

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

## Отчет по лабораторной работе №1

Специальность ИИ26(з)

Выполнил  
Т.В.  
Данилюк,  
студент группы ИИ26

Проверила  
К.В. Андренко,  
ст. преп. кафедры ИИТ,  
«       2026 г.

Брест 2026

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

**Задание 1. Для заданного массива вещественных чисел найти среднее значение и стандартное отклонение.**

Задачи лабораторной 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  $-0.5 \leq x_1, x_2 \leq 1.5$
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)

Код программы:

Содержимое в файле main.py

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

class PerceptronADALINE:
    def __init__(self, input_size, learning_rate=0.01):
        np.random.seed(42)
        self.w = np.random.uniform(-0.5, 0.5, input_size)
        self.w0 = np.random.uniform(-0.5, 0.5)
        self.lr = learning_rate

    # ===== ЛИНЕЙНЫЙ ВЫХОД (ADALINE) =====
    def net(self, X):
        return np.dot(X, self.w) + self.w0

    # ===== ПОРОГОВАЯ ФУНКЦИЯ (ПЕРСЕПТРОН) =====
    def step(self, S):
        return np.where(S >= 0.5, 1, 0)

    # ===== ЭТАП 1: ОБУЧЕНИЕ ADALINE =====
    def train_adaline(self, X, target, epochs=200):
```

```

mse_history = []

for epoch in range(epochs):
    errors = []

    for i in range(len(X)):
        S = self.net(X[i])      # ЛИНЕЙНЫЙ выход!
        delta = target[i] - S   # ошибка ДО активации

        # дельта-правило
        self.w += self.lr * delta * X[i]
        self.w0 += self.lr * delta

    errors.append(delta ** 2)

    mse_history.append(np.mean(errors))

return mse_history

# ===== ЭТАП 2: РАБОТА ПЕРСЕПТРОНА =====
def predict(self, X):
    S = self.net(X)
    return self.step(S)

# ===== ДАННЫЕ =====
X = np.array([[1, 4],
              [-1, 4],
              [1, -4],
              [-1, -4]])

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

# ===== ОБУЧЕНИЕ =====
p = PerceptronADALINE(input_size=2, learning_rate=0.01)
mse = p.train_adaline(X, target, epochs=200)

# ===== ГРАФИК MSE =====
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 2)
plt.plot(mse, linewidth=2)
plt.title("MSE во время обучения (ADALINE)")
plt.xlabel("Эпоха")
plt.ylabel("Ошибка")
plt.grid(True)

# ===== ВИЗУАЛИЗАЦИЯ РАЗДЕЛЕНИЯ =====
plt.subplot(1, 2, 1)

plt.scatter(X[:3, 0], X[:3, 1], s=100, label='Класс 0')
plt.scatter(X[3:, 0], X[3:, 1], s=100, label='Класс 1')

x_line = np.linspace(-6, 6, 200)
y_line = (0.5 - p.w0 - p.w[0] * x_line) / p.w[1]
plt.plot(x_line, y_line, 'r', label='Разделяющая прямая')

plt.xlim(-6, 6)

```

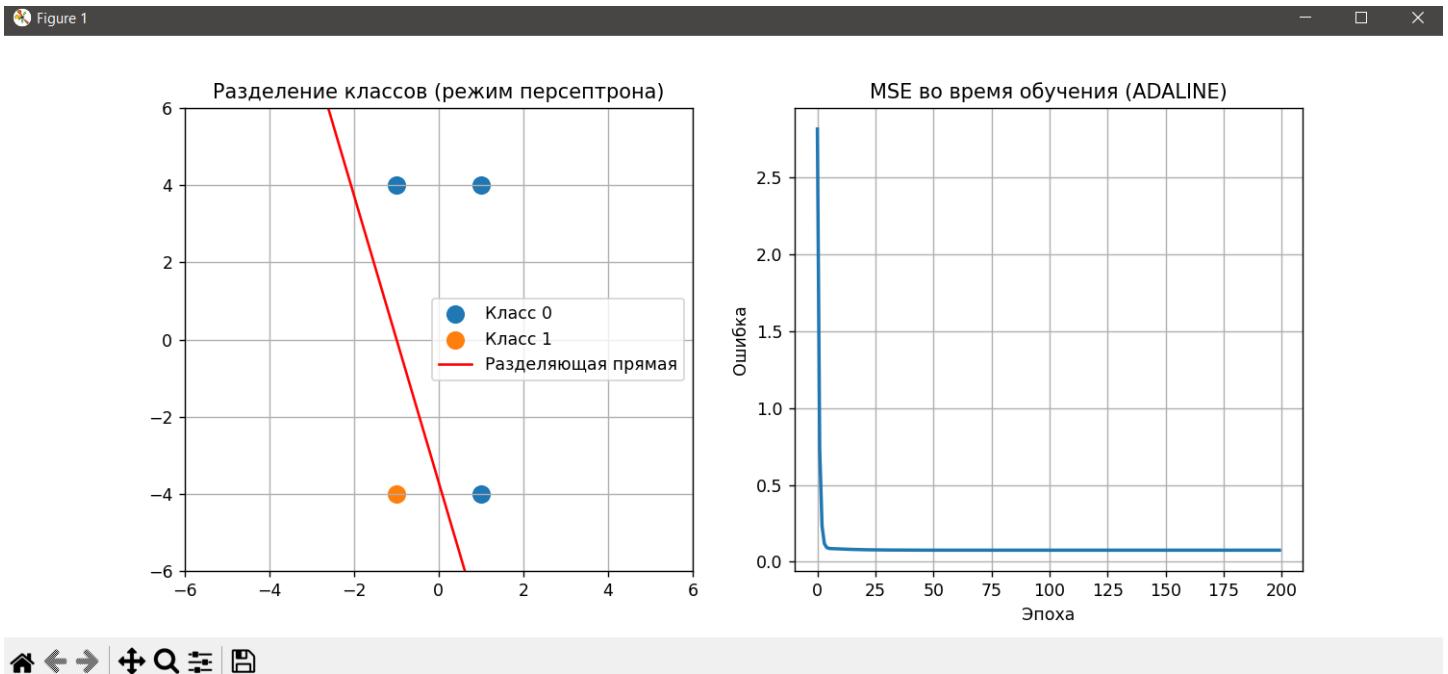
```
plt.ylim(-6, 6)
plt.grid(True)
plt.legend()
plt.title("Разделение классов (режим персептрана)")

plt.show()

# ===== РЕЖИМ ТЕСТИРОВАНИЯ =====
print("\nРежим работы персептрана")
while True:
    user_input = input("Введите x1 x2 или 'q': ")
    if user_input.lower() == 'q':
        break

    try:
        x1, x2 = map(float, user_input.split())
        pred = p.predict(np.array([x1, x2]))
        print(f"Класс: {pred}")
    except:
        print("Ошибка ввода")
```

## Рисунки с результатами работы программы



```
Режим работы персептрана
Введите x1 x2 или 'q': 1 4
Класс: 0
Введите x1 x2 или 'q': -1 4
Класс: 0
Введите x1 x2 или 'q': 1 -4
Класс: 0
Введите x1 x2 или 'q': -1 -4
Класс: 1
Введите x1 x2 или 'q': q

Process finished with exit code 0
```

**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).