

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

Лабораторная работа №1

По дисциплине «Модели решения задач в интеллектуальных системах»

Тема: «Бинарная классификация»

Выполнил:

Студент 3 курса

Группы ИИ-26

Петручик Д.А

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:

1. Реализовать алгоритм обучения однослойной нейронной сети с использованием **MSE** в качестве функции ошибки.
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 $-0.5 \leq x_1, x_2 \leq 1.5$

10.

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

x_1, x_2 - входные данные сети, e - эталонные значения

Код программы:

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

X = np.array([[6, 6], [-6, 6], [6, -6], [-6, -6]] )
e = np.array([0, 0, 1, 0])

learning_rates = [0.005, 0.01, 0.02]
epochs = 200
np.random.seed(42)

models = {}

for lr in learning_rates :
    w = np.random.uniform(-0.5, 0.5, 2)
```

```

w0 = np.random.uniform(-0.5, 0.5)
mse_history = []

for epoch in range(epochs) :
    epoch_errors = []
    for i in range(len(X)) :
        S = np.dot(X[i], w) + w0
        delta = e[i] - S
        w += lr * delta * X[i]
        w0 += lr * delta
        epoch_errors.append(delta * *2)
    mse_history.append(np.mean(epoch_errors))

models[lr] = (w.copy(), w0, mse_history)

plt.figure(figsize = (8, 4))
for lr, (_, _, mse_history) in models.items() :
    plt.plot(mse_history, linewidth = 2, label = f"lr={lr}")

    plt.title("График изменения ошибки (MSE)")
    plt.xlabel("Эпоха")
    plt.ylabel("Среднеквадратичная ошибка")
    plt.legend()
    plt.grid(True)
    plt.show()

best_lr = min(models, key = lambda k : models[k][2][-1])
w, w0, mse_history = models[best_lr]

print(f"\nВыбрана модель с learning_rate = {best_lr}")
print(f"Веса: w1 = {w[0]:.4f}, w2 = {w[1]:.4f}")
print(f"Смещение: w0 = {w0:.4f}")
print(f"Финальная MSE: {mse_history[-1]:.6f}")

plt.figure(figsize = (8, 6))

for i in range(len(X)) :
    if e[i] == 0 :
        plt.scatter(X[i, 0], X[i, 1], color = 'blue', s = 100,
                    label = 'Класс 0' if i == 0 else "")
    else :
        plt.scatter(X[i, 0], X[i, 1], color = 'red', s = 100,
                    label = 'Класс 1')

    x_line = np.linspace(-10, 10, 100)
    if w[1] != 0 :
        y_line = (0.5 - w0 - w[0] * x_line) / w[1]
        plt.plot(x_line, y_line, 'g--', label = 'Разделяющая
поверхность (S=0.5)')

    plt.xlim(-10, 10)
    plt.ylim(-10, 10)
    plt.title("Визуализация разделения областей 2-х классов")
    plt.axhline(0, color = 'black', alpha = 0.3)
    plt.axvline(0, color = 'black', alpha = 0.3)
    plt.xlabel("x1")
    plt.ylabel("x2")
    plt.legend()
    plt.grid(True)
    plt.show()

    print("\nРежим тестирования:")
    print("Формат ввода: x1, x2 (например: 6, 6 или -6, -6)")

    while True:

```

```

user_input = input("\nВведите x1, x2 через запятую или 'q' для выхода: ")
if user_input.lower() == 'q' :
    break

    try :
        test_point = np.array([float(c.strip()) for c in user_input.split(',')])
        S_test = np.dot(test_point, w) + w0
        y_final = 1 if S_test >= 0.5 else 0
        print(f"Результат: Класс {y_final} (S = {S_test:.4f})")

    if y_final == 1:
print("→ Точка относится к классу 1")
    else:
print("→ Точка относится к классу 0")

except ValueError :
print("Ошибка ввода! Введите два числа через запятую.")
except Exception as e :
print(f"Ошибка: {e}")

```

Результат тестирования:

```

Обучение завершено!
Веса: w1 = 0.0653, w2 = -0.0413
Смещение: w0 = 0.2500
Финальная MSE: 0.155000

Режим тестирования:
Формат ввода: x1, x2 (например: 6, 6 или -6, -6)

Введите x1, x2 через запятую или 'q' для выхода: 6,6
Результат: Класс 0 (S = 0.3937)
→ Точка относится к классу 0

Введите x1, x2 через запятую или 'q' для выхода: -6,6
Результат: Класс 0 (S = -0.3898)
→ Точка относится к классу 0

Введите x1, x2 через запятую или 'q' для выхода: 6,-6
Результат: Класс 1 (S = 0.8898)
→ Точка относится к классу 1

Введите x1, x2 через запятую или 'q' для выхода: -6,-6
Результат: Класс 0 (S = 0.1063)
→ Точка относится к классу 0

```

График среднеквадратичной ошибки:

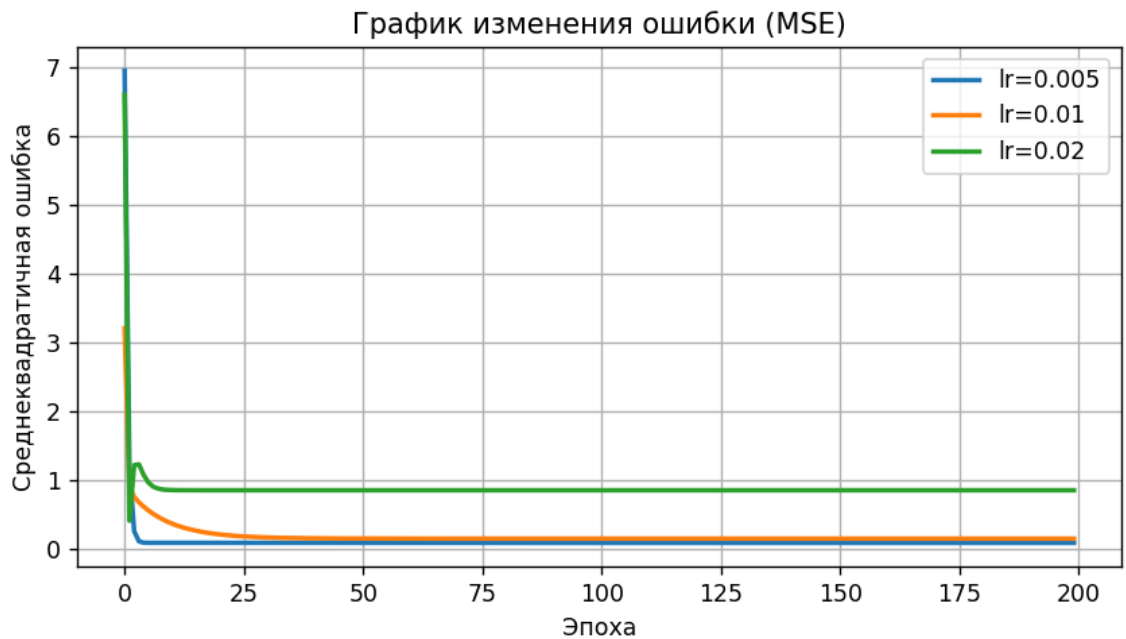
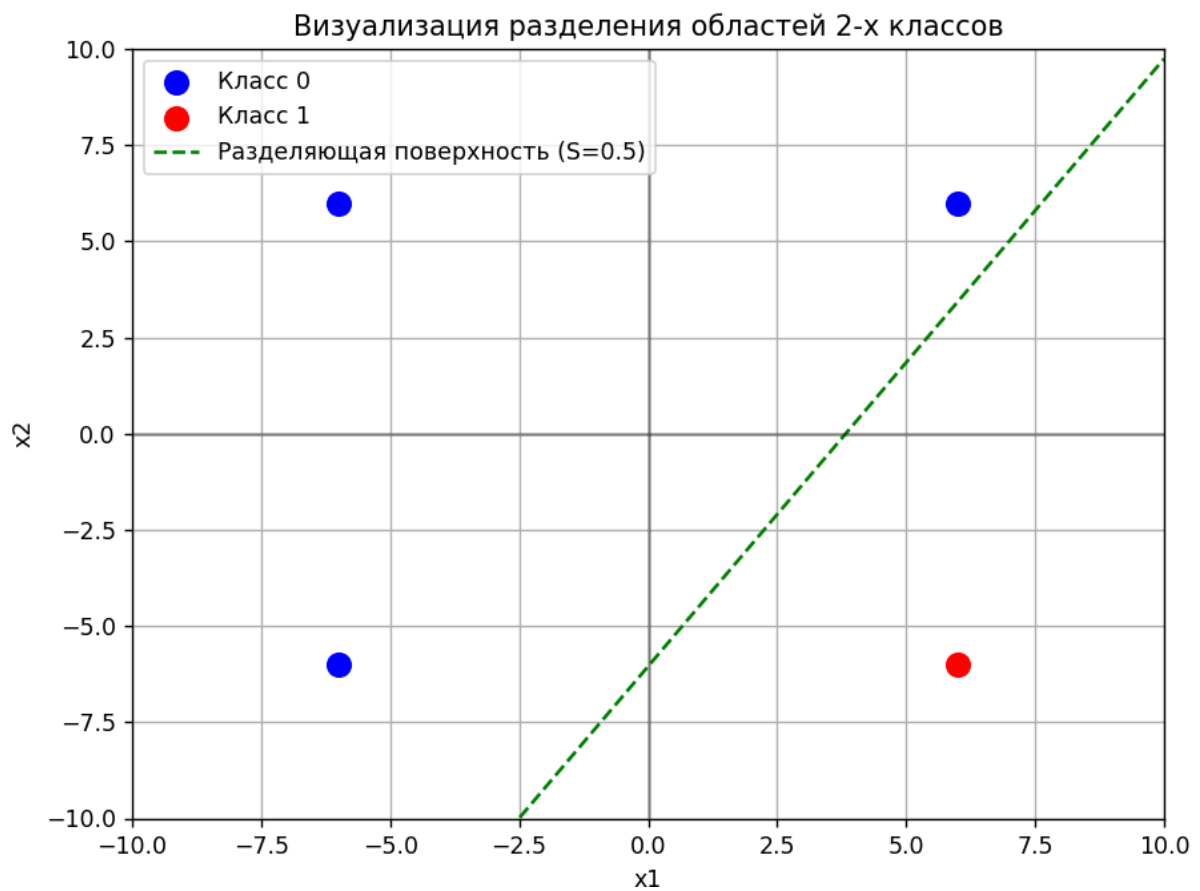


График с разделяющей поверхностью:



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