Laboratorium 4 - Perceptron, problem xor

```
import numpy as np

x = np.array(
    [[0,0],
    [0,1],
    [1,0],
    [1,1]])

d = np.array([1,0,0,1])
```

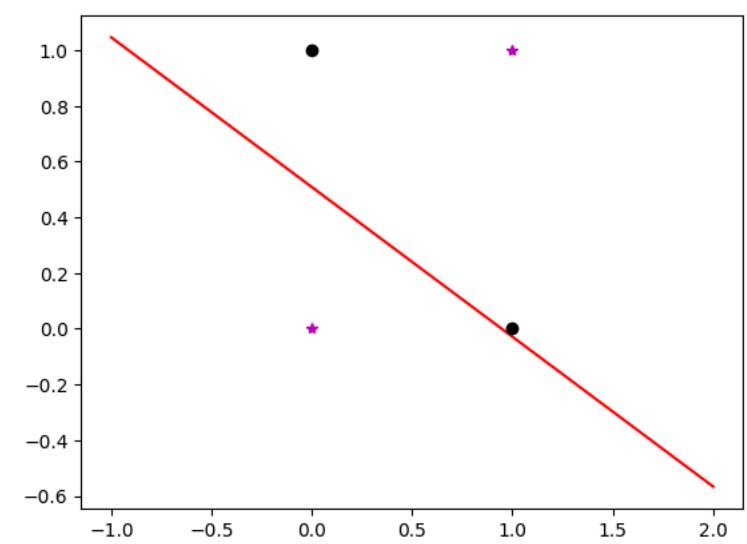
Zadanie 1

Sprawdzenie rozpoznawania punktów dla 1 perceptronu

```
w = np.random.random(3)


for i in range( len(x) ):
  xx = x[i]
  dd = d[i]

  s = xx[0]*w[1] + xx[1]*w[2] + w[0]*(-1)
  if s >= 0:
    y = 1
  else:
    y = 0

  print("x1: ", xx[0], " x2: ", xx[1], " d: ", dd, " y: ", y)
```

```
x1:  0  x2:  0  d:  1  y:  0
x1:  0  x2:  1  d:  0  y:  1
x1:  1  x2:  0  d:  0  y:  1
x1:  1  x2:  1  d:  1  y:  1
```

```
mi = 0.1
for a in range(100):
  for i in range( len(x) ):
    xx = x[i]
    dd = d[i]

    s = xx[0]*w[1] + xx[1]*w[2] + w[0]*(-1)
    if s >= 0:
      y = 1
    else:
      y = 0

    w[0] = w[0] + mi*(dd-y)*(-1)
    w[1] = w[1] + mi*(dd-y)*xx[0]
    w[2] = w[2] + mi*(dd-y)*xx[1]


import matplotlib.pyplot as plt
xx = np.arange(-1,3)
yy = -(w[1]/w[2]) * xx + (w[0]/w[2])
plt.plot(xx,yy, 'r-')

for i in range( len(x) ):
  if d[i] == 0:
    plt.plot(x[i, 0], x[i, 1], 'ko')
  else:
    plt.plot(x[i, 0], x[i, 1], 'm*')
```



Zadanie 2

```
import numpy as np

x = np.array(
    [[0,0],
    [0,1],
    [1,0],
    [1,1]])

d1 = np.array([0,0,0,1])
d2 = np.array([1,0,0,0])

w1 = np.random.random(3)
w2 = np.random.random(3)


for i in range(len(x)):
  xx = x[i]
  dd1 = d1[i]

  s1 = xx[0]*w1[1] + xx[1]*w1[2] + w1[0]*(-1)
  if s1 >= 0:
    y1 = 1
  else:
    y1 = 0

  print("x1: ", xx[0], " x2: ", xx[1], " d1: ", dd1, " y1: ", y1)

print("=======")

for i in range(len(x)):
  xx = x[i]
  dd2 = d2[i]

  s2 = xx[0]*w2[1] + xx[1]*w2[2] + w2[0]*(-1)
  if s2 >= 0:
    y2 = 1
  else:
    y2 = 0

  print("x1: ", xx[0], " x2: ", xx[1], " d2: ", dd2, " y2: ", y2)
```

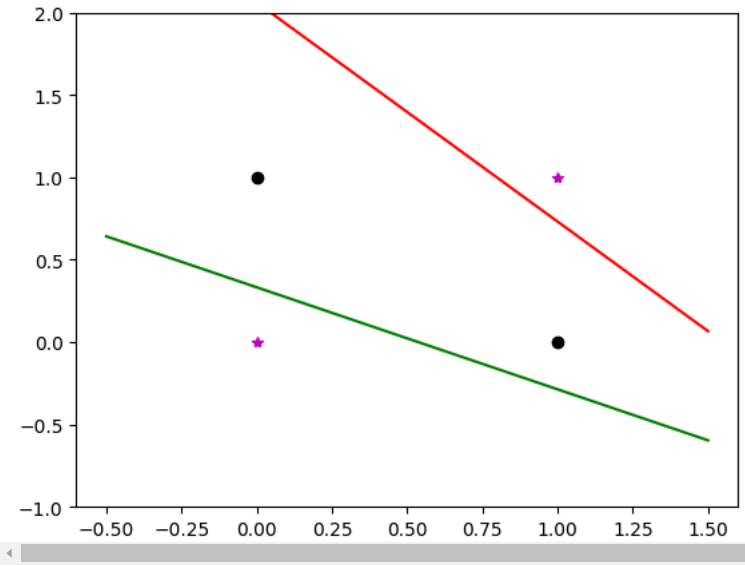
```
x1:  0  x2:  0  d1:  0  y1:  0
x1:  0  x2:  1  d1:  0  y1:  0
x1:  1  x2:  0  d1:  0  y1:  0
x1:  1  x2:  1  d1:  1  y1:  1
=======
x1:  0  x2:  0  d2:  1  y2:  1
x1:  0  x2:  1  d2:  0  y2:  0
x1:  1  x2:  0  d2:  0  y2:  0
x1:  1  x2:  1  d2:  0  y2:  0
```

```
mi = 0.1
y1 = np.zeros(4)
y2 = np.zeros(4)
for a in range(100):
```

```
  for i in range( len(x) ):
    xx = x[i]
    dd1 = d1[i]

    s1 = xx[0]*w1[1] + xx[1]*w1[2] + w1[0]*(-1)
    if s1 >= 0:
      y1[i] = 1
    else:
      y1[i] = 0

    w1[0] = w1[0] + mi*(dd1-y1[i])*(-1)
    w1[1] = w1[1] + mi*(dd1-y1[i])*xx[0]
    w1[2] = w1[2] + mi*(dd1-y1[i])*xx[1]

  for i in range( len(x) ):
    xx = x[i]
    dd2 = d2[i]

    s2 = xx[0]*w2[1] + xx[1]*w2[2] + w2[0]*(-1)
    if s2 >= 0:
      y2[i] = 1
    else:
      y2[i] = 0

    w2[0] = w2[0] + mi*(dd2-y2[i])*(-1)
    w2[1] = w2[1] + mi*(dd2-y2[i])*xx[0]
    w2[2] = w2[2] + mi*(dd2-y2[i])*xx[1]
#print(y1)
#print(y2)

y1 = y1.reshape(4, 1)
y2 = y2.reshape(4, 1)

y = np.concatenate((y1, y2), axis=1)
print(y)
```

```
[[0. 1.]
 [0. 0.]
 [0. 0.]
 [1. 0.]]
```

```
import matplotlib.pyplot as plt
xx = np.arange(-0.5,2.5)
plt.ylim(-1, 2)

yy1 = -(w1[1]/w1[2]) * xx + (w1[0]/w1[2])
plt.plot(xx,yy1, 'r-')

yy2 = -(w2[1]/w2[2]) * xx + (w2[0]/w2[2])
plt.plot(xx,yy2, 'g-')


for i in range( len(x) ):
  if d1[i] == 1 or d2[i]==1:
    plt.plot(x[i, 0], x[i, 1], 'm*')
  else:
    plt.plot(x[i, 0], x[i, 1], 'ko')
```



Zadanie 3 - trzeci perceptron

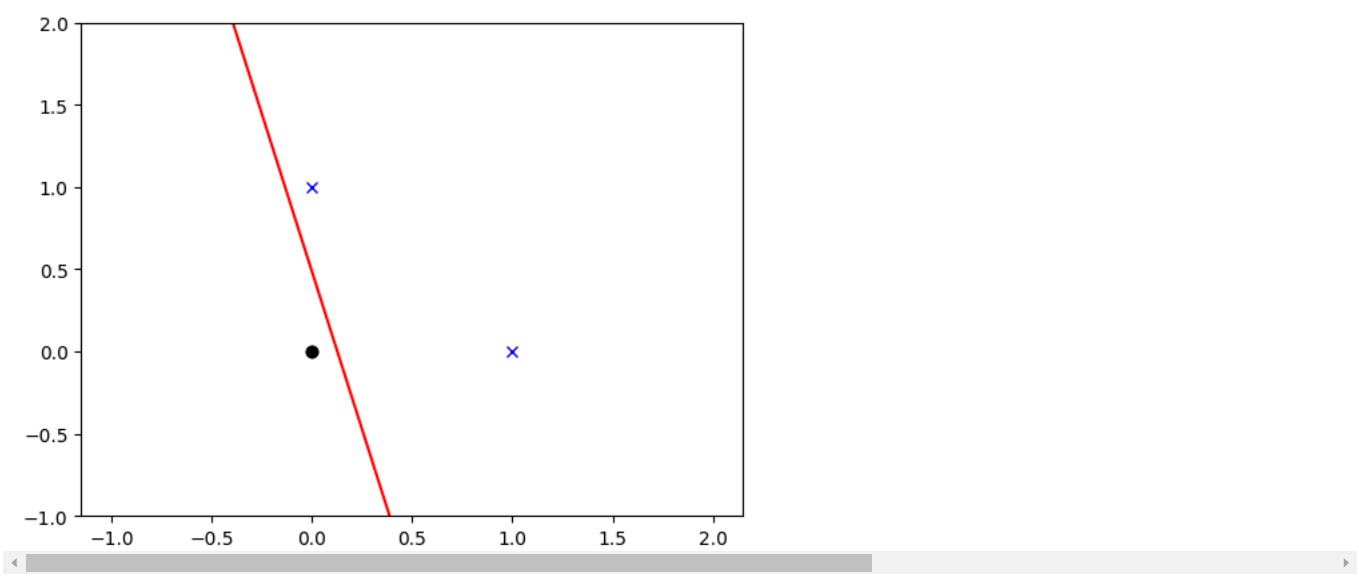
```
w3 = np.random.random(3)
```

```
for i in range( len(x) ):
  yy = y[i]
  dd3 = d[i]

  s3 = yy[0] * w3[1] + yy[1] * w3[2] + w3[0] * (-1)
  if s3 >= 0:
    y3 = 1
  else:
    y3 = 0
  print("y1: ", yy[0], " y2: ", yy[1], " d3: ", dd3, " y: ", y3)
```

```
y1:  0.0  y2:  1.0  d3:  1  y:  0
y1:  0.0  y2:  0.0  d3:  0  y:  0
y1:  0.0  y2:  0.0  d3:  0  y:  0
y1:  1.0  y2:  0.0  d3:  1  y:  1
```

```
mi = 0.1
y3 = np.zeros(3)

for a in range(10):
  for i in range(len(x)):
      yy = y[i]
      dd3 = d[i]

      s3 = yy[0] * w3[1] + yy[1] * w3[2] + w3[0] * (-1)
      if s3 >= 0:
        y3 = 1
      else:
        y3 = 0

      w3[0] = w3[0] + mi * (dd3 - y3) * (-1)
      w3[1] = w3[1] + mi * (dd3 - y3) * yy[0]
      w3[2] = w3[2] + mi * (dd3 - y3) * yy[1]
```

```
import matplotlib.pyplot as plt
xx = np.arange(-1,3)
yyy = -(w3[1]/w3[2]) * xx + (w3[0]/w3[2])
plt.plot(xx,yyy, 'r-')
plt.ylim(-1, 2)

for i in range( len(x) ):
  if d[i] == 1:
    plt.plot(y[i, 0], y[i, 1], 'bx')
  else:
    plt.plot(y[i, 0], y[i, 1], 'ko')
```




```
def predict (x, w):
```

```
  s = x[0]*w[1] + x[1]*w[2] + w[0]*(-1)
  if s >= 0:
    return 1
  else:
    return 0
```

Zadanie 4

```
print("x1   x2    y1    y2    d")

for i in range( len(x) ):
  xx = x[i]

  y1_p = predict(xx, w1)
  y2_p = predict(xx, w2)


  y_p = predict([y1_p, y2_p], w3)
  print("- - - - - - - - - - - -")
  print(xx[0], "  ", xx[1],"   ",  y1_p,"   ", y2_p, "   ", y_p)
```

```
x1   x2    y1    y2    d
- - - - - - - - - - - -
0    0     0     1     1
- - - - - - - - - - - -
0    1     0     0     0
- - - - - - - - - - - -
1    0     0     0     0
- - - - - - - - - - - -
1    1     1     0     1
```