МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

Кафедра «Инфокогнитивные технологии»

Практические и лабораторные занятия по дисциплине

«Проектирование интеллектуальных систем»

Лабораторная работа № 2

**«Распознавание изображений с помощью персептрона»**

Группа 224-321

Студент Пахомов Денис Владимирович

Преподаватель Кружалов Алексей Сергеевич

Москва 2023

**Краткое описание**

Разработка программы, которая обучает искусственную нейронную сеть (персептрон) распознавать два или более черно-белых изображения.

**Цель работы**

Изучить принципы работы и алгоритм обучения простейших искусственных нейронных сетей (НС).

**Порядок выполнения работы**

* Изучить теоретическое введение.
* Сформировать обучающую выборку из 10+ изображений.
* Разработать компьютерную программу (среда разработки выбирается студентом самостоятельно).
* Провести серию из 5+ испытаний с различными исходными данными, выявить ограничения и недостатки однослойных НС для решения задач распознавания.
* Оформить отчет по лабораторной работе.

**Требования к функциональности компьютерной программы**

* В программе должна быть реализована возможность задания обучающей выборки из внешних файлов изображений.
* Изображения должны быть черно-белыми (bitmap) и размером не менее 9 (3x3) пикселей.
* Программа должна иметь два режима работы: обучения и распознавания.
* Обучение должно производиться по стандартному алгоритму обучения персептрона с использованием дельта-правила.
* В программе должны задаваться следующие настройки:
  + количество входов нейрона, которое соответствует общему числу пикселей изображения,
  + коэффициент скорости обучения (если его значение постоянно),
  + правильные варианты элементов обучающей выборки,
  + размер ошибки, при котором обучение персептрона завершается (опционально).
* На экранной форме режима обучения должны отображаться:
  + элементы обучающей выборки (изображения),
  + настройки алгоритма обучения,
  + текущие (итоговые) веса нейронов и значение порога активационной функции,
  + протоколы результатов обучения (значения весов для каждой итерации).
* На экранной форме режима распознавания должны отображаться:
  + распознаваемое изображение (должно выбираться из всего множества),
  + результат распознавания,
  + веса нейронов и значение порога активационной функции,
  + значения выходов всех нейронов до и после применения активационной функции.

**Описание выбранной задачи:**

Разработка осуществлялась на языке C# WinForms приложения. Форма приложения представлена на рисунке 1.

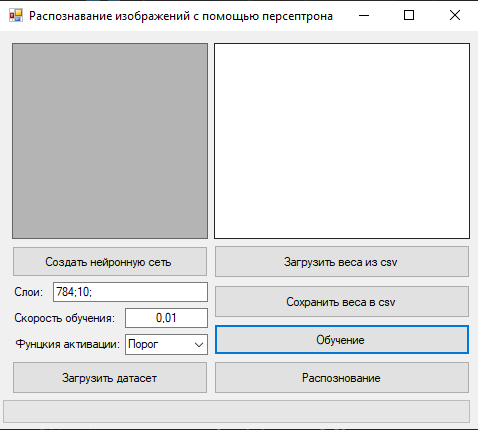


Рисунок 1 – Форма приложения

**Блок-схемы:**

* 1. Блок-схема работы функции forward

**Изображение выглядит как диаграмма

Автоматически созданное описание**

* 1. Блок-схема алгоритма обучения

Изображение выглядит как диаграмма

Автоматически созданное описание

**Эксперименты:**

**Общие параметры:**

*Входное изображение:* 28х28

*Выходные нейроны*: 10

**Параметры**:

*Функция активации*: ReLU

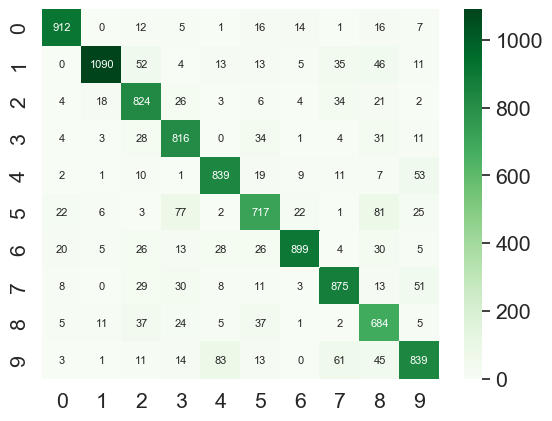
*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.001*

*Accuracy: 84.95%*

Результаты работы алгоритма представлены на рисунке 1.



Изображение выглядит как календарь

Автоматически созданное описание

Рисунок 1 – Эксперимент при функции активации ReLU

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.001*

Accuracy: 82.52%

Результаты работы алгоритма представлены на рисунке 2.

Изображение выглядит как календарь

Автоматически созданное описание

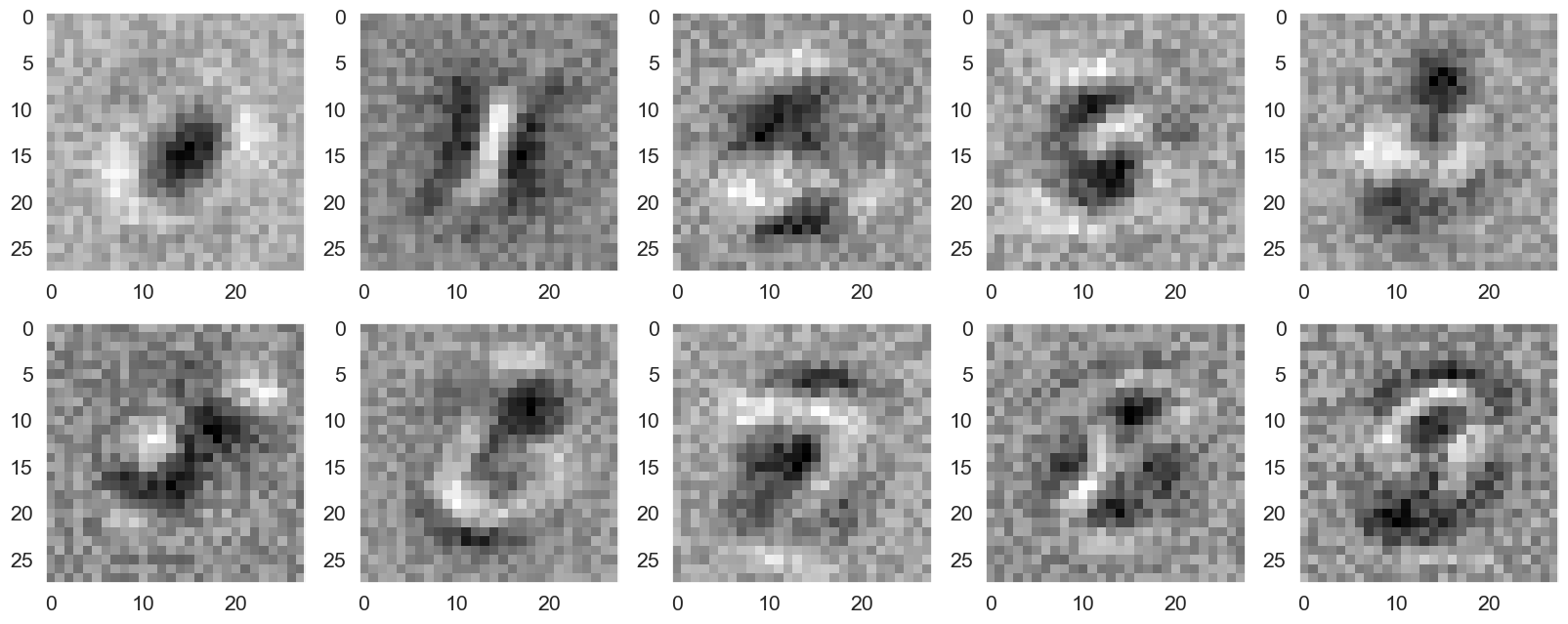


Рисунок 2 – Эксперимент при функции активации Sigmoid

*Функция активации*: dReLU

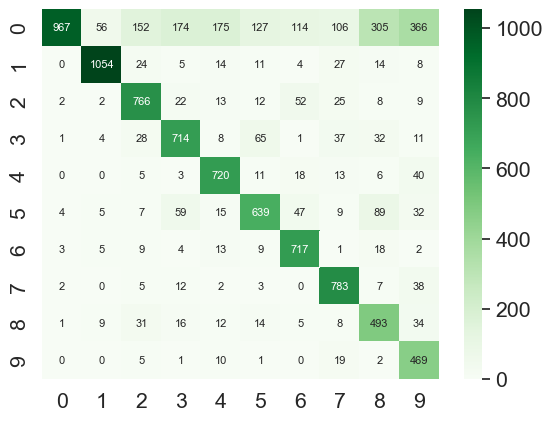
*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.001*

Accuracy: 73.22%

Результаты работы алгоритма представлены на рисунке 3.



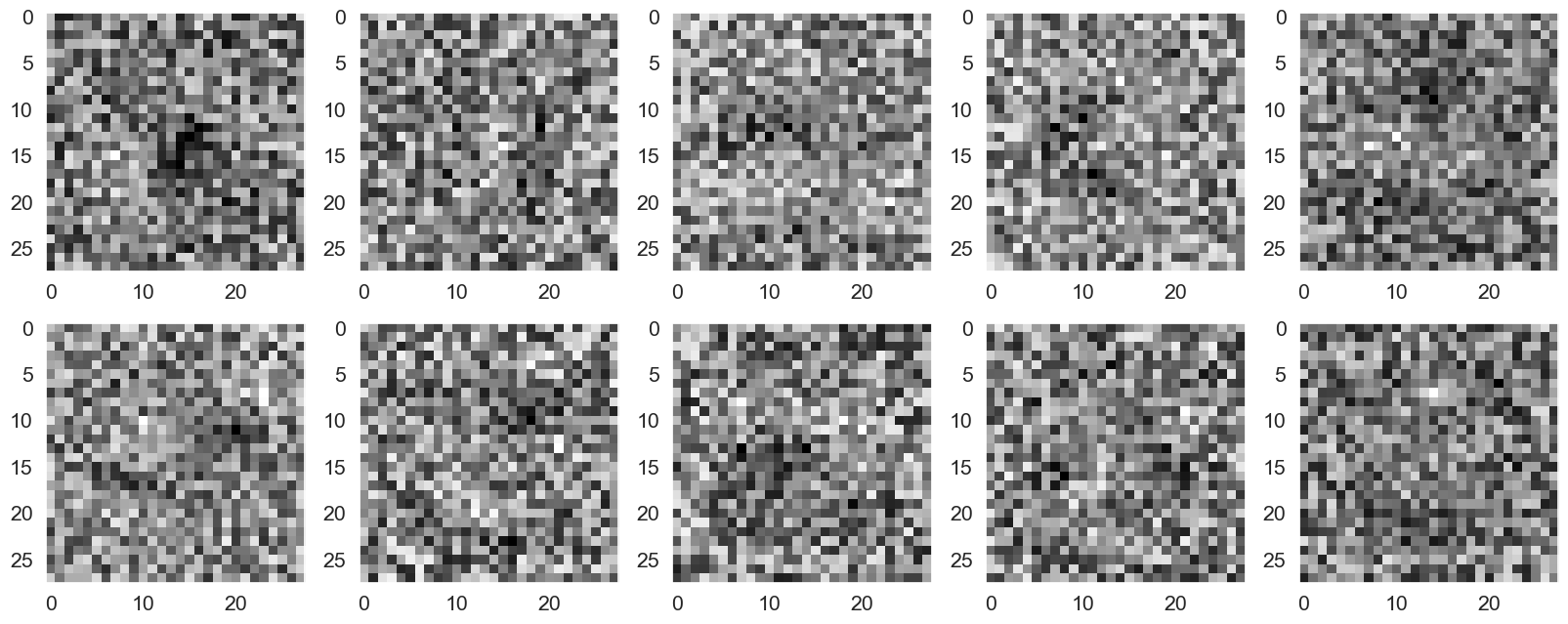


Рисунок 3 – Эксперимент при функции активации dReLU

*Функция активации*: ReLU

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.01*

Accuracy: 83.21%

Результаты работы алгоритма представлены на рисунке 4.

Изображение выглядит как календарь

Автоматически созданное описание

Изображение выглядит как календарь

Автоматически созданное описание

Рисунок 4 – Эксперимент при скорости обучения 0.01

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.01*

Accuracy: 87.15%

Результаты работы алгоритма представлены на рисунке 5.

Изображение выглядит как календарь

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Эксперимент при скорости обучения 0.01

*Функция активации*: dReLU

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.01*

Accuracy: 73.67%

Результаты работы алгоритма представлены на рисунке 6.

Изображение выглядит как календарь

Автоматически созданное описание

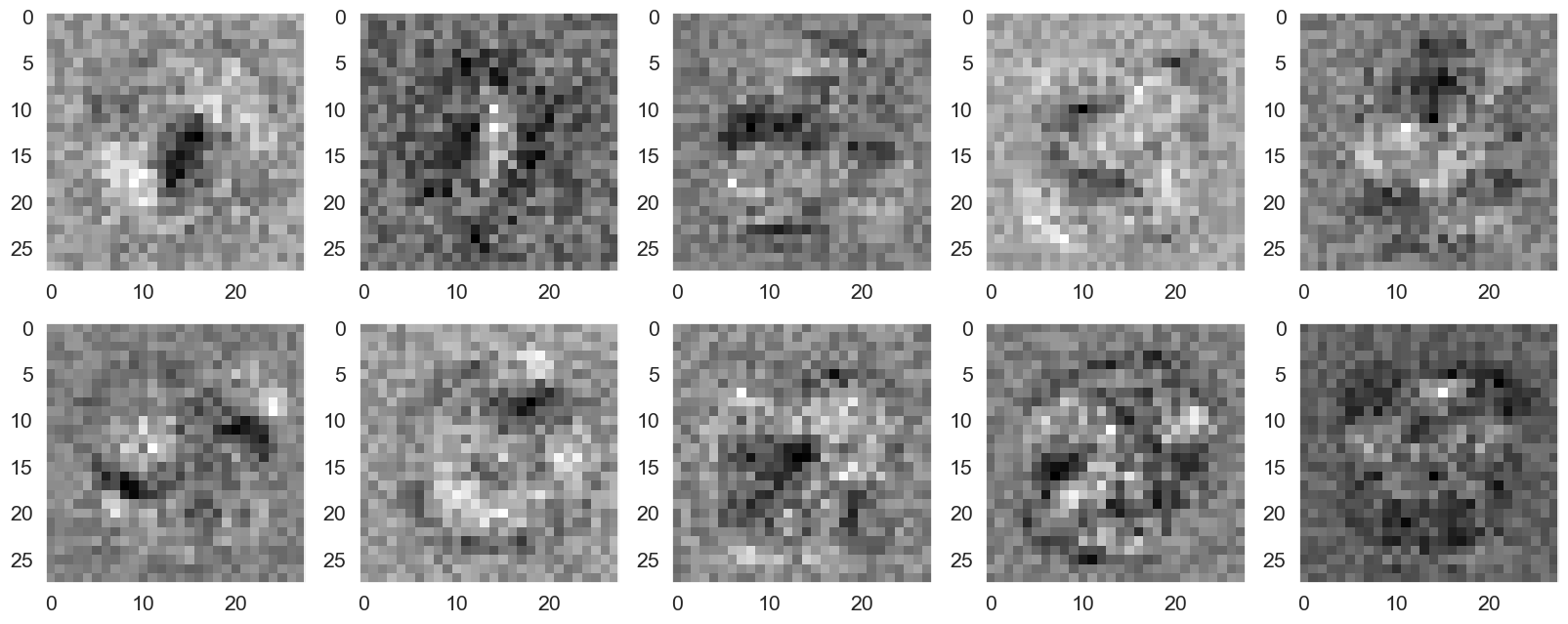


Рисунок 6 – Эксперимент при скорости обучения 0.01

*Функция активации*: ReLU

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.1*

Accuracy: 17.49%

Результаты работы алгоритма представлены на рисунке 7.

Изображение выглядит как календарь

Автоматически созданное описание

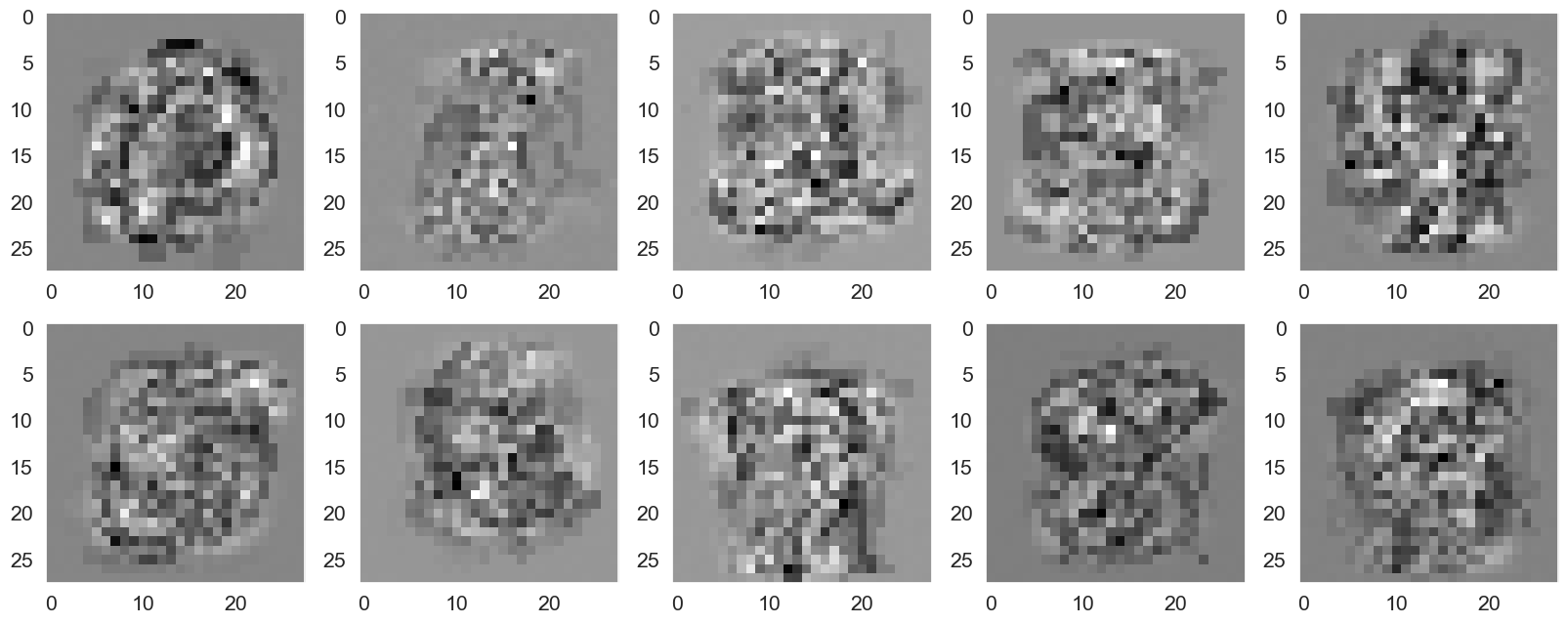
****

Рисунок 7 – Эксперимент при скорости обучения 0.1

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.1*

Accuracy: 84.91%

Результаты работы алгоритма представлены на рисунке 8.

**Изображение выглядит как календарь

Автоматически созданное описание**

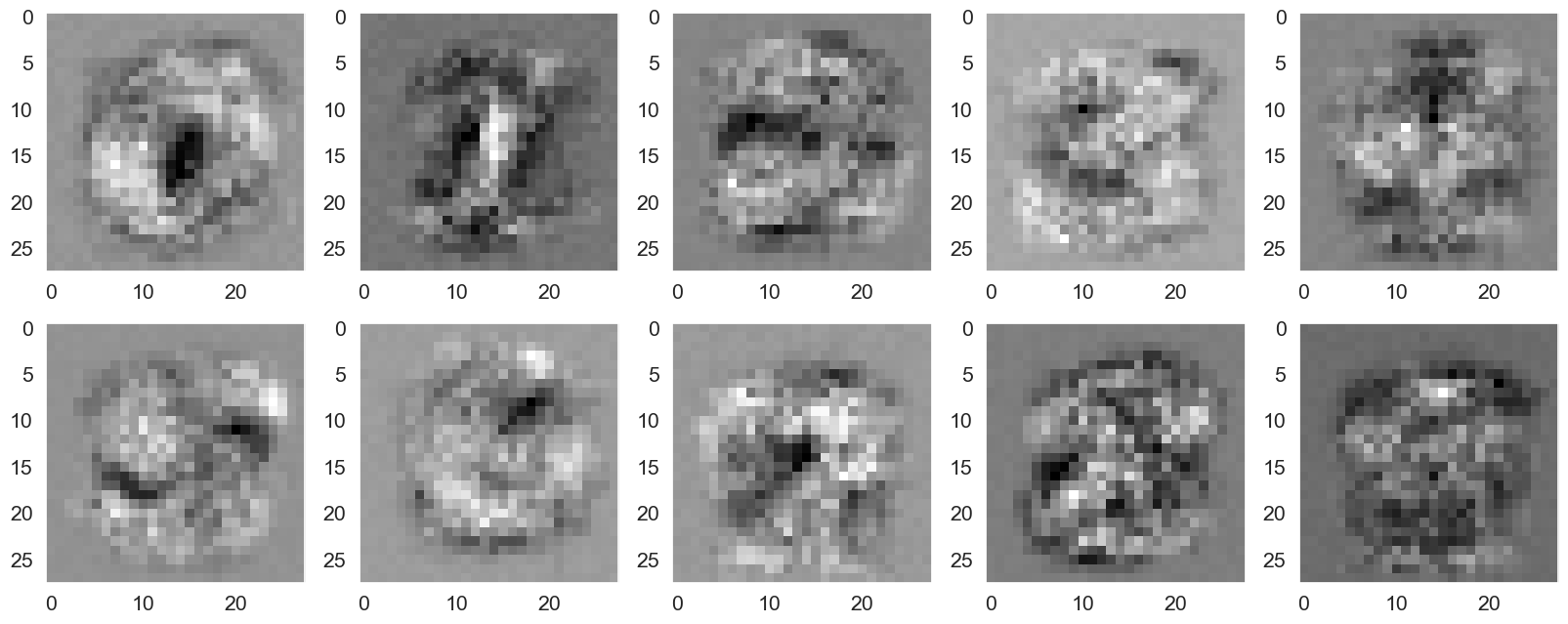
****

Рисунок 8 – Эксперимент при скорости обучения 0.1

*Функция активации*: dReLU

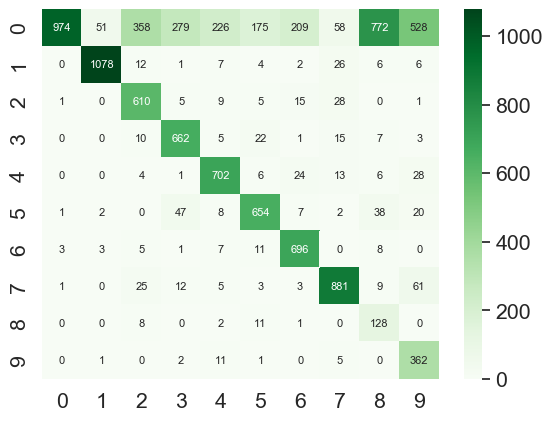
*Кол-во обучающих данных: 5000*

*Кол-во эпох: 1*

*Скорость обучения: 0.1*

Accuracy: 67.47%

Результаты работы алгоритма представлены на рисунке 9.



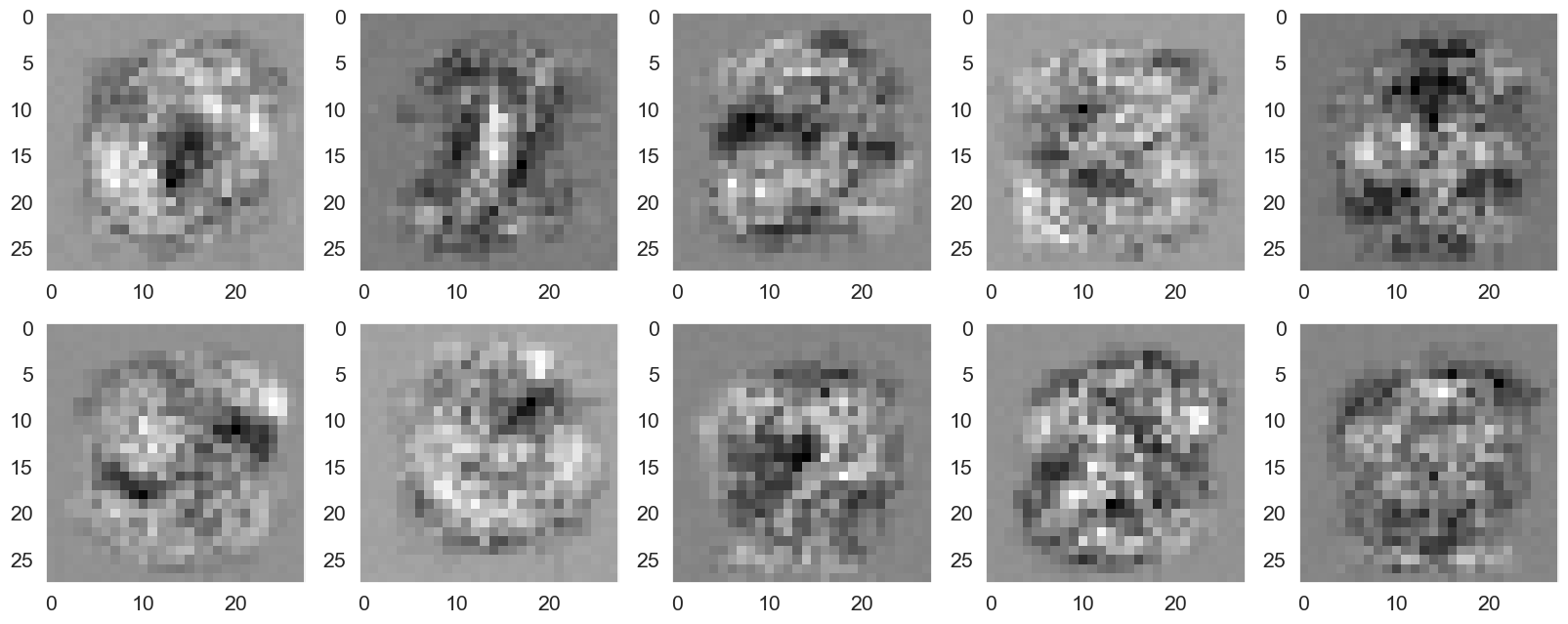


Рисунок 9 – Эксперимент при скорости обучения 0.1

Таблица 1 -Результаты экспериментов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Активация | Кол-во эпох | Кол-во данных | Скорость обучения | Accuracy |
| 1 | ReLU | 1 | 5000 | 0.001 | *84.95%* |
| 2 | Sigmoid | 1 | 5000 | 0.001 | 82.52% |
| 3 | dReLU | 1 | 5000 | 0.001 | 73.22% |
| 4 | ReLU | 1 | 5000 | 0.01 | 83.21% |
| 5 | Sigmoid | 1 | 5000 | 0.01 | 87.15% |
| 6 | dReLU | 1 | 5000 | 0.01 | 73.67% |
| 7 | ReLU | 1 | 5000 | 0.1 | 17.49% |
| 8 | Sigmoid | 1 | 5000 | 0.1 | 84.91% |
| 9 | dReLU | 1 | 5000 | 0.1 | 67.47% |

**Вывод:**

Однослойный перцептрон подходит для задач мультиклассовой классификации, при маленькой скорости обучения и подходящей функции активации, значение точности предсказаний может приближаться к 90%, выборка была ограничена до 5000 изображений и 1 эпохи обучения.

Наибольший результат показывает перцептрон с sigmoid функцией активации и скоростью обучения 0.01, а наименьший результат показывает перцептрон с ReLU функцией активации и скоростью обучения 0.1.

**Программный код представлен на GitHub:**

*https://github.com/GongniR/DoIS/blob/main/LW\_2*