Лабораторная работа № 1 по Нейроинформатике

Персептроны. Процедура обучения Розенбланта

Выполнила: Тимофеева Наталья М8О-408Б-19

Вариант № 16

Часть 1

Бинарная классификация

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras import layers
from tensorflow.python.keras import activations
from matplotlib import pyplot as plt
```

Данные для обучения

```
data_xy = np.array([[-5, 1], [-1.2, 4.1], [-5, -0.4], [-0.8, -0.4],
[2.7, -1.8], [2.8, -0.3]])
labels = np.array([0, 0, 0, 1, 1, 1])

model = keras.models.Sequential()
model.add(keras.layers.Dense(1, input_dim = 2, activation='sigmoid',
                             kernel_initializer =
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0),
                             bias_initializer =
keras.initializers.Zeros()))
model.compile(loss = 'mse', optimizer = 'adam', metrics = ['mae'])

model.fit(data_xy, labels, batch_size = 1, epochs = 500)
```

```
Epoch 1/500
6/6 [==============================] - 0s 990us/step - loss: 0.1957 -
mae: 0.4164
Epoch 2/500
6/6 [==============================] - 0s 798us/step - loss: 0.1928 -
mae: 0.4126
Epoch 3/500
6/6 [==============================] - 0s 796us/step - loss: 0.1898 -
mae: 0.4087
Epoch 4/500
6/6 [==============================] - 0s 997us/step - loss: 0.1877 -
mae: 0.4058
Epoch 5/500
6/6 [==============================] - 0s 1ms/step - loss: 0.1850 -
mae: 0.4017
Epoch 6/500
6/6 [==============================] - 0s 1ms/step - loss: 0.1822 -
```

```
Epoch 490/500
6/6 [==============================] - 0s 1ms/step - loss: 0.0277 -
mae: 0.1170
Epoch 491/500
6/6 [==============================] - 0s 998us/step - loss: 0.0276 -
mae: 0.1168
Epoch 492/500
6/6 [==============================] - 0s 2ms/step - loss: 0.0275 -
mae: 0.1167
Epoch 493/500
6/6 [==============================] - 0s 2ms/step - loss: 0.0275 -
mae: 0.1165
Epoch 494/500
6/6 [==============================] - 0s 1ms/step - loss: 0.0274 -
mae: 0.1164
Epoch 495/500
6/6 [==============================] - 0s 1ms/step - loss: 0.0273 -
mae: 0.1162
Epoch 496/500
6/6 [==============================] - 0s 2ms/step - loss: 0.0273 -
mae: 0.1160
Epoch 497/500
6/6 [==============================] - 0s 1ms/step - loss: 0.0272 -
mae: 0.1159
Epoch 498/500
6/6 [==============================] - 0s 996us/step - loss: 0.0272 -
mae: 0.1158
Epoch 499/500
6/6 [==============================] - 0s 2ms/step - loss: 0.0271 -
mae: 0.1156
Epoch 500/500
6/6 [==============================] - 0s 2ms/step - loss: 0.0271 -
mae: 0.1154

<keras.callbacks.History at 0x2a3ecfd60b0>
```

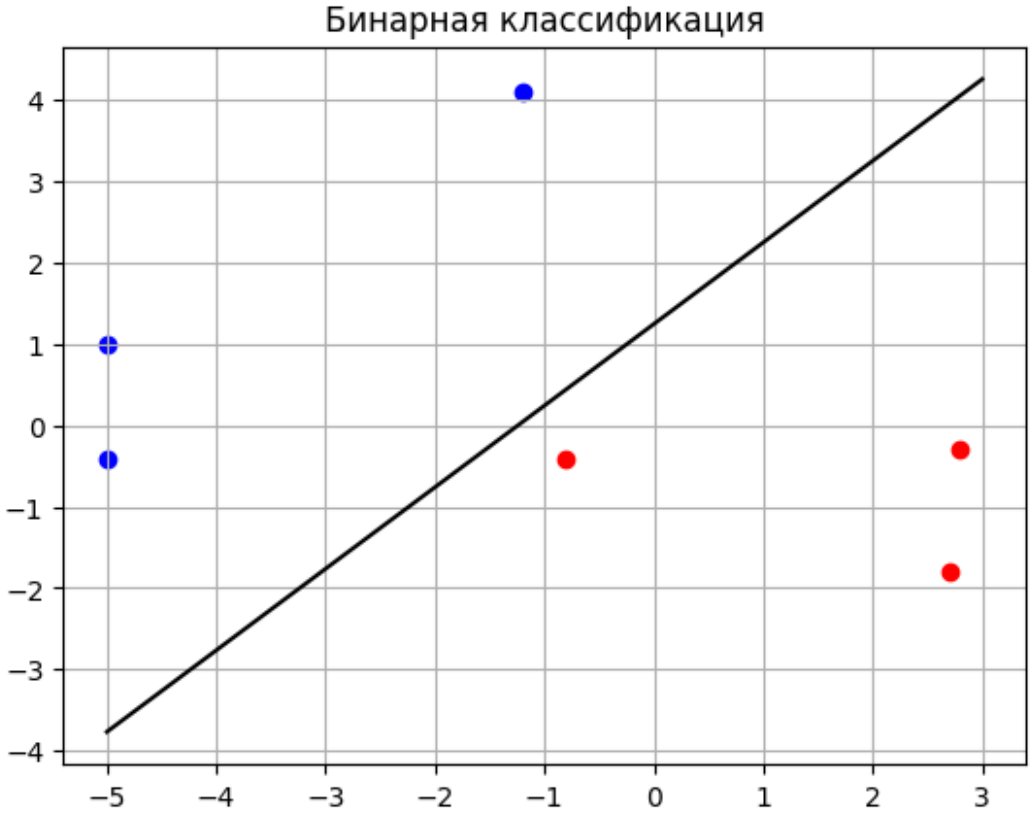
```
weights = model.layers[0].get_weights()
weights
```

```
[array([[ 0.6254261 ],
        [-0.62328756]], dtype=float32),
 array([0.77859956], dtype=float32)]
```

```
def fun(x, w):
    b = w[1][0]
    return (-x * w[0][0] - b) / w[0][1]

pl_x = np.linspace(-5, 3, 200)
pl_y = [fun(x, weights) for x in pl_x]
```

```
x1 = [i[0] for i in data_xy[:3]]
y1 = [i[1] for i in data_xy[:3]]
x2 = [i[0] for i in data_xy[3:]]
y2 = [i[1] for i in data_xy[3:]]
fig, ax = plt.subplots()
ax.scatter(x1, y1, c = 'blue')
ax.scatter(x2, y2, c = 'red')
ax.plot(pl_x, pl_y, c = 'black')
ax.set_title('Бинарная классификация')
plt.grid()
```



Лабораторная работа № 1 по Нейроинформатике

Персептроны. Процедура обучения Розенбланта

Выполнила: Тимофеева Наталья М8О-408Б-19

Вариант № 16

Часть 2

Разбиение на 4 класса

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras import layers
from tensorflow.python.keras import activations
from matplotlib import pyplot as plt
```

Данные для обучения

```
data_xy = np.array([[4.1, -2.2], [2.5, 2.5], [-1.2, 0.6], [-4.3, -
4.5], [0.3, 2.7], [4.3, -3.8], [0.6, -0.4], [-4.9, -1.7]])
labels = np.array([[1, 1], [0, 1], [0, 1], [1, 0], [0, 1], [1, 1], [0,
1], [0, 0]])

model = keras.models.Sequential()
model.add(keras.layers.Dense(2, input_dim = 2, activation='sigmoid',
                             kernel_initializer =
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0),
                             bias_initializer =
keras.initializers.Zeros()))
model.compile(loss = 'mse', optimizer = 'adam', metrics = ['mae'])

model.fit(data_xy, labels, batch_size = 1, epochs = 500)
```

```
Epoch 1/500
8/8 [==============================] - 0s 997us/step - loss: 0.2564 -
mae: 0.4736
Epoch 2/500
8/8 [==============================] - 0s 855us/step - loss: 0.2532 -
mae: 0.4705
Epoch 3/500
8/8 [==============================] - 0s 1ms/step - loss: 0.2498 -
mae: 0.4673
Epoch 4/500
8/8 [==============================] - 0s 854us/step - loss: 0.2463 -
mae: 0.4641
Epoch 5/500
8/8 [==============================] - 0s 1ms/step - loss: 0.2432 -
mae: 0.4611
Epoch 6/500
```

```
mae: 0.1124
Epoch 490/500
8/8 [==============================] - 0s 997us/step - loss: 0.0214 -
mae: 0.1123
Epoch 491/500
8/8 [==============================] - 0s 672us/step - loss: 0.0214 -
mae: 0.1121
Epoch 492/500
8/8 [==============================] - 0s 855us/step - loss: 0.0213 -
mae: 0.1119
Epoch 493/500
8/8 [==============================] - 0s 712us/step - loss: 0.0212 -
mae: 0.1117
Epoch 494/500
8/8 [==============================] - 0s 855us/step - loss: 0.0212 -
mae: 0.1115
Epoch 495/500
8/8 [==============================] - 0s 860us/step - loss: 0.0211 -
mae: 0.1113
Epoch 496/500
8/8 [==============================] - 0s 712us/step - loss: 0.0211 -
mae: 0.1111
Epoch 497/500
8/8 [==============================] - 0s 997us/step - loss: 0.0210 -
mae: 0.1109
Epoch 498/500
8/8 [==============================] - 0s 712us/step - loss: 0.0209 -
mae: 0.1107
Epoch 499/500
8/8 [==============================] - 0s 712us/step - loss: 0.0209 -
mae: 0.1106
Epoch 500/500
8/8 [==============================] - 0s 712us/step - loss: 0.0208 -
mae: 0.1104

<keras.callbacks.History at 0x1e4d3554700>

w, b = model.layers[0].get_weights()
w

array([[ 0.3498479 ,  0.6375578 ],
       [-0.93408614,  0.44285998]], dtype=float32)

b

array([-1.2312579,  1.5048833], dtype=float32)

x1 = [data_xy[7][0]]
y1 = [data_xy[7][1]]
x2 = [i[0] for i in data_xy[[1, 2, 4, 6]]]
y2 = [i[1] for i in data_xy[[1, 2, 4, 6]]]
```

```
x3 = [data_xy[3][0]]
y3 = [data_xy[3][1]]
x4 = [i[0] for i in data_xy[[0, 5]]]
y4 = [i[1] for i in data_xy[[0, 5]]]

fig, ax = plt.subplots()
ax.scatter(x1, y1, c = 'blue')
ax.scatter(x2, y2, c = 'green')
ax.scatter(x3, y3, c = 'red')
ax.scatter(x4, y4, c = 'yellow')
pl_x = np.linspace(-5, 5, 200)
plt.plot(pl_x, (-pl_x * w[0][0] - b[0]) / w[1][0])
plt.plot(pl_x, (-pl_x * w[0][1] - b[1]) / w[1][1])
ax.set_title('Разбиение на 4 класса')
plt.grid()
```