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 № 4

**«Обучение нейронной сети с помощью генетических алгоритмов»**

Группа 224-321

Студент Пахомов Денис Владимирович

Преподаватель Кружалов Алексей Сергеевич

Москва 2023

**Краткое описание**

Разработка компьютерной программы, которая обучает искусственный нейрон распознавать изображения с помощью генетических алгоритмов.

**Цель работы**

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

**Порядок выполнения работы**

* Предварительно выполнить лабораторные работы:
  + Решение оптимизационных задач с помощью генетических алгоритмов
  + Распознавание изображений с помощью персептрона
* Сформировать обучающую выборку из множества изображений.
* Разработать компьютерную программу (среда разработки выбирается студентом самостоятельно).
* Провести серию из 5+ испытаний с различными исходными данными, выявить трудности, ограничения и недостатки обучения НС с помощью генетических алгоритмов.
* 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 1616 (4x4)(4x4) пикселей.
* Программа должна иметь два режима работы: обучения и распознавания.
* Обучение НС должно производиться с помощью генетического алгоритма.

На экранной форме режима обучения должны отображаться:

* элементы обучающей выборки (изображения),
* правильные варианты элементов обучающей выборки,
* текущие (итоговые) веса нейронов и значение порога активационной функции,
* размер ошибки, при котором обучение нейрона завершается,
* режим обучения должен иметь два варианта работы:
  + пошаговый - на экране должны отображаться все представители (хромосомы) одного поколения до и после применения каждого оператора (скрещивания, селекции, редукции и мутации).
  + циклический - на экране должны отражаться только агрегированные данные по каждому поколению и итоговый набор хромосом.

На экранной форме режима распознавания должны отображаться:

* распознаваемое изображение (должно выбираться из всего множества),
* результат распознавания,
* веса нейронов и значение порога активационной функции,
* значения выхода нейрона.

**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

Изображение выглядит как диаграмма

Автоматически созданное описание

* 1. Блок-схема ГА

Изображение выглядит как текст, диаграмма, Технический чертеж, План

Автоматически созданное описание

**Эксперименты:**

**Общие параметры:**

*Входное изображение:* 28х28

*Выходные нейроны*: 2

*Кол-во эпох: 1*

Таблица 1 -Результаты экспериментов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Кол-во  инд. | Кол-во поколений | Вероятность  мутации | Вероятность  кроссовер | Accuracy |
| 1 | 50 | 5 | 0.1 | 1 | 79.72 % |
| 2 | 50 | 5 | 0.5 | 1 | 77.25 % |
| 3 | 50 | 5 | 1 | 1 | 47.42% |
| 4 | 50 | 5 | 0.1 | 0.3 | 46.49% |
| 5 | 50 | 5 | 0.1 | 0.6 | 84.74% |
| 6 | 50 | 5 | 0.1 | 0.9 | 97.59% |
| 7 | 50 | 10 | 0.1 | 1 | 99.11% |
| 8 | 50 | 15 | 0.1 | 1 | 76.56% |
| 9 | 50 | 20 | 0.1 | 1 | 88.37% |
| 10 | 10 | 5 | 0.1 | 1 | 78.75% |
| 11 | 20 | 5 | 0.1 | 1 | 78.45% |
| 12 | 100 | 5 | 0.1 | 1 | 90.82% |

**Параметры**:

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 79.72*

Результаты работы алгоритма представлены на рисунке 1.

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание

Рисунок 1 – Эксперимент номер 1

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 0.5*

*Вероятность кроссовера: 1*

*Accuracy: 77.25*

Результаты работы алгоритма представлены на рисунке 2.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, Симметрия, прямоугольный

Автоматически созданное описание

Рисунок 2 – Эксперимент номер 2

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 1*

*Вероятность кроссовера: 1*

*Accuracy: 47.*

Результаты работы алгоритма представлены на рисунке 3.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, рукописный текст, линия

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание

Рисунок 3 – Эксперимент номер 3

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 0.3*

*Accuracy: 46.*

Результаты работы алгоритма представлены на рисунке 4.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, линия, Шрифт

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, Симметрия, прямоугольный

Автоматически созданное описание

Рисунок 4 – Эксперимент номер 4

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 0.6*

*Accuracy: 84.*

Результаты работы алгоритма представлены на рисунке 5.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, линия, Шрифт

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание

Рисунок 5 – Эксперимент номер 5

*Кол-во индивид.: 50*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 0.9*

*Accuracy: 97.59*

Результаты работы алгоритма представлены на рисунке 6.

Изображение выглядит как текст, снимок экрана, Прямоугольник, прямоугольный

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание

Рисунок 6 – Эксперимент номер 6

*Кол-во индивид.: 50*

*Кол-во поколений: 10*

*Вероятность мутации: 0.1*

*Кол-во эпох: 1*

*Accuracy: 99.11*

Результаты работы алгоритма представлены на рисунке 7.

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, линия, Шрифт, График

Автоматически созданное описание

**Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание**

Рисунок 7 – Эксперимент номер 7

*Кол-во индивид.: 50*

*Кол-во поколений: 15*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 76.56*

Результаты работы алгоритма представлены на рисунке 8.

**Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание**

**Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описание**

**Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание**

Рисунок 8 – Эксперимент номер 8

*Кол-во индивид.: 50*

*Кол-во поколений: 20*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 88.37*

Результаты работы алгоритма представлены на рисунке 9.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, линия

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, Симметрия, прямоугольный

Автоматически созданное описание

Рисунок 9 – Эксперимент номер 9

*Кол-во индивид.: 10*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 78.75*

Результаты работы алгоритма представлены на рисунке 10.

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, линия, снимок экрана, Шрифт

Автоматически созданное описание

**Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание**

Рисунок 10 – Эксперимент номер 10

*Кол-во индивид.: 20*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 78.45*

Результаты работы алгоритма представлены на рисунке 11.

**Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание**

**Изображение выглядит как текст, рукописный текст, Шрифт, линия

Автоматически созданное описание**

**Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание**

Рисунок 11 – Эксперимент номер 11

*Кол-во индивид.: 100*

*Кол-во поколений: 5*

*Вероятность мутации: 0.1*

*Вероятность кроссовера: 1*

*Accuracy: 90.82*

Результаты работы алгоритма представлены на рисунке 12.

Изображение выглядит как текст, снимок экрана, Прямоугольник, зеленый

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, линия, График

Автоматически созданное описание

Изображение выглядит как шаблон, снимок экрана, прямоугольный, Симметрия

Автоматически созданное описание

Рисунок 12 – Эксперимент номер 12

**Вывод:**

Однослойный перцептрон, обученный с помощью генетического алгоритма, подходит для задач бинарной классификации, большую долю успеха обучения с помощью ГА занимается формирование первоначальной популяции. На графиках видно, что при поступлении нового изображения функция приспособленности начинает уменьшаться, а потом снова расти.

Наибольший результат показывает перцептрон обученный с помощью ГА с 10% шансом мутации, 100 % шансом мутации, популяция состоит из 50 индивидуалов, а кол-во поколений равно 10. Обратная ситуация наблюдается при обученный с помощью ГА с 10% шансом мутации, 30 % шансом мутации, популяция состоит из 50 индивидуалов, а кол-во поколений равно 5.

Балансирую параметры ГА алгоритма, можно добиться высоких показателей для задачи классификации

**Программный код представлен на GitHub:**

*https://github.com/GongniR/Mag\_2\_semester/tree/main/DoIS/LW\_4*