**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт**

(Национальный Исследовательский Университет)»

Институт №3 «Системы управления, информатика и электроэнергетика».

Кафедра 311

Направление подготовки Прикладная информатика (09.03.03)

Группа М3О-418Бк-19

**Отчет к лабораторной работе №2**

По дисциплине: Интеллектуальные системы

Работу выполнил студент 4-го курса:

Дмитриев Дмитрий Андреевич

Москва, 2023 г.

**Оглавление**

[**Введение** 2](#_Toc114124284)

[**Глава 1. Решение** 4](#_Toc114124285)

[**Глава 2. Результат работы программы** 5](#_Toc114124286)

[**Глава 3. Листинг программы** 6](#_Toc114124287)

[**Вывод** 7](#_Toc114124288)

[**Список литературы** 8](#_Toc114124289)

**Лабораторная работа №** **2**

**Алгоритм Кохонена**

# **Введение**

Алгоритм обучения без учителя предполагает наличие лишь вводных данных. Алгоритмы обучения нейронных сетей без учителя корректируют весовые коэффициенты таким образом, чтобы НС могла из схожих по некоему принципу данных на «входе» выдать результат, обнаруживающий другие взаимосвязи и закономерности между этими данными. В процессе обучения происходит выделение параметров, характерных для моделей обучающего материала, и дальнейшее объединение этих моделей в группировки по схожим признакам.

Данные, которые поступают на «вход», после обработки нейросетью сложатся в тот или иной ответ. Однако до обучения нельзя предугадать, в какой форме этот ответ поступит. Соответственно, сам процесс обучения должен обуславливать трансформацию результата в понятную форму. Это не представляет сложностей. Как правило, можно легко отследить, какую взаимосвязь задала данным нейросеть в процессе их обработки.

Алгоритмы обучения нейросетей без учителя используют данные без классификации или меток. НС сама выстраивает логическую цепочку и усваивает понимание этих действий, ориентируясь лишь на вводные данные. По сути, это повторяет человеческое самообучение: индивид, предпринимая какие-либо действия, делает выводы о правильности либо ошибочности решения, ориентируясь на последствия.

**Задание:**

Реализовать алгоритм Кохонена.

**Глава 1. Решение [1, 2]**

Конкурентное обучение — это форма обучения без учителя, в которой составные элементы соревнуются за получение удовлетворительного результата, и только один из них может выиграть соревнование.

Когда обучающий пример вводится в сетку, определяется Best Matching Unit (BMU) (победитель конкурса). BMU — это ячейка, веса которой ближе всего к обучающему примеру.

Алгоритм нахождения ячейки победителя:

- Инициализация всех весов сетки. Повторяем до тех пор, пока не будет достигнута максимальная эпоха.

- Перемешиваем обучающие примеры.

- Для каждого тренировочного экземпляра x находим ячейку, веса которой ближе всего к обучающему примеру: наиболее подходящий блок BMU.

Алгоритм решения:

1) Обновляю весовой вектор BMU и его соседних ячеек (рисунок 1).

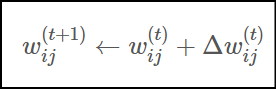


Рисунок 1 – Подсчёт весового вектора

2) Сравниваю весовой вектор каждой ячейки с входным вектором (рисунок 2).



Рисунок 2 – Подсчёт приращения весового вектора

Нахожу функцию приращения весов на каждой итерации (рисунок 3).

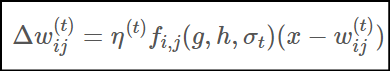


Рисунок 3 – Подсчёт приращения весового вектора на итерации

3) Задаю следующие переменные:

- t номер эпохи (итерация);

- (g, h) координаты BMU;

- η скорость обучения (рисунок 4);

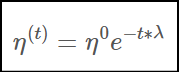


Рисунок 4 – Подсчёт скорости обучения на каждой итерации

- радиус окрестности (рисунок 5);

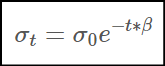
****

Рисунок 5 – Подсчёт радиуса окрестности

- функция расстояния окрестности (рисунок 6).

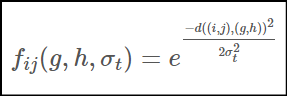


Рисунок 6 – Подсчёт функции расстояния окрестности

4) Создаю сетку SOM и визуализирую, как близкие по весу цвета располагаются в соседних ячейках (рисунок 7).

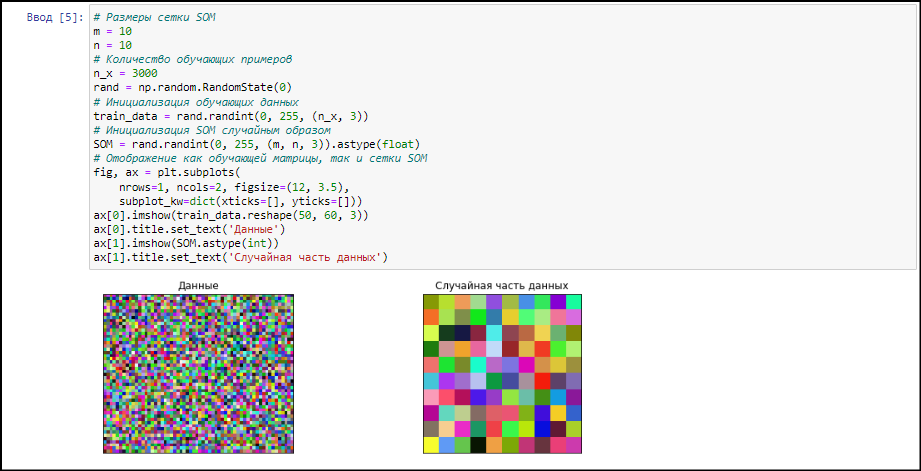


Рисунок 7 – Создание сетки SOM

Обучаю сетку SOM и визуализирую, как близкие по весу цвета располагаются в соседних ячейках (рисунок 8).



Рисунок 8 – Обучение сетки SOM

**Глава 2. Результат работы программы**

Пример работы программы (рисунок 7).

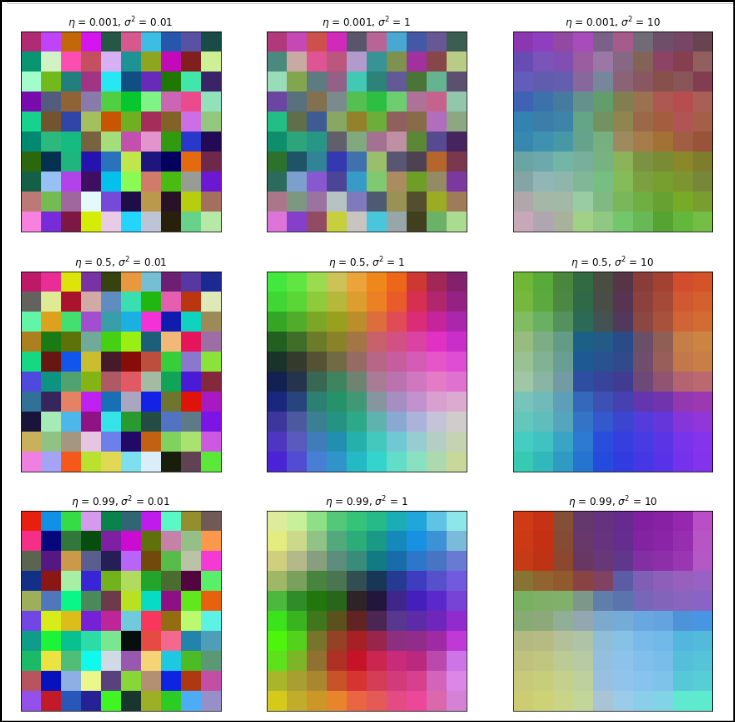
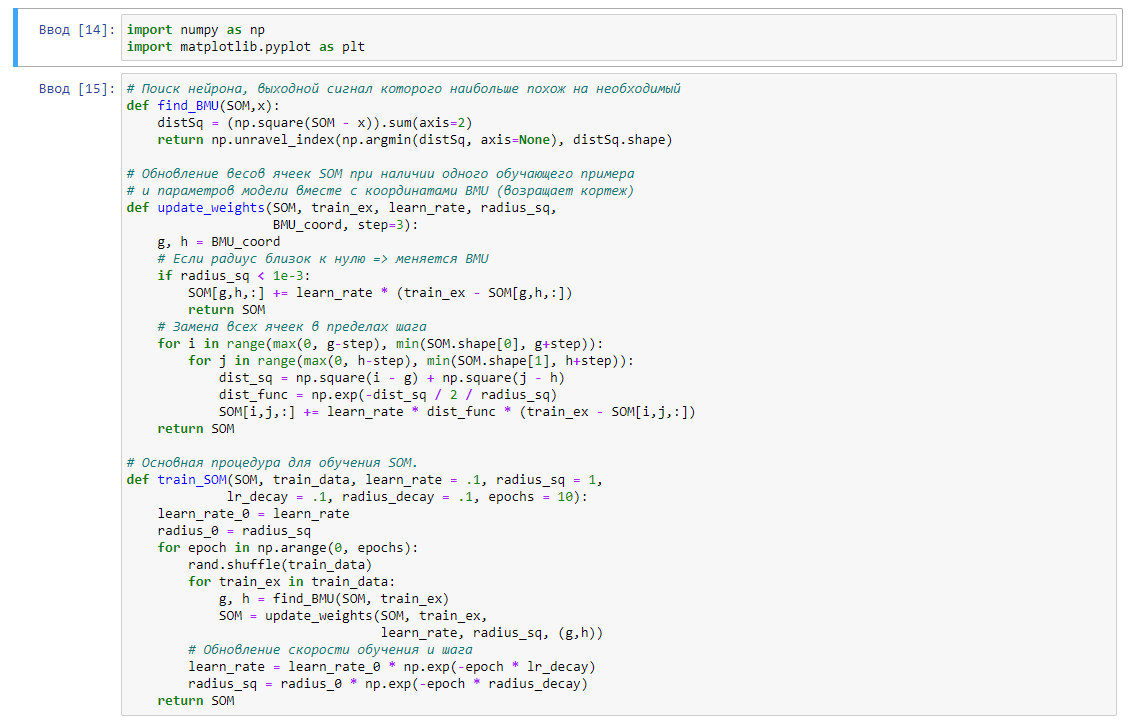
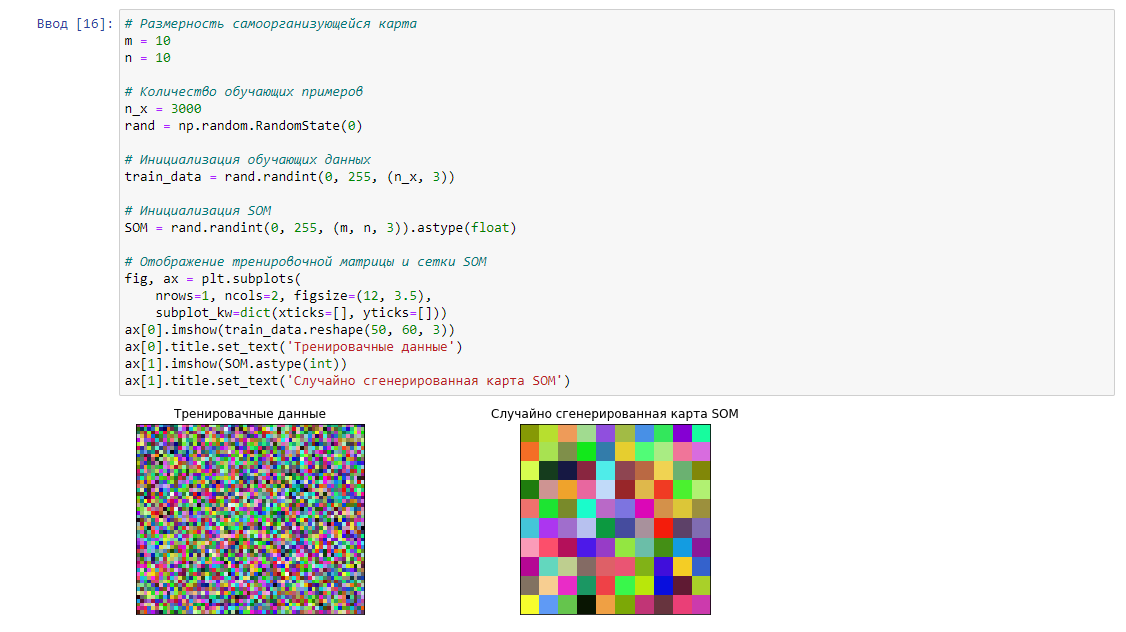
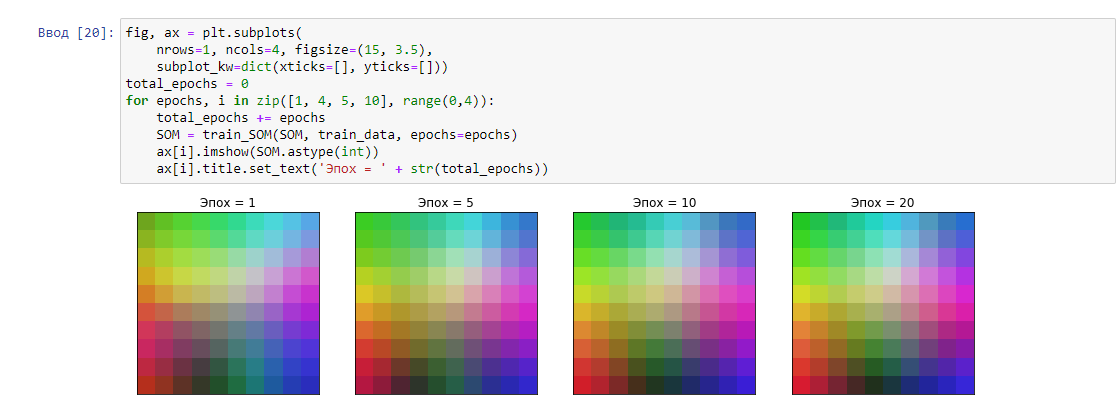


Рисунок 7 – Вывод программы

# **Глава 3. Листинг программы**







# **Вывод**

В ходе выполнения лабораторной работы я познакомился с алгоритмом Кохонена. Я успешно реализовал модель нейронной сети данного алгоритма. При обучении без учителя нейросети была предложен входной вектор данных. Данные поданные на вход сети, получили правильный выход, который выдала НС после обработки внутри своей структуры.

# **Список литературы**

1. Python documentation. [Электронный ресурс] // Python. URL: https://docs.python.org/3/
2. Алгоритмы обучения нейронной сети: наиболее распространенные варианты. [Электронный ресурс] // gb. URL: https://gb.ru/blog/algoritmy-obucheniya-nejronnoj-seti/