

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет»
(национальный исследовательский университет)
Институт естественных и точных наук
Кафедра прикладной математики и программирования

Отчет по лабораторной работе № 1
по дисциплине «Современные нейросетевые технологии»

Авторы работы
Студент группы ЕТ-122
_____/ Матвеева А.В.
«____» _____ 2025 г.

Руководитель работы,
_____/ Кичеев Д.М.
«____» _____ 2025 г.

Челябинск 2025

Задание:

Цель: настоящей работы состоит в том, чтобы изучить метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой).

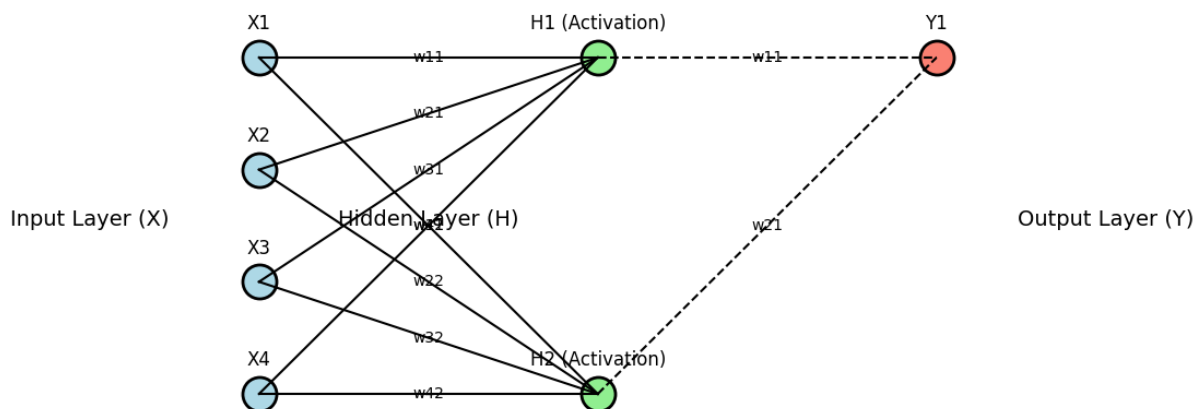
Задачи:

Выполнение практической работы предполагает решение *следующих задач*:

1. Изучение общей схемы метода обратного распространения ошибки.
2. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
3. Проектирование и разработка программной реализации.
4. Тестирование разработанной программной реализации.
5. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

1. В ходе выполнения лабораторной работы мной был изучен метод обратного распространения ошибки, были выведены необходимые формулы.

Образец простой полносвязной сети



2. Были найдены формулы обучения каждого этапа модели, формулы расчета ошибки для модели множественной классификации и другие необходимые формулы.

Были выведены формулы ошибки на последнем слое:

$$\delta^{(2)} = \frac{\partial L}{\partial z^{(2)}} = \hat{y} - y$$

Расчет градиентов функции ошибки по параметрам выходного слоя

$$\frac{\partial L}{\partial W^{(2)}} = \delta^{(2)} \cdot (a^{(1)})^\top$$

Расчет ошибки скрытого слоя:

$$\delta^{(1)} = \frac{\partial L}{\partial z^{(1)}} = (W^{(2)})^\top \delta^{(2)} \cdot \sigma'(z^{(1)})$$

Расчет градиентов функции для параметров последнего и скрытого слоя, например:

$$\frac{\partial L}{\partial W^{(1)}} = \delta^{(1)} \cdot x^\top$$

Формулы обновления параметров:

$$W^{(1)} := W^{(1)} - \eta \frac{\partial L}{\partial W^{(1)}}$$

В ходе выполнения работы был пошагово описан процесс обучения нейронной сети, включая метод обратного распространения ошибки.

3. Была разработана программная реализация вышеуказанного кода.

Например:

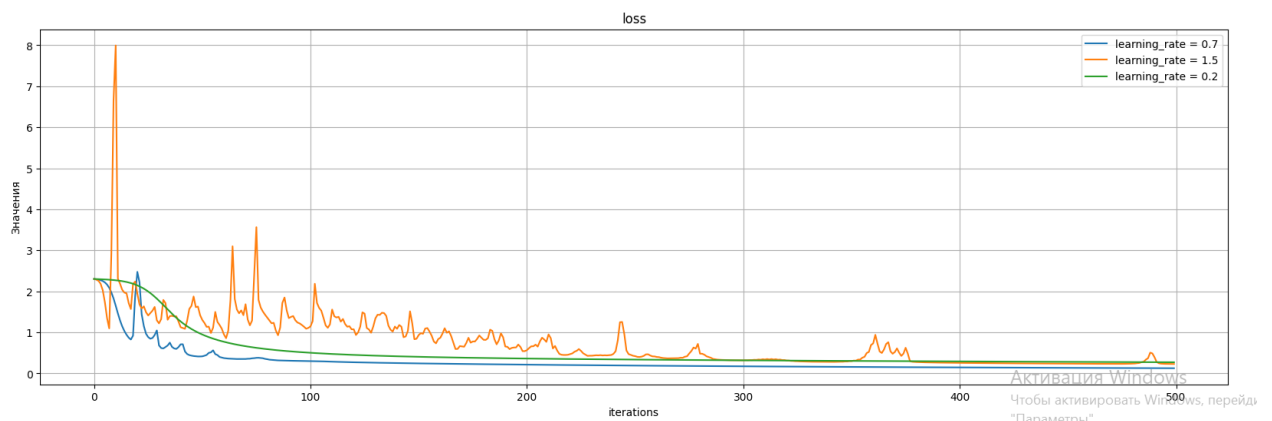
```
[ ] def relu_derivative(z):
    return (z > 0).astype(float)
```

```
def backward_propagation(X, Y, Z1, A1, A2, W2):
    """
    Обратное распространение.
    """
    m = X.shape[1]
    dZ2 = A2 - Y
    dW2 = np.dot(dZ2, A1.T) / m
    db2 = np.sum(dZ2, axis=1, keepdims=True) / m

    dA1 = np.dot(W2.T, dZ2)
    dZ1 = dA1 * relu_derivative(Z1)
    dW1 = np.dot(dZ1, X.T) / m
    db1 = np.sum(dZ1, axis=1, keepdims=True) / m

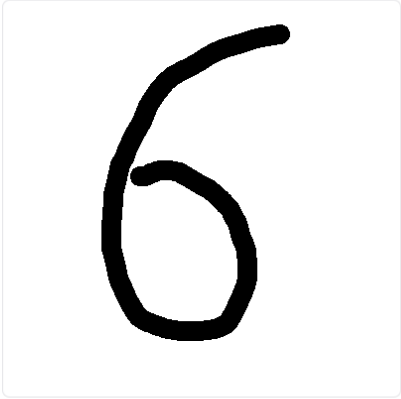
    return dW1, db1, dW2, db2
```

4. Программа была протестирована на различных гиперпараметрах.



Дополнительно, было разработано приложение, которое позволяет пользователю рисовать цифру и модель классифицирует данную цифру. Приложение было развернуто на Hugging Face. Код, модель, ноутбук с кодом разработки выложены на github.

Нарисуйте число



Layer 1

Clear

Submit

output

Предсказанное число: 6

Вероятности:

Класс 0:	0.0%
Класс 1:	0.01%
Класс 2:	0.0%
Класс 3:	0.01%
Класс 4:	0.03%
Класс 5:	11.84%
Класс 6:	87.86%
Класс 7:	0.0%
Класс 8:	0.26%
Класс 9:	0.0%

Вывод:

В данном проекте были выведены формулы для реализации метода обучения нейронных сетей - обратного распространения ошибки. На основе этих формул была создана нейросеть с одним скрытым кодом. Модель была обучена и протестирована. Дополнительно создано и развернуто приложение для тестирования модели.