blad sumaryczny
0.009305767833717853 blad sumaryczny
0.00798921444925493 blad sumaryczny
0.0069948425302024456 blad sumaryczny
0.006217630869051352 blad sumaryczny
0.005593692244207606 blad sumaryczny
0.005081932551496992 blad sumaryczny
0.004654721616590174 blad sumaryczny
0.004292799465244527 blad sumaryczny
0.0039823321223810245 blad sumaryczny
0.0037131281529588244 blad sumaryczny
0.0034775149965932768 blad sumaryczny
0.003269606897218701 blad sumaryczny
0.003084813984889352 blad sumaryczny
0.0029195046800876206 blad sumaryczny
0.0027707683367679684 blad sumaryczny
0.002636245051410805 blad sumaryczny
0.0025140014723105423 blad sumaryczny
0.0024024387358102927 blad sumaryczny
0.002300223239131779 blad sumaryczny
0.0022062339070208657 blad sumaryczny
0.0021195215452525173 blad sumaryczny
0.0020392771697631926 blad sumaryczny
0.001964807082611598 blad sumaryczny
0.00189551307654693 blad sumaryczny
0.0018308765786497296 blad sumaryczny
0.001770445848569357 blad sumaryczny
0.0017138255666907186 blad sumaryczny
0.00166066830778837 blad sumaryczny
0.0016106675137854622 blad sumaryczny
0.0015635516671044635 blad sumaryczny
0.0015190794321155864 blad sumaryczny
0.0014770355822293793 blad sumaryczny
0.0014372275684265269 blad sumaryczny
0.0013994826144788494 blad sumaryczny
0.001363645246976409 blad sumaryczny
0.0013295751861394818 blad sumaryczny
0.0012971455374454065 blad sumaryczny
0.0012662412352213892 blad sumaryczny
0.0012367576982090529 blad sumaryczny
0.0012085996641964239 blad sumaryczny
0.0011816801765200304 blad sumaryczny
0.0011559196998577243 blad sumaryczny
0.001131245346487013 blad sumaryczny
0.0011075901972505825 blad sumaryczny
0.0010848927039868394 blad sumaryczny
0.0010630961622565599 blad sumaryczny
0.0010421482449120205 blad sumaryczny
0.0010222000588479534 blad sumaryczny
0.0010026084255142264 blad sumaryczny
0.000983930257079372 blad sumaryczny
0.0009659275603371622 blad sumaryczny
0.000948564526940398 blad sumaryczny
0.0009318078285086653 blad sumaryczny
0.0009156264059760274 blad sumaryczny
0.0008999912800251491 blad sumaryczny
0.0008848753801880893 blad sumaryczny
0.0008702533905056312 blad sumaryczny
0.0008561016099050036 blad sumaryczny
0.0008423978256853795 blad sumaryczny
0.0008291211986990523 blad sumaryczny
0.00081625215898708 2 blad sumaryczny
0.0008037723107764177 blad sumaryczny
0.0007916643458742272 blad sumaryczny
0.0007799119646067676 blad sumaryczny
0.0007684998035477463 blad sumaryczny
0.0007574133693661176 blad sumaryczny
0.0007466389781977652 blad sumaryczny
0.0007361637000107993 blad sumaryczny
0.0007259753074915899 blad sumaryczny
0.0007160622290290227 blad sumaryczny
0.0007064135054190341 blad sumaryczny
0.0006970187499508316 blad sumaryczny
0.0006878681115707393 blad sumaryczny
0.0006789522408507389 blad sumaryczny
0.0006702622585157622 blad sumaryczny
0.0006617897263083519 blad sumaryczny
0.0006535266199908076 blad sumaryczny
0.0006454653043041946 blad sumaryczny
0.00063759850972096 blad sumaryczny
0.0006299193108430343 blad sumaryczny
0.0006224211063114026 blad sumaryczny
0.00061509760010515 8 blad sumaryczny
0.00060794278411935 blad sumaryczny
0.0006009509219208938 blad sumaryczny
```

```
0.0005941165335906414 blad sumaryczny
0.0005874343815679599 blad sumaryczny
0.0005808994574213056 blad sumaryczny
0.0005745069694750335 blad sumaryczny
0.0005682523312284596 blad sumaryczny
0.0005621311505087047 blad sumaryczny
0.0005561392193036952 blad sumaryczny
0.0005502725042261188 blad sumaryczny
```