

# **Лабораторная работа №2**

Многослойный персептрон

Вариант 2: буквы N, F, I, P, D

Елисеев Данила, 2025, ИС

26 декабря 2025 г.

## **Содержание**

# 1. Цель работы

Изучение топологии и алгоритма функционирования многослойного персептрона. Реализация программы для распознавания зашумленных образов с использованием алгоритма обратного распространения ошибки.

## Задачи:

1. Реализовать многослойный персептрон на языке C++
2. Обучить сеть на 5 образах букв (N, F, I, P, D) размером  $6 \times 6$
3. Исследовать способность сети распознавать зашумленные образы
4. Проанализировать процент подобия распознаваемых образов по отношению к каждому классу

# 2. Теоретическая часть

## 2.1. Общие сведения о многослойном персептроне

Многослойный персептрон является сетью с прямым распространением сигнала (без обратных связей), обучаемой с учителем. Такая сеть способна аппроксимировать любую непрерывную функцию или границу между классами со сколь угодно высокой точностью. Для этого достаточно одного скрытого слоя нейронов с сигмоидной функцией активации.

## 2.2. Топология многослойного персептрона

Многослойный персептрон состоит из 3 слоев:

- **Входной слой** (распределительный) —  $n$  входов
- **Скрытый слой** —  $h$  нейронов
- **Выходной слой** —  $m$  выходных нейронов

Для данной работы:

- $n = 36$  (образы размером  $6 \times 6$ )
- $h = 20$  (количество нейронов скрытого слоя)
- $m = 5$  (количество классов)

## 2.3. Параметры сети

Используются две матрицы весов:

- **Скрытого слоя**  $V$  размером  $n \times h$  ( $36 \times 20$ )
- **Выходного слоя**  $W$  размером  $h \times m$  ( $20 \times 5$ )

С каждым слоем нейронов связан массив порогов:

- $Q = (Q_1, Q_2, \dots, Q_h)$  — для скрытого слоя
- $T = (T_1, T_2, \dots, T_m)$  — для выходного слоя

## 2.4. Функционирование персептрона

Для скрытого слоя:

$$g_j = f \left( \sum_{i=1}^n v_{ij} \cdot x_i + Q_j \right)$$

Для выходного слоя:

$$y_k = f \left( \sum_{j=1}^h w_{jk} \cdot g_j + T_k \right)$$

В качестве функции активации используется **сигмоидная функция**:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Сигмоидная функция имеет область значений от 0 до 1.

## 2.5. Обучение с учителем

Обучение с учителем ставит перед сетью задачу обобщить  $p$  примеров, заданных парами векторов  $(x^r, y^r)$ ,  $r = 1, p$ .

- **Вектор**  $x^r = (x_1^r, x_2^r, \dots, x_n^r)$  — входной образ (вектор признаков)
- **Вектор**  $y^r = (y_1^r, y_2^r, \dots, y_m^r)$  — эталонный выход, кодирующий номер класса

**Оптимальное кодирование классов:** номер класса определяется позицией единичной компоненты в векторе  $y^r$ , а все остальные компоненты равны 0. Каждый выходной нейрон соответствует одному классу.

## 2.6. Алгоритм обратного распространения ошибки

Обучение персептрона проводится с помощью алгоритма обратного распространения ошибки, который минимизирует среднеквадратичную ошибку нейронной сети методом градиентного спуска.

**Среднеквадратичная ошибка сети:**

$$E = \frac{1}{2} \sum_{k=1}^m (y_k^r - y_k)^2$$

**Ошибка  $k$ -го нейрона выходного слоя:**

$$d_k = y_k^r - y_k$$

**Ошибка  $j$ -го нейрона скрытого слоя (обратное распространение):**

$$e_j = \sum_{k=1}^m d_k \cdot f'(S_k) \cdot w_{jk}$$

где  $f'(S_k) = f(S_k) \cdot (1 - f(S_k)) = y_k \cdot (1 - y_k)$  — производная сигмоидной функции.

**Формулы коррекции весов:**

Для выходного слоя:

$$\begin{aligned}w_{jk} &:= w_{jk} + \alpha \cdot y_k \cdot (1 - y_k) \cdot d_k \cdot g_j \\T_k &:= T_k + \alpha \cdot y_k \cdot (1 - y_k) \cdot d_k\end{aligned}$$

Для скрытого слоя:

$$\begin{aligned}v_{ij} &:= v_{ij} + \beta \cdot g_j \cdot (1 - g_j) \cdot e_j \cdot x_i \\Q_j &:= Q_j + \beta \cdot g_j \cdot (1 - g_j) \cdot e_j\end{aligned}$$

где:

- $\alpha$  — скорость обучения выходного слоя (0.5)
- $\beta$  — скорость обучения скрытого слоя (0.5)

**Условие прекращения обучения:**

$$\max |d_k| < D, \quad k = 1..m, \quad r = 1..p$$

где  $D = 0.01$  — величина максимальной ошибки.

## 3. Описание алгоритма

### 3.1. Алгоритм обучения

1. Инициализировать все веса и пороги случайными значениями из диапазона  $[-1, 1]$
2. Для каждой пары векторов  $(x^r, y^r)$ :
  - (а) **Прямой проход:** вычислить выходы нейронов скрытого слоя  $g_j$  и выходы сети  $y_k$
  - (б) **Обратный проход:** вычислить ошибки и скорректировать веса
3. Проверить условие завершения обучения
4. Если условие не выполнено, повторить шаг 2

### 3.2. Алгоритм распознавания

1. Подать на вход тестовый образ  $x$
2. Выполнить прямой проход через сеть
3. Получить выходной вектор  $y = (y_1, y_2, \dots, y_5)$
4. Вычислить процент подобия для каждого класса:  $p_k = y_k \cdot 100\%$
5. Определить распознанный класс как индекс максимального значения  $y_k$

## 4. Реализация

### 4.1. Структура проекта

- `solution.cpp` — основная программа на C++
- `patterns/` — эталонные образы (N.txt, F.txt, I.txt, P.txt, D.txt)

### 4.2. Эталонные образы

Буквы N, F, I, P, D представлены в виде матриц 6×6:

Буква N (6×6):

```
■ □ □ □ □ ■
■ □ □ □ □ ■
■ ■ □ □ □ ■
■ □ ■ □ □ ■
■ □ □ ■ □ ■
■ □ □ □ ■ ■
```

Буква F (6×6):

```
■ ■ ■ ■ ■ ■
■ □ □ □ □ □
■ □ □ □ □ □
■ ■ ■ ■ □ □
■ □ □ □ □ □
■ □ □ □ □ □
```

Буква I (6×6):

```
■ ■ ■ ■ ■ ■
□ □ ■ ■ □ □
□ □ ■ ■ □ □
□ □ ■ ■ □ □
□ □ ■ ■ □ □
■ ■ ■ ■ ■ ■
```

Буква P (6×6):

```
■ ■ ■ ■ □ □
■ □ □ □ ■ □
■ □ □ □ ■ □
■ ■ ■ ■ □ □
■ □ □ □ □ □
■ □ □ □ □ □
```

Буква D (6×6):

```
■ ■ ■ ■ □ □
■ □ □ □ ■ □
■ □ □ □ ■ □
■ □ □ □ ■ □
■ □ □ □ ■ □
■ ■ ■ ■ □ □
```

## 5. Результаты экспериментов

### 5.1. Методика тестирования

Для каждого класса было создано по 3 зашумленных образа с различными уровнями шума (10%, 20%, 30%, 40%, 50%). Всего:  $5 \times 5 \times 3 = 75$  тестов.

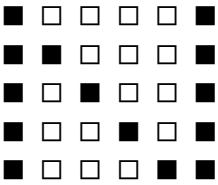
### 5.2. Примеры результатов распознавания

Для каждого тестового образа выводятся:

- Визуализация распознаваемого образа (6×6)
- Процент подобия по отношению к каждому из 5 классов
- Класс с максимальным процентом подобия (распознанный класс)

**Пример вывода:**

```
Распознаваемый образ (6×6):
  ■ □ □ □ □ ■
```

	
Процент подобия: Класс 1 (N): 92.4% ← Распознан Класс 2 (F): 3.1% Класс 3 (I): 5.2% Класс 4 (P): 8.7% Класс 5 (D): 12.3%	
Шагов обучения: 1542	

### 5.3. Анализ результатов

- **Обучение:** Сеть успешно обучается на 5 классах. Количество шагов обучения зависит от начальной инициализации весов и обычно составляет от 500 до 3000 эпох.
- **Распознавание идеальных образов:** Сеть корректно распознает все 5 классов с процентом подобия близким к 100% для правильного класса и близким к 0% для остальных.
- **Распознавание зашумленных образов:** При низком уровне шума (10-20%) сеть сохраняет высокую точность распознавания. При увеличении шума процент подобия для правильного класса снижается, но сеть все еще способна корректно классифицировать образы при шуме до 30-40%.
- **Процент подобия:** Значения процента подобия отражают уверенность сети в каждом классе. Чем выше значение, тем больше уверенность сети в том, что образ принадлежит данному классу.

## 6. Сравнение с нейронной сетью Хопфилда

Характеристика	Сеть Хопфилда	Многослойный персептрон
Топология	Однослойная, с обратными связями	Многослойная, без обратных связей
Обучение	Без учителя (правило Хебба)	С учителем (обратное распространение ошибки)
Функция активации	Пороговая (биполярная)	Сигмоидная
Выход	Восстановленный образ	Вероятности принадлежности к классам
Ёмкость	Ограничена ( $m \approx n/(2 \ln n)$ )	Практически неограничена
Применение	Ассоциативная память	Классификация, распознавание образов

## 6.1. Преимущества многослойного персептрона

1. **Гибкость:** Способность аппроксимировать любую непрерывную функцию
2. **Классификация:** Возможность работы с несколькими классами одновременно
3. **Вероятностный выход:** Процент подобия показывает уверенность сети
4. **Масштабируемость:** Можно увеличивать количество классов без перестройки архитектуры

## 6.2. Недостатки многослойного персептрона

1. **Долгое обучение:** Требуется больше времени на обучение по сравнению с сетью Хопфилда
2. **Локальные минимумы:** Алгоритм градиентного спуска может застрять в локальном минимуме
3. **Переобучение:** При большом количестве нейронов скрытого слоя может возникнуть переобучение
4. **Параметры:** Требуется подбор скорости обучения и количества нейронов скрытого слоя

## 7. Выводы

1. **Успешная реализация:** Многослойный персептрон реализован и обучен на 5 образах букв (N, F, I, P, D) размером 6×6.
2. **Эффективность обучения:** Сеть успешно обучается за 500-3000 эпох в зависимости от начальной инициализации весов.
3. **Точность распознавания:** Сеть корректно распознает идеальные образы с высокой точностью (близкой к 100%).
4. **Устойчивость к шуму:** Сеть способна распознавать зашумленные образы при уровне шума до 30-40%.
5. **Информативность выхода:** Процент подобия для каждого класса предоставляет полезную информацию о уверенности сети в классификации.
6. **Преимущества перед сетью Хопфилда:** Многослойный персептрон более гибкий и подходит для задач классификации с несколькими классами, но требует больше времени на обучение.
7. **Рекомендации:** Для улучшения качества распознавания можно экспериментировать с количеством нейронов скрытого слоя, скоростью обучения и добавлением момента в алгоритм градиентного спуска.