

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Ижевский государственный технический университет имени М. Т. Калашникова»

Кафедра «Программное обеспечение»

**ИЗУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ,  
НЕЧЕТКОЙ ЛОГИКИ И ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ  
НА ОСНОВЕ МЕТОДА ПРОЕКТОВ**

Учебно-методическое пособие по дисциплинам  
«Нейрокомпьютерные системы»,  
«Нечеткая логика и генетические алгоритмы»,  
«Методы оптимизации. Нейронные сети»,  
«Системы искусственного интеллекта»,  
«Математические основы искусственного интеллекта»



Ижевск  
ИжГТУ имени М. Т. Калашникова  
2013

С о с т а в и т е л ь А. В. Коробейников, канд. техн. наук, доц. кафедры «Программное обеспечение» ИжГТУ

Рекомендовано к использованию на заседании кафедры «Программное обеспечение» ИжГТУ (протокол №37 от 15 октября 2013 г.).

**Изучение нейронных сетей, нечеткой логики и генетических алгоритмов на основе метода проектов:** учебно-методическое пособие по дисциплинам «Нейрокомпьютерные системы», «Нечеткая логика и генетические алгоритмы» «Системы искусственного интеллекта», «Математические основы искусственного интеллекта» / сост. А. В. Коробейников. – Ижевск : Изд-во ИжГТУ, 2013. – 53 с.

Пособие предназначено для выполнения курсовой работы по дисциплинам «Нейрокомпьютерные системы» и «Нечеткая логика и генетические алгоритмы» для магистрантов, обучающихся по направлению подготовки 231000.68 «Программная инженерия» по программам «Системы мультимедиа и компьютерной графики» и «Разработка программно-информационных систем».

Пособие предназначено для выполнения курсовой работы по дисциплинам «Математические основы искусственного интеллекта» для студентов, обучающихся по направлению подготовки 231000.68 «Программная инженерия».

Пособие предназначено для выполнения лабораторных работ и самостоятельной работы студентов, обучающихся по специальности 230105.65 «Программное обеспечение вычислительной техники и автоматизированных систем» по дисциплине «Нейрокомпьютерные системы» и направлению подготовки 230100.62 «Информатика и вычислительная техника» при изучении дисциплины «Методы оптимизации. Нейронные сети».

Пособие предназначено для выполнения лабораторных работ и самостоятельной работы студентов, обучающихся по направлению подготовки 230100.62 «Информатика и вычислительная техника» и специальности 230105.65 «Программное обеспечение вычислительной техники и автоматизированных систем» при изучении дисциплины «Системы искусственного интеллекта» очной и заочной форм обучения.

Пособие предназначено для выполнения курсовой работы студентов, обучающихся по направлению подготовки 231000.62 «Программная инженерия» по дисциплине «Математические основы искусственного интеллекта».

В пособии используются образовательные технологии: метод проектов, дистанционное обучение, обучение в малых группах. Методическое пособие может использоваться как совместно с разработанным в среде «Moodle» дистанционным курсом для студентов заочного или очного обучения, так для очных студентов без дистанционного курса.

© ФГБОУ ВПО «Ижевский государственный технический университет имени М. Т. Калашникова», 2013

© Коробейников А. В., 2013

## **Оглавление**

1. Введение	4
2. Выбор задачи	15
3. Выбор типа нейросети	20
4. Выбор метода обучения нейросети	31
5. Разработка программы	40
6. Проведение экспериментов	42
7. Оформление пояснительной записки	47
Приложение. Форма титульного листа пояснительной записки	52

# 1. Введение

## 1.1. Цель и задачи

Цель курсовой работы: решение практических задач с использованием нейронных сетей, нечеткой логики и генетических алгоритмов.

Задачи курсовой работы:

- 1) решать практические задачи с помощью нейросетей;
- 2) формировать обучающую выборку;
- 3) оценивать применимость различных типов нейросетей и методов обучения для решения определенной задачи;
- 4) разрабатывать топологию сети;
- 5) проводить эксперименты с разработанной нейросетью;
- 6) оценивать результаты обучения и работы нейросети;
- 7) получить опыт дистанционного образования;
- 8) получить опыт обучения на основе метода проектов;
- 9) получить опыт командной работы над решением задачи.

## 1.2. Теоретические основы

### 1.2.1. Нейронные сети

**Нейрокомпьютер.** Нейрокомпьютер – устройство переработки информации на основе принципов работы естественных нейронных систем. Эти принципы были формализованы, что позволило говорить о теории искусственных нейронных сетей. Цель направления исследования заключается в построении реальных физических устройств, что позволит не просто моделировать искусственные нейронные сети на обычном компьютере, но так изменить принципы работы компьютера, что станет возможным говорить о том, что они работают в соответствии с теорией искусственных нейронных сетей.

**История нейроинформатики.** Термины нейрокибернетика, нейроинформатика, нейрокомпьютеры вошли в научный обиход недавно – в середине 80-х годов XX века. Однако электронный и биологический мозг постоянно сравнивались на протяжении всей истории существования вычислительной техники. Знаменитая книга Н. Винера «Кибернетика» (1948) имеет подзаголовок «Управление и связь в животном и машине».

Первыми нейрокомпьютерами были перцептроны Розенблатта:

«Марк-1» (1958) и «Тобермори» (1961–1967), а также «Адалин», разработанный Уидроу и Хоффом (1960) на основе дельта-правила (формулы Уидроу). В настоящее время «Адалин» является стандартным элементом многих систем обработки сигналов и связи. В этом же ряду первых нейрокомпьютеров находится программа «Кора», разработанная в 1961 году под руководством Бонгарда.

Большую роль в развитии нейровычислений сыграла монография Розенблатта (1958).

Идея нейробионики (создания технических средств на нейросетевых принципах) стала интенсивно реализовываться в начале 1980-х годов. Импульсом было следующее противоречие: размеры элементарных деталей компьютеров сравнивались с размерами элементарных «преобразователей информации» в нервной системе, было достигнуто быстроедействие отдельных электронных элементов в миллионы раз большее, чем у биологических систем, а эффективность решения задач, особенно связанных задач ориентировки и принятия решений в естественной среде, у живых систем пока недостижимо выше.

Другой импульс развитию нейрокомпьютеров дали теоретические разработки 1980-х годов по теории нейронных сетей (сети: Хопфилда, Кохонена; алгоритмы обучения: обратное распространения ошибки, генетический и стохастический).

**Основная идея – коннекционизм.** В отличие от цифровых систем, представляющих собой комбинации процессорных и запоминающих блоков, нейропроцессоры содержат память, распределённую в связях между очень простыми процессорами, которые часто могут быть описаны как формальные нейроны или блоки из однотипных формальных нейронов. Тем самым основная нагрузка на выполнение конкретных функций процессорами ложится на архитектуру системы, детали которой в свою очередь определяются межнейронными связями. Подход, основанный на представлении как памяти данных, так и алгоритмов системой связей (и их весами), называется коннекционизмом.

Основные преимущества нейрокомпьютеров:

- 1) все алгоритмы нейроинформатики обладают высоким параллелизмом, а это уже залог высокого быстрогодействия;
- 2) нейросистемы можно легко сделать очень устойчивыми к помехам и разрушениям;

3) устойчивые и надёжные нейросистемы могут создаваться и из ненадёжных элементов, имеющих значительный разброс параметров.

Разработчики нейрокомпьютеров стремятся объединить устойчивость, быстродействие и параллелизм аналоговых вычислительных машин с универсальностью современных компьютеров.

**Применение нейросетей.** Необходимо решить три задачи:

1) определить архитектуру сети: нейроны какого типа используются; как они связаны между собой; какие нейроны взяты в качестве входов и выходов;

2) сформировать обучающую выборку; обучающий образец:  $X^k$  – входной образ,  $T^k$  – верный (требуемый, целевой) выход,  $k$  – номер образца;  $K$  – общее число образцов;

3) определить метод обучения нейросети:

– формализованный метод итеративной коррекции значений весов или непосредственное вычисление значений весов;

– обучение с учителем (известно  $T^k$ ) или без учителя (неизвестно  $T^k$ ).

### 1.2.2. Метод проектов

Метод проектов – это способ достижения учебных целей через детальную разработку проблемы, которая должна завершиться вполне реальным, осязаемым практическим результатом, оформленным тем или иным образом; это совокупность приёмов, действий учащихся в их определённой последовательности для достижения поставленной задачи — решения проблемы, лично значимой для учащихся и оформленной в виде некоего конечного продукта.

Основное предназначение метода проектов состоит в предоставлении учащимся возможности самостоятельного приобретения знаний в процессе решения практических задач или проблем, требующего интеграции знаний из различных предметных областей. Если говорить о методе проектов как о педагогической технологии, то эта технология предполагает совокупность исследовательских, поисковых, проблемных методов, творческих по своей сути. Преподавателю в рамках проекта отводится роль разработчика, координатора, эксперта, консультанта.

То есть, в основе метода проектов лежит развитие познавательных навыков учащихся, умений самостоятельно конструировать

свои знания, ориентироваться в информационном пространстве, развитие критического и творческого мышления.

Разработанный еще в первой половине XX века на основе прагматической педагогики Джона Дьюи метод проектов становится особенно актуальным в современном информационном обществе. Метод проектов не новость в мировой педагогике: он начал использоваться в практике обучения значительно раньше выхода в свет известной статьи американского педагога В. Килпатрика «Метод проектов» (1918), в которой он определил это понятие как «от души выполняемый замысел». В России метод проектов был известен еще в 1905 году. Под руководством С.Т. Шацкого работала группа российских педагогов по внедрению этого метода в образовательную практику. После революции метод проектов применялся в школах по личному распоряжению Н. К. Крупской до 1931 г.

### **1.2.3. Дистанционное обучение**

Дистанционное обучение (ДО) – взаимодействие учителя и учащихся и учащихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты (цели, содержание, методы, организационные формы, средства обучения) и реализуемое специфичными средствами интернет-технологий или другими средствами, предусматривающими интерактивность. ДО - это одна из форм обучения, в которой информационные технологии являются ведущим средством.

Современное дистанционное обучение строится на использовании следующих основных элементов:

- среды передачи информации (почта, телевидение, радио, информационные коммуникационные сети);
- методов, зависящих от технической среды обмена информацией.

В настоящее время перспективным является интерактивное взаимодействие с учащимся посредством информационных коммуникационных сетей, из которых массово выделяется среда интернет-пользователей. В 2003 году инициативная группа *ADL* начала разработку стандарта дистанционного интерактивного обучения *SCORM*, который предполагает широкое применение интернет-технологий. Введение стандартов способствует как углублению требований к составу дистанционного обучения, так и требований к программно-

му обеспечению (ПО). В настоящее время имеются отечественные разработки ПО, которые достаточно широко применяются как отечественными, так и зарубежными организациями, предоставляющими услуги по дистанционному обучению.

Дистанционное обучение позволяет:

- снизить затраты на проведение обучения (не требуется затрат на аренду помещений, поездок к месту учебы, как учащихся, так и преподавателей и т. п.);
- проводить обучение большого количества человек;
- повысить качество обучения за счет применения современных средств, объемных электронных библиотек и т.д.
- создать единую образовательную среду (особенно актуально для корпоративного обучения).

#### **1.2.4. Обучение в малых группах**

Обучение в малых группах или обучение в сотрудничестве (*cooperative learning*), широко используется в Западной Германии, Нидерландах, в Великобритании, Австралии, Израиле, Японии. Основная идея этой технологии - создать условия для активной совместной учебной деятельности учащихся в разных учебных ситуациях. Практика показывает, что вместе учиться не только легче и интереснее, но и значительно эффективнее. Причем важно, что эта эффективность касается не только академических успехов студентов, их интеллектуального развития, но и нравственного. Помочь другу, вместе решить любые проблемы, разделить радость успеха или горечь неудачи – также естественно, как смеяться, петь, радоваться жизни.

Существует много разнообразных вариантов обучения в сотрудничестве. Преподаватель в своей практике может разнообразить и эти варианты своим творчеством, применительно к своим студентам, но при одном обязательном условии: при четком соблюдении основных принципов обучения в сотрудничестве:

1) Группы учащихся формируются преподавателем до занятия, разумеется, с учетом психологической совместимости. При этом в каждой группе должен быть сильный студент, средний и слабый (если группа состоит из трех учащихся), девочки и мальчики. Если группа на протяжении ряда занятий работает слаженно, дружно, нет необходимости менять их состав (это так называемые базовые груп-



пы). Если работа по каким-то причинам не очень клеится, состав группы можно менять от занятия к занятию;

2) Группе дается одно задание, но при его выполнении предусматривается распределение ролей между участниками группы (роли обычно распределяются самими студентами, но в некоторых случаях преподаватель может дать рекомендации);

3) Оценивается работа не только одного студента, но и всей группы; важно, что оцениваются не только и иногда не столько знания, сколько усилия учащихся (у каждого своя «планка»);

4) Преподаватель сам выбирает студента группы, который должен отчитаться за задание. В ряде случаев это бывает слабый студент. Если слабый студент в состоянии обстоятельно доложить результаты совместной работы группы, ответить на вопросы других групп, значит, цель достигнута и группа справилась с заданием, ибо цель любого задания – не формальное его выполнение (правильное / неправильное решение), а овладение материалом каждым студентом группы.

Умение пользоваться обучением в сотрудничестве – показатель высокой квалификации преподавателя, его прогрессивной методики обучения и развития учащихся

#### **1.2.5. Автоматизированная обучающая система «Moodle»**

«Moodle» – система управления курсами, автоматизированная обучающая система (АОС), также известная как система управления обучением, виртуальная обучающая среда. Система «Moodle» – специализированная система управления содержимым (content management system – CMS). Представляет собой свободное (распространяющееся по лицензии *GNU GPL*) веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения.

Система реализует философию «педагогике социального конструкционизма» и ориентирована прежде всего на организацию взаимодействия между преподавателем и учениками, хотя подходит и для организации традиционных дистанционных курсов, а также поддержки очного обучения.

«Moodle» переведена на десятки языков, в том числе и русский и используется почти в 50 тысячах организаций из более чем 200 стран мира. Лидером и идеологом системы является *M. Dougiamas* из Австралии. Проект является открытым и в нем участвует и множество

других разработчиков. Русификацию «Moodle» осуществляет команда добровольцев из России, Белоруссии и Украины.

«Moodle» написана на *PHP* с использованием *SQL*-базы данных (*MySQL*, *PostgreSQL*, *Microsoft SQL Server* и др. БД – используется *ADODB XML*). «Moodle» может работать с объектами *SCO* и отвечает стандарту *SCORM*.

Благодаря развитой модульной архитектуре, возможности «Moodle» могут легко расширяться сторонними разработчиками. Помимо языковой поддержки и шаблонов оформления, «Moodle» позволяет подключать также различные типы модулей.

### **1.3. Программное обеспечение**

При выполнении курсовой работы используется ПО:

- 1) среда разработки ПО на языке высокого уровня, выбранная студентами (*freeware*);
- 2) пакеты моделирования нейросетей [2] (*freeware*);
- 3) ПО АОС «Moodle», содержащая дистанционный курс по на выполнению данной курсовой работы на основе метода проектов.

### **1.4. Порядок выполнения проекта**

#### **1.4.1. Организация команд**

Для выполнения проекта необходимо организовать команды по 2-3 студента. При выполнении проекта распределить работу по проекту на участников команды: у каждого участника проекта должна быть своя часть проекта либо все части выполняются студентами совместно. В защите проекта должны участвовать все студенты команды.

#### **1.4.2. Этапы выполнения проекта**

- 1) выбор задачи (модуль 1);
- 2) выбор типа нейросети (модуль 2);
- 3) выбор метода обучения нейросети (модуль 3);
- 4) разработка программы (модуль 4);
- 5) проведение экспериментов (модуль 5);
- 6) оформление пояснительной записки (модуль 6);
- 7) защита проекта.

### 1.4.3. Календарный план выполнения проекта

День	Модуль					
	1	2	3	4	5	6
1	1					
2						
3						
4						
5						
6						
7						
8		1				
9						
10						
11						
12						
13						
14						
15			1			
16						
17						
18						
19						
20						
21						
22				1		
23						
24						
25						
26						
27						
28						
29					1	
30						
31						
32						
33						
34						
35						
36						1
Итого	7	7	4	10	4	4

1 Вебинары с указанием продолжительности (час)  
 Самостоятельная работа над модулем

#### **1.4.4. Требования для получения оценки по проекту**

При выполнении курсовой работы на основе метода проектов с использованием дистанционного курса за каждый пункт практического задания и за каждый вопрос обсуждения в форуме ставится оценка.

Оценка за практический пункт имеет минимальное значение 0, а максимальная оценка за пункт зависит от сложности пункта. Максимальная оценка за участие в обсуждении вопросов в форуме составляет 1.

При завершении выполнения курсовой работы все набранные оценки суммируются. Максимально возможно набрать 120 баллов.

Затем набранные баллы переводятся в оценку за курсовую работу в зависимости от набранных баллов по следующей шкале:

- 0...59 баллов – неудовлетворительно;
- 60...79 – удовлетворительно;
- 80...99 – хорошо;
- 100...120 – отлично.

Затем необходимо защитить выполненную курсовую работу. Защита работы выполняется всеми студентами. Каждый студент должен ответить на несколько вопросов по части курсовой работы, которую он непосредственно выполнял, а также иметь представление о частях других студентов команды проекта.

В результате защиты курсовой работы оценка, полученная по сумме набранных баллов может быть снижена, в случае неудовлетворительных ответов на вопросы. Такое снижение происходит для каждого студента независимо.

#### **1.5. Предварительное обсуждение**

##### **1.5.1. Модель биологического нейрона и искусственного нейрона**

Опишите особенности модели биологического нейрона. Насколько точно модель искусственного нейрона соответствует биологическому нейрону?

##### **1.5.2. Устройство компьютера и строение мозга**

Сравните устройство современного компьютера и строение мозга. Возможно ли создание блоков компьютера на основе искусственных нейронов?

### 1.5.3. Преимущества и проблемы нейροкомпьютерных вычислений

Опишите преимущества нейροкомпьютерных вычислений. Какие проблемы возникают при использовании нейросетей для решения задач?

### 1.6. Тезисы вебинара

- 1) цели и задачи выполнения курсовой работы;
- 2) основные идеи и история развития нейронные сети;
- 3) метод проектов;
- 4) дистанционное обучение;
- 5) обучение в малых группах;
- 6) автоматизированная обучающая система «Moodle»;
- 7) порядок выполнения и защиты проекта;
- 8) выполнение самостоятельной работы «Персептроны».

### 1.7. Список литературы

- 1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 44 с.
- 2) Круглов, В.В., Борисов, В. В. Искусственные нейронные сети. – М.: Горячая линия – Телеком, 2002. – 382 с.: ил.
- 3) Хайкин, С. Нейронные сети: полный курс. – 2-е изд. – М. ; СПб. ; Киев : Вильямс, 2006. – 1104 с.
- 4) Уоссермен, Ф. Нейрокомпьютерная техника : Теория и практика. – М. : Мир, 1992. – 184 с.
- 5) Круглов, В. В. Нечеткая логика и искусственные нейросети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : Физматлит, 2001. – 224 с.
- 6) Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2006. – 452 с.
- 7) Осовский, С. Нейронные сети для обработки информации. // М.: Финансы и статистика, 2004. – 344 с.
- 8) Злобин, В. К., Ручкин, В. Н. Нейросети и нейрокомпьютеры. – СПб.: БХВ-Петербург, 2011. – 256 с.
- 9) Заенцев, И. В. Нейронные сети. Основные модели. – Воронеж:

ВГУ, 1999 г., 76 с.

10) Барский, А. Б. Нейронные сети: распознавание, управление, принятие решений. – М: Финансы и статистика, 2004. – 175 с. :ил.

11) Комарцова, Л. Г., Максимов А. В., Нейрокомпьютеры. – М.: Издательство МГТУ им. Н. Э. Баумана, 2004. – 400 с.: ил.

12) Моисеева М.В., Полат Е.С., Бухаркина М.Ю., Нежурина М.И. Интернет-обучение: технологии педагогического дизайна / Под ред. М.В. Моисеевой. – М.: Издательский дом «Камерон», 2004. – 216 с.

13) Ступницкая М.А. Что такое учебный проект? / М. А. Ступницкая. – М. : Первое сентября, 2010. – 44 с.

14) Е. С. Полат, М. Ю. Бухаркина, М. В. Моисеева, А. Е. Петров Новые педагогические и информационные технологии в системе образования / М.: Академия, 2002. – 272 с.

15) Анисимов А.М. Работа в системе дистанционного обучения Moodle / Учебное пособие. 2-е издание. – Харьков: ХНАГХ, 2009. – 292 с.

## **1.8. Практические задания**

### **1.8.1. Самостоятельная работа «Персептроны»**

Для допуска к дистанционному курсу по выполнению курсовой работы на основе метода проекта согласно рабочей программы необходимо выполнить самостоятельную работу №1 «Персептроны», приведенную в учебно-методическом пособии [1].

Формат результата: *doc, rtf, odt*.

## **2. Выбор задачи**

### **2.1. Цель и задачи модуля**

Цель модуля: выбор задачи и формализация обучающей выборки.

Задачи модуля:

- 1) обзор существенных практических задач требующих решения;
- 2) выбор практической задачи, требующей решения;
- 3) формализация обучающей выборки; определение конкретной формы представления обучающих образцов.

### **2.2. Теоретические сведения**

#### **2.2.1. Задачи решаемые искусственными нейросетями**

Нейронные сети превосходят последовательные машины в решении тех же задач, в которых машину превосходит человек. Задачи, требующие большого объема вычислений или высокой точности лучше выполняются обычной ЭВМ.

Задачи, успешно решаемым ИНС на данном этапе их развития:

- 1) системы управления и регулирования с предсказанием в реальном времени: роботами; самолётами и ракетами; технологическими процессами непрерывного производства в энергетике, металлургии и др.; гибридным двигателем автомобиля; пневмоцилиндром; сварочным аппаратом; электропечью; турбогенератором;
- 2) распознавание образов (классификация): изображений; человеческих лиц; букв и иероглифов; отпечатков пальцев в криминалистике; речи; сигналов радара и сонара; элементарных частиц и происходящих с ними физических процессов; заболеваний по симптомам в медицине; местностей в геологии, по косвенным признакам; признаков опасности в системах безопасности; психодиагностика;
- 3) формирование моделей и различных нелинейных и трудно описываемых математически систем, прогнозирование развития этих систем во времени: погоды; курса акций и других финансовых показателей; исхода лечения; политических событий; поведения реального или потенциального противника в военном конфликте и в экономической конкуренции; устойчивости супружеских отношений;
- 4) оптимизация – поиск наилучших вариантов (при конструировании технических устройств; при выборе экономической стратегии; при подборе команды; при лечении больного);

- 5) обработка сигналов при наличии больших шумов;
- 6) информационная безопасность: обнаружение и предотвращение мошенничества в телекоммуникационных сетях;
- 7) разнообразные конечные автоматы (системы массового обслуживания и коммутации; телекоммуникационные системы);
- 8) принятие решений и диагностика, исключаящие логический вывод; особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере;
- 9) ассоциативный поиск информации и создание ассоциативных моделей; ассоциативная память; синтез речи; формирование естественного языка.
- 10) оптимизация в математике, технике, науке, медицине, экономике;
- 11) аппроксимация функций для решения задач моделирования;
- 12) кластеризация (категоризация) выборки образцов на кластеры (категории) подобных между собой образцов для применения при сжатии данных, исследования статистических свойств данных, извлечения знаний (*data mining*).

Уникальное свойство нейросетей – универсальность. Хотя почти для всех перечисленных задач существуют эффективные математические методы решения и несмотря на то, что ИНС проигрывают специализированным методам для конкретных задач, благодаря универсальности и перспективности для решения глобальных задач, например, построения искусственного интеллекта и моделирования процесса мышления, они являются важным направлением исследования, требующим тщательного изучения.

### **2.2.2. Формализация обучающей выборки**

**Обучающая выборка.** Для применения ИНС одной из задач является формирование обучающей выборки. Для этого необходимо в формализованном виде представить данные для обучения ИНС. Обучающая выборка, представляется; обучающим образцом:  $X^k$  – вектор входов (входной образ),  $T^k$  – вектор верных выходов (требуемый, целевой образ);  $k$  – номер образца;  $K$  – общее число образцов. При обучении без учителя целевой выход  $T^k$  отсутствует (неизвестен).

Обучающая выборка может быть представлена:

- в виде экспериментальных данных;



- в виде эталонов;
- в виде аналитических зависимостей (формул).

Примеры реализации масштабирования данных приведен в [1].

**Масштабирование данных.** Данные, подаваемые на вход сети и снимаемые с выхода, должны быть правильно подготовлены. Один из распространенных способов – масштабирование.

Масштабирование используется, чтобы привести данные в допустимый диапазон. Если этого не сделать, то возможно несколько проблем:

- 1) нейроны входного слоя или окажутся в постоянном насыщении ( $s^k$  – взвешенная сумма велика по модулю) или будут все время заторможены ( $s^k$  – мала по модулю);
- 2) весовые коэффициенты примут очень большие или очень малые значения при обучении, как следствие, растянется процесс обучения и снизится точность.
- 3) невозможность обучения на реальных данных ИНС, на выходе которых используются нейроны с ограниченным диапазоном выходов; например для сигмоида:  $y^k \in (0,1)$ .

Пример реализации масштабирования данных приведен в [1].

### 2.2.3. Примерная тематика проектов

- 1) изучение работы одной из известной топологии нейросетей; разработка ПО;
- 2) изучение одного из методов обучения (оптимизации) двухслойной сигмоидальной нейросети; разработка ПО;
- 3) изучение существующих пакетов ПО моделирования нейросетей; использование пакетов ПО;
- 4) применение нейросетей различной топологии для решения практических задач; разработка ПО или использование пакетов ПО;
- 5) исследование альтернативных и новых топологий нейросетей, функций активации нейронов, методов обучения; разработка ПО или использование пакетов ПО;
- 6) ускорение работы нейросетей на основе их моделирования на графических акселераторах (технологии *CUDA*, *OpenCL*); разработка ПО или использование пакетов ПО;
- 7) исследование применения генетических алгоритмов для решения практических задач; разработка ПО или использование пакетов

ПО;

8) исследование применения нечеткой логики для решения практических задач); разработка ПО или использование пакетов ПО.

### **2.3. Предварительное обсуждение**

#### **2.3.1. Задачи, решаемые нейросетями**

Какие задачи на ваш взгляд решаются с помощью нейросетей, какие не решаются?

Максимальная оценка 1.

### **2.4. Тезисы вебинара**

- 1) цели и задачи выполнения модуля;
- 2) задачи решаемые искусственными нейросетями;
- 3) формирование обучающей выборки;
- 4) примерная тематика проектов;
- 5) выбор задачи для проекта.

### **2.5. Список литературы**

1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 44 с.

2) Уоссермен, Ф. Нейрокомпьютерная техника : Теория и практика. – М. : Мир, 1992. – 184 с.

3) Круглов, В. В. Нечеткая логика и искусственные нейросети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : Физматлит, 2001. – 224 с.

4) Заенцев, И. В. Нейронные сети. Основные модели. – Воронеж: ВГУ, 1999 г., 76 с.

5) Осовский, С. Нейронные сети для обработки информации. // М.: Финансы и статистика, 2004. – 344 с.

### **2.6. Практические задания**

#### **2.6.1. Описание возможных задач**

Перечислить и кратко описать практические задачи, решаемые с помощью нейросетей.

Формат результата: *doc, rtf, odt*. Максимальная оценка 3. Критерии оценки: количество задач (0-2), полнота описания (0-1).

#### 2.6.2. Выбор практической задачи

Выбрать практическую задачу, которая будет являться темой вашего проекта, и описать её.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 3. Критерии оценки: оригинальность задачи (0-1), полнота описания (0-2).

#### 2.6.3. Формат входных и выходных данных

Описать значение и формат входных и выходных данных вашей задачи.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 3. Критерии оценки: корректность решения (0-2), полнота описания (0-1).

### 2.7. Заключительное обсуждение

#### 2.7.1. Перекрестная оценка выбранных задач

Покажите другим командам описание своей задачи. Оцените выбранную задачу других команд. Оцените описание их задач.

Максимальная оценка 1.

#### 2.7.2. Обсуждение выполнения модуля

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

### **3. Выбор типа нейросети**

#### **3.1. Цель и задачи модуля**

Цель модуля: выбор типа нейросети и разработка топологии.

Задачи модуля:

- 1) обзор существующих типов нейронных сетей;
- 2) выбор типа нейросети для решения задачи;
- 3) разработка топологии сети.

#### **3.2. Теоретические сведения**

##### **3.2.1. Классификация топологий нейросетей**

При выборе топологии сети нужно определить:

- какие нейроны мы хотим использовать (число входов и функции активации);
- каким образом соединить нейроны между собой;
- что взять в качестве входов и выходов сети.

В многослойных сетях нейроны объединяются в слои – совокупность нейронов с единым типом, едиными принципами поступления входных и формирования выходных сигналов.

Среди нейронов сети выделяют:

- 1) входные нейроны или слои, на входы которых поступают входные данные ИНС;
- 2) выходные нейроны или слои, с выходов которых снимаются выходы ИНС;
- 3) промежуточные (скрытые) нейроны или слои, которые не подключены ни к входам ни к выходам ИНС.

Классификация топологий по характеру связей между нейронами:

- 1) полносвязные: каждый нейрон связан с каждым другим; связи нейронов составляют полный граф; сети имеют высокие вычислительные возможности, но сложное обучение;
- 2) слабосвязные (с локальными связями): нейроны имеют связи только с ограниченным числом нейронов, которые в топологии ИНС находятся в непосредственной близости (соседи, в окрестности нейрона); сети имеют сниженные вычислительные возможности, упрощенное обучение, большое число слоев;
- 3) многослойные (слоистые): каждый выход нейронов одного слоя имеет связи с входами нейрона следующего слоя; наиболее

изученный и популярный тип сетей;

Классификация топологий по числу слоев:

- 1) однослойные;
- 2) двухслойные;
- 3) и так далее.

Монотонные сети – специальный тип слоистых сетей, каждый слой которых разбит на два множества: тормозящие и возбуждающие, имеющие соответственные тормозящие и возбуждающие связи между нейронами.

Классификация топологий по наличию обратной связи слоев:

- 1) без обратных связей (прямого распространения);
- 2) с обратными связями; сети имеют высокие вычислительные возможности; обладают свойствами памяти – сеть помнит предыдущее состояние нейронов; работа сети состоит из нескольких циклов, до останова изменений в сети (стабилизации); но обучение сложное, и возможна потеря динамической устойчивости, из-за наличия обратной связи; на таких сетях реализуют конечные автоматы.

Классификация нейронов по типу выходов:

- 1) бинарные (цифровые) – нейроны имеют выходы принимающие только 2 значения:  $y^k = \{0,1\}$  или  $y^k = \{-1,+1\}$ ; пороговая функция активации;
- 2) аналоговые – выходы нейронов принимают диапазон значений: например, сигмоид  $y^k \in (0,1)$ ; имеют большие вычислительные возможности.

Классификация нейронов по типу функции активации [1].

Классификация топологий по способу переключения:

- 1) синхронные – переключается сразу вся сеть или целый слой; более популярны и изучены.
- 2) асинхронные – нейроны переключаются независимо друг от друга; соответствует биологическому способу переключения; менее популярны и изучены.

Классификация топологий по числу типов используемых нейронов:

- 1) однородные (гомогенные) – все нейроны ИНС одинакового типа (одинаковая функция активации);
  - 2) разнородные (гетерогенные) – более одного типа нейронов в ИНС.
- В некоторых случаях синапсы нейронов имеют фиксированные

значения, что позволяет упростить процедуру обучения и сократить затраты памяти на хранения результата, например, в сети Хемминга и когнитроне.

Выбор структуры ИНС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных задач уже существуют оптимальные, на сегодня, типы нейросетей, некоторые из которых приведены в 3.2.2.

В целом можно отметить, что вычислительные возможности сети возрастают:

- с увеличением числа слоев сети;
- с увеличением числа нейронов в слоях;
- с введением обратных связей;
- с введением нелинейных функций активаций нейронов;
- с увеличением числа связей нейронов и слоев.

Однако, при росте вычислительных возможностей ИНС, как правило, растет так же и сложность их обучения.

### **3.2.2. Основные типы нейросетей**

Под типом (архитектурой) ИНС понимают связанные с названием сети: топологию, метод обучения, особенности работы сети, решаемые сетью задачи.

Таким образом, при выборе типа сети, необходимо учитывать решаемую задачу, а так же иметь ввиду ограничения, накладываемые на методы обучения сети.

По типам сетей подробнее смотри в [1...5].

#### **3.2.2.1. Персептрон Розенблатта**

Персептрон (*to percept*; представлять) – математическая и компьютерная модель восприятия информации мозгом, предложенная Розенблаттом в 1957 году и реализованная в виде электронной машины «Марк-1» в 1960 году. Персептрон стал одной из первых моделей нейросетей, а «Марк-1» — первым в мире нейрокомпьютером. Несмотря на свою простоту, персептрон способен обучаться и решать довольно сложные задачи. Персептрон представляет собой однослойную сеть (нейрон) с пороговой функцией активации и бинарными входами. Основная математическая задача, с которой он справляется, – это линейное разделение любых нелинейных множеств.

Подробнее смотри в [1].

### **3.2.2.2. Многослойные персептроны**

В современной теории нейросетей под персептроном понимают многослойную слоистую нейросеть прямого распространения сигналов с функцией активации пороговой или сигмоидальной (нелинейный порог).

### **3.2.2.3. Двухслойный сигмоид**

Двухслойный сигмоид – частный случай многослойного персептрона – является – универсальным аппроксиматором нелинейного отображения  $X^k \rightarrow Y^k$ . Это наиболее изученная и самая популярная из современных ИНС.

Подробнее смотри в [1].

### **3.2.2.3. Сеть встречного распространения**

Хехт-Нильсен разработал сеть как средство для объединения неконтролируемого слоя Кохонена с контролируемым выходным слоем. Сеть предназначена для решения сложных классификаций, при минимизации числа нейронов и времени обучения. Обучение для сети похоже на сети с линейным разделением обучающих векторов (*LVQ – linear vector quantization*).

Состав сети: слой Кохонена (самоорганизующаяся карта); слой Гроссберга.

Первая сеть встречного распространения состояла из двунаправленного отображения между входным и выходным слоями. Данные поступают на входной слой для генерации классификации на выходном слое, выходной слой поочередно принимает дополнительный входной вектор и генерирует выходную классификацию на входном слое сети. Из-за такого встречно-распространенного потока информации сеть получила свое название. Много разработчиков используют однонаправленный вариант сети, когда существует лишь один путь прямого распространения от входного к выходному слою.

В сети встречного распространения объединены два алгоритма: самоорганизующаяся карта Кохонена и выходная звезда Гроссберга.

В процессе обучения сети встречного распространения входные векторы ассоциируются с соответствующими выходными векторами

(двоичными или аналоговыми). После обучения сеть формирует выходные сигналы, которые отвечают входным сигналам. Обобщающая способность сети дает возможность получать правильный выход, когда входной вектор неполный или искаженный.

#### **3.2.2.4. Ассоциативные нейронные сети**

К сетям реализующим ассоциативную память относятся сеть Хопфилда, сеть Хэмминга, двунаправленная ассоциативная память (ДАП).

ИНС Хопфилда – полносвязная бинарная однородная однослойная нейронная сеть с симметричной матрицей обратных связей. В процессе работы динамика таких сетей сходится к одному из положений равновесия. Эти положения равновесия являются локальными минимумами функционала, называемого энергией сети (в простейшем случае – локальными минимумами отрицательно определённой квадратичной формы на  $n$ -мерном кубе). Такая сеть может быть использована как автоассоциативная память, как фильтр, а также для решения некоторых задач оптимизации. В отличие от многих нейронных сетей, работающих до получения ответа через определённое количество тактов, сети Хопфилда работают до достижения равновесия, когда следующее состояние сети в точности равно предыдущему: начальное состояние является входным образом, а при равновесии получают выходной образ.

В сети Хопфилда матрица связей является симметричной по главной диагонали, а диагональные элементы матрицы полагаются равными нулю, что исключает эффект воздействия нейрона на самого себя и является необходимым для сети Хопфилда условием, динамической устойчивости в процессе работы сети.

#### **3.2.2.5. Радиальные нейронные сети**

Сети с радиальными базисными нейронами (*RBF – radial base function*) имеют ряд преимуществ перед рассмотренными многослойными сигмоидальными сетями прямого распространения. Во-первых, они моделируют произвольную нелинейную функцию с помощью всего одного промежуточного слоя, тем самым, избавляя разработчика от необходимости решать вопрос о числе слоев. Во-вторых, параметры линейной комбинации в выходном слое можно



полностью оптимизировать с помощью хорошо известных методов линейной оптимизации, которые работают быстро и не испытывают трудностей с локальными минимумами, так мешающими при обучении с использованием алгоритма обратного распространения ошибки. Поэтому сеть *RBF* обучается очень быстро – на порядок быстрее, чем с использованием алгоритма обратного распространения.

Недостатки сетей РБФ: данные сети обладают плохими экстраполирующими свойствами и получаются весьма громоздкими при большой размерности вектора входов.

Нейронная сеть радиальных базисных функций содержит в наиболее простой форме два слоя:

- 1) слой входных нейронов с радиально симметричной активационной функцией (функцией Гаусса);
- 2) выходной линейный нейрон, выполняет линейную комбинацию над выходами радиального слоя.

#### **3.2.2.6. Сети корреляционного анализа**

Сеть *PCA* реализует метод главных компонент (*principal component analysis* – *PCA*) на основе адаптивного обучения (*inline*-обучение). *PCA* – один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации. Изобретен Пирсоном в 1901 г. Применяется во многих областях, таких как распознавание образов, компьютерное зрение, сжатие данных и других. Вычисление главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных. Иногда метод главных компонент называют преобразованием Кархунена-Лоэва или преобразованием Хотеллинга. Структура сети – слой линейных нейронов.

Другой способ уменьшения размерности данных – это метод независимых компонент (*independent component analysis* – *ICA*). Структура сети *ICA* совпадает с сетью *PCA*. Отличие сетей в методах обучения.

#### **3.2.2.7. Когнитрон. Неокогнитрон**

Когнитрон (*cognitive* – познавательный) – ИНС на основе принципа самоорганизации. Своей архитектурой когнитрон похож на строение зрительной коры, имеет иерархическую многослойную организацию, в которой нейроны между слоями связаны только ло-

кально. Обучается конкурентным обучением (без учителя). Каждый слой мозга реализует различные уровни обобщения; входной слой чувствителен к простым образам, таким, как линии, и их ориентации в определенных областях визуальной области, в то время как реакция других слоев является более сложной, абстрактной и независимой от позиции образа. Аналогичные функции реализованы в когнитроне путем моделирования организации зрительной коры.

Когнитрон конструируется в виде слоев нейронов, соединенных синапсами. Предсинаптический нейрон в одном слое связан с постсинаптическим нейроном в следующем слое. Имеются два типа нейронов: возбуждающие узлы, которые стремятся вызвать возбуждение постсинаптического узла, и тормозящие узлы, которые тормозят это возбуждение. Возбуждение нейрона определяется взвешенной суммой его возбуждающих и тормозящих входов, однако в действительности механизм является более сложным, чем простое суммирование.

Данная нейронная сеть одновременно является как моделью процессов восприятия на микроуровне, так и вычислительной системой, применяющейся для технических задач распознавания образов.

Неокогнитрон является дальнейшим развитием идеи когнитрона и более точно отражает строение зрительной системы, позволяет распознавать образы независимо от их преобразований, вращений, искажений и изменений масштаба. Неокогнитрон может как самообучаться, так и обучаться с учителем. Неокогнитрон получает на входе двумерные образы, аналогичные изображениям на сетчатке глаза, и обрабатывает их в последующих слоях аналогично тому, как это было обнаружено в зрительной коре человека. Конечно, в неокогнитроне нет ничего, ограничивающего его использование только для обработки визуальных данных, он достаточно универсален и может найти широкое применение как обобщенная система распознавания образов.

### **3.2.2.8. Рекуррентные сети**

Рекуррентные сети – сети с обратной связью. К ним относят сети: персептронная сеть с обратной связью (сеть *RMLP* – *recurrent multi-layer perceptron*); рекуррентная сеть Элмана; сеть *RTRN* (*real time recurrent network*). Такие сети используются для обработки сигналов в реальном времени и для реализации конечных автоматов.

Нейронная сеть Элмана – один из видов рекуррентной сети, которая так же как и сеть Джордана получается из многослойного персептрона введением обратных связей, только связи идут не от выхода сети, а от выходов внутренних нейронов. Это позволяет учесть предисторию наблюдаемых процессов и накопить информацию для выработки правильной стратегии управления. Эти сети могут применяться в системах управления движущимися объектами, так как их главной особенностью является запоминание последовательностей.

Сети *RMLP* и *RTRN* отличаются от сети Элмана числом слоев и слоями, с которых берется сигнал для обратной связи. Общие принципы организации и обучения сетей совпадают.

### 3.2.2.9. Нечёткая логика

Нечёткая логика (*fuzzy logic*) – раздел математики, являющийся обобщением классической логики и теории множеств, базирующее на понятии нечёткого множества, впервые введённого Л. Заде в 1965 году как объекта с функцией принадлежности элемента к множеству, принимающей любые значения в интервале  $[0, 1]$ , а не только 0 или 1. На основе этого понятия вводятся различные логические операции над нечёткими множествами и формулируется понятие лингвистической переменной, в качестве значений которой выступают нечёткие множества.

Предметом нечёткой логики считается исследование рассуждений в условиях нечёткости, размытости, сходных с рассуждениями в обычном смысле, и их применение в вычислительных системах.

Символическая нечёткая логика основывается на понятии  $t$ -нормы. После выбора некоторой  $t$ -нормы (а её можно ввести несколькими разными способами) появляется возможность определить основные операции над пропозициональными переменными: конъюнкцию, дизъюнкцию, импликацию, отрицание и другие.

Нетрудно доказать теорему о том, что дистрибутивность, присутствующая в классической логике, выполняется только в случае, когда в качестве  $t$ -нормы выбирается  $t$ -норма Гёделя.

Кроме того, в силу определенных причин, в качестве импликации чаще всего выбирают операцию, называемую *residuum* (она также зависит от выбора  $t$ -нормы).

Определение основных операций, перечисленных выше, приво-

дит к формальному определению базисной нечёткой логики, которая имеет много общего с классической булевозначной логикой (точнее, с исчислением высказываний).

Основное понятие нечёткой логики в широком смысле – нечёткое множество, определяемое при помощи обобщенного понятия характеристической функции. Затем вводятся понятия объединения, пересечения и дополнения множеств (через характеристическую функцию; задать можно различными способами), понятие нечёткого отношения, а также одно из важнейших понятий – понятие лингвистической переменной.

Даже такой минимальный набор определений позволяет использовать нечёткую логику в некоторых приложениях, для большинства же необходимо задать ещё правило вывода и оператор импликации.

#### **3.2.2.10. Нечеткая логика и гибридные нейронные сети**

Нейросеть может иметь нейроны и слои, реализующие различные математические методы, например, нейроны для реализации нечеткой функции принадлежности, применяемые для реализации на базе ИНС положений теории нечеткого логического вывода. Такие ИНС принято называть гибридными. Для их обучения могут использоваться специальные методы обучения, связанные с реализуемыми математическими методами.

Поскольку нечеткие множества описываются функциями принадлежности, а  $t$ -нормы и  $k$ -нормы обычными математическими операциями, можно представить нечеткие логические рассуждения в виде нейронной сети. Для этого функции принадлежности надо интерпретировать как функции активации нейронов, передачу сигналов как связи, а логические  $t$ -нормы и  $k$ -нормы, как специальные виды нейронов, выполняющие математические соответствующие операции. Существует большое разнообразие подобных нейро-нечетких сетей (*neuro-fuzzy network*). Например, *ANFIS (adaptive neuro fuzzy inference system)* – адаптивная нейро-нечеткая система вывода.

### **3.3. Предварительное обсуждение**

#### **3.3.1. Число нейронов и число слоёв**

Что лучше большое число слоев или большое число нейронов?

Максимальная оценка 1.

### 3.3.2. Выбор типа нейросети

Какими критериями вы будете руководствоваться при выборе типа нейросети для вашей задачи?

Максимальная оценка 1.

### 3.4. Тезисы вебинара

- 1) цели и задачи модуля;
- 2) классификация топологий нейросетей;
- 3) обзор основных типов нейросетей;
- 4) выбор типа нейросети для проекта.

### 3.5. Список литературы

- 1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 44 с.
- 2) Уоссермен, Ф. Нейрокомпьютерная техника : Теория и практика. – М. : Мир, 1992. – 184 с.
- 3) Осовский, С. Нейронные сети для обработки информации. // М.: Финансы и статистика, 2004. – 344 с.
- 4) Круглов, В. В. Нечеткая логика и искусственные нейросети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : Физматлит, 2001. – 224 с.
- 5) Хайкин, С. Нейронные сети: полный курс. – 2-е изд. – М. ; СПб. ; Киев : Вильямс, 2006. – 1104 с.

### 3.6. Практические задания

#### 3.6.1. Обзор существующих типов нейросетей

Перечислите существующие типы нейросетей. Опишите их.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 3. Критерии оценки: количество типов (0-2), полнота описания (0-1).

#### 3.6.2. Выбор типа нейросети

Выбрать и описать тип нейросети для решаемой задачи.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 3. Критерии оценки: корректность решения (0-2), полнота описания (0-1).

#### 3.6.3. Разработка топологии сети

Разработать топологию сети выбранного типа для вашей задачи.

Формат результата: *odg*, *emf*. Максимальная оценка 5. Критерии

оценки: корректность решения (0-2), полнота описания (0-1), соответствие стандартам (0-2).

### **3.7. Заключительное обсуждение**

#### **3.7.1. Возможность улучшения топологии нейросети**

Можно ли улучшить топологию вашей нейросети?

Максимальная оценка 1.

#### **3.7.2. Перекрестная оценка выбранного типа нейросети**

Покажите другим командам описание выбранного типа нейросети для решаемой задачи. Оцените выбранный тип нейросети других команд. Оцените их описание.

Максимальная оценка 1.

#### **3.7.3. Обсуждение выполнения модуля**

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

## 4. Выбор метода обучения нейросети

### 4.1. Цель и задачи модуля

Цель модуля: выбор метода обучения нейросети.

Задачи модуля:

- 1) обзор методов обучения;
- 2) выбор метода обучения;
- 3) описание метода обучения.

### 4.2. Теоретические сведения

#### 4.2.1. Обучение нейросети

Обучение ИНС заключается в определении таких значений синапсов (весов), которые обеспечат решение требуемой задачи посредством ИНС заданной архитектуры.

Искусственные нейронные сети обучаются самыми разнообразными методами. К счастью, большинство методов обучения исходят из общих предпосылок и имеет много идентичных характеристик. Целью данного приложения является обзор некоторых фундаментальных алгоритмов, как с точки зрения их текущей применимости, так и с точки зрения их исторической важности. После ознакомления с этими фундаментальными алгоритмами другие, основанные на них, алгоритмы будут достаточно легки для понимания и новые разработки также могут быть лучше поняты и развиты.

Основные подходы к обучению:

- 1) обучение с учителем; на обучающей выборке  $\{(X^k, T^k), k = 1..K\}$ ;
- 2) обучение без учителя; известны только входы  $\{X^k, k = 1..K\}$ .

Обучающие алгоритмы могут быть классифицированы как алгоритмы обучения с учителем и без учителя. В первом случае существует учитель, который предъявляет входные образы сети, сравнивает результирующие выходы с требуемыми, а затем настраивает веса сети таким образом, чтобы уменьшить различия. Трудно представить такой обучающий механизм в биологических системах; следовательно, хотя данный подход привел к большим успехам при решении прикладных задач, он отвергается исследователями, полагающими, что искусственные нейронные сети обязательно должны использовать те же механизмы, что и человеческий мозг.

Во втором случае обучение проводится без учителя, при предъявлении входных образов сеть самоорганизуется посредством настройки своих весов согласно определенному алгоритму. Вследствие отсутствия указания требуемого выхода в процессе обучения результаты непредсказуемы с точки зрения определения возбуждающих образов для конкретных нейронов. При этом, однако, сеть организуется в форме, отражающей существенные характеристики обучающего набора. Например, входные образы могут быть классифицированы согласно степени их сходства так, что образы одного класса активизируют один и тот же выходной нейрон.

#### **4.2.2. Минимизация ошибки**

Наиболее изучено и более часто применяется практически обучение с учителем. При этом задача обучения сводится в задаче оптимизации (минимизации) многопараметрической нелинейной функции, где параметрами служат значения всех синапсов ИНС, а сама функция – ошибка сети, разница между верным ответом  $T^k$  и реальным ответом ИНС  $Y^k$ .

Методы оптимизации, применяемые для обучения с учителем:

- 1) градиентный алгоритм (обратное распространение ошибки) ;
- 2) стохастические алгоритмы оптимизации (тепловая машина);
- 3) алгоритмы на базе эволюционного подхода (генетический алгоритм).

Подробнее смотри в [1].

#### **4.2.3. Другие методы обучения**

##### **4.2.3.1. Обучение по дельта-правилу**

Обучение однослойных персептронов Розенблатта выполняется по дельта-правилу, которое является частным случаем градиентного метода. Важность метода, в том что метод являлся первым систематическим формальным методом обучения ИНС с учителем (1957). Ограничение метода – применимость только для однослойных сетей. Подробнее смотри в [1].

##### **4.2.3.2. Обучение по методу Уидроу-Хоффа**

Персептрон Розенблатта ограничивается бинарными выходами. Уидроу и Хоффом расширили алгоритм обучения персептрона на



случай непрерывных выходов. Кроме того, они разработали математическое доказательство того, что сеть при определенных условиях будет сходиться к любой функции, которую она может представить. Их первая модель – адалин (*adaline – adaptive linear neuron*) – имеет один выходной нейрон, более поздняя модель – мадалин (*multiple adaline*) – расширяет ее на случай с многими выходными нейронами.

Выражения, описывающие процесс обучения адалина, очень схожи с дельта-правилом.

$$w_i(k+1) = w_i(k) - \alpha \cdot (t^k - s^k) \cdot x^k,$$

где  $k$  – в различных задачах можно интерпретировать как номер обучающего образца или как момент времени при анализе сигналов.

Метод является частным случаем градиентного метода для ИНС, состоящего из однослойного нейрона с линейной функцией активации.

Параметр скорости обучения  $\alpha$  определяется на основе следа автокорреляционной матрицы вектора  $X$ .

Использование алгоритма Уидроу-Хоффа позволяет решать задачу аппроксимации функции, которое реализует метод наименьших квадратов, без решения системы уравнений. Такой подход используется при обработке сигналов в *online* режиме обучения и позволяет вычислять коэффициенты КИХ и БИХ фильтров в адаптивном режиме (КИХ – конечная – и БИХ – бесконечная импульсные характеристики).

Нейрон Уидроу-Хоффа широко используется в современных системах обработки сигналов, например при обработке звука.

#### **4.2.3.3. Обучение по методу Хэбба**

Работа Хэбба обеспечила основу для большинства алгоритмов обучения, которые были разработаны после ее выхода. Основываясь на физиологических и психологических исследованиях, Хэбб интуитивно выдвинул гипотезу о том, каким образом может обучаться набор биологических нейронов. Его теория предполагает только локальное взаимодействие между нейронами при отсутствии глобального учителя; следовательно, обучение является неуправляемым,

Хэбб предположил, что синаптическое соединение двух нейронов усиливается, если оба эти нейрона возбуждены. Это можно представить как усиление синапса в соответствии с корреляцией уровней возбужденных нейронов, соединяемых данным синапсом. По этой

причине алгоритм обучения Хэбба иногда называется корреляционным алгоритмом.

Идея алгоритма выражается следующим равенством:

$$w_{ij}(t+1) = w_{ij}(t) + s_i \cdot s_j,$$

где  $w_{ij}(t)$  – значение синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $s_i$  – уровень возбуждения предсинаптического нейрона;  $s_j$  – уровень возбуждения постсинаптического нейрона.

Последующее использование метода Хэбба для обучения нейронных сетей привело к большим успехам, но наряду с этим показало ограниченность метода; некоторые образы просто не могут использоваться для обучения этим методом. В результате появилось большое количество расширений и нововведений, большинство из которых в значительной степени основано на работе Хэбба.

Развитие идей, предложенных Хэббом привело к созданию корреляционных сетей:

- сеть анализа главных компонент (*principal component analysis* – PCA);
- сеть анализа независимых компонент (*independent component analysis* – ICA).

#### 4.2.3.4. Обучение входной звезды

Входная звезда (*instar*) Гроссберга выполняет распознавание образов, то есть она обучается реагировать на определенный входной вектор  $X^k$  и ни на какой другой. Это обучение реализуется путем настройки весов таким образом, чтобы они соответствовали входному вектору. Выход входной звезды определяется как взвешенная сумма ее входов, как это описано в предыдущих разделах. С другой точки зрения, выход можно рассматривать как свертку входного вектора с весовым вектором, меру сходства нормализованных векторов. Следовательно, нейрон должен реагировать наиболее сильно на входной образ, которому был обучен.

Процесс обучения выражается следующим образом:

$$w_i(t+1) = w_i(t) + \alpha(x_i - w_i(t)),$$

где  $w_i$  – вес входа  $x_i$ ; – вход;  $\alpha$  – скорость обучения, которая постепенно уменьшается в процессе обучения от 1 до 0.

После завершения обучения предъявление входного вектора  $X^k$  будет активизировать обученный входной нейрон. Это можно рассматривать как единый обучающий цикл, если  $\alpha$  установлена в 1, однако в этом случае исключается способность входной звезды к обобщению. Хорошо обученная входная звезда будет реагировать не только на определенный единичный вектор, но также и на незначительные изменения этого вектора. Это достигается постепенной настройкой нейронных весов при предъявлении в процессе обучения векторов, представляющих нормальные вариации входного вектора. Веса настраиваются таким образом, чтобы усреднить величины обучающих векторов, и нейроны получают способность реагировать на любой вектор этого класса.

Входная звезда с некоторыми модификациями используется в слое Кохонена сети встречного распространения. Входная звезда реализует обучение без учителя (самоорганизацию), используется для реализации задачи кластеризации. Такая задача кластеризации называется – линейное разделение векторов (*LVQ*).

#### 4.2.3.5. Обучение выходной звезды

Выходная звезда (*outstar*) Гроссберга формирует выходной сигнал всякий раз, когда она активирована.

Для того чтобы обучить нейрон выходной звезды, его веса настраиваются в соответствии с требуемым целевым вектором. Алгоритм обучения может быть представлен символически следующим образом:

$$w_i(t+1) = w_i(t) + \beta(t_i - w_i(t)),$$

где  $w_i$  – вес входа  $x_i$ ; – вход;  $\beta$  – скорость обучения, которая постепенно уменьшается в процессе обучения от 1 до 0.

Как и в случае входной звезды, веса выходной звезды, постепенно настраиваются над множеством векторов, представляющих собой обычные вариации идеального вектора. В этом случае выходной сигнал нейронов представляет собой статистическую характеристику обучающего набора и может в действительности сходиться в процессе обучения к идеальному вектору при предъявлении только искаженных версий вектора.

В отличие от входной звезды обучение выходной является обучением с учителем, несмотря на внешнюю схожесть формул.

Выходная звезда используется как слой Гроссберга в сети встречного распространения.

#### 4.2.3.5. Обучение нейронов ассоциативной памяти

При запоминании эталона с помощью ассоциативной памяти определение значений синапсов происходит за счет прямого вычисления. Это можно интерпретировать как нанесение отпечатка вектора, представляющего эталон на матрицу синапсов сети.

Для сети Хопфилда вычисление выполняется по формуле:

$$w_{ij} = \sum_{k=1}^K x_i^k \cdot x_j^k,$$

где  $k$  – номер эталона;  $K$  – число эталонов (объем памяти).

Произведение  $x_i$  и  $x_j$  можно интерпретировать как связь  $i$ -го и  $j$ -го входа эталона.

Объем памяти ограничен и для ортогональных эталонов составляет  $K=n^2$ , где  $n$  – размер вектора эталона (число входов). Однако, для реальных данных составляет  $K=0.15 \cdot n^2$ .

У сетей, реализующих ассоциативную память, упрощенное обучение, но работа сети выполняется в несколько циклов, пока выходы сети не перестанут изменяться (стабилизируются).

#### 4.2.3.6. Векторные операции при обучении

При использовании нейронов с линейной активации возможно непосредственное вычисление значений синапсов на основе матричных операций. Такой подход применяется, например, в сети с радиальными базисными нейронами (*RBF – radial base function*).

Для определения значений синапсов используются формулы:

$$t = w \cdot \Phi$$
$$w = \Phi^{-1} \cdot t$$

где  $t$  – вектор требуемых выходов сети;  $w$  – вектор синапсов выходного нейрона;  $\Phi$  – матрица значений радиальных функций входного слоя для каждого входного образца. Сложность представляет вычисление обратной матрицы от квадратной матрицы  $\Phi$ .

Однако при практической реализации используют матрицу  $G$  – прямоугольную матрицу, так как число радиальных функций меньше числа входных образцов:

$$t = w \cdot G$$

$$w = G^+ \cdot t = (G^T \cdot G)^{-1} \cdot G^T \cdot$$

где  $G^+$  – матрица псевдоинверсии матрицы  $G$  на практике рассчитывается с применением декомпозиции  $SVD$ , что упрощает вычисления.

#### 4.2.3.7. *Online* и *offline* режимы обучения

Как правило процедура обучения выполняется заранее, является длительной по времени и сложной по вычислениям: один из методов оптимизации суммарной ошибки ИНС. При этом вся обучающая выборка многократно используется для обучения. Цикл применения к каждому образцу обучающей выборки процедуры коррекции синапсов – эпоха обучения. После того, как ИНС обучена она может быть использована для работы. При этом коррекция синапсов больше не производится. Такой подход называется *offline*-обучение.

При обработке экспериментальных сигналов в реальном времени, сеть может встретить каждый из сигналов только один раз. В дальнейшем сам характер сигналов может кардинально измениться. Таким образом, нет смысла обучать сеть на старых по времени образцах. Такой подход называется *online*-обучение. Обучение проводится на каждом вновь поступившем образце в момент времени  $k$  в цикле:

- 1) этап работы ИНС для образца  $X^k$ ;
- 2) этап обучения (коррекция синапсов) ИНС для образца  $X^k$ ;

Важным параметром является скорость обучения сети: влияет на поведение сети в отношении пластичность-стабильность. При высоком значении скорости обучения сеть потеряет динамическую устойчивость, а при низком почти не будет обучаться на новых образцах.

*Online*- обучение используется, например, для нейрона Уидроу-Хоффа и для сетей  $PCA$  и  $ICA$ .

#### 4.2.3.8. Конкурентное обучение

Метод обучения используется совместно с конкурентной функцией активации слоя нейронов. При этом, процедуре коррекции синапсов подвергаются не все нейроны слоя, а только победитель (чемпион) – нейрон с максимальным выходом. В некоторых реализациях рассматривается группа победителей. Такой подход позволяет скорректировать веса отдельных нейронов, при этом не испортить уже

обученные значения других нейронов.

Метод используется в сети встречного распространения, для обучения слоя Кохонена.

### **4.3. Предварительное обсуждение**

#### **4.3.1. Обучение точными методами**

Возможно ли обучение нейросети для вашей задачи с помощью точных методов?

Максимальная оценка 1.

#### **4.3.2. Проблема локальных минимумов**

Как решить проблему локальных минимумов при обучении для вашей задачи?

Максимальная оценка 1.

### **4.4. Тезисы вебинара**

1) обучение с учителем и без (самоорганизация); точные и итерационные методы обучения;

2) обучение нейросети, как задача оптимизации; понятие о локальных минимумах;

3) основные методы обучения (оптимизации) с учителем: градиентный, стохастический, генетический;

4) соотношение процесса обучения и работы нейросети: сети с обратными связями;

5) выбор метода обучения для конкретной задачи и типа нейросети.

### **4.5. Список литературы**

1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 44 с.

2) Уоссермен, Ф. Нейрокомпьютерная техника : Теория и практика. – М. : Мир, 1992. – 184 с.

3) Осовский, С. Нейронные сети для обработки информации. // М.: Финансы и статистика, 2004. – 344 с.

4) Круглов, В. В. Нечеткая логика и искусственные нейросети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : Физматлит, 2001. – 224 с.

5) Хайкин, С. Нейронные сети: полный курс. – 2-е изд. – М. ;

СПб. ; Киев : Вильямс, 2006. – 1104 с.

6) Струченков, В. И. Методы оптимизации. – М. : Солон-Пресс, 2009. – 320 с.

7) Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2006. – 452 с.

#### **4.6. Практические задания**

##### **4.6.1. Обзор методов обучения**

Перечислить возможные методы обучения для выбранной нейросети. Описать каждый их перечисленных методов.

Формат результата: *doc, rtf, odt*. Максимальная оценка 3. Критерии оценки: количество методов (0-2), полнота описания (0-1).

##### **4.6.2. Выбор метода обучения**

Выбрать и описать выбранный метод обучения.

Формат результата : *doc, rtf, odt*. Максимальная оценка 3. Критерии оценки: корректность решения (0-2), полнота описания (0-1).

##### **4.6.3. Алгоритмы обучения и работы сети**

Разработать алгоритмы обучения и функционирования нейросети.

Формат результата: *odg, emf*. Максимальная оценка 5. Критерии оценки: корректность решения (0-2), полнота описания (0-1), соответствие стандартам (0-2).

#### **4.7. Заключительное обсуждение**

##### **4.7.1. Оценка выбранного метода обучения**

Оцените преимущества и недостатки выбранного метода обучения нейросети.

Максимальная оценка 1.

##### **4.7.2. Перекрестная оценка метода обучения**

Покажите другим командам описание своего метода обучения. Оцените выбранные методы обучения других команд. Оцените описание их описание.

Максимальная оценка 1.

##### **4.7.3. Обсуждение выполнения модуля**

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

## **5. Разработка программы**

### **5.1. Цель и задачи модуля**

Цель модуля: разработка и отладка программы.

Задачи модуля:

- 1) разработка структуры программы;
- 2) кодирование программы;
- 3) отладка программы.

### **5.2. Предварительное обсуждение**

#### **5.2.1. Ввод обучающих образцов и вывод результатов**

Как организовать ввод обучающих образцов и вывод результатов?

Максимальная оценка 1.

#### **5.2.2. Структура программы**

Какие подсистемы должна содержать программа, реализующая нейросеть?

Максимальная оценка 1.

### **5.3. Тезисы вебинара**

- 1) цели и задачи модуля;
- 2) выбор между разработкой ПО и использование пакета ПО для моделирования нейросети;
- 3) выбор структуры ПО; структурный или объектный подход; необходимость графического интерфейса пользователя;
- 4) кодирование программы;
- 5) отладка программы.

### **5.4. Список литературы**

1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 48 с.

2) Г.С. Иванова. Технология программирования: Учебник для вузов. – 2-е изд., стереотип. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 320 с.

### **5.6. Практические задания**



#### 5.6.1. Разработка структуры программы

Разработайте структуру вашей программы.

Формат результата: *odg, emf*. Максимальная оценка 5. Критерии оценки: корректность решения (0-2), полнота описания (0-1), соответствие стандартам (0-2).

#### 5.6.2. Кодирование программы

Реализуйте программно решение практической задачи с помощью нейросети. Приведите текст программы на языке высокого уровня.

Формат результата: *doc, rtf, odt, txt*. Максимальная оценка 15. Критерии оценки: корректность решения (0-5), полнота описания (0-5), соответствие стандартам (0-5).

#### 5.6.3. Отладка программы

Отладьте программу. Приведите результаты работы программы для 2-3 примеров.

Формат результата: *xls, ods*. Максимальная оценка 10. Критерии оценки: корректность решения (0-4), качество визуализации (0-2), число эпох обучения (0-2), сбалансированность примеров (0-2).

### 5.7. Заключительное обсуждение

#### 5.7.1. Оптимальность реализации алгоритмов

Оцените оптимальность реализации алгоритмов с точки зрения быстродействия и затрат памяти.

Максимальная оценка 1.

#### 5.7.2. Перекрестная оценка структуры программы

Покажите другим командам структуру своей программы. Оцените структуру программы других команд. Оцените описание их оформление.

Максимальная оценка 1.

#### 5.7.3. Обсуждение выполнения модуля

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

## 6. Проведение экспериментов

### 6.1. Цель и задачи модуля

Цель модуля: проведение экспериментов и оценка результатов.

Задачи модуля:

- 1) формирование обучающей выборки;
- 2) проведение экспериментов над обучающей выборкой;
- 3) оптимизация числа нейронов в скрытых слоях;
- 4) оценка результатов экспериментов;
- 5) выводы и предложения по улучшению работы нейросети.

### 6.2. Теоретические сведения

#### 6.2.1. Формирование обучающей выборки

$K$  – количество элементов в обучающем множестве – должно быть достаточным для обучения сети, чтобы под управлением алгоритма обучения сформировать набор параметров (весов) ИНС, дающий нужное отображение  $X^k \rightarrow Y^k$ , где  $Y^k$  – фактический вектор выходов ИНС. В ходе обучения веса ИНС корректируются так, чтобы  $Y^k \approx T^k$ . Количество пар в обучающем множестве не регламентируется. Если элементов слишком много или мало, сеть не обучится и не решит поставленную задачу.

В результате обучения ИНС (настройки весов) необходимо добиться чтобы:

- 1) обеспечивалось формирование правильных выходных сигналов в соответствии со всеми примерами обучающей выборки;
- 2) обеспечивалось формирование правильных выходных сигналов в соответствии со всеми возможными входными сигналами, которые не вошли в обучающую выборку (обобщение)

Второе требование в значительной степени осложняет формирование обучающей выборки. В общем виде эта задача в настоящее время не решена, однако для всех известных случаев было найдено частное решение.

Обучающая выборка должна содержать сбалансированное множество образцов – примеров решения задачи. Должны быть представлены как положительные, так и отрицательные примеры образцов различных классов для задач классификации. При решении

сложных важных практических задач, проводят предварительный статистический анализ обучающей выборки.

### **6.2.2. Оптимизация числа нейронов в скрытых слоях**

Если для решения задачи используется ИНС – двухслойная сигмоидальная нейросеть (универсальный аппроксиматор), то число нейронов в скрытом слое и размер обучающей выборке связаны формулой:

$$2 \cdot (l + n + m) \leq K \leq 10 \cdot (g + n + m),$$

где  $n$  – число входов ИНС;  $m$  – число выходов ИНС (число нейронов выходного слоя);  $g$  – число нейронов входного слоя ИНС;  $K$  – размер обучающей выборки.

### **6.2.3. Проблема переобученности сети**

Одной из проблем при обучении ИНС является проблема переобученности сети – слишком сильной адаптации нейросети к особенностям обучающей выборки, а как следствие, ухудшение свойств обобщения – снижения качества работ сети на данных, которые не вошли в обучающую выборку, но которые относятся к тому же классу задач, что и примеры в обучающей выборке. Такое происходит, если: вычислительные возможности ИНС превосходят требуемые в данной задаче и процедура обучения нейросети была выполнена слишком качественно.

Для борьбы с этой проблемой используют разделение обучающей выборки на две части:

- 1) выборка для коррекции весов (работа и обучение ИНС по каждому образцу);
- 2) выборка для оценки качества обучения (только работа ИНС по каждому образцу).

При обучении ИНС используется выборка для коррекции весов. В ходе обучения оценивается работа сети на выборке для оценки качества обучения. Оценка происходит путем сравнения суммарных ошибок работы ИНС на обеих выборках. Если результат обучения (суммарная ошибка ИНС) на выборке для коррекции весов становится на заданный порог меньше, чем ошибка на выборке для оценки качества, то выполняется один из вариантов:

- 1) останов обучения, сеть считается обученной; при продолжении обучения ошибка на оценочной выборке будет ухудшаться;

2) упрощение структуры ИНС (снижение числа нейронов и соединений между ними) и повторение процедуры обучения.

Таким образом, при формировании обучающей выборки, можно предусмотреть разбиение её на выборку для оценки качества и выборку для коррекции весов.

#### **6.2.4. Оценка результатов экспериментов**

Качество работы ИНС при решении практической задачи принято оценивать:

- 1) по суммарной ошибке ИНС по всей выборке;
- 2) по максимальной ошибке ИНС по всей выборке.

Команда проекта должна решить какой показатель более важен для выбранной задачи.

### **6.3. Предварительное обсуждение**

#### **6.3.1. Контрольная выборка**

Всегда необходимо использование контрольной выборки?

Максимальная оценка 1.

#### **6.3.2. Время обучения и работы нейросетей**

Как уменьшить время обучения и работы нейросетей?

Максимальная оценка 1.

#### **6.3.3. Критерии оценки работы нейросетей**

Предложите критерии оценки работы нейросетей.

Максимальная оценка 1.

### **6.4. Тезисы вебинара**

- 1) цели и задачи модуля;
- 2) формирование обучающей выборки;
- 3) проведение экспериментов над обучающей выборкой;
- 4) оптимизация числа нейронов в скрытых слоях;
- 5) проблема переобученности сети
- 6) оценка результатов экспериментов;
- 7) как улучшить работу нейросети;
- 8) представление результатов экспериментов; использование диаграмм; вычисление статистических показателей.

### **6.5. Список литературы**

1) Коробейников, А. В. Программирование нейронных сетей: учебно-методическое пособие для выполнения самостоятельных работ // Ижевск: Изд-во ИжГТУ, 2012. – 48 с.

2) Хайкин, С. Нейронные сети: полный курс. – 2-е изд. – М. ; СПб. ; Киев : Вильямс, 2006. – 1104 с.

3) Уоссермен, Ф. Нейрокомпьютерная техника : Теория и практика. – М. : Мир, 1992. – 184 с.

4) В. Гмурман Теория вероятностей и математическая статистика (комплект из 2 книг) // М. : Юрайт, 2013. – 884 с.

## **6.6. Практические задания**

### **6.6.1. Обучающая выборка**

Сформировать обучающую выборку для вашей задачи.

Формат результата: *xls, ods*. Максимальная оценка 5. Критерии оценки: размер выборки (0-2), сбалансированность выборки (0-1), корректность решения (0-2).

### **6.6.2. Эксперименты**

Провести эксперименты по решению задачи вашей программой по образцам обучающей выборки. Оформить результаты в виде таблицы.

Формат результата: *xls, ods*. Максимальная оценка 5. Критерии оценки: корректность решения (0-4), сбалансированность экспериментов (0-2), количество экспериментов (0-2), качество визуализации (0-2).

### **6.6.3. Анализ результатов**

Провести анализ результатов экспериментов.

Формат результата: *doc, rtf, odt*. Максимальная оценка 5. Критерии оценки: корректность анализа (0-2), количество пунктов анализа (0-2), полнота описания (0-1).

### **6.6.4. Выводы и предложения**

Выводы и предложения по улучшению работы нейросети.

Формат результата: *doc, rtf, odt*. Максимальная оценка 5. Критерии оценки: корректность выводов и предложений (0-2), полнота описания (0-1), количество выводов и предложений (0-2).

## **6.7. Заключительное обсуждение**

### **6.7.1. Обоснуйте объем обучающей выборки**

Обоснуйте необходимый объем обучающей выборки для вашей задачи.

Максимальная оценка 1.

#### 6.7.2. Улучшение качества решения задачи

Возможно ли улучшение качества решения вашей задачи? За счет каких модификаций нейросети этого добиться?

Максимальная оценка 1.

#### 6.7.3. Перекрестная оценка результаты экспериментов

Покажите другим командам результаты экспериментов по решению вашей задачи. Оцените результаты программ других команд. Оцените описание их оформление.

Максимальная оценка 1.

#### 6.7.4. Обсуждение выполнения модуля

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

## 7. Оформление пояснительной записки

### 7.1. Цель и задачи модуля

Цель модуля: оформление пояснительной записки.

Задачи модуля:

- 1) оформление пояснительной записки;
- 2) использование стандартов на документацию.

### 7.2. Теоретическая часть

#### 7.2.1. Требования к содержанию пояснительной записки

- 1) задание по курсовой работе;
- 2) состав команды проекта и части проекта каждого участника;
- 3) материалы по каждому модулю проекта:
  - задания модуля;
  - результаты выполнения каждого задания модуля;
  - выводы по результатам выполнения модуля;
- 4) список литературы;
- 5) исходный текст программы (приложение).

#### 7.2.2. Стандарты при оформлении программной документации Разработка базы данных:

- 1) диаграмма «сущность-связь» для БД (*ERD – Entity-Relationship Diagrams*); *MS Visio*;
- 2) концептуальная модель данных;
- 3) физическая модель данных;
- 4) *SQL*-скрипт генерации базы данных.

#### Разработка программы при процедурном подходе:

- 1) математическая постановка задачи;
- 2) словарь терминов;
- 3) иерархия функциональных диаграмм (*IDEF0*); *MS Visio*;
- 4) диаграмма потоков данных (*DFD – Data Flow Diagrams* или *SSADM*); *MS Visio*;
- 5) диаграммы переходов состояний (*STD – State Transition Diagram*); *MS Visio (ROOM)*;
- 6) диаграммы описания структурированных данных (Джексона, Орра); *MS Visio (Jackson)*;

7) диаграммы Насси-Шнейдермана для спецификации процессов (алгоритмов в блоках); *MS Visio*;

8) псевдокоды для спецификации процессов (алгоритмов в блоках); *MS Visio*;

9) структурная схема (ГОСТ); *MS Visio (Basic Flowchart)*;

10) функциональная схема (ГОСТ); *MS Visio (Basic Flowchart)*;

11) структурные карты Константайна.

**Разработка программы при объектном подходе:**

1) диаграмма вариантов использования (*use case diagram*);

2) диаграммы классов (*class diagram, object diagram*);

3) диаграммы последовательности действий (*sequence diagram*);

4) диаграммы кооперации (*communication diagram, collaboration diagram*);

5) диаграммы деятельности (*activity diagram*);

6) диаграммы состояний объектов (*state machine diagram*);

7) диаграмма пакетов (*package diagram*);

8) диаграмма компонентов (*component diagram*);

9) диаграмма размещения (*deployment diagram*).

**7.2.3. Программное обеспечение**

1) оформление пояснительной записки: *LibreOffice Writer (odt, rtf), freeware*; *MS Word 2003 (doc, rtf)*; и другие;

2) оформление таблиц и диаграмм: *LibreOffice Calc (ods), freeware*; *MS Excell 2003 (xls)*; и другие;

3) оформление графов и схем: *LibreOffice Draw (odg), freeware*; *Dia, freeware*; *yEd Graph Editor, freeware*; *MS Visio*; и другие;

4) оформление презентаций: *LibreOffice Impress (odf), freeware*; *MS PowerPoint 2003 (ppt)*; и другие;

5) оформление схем программной документации: *LibreOffice Draw (odg), freeware*; *Dia, freeware*; *yEd Graph Editor, freeware*; *MS Visio*; *Visual Paradigm*; *Sybase Power Designer*; *IBM Rational Rose*; и другие.

**7.2.4. Перечень документов ЕСПД**

1) ГОСТ 19.001-77 ЕСПД. Общие положения.

2) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.



- 3) ГОСТ 19.102-77 ЕСПД. Стадии разработки.
- 4) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.
- 5) ГОСТ 19.104-78 ЕСПД. Основные надписи.
- 6) ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.
- 7) ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.
- 8) ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
- 9) ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.
- 10) ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний.
- 11) ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.
- 12) ГОСТ 19.402-78 ЕСПД. Описание программы.
- 13) ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
- 14) ГОСТ 19.501-78 ЕСПД. Формуляр. Требования к содержанию и оформлению.
- 15) ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.
- 16) ГОСТ 19.503-79 ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
- 17) ГОСТ 19.504-79 ЕСПД. Руководство программиста.
- 18) ГОСТ 19.505-79 ЕСПД. Руководство оператора.
- 19) ГОСТ 19.506-79 ЕСПД. Описание языка.
- 20) ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.
- 21) ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в программные документы, выполняемые печатным способом.
- 22) ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
- 23) ГОСТ 19.781-90. Обеспечение систем обработки информации программное.

#### **7.2.4. Структура отчета по разработке программного обеспечения**

1) аналитический обзор: постановка задачи (техническое задание), обзор аналогов, обзор технологий, обзор программных средств, принятие основных технических решений; соответствует этапу «Техническое задание» ГОСТ «Стадии разработки»;

2) разработка системы: спецификация системы на уровне взаимодействия подсистем (смотри пункт 7.2.2); соответствует этапу «Эскизный проект» ГОСТ «Стадии разработки»;

3) разработка подсистем: подробная спецификация подсистем (смотри пункт 7.2.2); соответствует этапу «Технический проект» ГОСТ «Стадии разработки»;

4) результаты работы: изображения экранных форм, контрольные примеры работы системы; статистические параметры по испытанию системы, информация по внедрению и апробации системы, исходные тексты программы, скрипт для генерации базы данных; соответствует этапам «Рабочий проект» и «Внедрение» ГОСТ «Этапы разработки».

### **7.3. Предварительное обсуждение**

#### **7.3.1. Когда оформлять пояснительную записку**

Когда начинать оформление пояснительной записки: до, после или во время выполнения курсовой работы или её очередного модуля?

Максимальная оценка 1.

### **7.4. Тезисы вебинара**

- 1) цели и задачи модуля;
- 2) оформление пояснительной записки; когда оформлять отчет;
- 3) структура отчета по разработке ПО;
- 4) использование стандартов на документацию; программные документы.

### **7.5. Список литературы**

1) Г.С. Иванова. Технология программирования: Учебник для вузов. – 2-е изд., стереотип. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 320 с.

2) Буч Г., Рамбо Д., Джекобсон А. Язык *UML*: Руководство пользователя. – М.: ДМК Пресс, 2001.

3) Семейство стандартов ГОСТ 34. Информационная технология.

Комплекс стандартов на автоматизированные системы.

4) Семейство стандартов ЕСПД. ГОСТ 19. Единая система программной документации.

5) Семейство стандартов ГОСТ 24. Единая система стандартов автоматизированных систем управления. Система технической документации на АСУ.

6) Семейство стандартов ГОСТ Р ИСО/МЭК. Программная инженерия.

7) Семейство стандартов ГОСТ Р ИСО/МЭК. Информационная технология.

8) Семейство стандартов *IEEE* по *Software Engineering*. Под эгидой *Software Engineering Standards Committee of the IEEE Computer Society*.

9) Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998.

## **7.6. Практические задания**

### **7.6.1. Оформление пояснительной записки**

Оформить пояснительную записку по курсовой работе, включающий в себя все разработанные материалы.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 10. Критерии оценки: полнота описания (0-5), соответствие стандартам (0-5).

### **7.6.2. Создание глоссария**

Создать глоссарий терминов, которые вы посчитали важными при выполнении курсовой работы.

Формат результата: *doc*, *rtf*, *odt*. Максимальная оценка 2. Критерии оценки: корректность решения (0-1), количество терминов (0-1).

## **7.7. Заключительное обсуждение**

### **7.7.1. Перекрестная оценка пояснительных записок**

Покажите другим командам пояснительную записку по своей курсовой работе. Оцените пояснительные записки других команд.

### **7.7.2. Обсуждение выполнения модуля**

Что вызвало сложности при выполнении модуля? Чтобы вы предложили изменить в модуле?

## **Приложение. Форма титульного листа пояснительной записки**

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Ижевский государственный технический университет имени М. Т. Калашникова»

Кафедра «Программное обеспечение»

Пояснительная записка  
по курсовой работе по дисциплине  
«Нейрокомпьютерные системы»  
на тему  
«Генетический алгоритм обучения нейросетей»

Выполнили:

1) магистрант группы М03-191-1	Иванов И. И.;
2) магистрант группы М03-191-1	Петров П. П.;
3) магистрант группы М03-191-1	Сидоров С. С.

Принял

Коробейников А. В.

Ижевск  
2013

*Учебное издание*

**Изучение нейронных сетей,  
нечеткой логики и генетических алгоритмов  
на основе метода проектов**

Учебно-методическое пособие по дисциплинам  
«Нейрокомпьютерные системы»,  
«Нечеткая логика и генетические алгоритмы»,  
«Методы оптимизации. Нейронные сети»,  
«Системы искусственного интеллекта»,  
«Математические основы искусственного интеллекта»

Составитель:  
**Коробейников** Александр Васильевич

В редакции составителя