Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 02.09.2026 11:03:02
Уникальный программный ключ:
48505f11ec15acaa386f5219d3113d727fefda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Казанский (Приволжский) федеральный университет»
Елабужский институт (филиал) КФУ

УТВЕРЖДАЮ
Заместитель директора по
образовательной деятельности
_____С.Ю. Бахвалов
«08» 08 2026 г.
МП

Программа дисциплины (модуля)
Нейронные сети

Направление подготовки/специальность: 15.03.06 Мехатроника и робототехника

Направленность (профиль) подготовки (специальности): Физические основы мехатроники и робототехники

Квалификация: бакалавр

Форма обучения: заочная

Язык обучения: русский

Год начала обучения по образовательной программе: - 2026

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

2. Место дисциплины (модуля) в структуре ОПОП ВО

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

4.2. Содержание дисциплины (модуля)

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

6. Фонд оценочных средств по дисциплине (модулю)

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

9. Методические указания для обучающихся по освоению дисциплины (модуля)

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

13. Приложение №1. Фонд оценочных средств

14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)

15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. Анисимова Э.С. (кафедра математики и прикладной информатики, ВШИиОН Елабужский институт КФУ)

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-2	Способен разрабатывать, отлаживать, внедрять и сопровождать программное обеспечение мехатронных и робототехнических систем
ПК-2.1	Знать способы разработки, отладки и сопровождения программного обеспечения для мехатронных и робототехнических систем
ПК-2.2	Уметь разрабатывать, отлаживать и сопровождать программное обеспечение для мехатронных и робототехнических систем
ПК-2.2	Владеть навыками разработки, отладки и сопровождения программного обеспечения для мехатронных и робототехнических систем

Обучающийся, освоивший дисциплину (модуль):

Должен знать:
- основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем

Должен уметь:
- применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем

Должен владеть:
- способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в Блок 1 "Дисциплины (модули)" Б1.В.ДВ.02.01 основной профессиональной образовательной программы 15.03.06 «Мехатроника и робототехника» (Физические основы мехатроники и робототехники) и относится к дисциплинам по выбору, части, формируемой участниками образовательных отношений.

Осваивается на 2 курсе (лето).

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 4 зачетных(ые) единиц(ы) на 144 часа(ов).

Контактная работа - 10 часа(ов), в том числе лекции - 4 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 6 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 130 часа(ов).

Контроль (зачёт / экзамен) - 4 часа(ов).

Форма промежуточного контроля дисциплины: зачет на 2 курсе (лето).

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием

отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Нейронные сети. Основные понятия	4	1	0	1	32
2.	Тема 2. Многослойные нейронные сети	4	2	0	1	32
3.	Тема 3. Радиальные нейронные сети	4	1	0	2	34
4.	Тема 4. Самоорганизующиеся нейронные сети	4	1	0	2	32
	Итого: 144 часа (из них 4 часа контроль)		4	0	6	130

4.2 Содержание дисциплины (модуля)

Тема 1. Нейронные сети. Основные понятия

Биологические представления о нейроне. Понятие нейронная сеть. Искусственный нейрон. Классификация нейронных сетей. Архитектура нейронных сетей. История нейрокомпьютерных вычислений. Персептрон. Задача распознавания.

Тема 2. Многослойные нейронные сети

Понятие многослойной нейронной сети. Принципы построения многослойных нейронных сетей. Алгоритм обратного распространения ошибки. Недостатки алгоритма обратного распространения ошибки.

Тема 3. Радиальные нейронные сети

Понятие радиальной нейронной сети. Структура радиальной нейронной сети. Математические основы функционирования радиальных нейронных сетей. Обучение радиальной нейронной сети.

Тема 4. Самоорганизующиеся нейронные сети

Самоорганизующиеся нейронные сети. Структура сети Кохонена. Обучение сети Кохонена. Нейронные сети классификации.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержденный приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года № 245)

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);

- критерии оценивания сформированности компетенций;

- механизм формирования оценки по дисциплине (модулю);

- описание порядка применения и процедуры оценивания для каждого оценочного средства;

- критерии оценивания для каждого оценочного средства;

- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке Елабужского института КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра каждого из изданий основной и дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки Елабужского института КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Нейронные сети - http://statsoft.ru/home/textbook/modules/stneunet.html

Учебник – нейронные сети https://e.lanbook.com/book/122180

Введение в машинное обучение: полное руководство - https://www.machinelearningmastery.ru/machine-learning-introduction-a-comprehensive-guide-af6712cf68a3/

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	Лекционные занятия проводятся с использованием интерактивных технологий и предполагают активное участие студентов. Для подготовки к занятиям рекомендуется выделять в материале проблемные вопросы, затрагиваемые преподавателем в лекции, и группировать информацию вокруг них. Желательно выделять в используемой литературе постановки вопросов, на которые разными авторам могут быть даны различные ответы. На основании постановки таких вопросов следует собирать аргументы в пользу различных вариантов решения поставленных проблем.
лабораторные	Лабораторные занятия - это одна из разновидностей практического занятия, являющаяся

Вид работ	Методические рекомендации
работы	эффективной формой учебных занятий в организации высшего образования. Лабораторные занятия имеют выраженную специфику в зависимости от учебной дисциплины, углубляют и закрепляют теоретические знания. На этих занятиях студенты осваивают конкретные методы изучения дисциплины, обучаются экспериментальным способам анализа, умению работать с приборами и современным оборудованием. Лабораторные занятия дают наглядное представление об изучаемых явлениях и процессах, студенты осваивают постановку и ведение эксперимента, учатся умению наблюдать, оценивать полученные результаты, делать выводы и обобщения. Отчёт по итогам выполненных лабораторных работ выполняется на листах белой бумаги формата А4 в печатном или рукописном виде. При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру вверху. При оформлении отчёта в печатном виде желательно соблюдать следующие требования. Для заголовков: полужирный шрифт, 14 пт, центрированный. Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине. Во всех случаях тип шрифта - Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал. Поля: левое - 3 см, правое - 1 см, верхнее и нижнее - 2 см. Отчет должен содержать следующие элементы: 1) Титульный лист с обязательным указанием варианта; 2) Цель работы; 3) Задание; 4) Основная часть; 5) Вывод
самостоя-тельная работа	Самостоятельная работа студентов по дидактической сути представляет собой комплекс условий обучения, организуемых преподавателем и направленных на самоподготовку учащихся. Учебная деятельность протекает без непосредственного участия преподавателя и заключается в проработке лекционного материала, подготовке к лабораторным занятиям; изучении учебной литературы из основного и дополнительного списка.
зачет	Зачет является формой оценки качества освоения студентом образовательной программы по дисциплине. По результатам зачета студенту выставляется оценка "зачтено" или "не зачтено". Зачет может проводиться в форме устного опроса по билетам (вопросам) или без билетов, с предварительной подготовкой или без подготовки, по усмотрению кафедры. Преподаватель может проставить зачет без опроса или собеседования тем студентам, которые активно участвовали на лабораторных занятиях.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Учебная аудитория для проведения учебных занятий лекционного типа, семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации № 61

Комплект мебели для преподавателя – 1 шт., посадочные места для обучающихся – 30 шт., одноместные столы – 12 шт., компьютерные столы – 18 шт., компьютеры – 19 шт., интерактивная панель – 1 шт., меловая доска настенная – 1 шт., выход в интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду.

Помещение для самостоятельной работы № 10

Посадочные места для пользователей – 28 шт., металлические двусторонние стеллажи для книг – 11 шт., книжный шкаф открытый – 5 шт., проектор – 1 шт., ноутбуки для пользователей – 11 шт., шкаф каталожный – 8 шт., шкаф для одежды – 1 шт., ксерокс – 1 шт., рабочий стол библиотекаря – 1 шт., компьютер библиотекаря – 1 шт., вешалка для одежды – 1 шт., жалюзи рулонные «Омега» с фотопечатью – 4 шт., стенд настенный (бронированное стекло) – 4 шт., шкаф-витрина встроенный в арку – 2 шт., шкаф-витрина стеклянный – 2 шт., стеллаж трубчатый с деревянными полками – 2 шт., рабочий стол для инвалидов и лиц с ОВЗ – 2 шт., стол СИ-1 рабочий для инвалидов-колясочников – 1 шт., компьютер– 2 шт., наушники – 2 шт., устройство «Говорящая книга» (тифлоплеер) – 2 шт., видеоувеличитель – 2 шт., радиокласс – 1 шт., портативный тактильный дисплей - 1 шт., сканирующая читающая машина - 1 шт., сканер – 1 шт., веб-камера – 1 шт., выход в интернет, внутривузовская

компьютерная сеть, доступ в электронную информационно-образовательную среду.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 15.03.06 «Мехатроника и робототехника» и профилю подготовки "Физические основы мехатроники и робототехники".

Приложение №1
к рабочей программе дисциплины (модуля)
Б1.В.ДВ.02.01 Нейронные сети

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал) КФУ

Фонд оценочных средств по дисциплине
Нейронные сети

Направление подготовки: 15.03.06 Мехатроника и робототехника
Профиль подготовки: Физические основы мехатроники и робототехники
Квалификация выпускника: бакалавр
Форма обучения: заочная
Язык обучения: русский
Год начала обучения по образовательной программе: 2026

СОДЕРЖАНИЕ

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)
2. Критерии оценивания сформированности компетенций
3. Распределение оценок за формы текущего контроля и промежуточную аттестацию
4. Оценочные средства, порядок их применения и критерии оценивания
4.1. Оценочные средства текущего контроля
4.1.1. Реферат
4.1.1.1. Порядок проведения.
4.1.1.2 Критерии оценивания
4.1.1.3. Содержание оценочного средства
4.1.2. Лабораторные работы
4.1.2.1. Порядок проведения.
4.1.2.2 Критерии оценивания
4.1.2.3. Содержание оценочного средства
4.1.3 Тестирование
4.1.3.1. Порядок проведения и процедура оценивания
4.1.3.2. Критерии оценивания
4.1.3.3. Содержание оценочного средства
4.2. Оценочные средства промежуточной аттестации
4.2.1 Зачет
4.2.1.1. Порядок проведения.
4.2.1.2. Критерии оценивания.
4.2.1.3. Оценочные средства.

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
ПК-2 Способен разрабатывать, отлаживать, внедрять и сопровождать программное обеспечение мехатронных и робототехнических систем	Знать основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем Уметь применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем Владеть способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем	**Текущий контроль:** Реферат по темам Тема 1. Нейронные сети. Основные понятия Тема 2. Многослойные нейронные сети Тема 3. Радиальные нейронные сети Тема 4. Самоорганизующиеся нейронные сети Лабораторные работы по темам Тема 1. Нейронные сети. Основные понятия Тема 2. Многослойные нейронные сети Тема 3. Радиальные нейронные сети Тема 4. Самоорганизующиеся нейронные сети Тестирование по темам Тема 1. Нейронные сети. Основные понятия Тема 2. Многослойные нейронные сети Тема 3. Радиальные нейронные сети Тема 4. Самоорганизующиеся нейронные сети **Промежуточная аттестация:** *Зачёт*

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично)	**Средний уровень (хорошо)**	**Низкий уровень (удовлетворительно)**	**Ниже порогового уровня (неудовлетворительно)**
ПК-2	Знает основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем	Знает основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем, допуская незначительные ошибки в рассуждениях и выводах	Знает основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем, допуская типичные ошибки в рассуждениях и выводах	Не знает основные понятия теории нейронных сетей; способы применения нейронных сетей в разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем.
	Умеет применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических	Умеет применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем, допуская	Умеет применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических	Не умеет применять нейросетевые технологии при разработке, отладке и сопровождении программного обеспечения для

	систем	незначительные ошибки.	систем, допуская типичные ошибки	мехатронных и робототехнических систем
	Владеет способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем	Владеет способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем, допуская незначительные ошибки в программной реализации	Владеет способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем, допуская типичные ошибки в программной реализации	Не владеет способностью реализовывать методы нейросетевой обработки при разработке, отладке и сопровождении программного обеспечения для мехатронных и робототехнических систем

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

4 семестр:
Текущий контроль:
Реферат
Тема 1. Нейронные сети. Основные понятия
Тема 2. Многослойные нейронные сети
Тема 3. Радиальные нейронные сети
Тема 4. Самоорганизующиеся нейронные сети

Лабораторные работы
Тема 1. Нейронные сети. Основные понятия
Тема 2. Многослойные нейронные сети
Тема 3. Радиальные нейронные сети
Тема 4. Самоорганизующиеся нейронные сети

Тестирование
Тема 1. Нейронные сети. Основные понятия
Тема 2. Многослойные нейронные сети
Тема 3. Радиальные нейронные сети
Тема 4. Самоорганизующиеся нейронные сети

Промежуточная аттестация - зачет
Промежуточная аттестация проводится после завершения изучения дисциплины или ее части в форме, определяемой учебным планом образовательной программы с целью оценить работу обучающегося, степень усвоения теоретических знаний, уровень сформированности компетенций.
Преподаватель, принимающий зачет обеспечивает случайное распределение вариантов зачетных заданий между обучающимися с помощью билетов и/или с применением компьютерных технологий; вправе задавать обучающемуся дополнительные вопросы и давать дополнительные задания помимо тех, которые указаны в билете.
Зачет проводится по билетам. В каждом билете оценочные средства одного вида: устный или письменный ответ на вопрос.

Для зачета:
Зачтено
Не зачтено

4. Оценочные средства, порядок их применения и критерии оценивания
4.1. Оценочные средства текущего контроля
4.1.1. Реферат
Тема 1. Нейронные сети. Основные понятия
Тема 2. Многослойные нейронные сети
Тема 3. Радиальные нейронные сети
Тема 4. Самоорганизующиеся нейронные сети

4.1.1.1. Порядок проведения.

Обучающиеся самостоятельно пишут работу на заданную тему и сдают преподавателю в письменном виде. В работе производится обзор материала в определённой тематической области либо предлагается собственное решение определённой теоретической или практической проблемы. Оцениваются проработка источников, изложение материала, формулировка выводов, соблюдение требований к структуре и оформлению работы, своевременность выполнения. В случае публичной защиты реферата оцениваются также ораторские способности

Требования к реферату

При оформлении текста реферата следует придерживаться следующих параметров:

поля: левое – 35 мм, правое – 15 мм, верхнее – 25 мм, нижнее – 25 мм;

ориентация страницы: книжная;

шрифт: TimesNewRoman;

кегль: 14 пт (пунктов);

красная строка:1 мм;

междустрочный интервал: полуторный;

выравнивание основного текста и сносок: по ширине.

Иллюстрации в виде рисунков, фотоснимков, схем и т.п. могут располагаться органично с текстом (возможно ближе к иллюстрируемой части) либо на отдельных листах. В любом случае выполняется нумерация (сквозная для всех разделов), которая располагается вверху. Подрисуночную нумерацию и надпись располагать внизу.

Заканчивается пояснительная записка библиографическим списком источников, к которым обращался студент во время работы над разрабатываемой темой.

Объем информационно-технологической документации не регламентируется – он диктуется достаточностью для практического применения. Карточки задания для самоконтроля (если таковы имеются) вкладываются в прозрачные файлы.

Реферат по своему структурному содержанию должен содержать следующие элементы:

- титульный лист;
- содержание;
- введение;
- базовое понятия;
- историческая справка (особенности зарождения и развития, основоположники и т.д.);
- классификация (виды, формы и т.д.);
- общее и частное положения по применению в учебно-воспитательном процессе;
- глоссарий;
- список использованных источников
- приложения

4.1.1.2 Критерии оценивания

Оценка «отлично» ставится, если обучающийся:

Тему раскрыл полностью. Продемонстрировал превосходное владение материалом. Использовал надлежащие источники в нужном количестве. Структура работы соответствует поставленным задачам. Степень самостоятельности работы высокая.

Оценка «хорошо» ставится, если обучающийся:

Тему в основном раскрыл. Продемонстрировал хорошее владение материалом. Использовал надлежащие источники. Структура работы в основном соответствует поставленным задачам. Степень самостоятельности работы средняя.

Оценка «удовлетворительно» ставится, если обучающийся:

Тему раскрыл слабо. Продемонстрировал удовлетворительное владение материалом. Использованные источники и структура работы частично соответствуют поставленным задачам. Степень самостоятельности работы низкая.

Оценка «неудовлетворительно» ставится, если обучающийся:

Тему не раскрыл. Продемонстрировал неудовлетворительное владение материалом. Использованные источники недостаточны. Структура работы не соответствует поставленным задачам. Работа несамостоятельна

4.1.1.3. Содержание оценочного средства

1. Правило обучения Хебба.
2. Линейная нейронная сеть.
3. Рекуррентный метод наименьших квадратов
4. Линейная сеть с линей задержки
5. Распознавание символов
6. Концепция нейроуправления
7. Нейроэмуляторы и нейропредикторы
8. Нейронная сеть Элмана
9. Сети Хопфилда
10. Нейронная сеть Хэмминга
11. Адаптивные резонансные нейронные сети

12. Методы глобальной оптимизации
13. Метод имитации отжига
14. Стохастические методы обучений нейронных сетей
15. Метод роя частиц
16. Элементная база для аппаратной реализации нейрокомпьютеров
17. Сети на основе теории адаптивного резонанса
18. Машина Больцмана
19. Двунаправленная ассоциативная память
20. Многопрофильная модель релаксационной нейронной сети

4.1.2. Лабораторные работы

Тема 1. Нейронные сети. Основные понятия

Тема 2. Многослойные нейронные сети

Тема 3. Радиальные нейронные сети

Тема 4. Самоорганизующиеся нейронные сети

4.1.2.1. Порядок проведения.

В аудитории, оснащённой соответствующим оборудованием, обучающиеся проводят учебные эксперименты и тренируются в применении практико-ориентированных технологий. Оцениваются знание материала и умение применять его на практике, умения и навыки по работе с оборудованием в соответствующей предметной области.

Лабораторные работы проводятся преподавателем согласно разработанному и утвержденному на кафедре рабочей программе. Каждая лабораторно-практическая работа выполняется по определенной теме программы в соответствии с заданием.

Перед выполнением каждой работы студенты-бакалавры должны проработать соответствующий материал, используя конспекты теоретических занятий, периодические издания, учебно-методические пособия и учебники

На каждом занятии студенты выполняют работу в соответствии с ее содержанием и методическими указаниями.

По окончании занятий студенты оформляют отчет по каждой работе, соблюдая следующую форму:

- Наименование темы;
- Цель работы;
- Задание и содержание выполненной работы,
- Письменные ответы на контрольные вопросы.
- Выводы по проделанной работе.
- Список использованных источников.

4.1.2.2 Критерии оценивания

Зачтено ставится, если обучающийся:

Правильно выполнил все задания. Продемонстрировал высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий. Правильно выполнил большую часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий. Задания выполнил более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.

Не зачтено ставится, если обучающийся:

Задания выполнил менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.

4.1.2.3. Содержание оценочного средства

ЛАБОРАТОРНАЯ РАБОТА № 1
ИЗУЧЕНИЕ СВОЙСТВ ЛИНЕЙНОГО НЕЙРОНА И ЛИНЕЙНОЙ НЕЙРОННОЙ СЕТИ

1 ЦЕЛЬ РАБОТЫ

Изучить свойства линейного нейрона и линейной нейронной сети.

2 СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1 Модель нейрона

Искусственные нейронные сети (НС) представляют собой простейшие математические модели мозга. Понять основные принципы построения НС можно, рассматривая их как совокупность (сеть) отдельных структур (нейронов). Очень грубо структуру биологического нейрона можно описать следующим образом. Нейрон имеет сому – тело, дерево входов – дендриты, выход – аксон. На соме и на дендритах располагаются окончания аксонов других нейронов, называемых синапсами. Принятые синапсами входные сигналы стремятся либо возбудить нейрон, либо затормозить. Когда суммарное возбуждение достигает некоторого порога, нейрон возбуждается и посылает по аксону сигнал другим нейронам. Каждый синапс обладает уникальной синаптической силой, которая пропорционально своему значению изменяет передаваемый на нейрон входной сигнал. В соответствии с приведенным описанием математическая модель нейрона представляет собой суммирующий пороговый элемент (рис. 1).

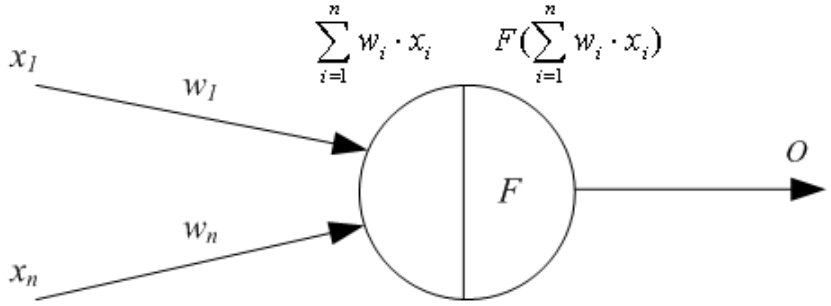

Рис. 1. Искусственный нейрон

Формула срабатывания нейрона:

$$O = F(< W^T, X >) = F(\sum_{i=1}^{n} w_i \cdot x_i),$$

$$O = \begin{cases} 1, (< W^T, X >) \geq 0, \\ 0, \textit{иначе.} \end{cases}$$

2.2 Алгоритм обучения по дельта-правилу

Обучение НС происходит на некоторой обучающей выборке, для каждого образца которой определяются и сравниваются с желаемыми значениями все текущие выходы. Если разница недопустима, то веса изменяются. Окончанием обучения считается ситуация, когда общая ошибка на всех образцах допустима.

Все алгоритмы обучения нейросетей являются разновидностями алгоритма обучения по методу коррекции ошибки, которая осуществляется по-разному. Идея изменения весов НС сводится к нахождению общей меры качества сети, в качестве которой обычно выбирают функцию ошибки сети. Тогда, чтобы подобрать нужные веса, необходимо минимизировать функцию ошибки. Самым распространенным методом поиска минимума является метод градиентного спуска. Для случая функции с одной переменной веса изменяются в направлении, противоположном производной, т. е. справедлива формула

$$W^{n+1} = W^n - h \cdot F'(W),$$

где h – некоторый уровень обучения, шаг изменения;
$F'(W)$ – производная функции качества НС для одной переменной.

Для функции F от n переменных и единичного вектора e в пространстве R^n $||e|| = 1$, $e \in R^n$, дифференциал выражается формулой

$$\partial_e F(W) = \frac{\lim_{t \to 0} (F(W + et) - F(W))}{t}.$$

Для случая $e = (0, 0...1...0)$ определим частный дифференциал

$$\partial_i F(W) = \frac{\lim_{t\to 0}(F(w_1, w_2, ..., w_{i+et}, ..., w_n) - F(W))}{t}.$$

Таким образом, антиградиент – это набор следующих дифференциалов:

$$\partial F(W) = ((-\partial F(w_1), -\partial F(w_2), ..., -\partial F(w_i), ..., -\partial F(w_n))^T.$$

Для определения обобщенной функции ошибки рассмотрим обучающую выборку $\{(x^k, y^k)\}$, где $k = 1, ..., K$. Накопленная по всем эпохам ошибка

$$E = \sum_{k=1}^{K}(E^k) = \sum_{k=1}^{K}\left(\sum_{i=1}^{m} 1/2\|O_i - Y_i\|^2\right).$$

Формула модификации весов НС

$$W^{n+1} = W^n - h \cdot \partial E / \partial W$$

уточняется для различных видов функции активации. Для линейной функции $F(t) = t$, НС формирует каждый выход как скалярное произведение весов на вектор входов: $O_i = < W_i, X_i >$ и градиент будет равен:

$$\partial E / \partial W = -(Y_i - O_i) \cdot X,$$

где Y_i – желаемый выход; O_i – полученный выход; X – вектор выхода.

Таким образом, получаем формулу изменения весов

$$W^{n+1} = W^n - h \cdot (Y_i - O_i) \cdot X.$$

Если значением δ назвать разницу $(Y_i - O_i)$, то получим формулу

$$W^{n+1} = W^n - h \cdot \delta \cdot X,$$

что является *алгоритмом обучения по* δ *-правилу*.

2.3 Описание основных функций

Для работы с нейронными сетями необходимо инсталлировать MATLAB и обладать первоначальными знаниями относительно языка системы. Поскольку MATLAB представляет собой интерпретатор, то обучение инструментарию нейронных сетей заключается в основном в изучении функций и их параметров. Узнать возможности нейрона как классификатора простых линейно сепарабельных задач можно путем проведения экспериментов с моделью одного линейного нейрона.

Функция *newp*. Для того чтобы создать нейрон, используют функцию newp, имеющую следующий синтаксис:

$$net = newp(PR, S, TF, LF),$$

где *PR* – матрица минимальных и максимальных *R* входных элементов; *S* – количество нейронов (при создании одного нейрона *S*=1); *TF* – функция активации (transfer function); *LF* – имя функции обучения нейрона.

В случае если параметры функции *newp* не заданы, их значения определяются посредством ввода значений в диалоговые окна. Построенный нейрон характеризуется функциями весов (weight function), входов сети (net input function) и определенной функцией активации. Функция весов – это умножение весов на входной сигнал, функция входов сети – их сумма. Веса задаются как для входов нейрона, так и для фиксированного входа, задающего порог срабатывания (bias). Вектор весов инициализируется нулями. Для обучения используются функции, рассмотренные ниже.

Функция *learnp* настраивает веса нейрона. Синтаксис функции обучения довольно сложен:

$$[dW, LS] = learnp(W, P, Z, N, A, T, E, gW, gA, D, LP, LS),$$
$$[db, LS] = learnp(b, ones(1, Q), Z, N, A, T, E, gW, gA, D, LP, LS),$$
$$info = learnp(code).$$

Функция *learnp (W, P, Z, N, A, T, E, gW, gA, D, LP, LS)* имеет несколько входов, где вектор *W* – вектор весов; *P* – вектор входов; *Z* – вектор взвешенных входов; *N* – вектор сети; *A* – вектор выхода; *T* – вектор желаемых выходов; *E* – вектор ошибок; *gW* – вектор изменения весов; *gA* – изменения выходов. Функция возвращает значения: *dW* – изменения матрицы весов; *LS* – новый уровень обученности.

Функция *learnp* может быть использована с параметрами по умолчанию:

$$dW = learnp([\,], p, [\,], [\,], [\,], [\,], e, [\,], [\,], [\,], [\,], [\,]).$$

Использование пустого списка [ ] означает параметр по умолчанию.

Функция *learnp* вычисляет изменение весов *dW* для заданного нейрона в соответствии с правилом обучения персептрона:

$$dW = \begin{cases} 0, \text{ если ошибка } e = 0, \\ p', \text{ если } e = 1, \\ -p', \text{ если } e = -1, \end{cases}$$

т. е. $dW = e \cdot p'$.

Функция *learnpn* настраивает нормализованные веса:

$$[dW, LS] = learnpn(W, P, Z, N, A, T, E, gW, gA, D, LP, LS).$$

Функция *learnpn* вычисляет изменение весов *dW* для данного нейрона и его входа *P* и ошибки *E* в соответствии с нормализованным правилом обучения персептрона:

$$pn = p / \sqrt{(1 + p(1)^2 + p(2)^2 + \ldots + p(R)^2)},$$
$$dW = \begin{cases} 0, \text{ если ошибка } e = 0, \\ pn', \text{ если } e = 1, \\ -pn', \text{ если } e = -1, \end{cases}$$

т. е. $dW = e \cdot pn'$.

Линейный нейрон имеет одно существенное ограничение. Входные векторы должны быть линейно разделимы. Если векторы невозможно отделить прямой или гиперплоскостью, то персептрон не способен решить задачу классификации.

Функция *adapt* адаптирует НС к условиям задачи:

$$[net, Y, E, Pf, Af] = adapt(net, P, T, P_i, A_i).$$

Параметры функции *adapt*: *net* – имя сети; *P* – входы сети; *T* – желаемый выход; P_i – исходные условия задержки; A_i – исходные условия задержки для слоя. Функция возвращает параметры адаптированной сети *net.adaptParam*: *net* – измененная сеть; *Y* – выход сети; *E* – ошибки сети; *Pf* – условия задержки входов; *Af* – условия задержки слоя. Параметры P_i и *Pf* необязательные и необходимы только для сетей, имеющих задержки на входах и слое.

Функция *train* также обучает НС и использует следующий синтаксис:

$$[net, tr] = train(net, P, T, P_i, A_i).$$

Функция *train* имеет следующие параметры: *net* – сеть; *P* – входы сети; *T* – желаемый выход; P_i – исходные условия задержки входа; A_i – исходные условия задержки слоя.

Функция *sim* имитирует нейронную сеть:

$$[Y, Pf, Af] = sim(net, P, P_i, A_i),$$

где *net* – сеть; *P* – входы сети; P_i — исходные условия задержки входов сети; A_i – исходные условия задержки слоя. Функция возвращает *Y* – выходы сети; *Pf* – окончательные условия задержки входов; *Af* – окончательные условия задержки слоя.

Функции активации. Ниже представлены назначения этих функций.

Функция	Назначение
hardlim	Возвращает 1, если на входе положительное число и 0 в противном случае.
tansig	Вычисляет гиперболический тангенс от входа.
purelin	Вычисляет выход слоя от сетевого входа.

Функции графического интерфейса и вспомогательные функции. Назначение этих функций представлено ниже.

Функция	Назначение
axis([X_{min} X_{max} Y_{min} Y_{max}])	Устанавливает диапазоны координатных осей.
title('строка')	Выводит в графическое окно рисунков заголовок графика.
rand(M, N)	Возвращает матрицу размерности *M* на *N* со случайными значениями.
xlabel('строка') *ylabel('строка')*	Подписывают наименование координатных осей.
cla reset	Очищает координатную сетку в окне рисунков.
hold on *hold off*	Включают и отключают режим добавления графиков на координатную сетку.
text(X, Y, 'строка')	Выводит строку, начиная с указанных координат в поле рисунков.
pause (n)	Ожидает пользовательского ответа *n* секунд.
plot(X, Y, 'цвет и символ')	Изображает на координатной сетке точки с координатами, заданными векторами *X, Y,* с помощью указанного символа и цвета.
plotpv(P, V)	Изображает точки *P* указанными маркерами *T,* где *P* – матрица входных векторов размерностью *R* на *Q* (*R* должен быть 3 или меньше), *T* – матрица двоичных векторов размерностью 5 на *Q (S* должен быть 3 или меньше).
plotes (WV, BV, ES, V)	Изображает поверхность ошибки на отдельном входе, где *WV* – вектор строк значений весов *W* размерности *N, BV* – вектор строк значений порогов *B* размерности *M, ES* – матрица ошибки размерности *M* на *N, V* – угол зрения по умолчанию [-37, 5, 30].
plotsom(POS)	Изображает позицию нейрона красной точкой, связывая синей линией нейроны, находящиеся друг от друга на расстоянии 1. *POS* – матрица *S N*-размерных нейронов.
ind2vec *vec2ind*	Позволяют представить индексы либо собственно значениями индексов, либо векторами, строки которых содержат 1 в позиции индекса.
full	Преобразует разреженную матрицу в полную.
maxlinlr(P)	Функция возвращает максимальный уровень обученности линейного слоя без *bias,* который обучался только на векторе *P.*
trainlm	Выполняет обучение многослойной НС методом Левенберга-Марквардта.
netprod	Входная сетевая функция, которая вычисляет выход сетевого слоя, умножая входной вектор на веса и прибавляя *bias.*
init	Итеративно инициализирует НС.

Структура данных описания нейронных сетей. Структура данных *net* – это описание обученной НС. Обучение осуществляется в соответствии со следующими параметрами, значения которых либо устанавливаются пользователем, либо по умолчанию.

Структура данных	Комментарий
net.trainParam. epochs 100	Максимальное количество эпох обучения.
net. trainParam.goal 0	Целевое значение ошибки.
net.trainParam.max_fail 5	Максимальное значение ошибки.

net.trainParam.mem reduc 1	Фактор оптимизации процесса обучения: оптимизация использования памяти или времени процессора.
net. trainParam. min_grad 1e-10	Минимальное значение градиента.
net.trainParam.show 25	Количество эпох между показами.
net.trainParam.time inf	Максимальное время обучения в секундах.
TR	Структура данных, содержащая значения об обученности НС в текущую эпоху.
TR. epoch	Номер эпохи.
TR.perf	Уровень обученности *(Trainingperformance).*
TR.vperf	Степень качества *(Validation performance).*
TR.tperf	Результативность обработки теста *(Testperformance).*
TR.mu	Значение адаптивности.

Структура данных описания адаптированной НС *net.adaptfcn* включает в себя следующие поля *net.adapt.param: NET* – адаптированная НС; *Y* – выходы НС; *E* – ошибки НС; *Pf* – окончательные входные значения задержек; *Af* – окончательные выходные задержки; *TR* – результат обучения (эпохи и целевая ошибка). Проведем в среде *Matlab toolbox* эксперименты, используя рассмотренные функции.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

ПРИМЕР 1 Создание нейронов, реализующих функции логического И и логического ИЛИ

Создадим нейрон с одним двухэлементным входом (интервалы первого и второго элементов [0; 1]). Определим два первых параметра функции *newp,* а в качестве значений третьего и четвертого параметра (типа функции активации и имени процедуры обучения) воспользуемся значениями по умолчанию.

```
% создание нейрона с одним двухэлементным входом (интервал
% первого элемента [0; 1] и интервал второго элемента [-2; 2]
net = newp([0 1; -2 2], 1);
```

Для того чтобы исследовать поведение нейрона, необходимо имитировать его работу с помощью функции *sim.* Для определения последовательности значений входа создадим последовательность *P1.*

```
% создание последовательности значений входа
P1 = {[0; 0] [0; 1] [1; 0] [1; 1]};
% имитация работы нейрона net на последовательности входов P
% желаемых выходов – T1, которая позволит нам провести адаптацию
% нейрона (обучить его) через 20 проходов.
Y = sim (net, P1);
% создание последовательности выходов
T1 = {0, 0, 0, 1};
% установка количества проходов (циклов) адаптации
net.adaptParam.passes = 20;
% адаптация нейрона net для обучающей выборки <P1; T1>
net = adapt (net, P1, T1);
% симуляция работы нейрона net на последовательности входов  P1
Y = sim (net, P1);
```

В результате мы получим нейрон, выполняющий функцию логического И.
Для переобучения нейрона на выполнение функции ИЛИ переопределим входы *P* и выходы *T.*

```
% создание последовательности входов
P2 = [0 0 1 1; 0 1 0 1];
% создание последовательности выходов (реакций) для нейрона,
% выполняющего функцию логического ИЛИ
T2 = [0, 1, 1, 1];
% Инициализируем нейрон, обучим его на 20 проходах (эпохах)
% инициализация нейрона net
net = init (net);
% имитация работы нейрона net на последовательности входов P2
Y = sim (net, P2);
```

% установка количества проходов
net.trainParam.epochs = 20;
% обучение нейрона net на обучающей выборке <P2, T2>
net = train(net, P2, T2);
% имитация работы нейрона net на последовательности входов P2
Y = sim (net, P2);

Функция train выводит график обучения нейрона (рис. 2).

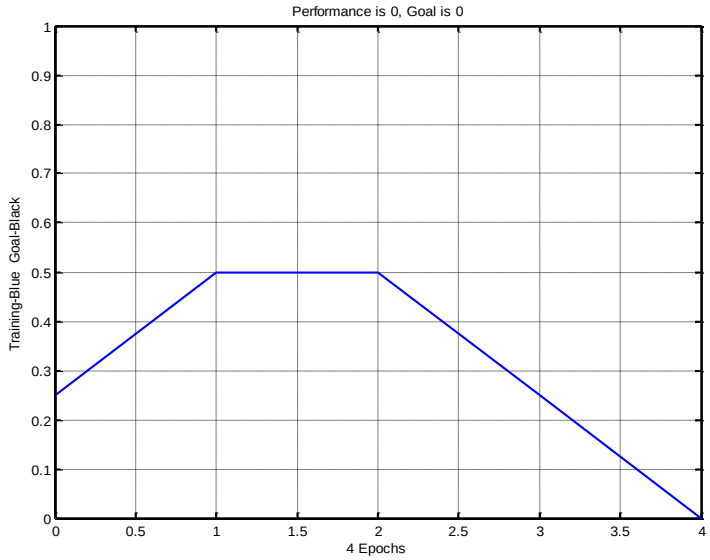

Рис. 2. График обучения нейрона

Для случайного изменения весов и порога срабатывания используем функцию *init*. По умолчанию для создаваемого нейрона указана функция *hardlim*.

ПРИМЕР 2 Обучение нейрона классификации векторов на две категории

Начнем с классификации векторов на основе двухвходового нейрона. Будем использовать функцию *newp* для создания нейрона, *sim* для имитации его работы, *adapt для* адаптации (обучения) нейронной сети. Обучим двухвходовый нейрон классифицировать входные векторы на две категории.

% определение четырех двухэлементных входов
P = [-0.5 -0.5 0.4 -0.2; -0.5 0.5 -0.4 1.0];
% зададим желаемые выходы нейрона для определенных векторов
T = [1 1 0 0];
% изобразим входные векторы (точки) на плоскости:
plotpv(P, T);

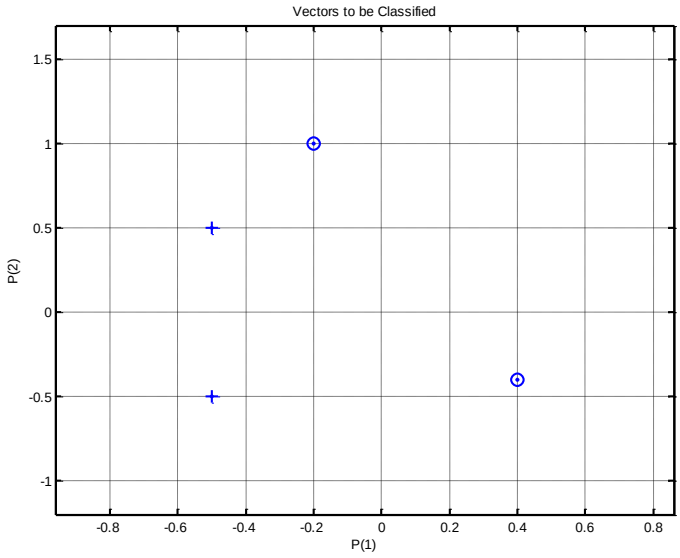

Рис. 3. Исходные векторы, предназначенные для классификации нейроном

Каждый из четырех входных векторов на плоскости *P* определяется двумя координатами, представленными как двухэлементные столбцы в матрице *P* (рис. 3).

Создадим один линейный нейрон с двумя входами, значения которых лежат в интервале [-1, 1].

```
% создание линейного нейрона с двумя входами из интервала [-1, 1]
net = newp([-1 1; -1 1], 1);
```

Нейрон по умолчанию имеет функцию активации *hardlim* и такой нейрон разделяет входные векторы прямой линией.

Определим координаты линии классификации: веса *(IW)* и порог срабатывания нейрона *(b).*

```
% получение управляющей структуры linehandle для изображения
% разделяющей линии в координатах весов (IW) и порога
% срабатывания нейрона (b)
linehandle = plotpc (net.IW{1}, net.b{1});
% изображение разделяющей прямой
plotpc(net.IW{1}, net.b{1});
```

Если исходным весам задать нулевые значения, то любые входы дадут одинаковые выходы, и линия классификации не будет видна на плоскости. Проведем обучение:

```
% очистка координатных осей
cla;
% изображение входных векторов двух категории, категория задается
% элементами вектора T
plotpv(P, T);
% получение управляющей структуры linehanctle
linehandle = plotpc(net.IW{1}, net.b{1});
% присвоение начального значения ошибки
E = 1;
% инициализация нейрона
net = init (net);
% получение управляющей структуры linehandle
linehandle = plotpc (net.IW{1}, net.b{1});
% организация цикла пока ошибка не равна 0
while (mse(E))',
% адаптация нейрона net на обучающей выборке <P, T>,
% функция возвращает адаптированный нейрон net,
% выход Y, ошибку E
[net, Y, E] = adapt (net, P, T);
```

```
% изображение разделяющей прямой нейрона после адаптации
linehandle = plotpc(net.IW{1}, net.b{1}, linehandle);
% очистка окна графиков
drawnow;
% конец цикла while
end;
```

Функция *adapt* возвращает новый объект – сеть, которая выполняет классификацию, выход сети и ошибку (рис. 4).

Проведем классификацию нового вектора с помощью обученного нейрона на основе функции *sim*. Определим новый вектор *P*.

```
% определение вектора P
P = [0, 6; 1, 1];
% имитация работы нейрона net, получение отклика нейрона a
a = sim (net, P);
% изображение входа P, отнесенного нейроном к категории a
plotpv (P, a);
```

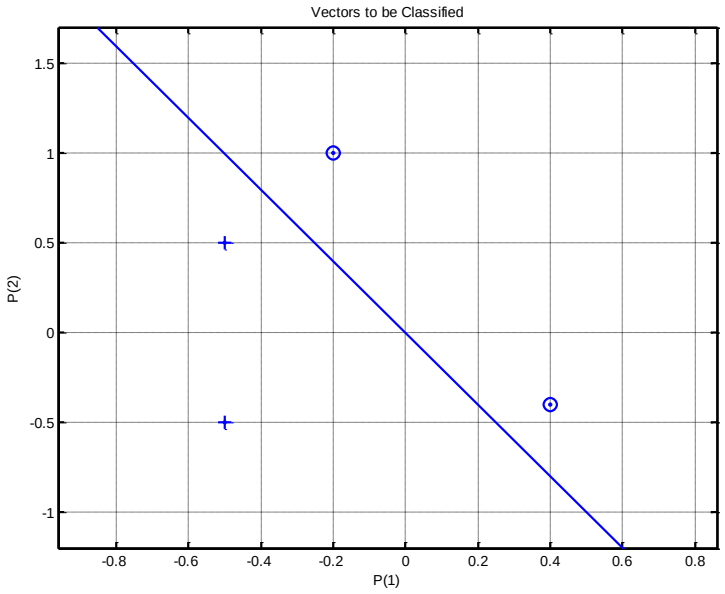

Рис. 4. Прямая, разделяющая исходные векторы на классы

Обученный нейрон можно использовать для классификации любого вектора:

```
% включить режим добавления графиков в графическом окне
hold on;
% изображение входных точек в соответствии с категориями T
plotpv(P, T);
% изображение разделяющей поверхности
plotpc (net.IW{1}, net.b{1});
% отключение режима добавления графиков
hold off;
```

Нейрон классифицирует новую точку, как принадлежащую категории «0» (представлена кружком), а не категории «1» (представлена +).

ПРИМЕР 3 Создание слоя линейных нейронов

Рассмотрим последовательность из 10 шагов (для выхода *T1*, который известным образом зависит от входов *P1):*

```
% последовательность входов
```

```
P1 = {-1 0 0 0 1 1 -1 0 -1 1};
% последовательность выходов
T1 = {-1 -1 1 0 1 2 0 -1 -1 1};
```

Используем функцию *newlin*, чтобы создать нейрон со значениями входа в интервале [-1; 1], задержками входа от 0 до 1 и уровнем обучения 0,1.

```
% создание линейного слоя из одного нейрона со значениями входа
% в интервале [-1; 1], задержками входа от 0 до 1
% и уровнем обучения 0,1.
net = newlin ([-1 1], 1, [0 1], 0.1);
```

Адаптируем нейрон к задаче одним проходом через последовательность входа. Измерим среднюю квадратичную ошибку с помощью функции *mse(e)*.

```
% адаптация нейрона к последовательности P1
[net, y, e, pf] = adapt (net, P1, T1);
% измерение ошибки
mse (e)
```

Получим довольно большую ошибку. Вновь адаптируем сеть на 10 шагах последовательности, используя предыдущее значение *pf* как новое исходное значение задержки:

```
P2 = {1 -1 -1 1 1 -1 0 0 0 1};
T2 = {2 0 -2 0 2 0 -1 0 0 1};
% адаптация с начальным вектором задержки pf
[net, y, e, pf] = adapt (net, P2, T2, pf);
mse (e)
```

Адаптируем сеть на 100-разовом прогоне последовательности:

```
% формирование новой последовательности входов
P3 = [P1 P2];
% формирование новой последовательности выходов
T3 = [T1 T2];
% установка количества проходов
net.adaptParam.passes = 100;
% адаптация нейрона
[net, y, e] = adapt (net, P3, T3);
```

Получим приемлемую ошибку, значит сеть обучена зависимости входов от выходов.

4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляет собой математическая модель нейрона?Как выглядит формульное представление алгоритма обучения по дельта-правилу?
2. Какое ограничение имеет линейный нейрон?
3. Можно ли обучить линейный нейрон выполнять логическую функцию исключающего ИЛИ?
4. Какие функции используются для настройки весов персептрона в среде MATLAB? В чем их отличие?

ЛАБОРАТОРНАЯ РАБОТА № 2
ИЗУЧЕНИЕ МНОГОСЛОЙНОГО НЕЛИНЕЙНОГО ПЕРСЕПТРОНА И АЛГОРИТМА ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

1 ЦЕЛЬ РАБОТЫ

Изучить возможности многослойного персептрона как аппроксиматора и классификатора.

2 СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1 Алгоритм обратного распространения ошибки

Пусть определена трехслойная нейронная сеть с n входами, m выходами и l скрытыми между ними элементами, тогда необходимо рассмотреть и построить два слоя весов: от входов к скрытым элементам и к выходу, т. е. (W_1, W_2).

Назначение алгоритма обратного распространения ошибки – настройка всех слоев многослойной структуры. Рассмотрим работу алгоритма на примере сети с одним скрытым слоем и одним выходом (рис. 5). Преобразования входных сигналов, задаваемые нейронной сетью, определяются следующими формулами:

$$F(< W, X >) = 1/(1+\exp(-W^T X));$$
$$O_1 = 1/(1+\exp(-W_1^T X));$$
$$O_2 = 1/(1+\exp(-W_2^T O_1)).$$

Общая функция ошибки зависит от весов всех слоев, в нашем случае от вектора W_2 и от матрицы W_1:

$$E(W_1, W_2) = 1/2(Y - 1/(1+\exp(-W_2^T O_1)))^2,$$

где Y – выход, который задан в обучающей выборке.

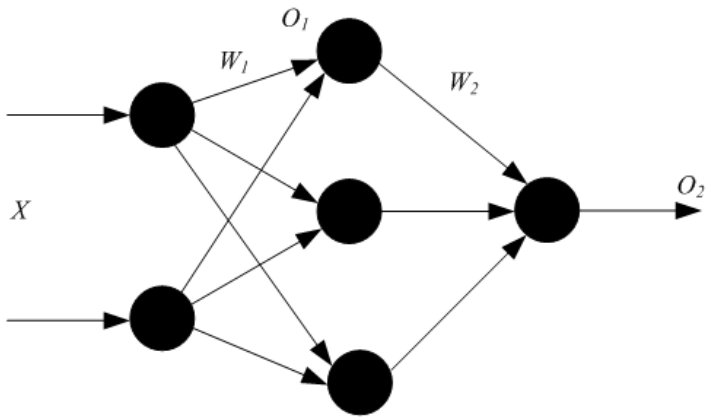

Рис. 5. Многослойный персептрон

Теперь необходимо определить приращение каждого веса с помощью частных производных:

$$\frac{\partial E(W_2, W_1)}{\partial W_2};$$
$$\frac{\partial E(W_2, W_1)}{\partial W_1}.$$

Для многослойной архитектуры частные производные ошибки по матрице весов каждого слоя определяются по формуле сложной производной. В случае униполярной сигмоиды правило изменения весов будет следующим:

$$W_2 = W_2 + h \cdot (Y - O_2) \cdot O_2 \cdot (1 - O_2) \cdot O_1;$$
$$\partial = (Y - O_2) \cdot O_2 \cdot (1 - O_2);$$
$$W_1 = W_1 + h \cdot \partial \cdot W_1 \cdot (1 - O_1) \cdot O_1 \cdot X.$$

Таким образом, метод обратного распространения ошибки позволяет изменять веса промежуточных слоев, хотя желаемые значения на промежуточных слоях не заданы.

2.2 Описание основных функций

Функция *newff* создает нейронную сеть прямого распространения сигнала, обучаемую с помощью алгоритма обратного распространения ошибки:

net = newff(PR, [S1 S2 SNl], {TF1 TF2 TFNl), BTF, BLF, PF).

Рассмотрим параметры функции *newff*: *PR* – матрица интервалов значений для *R* входных элементов, задаваемых минимальным и максимальным значениями; S_i – размер *i*-го слоя, для *N* слоев; TF_i – функция активации *i*-го слоя, по умолчанию используется функция *tansig* – гиперболический тангенс; *BTF* – функция обучения сети методом обратного распространения ошибки, по умолчанию используется функция *traingdx*; *BLF* – функция изменения весов при обучении, по умолчанию используется *learngdm*; *PF* – функция измерения ошибки, по умолчанию *mse.* Функция *newff* возвращает многослойную нейронную сеть прямого и обратного распространения сигнала и ошибки соответственно. Функции активации могут быть выбраны из следующего перечня: гиперболический тангенс *tansig*, логистическая сигмоида *logsig* или линейная функция *purelin.*

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

ПРИМЕР 1 Создание и обучение нейронной сети с помощью алгоритма обратного распространения ошибки

Зададим с помощью графика исходную функцию:

```
% входы НС
P = [0 1 2 3 4 5 6 7 8];
% желаемые реакции НС
T = [0 0.44 0.88 0.11 -0.66 -0.95 -0.45 0.18 0.92];
% изображение аппроксимируемой функции
plot(P, T, 'o');
```

Используем функцию *newff*, чтобы создать двухслойную сеть прямого распространения. Пусть сеть имеет входы с интервалом значений от 0 до 8, первый слой с 10 нелинейными сигмоидальными, второй – с одним линейным нейронами. Используем для обучения алгоритм обратного распространения ошибки (backpropagation) Левенберга – Марквардта (рис. 6).

```
% создание двухслойной НС прямого распространения с интервалом
% значений входов от 0 до 8, причем первый слой содержит
% 10 нелинейных сигмоид, а второй — один линейный нейрон.
% Для обучения используется алгоритм обратного распространения
% ошибки (backpropagation).
net = newff([0 8], [10 1], {'tansig' 'purelin'},'trainlm');
% имитация работы необученной НС
yl = sim (net, P);
% изображение результатов работы необученной НС
plot(P, T, 'o', P, yl, 'x') ;
% Обучим сеть на 100 эпохах с целевой ошибкой 0.01:
% установка количества проходов
net.trainParam.epochs = 50;
% установка целевого значения ошибки
net.trainParam.goal = 0.01;
% обучение НС (рис. 6)
net = train(net, P, T) ;
% имитация работы обученной НС
y2 = sim(net,  P);
% изображение результатов работы НС (рис. 7)
plot(P, T, 'o', P, yl, 'x', P, y2, '+');
```

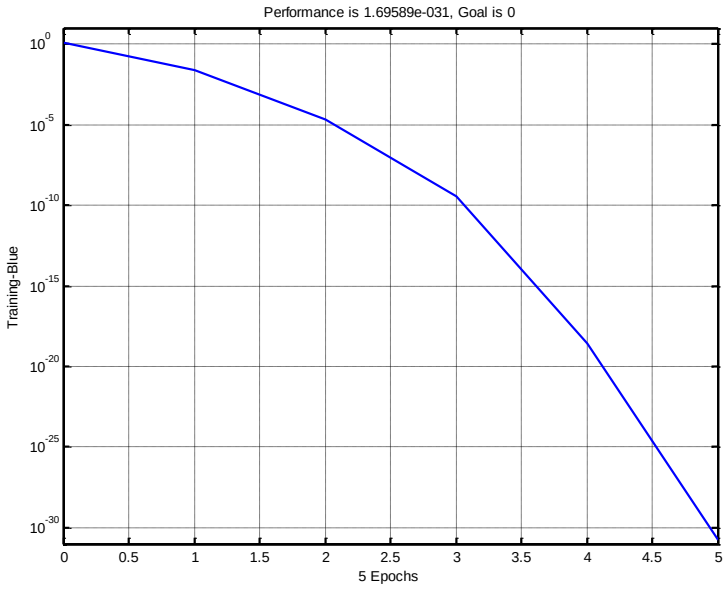

Рис. 6. График обучения двухслойного персептрона

Для исследования работы алгоритма обратного распространения ошибки воспользуемся примером, встроенным в Matlab toolbox, набрав команду demo.

В появившемся диалоговом окне необходимо последовательно выбирать пункты меню: Toolboxes->Neural Network->Other Demos->Other Neural Network Design textbook demos->Table of Contents->10-13->Backpropagation Calculation.

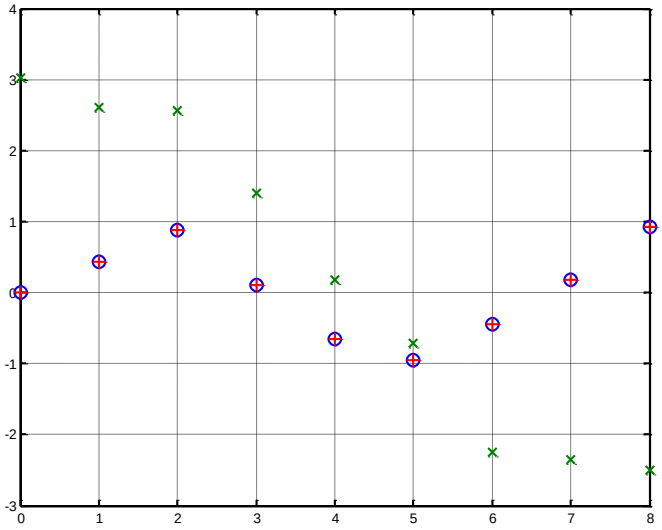

Рис. 7. Результат аппроксимации векторов двухслойным персептроном

В примере используется двухслойный персептрон с двумя нелинейными нейронами в первом слое и одним во втором. Действие алгоритма обратного распространения ошибки разбито на следующие шаги: назначение входа и желаемого выхода, прямой проход входного сигнала до выхода, обратное распространение ошибки, изменение весов. Переменные, позволяющие проследить работу алгоритма обратного распространения ошибки, обозначены следующим образом:

P – входной сигнал;

$W_1(i)$ – вектор весов первого слоя, $W_1(1)$ – вес связи, передающий входной сигнал на первый нейрон, а $W_1(2)$ – на второй;

$W_2(i)$ – вектор весов второго слоя, $W_2(1)$ – вес связи, передающий входной сигнал с первого нейрона во второй слой, а $W_2(2)$ – со второго;

$B_1(i)$ – вектор пороговых значений *(bias)* нейронов первого слоя, i = 1, 2;
B_2 – пороговое значение *(bias)* нейрона второго слоя;
$N_1(i)$ – вектор выходов первого слоя, i = 1, 2;
N_2 – выход второго слоя;
$A_1(i)$ – вектор выходных сигналов первого слоя после выполнения функции активации (сигмоиды), i = 1, 2;
A_2 – выход второго слоя после выполнения функции активации (линейной);
lr – коэффициент обучаемости.

Пусть входной сигнал P = 1,0, а желаемый выход $t = 1 + \sin(p * \pi / 4) = 1{,}707$.
Результаты выполнения этапов алгоритма представлены в табл. 1.

Таблица 1

Результаты поэтапного выполнения алгоритма обратного распространения ошибки

Этап	Прямое распространение входного сигнала	Обратное распространение ошибки	Изменение весов
$A_1(1)$, $A_1(2)$	$Logsig(W_1P+B_1)$ = = [0,321, 0,368]	Не выполняется	Не выполняется
A_2	$purelin(W_1P+B_1)$ = = 0,446	*То же*	*То же*
e	$t - A_2$ = 1,261	»	»
$N_1(1)$, $N_1(2)$	Не выполняется	$\frac{\partial \log sim(N_1)}{\partial N_1 \cdot W_2 \cdot N_2} = = [0{,}049,\ 0{,}100]$	»
N_2	То же	$-2 \cdot \frac{\partial purelin(N_2)}{\partial N_2 \cdot e} = -2{,}522$	»
$W_1(1)$ $W_1(2)$	»	Не выполняется	$W_1 = W_1 - \mathrm{lr} \cdot N_1 \times P = = [-0{,}265, -0{,}420]$
$B_1(1)$, $B_1(2)$	»	То же	$B_1 = B_1 - \mathrm{lr} \cdot N_1 = = [-0{,}475, -0{,}140]$
B_2	»	»	$B_2 = B_2 - \mathrm{lr} \cdot N_2 = 0{,}732$
$W_2(2)$	»	»	$W_2 = W_2 - \mathrm{lr} \cdot N_2 \times N_1 = = [0{,}171,\ 0{,}077]$

4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каким алгоритмом обучают многослойные НС?
2. Из каких основных этапов состоит алгоритм обратного распространения ошибки?
3. Почему алгоритм обратного распространения ошибки относится к классу алгоритмов градиентного спуска?
4. Как влияет функция принадлежности на правило изменения весов в обратном алгоритме распространения ошибки?
5. Какая функция в среде MATLAB создает НС прямого распространения?
6. Какие функции активации могут быть назначены для нейронов НС прямого распространения?

ЛАБОРАТОРНАЯ РАБОТА № 3
ИЗУЧЕНИЕ РАДИАЛЬНЫХ БАЗИСНЫХ, ВЕРОЯТНОСТНЫХ НЕЙРОННЫХ СЕТЕЙ, СЕТЕЙ РЕГРЕССИИ

1 ЦЕЛЬ РАБОТЫ

Изучить модель вычислений радиального базисного нейрона, структуру и функции сетей регрессии, вероятностных нейронных сетей.

2 СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1 Радиально-базисные сети. Сети регрессии. Вероятностные НС

Рассмотрим радиальный базисный нейрон с R входами. Структура нейрона представлена на рис. 8. Радиальный базисный нейрон (РБН) вычисляет расстояние между векторами входов X и вектором весов W, затем умножает его на фиксированный порог b. Функция активации РБН, полученная в среде MATLAB, представлена на рис. 10. Радиальная базисная функция имеет максимум, равный 1, когда ее входы нулевые. Следовательно, радиальный базисный нейрон действует как детектор, который получает на выходе 1, когда вход X идентичен его вектору весов W. Фиксированный порог b даст возможность управлять чувствительностью нейрона. Например, если нейрон имеет порог 0,1, то выход равен 0,5 для любого входного вектора X, находящегося на векторном расстоянии 8,326 (8,326/b) от W.

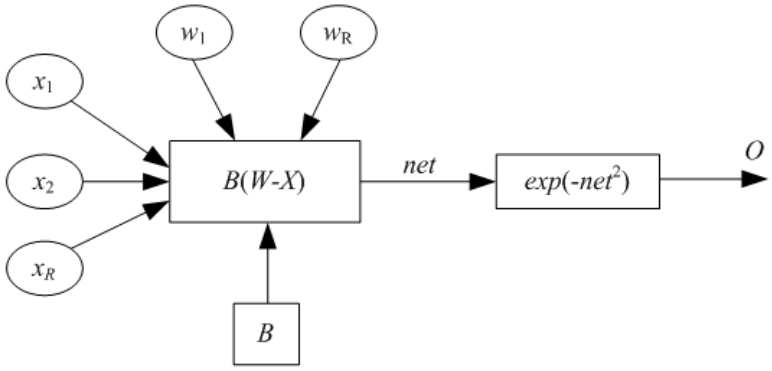

Рис. 8. Радиальный базисный нейрон

Радиальная базисная нейронная сеть (РБНС) состоит из двух слоев: скрытого радиального базисного слоя из S^1 нейронов и выходного линейного слоя из S^2 нейронов. Элементы первого слоя РБНС вычисляют расстояния между входным вектором и векторами весов первого слоя, сформированных из строк матрицы $W^{2,1}$. Вектор порогов B и расстояния поэлементно умножаются. Выход первого слоя можно выразить формулой

$$A^1 = radbas(\|W - X\| \cdot B),$$

где A^1 – выход первого слоя; функция *radbas* – радиально-базисная функция; W – матрица весов первого слоя сети; X – входной вектор; B – вектор порогов первого слоя.

Согласно формуле радиальные базисные нейроны с вектором весов, близким к X, сгенерируют значения, близкие к 1. Если нейрон имеет выход 1, то это значение весами второго слоя будет передано на его линейные нейроны. Фактически радиальный базисный нейрон с выходом 1 превращает выходы всех остальных нейронов в нули. Тем не менее, типичным является случай, когда несколько нейронов дают на выходах значимый результат, хотя и с разной степенью.

Радиальные базисные нейронные сети обучаются в три этапа. Опишем этапы обучения.

Первый этап – выделение центров (весов). Центры, представленные в РБН-слое, оптимизируются первыми с помощью обучения без учителя. Центры могут быть выделены разными алгоритмами, в частности обучением Кохонена. Алгоритмы должны разместить центры, отражая кластеризацию исходных данных.

Второй этап – назначение отклонений. Отклонения могут быть назначены различными алгоритмами, например алгоритмом «ближайшего соседа».

Третий этап – линейная оптимизация. Можно использовать методы обучения по дельта-правилу, обратному распространению ошибки.

Нейронные сети регрессии (НСР) имеют такой же, как и РБНС, первый слой, но второй слой строится специальным образом. Для аппроксимации функций часто используются обобщенные сети регрессии (generalized regression neuron networks). Второй слой, как и в случае РБНС, выполняет поэлементное произведение строки $W_{1,2}$ и вектора выхода первого слоя a^1. Он имеет столько нейронов, сколько существует целевых пар <входной вектор/целевой вектор>. Матрица весов W – это набор целевых строк. Целевое значение – это значение аппроксимируемой функции в обучающей выборке. Предположим, имеется один входной вектор x_i, который сгенерирует на выходе первого слоя выход, близкий к 1. В результате выход второго слоя будет близок к t_i одному из значений аппроксимируемой функции, использованной при формировании второго слоя.

Сети регрессии иногда называют *Байесовскими вероятностными сетями регрессии,* или обобщенными НС регрессии. Некоторые реализации сетей регрессии имеют четыре слоя: входной, выходной, слои радиальных центров, элементов регрессии. Радиальный слой представляет собой центры-кластеры известных обучающих

данных и содержит такое же количество элементов, как обучающая выборка; РБН обучаются алгоритмом кластеризации. Слой регрессии имеет только на один элемент больше, чем выходной слой, и содержит линейные элементы одного из двух типов. Элемент первого типа вычисляет условную вероятность каждого выходного атрибута, элемент второго типа вычисляет плотность вероятности. Выходной слой выполняет специальные функции деления. Каждый элемент делит выходы, ассоциированные первым типом, с помощью элементов второго типа.

Байесовские вероятностные НС используются только для проблем классификации. Они содержат четыре слоя: входной, выходной, слой РБН и элементов линейной классификации. Слои могут содержать квадратную матрицу потерь, включение которой возможно, только если третий и четвертый слои состоят из одинакового числа элементов. Радиальные базисные нейроны в таких сетях используются для хранения образцов, взятых из обучающей выборки, которая берется полностью. Следовательно, первый скрытый слой содержит такое же количество элементов, что и обучающая выборка. Так как элементы слоя классификации связаны с выходом каждого класса, можно оценить вероятность принадлежности последнему. Если используется матрица потерь, то цена решения минимальна. Такие сети обычно быстро тренируются, но медленно вычисляют из-за большого размера.

Вероятностные нейронные сети (ВНС, probabilistic neuron networks) используются для решения проблемы классификации. Первым слоем в архитектуре ВНС является слой радиальных базисных нейронов, который вычисляет расстояние и вектор индикаторов принадлежности другим входным векторам, используемым при обучении. Второй слой суммирует эти значения для каждого класса входов и формирует выходы сети, как вектор вероятностей. Далее специальная функция активации (compete) определяет максимум вероятностей на выходе второго слоя и устанавливает данный выход в 1, а остальные выходы в 0. Матрица весов первого слоя $W^{1,1}$ установлена в соответствии с обучающими парами. Блок расчета расстояний получает вектор, элементы которого показывают, насколько близок входной вектор к векторам обучающего множества. Элементы вектора умножаются на вектор порогов и преобразуются радиальной базисной функцией. Входной вектор, близкий к некоторому образцу, устанавливается в 1 в выходном векторе первого слоя. Если входной вектор близок к нескольким образцам отдельного класса, то несколько элементов выходного вектора первого слоя будут иметь значения, близкие к 1.

Веса второго слоя $W^{1,1}$ устанавливаются по матрице T целевых векторов, каждый вектор которой включает значение 1 в строке, связанной с определенным классом входов, и нули в остальных позициях. Произведения $T\,a^1$ суммируют элементы выходного вектора первого слоя a^1 для каждого из K классов. Затем функция активации второго слоя (compete) установит значение 1 в позицию, соответствующую большему элементу выходного вектора, и 0 во все остальные. Следовательно, сеть классифицирует входные векторы, назначая входу единственный класс на основе максимальной вероятности принадлежности.

2.2 Описание основных функций

Функция *newrb* создает радиальную базисную сеть и имеет следующий синтаксис:

net = newrb(P, T, goal, spread).

Радиальные базисные сети используют для аппроксимации функций. Функция *newrb* конструирует скрытый (первый) слой из радиальных базисных нейронов и использует значение средней квадратичной ошибки *(goal)*. Функция *newrb(P, T, goal, spread)* имеет следующие аргументы: *P* – матрица *Q* входных векторов размерности *R* на *Q*; *T* – матрица *Q* векторов целевых классов *S* на *Q; goal* – средняя квадратичная ошибка, по умолчанию 0,0; *spread* – разброс радиальной базисной функции, по умолчанию 1,0. Функция создает и возвращает в качестве объекта радиальную базисную сеть. Большое значение разброса приводит к большей гладкости аппроксимации. Слишком большой разброс требует много нейронов, для того чтобы подстроиться под быстро изменяющуюся функцию, слишком малый – для достижения гладкости аппроксимации. Подобрать значение разброса можно с помощью многократных вызовов функции *newrb*. Создадим в среде MATLAB радиальную базисную сеть:

net = newrbe(P, T, spread).

Функция *newrbe* проектирует радиальную базисную сеть с нулевой ошибкой для заданных векторов. Функция *newrbe(P, T, spread)* имеет три параметра: *P* – матрица *Q* входных векторов размерности *R* на *Q*; *T* – матрица *Q* целевых векторов – описателей класса размерности *S* на *Q*; *spread* – разброс радиальной базисной функции, по умолчанию 1,0. Функция создает радиальную базисную сеть.

Функция ***newgrnn*** проектирует НС регрессии – это вид радиальной базисной сети, которая часто используется для аппроксимации функций и быстро строит сеть для аппроксимации:

net = newgrnn(P, T, spread).

Функция *newgrnn(P, T, spread)* имеет следующие входы: *P* – матрица *Q* входных векторов размерности *R* на *Q*; *T* – матрица *Q* целевых векторов классов размерности *S* на *Q*; *spread* – разброс радиальных базисных функций, по умолчанию 1,0. Функция возвращает НС регрессии. Чем больше разброс, тем более гладкой будет функция аппроксимации. Для того чтобы настроить функцию аппроксимации на исходные данные, используют разброс меньший, чем типичное расстояние между входными векторами.

Функция *newgrnn* создает двухслойную сеть. Первый слой содержит радиальные базисные нейроны, вычисляющие расстояние между входами и весами с помощью *netprod*. Второй слой имеет нейроны с функцией активации *purelin*. Только у первого слоя существует *bias*.

Функция ***newpnn*** создает вероятностную НС, проектируемую с помощью функции

net = newpnn(P, T, spread).

Вероятностная НС – это вид радиальной базисной сети, как и НС регрессии, но данные сети используются для решения задачи классификации, а не аппроксимации. Функция *net = newpnn(P, T, spread)* имеет такие же параметры, как и вышеописанная функция *newgrnn*. Если разброс близок к нулю, вероятностная НС действует как классификатор на основе принципа выбора «ближайшего соседа», в противном случае сеть принимает в расчет несколько близких векторов.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

ПРИМЕР 1 Создание и обучение НС регрессии

Рассмотрим проектирование НС регрессии. Определим:

```
% входы НС регрессии
P = [1 2 3];
% выходы НС регрессии
T = [3.0 5.1 4.8];
```

Воспользуемся функцией *newgrnn* для создания НС регрессии. Зададим разброс радиальных базисных функций (переменная *spread)* меньше, чем 1, для того чтобы получить хорошую аппроксимацию данных и более гладкую функцию.

```
% установка разброса радиальных базисных функций
spread = 0.8;
% создание НС регрессии
net = newgrnn(P, T, spread);
% имитация работы НС регрессии
A = sim (net, P);
% изображение аппроксимируемой функции
plot(P, T, '.', 'markersize', 30);
% установка режима добавления графиков на координатные оси
hold on;
% изображение работы необученной НС регрессии
plot(P, A, '.', 'markersize', 30, 'color', [1 0 0]);
% очистка координатных осей
cla reset;
% установка нового входа НС регрессии
p = 4.5;
% получение отклика НС регрессии
a = sim (net, p);
% изображение аппроксимируемой функции
plot(P, T, '.', 'markersize', 30);
% установка диапазонов осей X и Y
axis ([0 9 -1 4]);
% включение режима добавления графиков
 hold on;
% изображение отклика НС регрессии на вход p
plot(p, a, '.', 'markersize', 30, 'color', [1 0 0]);
% написать заголовок графика
title('Новый входной вектор ');
% пометить ось X
xlabel('P и p');
```

```
% пометить ось Y
ylabel('T и a');
% очистить координатную сетку
cla reset;
% определить последовательность входов P2
P2 = 0: 0.2 : 9;
```

Сымитируем отклик сети для различных значений, чтобы увидеть результат аппроксимации (рис. 9).

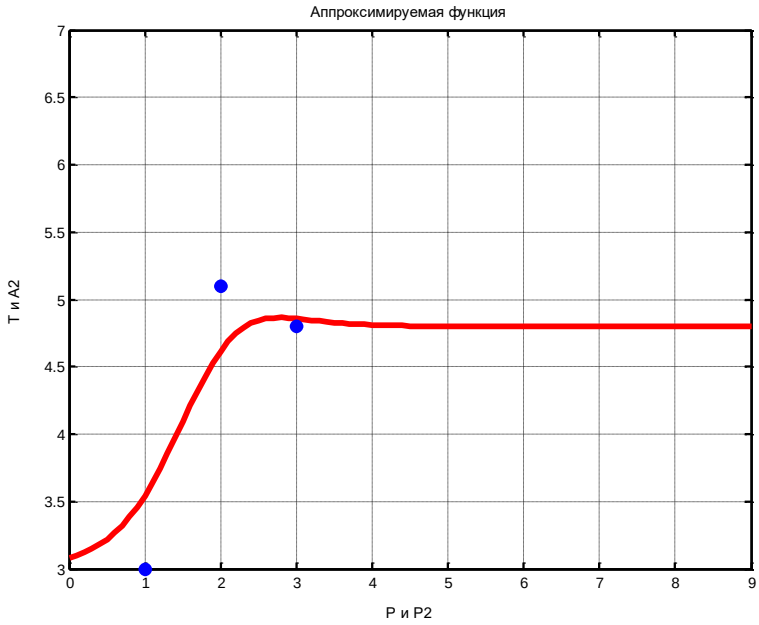

Рис. 9. Аппроксимация точек с помощью нейронной сети регрессии

```
% получить отклик НС регрессии на последовательность входов P2
A2 = sim (net, P2);
% изобразить отклик НС регрессии
plot(P2, A2, 'linewidth', 4, 'color', [1 0 0]);
% включить режим добавления графиков
hold on;
% изобразить аппроксимируемую функцию
plot(P, T, '.', 'markersize', 30);
% установить диапазон осей
axis ([0 9 3 7]);
% озаглавить график
title ('Аппроксимируемая функция');
% пометить ось X
 xlabel('P и P2');
% пометить ось Y
ylabel('T и A2');
```

ПРИМЕР 2 Использование РБНС для аппроксимации функций

Рассмотрим аппроксимацию функций на основе радиальной базисной сети.

```
% определение диапазона значений радиальной базисной функции
P = -4 : 0.1 : 4;
% вычисление радиальной базисной функции на диапазоне P
a = radbas (P);
% изображение РБФ
plot (P, a);
% озаглавить  график
title('Радиальная базисная функция');
```

```
% пометить ось X
xlabel('Вход P');
% пометить ось Y
ylabel ('Выход a');
```

На рис. 10 изображена радиальная базисная функция.

Функция *newrbe* создаст необходимую сеть. Зададим аппроксимируемую функцию как набор точек:

```
% определение последовательности аргументов
% аппроксимируемой функции P
P = -1 : 0.1 : 1;
% определение значений аппроксимируемой функции,
% соответствующих P
T = [-.6662 -.3766 -.1129 .2111 .6565 .3301 .3649 .2006...
-.1913 -.3994 -.5022 -.4531 -.1133 .0866 .3333 .4955...
.3488 .2833 -.1112 -.6685 -.3255];
```

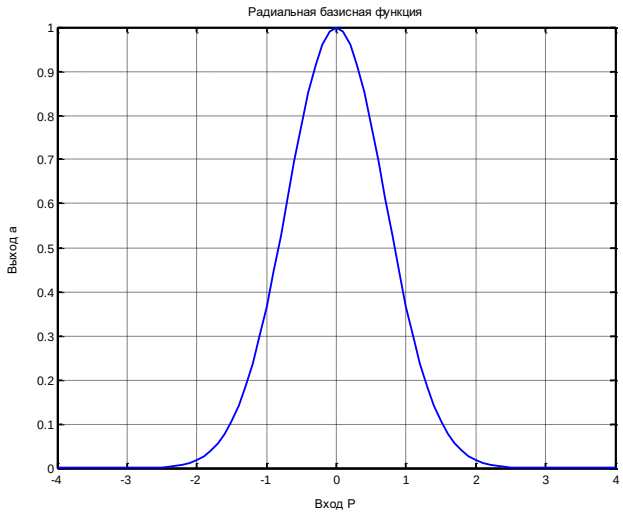

Рис. 10. Радиальная базисная функция

Изобразим график функции (рис. 11).

```
% изображение аппроксимируемой функции
plot(P, T, '*');
% озаглавить график
title('Обучающая выборка');
% пометить ось X
xlabel('Входной вектор P');
% пометить ось Y
ylabel('Вектор T');
```

Далее необходимо найти функцию, которая хорошо описывает заданные 21 точку.
Функция *newrb* создает РБНС для аппроксимации:

```
% целевой среднеквадратичной ошибки
e = 0.02;
% разброса РБФ
sp = 1;
% создание РБФ
net = newrb(P,T,e,sp);
% определение вектора входов
X = -1 : 0.01 : 1;
```

Сымитируем работу сети.

% формирование отклика Y
Y = sim(net, X);
% включение режима добавления графика
hold on;
% изображение результатов аппроксимации
plot(X, Y);
% отключение режима добавления графика
hold off;

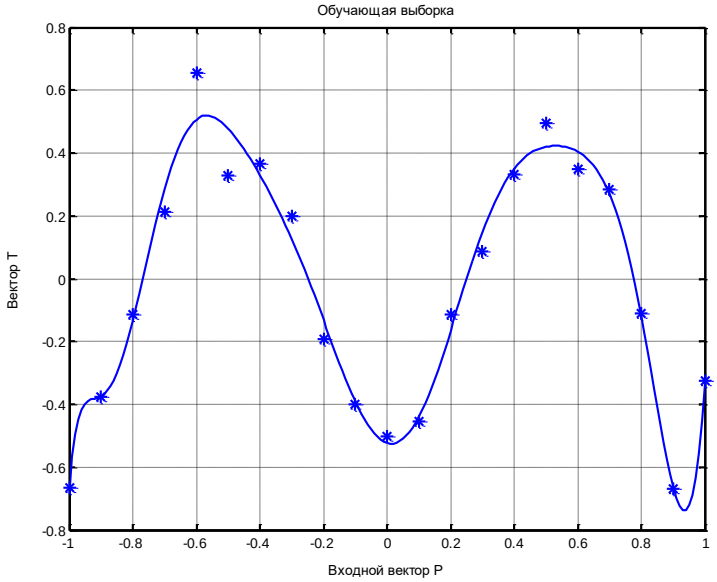

Рис. 11. Результат аппроксимации с помощью радиальной базисной нейронной сети

ПРИМЕР 3 Использование вероятностной НС для классификации векторов

Рассмотрим задачу классификации с набором входов *P* и множеством классов, обозначенным *Tc*.

% определение входов
P = [1 2 3 4 5 6 7];
% определение желаемых выходов
Tc =[1 1 3 3 2 1 1];
% конвертирование индексов в векторы, содержащие 1
% в индексных позициях
T = ind2vec (Tc);
% создание вероятностной НС
net = newpnn (P, T);
% имитация работы вероятностной НС
Y = sim(net, P);
% конвертирование номеров классов в векторы.
Yc = vec2ind(Y);

4.1.3 Тестирование
Тема 1. Нейронные сети. Основные понятия
Тема 2. Многослойные нейронные сети
Тема 3. Радиальные нейронные сети
Тема 4. Самоорганизующиеся нейронные сети

4.1.3.1. Порядок проведения и процедура оценивания
Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется

фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий. Ниже приведены примерные задания.

4.1.3.2. Критерии оценивания

Оценка «отлично» ставится, если обучающийся:
Дал 86% правильных ответов и более.
Оценка «хорошо» ставится, если обучающийся:
Дал От 71% до 85 % правильных ответов.
Оценка «удовлетворительно» ставится, если обучающийся:
Дал От 56% до 70% правильных ответов.
Оценка «не удовлетворительно» ставится, если обучающийся:
Дал 55% правильных ответов и менее.

4.1.3.3. Содержание оценочного средства

Вариант 1

1. Что является основным строительным блоком нейронной сети?
   а) Нейрон
   б) Синапс
   в) Аксон
   г) Дендрит
2. Какая функция активации обычно используется в выходном слое для задач классификации?
   а) ReLU
   б) Sigmoid
   в) Tanh
   г) Softmax

3. Что такое обратное распространение ошибки (backpropagation)?
   а) Метод обучения нейронной сети
   б) Тип нейрона
   в) Функция активации
   г) Архитектура сети
4. Что такое эпоха (epoch) в обучении нейронной сети?
   а) Один проход по всему набору данных
   б) Один проход по одному примеру данных
   в) Один шаг градиентного спуска
   г) Время обучения сети
5. Что такое переобучение (overfitting)?
   а) Модель слишком хорошо обобщает данные
   б) Модель плохо обобщает данные
   в) Модель не обучается
   г) Модель обучается слишком быстро
6. Что такое недообучение (underfitting)?
   а) Модель слишком хорошо обобщает данные
   б) Модель плохо обобщает данные
   в) Модель не обучается
   г) Модель обучается слишком быстро
7. Для чего используется функция потерь (loss function)?
   а) Измерение ошибки модели
   б) Активация нейронов
   в) Обучение сети
   г) Генерация данных
8. Что такое градиентный спуск?
   а) Алгоритм оптимизации
   б) Функция активации
   в) Тип нейронной сети
   г) Метод проверки модели

9. Какие слои обычно присутствуют в сверточной нейронной сети (CNN)?
   а) Сверточные слои и полносвязные слои
   б) Только полносвязные слои
   в) Только сверточные слои
   г) Рекуррентные слои
10. Что такое pooling в CNN?
   а) Снижение размерности данных
   б) Увеличение размерности данных
   в) Извлечение признаков
   г) Активация нейронов
11. Что такое рекуррентная нейронная сеть (RNN)?
   а) Сеть с циклическими связями
   б) Сеть без связей
   в) Сеть только с полносвязными слоями
   г) Сеть только со сверточными слоями
12. Для чего обычно используются RNN?
   а) Обработка последовательностей данных
   б) Классификация изображений
   в) Классификация текстов
   г) а и в
13. Что такое LSTM?
   а) Тип рекуррентной нейронной сети
   б) Тип сверточной нейронной сети
   в) Функция активации
   г) Алгоритм оптимизации
14. Что такое параметр learning rate?
   а) Скорость обучения модели
   б) Размер мини-пакета
   в) Количество эпох
   г) Размер сети
15. Что такое регуляризация?
   а) Метод предотвращения переобучения
   б) Метод ускорения обучения
   в) Метод улучшения точности
   г) Метод увеличения размера сети
16. Что такое Dropout?
   а) Метод регуляризации
   б) Функция активации
   в) Тип нейронной сети
   г) Алгоритм оптимизации
17. Что такое активационная функция?
   а) Функция, которая добавляет нелинейность в модель
   б) Функция, которая линейно преобразует данные
   в) Функция, которая измеряет ошибку
   г) Функция, которая оптимизирует веса
18. Что такое весовые коэффициенты (weights) в нейронной сети?
   а) Параметры, которые определяют силу связей между нейронами
   б) Входные данные
   в) Выходные данные
   г) Функция активации
19. Что такое bias (смещение) в нейронной сети?
   а) Постоянное значение, добавляемое к взвешенной сумме входов нейрона
   б) Случайное значение
   в) Вес связи
   г) Функция активации
20. Что такое мини-пакет (mini-batch) в градиентном спуске?
   а) Подмножество тренировочных данных

б) Весь тренировочный набор данных
в) Один пример тренировочных данных
г) Выходные данные модели

Ответы: 1-а, 2-г, 3-а, 4-а, 5-б, 6-б, 7-а, 8-а, 9-а, 10-а, 11-а, 12-г, 13-а, 14-а, 15-а, 16-а, 17-а, 18-а, 19-а, 20-а

Вариант 2

1. Для решения какой задачи лучше подходит сверточная нейронная сеть (CNN)?
   а) Обработка текста
   б) Распознавание речи
   в) Классификация изображений
   г) Предсказание временных рядов
2. Для решения какой задачи лучше подходит рекуррентная нейронная сеть (RNN)?
   а) Классификация изображений
   б) Распознавание речи
   в) Классификация текстов
   г) б и в
3. Какие типы данных обычно используются для обучения нейронных сетей?
   а) Численные данные
   б) Текстовые данные
   в) Изображения
   г) Все вышеперечисленное
4. Что такое трансферное обучение (transfer learning)?
   а) Использование предварительно обученной модели для решения новой задачи
   б) Обучение модели с нуля
   в) Передача данных между моделями
   г) Изменение архитектуры сети
5. Что такое генеративно-состязательные сети (GAN)?
   а) Два нейронных сети, которые соревнуются друг с другом
   б) Один нейронный сеть
   в) Сеть с циклическими связями
   г) Сеть без связей
6. Для чего обычно используются GAN?
   а) Генерация новых данных
   б) Классификация данных
   в) Обработка естественного языка
   г) а и б
7. В какой области применяются нейронные сети?
   а) Распознавание образов
   б) Обработка естественного языка
   в) Финансовое моделирование
   г) Все вышеперечисленное
8. Что такое автоэнкодер (autoencoder)?
   а) Нейронная сеть, которая учится кодировать и декодировать данные
   б) Нейронная сеть для классификации
   в) Нейронная сеть для регрессии
   г) Нейронная сеть для обработки временных рядов
9. Для чего обычно используется автоэнкодер?
   а) Сжатие данных
   б) Генерация данных
   в) Выявление аномалий
   г) Все вышеперечисленное
10. Что такое нейронная сеть прямого распространения (feedforward neural network)?
    а) Сеть без циклических связей
    б) Сеть с циклическими связями

в) Рекуррентная сеть
г) Сверточная сеть

11. Какая нейронная сеть лучше всего подходит для анализа последовательностей, таких как текст или звук?
    а) CNN
    б) RNN
    в) Feedforward NN
    г) Autoencoder
12. Что такое метрика точности (accuracy)?
    а) Процент правильно классифицированных образцов
    б) Среднеквадратичная ошибка
    в) Лосс-функция
    г) Время обучения
13. Какие методы используются для оценки производительности нейронных сетей?
    а) Метрики точности
    б) Кривые обучения
    в) Перекрестная проверка
    г) Все вышеперечисленное
14. Что такое перекрестная проверка (cross-validation)?
    а) Метод оценки обобщающей способности модели
    б) Метод обучения модели
    в) Метод оптимизации
    г) Метод регуляризации
15. Для решения задачи классификации изображений, какая из следующих архитектур наиболее подходит?
    а) Рекуррентная нейронная сеть
    б) Полносвязная нейронная сеть
    в) Сверточная нейронная сеть
    г) Автоэнкодер
16. Что такое гиперпараметры нейронной сети?
    а) Параметры, которые настраиваются во время обучения
    б) Параметры, которые устанавливаются до обучения
    в) Входные данные для сети
    г) Выходные данные сети
17. Как можно улучшить производительность нейронной сети?
    а) Изменение архитектуры
    б) Подбор гиперпараметров
    в) Добавление данных
    г) Все вышеперечисленное
18. Что такое word embedding?
    а) Представление слов в виде векторов
    б) Представление предложений в виде векторов
    в) Представление изображений в виде векторов
    г) Представление звука в виде векторов
19. Для чего используются word embeddings?
    а) Обработка естественного языка
    б) Обработка изображений
    в) Обработка звука
    г) а и б
20. Что такое нейронная сеть с обратной связью?
    а) Сеть, в которой выходные данные влияют на последующие входы
    б) Сеть без обратных связей
    в) Сверточная нейронная сеть
    г) Полносвязная нейронная сеть

Ответы: 1-в, 2-г, 3-г, 4-а, 5-а, 6-а, 7-г, 8-а, 9-г, 10-а, 11-б, 12-а, 13-г, 14-а, 15-в, 16-б, 17-г, 18-а, 19-а, 20-а

4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какую функцию называют радиальной базисной функцией?
2. Из каких слоев состоит радиально-базисная НС?
3. Из каких слоев состоит НС регрессии?
4. Из каких слоев состоит вероятностная НС?
5. Какие виды НС предназначены для решения задачи аппроксимации функций, а какие – для классификации объектов?
6. Какие типы НС создаются в среде MATLAB с помощью функций *newrb, newrbe, newgrnn, newpnn*?

4.2. Оценочные средства промежуточной аттестации

По дисциплине предусмотрен зачет. Зачет проходит по билетам. В каждом билете два теоретических вопроса. Зачет проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций.

4.2.1Зачет

4.2.1.1. Порядок проведения.

Устный или письменный ответ на вопрос направлен на проверку знаний основных разделов по дисциплине «Нейронные сети».

4.2.1.2. Критерии оценивания.

Зачтено ставится, если обучающийся:

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован, освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала, умение формулировать свои мысли, обсуждать дискуссионные положения. Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован . Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения. Или тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

Не зачтено ставится, если обучающийся:

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные

4.2.1.3. Оценочные средства.

Вопросы для устного или письменного ответа

1. Биологические представления о нейроне.
2. Понятие нейронная сеть.
3. Искусственный нейрон.
4. Классификация нейронных сетей.
5. Архитектура нейронных сетей.
6. История нейрокомпьютерных вычислений.
7. Персептрон.
8. Задача распознавания.
9. Понятие многослойной нейронной сети.
10. Принципы построения многослойных нейронных сетей.
11. Алгоритм обратного распространения ошибки.
12. Недостатки алгоритма обратного распространения ошибки.
13. Понятие радиальной нейронной сети.
14. Структура радиальной нейронной сети.
15. Математические основы функционирования радиальных нейронных сетей.
16. Обучение радиальной нейронной сети.
17. Самоорганизующиеся нейронные сети. Структура сети Кохонена. Обучение сети Кохонена.
20. Нейронные сети классификации.

Краткие ответы:

1. Биологические представления о нейроне.

Нейрон – это основная структурная и функциональная единица нервной системы. Он состоит из тела клетки (сомы), дендритов и аксона. Дендриты – это разветвлённые отростки, принимающие сигналы от других нейронов. Аксон – это длинный отросток, передающий сигналы к другим нейронам или мышцам. Сигналы передаются через синапсы – специализированные контакты между нейронами. Сила сигнала определяется силой синаптической связи, которая может изменяться в зависимости от активности нейрона. Этот процесс называется синаптической пластичностью и лежит в основе обучения

и памяти в биологической нервной системе. Биологические нейроны работают с помощью сложных электрохимических процессов. Потенциал действия – это электрический сигнал, распространяющийся по аксону. Когда потенциал действия достигает синапса, он вызывает высвобождение нейротрансмиттеров, которые влияют на активность других нейронов. Взаимодействие большого количества нейронов формирует сложные сети, ответственные за обработку информации и управление поведением. Изучение биологических нейронов вдохновило создание искусственных нейронных сетей.

2. Понятие нейронная сеть.

Нейронная сеть – это вычислительная модель, вдохновлённая структурой и функцией биологической нервной системы. Она состоит из множества взаимосвязанных простых процессирующих элементов – искусственных нейронов – организованных в слои. Эти слои обычно подразделяются на входной, один или несколько скрытых и выходной. Каждый нейрон принимает сигналы от нейронов предыдущего слоя, обрабатывает их и передаёт результат нейронам следующего слоя. Обработка информации происходит благодаря взвешенным связям между нейронами. Веса этих связей изменяются в процессе обучения сети. Обучение заключается в настройке весов таким образом, чтобы сеть максимально точно выполняла поставленную задачу. Нейронные сети способны распознавать образы, обрабатывать текст, предсказывать события и выполнять множество других сложных задач. Их способность к обучению и адаптации делает их мощным инструментом в различных областях.

3. Искусственный нейрон.

Искусственный нейрон – это математическая модель, имитирующая работу биологического нейрона. Он принимает на вход несколько сигналов (x1, x2, …, xn), каждый из которых умножается на свой вес (w1, w2, …, wn). Полученные произведения суммируются, к сумме добавляется смещение (bias), и результат подаётся на функцию активации (f). Функция активации вводит нелинейность в модель, позволяя сети аппроксимировать сложные нелинейные зависимости. Результат работы функции активации является выходным сигналом искусственного нейрона. Простые искусственные нейроны, объединённые в сложную структуру, позволяют создавать мощные вычислительные системы – нейронные сети. Различные функции активации (sigmoid, ReLU, tanh и др.) обладают различными свойствами и подходят для решения разных задач. Выбор функции активации – важный этап в проектировании нейронных сетей.

4.Классификация нейронных сетей.

Классификация нейронных сетей основана на их архитектуре, методе обучения и применении. Персептроны - простейший тип. Многослойные персептроны (MLP) содержат скрытые слои. Свёрточные сети (CNN) обрабатывают изображения. Рекуррентные сети (RNN) анализируют последовательности. LSTM и GRU - подтипы RNN для длинных последовательностей. Радиально-базисные функции (RBF) аппроксимируют функции. Самоорганизующиеся карты (SOM) кластеризуют данные. Генеративно-состязательные сети (GAN) создают новые данные. Обучение бывает с учителем, без учителя и с подкреплением. Выбор архитектуры зависит от задачи. CNN для изображений, RNN для текста. MLP универсальны. RBF для аппроксимации, SOM для кластеризации, GAN для генерации.

5.Архитектура нейронных сетей.

Архитектура нейронной сети определяет её структуру. Нейроны – базовые единицы. Они организованы в слои. Связи соединяют нейроны, а веса определяют силу связей. Входной слой принимает данные. Скрытые слои выполняют преобразования. Выходной слой формирует результат. Функции активации вносят нелинейность. ReLU, сигмоид, tanh – примеры функций активации. Глубокие сети имеют много скрытых слоёв. CNN используют свёртку, RNN – рекуррентные связи. Выбор архитектуры зависит от задачи и данных. Количество нейронов влияет на сложность модели.

6.История нейрокомпьютерных вычислений.

История нейросетей началась с модели МакКаллоха-Питтса. Розенблатт создал персептрон. Минский и Пейперт выявили ограничения персептрона. Алгоритм обратного распространения ошибки возродил интерес к нейросетям. Появились сети Хопфилда и карты Кохонена. В 90-х развивались методы обучения. Глубокое обучение стало доминирующим подходом в 2000-х. GPU и большие данные ускорили развитие. AlexNet показал эффективность глубоких сетей. ResNet, GAN, трансформеры – современные архитектуры. Исследования направлены на повышение эффективности и интерпретируемости. Нейроморфные и квантовые вычисления – будущее нейросетей.

7.Персептрон

Персептрон – ранная модель нейронной сети. Это линейный классификатор. Он состоит из входа, весов и выхода. Входы умножаются на веса. Сумма взвешенных входов подаётся на функцию активации. Обучение корректирует веса для минимизации ошибки. Персептрон ограничен линейно разделяемыми задачами. Минский и Пейперт описали эти ограничения. Несмотря на ограничения, персептрон важен для истории нейросетей. Он стал основой для более сложных моделей. Понимание персептрона помогает понять нейросети.

8.Задача распознавания

Задача распознавания – определить класс объекта. Распознают изображения, звук, текст. Распознавание образов выделяет объекты на изображениях. Распознавание речи преобразует звук в текст. Распознавание текста извлекает информацию. Распознавание рукописного текста переводит его в печатный. Используются нейросети, SVM, k-NN, деревья решений. CNN для изображений, RNN для речи и текста. SVM строят гиперплоскости. k-NN используют ближайших соседей. Деревья решений – это правила. Распознавание применяется в компьютерном зрении, обработке языка, биометрии.

9. Понятие многослойной нейронной сети. Многослойный персептрон (MLP) состоит из нескольких слоёв нейронов. MLP имеет входной, один или несколько скрытых и выходной слои. Нейроны связаны весами, которые настраиваются во время обучения. MLP может аппроксимировать нелинейные функции. Это делает их подходящими для решения сложных задач. MLP используется в распознавании образов, обработке естественного языка и прогнозировании. Каждый слой выполняет преобразование входных данных. Функции активации вносят нелинейность в модель. Обучение MLP направлено на минимизацию ошибки. Архитектура MLP определяется количеством слоёв и нейронов. Выбор архитектуры зависит от конкретной задачи. Глубокое обучение использует MLP с большим количеством скрытых слоёв. MLP – основа многих современных нейронных сетей. Обучение MLP может быть вычислительно затратным.

10. Принципы построения многослойных нейронных сетей. Выбор количества слоёв и нейронов – важный этап построения MLP. Количество слоёв определяет глубину сети. Количество нейронов в каждом слое влияет на capacity сети. Функции активации вводят нелинейность и влияют на обучение. ReLU, сигмоид и tanh – распространённые функции активации. Архитектура сети должна соответствовать сложности задачи. Слишком сложная сеть может переобучиться. Слишком простая сеть может недообучиться. Необходимо балансировать между сложностью и обобщающей способностью. Инициализация весов влияет на процесс обучения. Методы регуляризации помогают предотвратить переобучение. Скорость обучения – важный гиперпараметр. Оптимизаторы, такие как Adam и SGD, используются для настройки весов. Выбор архитектуры часто осуществляется эмпирически.

11. Алгоритм обратного распространения ошибки. Backpropagation – основной алгоритм обучения нейронных сетей. Он используется для вычисления градиента функции ошибки. Градиент указывает направление изменения весов для уменьшения ошибки. Алгоритм основан на цепном правиле дифференцирования. Ошибка распространяется обратно от выходного слоя к входному. Веса корректируются пропорционально градиенту ошибки. Скорость обучения определяет величину корректировки весов. Backpropagation позволяет эффективно обучать сложные сети. Алгоритм итеративно обновляет веса. Он является основой для многих методов глубокого обучения. Backpropagation может быть вычислительно затратным для больших сетей. Выбор подходящей скорости обучения важен для успешного обучения. Существуют различные модификации backpropagation.

12. Недостатки алгоритма обратного распространения ошибки. Backpropagation может сходиться к локальным минимумам. Это означает, что найденное решение не является глобально оптимальным. Скорость обучения может быть медленной, особенно для глубоких сетей. Алгоритм чувствителен к начальным значениям весов. Проблема исчезающего градиента может затруднить обучение глубоких сетей. Backpropagation требует больших вычислительных ресурсов. Выбор скорости обучения может быть сложной задачей. Слишком большая скорость обучения может привести к расходимости. Слишком маленькая скорость обучения может замедлить сходимость. Алгоритм чувствителен к выбросам в данных. Необходимо тщательно подбирать параметры алгоритма.

13. Понятие радиальной нейронной сети. Радиальные нейронные сети (RBF) используют радиально-базисные функции активации. RBF сети обычно состоят из трёх слоёв: входного, скрытого и выходного. Скрытый слой содержит RBF нейроны. RBF функции измеряют сходство входного вектора с центром

нейрона. Чем ближе вход к центру, тем больше выход нейрона. RBF сети эффективны для аппроксимации функций. Они могут использоваться для классификации и регрессии. RBF сети часто обучаются быстрее, чем MLP. Выбор RBF функции влияет на производительность сети. Гауссова функция – распространённый выбор RBF функции.

14. Структура радиальной нейронной сети. Входной слой RBF сети принимает входные данные. Скрытый слой содержит RBF нейроны. Каждый RBF нейрон имеет центр и ширину. Центр определяет область входного пространства, на которую реагирует нейрон. Ширина определяет степень влияния входного вектора. Выходной слой выполняет линейную комбинацию выходов RBF нейронов. Веса выходного слоя настраиваются во время обучения. Количество RBF нейронов определяет сложность модели. Выбор количества нейронов – важный этап построения RBF сети. Структура RBF сети проще, чем структура MLP.

15. Математические основы функционирования радиальных нейронных сетей. RBF функции вычисляют расстояние между входным вектором и центром нейрона. Обычно используется Евклидово расстояние. Ширина RBF функции определяет степень размытости вокруг центра. Гауссова функция – популярный выбор RBF функции. Выход RBF нейрона уменьшается с увеличением расстояния до центра. Выходной слой выполняет взвешенное суммирование выходов RBF нейронов. Веса выходного слоя определяют вклад каждого RBF нейрона в окончательный результат. Обучение RBF сети сводится к выбору центров, ширин и весов выходного слоя. Математический аппарат RBF сетей относительно прост. RBF сети хорошо подходят для аппроксимации функций.

16. Обучение радиальной нейронной сети. Обучение RBF сети включает определение центров RBF нейронов. Выбор ширины RBF функций также важен. Настройка весов выходного слоя завершает обучение. Центры могут выбираться случайно из обучающей выборки. Кластеризация данных может помочь определить центры. Ширина RBF функции определяет область влияния нейрона. Слишком узкие функции могут привести к переобучению. Слишком широкие функции могут привести к недообучению. Веса выходного слоя обычно обучаются с помощью метода наименьших квадратов. Градиентный спуск также может использоваться для настройки весов. Обучение RBF сети часто быстрее, чем обучение MLP. Правильный выбор параметров RBF сети важен для достижения хорошей производительности. Существуют различные алгоритмы обучения RBF сетей. Обучение может быть выполнено в один или два этапа. RBF сети хорошо подходят для задач аппроксимации функций.

17. Самоорганизующиеся нейронные сети. Самоорганизующиеся нейронные сети (SONN) обучаются без учителя. SONN способны выявлять структуру данных самостоятельно. Они используются для кластеризации, снижения размерности и визуализации. Карты Кохонена являются примером SONN. SONN находят скрытые закономерности в данных. Они могут обрабатывать большие объемы данных. Обучение SONN основано на конкуренции между нейронами. Нейроны конкурируют за право реагировать на входной стимул. Победитель корректирует свои веса, чтобы стать ближе к входному стимулу. Соседи победителя также корректируют свои веса. Этот процесс позволяет сети адаптироваться к структуре данных. SONN могут использоваться для предварительной обработки данных. Они могут помочь улучшить производительность других алгоритмов машинного обучения. SONN являются мощным инструментом для анализа данных.

18. Структура сети Кохонена. Сеть Кохонена (SOM) представляет собой двумерную решетку нейронов. Каждый нейрон имеет вектор весов той же размерности, что и входные данные. Нейроны расположены в узлах решетки. SOM сохраняет топологию входных данных. Близкие входные векторы активируют близкие нейроны на карте. Сеть Кохонена использует конкурентное обучение. Нейроны конкурируют за право быть "победителем" для данного входного вектора. Победитель - нейрон, вектор весов которого наиболее близок к входному вектору. Веса победителя и его соседей корректируются в процессе обучения. Размер карты Кохонена определяется пользователем. Большие карты позволяют более детально представить структуру данных. SOM может использоваться для визуализации многомерных данных.

19. Обучение сети Кохонена. Обучение сети Кохонена – итеративный процесс. На каждом шаге предъявляется входной вектор. Определяется нейрон-победитель, вектор весов которого наиболее близок к входному вектору. Веса победителя и его соседей корректируются. Корректировка весов основана на расстоянии до победителя. Чем ближе нейрон к победителю, тем сильнее корректируются его веса. Радиус соседства уменьшается в процессе обучения. Скорость обучения также уменьшается со

временем. Обучение продолжается до тех пор, пока веса не стабилизируются. Обученная сеть Кохонена отображает входные данные на двумерную карту. Карта сохраняет топологическую структуру данных.

20. Нейронные сети классификации. Нейронные сети классификации используются для решения задач классификации. Они назначают входные данные одному из predefined классов. MLP, CNN, RNN могут использоваться для классификации. Выходной слой сети классификации обычно имеет столько нейронов, сколько классов. Функция softmax часто используется на выходном слое. Softmax преобразует выходы нейронов в вероятности принадлежности к классам. Класс с наибольшей вероятностью выбирается в качестве предсказания. Обучение сети классификации выполняется с помощью размеченных данных. Размеченные данные содержат пары вход-выход. Цель обучения – минимизировать ошибку классификации. CNN эффективны для классификации изображений. RNN подходят для классификации последовательностей. Выбор архитектуры сети зависит от типа данных и сложности задачи.

Приложение 2
к рабочей программе дисциплины (модуля)
Б1.В.ДВ.02.01 Нейронные сети

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 15.03.06 Мехатроника и робототехника

Профиль подготовки: Физические основы мехатроники и робототехники

Квалификация выпускника: бакалавр

Форма обучения: заочная

Язык обучения: русский

Год начала обучения по образовательной программе: 2026

Основная литература:

1. Данилов, В. В. Нейронные сети : учебное пособие / В. В. Данилов. — Донецк : ДонГУ, 2020. — 158 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179953. — Режим доступа: для авториз. пользователей.
2. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 216 с. — ISBN 978-5-507-50568-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/447392. — Режим доступа: для авториз. пользователей.
3. Сириченко, А. В. Искусственные нейронные сети. Практикум : учебное пособие / А. В. Сириченко. — Москва : МИСИС, 2022. — 26 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/305447 . — Режим доступа: для авториз. пользователей.
4. Степанов, Ю. А. Системы искусственного интеллекта : учебное пособие / Ю. А. Степанов, А. В. Вылегжанина, Л. Н. Бурмин. — Кемерово : КемГУ, 2024. — 102 с. — ISBN 978-5-8353-3166-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/427532 . — Режим доступа: для авториз. пользователей.

Дополнительная литература

1. Осипов, Г. С. Методы искусственного интеллекта: монография / Г. С. Осипов. - Москва : Физматлит, 2011. - 296 с. - ISBN 978-5-9221-1323-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/544787 – Режим доступа: по подписке.
2. Сергеев, Н. Е. Системы искусственного интеллекта. Часть 1: Учебное пособие / Сергеев Н.Е. - Таганрог: Южный федеральный университет, 2016. - 118 с.: ISBN 978-5-9275-2113-5. - Текст: электронный. - URL: https://znanium.com/catalog/product/991954 . – Режим доступа: по подписке.
3. Теоретические основы информатики / Царев Р.Ю., Пупков А.Н., Самарин В.В [и др.]. - Красноярск: СФУ, 2015. - 176 с.: ISBN 978-5-7638-3192-4. - Текст: электронный. - URL: https://znanium.com/catalog/product/549801. – Режим доступа: по подписке.
4. Вышегуров, С. Х. Информатика [Электронный ресурс] : учеб. пособие / Новосиб. гос. аграр. ун-т. Агроном. фак.; сост.: И.И. Некрасова, С.Х. Вышегуров. - Новосибирск: Золотой колос, 2014. - 105 с. - Текст: электронный. - URL: https://znanium.com/catalog/product/516070. – Режим доступа: по подписке.
5. Ермакова, А.Н. Информатика [Электронный ресурс]: учебное пособие / А.Н. Ермакова, С.В. Богданова. - Ставрополь: Сервисшкола, 2013. - 184 с. - Текст: электронный. - URL: https://znanium.com/catalog/product/514863 – Режим доступа: по подписке.
6. Каймин, В. А. Информатика: Учебник / Каймин В. А. - 6-е изд. - Москва: НИЦ ИНФРА-М, 2015. - 285 с. (Высшее образование: Бакалавриат) ISBN 978-5-16-010876-6. - Текст: электронный. - URL: https://znanium.com/catalog/product/504525. – Режим доступа: по подписке.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.В.ДВ.02.01 Нейронные сети

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 15.03.06 -Мехатроника и робототехника

Профиль подготовки: Физические основы мехатроники и робототехники

Квалификация выпускника: бакалавр

Форма обучения: заочная

Язык обучения: русский

Год начала обучения по образовательной программе: 2026

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Программное обеспечение: операционная система Windows, Microsoft office, PyCharm, Kaspersky Free для Windows

Электронная библиотечная система «ZNANIUM.COM»

Электронная библиотечная система Издательства «Лань»

Электронная библиотечная система «Консультант студента»