|  |  |
| --- | --- |
| Picture 1 | **МИНОБРНАУКИ РОССИИ**  **федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  «Санкт-Петербургский государственный морской технический университет» (СПбГМТУ) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет цифровых промышленных технологий

Направление подготовки 09.03.01

"**Интеллектуальные технологии киберфизических систем**"

Лабораторная работа №3

Студент 2 курса группы 20221

Очного отделения

Руденко Вячеслав Сергеевич

Проверил:

Преподаватель CПбГМТУ

Кайнова Татьяна Денисовна

Санкт-Петербург

2024

Оглавление

[1. ОБЩАЯ ЧАСТЬ 3](#_Toc182400927)

1. [1.1 Цель работы 3](#_Toc182400928)
2. [1.2 Формулировка задачи 3](#_Toc182400929)
3. [1.3 Исходные данные 3](#_Toc182400930)

[2. ХОД РАБОТЫ 4](#_Toc182400931)

[Инициализация весов 4](#_Toc182400932)

[2.2 Прямое распространение (forward propagation) 5](#_Toc182400933)

[2.3 Обратное распространение ошибки (backpropagation) 7](#_Toc182400934)

[ПРИЛОЖЕНИЕ А 11](#_Toc182400935)

# ОБЩАЯ ЧАСТЬ

## 1.1 Цель работы

Целью работы является разработка и реализация алгоритма обучения многослойной нейронной сети с использованием метода обратного распространения ошибки. При этом сеть должна иметь не менее двух скрытых слоев, что позволит изучить принципы построения более сложных архитектур нейронных сетей.

1.2 Формулировка задачи

В рамках выполнения работы необходимо:

1. Реализовать алгоритм прямого и обратного проходов для многослойной нейронной сети.
2. Обеспечить наличие в сети:

* не менее двух скрытых слоев;
* более двух нейронов в каждом скрытом слое.

1. Реализовать процесс обучения сети на основе метода обратного распространения ошибки.
2. Проверить корректность работы алгоритма на наборе тренировочных данных.1.3 Исходные данные

# 2. ХОД РАБОТЫ

Используется фиксированный seed: np.random.seed(). Это позволяет при каждом запуске программы получать одни и те же случайные значения.

Скорость обучения (LEARNING\_RATE) задается как параметр, определяющий, насколько сильно будут изменяться веса на каждом шаге обучения.

Входные данные (x\_train) представляют собой массив, в котором каждая строка — это набор входных значений для сети. Они включают bias. Выходные данные (y\_train) содержат целевые значения, которым должны соответствовать выходы сети после обучения.

Все веса нейронной сети кладутся в массив n\_w, где:

* n\_w[0] — веса для первого скрытого слоя.
* n\_w[1] — веса для второго скрытого слоя.
* n\_w[2] и n\_w[3] — веса для следующих слоев, включая выходной.

Каждый нейрон сети имеет собственные веса, которые изначально инициализируются случайными значениями в диапазоне от -1.0 до 1.0.

Нейронная сеть имеет два скрытых слоя с двумя нейронами в каждом и один выходной слой. Для каждого слоя применяются различные функции активации:

Для скрытых слоев используется функция tanh, которая выдает значения в диапазоне от -1 до 1.

Для выходного слоя применяется логистическая сигмоида, которая выдает значения в диапазоне от 0 до 1.

n\_y — массив, который используется для хранения выходных значений каждого слоя.

n\_error — массив, который используется для хранения для хранения ошибок, вычисленных на этапе обратного прохода.

index\_list — список индексов обучающей выборки, который используется для перемешивания данных перед каждой эпохой.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была разработана персептронная модель, состоящая из нескольких нейронов, реализующих последовательные логические операции по блок-схеме. Программа успешно выполняет заданную функцию, отображает финальные значения весов для каждой операции и визуализирует разделяющие линии на графиках, что позволяет наглядно продемонстрировать, как модель обрабатывает различные комбинации входных значений. Таким образом, цель работы достигнута, и построенная модель может быть использована для визуализации и анализа логической зависимости, заданной на блок-схеме.

# ПРИЛОЖЕНИЕ А

Листинг кода

Листинг 1 - perseptron.py