

**Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Амурский государственный университет»**

Кафедра Информационных и управляющих систем

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ  
«НЕЙРОИНФОРМАТИКА»**

Основной образовательной программы по специальности 230201.65 – Информационные системы и технологии

УМКД разработан доцентом Акиловой Ириной Михайловной

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от « \_\_\_\_\_ » \_\_\_\_\_ 2012 г. № \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ / А.В.Бушманов

### **УТВЕРЖДЕН**

Протокол заседания УМСС специальности 230201.65 – Информационные системы и технологии

от « \_\_\_\_\_ » \_\_\_\_\_ 2012 г. № \_\_\_\_\_

Председатель УМСС \_\_\_\_\_ / В.В.Еремина /

## СОДЕРЖАНИЕ

Рабочая программа	3
Краткий конспект лекций	11
Методические указания по выполнению лабораторных работ	25
Методические рекомендации по проведению самостоятельной работы студентов	38
Фонд контрольных заданий для оценки качества знаний по дисциплине	38
Комплект экзаменационных билетов	41

## **РАБОЧАЯ ПРОГРАММА**

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Амурский государственный университет»

УТВЕРЖДАЮ  
Проректор по учебной работе

\_\_\_\_\_ В.В. Проказин

«\_\_» \_\_\_\_\_ 20\_\_ г.

## **РАБОЧАЯ ПРОГРАММА НЕЙРОИНФОРМАТИКА**

для специальности 230201.65 «Информационные системы и технологии»

Квалификация (степень) выпускника – инженер

Курс – 3

Семестр – 5

Лекции – 36 (час.)

Экзамен – 5

Лабораторные занятия – 18 (час.)

Самостоятельная работа – 66 (час.)

Общая трудоемкость дисциплины – 120 (час.)

Составитель – И.М.Акилова, доцент

Факультет математики и информатики

Кафедра информационных и управляющих систем

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования по специальности 230201.65 «Информационные системы и технологии»

Рабочая программа обсуждена на заседании кафедры информационных и управляющих систем

«\_\_» \_\_\_\_\_ 20\_\_ г., протокол № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ А.В. Бушманов

Рабочая программа одобрена на заседании учебно-методического совета специальности 230201.65 «Информационные системы и технологии»

«\_\_» \_\_\_\_\_ 20\_\_ г., протокол № \_\_\_\_\_

Председатель \_\_\_\_\_ В.В. Еремина

Рабочая программа переутверждена на заседании кафедры информационных и управляющих систем

«\_\_» \_\_\_\_\_ 20\_\_ г., протокол № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ А.В. Бушманов

СОГЛАСОВАНО  
Учебно-методическое  
управление

\_\_\_\_\_  
«\_\_» \_\_\_\_\_ 20\_\_ г.

СОГЛАСОВАНО  
Председатель учебно-методического  
совета факультета

\_\_\_\_\_ С.Г. Самохвалова  
«\_\_» \_\_\_\_\_ 20\_\_ г.

СОГЛАСОВАНО  
Заведующий выпускающей кафедрой

\_\_\_\_\_ А.В. Бушманов  
«\_\_» \_\_\_\_\_ 20\_\_ г.

СОГЛАСОВАНО  
Директор научной библиотеки

\_\_\_\_\_ Л.А. Проказина  
«\_\_» \_\_\_\_\_ 20\_\_ г.

## 1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины - дать представление о прикладных программных средствах, основанных на нейронных сетях; дать представления об инструментальном ПО для обучения нейронных сетей и экспериментов с ними; подготовить студентов к использованию нейросетевых технологий в научно-исследовательской деятельности.

В результате изучения дисциплины у студентов должны быть сформированы представления о разных моделях нейронных сетей, их особенностях; об особенностях решения задач с помощью нейронных сетей; об существующих прикладных системах, основанных на применении нейронных сетей; об этапах решения задач с помощью нейронных сетей.

После изучения дисциплины студент должен знать содержание:

- принципов постановки задач для решения с помощью нейронных сетей;
- методов представления данных для обучения и использования нейронных сетей;
- методов обучения нейронных сетей и оценки качества обучения нейронной сети;
- этапов решения задач с помощью нейронных сетей.

После изучения дисциплины студент должен приобрести умения и навыки:

- ориентироваться в различных типах прикладных систем, основанных на использовании нейронных сетей;
- ориентироваться а различных методах представления данных для обучения нейронной сети;
- выбирать и ставить задачу для решения ее нейронной сетью;
- выбирать модель нейронной сети для решения задачи.

## 2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Программа курса «Нейроинформатика» составлена в соответствии с требованиями государственного образовательного стандарта специализации – Информационные системы и технологии, специализации 230201.65, блок дисциплины по выбору ЕН.В.01.

Для успешного освоения дисциплины «Нейроинформатика» используются знания, умения, навыки и виды деятельности, полученные в ходе изучения дисциплин: «Математический анализ» (Б.2.Б.1), «Алгебра и геометрия» (Б.2.Б.2), «Информатика» (Б.2.Б.3) .

## 3 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 120 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах				Формы текущего контроля успеваемости Форма промежуточной аттестации
				Лек	Пр	Лаб	Сам	
1	Ведение	5	1-2	6		1	5	Подготовка к защите лабораторной работы
2	Модели нейронов и методы их обучения	5	3-4	4		1	5	Защита лабораторной работы
3	Однонаправленные многослойные сети сигмоидального типа	5	5-6	2		1	5	Подготовка к защите лабораторной работы
4	Проблемы практического использования искусственных нейронных сетей и их свойства	5	7-8	2		1	5	Защита лабораторной работы

5	Радиальные нейронные сети	5	9-10	4		2	5	Защита лабораторной работы
6	Рекуррентные сети как ассоциативные запоминающие устройства	5	11-12	4		2	5	Защита лабораторной работы
7	Рекуррентные сети на базе перцептрона	5	13-14	4		2	5	Защита лабораторной работы
8	Сети с самоорганизацией на основе конкуренции	5	14-15	6		2	10	Защита лабораторной работы
9	Сеть с самоорганизацией корреляционного типа	5	16-17	2		3	10	Защита лабораторной работы
10	Вероятностная нейронная сеть	5	17-18	2		3	11	Защита лабораторной работы
11	Всего по разделам	5	1-18	36		18	66	Экзамен

#### 4 СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

##### 4.1 Лекции

##### 1. ВВЕДЕНИЕ

Биологические основы функционирования нейрона. Первые модели нейронной сети. Прикладные возможности нейронных сетей. Определение искусственных нейронных сетей. Свойства биологических и искусственных нейронных сетей. Способы реализации нейросетей. Типы задач, решаемых нейронными сетями. Недостатки и ограничения нейронных сетей.

##### 2. МОДЕЛИ НЕЙРОНОВ И МЕТОДЫ ИХ ОБУЧЕНИЯ.

Перцептрон. Сигмоидальный нейрон. Нейрон типа «адалайн». Инстар и оутстар Гроссберга. Нейроны типа WTA. Модель нейрона Хемминга. Стахостическая модель нейрона.

##### 3. ОДНОНАПРАВЛЕННЫЕ МНОГОСЛОЙНЫЕ СЕТИ СИГМОИДАЛЬНОГО ТИПА.

Однослойная сеть. Многослойный перцептрон. Структура перцептронной сети. Алгоритм обратного распространения ошибки. Градиентные алгоритмы обучения сети: основные положения, алгоритм наискорейшего спуска, алгоритм переменной метрики, алгоритм Левенберга-Марквардта, алгоритм сопряженных градиентов. Подбор коэффициента обучения. Методы инициализации весов.

##### 4. ПРОБЛЕМЫ ПРАКТИЧЕСКОГО ИСПОЛЬЗОВАНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ И ИХ СВОЙСТВА.

Предварительный подбор архитектуры сети. Подбор оптимальной архитектуры сети. Методы наращивания сети. Подбор обучающих выборок. Добавление шума в обучающие выборки. Распознавание и классификация образов. Нейронная сеть для сжатия данных. Идентификация динамических объектов.

##### 5. РАДИАЛЬНЫЕ НЕЙРОННЫЕ СЕТИ..

Математические основы. Радиальная нейронная сеть. Методы обучения радиальных нейронных сетей. Пример использования радиальной сети. Методы подбора количества базисных функций: эвристические методы, метод ортогонализации Грэма-Шмидта.

##### 6. РЕКУРРЕНТНЫЕ СЕТИ КАК АССОЦИАТИВНЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА.

Сети с обратными связями. Послойность сети и матричное умножение. Расчет градиента квадратичной формы. Выбор начальной точки и длины шага. Сеть Хопфилда. Сеть Хемминга. Сеть ART

##### 7. РЕКУРРЕНТНЫЕ СЕТИ НА БАЗЕ ПЕРЦЕПТРОНА.

Перцептронная сеть с обратной связью: структура сети RMLP, алгоритм обучения сети RMLP, подбор коэффициента обучения, коэффициент усиления сигнала. Рекуррентная сеть Элмана: структура сети, алгоритм обучения сети Элмана, обучение с учетом момента.

#### 8. СЕТИ С САМООРГАНИЗАЦИЕЙ НА ОСНОВЕ КОНКУРЕНЦИИ.

Нейронные сети встречного распространения. Сети Кохонена. Обучение слоя Кохонена. Примеры обучения сети Кохонена. Применение сети Кохонена для сжатия данных. Слой Гроссбера: обучение слоя Гроссбера, модификации.

#### 9. СЕТЬ С САМООРГАНИЗАЦИЕЙ КОРРЕЛЯЦИОННОГО ТИПА.

Энергетическая функция корреляционных сетей. Нейронные сети РСА: математическое введение, определение первого главного элемента, алгоритм определения множества главных компонентов.

#### 10. ВЕРОЯТНОСТНАЯ НЕЙРОННАЯ СЕТЬ.

Сети, использующие статистический подход. Метод «модельной закалки». Пример алгоритма минимизации функции. Машина Больцмана. Архитектура нейронной сети PNN. Пример модульной нейронной сети.

### 5 САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	Раздел дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в часах
1	Предсказание стационарного сигнала	Выполнение лабораторной работы, оформление отчета.	22
2	Моделирование стационарного фильтра	Выполнение лабораторной работы, оформление отчета.	22
3	Распознавание образов	Выполнение трех лабораторных работ, оформление отчетов.	22

### 6 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: модульно-рейтинговое обучение, технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Реализация данной модели предполагает использование следующих технологий стратегического уровня (задающих организационные формы взаимодействия субъектов образовательного процесса), осуществляемых с использованием определенных тактических процедур:

- лекционные (вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция);
- практические (углубление знаний, полученных на теоретических занятиях, решение задач);
- тренинговые (формирование определенных умений и навыков, формирование алгоритмического мышления);
- активизации познавательной деятельности (приемы технологии развития критического мышления через чтение и письмо, работа с литературой, подготовка презентаций по темам домашних работ);
- самоуправления (самостоятельная работа студентов, самостоятельное изучение материала).

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), использование мультимедиа-средств при проведении лекционных и практических занятий.

Формы проведения лекционно-практических занятий по дисциплине представлены в таблице (рекомендуемые).

## **7 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

7.1 Оценочные средства для текущего контроля успеваемости

7.1.1 Контрольные вопросы допуска к выполнению практических работ

7.1.2 Отчеты о выполнении индивидуальных вариантов заданий практических работ

7.2 Оценочные средства для промежуточной аттестации

Вопросы к экзамену:

1. Что такое нейронные сети (НС)? Проблемы, возникающие при моделировании. Свойства биологических и искусственных НС.
2. Типы задач, решаемых нейронными сетями. Недостатки и ограничения НС.
3. Биологический нейрон. Структура, функции.
4. Формальный нейрон. Виды функций активации. Ограниченность модели формального нейрона.
5. Многослойный перцептрон. Структура, алгоритм работы. Этапы решения задачи с помощью НС.
6. Формализация условий задачи для НС. Примеры. Подготовка входных и выходных данных. Выбор количества слоев.
7. Обучение однослойного перцептрона. Выбор шагов по  $W, \Theta$ .
8. Проблема "исключающего ИЛИ" и ее решение.
9. Перцептронная представляемость.
10. Метод обратного распространения ошибки.
11. Паралич сети. Выбор шага по параметрам. Локальные минимумы. Временная неустойчивость.
12. Примеры применения перцептронов.
13. Динамическое добавление нейронов. Способность НС к обобщению.
14. Обучение без учителя. Сеть с линейным поощрением.
15. Задача классификации. Сеть Кохонена.
16. Обучение слоя Кохонена. Метод выпуклой комбинации. Примеры обучения.
17. Режимы работы сети Кохонена. Применение для сжатия данных.
18. Сеть встречного распространения. Схема, обучение, свойства.
19. Генетические алгоритмы для обучения НС. Положительные качества и недостатки.
20. Послойность сети и матричное умножение. Расчет градиента квадратичной формы с помощью НС. Выбор начальной точки и длины шага.
21. Сети с обратными связями. Сеть Хопфилда. Вычислительная энергия и ее минимизация.
22. Этапы решения задачи сетью Хопфилда. Устойчивость, сходимость к эталонам.
23. Соотношение стабильности пластичности при запоминании. Сеть АРТ 1. Структура, описание элементов сети.
24. Работа сети АРТ 1. Запоминание и классификация векторов сетью.
25. Метод имитации отжига.

7.3 Учебно-методическое обеспечение самостоятельной работы

7.3.1 Раздаточный материал с заданиями и методическими указаниями по выполнению практических работ

## **8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **а) ОСНОВНАЯ ЛИТЕРАТУРА:**

8.1 Осовский С. Нейронные сети для обработки информации/ С. Осовский ; пер. с пол. И. Д. Рудинского. -М.: Финансы и статистика, 2002. -344 с.:а-рис.

8.2 Хайкин С. Нейронные сети : Полный курс/ С. Хайкин ; пер. с англ., ред. Н. Н. Кусуль, пер. с англ. А. Ю. Шелестова. -2-е изд., испр.. -М.: Вильямс, 2006. -1104 с.:а-рис.

8.3 Каллан Р. Основные концепции нейронных сетей : [учеб. пособие]/ Р. Каллан ; пер. с англ. А. Г. Сивак. -М.: Вильямс, 2001. -288 с.:а-ил.

### **б) ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:**

8.4 Матвеев М.Г. Модели и методы искусственного интеллекта: учеб. пособие/ М.Г.Матвеев, А.С.Свиридов, Н.А.Алейникова. – М.:Финансы и статистика: ИНФРА-М, 2008. – 448 с.

8.5 Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособие / Г.В.Рыбина. – М.:Финансы и статистика: ИНФРА-М, 2010. – 432 с.

8.6 Сергиевский Г.М., Волченков Н.Г. Функциональное и логическое программирование: учеб. пособие: М.: Академия, 2010. – 320 с.

### **г) ПЕРИОДИЧЕСКИЕ ИЗДАНИЯ:**

8.7. «Нейрокомпьютеры: разработка, применение»

8.8 «Информационные технологии и вычислительные системы»

8.9 «Программные продукты и системы»

### **д) ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНТЕРНЕТ-РЕСУРСЫ**

8.10 Свободно распространяемая версия математического пакета MatLAB.

№ п/п	Наименование ресурса	Характеристика
1	<a href="http://www.iqlib.ru">http://www.iqlib.ru</a>	Интернет библиотека образовательных изданий, В которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знаний.
2	<a href="http://www.intuit.ru">http://www.intuit.ru</a>	Интернет-университет информационных технологий. В котором собраны электронные и видеокурсы по отраслям знаний
3	<a href="http://amursu.ru">http://amursu.ru</a>	Сайт АмГУ, Библиотека – электронная библиотека АмГУ

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

9.1 Лекционная аудитория, оборудованная мультимедийными средствами

9.2 Лаборатории, оборудованные рабочими местами пользователей ЭВМ

## 10 РЕЙТИНГОВАЯ ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

Семестровый модуль дисциплины						
№ п/п	Раздел дисциплины	Виды контроля	Сроки выполнения (недели)	Максимальное кол-во баллов	Посещение, активность на занятиях	Максимальное кол-во баллов за модуль
1	Модели нейронов и методы их обучения	ЛР № 1	1-3	7	1	6
2	Однонаправленные многослойные сети	ЛР № 2	4-5	7	1	6
3	Рекуррентные сети как ассоциативные запоминающие устройства	ЛР № 3	6-7	7	1	6
4	Рекуррентные сети на базе перцептрона	ЛР № 4	8-9	8	1	6
5	Сети с самоорганизацией на основе конкуренции	ЛР № 5	10-11	8	1	6
6	Сеть с самоорганизацией корреляционного типа	ЛР № 6	12-14	8	1	6
7	Вероятностная нейронная сеть	ЛР № 7	15-17	8	1	6
8	Всего		18	53	7	60
9	Промежуточная аттестация	Экзамен			0	40
Итого						100

### КРАТКИЙ КОНСПЕКТ ЛЕКЦИЙ

#### Лекция № 1 *Тема: Введение*

Нейронные сети - это раздел искусственного интеллекта, в котором для обработки сигналов используются явления, аналогичные происходящим в нейронах живых существ.

#### **Свойства нейронных сетей**

Важнейшее свойство нейронных сетей, свидетельствующее об их огромном потенциале и широких прикладных возможностях, состоит в параллельной обработке информации одновременно всеми нейронами. Благодаря этой способности при большом количестве межнейронных связей достигается значительное ускорение процесса обработки информации. Во многих ситуациях становится возможной обработка сигналов в реальном масштабе времени.

Кроме того, при большом числе межнейронных соединений сеть приобретает устойчивость к ошибкам, возникающим на некоторых линиях. Функции поврежденных связей берут на себя исправные линии, в результате чего деятельность сети не претерпевает существенных возмущений. Это свойство используется, в частности, при поиске оптимальной архитектуры нейронной сети путем разрыва отдельных связей.

Другое не менее важное свойство - способность к обучению и обобщению накопленных знаний. Нейронная сеть обладает чертами искусственного интеллекта. Натренированная на ограниченном множестве данных сеть способна обобщать полученную информацию и показывать хорошие результаты на данных, не использовавшихся в процессе обучения.

Характерная особенность сети состоит также в возможности ее реализации с применением технологии сверхбольшой степени интеграции. Различие элементов сети невелико, а их повторяемость огромна. Это открывает перспективу создания универсального процессора с однородной структурой, способного перерабатывать разнообразную информацию.

Использование перечисленных свойств на фоне развития устройств со сверхбольшой степенью интеграции (VLSI) и повсеместного применения вычислительной техники вызвало в последние годы огромный рост интереса к нейронным сетям и существенный прогресс в их исследовании. Создана база для выработки новых технологических решений, касающихся восприятия, искусственного распознавания и обобщения видеоинформации, управления сложными системами, обработки речевых сигналов и т.п. Искусственные нейронные сети в практических приложениях, как правило, используются в качестве подсистемы управления или выработки решений, передающей исполнительный сигнал другим подсистемам, имеющим иную методологическую основу.

### **Прикладные возможности нейронных сетей**

Любая нейронная сеть используется в качестве самостоятельной системы представления знаний, которая в практических приложениях выступает, как правило, в качестве одного из компонентов системы управления либо модуля принятия решений, передающих результирующий сигнал на другие элементы, не связанные непосредственно с искусственной нейронной сетью.

Функции, выполняемые сетями, подразделяются на несколько групп:

- 1) аппроксимация и интерполяция;
- 2) классификация и распознавание образов;
- 3) прогнозирование;
- 4) идентификация и оценивание;
- 5) ассоциативное управление.

### **Топология нейронных сетей**

Различные способы объединения нейронов между собой и организации их взаимодействия привели к созданию сетей разных типов.

Классифицируя нейронные сети по топологии, можно выделить три основных типа таких сетей:

- полносвязные сети ;
- многослойные, или слоистые сети ;
- слабосвязные сети (нейронные сети с локальными связями).

*Полносвязные сети* представляют собой искусственные нейронные сети, каждый нейрон которой передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются *всем* нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

В *многослойных сетях* нейроны объединяются в *слои*. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. В общем случае сеть состоит из  $Q$  слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов первого слоя (входной слой часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Вход нейронной сети можно рассматривать как выход «нулевого слоя» вырожденных нейронов, которые служат лишь в качестве распределительных точек, суммирования и преобразования сигналов здесь не производится. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев. Связи от выходов нейронов некоторого слоя  $q$  ко входам нейронов следующего слоя  $(q+1)$  называются последовательными.

В свою очередь среди слоистых сетей выделяют следующие типы:

1. *Монотонные*. Это специальный частный случай слоистых сетей с дополнительными условиями на связи и элементы. Каждый, слой, кроме последнего (выходного), разбит на два блока: возбуждающий (В) и тормозящий (Т). Связи между блоками тоже разделяются на тормозящие и возбуждающие. Если от блока  $A$  к блоку  $C$  ведут только возбуждающие связи, это означает, что любой выходной сигнал блока  $C$  является монотонной неубывающей функцией любого выходного сигнала блока  $A$ . Если же эти связи только тормозящие, то любой выходной сигнал блока  $C$  является не-возрастающей функцией любого

выходного сигнала блока *A*. Для элементов монотонных сетей необходима монотонная зависимость выходного сигнала элемента от параметров входных сигналов.

2. *Сети без обратных связей*. В таких сетях нейроны входного слоя получают входные сигналы, преобразуют их и передают нейронам 1-го скрытого слоя, далее срабатывает 1-й скрытый слой и т. д. до  $q$ -го слоя, который выдает выходные сигналы для интерпретатора и пользователя. Если не оговорено противное, то каждый выходной сигнал  $q$ -го слоя подается на вход всех нейронов  $(q+1)$ -го слоя; однако возможен вариант соединения  $q$ -го слоя с произвольным  $(q+r)$ -м слоем.

Следует отметить, что классическим вариантом слоистых сетей являются сети прямого распространения (рис. 2).

3. Сети с обратными связями. Это сети, у которых информация с последующих слоев передается на предыдущие.

В качестве примера сетей с обратными связями на рис. 3 представлены так называемые *частично-рекуррентные сети* Элмана и Жордана.

Известные сети можно разделить по принципу структуры нейронов на *гомогенные* (или *однородные*) и *гетерогенные*. Гомогенные сети состоят из нейронов одного типа с единой функцией активации. В гетерогенную сеть входят нейроны с различными функциями активации.

Важно отметить существование *бинарных* и *аналоговых* сетей. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический ноль («заторможенное» состояние) и логическая единица («возбужденное» состояние).

Еще одна классификация делит нейронных сетей на *асинхронные* и *синхронные*. В первом случае в каждый момент времени свое состояние меняет лишь один нейрон. Во втором – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Алгоритмически ход времени в нейронной сети задается итерационным выполнением над нейронами однотипных действий. Далее будут рассматриваться только синхронные НС.

Сети можно классифицировать также по числу слоев.

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуются НС. Чем сложнее НС, тем масштабнее задачи, подвластные ей.

Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные (на сегодняшний день) конфигурации. Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации. При этом он руководствуется несколькими основополагающими принципами:

- возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев;
- введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о так называемой динамической устойчивости сети;
- сложность алгоритмов функционирования сети (в том числе, на пример, введение нескольких типов синапсов – возбуждающих, тормозящих и др.) также способствует усилению мощи НС.

Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза НС сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора, хотя в литературе приведены доказательства того, что для любого алгоритма существует нейронная сеть, которая может его реализовать.

### **Биологические основы функционирования нейрона**

Тематика искусственных нейронных сетей относится к междисциплинарной сфере знаний, связанных с биокибернетикой, электроникой, прикладной математикой, статистикой, автоматикой и даже с медициной.

Искусственные нейронные сети возникли на основе знаний о функционировании нервной системы живых существ. Они представляют собой попытку использования процессов, происходящих в нервных системах, для выработки новых технологических решений.

Нервная клетка, сокращенно называемая *нейроном*, является основным элементом нервной системы. Изучение механизмов функционирования отдельных нейронов и их взаимодействия принципиально важно для познания протекающих в нервной системе процессов поиска, передачи и обработки информации. С этой точки зрения представляется необходимым построить и изучить модель биологического нейрона.

Как и у любой другой клетки, у нейрона имеется тело со стандартным набором органелл, называемое *сомой*, внутри которого располагается ядро. Из сомы нейрона выходят многочисленные отростки, играющие ключевую роль в его взаимодействии с другими нервными клетками. Можно выделить два типа отростков: многочисленные тонкие, густо ветвящиеся *дендриты* и более толстый, расщепляющийся на конце *аксон* (рис. 1).

Входные сигналы поступают в клетку через *синапсы*, тогда как выходной сигнал отводится аксоном через его многочисленные нервные окончания, называемые *коллатералами*. Коллатералы контактируют с сомой и дендритами других нейронов, образуя очередные синапсы. Очевидно, что синапсы, подключающие к клетке выходы других нейронов, могут находиться как на дендритах, так и непосредственно на теле клетки.

Передача сигналов внутри нервной системы - это очень сложный электрохимический процесс. С большим упрощением можно считать, что передача нервного импульса между двумя клетками основана на выделении особых химических веществ, называемых *нейромедиаторами*, которые формируются под влиянием поступающих от синапсов раздражителей. Эти вещества воздействуют на клеточную мембрану, вызывая изменение ее энергетического потенциала, причем величина этого изменения пропорциональна количеству нейромедиатора, попадающего на мембрану.

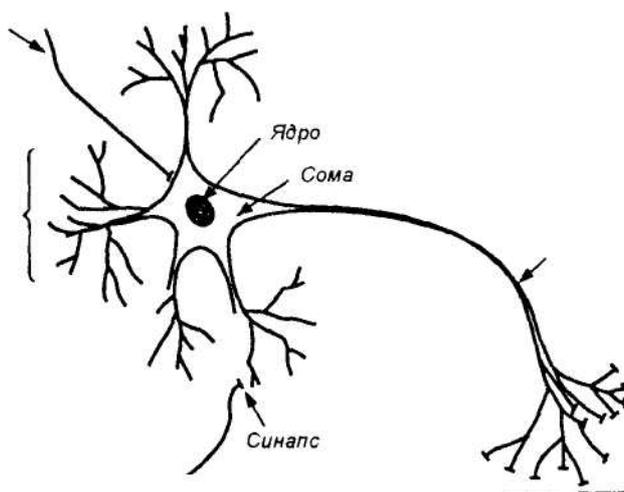


Рис. 1. Упрощенная структура биологической нервной клетки

Синапсы отличаются друг от друга размерами и возможностями концентрации нейромедиатора вблизи своей оболочки. По этой причине импульсы одинаковой величины, поступающие на входы нервной клетки через различные синапсы, могут возбуждать ее в разной степени. Мерой возбуждения клетки считается уровень поляризации ее мембраны, зависящий от суммарного количества нейромедиатора, выделенного на всех синапсах.

Каждому входу клетки можно сопоставить численные коэффициенты (веса), пропорциональные количеству нейромедиатора, однократно выделяемого на соответствующем синапсе. В математической модели нейрона входные сигналы должны умножаться на эти коэффициенты для того, чтобы корректно учитывать влияние каждого сигнала на состояние нервной клетки. Синаптические веса должны быть натуральными числами, принимающими как положительные, так и отрицательные значения. В первом случае синапс оказывает возбуждающее, а во втором - тормозящее действие, препятствующее возбуждению клетки другими сигналами. Таким образом, действие возбуждающего синапса может моделироваться положительным значением синаптического веса, а действие тормозящего синапса - отрицательным значением.

В результате поступления входных импульсов на конкретные синапсы и высвобождения соответствующих количеств нейромедиатора происходит определенное электрическое возбуждение нервной клетки. Если отклонение от состояния электрического равновесия невелико либо если баланс возбуждений и торможений является отрицательным, клетка самостоятельно возвращается в исходное состояние, и на ее выходе какие-либо изменения не регистрируются. В этом случае считается, что уровень возбуждения клетки был ниже порога ее срабатывания. Если же сумма возбуждений и торможений превысила порог активации клетки, значение выходного сигнала начинает лавинообразно нарастать, принимая характерный вид нервного импульса, пересылаемого аксоном на другие нейроны, подключенные к данной клетке. Величина этого сигнала не зависит от степени превышения порога. Клетка действует по принципу "все или ничего".

После выполнения своей функции нейромедиатор удаляется. Механизм удаления заключается либо во всасывании этой субстанции клеткой, либо в ее разложении, либо в удалении за пределы синапса.

Одновременно с генерацией нервного импульса в клетке запускается процесс рефракции. Он проявляется как стремительное возрастание порога активации клетки до значения "плюс бесконечность", в результате чего сразу после генерации импульса нейрон теряет способность вырабатывать очередной сигнал даже при сильном возбуждении. Такое состояние сохраняется в течение времени  $\Delta t_r$ , называемого периодом абсолютной рефракции. По окончании этого срока наступает период относительной рефракции  $\Delta t_w$ , за который порог срабатывания возвращается к первоначальному значению. В это время клетку можно активировать, но только с приложением более сильных возбуждений. В естественных процессах, как правило, выполняется отношение  $\Delta t_r \gg \Delta t_w$ .

Количество взаимодействующих друг с другом нервных клеток чрезвычайно велико. Считается, что человеческий мозг содержит около  $10^{11}$  нейронов, каждый из которых выполняет относительно примитивные функции суммирования весовых коэффициентов входных сигналов и сравнения полученной суммы с пороговым значением. Каждый нейрон имеет свои веса и свое пороговое значение. Они определяются местонахождением нейрона и решаемой им задачей и могут интерпретироваться аналогично содержанию локальной памяти процессора.

Громадное количество нейронов и межнейронных связей (до 1000 входов в каждый нейрон) приводит к тому, что ошибка в срабатывании отдельного нейрона остается незаметной в общей массе взаимодействующих клеток. Нейронная сеть проявляет высокую устойчивость к помехам - это "стабильная" сеть, в которой отдельные сбои не оказывают

существенного влияния на результаты ее функционирования. Таково главное отличие нейронных систем от обычных электронных систем, созданных человеком. Следует подчеркнуть, что ни одна современная технология не позволяет построить искусственную нейронную сеть, близкую по масштабам к нейронной сети мозга. Однако изучение и копирование биологических нервных систем позволяют надеяться на создание нового поколения электронных устройств, имеющих аналогичные характеристики.

Другая важная особенность нервных систем - высокая скорость их функционирования, несмотря на относительно длительный цикл срабатывания каждой отдельной клетки, измеряемый в миллисекундах и показанный на рис. 2. Она достигается благодаря параллельной обработке информации в мозге огромным количеством нейронов, соединенных многочисленными межнейронными связями. Такие операции, как распознавание образов и звуков либо принятие решений, выполняются человеческим мозгом за промежутки времени, измеряемые миллисекундами. Достижение такого результата при использовании полупроводниковой технологии VLSI все еще выходит за границы современных технических возможностей, хотя цикл срабатывания отдельных исполнительных элементов СБИС является достаточно коротким и имеет порядок  $10^{-8}$  с. Если удастся, взяв за образец нервную систему, создать устройство с высокой степенью параллельности выполнения независимых операций, то скорость его функционирования может быть существенно увеличена и приближена к уровню, наблюдаемому в процессах обработки информации биологическими объектами и могут интерпретироваться аналогично содержанию локальной памяти процессора.

Лекция № 2 *Тема: Модели нейронов и методы их обучения..*

Исходя из биологических основ функционирования нейрона каждый нейрон можно считать своеобразным процессором. С каждым процессором (т.е. обрабатывающим элементом сети) связывается набор входящих связей, по которым к данному элементу поступают сигналы от других элементов сети, и набор исходящих связей, по которым сигналы данного элемента передаются другим элементам. Некоторые элементы предназначены для получения сигналов из внешней среды (и поэтому называются входными элементами), а некоторые — для вывода во внешнюю среду результатов вычислений (и поэтому такие элементы сети называются выходными элементами). В случае программного моделирования реальных процессов на входные элементы обычно подаются уже предварительно подготовленные данные из некоторого файла данных, а не от непосредственно связанных с внешней средой датчиков.

Процессор суммирует с соответствующими весами сигналы, приходящие от других нейронов, выполняет нелинейную (например, пороговую) решающую функцию и передает результирующее значение связанным с ним нейронам. В соответствии с действующим правилом "все или ничего" в простейших моделях нейронов выходной сигнал принимает двоичные значения: 0 или 1. Значение 1 соответствует превышению порога возбуждения нейрона, а значение 0 — возбуждению ниже порогового уровня.

В одной из первых моделей нейрона, называемой моделью МакКаллока-Питса (предложенной в 1943 г.), нейрон считается бинарным элементом. Структурная схема этой модели представлена на рис. 1.

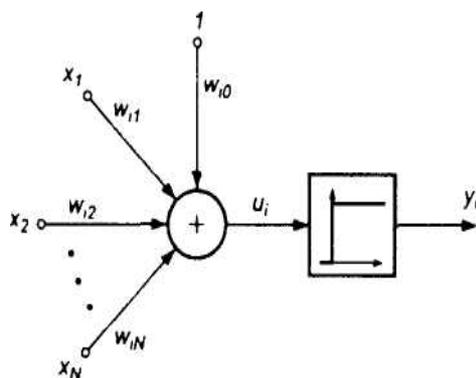


Рис. 1. Модель нервной клетки по МакКаллоку-Питсу

Входные сигналы  $x_j$  ( $j = 1, 2, \dots, N$ ) суммируются с учетом соответствующих весов  $w_{ij}$  (сигнал поступает в направлении от узла  $i$  к узлу  $j$ ) в сумматоре, после чего результат сравнивается с пороговым значением  $w_{i0}$ . Выходной сигнал нейрона  $y_i$ , определяется при этом зависимостью

$$y_i = f\left(\sum_{j=1}^N w_{ij}x_j(t) + w_{i0}\right). \quad (1)$$

Аргументом функции выступает суммарный сигнал  $u_i = \sum_{j=1}^N w_{ij}x_j(t) + w_{i0}$ . Функция  $f(u_i)$  называется функцией активации. В модели МакКаллока-Питса это пороговая функция вида

$$f(u) = \begin{cases} 1 & \text{для } u > 0 \\ 0 & \text{для } u \leq 0 \end{cases} \quad (2)$$

Коэффициенты  $w_{ij}$ , присутствующие в формуле (1), представляют веса синаптических связей. Положительное значение  $w_{ij}$  соответствует возбуждающим синапсам, отрицательное значение  $w_{ij}$  – тормозящим синапсам, тогда как  $w_{ij} = 0$  свидетельствует об отсутствии связи между  $i$ -м и  $j$ -м нейронами.

Модель МакКаллока-Питса — это дискретная модель, в которой состояние нейрона в момент  $(t + 1)$  рассчитывается по значениям его входных сигналов в предыдущий момент  $t$ . Построение дискретной модели обосновывается проявлением рефракции у биологических нейронов, приводящей к тому, что нейрон может изменять свое состояние с конечной частотой, причем длительность периодов бездействия зависит от частоты его срабатывания.

Через несколько лет Д. Хебб в процессе исследования ассоциативной памяти предложил теорию обучения (подбора весов  $w_{ij}$ ) нейронов. При этом он использовал наблюдение, что веса межнейронных соединений при активации нейронов могут возрастать. В модели Хебба приращение веса  $\Delta w_{ij}$  в процессе обучения пропорционально произведению выходных сигналов  $y_i$  и  $y_j$  нейронов, связанных весом  $w_{ij}$ :

$$w_{ij}(k+1) = w_{ij}(k) + \eta y_i(k)y_j(k), \quad (3)$$

где  $k$  означает номер цикла, а  $\eta$  – это коэффициент обучения.

В начале 60-х годов Б. Видроу предложил теоретическое обоснование и сформулировал принципы практической реализации адаптивных устройств обработки сигналов, что стало существенным вкладом в развитие нейронных сетей, функционирующих в режимах "онлайн" и "оффлайн".

В 1962 г. была опубликована книга Ф. Розенблатта, в которой представлена теория динамических нейронных систем для моделирования мозговой деятельности, основанная на персептронной модели нервной клетки. В этой теории использовалось представление нейрона моделью МакКаллока-Питса, в которой функция активации принимала двоичные значения 0 и 1.

Ограниченные возможности одиночного персептрона и составляемых из таких элементов одноуровневых сетей подверглись критике в книге М. Минского и С. Пейперта, что вызвало резкое снижение финансирования этой сферы научных исследований и привело в результате к замедлению развития искусственных нейронных сетей. Только отдельные научные группы, сконцентрированные вокруг таких ученых, как Гроссберг, Видроу, фон дер Мальсбург, Амари, Фукушима и Кохонен, продолжали работу в этой области. И только бурное развитие в 80-х годах технологии производства полупроводниковых устройств сверхвысокой степени интеграции

(VLSI) привело к резкому возрастанию интереса к средствам параллельной обработки информации, которыми считаются и искусственные нейронные сети. Начиная с опубликованных в 1982 г. работ Дж. Хопфилда, теория нейронных сетей развивается в стремительном темпе, а количество научных центров, занимающихся этой междисциплинарной сферой знаний, непрерывно увеличивается. Доработка или, точнее, повторное открытие принципа обратного распространения в применении к обучению многослойных сетей сняли те ограничения, которые стали главным объектом критики в книге М. Минского и С. Пейперта. Масштабное увеличение финансирования этой научной отрасли предопределило существенный прогресс как в теории, так и в практических приложениях. С учетом взрывного развития вычислительных систем это создало базу для реализации новых технологических решений в сфере технического распознавания образов, восприятия и объяснения, в управлении сложными системами, для обработки речевых сообщений и т.п. В настоящее время искусственные нейронные сети представляют собой высоко-развитую (особенно в теоретическом аспекте) отрасль знаний.

### **Правило вычисления сигнала активности**

Для всех элементов имеется правило вычисления выходного значения, которое предполагается передать другим элементам или во внешнюю среду (если речь идет о выходном элементе, представляющем конечный результат вычислений). Это правило называют функцией активности, а соответствующее выходное значение называют активностью соответствующего элемента. Активность может представляться либо некоторым действительным значением произвольного вида, либо действительным значением из некоторого ограниченного интервала значений (например, из интервала  $[0,1]$ ), или же некоторым значением из определенного дискретного набора значений (например,  $\{0, 1\}$  или  $\{+1,-1\}$ ). На вход функции активности поступает значение комбинированного ввода данного элемента. Примеры функций активности:

- 1) Тожественная функция
- 2) Пороговая функция
- 3) Сигмоидальная функция

Лекция № 3 **Тема: Однонаправленные многослойные сети.**

В соответствии с принципами функционирования биологических нейронов созданы различные математические модели, которыми в большей или меньшей степени реализуются свойства природной нервной клетки. Обобщенная схема, составляющая основу большинства таких моделей, восходит к представленной модели МакКаллока-Питса. Свойства нелинейной функции, особенно ее непрерывность, оказывают определяющее влияние на выбор способа обучения нейрона (подбор весовых коэффициентов). Другим важным фактором становится выбор стратегии обучения. Можно выделить два подхода: *обучение с учителем* и *обучение без учителя*.

При *обучении с учителем* предполагается, что, помимо входных сигналов, составляющих вектор  $x$ , известны также и ожидаемые выходные сигналы нейрона  $d_i$ , составляющие вектор  $d$ . В подобной ситуации подбор весовых коэффициентов должен быть организован так, чтобы фактические выходные сигналы нейрона  $y_i$  принимали бы значения, как можно более близкие к ожидаемым значениям  $d_i$ . Ключевым элементом процесса обучения с учителем является знание ожидаемых значений  $d_i$ , выходного сигнала нейрона.

Если такой подход невозможен, остается выбрать стратегию обучения без учителя. Подбор весовых коэффициентов в этом случае проводится на основании либо конкуренции нейронов между собой (стратегии "*Winner Takes All - WTA*" (Победитель получает все) или "*Winner Takes Most - WTM*" (Победитель получает больше), либо с учетом корреляции обучающих и выходных сигналов (обучение по Хеббу). При *обучении без учителя* на этапе адаптации нейрона мы не можем прогнозировать его выходные сигналы, тогда как при обучении с учителем результат обучения предопределен заранее благодаря априори заданным обучающим выборкам.

- 1) **Многослойный персептрон**

Простой персептрон - это обычная модель МакКаллока-Питса с соответст-

вующей стратегией обучения.

Весовые коэффициенты входов сумматора, на которые поступают входные сигналы  $x_j$ , обозначаются  $w_{ij}$ , а пороговое значение, поступающее с так называемого поляризатора, –  $w_{i0}$ . Нелинейная функция активации персептрона представляет собой дискретную функцию ступенчатого типа, вследствие чего выходной сигнал нейрона может принимать только два значения - 0 или 1 в соответствии с правилом

$$y_i(u_i) = \begin{cases} 1 & \text{для } u \geq 0 \\ 0 & \text{для } u < 0 \end{cases} \quad (1)$$

где  $u_i$  обозначен выходной сигнал сумматора

$$u_i = \sum_{j=0}^N w_{ij} x_j \quad (2)$$

В приведенной формуле подразумевается, что имеющий длину  $N$  вектор  $x$  дополнен нулевым членом  $x_0 = 1$ , формирующим сигнал поляризации,  $x = [x_0, x_1, \dots, x_N]$ . Обучение персептрона требует наличия учителя и состоит в таком подборе весов  $w_{ij}$ , чтобы выходной сигнал  $y_i$  был наиболее близок к заданному значению  $d_i$ . Это обучение гетероассоциативного типа, при котором каждой обучающей выборке, представляемой вектором  $x$  априори поставлено в соответствие ожидаемое значение  $d_i$  на выходе  $i$ -го нейрона.

## 2) Сигмоидальный нейрон

Нейрон сигмоидального типа (рис. 2) имеет структуру, подобную модели МакКаллока-Питса, с той разницей, что функция активации является непрерывной и может быть выражена в виде сигмоидальной униполярной или биполярной функции. Униполярная функция, как правило, представляется формулой

$$f(x) = \frac{1}{1 + e^{-\beta x}}, \quad (7)$$

тогда как биполярная функция задается в виде

$$f(x) = th(\beta x) = \tanh(\beta x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}} \quad (8)$$

В этих формулах параметр  $\beta$  подбирается пользователем. Его значение влияет на форму функции активации. На рис. 3 представлены графики сигмоидальной функции от переменной  $x$  для различных значений  $\beta$

При малых величинах  $\beta$  график функции достаточно пологий, но по мере роста значения  $\beta$  крутизна графика увеличивается. При  $\beta \rightarrow \infty$  сигмоидальная функция превращается в функцию ступенчатого типа, идентичную функции активации персептрона. На практике чаще всего для упрощения используется значение  $\beta = 1$ .

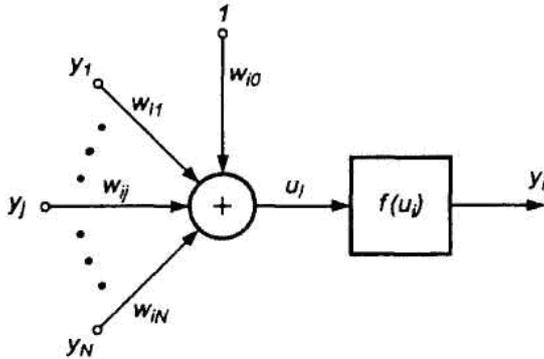
## 3) Модель нейрона Хебба. Сеть прямого распространения сигнала.

Д. Хебб в процессе исследования нервных клеток заметил, что связь между двумя клетками усиливается, если обе клетки пробуждаются (становятся активными) в один и тот же момент времени. Если  $j$ -я клетка с выходным сигналом  $y_j$  связана с  $i$ -й клеткой, имеющей выходной сигнал  $y_i$ , связью с весом  $w_{ij}$ , то на силу связи этих клеток влияют значения выходных сигналов  $y_i$  и  $y_j$ .

Д. Хебб предложил формальное правило, в котором отразились результаты его наблюдений. В соответствии с правилом Хебба, вес  $w_{ij}$  нейрона изменяется пропорционально произведению его входного и выходного сигналов

$$\Delta w_{ij} = \eta y_j y_i, \quad (31)$$

где  $\eta$  - это коэффициент обучения, значение которого выбирается в интервале (0,1). Правило Хебба может применяться для нейронных сетей различных типов с разнообразными функциями активации моделей отдельных нейронов.



Структурная схема нейрона Хебба, представленная на рис. 11, соответствует стандартной форме модели нейрона. Связь с весом  $w_{ij}$ , способ подбора значения которого задается отношением (31), соединяет входной сигнал  $y_j$  с сумматором  $i$ -го нейрона, вырабатывающего выходной сигнал  $y_i$ . Обучение нейрона по правилу Хебба может проводиться как с учителем, так и без него. Во

втором случае в правиле Хебба используется фактическое значение  $y_i$  выходного сигнала нейрона. При обучении с учителем вместо значения выходного сигнала  $y_i$  используется ожидаемая от этого нейрона реакция  $d_i$ . В этом случае правило Хебба записывается в виде

$$\Delta w_{ij} = \eta y_j d_i. \quad (32)$$

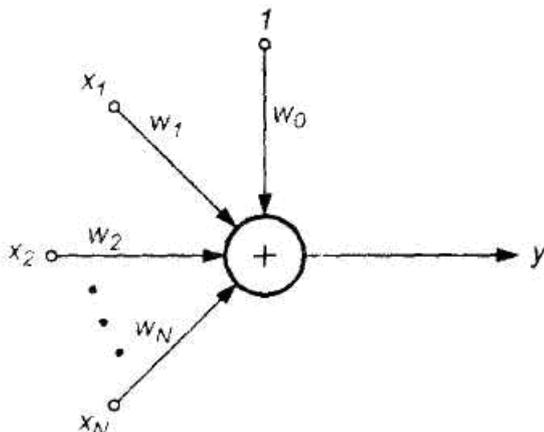
Правило Хебба характеризуется тем, что в результате его применения веса могут принимать произвольно большие значения, поскольку в каждом цикле обучения происходит суммирование текущего значения веса и его приращения  $\Delta w_{ij}$ :

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}. \quad (33)$$

Один из способов стабилизации процесса обучения по правилу Хебба состоит в учете для уточнения веса последнего значения  $w_{ij}$ , уменьшенного на коэффициент забывания  $\gamma$ . При этом правило Хебба представляется в виде

$$w_{ij}(t+1) = w_{ij}(t)(1 - \gamma) + \Delta w_{ij} \quad (34)$$

Значение коэффициента забывания  $\gamma$  выбирается, как правило, из интервала (0, 1) и чаще всего составляет некоторый процент от коэффициента обучения  $\eta$ . Применение больших значений  $\gamma$  приводит к тому, что нейрон забывает значительную часть того, чему он обучился в прошлом. Рекомендуемые значения коэффициента забывания  $\gamma < 0,1$ , при которых нейрон сохраняет большую часть информации, накопленной в процессе обучения, и получает возможность стабилизировать значения весов на определенном уровне.



При обучении линейного нейрона по правилу Хебба стабилизация не происходит даже при вводе коэффициента забывания. Выходной сигнал нейрона, структурная схема которого приведена на рис. 13, определяется выражением

$$y = \sum_j w_j x_j = w^T x = x^T w \quad (35)$$

Рис. 12. Модель линейного нейрона Хебба

Если согласно правилу Хебба

$$\Delta w = \eta x y \quad (36)$$

подставить выражение (35) в формулу (36) и выбрать для упрощения  $\eta = 1$ , то получим приращение вектора весов  $\Delta w$  в виде

$$\Delta w = C w, \quad (37)$$

где  $C = x x^T$  - это матрица корреляции, которая по определению является симметричной и положительно полуопределенной и, следовательно, имеет собственные натуральные и неотрицательные значения. При выполнении операций, описываемых зависимостью (37) и повторяемых на положительно полуопределенной матрице  $C$ , процесс становится расходящимся, а значения компонентов вектора  $w$  стремятся к бесконечности.

Нестабильность правила Хебба в процессе обучения можно устранить ограничением вектора весов за счет операции ренормализации, т.е. таким подбором пропорционального коэффициента  $\alpha$  на каждом шаге обучения, чтобы  $w' = \alpha w$  при  $\|w'\| = 1$ . Этот метод достаточно сложен и требует дополнительных трудозатрат на этапе обучения.

Е. Ойя модифицировал правило Хебба таким образом, что и без ренормализации процесса обучения вектор весов самостоятельно стремится к  $\|w\| = 1$ . В соответствии с правилом Ойи уточнение весов производится согласно выражению

$$\Delta w = \eta y (x_i - w_i). \quad (38)$$

Это правило напоминает обратное распространение, поскольку сигнал  $x_i$  модифицируется обратным сигналом, связанным с выходным сигналом  $y$  нейрона. Для каждого отдельно взятого нейрона правило Ойя может считаться локальным, так как в процессе модификации  $x_i$  принимается во внимание только тот весовой коэффициент, значение которого подбирается в текущий момент времени.

Доказательство ограниченности весов, уточняемых по правилу Ойя, можно получить, заменяя скалярное выражение (38) векторной формой, которая с учетом упрощения  $\eta = 1$  и в соответствии с (38) приобретает вид:

$$\Delta w = C w - (w^T C w) w \quad (39)$$

Стабильность процесса обучения достигается, когда при достаточно длительном обучении обеспечивается  $\|\Delta w\| = 0$ , т.е.

$$C w = (w^T C w) w. \quad (40)$$

Если собственное значение корреляционной матрицы  $C$  обозначить  $\lambda$ , а вектор  $w$  подбирать как связанный с ней собственный вектор, то по определению собственного значения имеем  $C w = \lambda w$ . Подставляя это выражение в формулу (39), получаем:

$$\lambda = w^T C w = w^T \lambda w = \lambda |w|^2. \quad (41)$$

Из (41) следует, что применение для обучения модифицированного правила Хебба приводит к ограничению модуля вектора  $w$  единицей  $|w| = 1$ , обеспечивающему ограниченность значений весовых коэффициентов.

**Лекция № 4 Тема: Проблемы практического использования искусственных нейронных сетей и их свойства**

Рассмотрим случай запоминаемых пар образцов. Идея заключается в том, чтобы выбрать нужный образец из памяти, даже если у нас нет всей необходимой информации для начала поиска сохраненного образца. Например, вы хотите найти книгу в библиотеке, но не помните ее названия. При этом если вы знаете имя автора и описание того, чему книга посвящена, этого уже достаточно (с большой долей уверенности!), чтобы найти ассоциируемый с этой информацией объект.

Когда сохраняемая в памяти пара ассоциируемых образцов создается одинаковыми образцами, память называется автоассоциативной, а если образцы являются разными, то память называется гетероассоциативной. В этой главе будут рассмотрены три модели нейронных сетей для автоассоциации образцов.

### *Дискретная сеть Хопфилда*

Сеть Хопфилда (Hopfield) является автоассоциативной сетью, ведущей себя подобно памяти, которая может вспомнить сохраненный образец даже по подсказке (в виде вводимых данных), представляющей собой искаженную помехами версию нужного образца. Например, сеть может сохранить набор изображений букв, а когда сети будет представлена искаженная версия сохраненного символа, сеть должна оказаться способной найти истинный экземпляр. Дискретная сеть Хопфилда имеет следующие характеристики.

- Один слой элементов (входные элементы, представляющие входной образец, не учитываются).
- Каждый элемент связывается со всеми другими элементами, но элемент не связывается с самим собой.
- За один шаг обновляется только один элемент, в отличие, например, от сети с обратным распространением ошибок, где все элементы слоя могут изменяться одновременно, если сеть реализована в виде аппаратных средств с соответствующими параллельными возможностями.
- Элементы обновляются в случайном порядке, но в среднем каждый элемент должен обновляться в одной и той же мере. Например, в случае сети из 10 элементов после 100 обновлений каждый элемент должен обновиться приблизительно 10 раз.
- Вывод элемента ограничен значениями 0 или 1.

Сеть Хопфилда является рекуррентной в том смысле, что для каждого входного образца выход сети повторно используется в качестве ввода до тех пор, пока не будет достигнуто устойчивое состояние. Пример сети Хопфилда показан на рис. 1. Удобно считать, что сеть Хопфилда не имеет входных элементов, так как входной вектор просто определяет начальные значения активности элементов. Например, если ввод является двоичным, то входной вектор  $[1 \ 1 \ 0 \ 1]$  означает, что значения активности для элементов  $\{1, 2, 4\}$  будут равны 1, а для элемента  $\{3\}$  активность будет равна 0. Элемент обновляется тогда, когда все элементы передадут свои значения активности по имеющимся взвешенным связям, после чего вычисляется сумма произведений (т.е. берется скалярное произведение). Значение активности элемента получается на основе использования некоторого правила активизации. Каждый элемент сети Хопфилда имеет состояние, характеризующееся значением активности, которое должен посылать данный элемент другим элементам, а состояние сети в любой момент времени задается вектором состояний всех ее элементов.

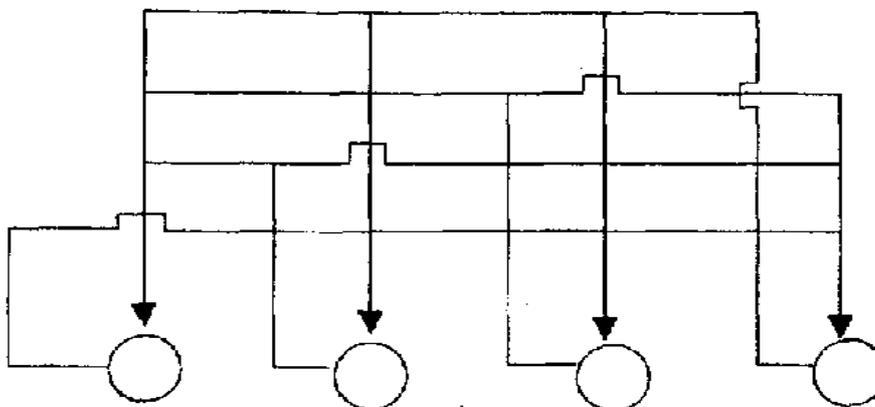


Рис. 1. Сеть Хопфилда с четырьмя элементами. Для каждого элемента входного вектора имеется свой элемент сети. Элементы сети связаны со всеми остальными ее элементами, но не сами с собой. Связи являются двунаправленными

В качестве входных данных сети Хопфилда можно использовать двоичные, но здесь мы будем использовать +1 для обозначения состояния "включено" и -1 — для состояния "выключено". Комбинированный ввод элемента вычисляется по формуле:

$$net_i = \sum_{j=1}^n s_j w_{ij},$$

где  $s_j$  обозначает состояние элемента с номером  $j$ . Когда элемент обновляется, его состояние изменяется в соответствии с правилом

$$s_j = \begin{cases} +1, & \text{если } net_j > 0 \\ -1, & \text{если } net_j < 0 \end{cases}$$

Эта зависимость называется сигнум-функцией и в более краткой форме она записывается в виде

$$s_j = \text{sgn}(net_j).$$

Если комбинированный ввод оказывается равным нулю, то элемент остается в состоянии, в котором он пребывал перед обновлением.

Сеть работает очень просто. Входной вектор задает начальные состояния всех элементов. Элемент для обновления выбирается случайным образом. Выбранный элемент получает взвешенные сигналы от всех остальных элементов и изменяет свое состояние. Выбирается другой элемент, и процесс повторяется. Сеть достигает предела, когда ни один из ее элементов, будучи выбранным для обновления, не меняет своего состояния.

Весовые значения для сети Хопфилда определяются непосредственно из учебных данных без необходимости проведения обучения в более привычном смысле. Сеть Хопфилда ведет себя как память, и процедура сохранения отдельного вектора представляет собой вычисление прямого произведения вектора с ним самим. В результате этой процедуры создается матрица, задающая весовые значения для сети Хопфилда, в которой все диагональные элементы должны быть установлены равными нулю (поскольку диагональные элементы задают автосвязи элементов, а элементы сами с собой не связаны). Таким образом, весовая матрица, соответствующая сохранению вектора  $x$ , задается формулой  $W = x^T x$ .

#### **Двунаправленная ассоциативная память**

Сетью, имеющей много общего с сетью Хопфилда, является двунаправленная ассоциативная память (сеть ВАМ — Bidirectional Associate Memory), предложенная Коско. Сеть ВАМ является гетероассоциативной рекуррентной сетью. Сеть сохраняет пары образцов и может восстановить образец, когда ассоциированный с ним образец предлагается ей в качестве подсказки. В этой сети два слоя элементов — по одному для каждого из образцов пары — и оба слоя соединяются двунаправленными связями (т.е. активность может передаваться по связям в обоих направлениях) (рис. 2).

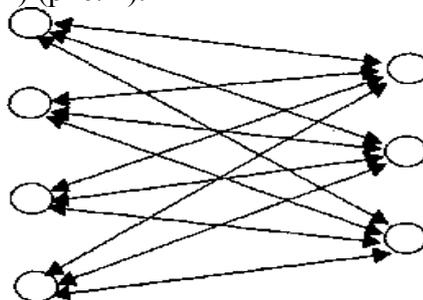


Рис. 2 Двунаправленная ассоциативная память. Элементы слева представляют образцы размерности 4, а элементы справа - ассоциированные с ними образцы размерности 3

Рассмотрим только дискретную биполярную сеть ВАМ, но можно рассмотреть и непрерывные значения. Чтобы сохранить образец  $s$  и ассоциируемый с ним образец  $t$ , рассматривается прямое произведение, определяющее весовые значения. Процедура точно такая же, как и в сети Хопфилда, но теперь матрица уже не обязана быть квадратной, а диагональные элементы не обнуляются. Весовой матрицей для одной пары будет матрица

$$W = s^T t.$$

Чтобы сохранить несколько пар, все соответствующие произведения, определяющие весовые значения, складываются, точно так же, как это делается для сети Хопфилда.

Процедура нахождения в памяти элемента подобна соответствующей процедуре сети Хопфилда. В следующем описании  $i$  обозначает один слой элементов, а  $j$  — ассоциированный слой элементов.

- Устанавливаются значения активности элементов слоя  $i$  в соответствии со значениями, задаваемыми входным образцом.

- Распространяется активность на слой  $j$ . Комбинированный ввод

элемента слоя  $j$  равен  $u_j = \sum_{i=1}^n s_i w_{ij}$

- Вычисляется новое состояние для каждого элемента слоя  $j$ :

$$t_j = f(u_j).$$

- Распространяется активность на слой  $i$ . Комбинированный ввод элемента слоя  $i$  равен:  $s_i = f(u_i)$ .

- Это двустороннее распространение сигналов активности повторяется до тех пор, пока не будет достигнуто устойчивое состояние. Активность для каждого слоя определяется относительно некоторой пороговой величины  $\theta$ .

$$t_j = f(u_j) = \begin{cases} 1, & \text{если } u_j > \theta_j \\ t_j, & \text{если } u_j = \theta_j \\ -1, & \text{если } u_j < \theta_j \end{cases} \quad s_i = f(u_i) = \begin{cases} 1, & \text{если } u_i > \theta_i \\ s_i, & \text{если } u_i = \theta_i \\ -1, & \text{если } u_i < \theta_i \end{cases}$$

Все элементы сети сначала имеют нулевые значения активности. Обратите внимание на то, что распространение может начаться с любого уровня, так как и  $s$  может использоваться для вызова  $t$ , и, наоборот,  $t$  может использоваться для вызова  $s$ .

**Автоассоциативное обратное распространение ошибок**

Стандартная сеть с прямой связью и обратным распространением ошибок может быть обучена автоассоциативным методом выполнению задач типа сжатия изображений. Этот метод дает возможность сделать целевой образец таким же, как и учебный образец, так что сеть учится воспроизводить в выходном слое то, что подается ей на рассмотрение во входном слое. Базовая архитектура сети показана на рис. 4.

Сеть обучается стандартным образом, но целью обучения является ассоциация каждого учебного образца с самим собой. После успешного окончания обучения сеть может работать как два механизма: первый слой весов может обеспечивать сжатие образца, а второй слой весов может восстанавливать полный образец из его сжатого представления. Конечно, в результате реконструкции могут быть и потери информации, поскольку сеть не

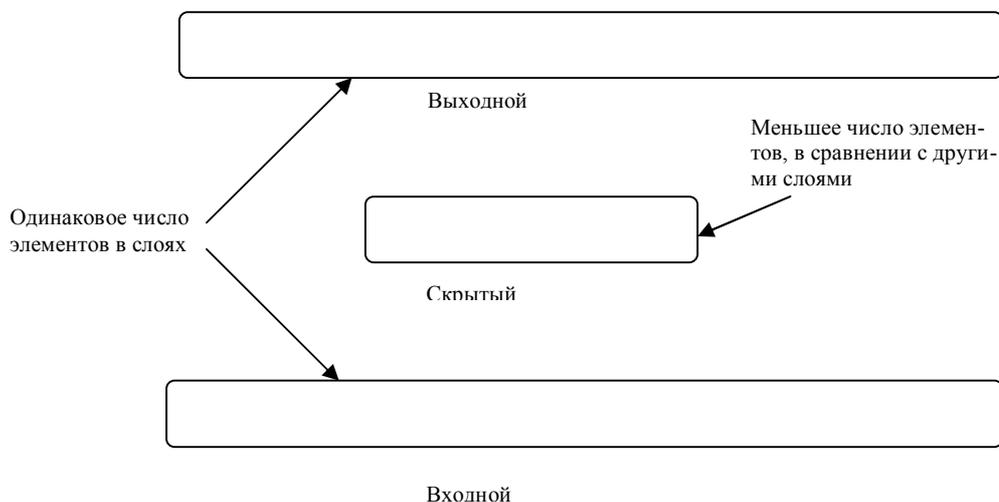


Рис. 4. Автоассоциативная сеть с прямой связью

будет в совершенстве создавать на выходном уровне все учебные экземпляры. Образец сжимается после его подачи на вход сети и распространения сигналов, по которым вычисляются значения активности скрытых элементов. Для сети с архитектурой типа 20-10-20 (20 входных элементов, 10 скрытых элементов и 20 выходных элементов) отношением сжатия будет 2-1, поскольку для данного входного вектора сжатый вектор задается значениями активности скрытых элементов. Чтобы восстановить входной вектор, сжатый вектор предъявляется скрытому слою, и активность распространяется на выходной слой.

Иногда может оказаться выгодным обработать данные с помощью автоассоциативной сети перед тем, как рассмотреть их с помощью другой сетевой модели (и даже перед тем, как выполнить их классификацию с помощью другой сети с прямой связью). Известно, например, что автоассоциативная сеть с прямой связью, имеющая один скрытый слой, вычисляет по сути главные компоненты. Анализ главных компонент является основным методом статистической обработки данных и используется для того, чтобы удалить избыточность данных и провести кластеризацию. В результате анализа главных компонент происходит также декорреляция признаков векторов (признаки характеризуют положение элементов), что иногда оказывается полезным при обучении другой сети с прямой связью. Идея заключается в преобразовании учебных данных в некоторое сокращенное описание, где компонентами такого сокращенного описания выступают скрытые элементы.

Анализ главных компонент можно использовать для выполнения сжатия с потерями. Трансформация двумерных образцов в одномерные происходит следующим образом:

$$\begin{bmatrix} 1 & 1 \\ 3 & 3 \\ 4 & 4 \\ 8 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \end{bmatrix} = [1,414 \quad 4,243 \quad 5,657 \quad 11,314].$$

Обратное отображение для 1.414 задается с помощью следующего выражения:

$$[1,414] \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} = [1 \quad 1].$$

Вывод: Ассоциативная сеть действует подобно памяти.

- Автоассоциация связывает образец с ним самим. Например, автоассоциативная память может использоваться для того, чтобы восстановить истинную версию сохраненного образца по искаженной версии этого образца.
- Гетероассоциация представляет собой связывание двух разных образцов. Один образец может использоваться в качестве подсказки, по которой из памяти извлекается другой образец.
- Когда автоассоциативное обучение происходит путем пропускания сигналов через узкий канал типа скрытого слоя автоассоциативной сети с прямой связью, происходит в некотором смысле сжатие данных. Сжатие может оказаться полезной формой предварительной обработки данных перед использованием их для обучения сети с другой архитектурой.
- Сеть Хопфилда может использоваться для автоассоциации, а двунаправленная ассоциативная память — для гетероассоциации.
- Весовые значения дискретной сети Хопфилда и двунаправленной ассоциативной памяти могут быть найдены с помощью простых матричных вычислений, поэтому такие сети не требуют длительного обучения.
- От числа элементов в ассоциативной сети зависит число образцов, которые могут быть сохранены.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

*Лабораторная работа № 1*

### Тема: Создание однонаправленной сети

Цель занятия – продемонстрировать основные этапы реализации нейронно- сетевого подхода для решения конкретной задачи. Можно выделить 4 основных этапа:

1. Подготовка данных для тренировки сети.
2. Создание сети.
3. Обучение сети.
4. Тестирование сети.
5. Моделирование сети. (Использование сети для решения поставленной задачи.)

В качестве примера рассмотрим следующую задачу:

Задан массив, состоящий из нескольких значений функции  $y = Ce^{-\frac{(x-A)^2}{S}}$  ( $S > 0$ ) на интервале (0,1). Создать нейронную сеть такую, что при вводе этих значений на входы сети на выходах получались бы значения параметров C, A и S.

1. Подготовка данных для обучения сети

В первую очередь необходимо определиться с размерностью входного массива. Выберем количество значений функции равным  $N=21$ , т.е. в качестве входных векторов массива используем значения функции  $y$  в точках  $x=0.05; \dots 1.0$ . Для обучения сети необходимо сформировать массив входных векторов для различных наборов параметров C, A и S. Каждый набор этих параметров является вектором-эталоном для соответствующего входного вектора.

Для подготовки входного и эталонного массивов воспользуемся следующим алгоритмом. Выбираем случайным образом значения компонент вектора – эталона C, A S и вычисляем компоненты соответствующего входного вектора. Повторяем эту процедуру M раз и получаем массив входных векторов в виде матрицы размерностью  $N \times M$  и массив векторов – эталонов в виде матрицы размерностью в нашем случае  $3 \times M$ . Полученные массивы мы можем использовать для обучения сети.

Прежде чем приступить к формированию обучающих массивов необходимо определиться с некоторыми свойствами массивов.

1. Диапазон изменения параметров C, A S. Выберем диапазоны изменения параметров C, A, S равными (0.1, 1). Значения, близкие к 0 и сам 0 исключим в связи с тем, что функция не определена при  $S=0$ . Второе ограничение связано с тем, что при использовании типичных передаточных функций желательно, чтобы компоненты входных и выходных векторов не выходили за пределы диапазона (-1,1). В дальнейшем мы познакомимся с методами нормировки, которые позволяют обойти это ограничение.

2. Количество входных и эталонных векторов выберем равным  $M=100$ . Этого достаточно для обучения а процесс обучения не займет много времени.

Тестовые массивы и эталоны подготовим с помощью программы *mas1*:

```
% формирование входных массивов (входной массив P) и (эталонны T)
P=zeros(100,21);
T=zeros(3,100);
x=0:5.e-2:1;
for i=1:100
c=0.9*rand+0.1;
a=0.9*rand+0.1;
s=0.9*rand+0.1;
T(1,i)=c;
T(2,i)=a;
T(3,i)=s;
P(i,:)=c*exp(-((x-a).^2/s));
end;
P=P';
```

С помощью этой программы формируется матрица P из  $M=100$  входных векторов-столбцов, каждый из которых сформирован из 21 точки исходной функции со случайно выбранными значениями параметров C, A, S, и матрица T эталонов из 100 эталонных векторов-столбцов, каждый из которых сформирован из 3 соответствующих эталонных значений.

Матрицы P и T будут использованы при обучении сети. Следует отметить, что при каждом новом запуске этой программы будут формироваться массивы с новыми значениями компонентов векторов, как входных, так и эталонных.

## 2. Создание сети

Вообще, выбор архитектуры сети для решения конкретной задачи основывается на опыте разработчика. Поэтому предложенная ниже архитектура сети является одним вариантом из множества возможных конфигураций. Для решения поставленной задачи сформируем трехслойную сеть обратного распространения, включающую 21 нейрон во входном слое (по числу компонент входного вектора) с передаточной функцией `logsig`, 15 нейронов во втором слое с передаточной функцией `logsig` и 3 нейрона в выходном слое (по числу компонент выходного вектора) с передаточной функцией `purelin`. При этом в качестве обучающего алгоритма выбран алгоритм `Levenberg-Marquardt (trainlm)`. Этот алгоритм обеспечивает быстрое обучение, но требует много ресурсов. В случае, если для реализации этого алгоритма не хватит оперативной памяти, можно использовать другие алгоритмы (`trainbfg`, `trainrp`, `trainscg`, `traincgb`, `traincgf`, `traincgp`, `trainoss`, `traingdx`). По умолчанию используется `trainlm`. Указанная сеть формируется с помощью процедуры:

```
net=newff(minmax(P),[21,15,3],{'logsig' 'logsig' 'purelin'},'trainlm');
```

Первый аргумент - матрица Mx2 минимальных и максимальных значений компонент входных векторов вычисляется с помощью процедуры `minmax`. Результатом выполнения процедуры `newff` является объект – нейронная сеть `net` заданной конфигурации. Сеть можно сохранить на диске в виде `mat`. файла с помощью команды `save` и загрузить снова с помощью команды `load`. Более подробную информацию о процедуре можно получить, воспользовавшись командой `help`.

## 3. Обучение сети

Следующий шаг – обучение созданной сети. Перед обучением необходимо задать параметры обучения. Задаем функцию оценки функционирования `sse`.

```
net.performFcn='sse';
```

В этом случае в качестве оценки вычисляется сумма квадратичных отклонений выходов сети от эталонов. Задаем критерий окончания обучения – значение отклонения, при котором обучение будет считаться законченным:

```
net.trainParam.goal=0.01;
```

Задаем максимальное количество циклов обучения. После того, как будет выполнено это количество циклов, обучение будет завершено:

```
net.trainParam.epochs=1000;
```

Теперь можно начинать обучение:

```
[net,tr]=train(net,P,T);
```

Таким образом, обучение сети окончено. Теперь эту сеть можно сохранить в файле `nn1.mat`:

```
save nn1 net;
```

## 4. Тестирование сети

Перед тем, как воспользоваться нейронной сетью, необходимо исследовать степень достоверности результатов вычислений сети на тестовом массиве входных векторов. В качестве тестового массива необходимо использовать массив, компоненты которого отличаются от компонентов массива, использованного для обучения. В нашем случае для получения тестового массива достаточно воспользоваться еще раз программой `mas1`.

Для оценки достоверности результатов работы сети можно воспользоваться результатами регрессионного анализа, полученными при сравнении эталонных значений со значениями, полученными на выходе сети когда на вход поданы входные векторы тестового массива. В среде MATLAB для этого можно воспользоваться функцией `postreg`. Следующий набор команд иллюстрирует описанную процедуру:

```
>> mas1 - создание тестового массива P
```

```
>> u=sim(net,P); - обработка тестового массива
```

>> [m,b,r]=postreg(y(1,:),T(1,:)); - регрессионный анализ результатов обработки.

Сравнение компонентов С эталонных векторов с соответствующими компонентами выходных векторов сети. Если все точки легли на прямую, то это говорит о правильной работе сети на тестовом массиве.

>> [m,b,r]=postreg(y(2,:),T(2,:));

>> [m,b,r]=postreg(y(3,:),T(3,:));

Сохраним обученную сеть net на диске в файл nn1.mat save nn1 net

5. Моделирование сети. (Использование сети для решения поставленной задачи)

Для того, чтобы применить обученную сеть для обработки данных, необходимо воспользоваться функцией sim:

p=p';

Y=sim(net,p);

где p – набор входных векторов, Y – результат анализа в виде набора выходных векторов.

Например, пусть C=0.2, A=0.8, S=0.7,

Тогда

$p=0.2*\exp(-((x-0.8).*2/0.7));$

Подставив этот входной вектор в качестве аргумента функции sim :

Y=sim(net,p)

Получим:

Y=0.2048 (C), 0.8150 (A), 0.7048 (S) >>

Что весьма близко к правильному результату (0.2; 0.8; 0.7).

*Лабораторная работа № 2*

### **Тема: Сети кластеризации и классификации данных**

В процессе анализа больших информационных массивов данных неизменно возникают задачи, связанные с исследованием топологической структуры данных, их объединением в группы (кластеры), распределением по классам и т. п. Это могут быть экономические, финансовые, научно-технические, медицинские и другие приложения, где требуется решение таких практических задач, как сжатие данных, их хранение и поиск, определение характеристик объекта по ограниченному набору признаков. Такие задачи могут быть успешно решены с применением специального класса самоорганизующихся нейронных сетей.

#### **Самоорганизующиеся нейронные сети**

Свойство самоорганизации является одним из наиболее привлекательных свойств нейронных сетей. Таким свойством обладают самоорганизующиеся нейронные сети, описанные финским ученым Т. Кохоненом. Нейроны самоорганизующейся сети могут быть обучены выявлению групп (кластеров) векторов входа, обладающих некоторыми общими свойствами. При изучении самоорганизующихся нейронных сетей, или *сетей Мошна*, существенно различать сети с неупорядоченными нейронами, которые часто называют *слоями Кохонена*, и сети с упорядочением нейронов, которые часто называют *картами Кохонена*. Последние отражают структуру данных таким образом, что близким кластерам данных на карте соответствуют близко расположенные нейроны.

Для создания самоорганизующихся нейронных сетей, являющихся слоем или картой Кохонена, предназначены М-функции newsc и newsom соответственно.

По команде help selforg можно получить следующую информацию об М-функциях, входящих в состав ППП Neural Network Toolbox и относящихся к построению сетей Кохонена:

<i>Self-organizing networks</i>	<i>Самоорганизующиеся сети</i>
<b>New networks</b>	<b>Формирование сети</b>
newsc newsom	Создание слоя Кохонена <b>Создание карты Кохонена</b>
<b>Using networks</b>	<b>Работа с сетью</b>

sim init adapt train	Моделирование Инициализация Адаптация Обучение
<b>Weight functions</b>	<b>Функции расстояния и взвешивания</b>
negdist	Отрицательное евклидово расстояние
<b>Net input functions</b>	<b>Функции накопления</b>
netsum	Сумма взвешенных входов
<b>Transfer functions</b>	<b>Функции активации</b>
compet	Конкурирующая функция активации
<b>Topology functions</b>	<b>Функции описания топологии сети</b>
gridtop hextop randtop	Прямоугольная сетка Гексагональная сетка <b>Сетка со случайно распределенными узлами</b>
<b>Distance functions</b>	<b>Функции расстояния</b>
dist boxdist mandist linkdist	Евклидово расстояние Расстояние максимального координатного смещения Расстояние суммарного координатного смещения Расстояние связи
<b>Initialization functions</b>	<b>Функции инициализации сети</b>
initlay initwb initcon midpoint	Послойная инициализация Инициализация весов и смещений Инициализация смещений с учетом чувствительности нейронов Инициализация весов по правилу средней точки
<b>Learning functions</b>	<b>Функции настройки параметров</b>
learnk learncon learnsom	Правило настройки весов для слоя Кохонена Правило настройки смещений для слоя Кохонена Правило настройки весов карты Кохонена
<b>Adapt functions</b>	<b>Функции адаптации</b>
adaptwb	Адаптация весов и смещений
<b>Training functions</b>	<b>Функции обучения</b>
trainwb1	Повекторное обучение весов и смещений
<b>Demonstrations</b>	<b>Демонстрационные примеры</b>
democ1 demosm1 demosm2	Настройка слоя Кохонена Одномерная карта Кохонена Двумерная карта Кохонена

### **Слой Кохонена**

Рассмотрим самоорганизующуюся нейронную сеть с единственным слоем, задача которой заключается в том, чтобы правильно сгруппировать (кластеризировать) поступающие на нее векторы входа.

Создание сети

Для формирования слоя Кохонена предназначена М-функция newsc. Покажем, как она работает, на простом примере. Предположим, что задан массив из четырех двухэлементных векторов, которые надо разделить на 2 класса:

```

p = [.1 .8 .1 .9; .2 .9 .1 .8]
P =
    0.1000  0.8000  0.1000  0.9000
    0.2000  0.9000  0.1000  0.8000

```

В этом примере нетрудно видеть, что 2 вектора расположены вблизи точки (0,0) и 2 вектора - вблизи точки (1,1). Сформируем слой Кохонена с двумя нейронами для анализа двух-элементных векторов входа с диапазоном значений от 0 до 1:

```
net = newc([0 1; 0 1],2);
```

Первый аргумент указывает диапазон входных значений, второй определяет количество нейронов в слое. Начальные значения элементов матрицы весов задаются как среднее максимального и минимального значений, т. е. в центре интервала входных значений; это реализуется по умолчанию с помощью М-функции `midpoint` при создании сети. Убедимся, что это действительно так:

```

wts = net.IW{1,1}
wts =
    0.5000  0.5000
    0.5000  0.5000

```

Определим характеристики слоя Кохонена:

```

net.layers{1}
ans =
    dimensions: 2
    distanceFcn: 'dist'
    distances: [2x2 double]
    initFcn: 'initwb'
    netInputFcn: 'netsum'
    positions: [0 1]
    size: 2
    topologyFcn: 'hextop'
    transferFcn: 'compet'
    userdata: [1x1 struct]

```

Из этого описания следует, что сеть использует функцию евклидова расстояния `dist`, функцию инициализации `initwb`, функцию обработки входов `netsum`, функцию активации `compet` и функцию описания топологии `hextop`.

Характеристики смещений следующие:

```

net.biases{1}
ans =
    initFcn: 'initcon'
    learn: 1
    learnFcn: 'learncon'
    learnParam: [1x1 struct]
    size: 2
    userdata: [1x1 struct]

```

Смещения задаются функцией `initcon` и для инициализированной сети равны

```

net.b{1}
ans =
    5.4366
    5.4366

```

Функцией настройки смещений является функция `learncon`, обеспечивающая настройку с учетом параметра активности нейронов.

Элементы структурной схемы слоя Кохонена могут быть получены с помощью оператора `gensim(net)`

Они наглядно поясняют архитектуру и функции, используемые при построении слоя Кохонена. Теперь, когда сформирована самоорганизующаяся нейронная сеть, требуется обучить ее

решению задачи кластеризации данных. Напомним, что каждый нейрон блока `compnet` конкурирует за право ответить на вектор входа  $p$ . Если все смещения равны 0, то нейрон с вектором веса, самым близким к вектору входа  $p$ , выигрывает конкуренцию и возвращает на выходе значение 1; все другие нейроны возвращают значение 0.

#### *Правило обучения слоя Кохонена*

Правило обучения слоя Кохонена, называемое также *правилом Кохонена*, заключается в том, чтобы настроить нужным образом элементы матрицы весов.

Правило Кохонена представляет собой рекуррентное соотношение, которое обеспечивает коррекцию строки  $i$  матрицы весов добавлением взвешенной разности вектора входа и значения строки на предыдущем шаге. Таким образом, вектор веса, наиболее близкий к вектору входа, модифицируется так, чтобы расстояние между ними стало еще меньше. Результат такого обучения будет заключаться в том, что победивший нейрон, вероятно, выигрывает конкуренцию и в том случае, когда будет представлен новый входной вектор, близкий к предыдущему, и его победа менее вероятна, тогда будет представлен вектор, существенно отличающийся от предыдущего. Когда на вход сети поступает все большее и большее число векторов, нейрон, являющийся ближайшим, снова корректирует свой весовой вектор. В конечном счете, если в слое имеется достаточное количество нейронов, то каждая группа близких векторов окажется связанной с одним из нейронов слоя. В этом и заключается свойство самоорганизации слоя Кохонена.

Настройка параметров сети по правилу Кохонена реализована в виде М-функции `learnk`.

#### *Правило настройки смещений*

Одно из ограничений всякого конкурирующего слоя состоит в том, что некоторые нейроны оказываются незадействованными. Это проявляется в том, что нейроны, имеющие начальные весовые векторы, значительно удаленные от векторов входа, никогда не выигрывают конкуренции, независимо от того как долго продолжается обучение. В результате оказывается, что такие векторы не используются при обучении и соответствующие нейроны никогда не оказываются победителями. Такие нейроны-неудачники называют "*мертвыми*" нейронами, поскольку они не выполняют никакой полезной функции. Чтобы исключить такую ситуацию и сделать нейроны чувствительными к поступающим на вход векторам, используются смещения, которые позволяют нейрону стать конкурентным с нейронами-победителями. Этому способствует положительное смещение, которое добавляется к отрицательному расстоянию удаленного нейрона.

Соответствующее правило настройки, учитывающее нечувствительность мертвых нейронов, реализовано в виде М-функции `learncon`.

Параметр скорости настройки по умолчанию равен 0.001, и его величина обычно на порядок меньше соответствующего значения для М-функции `learnk`. Увеличение смещений для неактивных нейронов позволяет расширить диапазон покрытия входных значений, и неактивный нейрон начинает формировать кластер. В конечном счете он может начать притягивать новые входные векторы.

Это дает два преимущества. Если нейрон не выигрывает конкуренции, потому что его вектор весов существенно отличается от векторов, поступающих на вход сети, то его смещение по мере обучения становится достаточно большим и он становится конкурентоспособным. Когда это происходит, его вектор весов начинает приближаться к некоторой группе векторов входа. Как только нейрон начинает побеждать, его смещение начинает уменьшаться. Таким образом, задача активизации "мертвых" нейронов оказывается решенной. Второе преимущество, связанное с настройкой смещений, состоит в том, что они позволяют выравнивать значения параметра активности и обеспечить притяжение приблизительно одинакового количества векторов входа. Таким образом, если один из кластеров притягивает большее число векторов входа, чем другой, то более заполненная область притянет дополнительное количество нейронов и будет поделена на меньшие по размерам кластеры.

#### *Обучение сети*

Реализуем 10 циклов обучения. Для этого можно использовать функции `train` или `adapt`:

```
net.trainParam.epochs = 10;
```

```
net = train(net,p) ;
net.adaptParam.passes = 10;
[net,y,e] = adapt(net,mat2cell(p));
```

Заметим, что для сетей с конкурирующим слоем по умолчанию используется обучающая функция `trainwb1`, которая на каждом цикле обучения случайно выбирает входной вектор предъявляет его сети; после этого производится коррекция весов и смещений.

Выполним моделирование сети после обучения:

```
a = sim(net,p);
ac = vec2ind(a)
ac = 2 1 2 1
```

Видим, что сеть обучена классификации векторов входа на 2 кластера: первый расположен в окрестности вектора (0, 0), второй - в окрестности вектора (1, 1). Результирующие веса и смещения равны:

```
wts1 = net.IW{1,1}
b1 = net.b{1}
wts1 =
    0.58383    0.58307
    0.41712    0.42789
b1 = 5.4152
    5.4581
```

Заметим, что первая строка весовой матрицы действительно близка к вектору (1, 1), в то время как вторая строка близка к началу координат. Таким образом, сформированная сеть обучена классификации входов. В процессе обучения каждый нейрон в слое, весовой вектор которого близок к группе векторов входа, становится определяющим для этой группы векторов. В конечном счете, если имеется достаточное число нейронов, каждая группа векторов входа будет иметь нейрон, который выводит 1, когда представлен вектор этой группы, и 0 в противном случае, или, иными словами, формируется кластер. Таким образом, слой Кохонена действительно решает задачу кластеризации векторов входа.

*Пример:*

Функционирование слоя Кохонена можно пояснить более наглядно, используя графику системы MATLAB. Рассмотрим 48 случайных векторов на плоскости, формирующих 8 кластеров, группирующихся около своих центров. На графике, приведенном на рис. 7.3, казано 48 двухэлементных векторов входа.

Сформируем координаты случайных точек и построим план их расположения шшю-скости:

```
c = 8 ; n = 6 ; % Число кластеров, векторов в кластере
d = 0.5 ; % Среднеквадратичное отклонение от центра кластера
x = [-10 10;-5 5]; % Диапазон входных значений
[r,q] = size(x); minv = min(x')'; maxv = max(x')';
v = rand(r, c). * ((maxv - minv)*ones(1,c) + minv*ones(1,c));
t = c*n;% число точек
v=[v v v v v v]; v = v + randn(r,t)*d; % Координаты точек
P=v;
Plot(P(1,:), P(2,:), 'k')
```

```
Title('Векторы выхода'), xlabel('P(1,:)'), ylabel('P(2,:)')
```

После обучения в течение 500 циклов получим:

```
net.trainParam.epochs=500;
net=train(net,P);
w=net.IW{1}
bn=net.b{1}
cn=exp(1)./bn
```

Центры кластеризации распределились по восьми областям, смещения отклонились в обе стороны от исходного значения так же, как и параметры активности нейронов.

Рассмотренная самонастраивающаяся сеть Кохонена является типичным примером сети, которая реализует процедуру обучения без учителя.

Лабораторная работа № 3:

**Тема: Сеть Хопфилда**

Всякий целевой вектор можно рассматривать как набор характерных признаков некоторого объекта. Если создать рекуррентную сеть, положение равновесия которой совпадало бы с этим целевым вектором, то такую сеть можно было бы рассматривать как ассоциативную память. Поступление на вход такой сети некоторого набора признаков в виде начальных условий приводило бы ее в то или иное положение равновесия, что позволяло бы ассоциировать вход с некоторым объектом. Именно такими ассоциативными возможностями и обладают *сети Хопфилда*. Они относятся к классу рекуррентных нейронных обладающих тем свойством, что за конечное число тактов времени они из произвольного начального состояния приходят в состояние устойчивого равновесия, называемое *аттрактором*. Количество таких аттракторов определяет объем ассоциативной памяти сети Хопфилда.

Спроектировать сеть Хопфилда - это значит создать рекуррентную сеть со множеством точек равновесия, таких, что при задании начальных условий сеть в конечном счете приходит в состояние покоя в одной из этих точек. Свойство рекурсии проявляется в том, что выход сети подается обратно на вход.

Метод синтеза сети Хопфилда основан на построении системы линейных дифференциальных уравнений первого порядка, которая задана в некотором замкнутом гиперкубе пространства состояний и имеет решения в вершинах этого гиперкуба.

По команде `help hopfield` можно получить следующую информацию об М-функциях, входящих в состав ППП Neural Network Toolbox и относящихся к построению модифицированных сетей Хопфилда:

Hopfield recurrent networks	Рекуррентная модифицированная сеть Хопфилда
<b>New networks</b>	<b>Формирование сети</b>
nethop	Создание модифицированной сети Хопфилда
<b>Weight functions</b>	<b>Операции с весовой функцией</b>
dotprod	Скалярное произведение
<b>Net input functions</b>	<b>Операции над входами</b>
netsum	Суммирование
<b>Transfer functions</b>	<b>Функции активации</b>
satlins	Симметричная линейная функция с ограничениями
<b>Demonstrations</b>	<b>Демонстрационные примеры</b>
demohop1	Пример двумерной модифицированной сети Хопфилда
demohop2 demohop3	Пример неустойчивой точки равновесия
demohop4	Пример трехмерной модифицированной сети Хопфилда Пример устойчивых паразитных точек равновесия

*Архитектура сети*

Архитектура модифицированной сети Хопфилда представлена на рис. 1.

Вход `p` устанавливает значения начальных условий. В сети используется линейная функция активации с насыщением *satlins*, которая описывается следующим образом:

$$a = \text{satlins}(n) = \begin{cases} -1, & n < -1; \\ n, & -1 \leq n \leq 1; \\ 1, & n > 1. \end{cases}$$

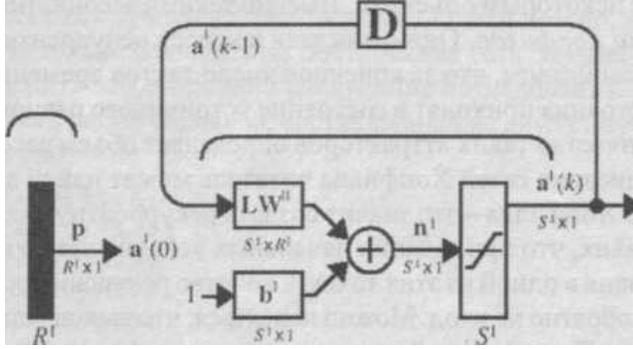


Рис. 1.

Эта сеть может быть промоделирована с одним или большим количеством векторов входа, которые задаются как начальные условия. После того как начальные условия заданы, сеть генерирует выход, который по обратной связи подается на вход. Этот процесс повторяется много раз, пока выход не установится в положение равновесия. Можно надеяться, что каждый вектор выхода в конечном счете сойдется к одной из точек равновесия, наиболее близкой к входному сигналу.

### **Распознавание образов**

#### *Постановка задачи*

Требуется создать нейронную сеть для распознавания 26 символов латинского алфавита. В качестве датчика предполагается использовать систему распознавания, которая выполняет оцифровку каждого символа, находящегося в поле зрения. В результате каждый символ будет представлен шаблоном размера 5x7.

Однако система считывания символов обычно работает неидеально и отдельные элементы символов могут оказаться искаженными.

Проектируемая нейронная сеть должна точно распознавать идеальные векторы входа и с максимальной точностью воспроизводить зашумленные векторы. М-функция `prprob` определяет 26 векторов входа, каждый из которых содержит 35 элементов, этот массив называется *алфавитом*. М-функция формирует выходные переменные `alphabet` и `targets`, которые определяют массивы алфавита и целевых векторов. Массив `targets` определяется как `eye(26)`. Для того чтобы восстановить шаблон для *i*-й буквы алфавита, надо выполнить следующие операторы:

```
[alphabet, targets] = prprob;
ti = alphabet(:, i);
letter{i} = reshape(ti, 5, 7)';
letter{i}
```

*Пример:*

Определим шаблон для символа А, который является первым элементом алфавита:

```
[alphabet, targets] = prprob;
i = 2;
ti = alphabet(:, i);
letter{i} = reshape(ti, 5, 7)';
letter{i}
ans =
    0 0 1 0 0
    0 1 0 1 0
    0 1 0 1 0
    1 0 0 0 1
    1 1 1 1 1
    1 0 0 0 1
    1 0 0 0 1
```

#### *Нейронная сеть*

На вход сети поступает вектор входа с 35 элементами; вектор выхода содержит 26 элементов, только один из которых равен 1, а остальные - 0. Правильно функционирующая сеть должна ответить вектором со значением 1 для элемента, соответствующего номеру символа в алфавите. Кроме того, сеть должна быть способной распознавать символы в условиях действия шума. Предполагается, что шум - это случайная величина со средним значением 0 и стандартным отклонением, меньшим или равным 0.2.

#### *Архитектура сети*

Для работы нейронной сети требуется 35 входов и 26 нейронов в выходном слое. Для решения задачи выберем двухслойную нейронную сеть с логарифмическими сигмоидальными функциями активации в каждом слое. Такая функция активации выбрана потому, что диапазон

выходных сигналов для этой функции определен от 0 до 1, и этого достаточно, чтобы сформировать значения выходного вектора.

Скрытый слой имеет 10 нейронов. Такое число нейронов выбрано на основе опыта и разумных предположений. Если при обучении сети возникнут затруднения, то можно увеличить количество нейронов этого уровня. Сеть обучается так, чтобы сформировать единицу в единственном элементе вектора выхода, позиция которого соответствует номеру символа, и заполнить остальную часть вектора нулями. Однако наличие шумов может приводить к тому, что сеть не будет формировать вектора выхода, состоящего точно из единиц и нулей. Поэтому по завершении этапа обучения выходной сигнал обрабатывается М-функцией `compet`, которая присваивает значение 1 единственному элементу вектора выхода, а всем остальным - значение 0.

#### *Инициализация сети*

Вызовем М-файл `prprob`, который формирует массив векторов входа `alphabet` размера  $35 \times 26$  с шаблонами символов алфавита и массив целевых векторов `targets`:

```
[alphabet,targets] = prprob;  
[R,Q] = size (alphabet) ;  
[S2,Q] = size (targets) ;
```

Двухслойная нейронная сеть создается с помощью команды `newff`:

```
S1 = 10;  
net = newff(minmax(alphabet),[S1 S2],{'logsig' 'logsig'},'traingdx');  
net.LW {2,1}=net.LW {2,1} *0.01;  
net.b {2}=net.b {2} *0.01;
```

#### *Обучение*

Чтобы создать нейронную сеть, которая может обрабатывать зашумленные векторы входа, следует выполнить обучение сети как на идеальных, так и на зашумленных векторах. Сначала сеть обучается на идеальных векторах, пока не будет обеспечена минимальная сумма квадратов погрешностей. Затем сеть обучается на 10 наборах идеальных и зашумленных векторов. Две копии свободного от шума алфавита используются для того, чтобы сохранить способность сети классифицировать идеальные векторы входа. К сожалению, после того, как описанная выше сеть обучилась классифицировать сильно зашумленные векторы, она потеряла способность правильно классифицировать некоторые векторы, свободные от шума. Следовательно, сеть снова надо обучить на идеальных векторах. Это гарантирует, что сеть будет работать правильно, когда на ее вход будет передан идеальный символ. Обучение выполняется с помощью функции `trainbpx`, которая реализует метод обратного распространения ошибки с возмущением и адаптацией параметра скорости настройки.

#### *Обучение в отсутствие шума*

Сеть первоначально обучается в отсутствие шума с максимальным числом циклов обучения 5000 либо до достижения допустимой средней квадратичной погрешки равной 0.1:

```
P = alphabet;  
T = targets;  
net.performFcn = 'sse' ;  
net.trainParam.goal = 0.1;  
net.trainParam.show = 20;  
net.trainParam.epochs = 5000;  
net.trainParam.me = 0.95;  
[net,tr] = train(net,P,T);
```

#### *Обучение в присутствии шума*

Чтобы спроектировать нейронную сеть, не чувствительную к воздействию шума, обучим ее с применением двух идеальных и двух зашумленных копий векторов алфавита. Целевые векторы состоят из четырех копий векторов. Зашумленные векторы имеют шум со средним значением 0.1 и 0.2. Это обучает нейрон правильно распознавать зашумленные символы и в то же время хорошо распознавать идеальные векторы.

При обучении с шумом максимальное число циклов обучения сократим до 300, а допустимую погрешность увеличим до 0.6:

```
netn = net;  
netn.trainParam.goal = 0.6;  
netn.trainParam.epochs = 300;  
T=[targets targets targets targets];  
for pass = 1:10  
P=[alphabet, alphabet, ...  
(alphabet + randn(R,Q)*0.1), ...  
(alphabet + randn(R,Q)*0.2)];  
[netn,tr] = train(netn,P,T);  
end
```

#### *Повторное обучение в отсутствие шума*

Поскольку нейронная сеть обучалась в присутствии шума, то имеет смысл повторить ее обучение без шума, чтобы гарантировать, что идеальные векторы входа классифицируются правильно.

```
netn.trainParam.goal = 0.1; % Предельная среднеквадратичная погрешность  
netn.trainParam.epochs = 500; % Максимальное количество циклов обучения  
net.trainParam.show = 5; % Частота вывода результатов на экран  
[netn,tr]=train(netn,P,T);
```

#### *Эффективность функционирования системы*

Эффективность нейронной сети будем оценивать следующим образом. Рассмотрим 2 структуры нейронной сети: сеть 1, обученную на идеальных последовательностях, и сеть 2, обученную на зашумленных последовательностях. Проверка функционирования производится на 100 векторах входа при различных уровнях шума.

Приведем фрагмент сценария `appc1`, который выполняет эти операции:

```
noise_range = 0:.05:.5; max_test =  
100;  
network1 = [ ];  
network2 = [];  
T = targets;  
% Выполнить тест  
for noiselevel = noise_range  
errors1 = 0;  
errors2 = 0;  
for i=1:max_test  
P = alphabet + randn(35,26)*noiselevel;  
% Тест для сети 1  
A = sim(net,P);  
AA = compet(A);  
errors1 = errors1 + sum(sum(abs(AA-T)))/2;  
% Тест для сети 2  
An = sim(netn,P);  
AAn = compet(An);  
errors2 = errors2 + sum(sum(abs(AAn-T)))/2;  
echo off  
end
```

% Средние значения ошибок (100 последовательностей из 26 векторов целей)

```
network1 = [network1 errors1/26/100];  
network2 = [network2 errors2/26/100];  
end
```

Тестирование реализуется следующим образом. Шум со средним значением 0 и стандартным отклонением от 0 до 0.5 с шагом 0.05 добавляется к векторам входа. Для каждого уровня шума формируется 100 зашумленных последовательностей для каждого символа вычисляется выход сети. Выходной сигнал обрабатывается М-функцией `compnet` с той целью, чтобы выбрать только один из 26 элементов вектора выхода. После этого оценивается количество ошибочных классификаций и вычисляется процент ошибки.

Сеть 1 обучена на идеальных векторах входа, а сеть 2 - на зашумленных. Обучение сети на зашумленных векторах входа значительно снижает погрешность распознавания реальных векторов входа. Сети имеют очень малые погрешности, если среднеквадратичное значение шума находится в пределах от 0.00 до 0.05. Когда к векторам был добавлен шум со среднеквадратичным значением 0.2, в обеих сетях начали возникать заметные ошибки. При этом погрешности нейронной сети, обученной на зашумленных векторах, на 3-4% ниже, чем для сети, обученной на идеальных входных последовательностях.

Если необходима более высокая точность распознавания, сеть может быть обучена либо в течение более длительного времени, либо с использованием большего количества нейронов в скрытом слое. Можно также увеличить размер векторов, чтобы пользоваться шаблоном с более мелкой сеткой, например 10x14 точек вместо 5x7.

### МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

По выбранной теме студенты выполняют реферативную работу.

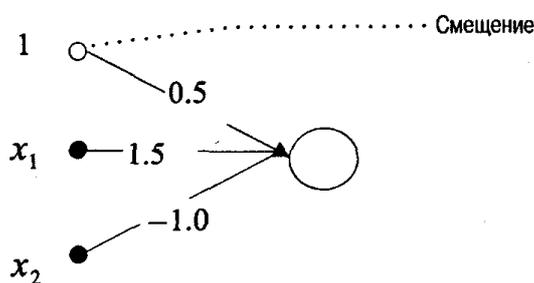
Реферативная работа включает следующие разделы:

1. Обоснование актуальности выбранной тематики и описание целей выполнения работы.
2. Систематизация и анализ найденных в научной печати, в сети Интернет и других источниках материалов.
3. Выводы.
4. Предложения по использованию результатов работы в конкретных областях и возможные направления дальнейших исследований.

### ФОНД КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЗНАНИЙ ПО ДИСЦИПЛИНЕ

Задание № 1.

Вычислите комбинированный сетевой ввод для элемента на рисунке и соответствующее выходное значение при использовании пороговой функции и входного вектора [0.7 2.5].

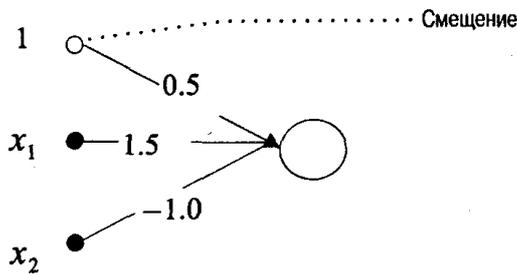


Задание №2.

Вычислите выходное значение, используя в качестве функции активности сигмоидальную функцию. Входной вектор [0.7 2.5], весовая матрица  $w = \begin{pmatrix} 0,5 \\ 1,5 \\ -1,0 \end{pmatrix}$ .

Задание № 3.

Вычислите комбинированный ввод для сети с архитектурой, показанной на рисунке, но с набором весовых значений  $[-0.2 \ 0.03 \ 1.2]$  и входным вектором  $[0.7 \ 2.5]$ .



Задание № 4.

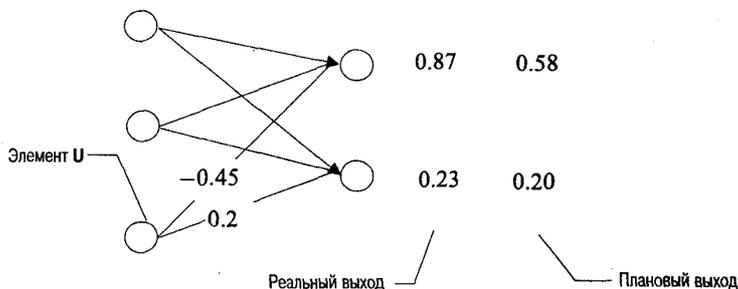
Представьте полностью прямой и обратный проходы в сети с прямой связью, использующей алгоритм обратного распространения ошибок, для входного образца  $[0.1 \ 0.9]$  и целевого выходного значения  $0.9$  в предположении, что сеть имеет архитектуру 2-2-1 (т.е. два входных, два скрытых один выходной элемент) с весовыми коэффициентами

для первого слоя,  $\begin{bmatrix} 0,1 & 0,1 \\ -0,2 & -0,1 \\ 0,1 & 0,3 \end{bmatrix}$  для второго слоя.

слоя,  $\begin{bmatrix} 0,2 \\ 0,2 \\ 0,3 \end{bmatrix}$  для второго слоя.

Задание № 5.

На рисунке



показаны скрытый и выходной слой сети с прямой связью. Вычислите ошибку для скрытого элемента U при условии, что значение его активности для обрабатываемого сетью образца равно  $0.64$ .

Задание № 6.

Сеть типа 2-2-1 с радиальными базисными функциями используется для решения проблемы XOR. Первый слой весов задан матрицей  $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ .

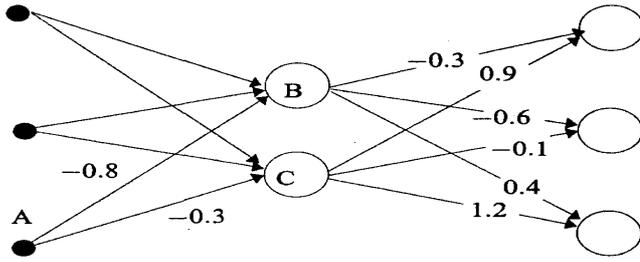
Для каждого вводимого образца XOR вычислите значения активности для всех скрытых элементов, если функция активности имеет вид  $\varphi(net) = \exp[-net^2]$ , где  $net$  является евклидовой нормой.

Задание № 7.

На рисунке показана сеть с обратным распространением ошибок во время обработки учебного вектора  $[1.0 \ 0.9 \ 0.9]$ , для которого целевым выходным вектором является  $[0.1 \ 0.9 \ 0.1]$ . Пусть выходом элемента В является значение  $0.6$ , а выходом элемента С — значение  $0.8$ . Предположим, что функцией активности является сигмоид.

(а) Вычислите фактический выходной вектор.

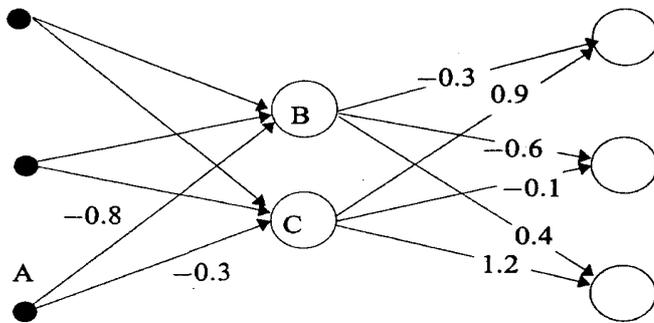
- (b) Вычислите значения ошибок для каждого выходного элемента.  
 (c) Вычислите значения ошибок для каждого скрытого элемента.



Задание № 8.

На рисунке показана сеть с обратным распространением ошибок во время обработки учебного вектора  $[0.1 \ 0.5 \ 0.9]$ , для которого целевым выходным вектором является  $[0.1 \ 0.9 \ 0.1]$ . Пусть выходом элемента В является значение 0.6, а выходом элемента С — значение 0.8. Предположим, что функцией активности является сигмоид.

- (a) Вычислите фактический выходной вектор.  
 (b) Вычислите значения ошибок для каждого выходного элемента.  
 (c) Вычислите значения ошибок для каждого скрытого элемента.



Задание № 9.

Для обучения сети SOFM с тремя входными и двумя кластерными элементами используются четыре учебных вектора:  $[0.8 \ 0.7 \ 0.4]$ ,  $[0.6 \ 0.9 \ 0.9]$ ,  $[0.3 \ 0.4 \ 0.1]$ ,  $[0.1 \ 0.1 \ 0.3]$  и на-

чальные весовые значения  $\begin{bmatrix} 0,5 & 0,4 \\ 0,6 & 0,2 \\ 0,8 & 0,5 \end{bmatrix}$ .

Начальный радиус выбирается равным 0, а норма обучения  $\eta$  — ной 0.5. Вычислите изменения весовых значений в ходе первого цикла обработки данных, рассматривая учебные векторы в указанном порядке.

Задание № 10.

Пусть учебными векторам являются вектора  $a=[1 \ 4]$ ,  $p_1=[2 \ 1]$ ,  $p_2=[6 \ 6]$ .

- (a) Используя скалярное произведение, покажите, что  $a$  ближе к  $p_2$ , чем к  $p_1$ .  
 (б) Если  $p$  оказывается прототипом-победителем в некоторой сети SOFM, то покажите, как должен двигаться вектор  $p$  в предположении, что вектор  $a$  предлагается на рассмотрение сети два раза подряд (т.е. в случае, когда нет других учебных образцов). Считайте, что  $\eta=1$ .

Задание № 11.

Пусть учебными векторам являются вектора  $x = [0.2 \ -1.4 \ 2.3]$ ,  $p_1=[0.6 \ -4.0 \ 7.0]$ ,  $p_2=[0.1 \ -1.0 \ 2.2]$ .

- (a) К какому из прототипов оказывается ближе всего вектор  $x$  в смысле евклидова расстояния?  
 (б) К какому из прототипов оказывается ближе всего вектор  $x$  в смысле скалярного произведения?

(в) Скорректируйте весовой вектор прототипа-победителя из п. (а) в соответствии с алгоритмом обучения SOFM при норма обучения 0.8.

Задание 12.

Пусть учебными векторам являются вектора  $x = [0.2 \ -1.4 \ -0.3 \ 0.8]$ ,  
 $p_1 = [0.3 \ -3.0 \ 1.0 \ 0.2]$ ,  $p_2 = [0.4 \ -1.4 \ -2.0 \ 3.0]$ .

(а) К какому из прототипов оказывается ближе всего вектор  $x$  в смысле евклидова расстояния?

(б) К какому из прототипов оказывается ближе всего вектор  $x$  в смысле скалярного произведения?

(в) Скорректируйте весовой вектор прототипа-победителя из п. (а) в соответствии с алгоритмом обучения SOFM при норма обучения 0.8.

Задание № 13.

Найдите набор весовых значений сети Хопфилда, соответствующий сохранению образца [1-111].

Задание № 14.

Найдите устойчивое состояние сети Хопфилда при условии, что входным образцом является [-1-1 1 1].

Задание № 15.

Определите весовую матрицу сети Хопфилда, соответствующую сохранению следующих двух векторов [-1 1 -1], [1 -1 1].

Задание № 16.

Определите весовые значения сети Хопфилда, соответствующие сохранению образца [1 1 1 -1].

Задание № 17.

Определите весовые значения сети Хопфилда для сохранения следующих образцов: [-1 1 -1 -1 1 -1 -1 -1], [-1 -1 1 -1 -1 -1 1 -1].

Проверьте устойчивость сети при предоставлении ей на вход в качестве подсказок сохраненных образцов.

Задание № 18.

Проверьте устойчивость сети при предоставлении ей на вход в качестве подсказок следующих образцов: [-1 1 -1 -1 1 -1 -1 1],

[1 -1 1 1 -1 1 -1 -1].

Задание № 19.

Определите весовые значения сети Хопфилда для сохранения следующих образцов:

[-1 1 -1 -1 1 -1 -1 -1],

[-1 -1 1 -1 -1 -1 1 -1],

[1 -1 -1 1 -1 1 -1 -1].

Проверьте устойчивость сети при предоставлении ей на вход в качестве подсказки первого сохраненного образца.

## КОМПЛЕКТ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Что такое нейронные сети?

2. Сеть Хопфилда.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

1. Свойства биологических и искусственных нейронных сетей.
2. Двухнаправленная ассоциативная память.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

1. Способы реализации нейросетей
2. Автоассоциативное обратное распространение ошибок. Базовая архитектура сети.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

1. Типы задач, решаемых нейронными сетями. Недостатки и ограничения нейронных сетей.
2. Рекуррентные сети. Базовая архитектура сети.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 5

1. Биологический нейрон. Структура, функции.

2. Обратное распространение во времени. Базовая архитектура рекуррентной сети с обратным распространением ошибок.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 6

1. Формальный нейрон. Ограниченность модели формального нейрона.
2. Кластеризация образцов. Сеть Кохонена.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 7

1. Виды функций активации.
2. Обучение слоя Кохонена.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 8

1. Структура связей в нейронных сетях. Правило распространения сигналов в сети.
2. Примеры применения перцептронов.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 9

1. Структура связей. Правило комбинирования входящих сигналов.

2. Самоорганизующаяся карта признаков. Базовая архитектура сети.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 10

1. Свойства биологических нейронных сетей.
2. Многослойный перцептрон. Базовая архитектура сети.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 11

1. Свойства искусственных нейронных сетей.
2. Алгоритм обратного распространения ошибки.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 12

1. Недостатки и ограничения нейронных сетей.
2. Сети с радиальными и базисными функциями.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

« \_\_\_\_ » \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 13

1. Этапы решения задачи с помощью нейронной сети.

2. Самоорганизующаяся карта признаков. Свойства самоорганизующейся карты признаков.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

«\_\_\_» \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 14

1. Формальный нейрон. Виды функций активации.
2. Сеть Хопфилда. Вычислительная энергия сети Хопфилда.

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра ИУС

Утверждено на заседании кафедры

«\_\_\_» \_\_\_\_\_ 200\_г.

Заведующий кафедрой

Утверждаю \_\_\_\_\_

Факультет МиИ

Курс II

Дисциплина

Нейроинформатика

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 15

1. Проблемы, возникающие при моделировании нейронных сетей.
2. Обучение по алгоритму обратного распространения ошибки.