

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

Лабораторная работа №1
За шестой семестр
По дисциплине: «Модели решения задач в ИС»
Тема: «Бинарная классификация»

Выполнил:
Студент 3 курса
Группы ИИ-26(1)
Пилипук М.А.
Проверил:
Андренко К.В.

Брест 2026

Цель: 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 среднеквадратической ошибки (MSE).

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 , например - $0.5 \leq x_1, x_2 \leq 1.5$

Код программы содержится в файле `main.py`

Результаты работы программы для набора входных данных $x_1 = (6, -6, 6, -6)$, $x_2 = (1, 1, -1, -1)$ и эталонным выходом $y = (0, 0, 0, 1)$.

Для обучения сети используется формула потерь MSE и формулы градиентов.

В результате обучения получаем следующие графики:

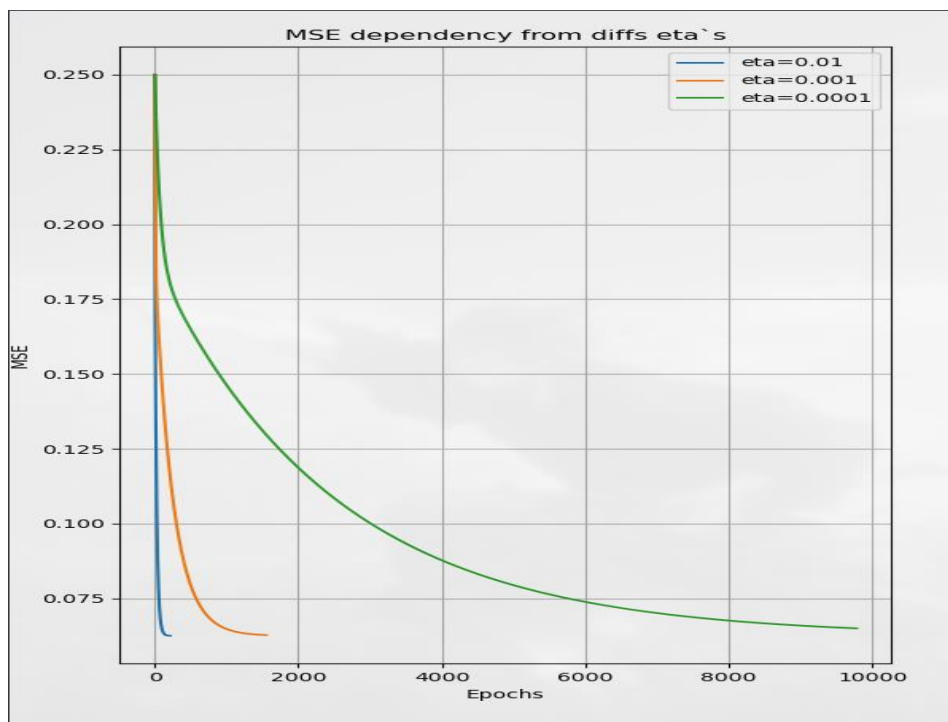


График зависимости MSE от номера эпохи для разных значений шага обучения.

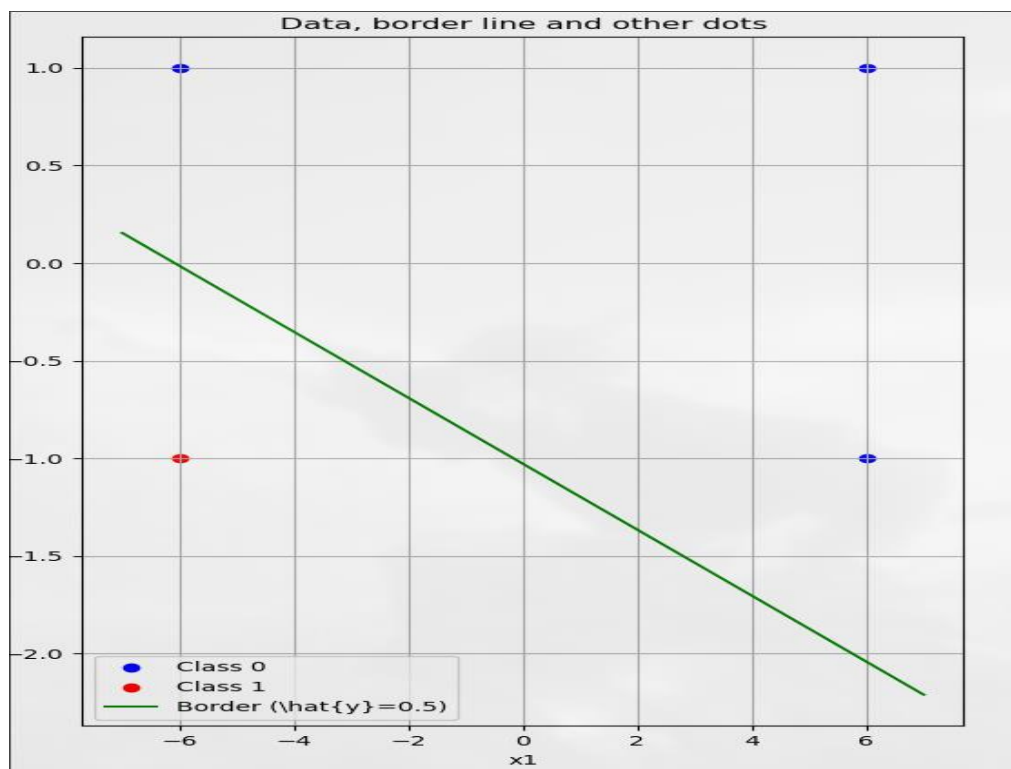


График визуализации результатов классификации.

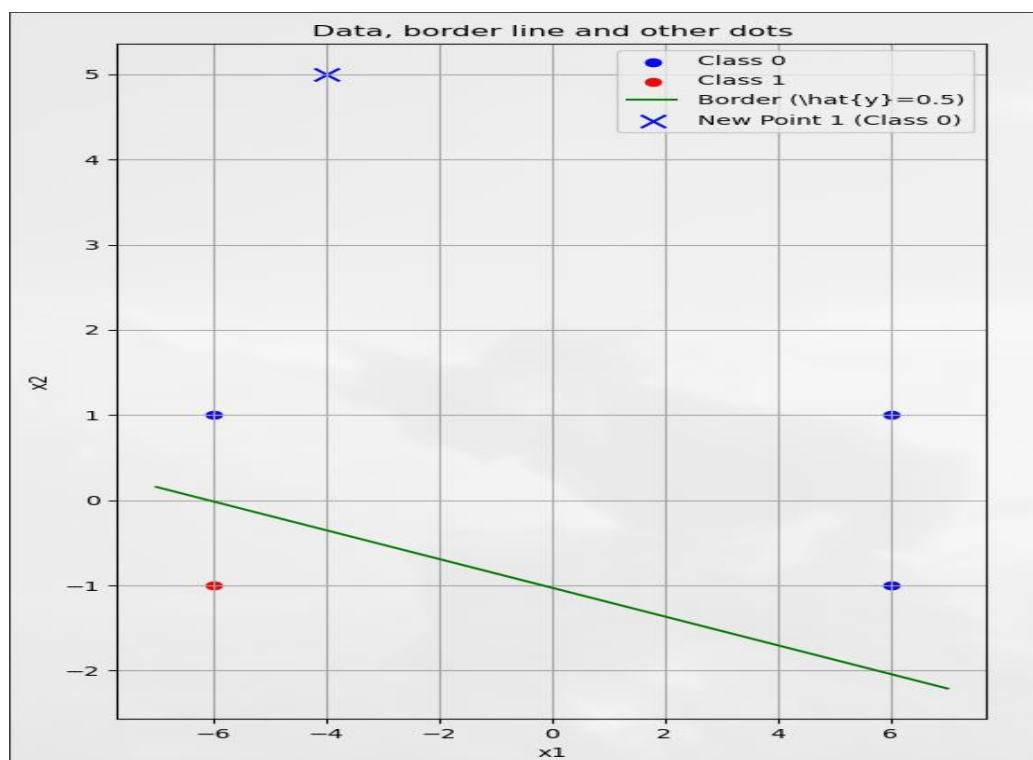


График отображения пользовательского входного вектора.

Вывод: в ходе данной лабораторной работы я изучил концепцию простой однослойной нейронной сети и применил её для решения задачи бинарной классификации. Для выполнения работы также изучил алгоритм MSE для расчёта ошибки и градиенты для корректировки весов. Было проведено тестирование с различными величинами шага обучения, в результате которого можно сделать

вывод, что большие значения шага дают наиболее быстрый результат обучения, тогда как наименьшие дают максимально точный результат, жертвуя скоростью. Это прекрасно проиллюстрировано на графике, где для шага 0.01 понадобилось всего ~262 эпох, тогда как для 0.0001 - практически 10000.