

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
По дисциплине: «Модели решения задач в И С»
Тема: «Бинарная классификация»

Выполнил:
Студент 3 курса
Группы ИИ-26
Шоева Е.Т.
Проверила:
Андренко К.В.

Цель работы: Изучить принципы бинарной классификации и реализовать однослойную нейронную сеть (персептрон) для решения задачи классификации с использованием **пороговой** функции активации, а также исследовать процесс обучения модели с применением среднеквадратичной ошибки (MSE).

Ход работы

Задачи лабораторной работы:

1. Реализовать алгоритм обучения однослойной нейронной сети с использованием **MSE** в качестве функции ошибки (**дельта-правило**).
2. Провести обучение сети с **разными значениями шага обучения** и построить **график зависимости MSE** от номера эпохи.
3. Выполнить визуализацию результатов классификации:
 - исходные точки обучающей выборки,
 - разделяющую линию (границу между двумя классами).
4. Реализовать режим функционирования сети:
 - пользователь задаёт произвольный входной вектор,
 - сеть вычисляет выходной класс,
 - соответствующая точка отображается на графике,
 - для корректной визуализации рекомендуется выбирать значения из диапазона **ВСТАВИТЬ СВОЙ ДИАПАЗОН**, например $-0.5 \leq x_1, x_2 \leq 1.5$
5. Написать вывод по выполненной работе.

Допускается применение **математических** и **графических** библиотек

ML-библиотеки и **ML-фреймворки** использовать **нельзя** (например: scikit-learn, TensorFlow, PyTorch - запрещены)

Вариант:

12.

x_1	x_2	e
6	2	0
-6	2	0
6	-2	1
-6	-2	0

Код:

```
import numpy as np
import matplotlib.pyplot as plt

X = np.array([
    [6, 2],
    [-6, 2],
    [6, -2],
    [-6, -2]
], dtype=float)

y = np.array([0, 0, 1, 0], dtype=float)

X = X / 6.0
X_bias = np.hstack([X, np.ones((X.shape[0], 1))])

learning_rates = [0.01, 0.05, 0.1]
epochs = 50

def net_output(X, w):
    return np.dot(X, w)
```

```

def mse(y_true, y_pred):
    return np.mean((y_true - y_pred) ** 2)

plt.figure(figsize=(9, 6))

for lr in learning_rates:
    w = np.random.randn(3) * 0.1
    mse_history = []

    for epoch in range(epochs):
        y_pred = net_output(X_bias, w)
        error = y - y_pred

        w = w + lr * np.dot(X_bias.T, error) / len(y)

        mse_history.append(mse(y, y_pred))

    plt.plot(mse_history, linewidth=2, label=f"lr = {lr}")

plt.xlabel("Номер эпохи")
plt.ylabel("MSE")
plt.title("Зависимость MSE от номера эпохи")
plt.legend()
plt.grid(True)
plt.xlim(0, epochs)
plt.ylim(0, max(mse_history) + 0.1)
plt.show()

lr = 0.1
w = np.random.randn(3) * 0.1

for epoch in range(epochs):
    y_pred = net_output(X_bias, w)
    error = y - y_pred
    w = w + lr * np.dot(X_bias.T, error) / len(y)

print("Финальные веса:", w)

plt.figure(figsize=(7, 7))

X_plot = X * 6

for i in range(len(X_plot)):
    if y[i] == 0:
        plt.scatter(X_plot[i][0], X_plot[i][1], color='blue', label='Класс 0' if i == 0 else "")
    else:
        plt.scatter(X_plot[i][0], X_plot[i][1], color='red', label='Класс 1')

plt.xlim(-8, 8)
plt.ylim(-5, 5)

x_vals = np.linspace(-8, 8, 200)
x_vals_norm = x_vals / 6
y_vals = -(w[0] * x_vals_norm + w[2]) / w[1]
y_vals = y_vals * 6
plt.plot(x_vals, y_vals, color='green', label="Разделяющая линия")

x1 = float(input("Введите x1: "))
x2 = float(input("Введите x2: "))

user_norm = np.array([x1/6, x2/6, 1])
net = np.dot(user_norm, w)
user_class = 1 if net >= 0.5 else 0

```

```

print("Класс точки:", user_class)

if user_class == 0:
    plt.scatter(x1, x2, color='cyan', s=120, marker='x', label="Ваша точка (класс 0)")
else:
    plt.scatter(x1, x2, color='magenta', s=120, marker='x', label="Ваша точка (класс 1)")

plt.xlabel("x1")
plt.ylabel("x2")
plt.title("Классификация однослойной нейронной сетью")
plt.legend()
plt.grid(True)
plt.show()

```

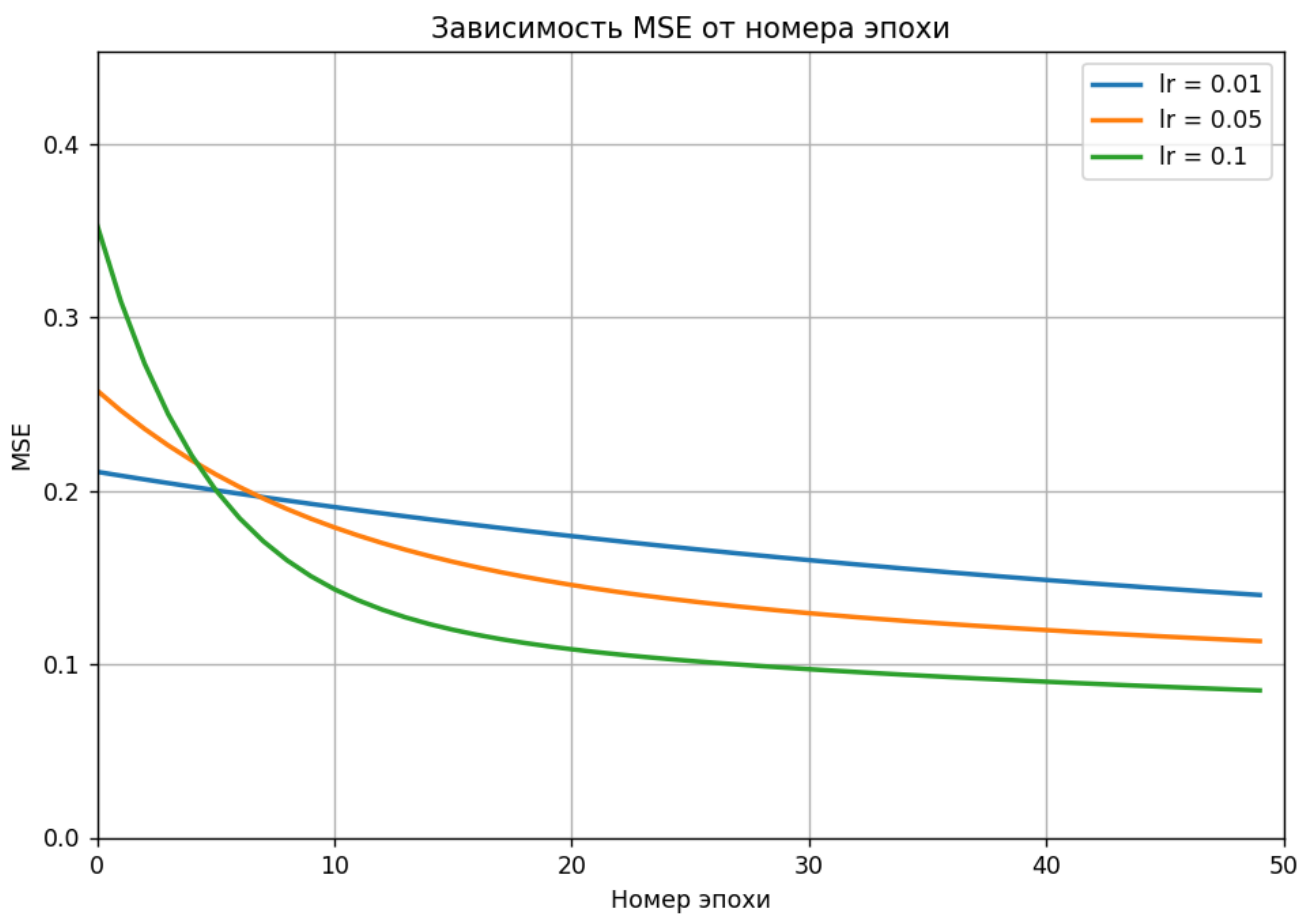
Вывод:

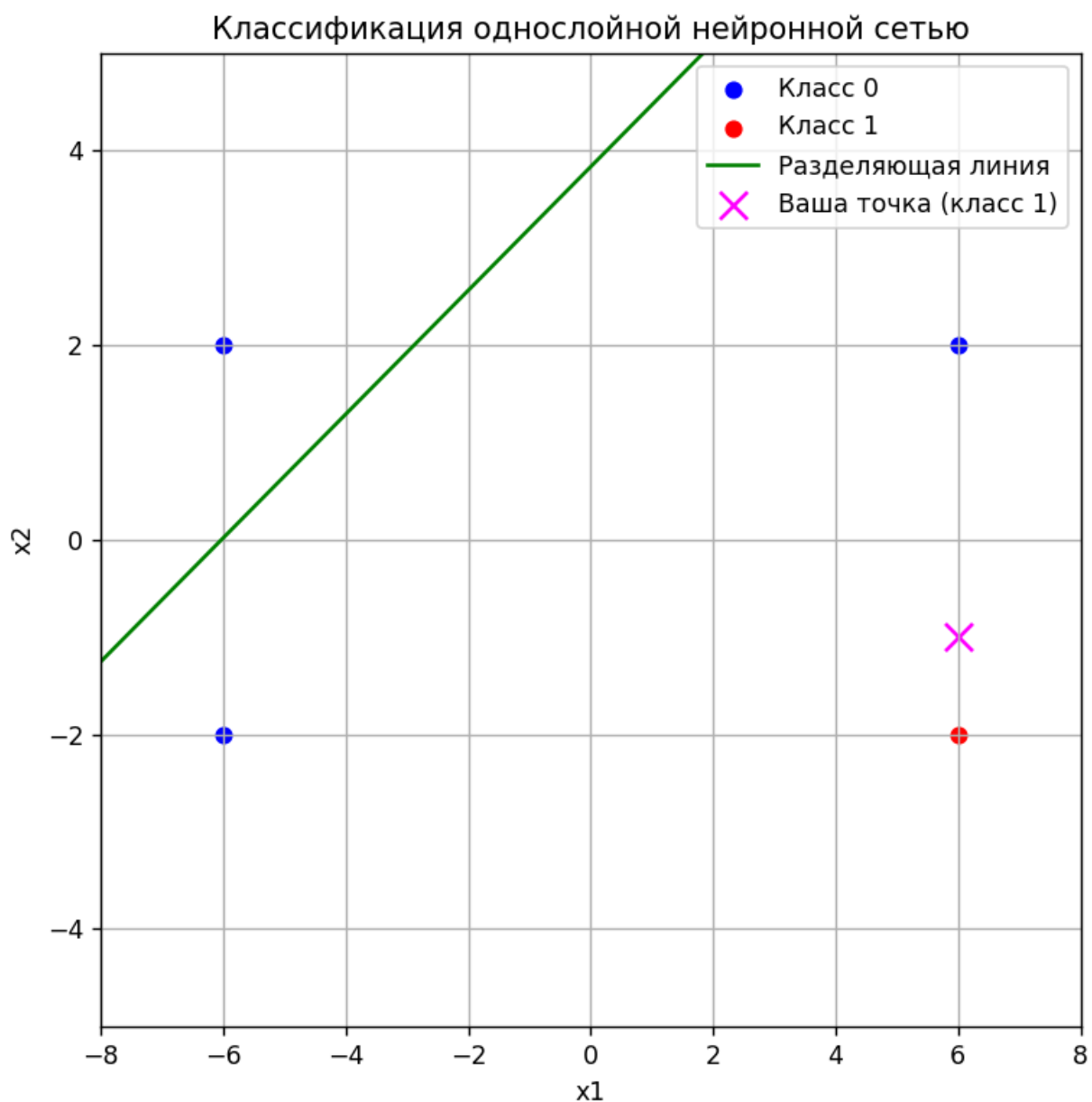
```

Финальные веса: [ 0.24761522 -0.38958532  0.2490609 ]
Введите x1: 6
Введите x2: -1
Класс точки: 1

```

Графики:





Вывод: Изучила принципы бинарной классификации и реализовала однослойную нейронную сеть (персептрон) для решения задачи классификации с использованием **пороговой** функции активации, а также исследовать процесс обучения модели с применением среднеквадратичной ошибки (MSE).