

## **Содержание**

Общие теоретические сведения.....	4
1. Лабораторная работа №1. Нейронная сеть Хопфилда.....	10
2. Лабораторная работа №2. Многослойный персептрон.....	15
3. Лабораторная работа №3. Сеть РБФ.....	23
4. Лабораторная работа №4. Конкурентная нейронная сеть.....	27
Литература.....	31

# НЕЙРОННЫЕ СЕТИ

## Общие теоретические сведения

Приведем одно из определений искусственных нейронных сетей [1]:

**Искусственная нейронная сеть (ИНС)** – это существенно параллельно распределенный процессор, который обладает способностью к сохранению и презентации опытного знания. Она сходна с мозгом в двух аспектах:

- знание приобретается сетью в процессе обучения;
- для сохранения знания используются силы межнейронных соединений, называемые также синаптическими соединениями.

Работа нейронной сети (НС) состоит в преобразовании входных сигналов во времени, в результате чего меняется внутреннее состояние сети и формируются выходные воздействия. Обычно НС оперирует цифровыми, а не символьными величинами. Большинство моделей НС требуют обучения. В общем случае, **обучение** — это такой выбор параметров сети, при котором сеть лучше всего справляется с поставленной проблемой. Обучение — это задача многомерной оптимизации и для ее решения существует множество алгоритмов.

Современные искусственные НС демонстрируют такие ценные свойства, как:

1. *Обучаемость*. Выбрав одну из моделей НС, создав сеть и выполнив алгоритм обучения, мы можем обучить сеть решению задачи, которая ей по силам.

2. *Способность к обобщению*. После обучения сеть становится нечувствительной к малым изменениям входных сигналов (шуму или вариациям входных образов) и дает правильный результат на выходе.

3. *Способность к абстрагированию*. Если предъявить сети несколько искаженных вариантов входного образа, то сеть сама может создать на выходе идеальный образ, с которым она никогда не встречалась.

К задачам, успешно решаемым НС на данном этапе их развития относятся:

- распознавание зрительных, слуховых образов;
- ассоциативный поиск информации и создание ассоциативных моделей, синтез речи, формирование естественного языка;
- формирование моделей и различных нелинейных и трудно описываемых математически систем, прогнозирование развития этих систем во времени;
- системы управления и регулирования с предсказанием;
- разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключающие логический вывод, особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере.

На рис. 1.1 показана структура пары типичных биологических нейронов. Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посыпая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

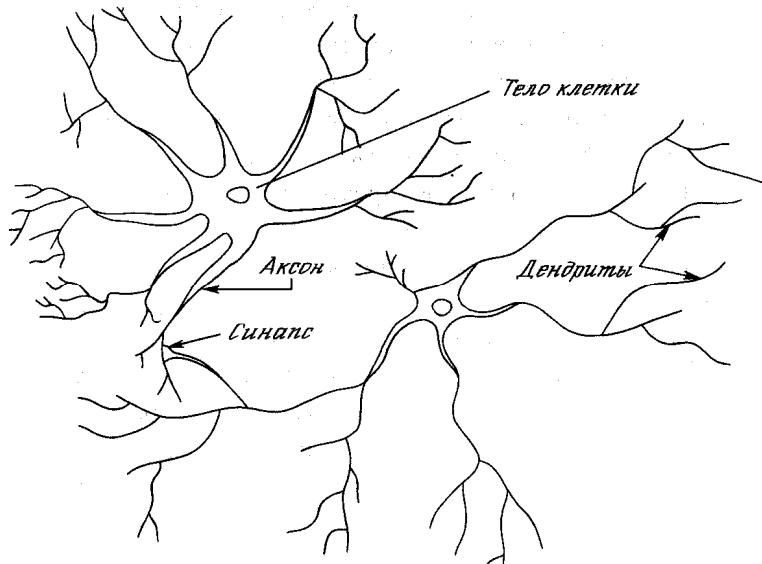


Рис. 1.1. Биологический нейрон

Основным элементом нейронной сети является **нейрон**, который осуществляет операцию нелинейного преобразования суммы произведений входных сигналов на весовые коэффициенты:

$$y = f\left(\sum_{i=1}^n w_i x_i + T\right), \quad (1.1)$$

где  $X = (x_1, x_2, \dots, x_n)$  – вектор входного сигнала;  $W = (w_1, w_2, \dots, w_n)$  – весовой вектор;  $T$  - порог;  $f$  – так называемая функция активации.

Схема нейронного элемента изображена на рис 1.2. Каждому  $i$ -му входу нейрона соответствует весовой коэффициент  $w_i$  (синапс), который характеризует силу синаптической связи по аналогии с биологическим нейроном.

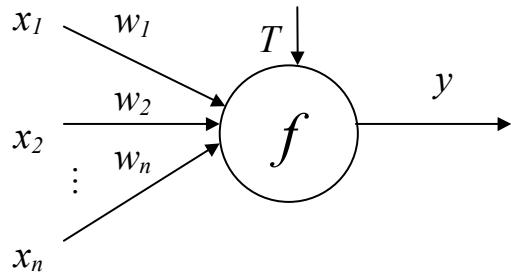


Рис. 1.2. Искусственный нейрон

Весовой вектор  $W$ , порог  $T$  и пороговая функция  $f$  определяют поведение любого нейрона – его реакцию на входные данные. Величина веса  $w_i$  определяет степень влияния входа  $i$  на выход нейрона, а знак – характер влияния. Каждый из входов связан с некоторым источником информации (рецептор, формирующий признаки, распределительная ячейка, выход другого нейрона). Положительные веса характерны для возбуждающих связей, способствующих повышению активности нейрона. Отрицательные, как правило, соответствуют тормозящим связям.

**Порог**, если он используется, является характеристикой, задающей начальный уровень активности (при нулевом входе) и помогающей настроить нейрон на пороговую функцию. Изменение порога эквивалентно сдвигу пороговой функции по оси абсцисс. Ряд авторов [2] вводят дополнительный вход нейрона  $x_0$ , всегда равный 1, и обозначают порог как его вес  $w_0$ . Это позволяет упростить выражение (1.1) и математическую запись некоторых алгоритмов обучения. Однако на практике при программной реализации это не приводит к экономии времени и способствует ошибкам, кроме этого, порог может настраиваться иначе, чем весовой вектор.

**Функция активации** используется для ограничения выхода нейрона в заданном диапазоне и для нелинейного преобразования взвешенной суммы. Последнее позволяет нейронному классификатору аппроксимировать любую нелинейную границу между классами в пространстве образов. Функция активации выбирается для конкретной задачи и является неизменной характеристикой отдельного нейрона.

Могут использоваться следующие функции активации и их гибриды:

1) линейная функция  $y = Ax$ ;

2) пороговая функция  $y = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$ ;

3) биполярная пороговая функция  $y = \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases}$ ;

4) сигмоидная функция (рис. 1.2)

$$y = \frac{1}{(1 + e^{-x})}; \quad (1.2)$$

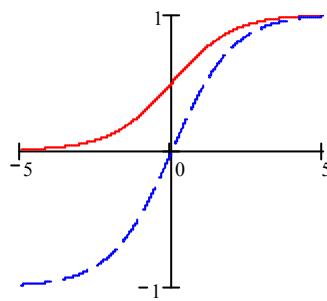


Рис. 1.2. Сигмоидные функции активации

5) биполярная сигмоидная функция (см. рис. 1.2)

$$y = \frac{2}{(1 + e^{-x})} - 1; \quad (1.3)$$

6) гиперболический тангенс (рис. 1.3)

$$y = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (1.4)$$

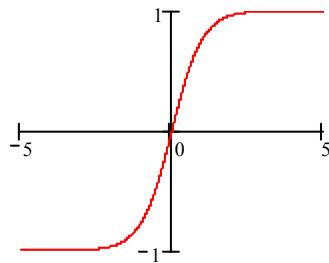


Рис. 1.3. Гиперболический тангенс

Для классификаторов чаще всего применяют функции (1.2) – (1.4), поскольку они нелинейные и хорошо дифференцируются.

Веса и порог конкретного нейрона являются настраиваемыми параметрами. Они содержат знания нейрона, определяющие его поведение. Процесс настройки этих знаний с целью получения нужного поведения называется обучением.

Нейронная сеть – совокупность нейронных элементов и связей между ними. Слоем нейронной сети называется множество нейронных элементов, на которые в каждый такт времени параллельно поступает информация от других нейронных элементов сети. Помимо слоев нейронов часто используется поня-

тие входного распределительного слоя. Распределительный слой передает входные сигналы на первый обрабатывающий слой нейронных элементов (рис. 1.4).

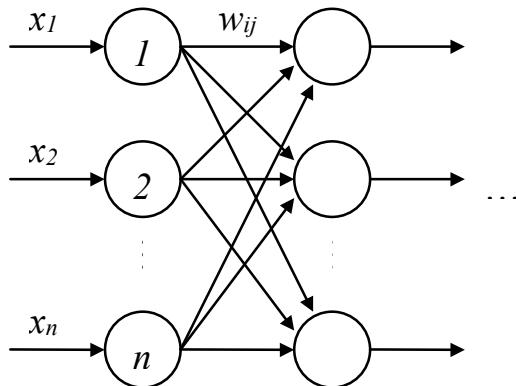


Рис. 1.4. Распределительный слой

У любой нейронной сети можно выделить 2 режима работы: обучение и воспроизведение. На этапе обучения настраиваются веса и пороги всех слоев. Воспроизведение является этапом обработки информации, следующим за обучением, при этом веса и пороги, как правило, не изменяются. Большое влияние на функционирование сети оказывает ее топология – архитектура слоев и связей между нейронами.

На способе обработки информации решающим образом сказывается наличие или отсутствие в сети обратных связей. Если обратных связей нет (каждый нейрон получает информацию только от нейронов предыдущих слоев), то обработка информации происходит в одном направлении за число тактов, равное числу слоев. При наличии обратных связей информация может проходить через сеть много раз до достижения какого-либо условия. В случае, если условие достигнуто, говорят, что сеть стабилизировалась. В общем случае сходимость не гарантируется. Тем не менее наличие обратных связей позволяет решать задачи с привлечением меньшего числа нейронов, что ускоряет процесс обучения.

Для лучшего понимания данного вопроса приведем одну из возможных классификаций НС в зависимости от различных характеристик [2]:

1. По типу входной информации:

- аналоговые НС (используют информацию в форме действительных чисел);
- двоичные НС (оперируют с информацией, представленной в двоичном виде);

2. По характеру обучения:

- с учителем (известно входное пространство решений НС);

- без учителя (НС формирует выходное пространство решений только на основе входных воздействий – самоорганизующиеся сети);

3. По характеру настройки синапсов:

- сети с фиксированными связями (весовые коэффициенты НС выбираются сразу, исходя из условия задачи);
- сети с динамическими связями (в процессе обучения происходит настройка синаптических связей);

4. По методу обучения

- НС с алгоритмом обратного распространения ошибки;
- НС с конкурентным обучением;
- НС, использующие правило Хебба;
- НС с гибридным обучением, в которых используются различные алгоритмы обучения;

5. По характеру связей:

- НС с прямыми связями;
- НС с обратным распространением информации;

6. По архитектуре и обучению:

- персептронные сети с прямыми связями;
- самоорганизующиеся НС (НС Кохонена, НС адаптивного резонанса, рециркуляционные НС);
- НС с обратными связями (НС Хопфилда, НС Хэмминга, двунаправленная ассоциативная память, рекуррентные НС);
- гибридные НС (НС встречного распространения, НС с радиально-базисной функцией активации).

# Лабораторная работа №1

## Нейронная сеть Хопфилда

### 1. Цель работы

Изучение топологии, алгоритма функционирования сети Хопфилда.

### 2. Теоретические сведения

Сеть Хопфилда – однослойная, симметричная, нелинейная, автоассоциативная нейронная сеть, которая запоминает бинарные / биполярные образы. Сеть характеризуется наличием обратных связей. Топология сети Хопфилда показана на рис. 2.1. Информация с выхода каждого нейрона поступает на вход всех остальных нейронов. Образы для данной модификации сети Хопфилда кодируются биполярным вектором, состоящим из 1 и -1.

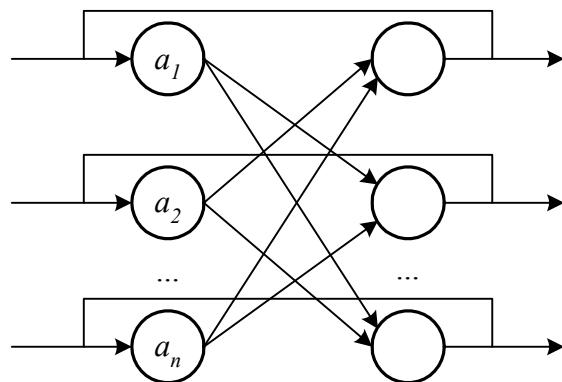


Рис. 2.1. Топология сети Хопфилда

**Обучение.** Обучение сети осуществляется в соответствии с соотношением

$$w_{ij} = \begin{cases} \sum_{k=1}^m a_i^k a_j^k, & i \neq j, \text{ для } i, j = \overline{1, n}, \\ 0, & i = j \end{cases} \quad (2.1)$$

где  $w_{ij}$  – вес связи от  $i$ -го нейрона к  $j$ -му;

$n$  – количество нейронов в сети;

$m$  – количество образов, используемых для обучения сети;

$a_i^k$  –  $i$ -й элемент  $k$ -го образа из обучающей выборки.

Матрица весовых коэффициентов

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}. \quad (2.2)$$

В качестве матрицы весовых коэффициентов Хопфилд использовал симметричную матрицу ( $w_{ij}=w_{ji}$ ) с нулевой главной диагональю ( $w_{ii}=0$ ). Последнее условие соответствует отсутствию обратной связи нейронного элемента на себя. В качестве функции активации нейронных элементов может использоваться как пороговая, так и непрерывная функция, например сигмоидная или гиперболический тангенс.

Будем рассматривать нейронную сеть Хопфилда с дискретным временем. Тогда при использовании пороговой функции активации она называется нейронной сетью с дискретным состоянием и временем. Нейронная сеть с непрерывной функцией активации называется нейронной сетью с непрерывным состоянием и дискретным временем. При использовании непрерывного времени модель Хопфилда называется непрерывной.

Для описания функционирования таких сетей Хопфилд использовал аппарат статистической физики. При этом каждый нейрон имеет два состояния активности ( $1, -1$ ), которые аналогичны значениям спина некоторой частицы. Весовой коэффициент  $w_{ji}$  можно интерпретировать как вклад поля  $j$  – частицы в величину потенциала  $i$  – частицы. Хопфилд показал, что поведение такой сети аналогично поведению лизингового спинового стекла. При этом он ввел понятие вычислительной энергии, которую можно интерпретировать в виде ландшафта с долинами и впадинами. Структура соединений сети определяет очертания ландшафта. Нейронная сеть выполняет вычисления, следя по пути, уменьшающему вычислительную энергию сети. Это происходит до тех пор, пока путь не приведет на дно впадины. Данный процесс аналогичен скатыванию капли жидкости по склону, когда она минимизирует свою потенциальную энергию в поле тяготения. Впадины и долины в сети Хопфилда соответствуют наборам информации, которую хранит сеть. Если процесс начинается с приближенной или неполной информации, то он следует по пути, который ведет к ближайшей впадине. Это соответствует операции ассоциативного распознавания.

Матрица весов является диагонально симметричной, причем все диагональные элементы равны 0.

**Воспроизведение.** Нейронная сеть Хопфилда может функционировать синхронно и асинхронно. Для воспроизведения используется соотношение

$$a_i(t+1) = f\left(\sum_{j=1}^n w_{ij} a_j(t)\right), \quad (2.3)$$

где  $a_j(t)$  – выход  $j$ -го нейрона в момент времени  $t$ , а  $f$  – бинарная / биполярная функция активации;

$$f(x) = \begin{cases} 1 & x > 0, \\ -1 & x \leq 0. \end{cases} \quad (2.4)$$

При работе в синхронном режиме на один такт работы сети все нейроны одновременно меняют состояние по формуле (2.4). В случае асинхронной работы состояние меняет только один случайный нейрон. Итерации продолжаются до тех пор, пока сеть не придет в стабильное состояние.

Во время воспроизведения исходным вектором  $a(0)$  является некоторый тестовый образ, не совпадающий с образами из обучающей выборки. В процессе функционирования по формуле (2.4) сеть должна прийти в состояние, соответствующее образу из обучающей выборки, наиболее похожему на тестовый.

Максимальное количество образов, которое можно запомнить в матрице  $W$  не превышает

$$m = \frac{n}{2 \ln n + \ln \ln n}, \quad (2.5)$$

где  $n$  – количество нейронов, что следует отнести к недостаткам этой сети.

### 3. Задание

1. Ознакомьтесь с теоретической частью.
2. На языке С, С++ напишите программу, реализующую нейронную сеть Хопфилда.
3. Произведите обучение сети Хопфилда на заданный тип образов. Для запоминания в соответствии с вариантом задано 3 образа (бинарные изображения размером  $10 \times 10$ ).
4. Подайте на вход сети ряд тестовых образов, в которые внесено зашумление (процент зашумления образа – 10%, 20%, 30%, 35%, 40%, 45%, 50%, 60%, 70%, 80%, 90%, 100%). Тестовых образов должно быть не менее 10 для каждого из классов с одним и тем же процентом зашумления.
5. Проанализируйте результаты, при каком проценте зашумления тестовые образы распознаются верно.
6. Напишите отчет.

## Содержание отчета:

- топология сети Хопфилда;
- описание алгоритма работы сети;
- тестируемые образы (3 образа);
- искаженные образы (процент зашумления образа – 10%, 20%, 30%, 35%, 40%, 45%, 50%, 60%, 70%, 80%, 90%, 100%);
- результаты распознавания, статистика;
- выводы.

Таблица 2.1

### Варианты задания

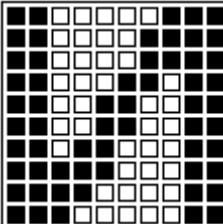
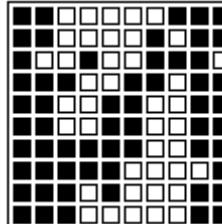
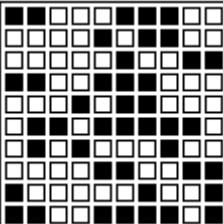
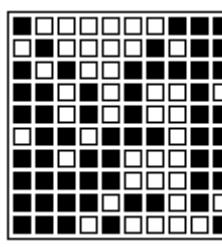
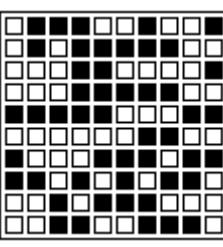
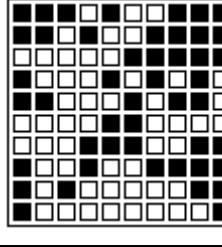
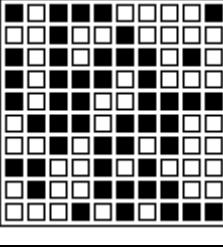
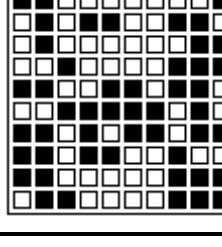
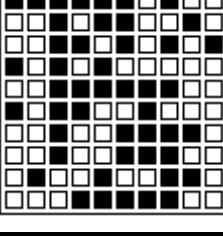
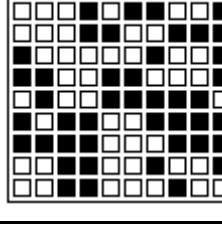
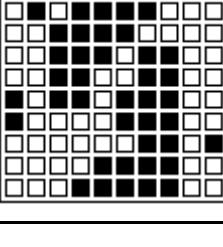
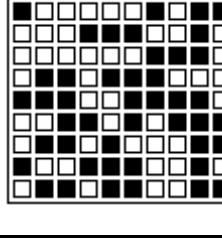
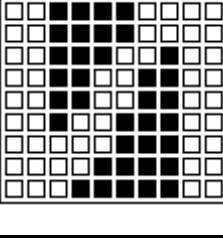
№ варианта	1-ый тестируемый образ	2-ой тестируемый образ	3-ий тестируемый образ
1	«А»	«И»	«Р»
2	«Б»	«К»	«С»
3	«В»	«Л»	«Т»
4	«Г»	«М»	«У»
5	«Д»	«Н»	«Х»
6	«Е»	«О»	«Ш»
7	«З»	«П»	«Б»

## 4. Контрольные вопросы

1. Топология сети Хопфилда.
2. Обучение сети Хопфилда.
3. Процесс воспроизведения информации в сети Хопфилда.
4. Зависимость максимального количества образов, запоминаемых сетью, от ее размера.
5. В чем причина некорректной работы при запоминании подобных образов?
6. Варианты использования сети Хопфилда.

Таблица 2.2

Пример задания тестируемого и искаженных образов

Тестируемый образ	Процент зашумления образа	Вид искаженного образа	Процент зашумления образа	Вид искаженного образа
	10%		50%	
	20%		60%	
	30%		70%	
	35%		80%	
	40%		90%	
	45%		100%	

# Лабораторная работа №2

## Многослойный персептрон

### 1. Цель работы

Изучение топологии, алгоритма функционирования многослойного персептрана.

### 2. Теоретические сведения

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 (1.2) – (1.4), т.е. многослойный персептрон обычно состоит из 3 слоев: первого распределительного, второго скрытого и третьего выходного (рис. 3.1).

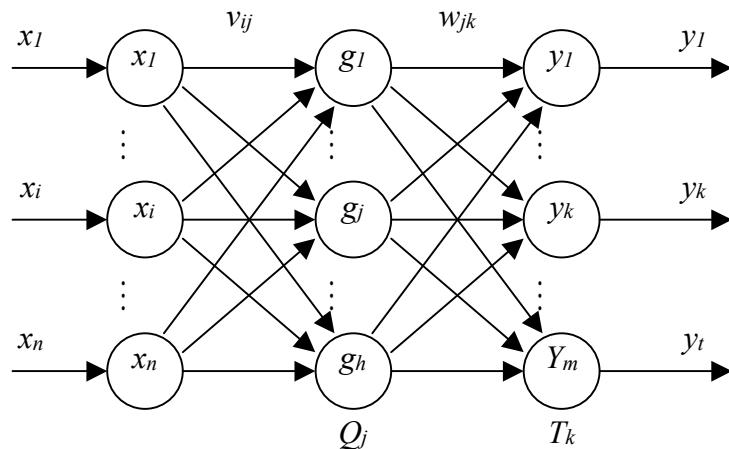


Рис. 3.1. Многослойный персептрон

Такая сеть имеет  $n$  входов и  $n$  нейронов распределительного слоя,  $h$  нейронов скрытого слоя и  $t$  выходных нейронов. Используются две матрицы весов: скрытого слоя  $v$  размером  $n \times h$  и выходного слоя  $w$  размером  $h \times t$ . Кроме этого, с каждым слоем нейронов связан массив порогов:  $Q$  – для скрытого слоя,  $T$  – для выходного. Эти данные представляют собой знания сети, настраиваемые в процессе обучения и определяющие ее поведение. Персептрон функционирует по следующим формулам:

$$g_j = f\left(\sum_{i=1}^n v_{ij} x_i + Q_j\right), \quad (3.1)$$

$$y_k = f\left(\sum_{j=1}^h w_{jk} g_j + T_k\right). \quad (3.2)$$

В качестве функции активации используется одна из функций (1.2) – (1.4). Вид функции определяет диапазон чисел, в котором работает сеть. В дальнейшем будет использоваться сигмоидная функция (1.2), имеющая область значений от 0 до 1.

Обучение с учителем ставит перед сетью задачу обобщить  $p$  примеров, заданных парами векторов  $(x^r, y^r)$ ,  $r = \overline{1, p}$ . Вектор  $x^r = (x_1^r, x_2^r, \dots, x_i^r, \dots, x_n^r)$  в случае задачи классификации задает входной образ (вектор признаков), а вектор  $y^r = (y_1^r, y_2^r, \dots, y_k^r, \dots, y_m^r)$ , задающий эталонный выход, должен кодировать номер класса. При этом есть множество вариантов кодирования. Оптимальным представляется кодирование, когда номер класса определяется позицией единичной компоненты в векторе  $y^r$ , а все остальные компоненты равны 0. Каждый выходной нейрон соответствует одному классу. Такой способ позволяет при классификации определять вероятность каждого класса по величине на выходе соответствующего нейрона (чем ближе к единице, тем вероятность больше).

Обучение персептрона проводится с помощью алгоритма обратного распространения ошибки, который минимизирует среднеквадратичную ошибку нейронной сети. Для этого с целью настройки синаптических связей используется метод градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Рассмотрим алгоритм обратного распространения ошибки.

1. На первом этапе происходит начальная инициализация знаний сети. Простейший вариант такой инициализации – присвоить всем весам и порогам случайные значения из диапазона  $[-1, 1]$ .

2. Далее для каждой пары векторов  $(x^r, y^r)$  выполняется следующее:

2.1. Для входного вектора рассчитываются выходы нейронов скрытого слоя и выходы сети по формулам (3.1), (3.2).

2.2. Происходит коррекция знаний сети, при этом главное значение имеет отклонение реально полученного выхода сети  $y$  от идеального вектора  $y^r$ . Согласно методу градиентного спуска, изменение весовых коэффициентов и порогов нейронной сети происходит по следующим формулам:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \frac{\partial E}{\partial w_{jk}(t)}, \quad (3.3)$$

$$T_k(t+1) = T_k(t) + \alpha \frac{\partial E}{\partial T_k(t)}, \quad (3.4)$$

где  $E$  - среднеквадратичная ошибка нейронной сети для одного образа, а  $\alpha$  – параметр, определяющий скорость обучения. Формулы записаны в терминах выходного слоя, аналогично выглядят формулы для скрытого слоя. Среднеквадратичная ошибка сети вычисляется как

$$E = \frac{1}{2} \sum_{k=1}^m (y_k^r - y_k)^2. \quad (3.5)$$

Ошибка  $k$ -го нейрона выходного слоя определяется как

$$d_k = \frac{\partial E}{\partial y_k} = y_k^r - y_k. \quad (3.6)$$

Выразим производные из формул (3.3), (3.4) через легко вычисляемые величины. Определим взвешенную сумму, аргумент функции активации как

$$S_k = \sum_{j=1}^h w_{jk} g_j + T_k. \quad (3.7)$$

Из соотношения (3.3)  $\frac{\partial E}{\partial w_{jk}}$  можно представить как:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial w_{jk}}, \quad (3.8)$$

где  $\frac{\partial E}{\partial y_k} = d_k = y_k^r - y_k$  – ошибка  $k$ -го нейрона;  $\frac{\partial y_k}{\partial S_k} = f'(S)$  – производная

функции активации;  $\frac{\partial S_k}{\partial w_{jk}} = g_j$  – значение  $j$ -го нейрона предыдущего слоя.

Получаем

$$\frac{\partial E}{\partial w_{jk}} = d_k f'(S_k) g_j. \quad (3.9)$$

Аналогично (3.8), с учетом того, что  $\frac{\partial S_k}{\partial T_k} = 1$ , получаем

$$\frac{\partial E}{\partial T_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial T_k} = d_k f'(S_k). \quad (3.10)$$

Веса и пороги скрытого слоя также корректируются по формулам, аналогичным (3.3), (3.4), с учетом (3.9) и (3.10). При этом главной трудностью является определение ошибки нейрона скрытого слоя. Эту ошибку явно определить по формуле, аналогичной (3.6), нельзя, однако существует возможность рассчитать ее через ошибки нейронов выходного слоя (отсюда произошло название алгоритма обратного распространения ошибки):

$$e_j = \frac{\partial E}{\partial g_j} = \sum_{k=1}^m \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial S_k} \frac{\partial S_k}{\partial g_j} = \sum_{k=1}^m d_k f'(S_k) w_{jk}. \quad (3.11)$$

Производные от функций активации тоже легко рассчитываются, например, для сигмоидной функции получаем

$$f'(S_k) = \left( \frac{1}{(1 + e^{-S})} \right)' = f(S_k)(1 - f(S_k)) = y_k(1 - y_k). \quad (3.12)$$

Аналогично:

$$(th(S_k))' = 1 - y_k^2;$$

$$\left( \frac{2}{(1 + e^{-x})} - 1 \right)' = \frac{(1 - y_k^2)}{2}.$$

Таким образом, можно записать окончательные выражения (3.3), (3.4) для двух слоев, использующих сигмоидную функцию:

$$w_{jk} = w_{jk} + \alpha y_k (1 - y_k) d_k g_j, \quad (3.13)$$

$$T_k = T_k + \alpha y_k (1 - y_k) d_k, \quad (3.14)$$

$$v_{ij} = v_{ij} + \beta g_j (1 - g_j) e_j x_i, \quad (3.15)$$

$$Q_j = Q_j + \beta g_j (1 - g_j) e_j. \quad (3.16)$$

В эти формулы вводится дополнительный параметр –  $\beta$  – скорость обучения скрытого слоя, который может отличаться от аналогичного параметра для выходного слоя. Рекомендуется изменять скорости обучения обратно пропорционально количеству шагов 2 алгоритма обучения, однако это не всегда оправдывает себя на практике.

3. После того, как коррекция знаний произведена для каждой пары векторов, можно оценить степень успешности обучения сети для определения момента завершения алгоритма. Для этого можно использовать в качестве критерия максимальную по модулю ошибку на выходе  $d_k$ , полученную на шаге 2. Условием прекращения обучения в этом случае будет

$$\max |d_k| < D, k = \overline{1, m}, r = \overline{1, p}, \quad (3.17)$$

где  $D$  – достаточно маленькая константа – величина максимальной ошибки, которую требуется достичь в процессе обучения. Если условие (3.17) не выполняется, то шаг 2 повторяется.

Способность персептрона-классификатора разделять образы в пространстве признаков прежде всего зависит от его скрытого слоя. Именно на этот слой возлагается задача сделать множество классов линейно разделимым для успешной работы выходного слоя. Очевидно, что чем больше нейронов в скрытом слое, тем большее количество примеров этот слой может разделять. Кроме этого, увеличение числа признаков входных образов также способствует успешному их разделению в пространстве признаков. Однако увеличение этих параметров приводит к росту ошибок сети и времени обучения. Увеличение размерности входов  $n$  приводит к росту ошибки аппроксимации сети, возникающей из-за обобщения данных. Увеличение числа нейронов  $h$  скрытого слоя приводит к росту ошибки, связанной со сложностью модели. Персептрону легче провести функцию через эталонные точки, однако при этом обобщающая способность сети ухудшается. Он хуже предсказывает поведение аппроксимируемой функции на образах, не входящих в обучающую выборку. Такое состояние сети называется переобучением.

Оптимальное соотношение между этими параметрами оценивают как

$$h \sim \sqrt{\frac{p}{n}}. \quad (3.18)$$

Эксперименты показывают, что обучение максимально успешно проходит на множестве классов, хорошо (желательно линейно) разделенных в пространстве признаков. Это достигается удачным подбором информативных признаков. Если классы в пространстве признаков хорошо кластеризуются, т.е. образы каждого класса составляют компактную группу, достаточно удаленную от других групп, то есть возможность уменьшить размер обучающей выборки  $p$ .

(используются только центры кластеров) и затем уменьшить число нейронов  $h$ . Это приводит к ускорению обучения и улучшает работу классификатора.

Другой проблемой является то, что алгоритм градиентного спуска не гарантирует нахождение глобального минимума среднеквадратичной ошибки сети (3.5), а гарантируется определение только локального минимума. Проблемы, возникающие в процессе градиентного спуска, можно проанализировать на примере функции ошибки, схематически изображенной на рис. 3.2.

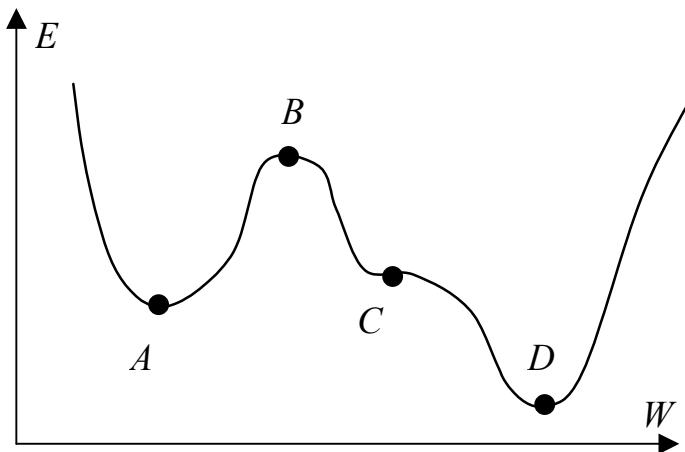


Рис. 3.2. Схематическая функция ошибки сети

На рис. 3.2 показаны четыре критические точки, производная функции ошибки в которых близка к нулю. Точка  $A$  соответствует локальному минимуму. Признаком достижения локального минимума в процессе обучения является полное прекращение уменьшения ошибки (3.17). В этом случае может помочь повторное обучение с другим начальным распределением знаний (тут может помочь случайная инициализация). Точка  $B$  — локальный максимум. В случае попадания в окрестность такой точки скорость резко падает, затем снова быстро растет. Не существует способа предсказать, в какую сторону лучше двигаться из точки с производной, близкой к нулю. Точка  $C$  — точка перегиба, характеризуется длительным уменьшением скорости. Точка  $D$  — глобальный минимум — цель алгоритма.

Существуют многочисленные способы оптимизации метода градиентного спуска, призванные улучшить поведение алгоритма в подобных критических точках. Эффективной модификацией является введение момента, накапливающего влияние градиента на веса со временем. Тогда величина  $\Delta w$  (3.3) в момент времени  $t$  будет вычисляться как

$$\Delta w(t) = \alpha \frac{\partial E}{\partial w} + \mu \Delta w(t - I), \quad (3.19)$$

где  $\mu$  - параметр, определяющий величину влияния момента. С использованием (3.19) скорость изменения весов возрастает на участках с постоянным знаком производной. В окрестностях минимума скорость резко падает за счет колебания знака.

Достоинства алгоритма – большая скорость в точках перегиба, возможность по инерции преодолевать небольшие локальные минимумы. Недостатки – еще один параметр, величину которого следует подбирать и настраивать.

Этот и другие алгоритмы оптимизации обучения персептрона позволяют улучшить работу сети в условиях плохой сходимости. Однако они усложняют процесс обучения, не гарантируя в то же время полного успеха во всех случаях. Успех обучения классификатора зависит от самого алгоритма обучения и качества обучающей выборки.

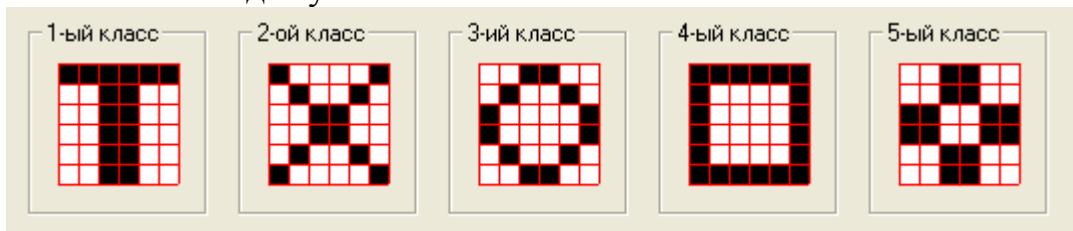
### 3. Задание

1. Ознакомьтесь с теоретической частью.
2. На языке C, C++ напишите программу, реализующую многослойный персептрон.
3. Произведите обучение многослойного персептрона. Исходные данные - 5 классов образов, размер идеального образа  $6 \times 6$  (в соответствии с вариантом).
4. Подайте на вход сети ряд тестовых образов, по 3 зашумленных образа каждого из 5 классов.
5. Проанализируйте результаты работы программы, которые должны иметь следующий вид:

- вывести распознаваемый зашумленный образ.



- вывести процент подобия распознаваемого зашумленного образа по отношению к каждому из 5 классов.



Подобие (%)	32.40036720418	37.85587763332	1.649962583682	5.446987201946	54.34485680307
-------------	----------------	----------------	----------------	----------------	----------------

- вывести количество шагов, затраченных на обучение сети на заданное количество классов.

6. Напишите отчет.

Содержание отчета:

- топология многослойного персептрона;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения многослойного персептрона;
- тестовые зашумленные образы;
- результаты воспроизведения: процент подобия по отношению к каждому из классов, количество шагов, затраченных на обучение;
- результаты сравнения многослойного персептрона с нейронной сетью Хопфилда;
- выводы: преимущества и недостатки многослойного персептрона.

Таблица 3.1

Варианты задания

№ варианта	1-ый класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	$\wedge$	$\vee$	$\exists$	$\subset$	$\supset$
4	$\otimes$	$\oplus$	$\times$	$\div$	$\pm$
5	$\leq$	$\geq$	$\neq$	$\equiv$	$\approx$
6	L	U	T	O	K
7	$\rightarrow$	$\leftrightarrow$	$\leftarrow$	$\downarrow$	$\lrcorner$

#### 4. Контрольные вопросы

1. Топология многослойного персептрона.
2. Процесс обучения.
3. Процесс воспроизведения.
4. Процедура обратного распространения ошибки.
5. Каким образом можно улучшить качество распознавания?
6. Достоинства и недостатки данного типа нейронной сети.

# Лабораторная работа № 3

## Сеть РБФ

### 1. Цель работы

Изучение топологии, алгоритма функционирования сети РБФ.

### 2. Теоретические сведения

Сеть РБФ (радиальная базисная функция) является аналогом многослойного персептрана (рис. 4.1). Скорость обучения такой сети гораздо выше, причем допускается полностью аналитический подход к расчету весовых коэффициентов. Однако эти положительные моменты сопровождаются рядом недостатков, главным из которых является ухудшение точности аппроксимации. Сеть обладает хорошей обобщающей способностью только для ограниченного класса аппроксимируемых функций. В качестве классификатора такая сеть может с успехом применяться в случае хорошей кластеризации классов в пространстве признаков.

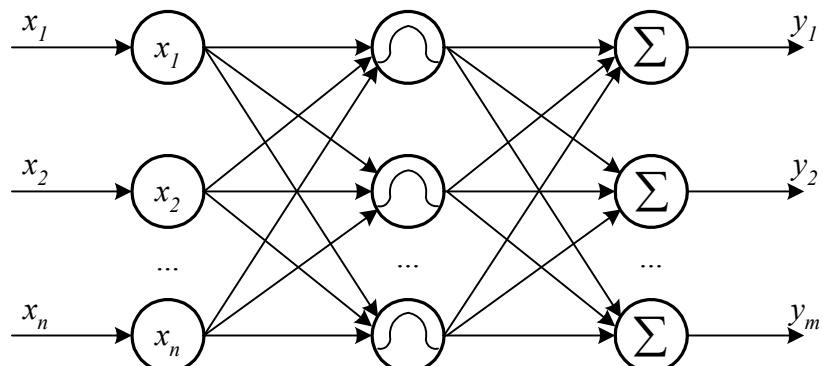


Рис. 4.1. Топология сети РБФ

Упрощение работы и обучения достигается за счет введения вместо скрытого слоя нейронов слоя РБФ ячеек. Классический закон, по которому такая ячейка функционирует, определяется формулой гауссового колокола:

$$g_j = \exp\left(\frac{-|x - t^j|^2}{\sigma_j^2}\right), \quad (4.1)$$

где  $x$  – входной вектор;  $t^j$  – вектор, определяющий математическое ожидание (центр кластера в пространстве признаков) РБФ ячейки;  $\sigma_j$  – среднеквадратическое отклонение или параметр, зависящий от величины разброса образов класс-

са от его центра (рис. 4.1). В данном выражении евклидово расстояние между векторами  $x$  и  $t^j$  вычисляется как

$$|x - t^j|^2 = (x_1 - t_1^j)^2 + (x_2 - t_2^j)^2 + \dots + (x_n - t_n^j)^2.$$

**Обучение.** РБФ ячейки обучаются путем подбора центра и отклонения каждой из них. Для классификатора в качестве центра выбирается центр кластера в пространстве признаков, компактно содержащего образы одного и того же класса. В простейшем случае, если класс задается одним идеальным образом, этот образ и будет являться вектором  $t$  – центром РБФ ячейки. Параметр разброса каждой ячейки выбирается в зависимости от величины радиуса кластера или расстояния до соседних центров. Ряд авторов рекомендует выбирать  $\sigma$  как половину расстояния до ближайшего центра ячейки, соответствующей другому классу. Количество РБФ ячеек выбирается таким образом, чтобы покрыть гауссовыми колоколами все классы.

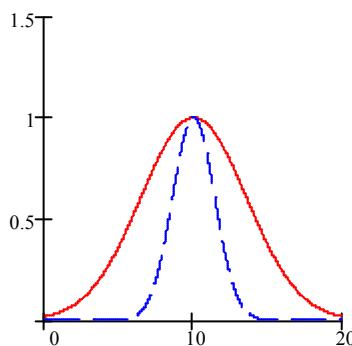


Рис. 4.2. Примеры функций РБФ ячеек с одинаковым центром и разным разбросом

Выходной слой РБФ сети обычно состоит из суммирующих ячеек

$$y_k = \sum_{j=1}^h w_{jk} g_j. \quad (4.2)$$

Это позволяет при определенных условиях использовать систему линейных уравнений для определения весов выходного слоя. В других обстоятельствах можно использовать алгоритм градиентного спуска для настройки весов выходного слоя (функция активации в данном случае линейная). С учетом того, что обучается только выходной слой нейронной сети, а скрытый уже настроен, обучение такой сети проходит на порядок быстрее, чем аналогичного многослойного персептрона.

В случае обучения выходного слоя по алгоритму градиентного спуска формулы расчета из лабораторной работы 2.2 значительно упрощаются. Формулы (3.1) и (3.2) заменяются соответственно на (4.1) и (4.2). Формула (3.13) упрощается до

$$w_{jk} = w_{jk} + \alpha d_k g_j, \quad (4.3)$$

поскольку функция активации в выходном слое сети РБФ линейная и ее производная равна 1. Для коррекции весовых коэффициентов выходных суммирующих ячеек используется только (4.3).

Контроль завершения алгоритма обучения производится аналогично.

**Воспроизведение.** Сеть функционирует по формулам (4.1) и (4.2).

Очевидно, что функция на выходе РБФ сети будет представлять собой суперпозицию гауссовых колоколов. В этом заключается ограничение данного класса классификаторов. Кроме этого, при неудачном выборе признаков большой проблемой является выбор количества РБФ ячеек, определение их центров и отклонения. С уменьшением числа РБФ ячеек улучшается обобщение данных сетью, но могут проявляться большие ошибки в эталонных точках.

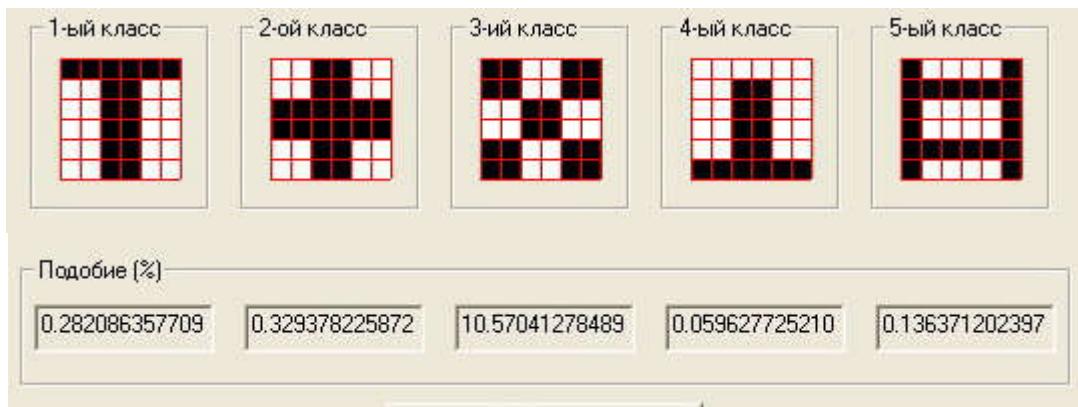
### 3. Задание

1. Ознакомьтесь с теоретической частью.
2. Написать программу на С, С++, реализующую сеть РБФ.
3. Произведите обучение сети РБФ. Исходные данные - 5 классов образов, размер идеального образа  $6 \times 6$  (в соответствии с вариантом).
4. Подайте на вход сети ряд тестовых образов, по 3 зашумленных образа каждого из 5 классов.
5. Проанализируйте результаты работы программы, которые должны иметь следующий вид:

- вывести распознаваемый зашумленный образ;



- вывести процент подобия распознаваемого зашумленного образа по отношению к каждому из 5 классов;



- вывести количество шагов, затраченных на обучение сети на заданное количество классов.

6. Напишите отчет.

Содержание отчета:

- топология сети РБФ;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения сети РБФ;
- тестовые зашумленные образы;
- результаты воспроизведения: процент подобия по отношению к каждому из классов, количество шагов, затраченных на обучение;
- результаты сравнения сети РБФ с многослойным персептроном;
- выводы: преимущества и недостатки сети РБФ.

Таблица 4.1  
Варианты задания

№ варианта	1-й класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	$\wedge$	$\vee$	$\exists$	$\subset$	$\supset$
4	$\otimes$	$\oplus$	$\times$	$\div$	$\pm$
5	$\leq$	$\geq$	$\neq$	$\equiv$	$\approx$
6	L	U	T	O	K
7	$\rightarrow$	$\leftrightarrow$	$\leftarrow$	$\downarrow$	$\lrcorner$

#### 4. Контрольные вопросы

1. Топология сети РБФ.
2. Обучение сети, выбор количества РБФ ячеек.
3. Процесс воспроизведения информации в сети РБФ.
4. Каким образом можно улучшить работу сети?
5. Достоинства и недостатки данного типа нейронной сети.

# Лабораторная работа № 4

## Конкурентная нейронная сеть

### 1. Цель работы

Изучение топологии, алгоритма функционирования конкурентной нейронной сети.

### 2. Теоретические сведения

Самоорганизующиеся нейронные сети обучаются без учителя. Они способны адаптироваться к входным данным, используя содержащиеся в этих данных зависимости. Такие сети используются для нахождения более компактного описания данных (сжатия), кластеризации, выделения признаков.

Конкурентная сеть является простейшей самоорганизующейся нейронной сетью (рис. 5.1).

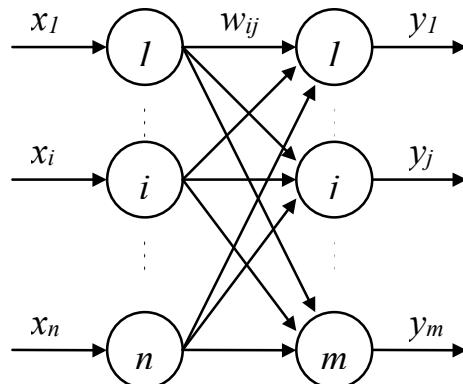


Рис. 5.1. Конкурентная нейронная сеть

Первый слой является распределительным. Нейроны второго слоя функционируют по формуле

$$y_j = \sum_{i=1}^n w_{ij} x_i = |w^j| \|x\| \cos \alpha, \quad (5.1)$$

где  $x = (x_1, x_2, \dots, x_i, \dots, x_n)$  – входной вектор;  $w^j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})$  – вектор весовых коэффициентов нейрона, а  $|x|$  и  $|w^j|$  – их модули,  $\alpha$  – угол между ними.

**Обучение.** При обучении нейронной сети при подаче каждого входного вектора определяется нейрон-победитель, для которого (5.1) максимально. Для этого нейрона синаптические связи усиливаются по формуле

$$w_{ij}(t+1) = w_{ij}(t) + \beta(x_i - w_{ij}(t)), \quad (5.2)$$

где  $\beta$  – скорость обучения.

Смысл этой формулы в том, что вектор весовых коэффициентов нейрона – победителя “поворачивается” в сторону входного вектора, тем самым активность нейрона усиливается. Удобно работать с нормированными входными и весовыми векторами, когда их модуль равен 1. Нормировка уравнивает шансы в конкуренции нейронов с разным модулем вектора весовых коэффициентов. Выражение (5.1) для нормированных векторов будет выглядеть как

$$y_j = \sum_{i=1}^n w_{ij} x_i = \cos \alpha, \quad (5.3)$$

а выражение (5.2):

$$w_{ij}(t+1) = \frac{w_{ij}(t) + \beta(x_i - w_{ij}(t))}{\left| w^j(t) + \beta(x - w^j(t)) \right|}. \quad (5.4)$$

Случайное начальное распределение весовых коэффициентов может привести к тому, что некоторые нейроны никогда не станут победителями, так как их весовые векторы окажутся удаленными от всех входных векторов. Существует ряд модификаций алгоритма обучения, позволяющих устранить этот недостаток. Хорошие результаты на практике показало частотно–зависимое конкурентное обучение. Согласно нему, нейрон-победитель определяется по минимуму произведения евклидового расстояния между входным и весовым вектором и количеством побед данного нейрона  $f_j$ :

$$d_v = \min_j (\left| x - w^j \right| f_j). \quad (5.5)$$

Шансы нейрона на победу уменьшаются с количеством побед, что дает преимущество другим нейронам.

Конкурентное обучение продолжается до тех пор, пока максимум евклидового расстояния между любым входным вектором и соответствующим ему вектором весов нейрона-победителя не достигнет заданного малого значения.

Конкурентная сеть позволяет разбить входную выборку нормированных векторов на  $m$  (количество выходных нейронов сети) кластеров, расположенных на поверхности гиперсферы в пространстве признаков единичного радиуса. Входные векторы, приводящие к победе одного и того же нейрона, относят к одному кластеру.

Кохонен предложил внести в правило конкурентного обучения (5.2) информацию о расположении нейронов в выходном слое. Для этого нейроны упорядочиваются в одномерные или двухмерные решетки. Вводится функция, корректирующая изменение весов в зависимости от расстояния до нейрона–победителя  $h(t, k, j)$  – сила влияния между нейроном–победителем  $k$  и нейроном  $j$  в момент времени  $t$ . Для  $j=k$  эта функция всегда равна 1 и уменьшается с ростом расстояния между  $k$  и  $j$  в решетке. С течением времени радиус влияния обычно сужается. С использованием этой функции веса меняются для всех нейронов сети, а не только для нейрона–победителя:

$$w_{ij}(t+1) = w_{ij}(t) + \beta h(t, k, j)(x_i - w_{ij}(t)). \quad (5.6)$$

В качестве функции  $h(t, k, j)$  может использоваться гауссовый колокол (4.1) с параметром  $\sigma$ , зависящим от времени или функция вида “мексиканская шляпа”.

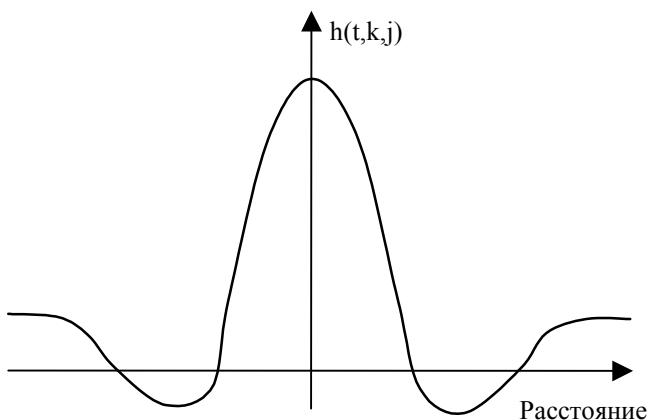


Рис. 5.2. Пример функции расстояния в сетях Кохонена

В результате модификации конкурентного обучения сеть Кохонена не только кластеризирует входные примеры, но и упорядочивает их в виде одномерной или двухмерной решетки. Это позволяет получить дополнительную информацию о близости кластеров. Если два кластера проецируются на соседние нейроны решетки, это говорит об их близости в исходном пространстве признаков. Обратное неверно. Из-за уменьшения размерности пространства отношение близости сохраняется только для ограниченного числа кластеров.

### 3. Задание

1. Ознакомьтесь с теоретической частью.

2. Написать программу на С, С++, реализующую конкурентную нейронную сеть.

3. Обучить конкурентную сеть с использованием правила (5.5) на количество образов, превышающих количество нейронов сети. Рекомендуется использовать нормированные векторы. Исходные данные - 5 классов образов, размер идеального образа  $6 \times 6$  (в соответствии с вариантом).

4. Убедиться, что похожие образы были спроектированы сетью в один кластер (подача их на вход активизирует один и тот же нейрон).

5. Подать на вход тестовые образы, отличные от образов из обучающей выборки. Сделать выводы.

6. Напишите отчет.

Содержание отчета:

- топология конкурентной нейронной сети;
- основные формулы обучения и воспроизведения;
- идеальные образы для обучения сети;
- тестовые зашумленные образы;
- результаты воспроизведения;
- результаты сравнения конкурентной нейронной сети с сетью РБФ и многослойным персепtronом;
- выводы: преимущества и недостатки конкурентной нейронной сети.

Таблица 5.1

Варианты задания

№ варианта	1-ый класс	2-ой класс	3-ий класс	4-ый класс	5 класс
1	2	3	4	5	7
2	N	F	I	P	D
3	$\wedge$	$\vee$	$\exists$	$\subset$	$\supset$
4	$\otimes$	$\oplus$	$\times$	$\div$	$\pm$
5	$\leq$	$\geq$	$\neq$	$\equiv$	$\approx$
6	L	U	T	O	K
7	$\rightarrow$	$\leftrightarrow$	$\leftarrow$	$\downarrow$	$\lrcorner$

#### 4. Контрольные вопросы

1. Смысл самообучения.
2. Обучение конкурентной нейронной сети.
3. Определение нейрона-победителя.
4. Сеть Кохонена.
5. Использование самоорганизующихся сетей.
6. Достоинства и недостатки данного типа нейронной сети.

## Литература

1. Aleksander I., Morton H. An Introduction to Neural Computing. — London: Chapman&Hall, 1990.
2. Головко В.А. Нейронные сети: обучение, организация и применение. Учеб. пособие для вузов.— М.:ИПРЖР, 2001. – 256 с.
3. Bishop C.M. Neural Networks for Pattern Recognition.— Oxford: Clarendon press, 1995.— 482 p.
4. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Natl. Acad. Sci. USA.— 1982.— Vol. 79.— P. 2554.
5. Kohonen T. Self-organization and associative memory. – Springer-Verlag, 1989.— 312 p.
6. Kohonen T. Self-organized formation of topologically correct feature maps // Biol. Cybernetics.— 1982.— Vol. 43.— P. 56-69.
7. Kohonen T. Self-organizing maps. –Springer-Verlag, 1995.— 362 p.
8. Rumelhart D.E., Hinton G.E., Wiliams R.J. Learning internal representation by error propagation: McClelland J.L. and Rumelhart D.E. (Eds). Parallel Distributed Processing: Exploration in the Microstructure of Cognition.— MIT Press, Cambridge MA.— 1986.— Vol. 1.— P. 318-362.
9. Хайкин С. Нейронные сети: полный курс, 2-е изд.: Пер. с англ. — М. : Издательский дом «Вильямс», 2006. — 1104 с.
10. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применение в экономике и бизнесе. —М.: Мир, 1998.— 222 с.