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 № 3

**«Распознавание изображений на базе НС обратного распространения»**

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 Python.

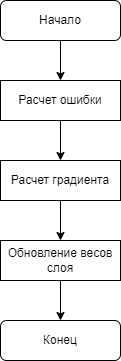
**Блок-схемы:**

* 1. Блок-схема работы функции forward

**Изображение выглядит как диаграмма

Автоматически созданное описание**

* 1. Блок-схема алгоритма обучения



* 1. Блок-схема алгоритма обучения

Изображение выглядит как диаграмма

Автоматически созданное описание

**Эксперименты:**

Таблица 1 - Результаты экспериментов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Активация  Скрытых слоев | Кол-во эпох | Кол-во данных | Скорость обучения | Accuracy |
| 1 | ReLU | 6 | 60000 | 0.001 | *92.04%* |
| 2 | Sigmoid | 6 | 60000 | 0.001 | 23.31% |
| 4 | ReLU | 6 | 60000 | 0.01 | 96.89% |
| 5 | Sigmoid | 6 | 60000 | 0.01 | 91.38% |
| 7 | ReLU | 6 | 60000 | 0.1 | 92.47% |
| 8 | Sigmoid | 6 | 60000 | 0.1 | 96.71% |

**Общие параметры:**

*Входное изображение:* 28х28

*Выходные нейроны*: 100 - 50 -10

*Порог ошибки: 1\*e-10*

**Параметры**:

*Функция активации*: ReLU

*Кол-во обучающих данных:* 60000

*Кол-во эпох: 6*

*Скорость обучения: 0.001*

*Accuracy: 92.04%*

Результаты работы алгоритма представлены на рисунке 1.

Изображение выглядит как текст, снимок экрана, прямоугольный, диаграмма

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 2 – Эксперимент при функции активации ReLU-0.001

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 60000*

*Кол-во эпох: 6*

*Скорость обучения: 0.001*

Accuracy: 23.31%

Результаты работы алгоритма представлены на рисунке 2.

Изображение выглядит как текст, снимок экрана, число, Прямоугольник

Автоматически созданное описание

Изображение выглядит как линия, График, диаграмма, текст

Автоматически созданное описание

Рисунок 2 – Эксперимент при функции активации Sigmoid-0.001

*Функция активации*: ReLU

*Кол-во обучающих данных: 60000*

*Кол-во эпох: 6*

*Скорость обучения: 0.01*

Accuracy: 96.89%

Результаты работы алгоритма представлены на рисунке 3.

Изображение выглядит как текст, снимок экрана, диаграмма, прямоугольный

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, График, диаграмма

Автоматически созданное описание

Рисунок 3 – Эксперимент при функции активации ReLU-0.01

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 60000*

*Кол-во эпох: 6*

*Скорость обучения: 0.01*

Accuracy: 91.38%

Результаты работы алгоритма представлены на рисунке 4.

Изображение выглядит как текст, снимок экрана, прямоугольный, диаграмма

Автоматически созданное описание

Изображение выглядит как снимок экрана, линия, диаграмма, График

Автоматически созданное описание

Эксперимент при функции активации Sigmoid-0.01

*Функция активации*: ReLU

*Кол-во обучающих данных: 60000*

*Кол-во эпох: 6*

*Скорость обучения: 0.1*

Accuracy: 92.47%

Результаты работы алгоритма представлены на рисунке 5.

Изображение выглядит как текст, снимок экрана, прямоугольный, диаграмма

Автоматически созданное описание

Изображение выглядит как диаграмма, текст, линия, снимок экрана

Автоматически созданное описание

Эксперимент при функции активации ReLU-0.1

*Функция активации*: Sigmoid

*Кол-во обучающих данных: 60000*

*Кол-во эпох: 6*

*Скорость обучения: 0.1*

Accuracy: 96.71%

Результаты работы алгоритма представлены на рисунке 6.

Изображение выглядит как текст, снимок экрана, диаграмма, прямоугольный

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, График, диаграмма

Автоматически созданное описание

Рисунок 6 – Эксперимент при функции активации Sigmoid-0.1

**Вывод:**

Многослойный перцептрон с обратным распространением ошибки подходит для задач мультиклассовой классификации, при средней скорости обучения и подходящей функции активации для скрытых слоев, значение точности предсказаний может приближаться к 97%.

Наибольший результат показывает перцептрон с ReLU функцией активации скрытых слоев и скоростью обучения 0.01, а наименьший результат показывает перцептрон с Sigmoid функцией активации и скоростью обучения 0.001.

**Программный код представлен на GitHub:**

*https://github.com/GongniR/Mag\_2\_semester/blob/main/DoIS/LW\_3/LW\_3.ipynb*