

Федеральное агентство по образованию

Московский инженерно-физический институт
(государственный университет)

О.А. Мишулина, А.Г. Трофимов, М.В. Щербинина

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО КУРСУ
«ВВЕДЕНИЕ В ТЕОРИЮ НЕЙРОННЫХ СЕТЕЙ»**

Рекомендовано к изданию УМО «Ядерные физика и технологии»

Москва 2007

УДК 004.032.26(076.5)

ББК 32.813я7

М 71

Мишулина О.А., Трофимов А.Г., Щербинина М.В. **Лабораторный практикум по курсу «Введение в теорию нейронных сетей»: учебно-методическое пособие.** – М.: МИФИ, 2007. – 112 с.

Пособие содержит описание лабораторных исследований с целью изучения функциональных возможностей, методов обучения и областей практического использования нейронных сетей различных парадигм. Изложение материала построено так, чтобы обеспечить возможность самостоятельного выполнения лабораторных работ, оформления результатов исследований и самоконтроля знаний по основным теоретическим вопросам.

Предназначено для студентов и аспирантов, изучающих курс «Введение в теорию нейронных сетей», а также специалистов, желающих самостоятельно ознакомиться с теорией нейронных сетей и освоить нейросетевые технологии.

Пособие подготовлено в рамках Инновационной образовательной программы.

Рецензент канд. техн. наук, доц. М.В. Марковский

ISBN 978-5-7262-0840-4

© *Московский инженерно-физический институт
(государственный университет), 2007*

Редактор М.В. Макарова

Подписано в печать 30.10.2007. Формат 60x84 1/16.

Печ. л. 7,0. Уч.-изд. л. 7,0. Тираж 150 экз.

Изд. № 3/32 Заказ № 0-615

*Московский инженерно-физический институт
(государственный университет).
115409, Москва, Каширское ш., 31*

*Типография издательства «Тровант».
г. Троицк Московской обл.*

С О Д Е Р Ж А Н И Е

Предисловие	4
Лабораторная работа 1	
Математические модели искусственных нейронных сетей. Сеть Хемминга	6
Лабораторная работа 2	
Распознавание образов. Персептрон Розенблатта	24
Лабораторная работа 3	
Многослойные нейронные сети. Метод обратного распространения ошибки	50
Лабораторная работа 4	
Применение многослойных нейронных сетей для решения прикладной задачи обработки данных.	67
Лабораторная работа 5	
Ассоциативная память. Сеть Хопфилда	74
Лабораторная работа 6	
Самообучение нейронных сетей. Кластеризация данных.	99
Рекомендуемая литература	112

ПРЕДИСЛОВИЕ

Лабораторный практикум сопровождает курс лекций «Введение в теорию нейронных сетей», который более десяти лет читается в МИФИ студентам, специализирующимся в области прикладной математики и информатики. Возможность построения высокоточных и эффективных нейросетевых систем обработки данных и огромное разнообразие их практических приложений требуют изучения будущими специалистами основ нейросетевых технологий. Лабораторный практикум знакомит студентов с наиболее распространенными нейросетевыми парадигмами, позволяет понять принципы выбора архитектуры, обучения и тестирования нейронных сетей. Для подготовки настоящего пособия использованы материалы предыдущего издания*, которые существенно переработаны и дополнены новой лабораторной работой.

В практикум включены шесть лабораторных работ.

Первая работа является вводной. Она знакомит с моделью технического нейрона и на примере простейшей нейронной сети – сети Хемминга – демонстрирует возможность нейросетевого решения задачи распознавания образов.

Во второй лабораторной работе рассматриваются более сложные нейросетевые парадигмы, ориентированные на решение задач распознавания образов в разных формальных постановках. Архитектуры изучаемых нейронных сетей выстраиваются с использованием элементарных персептронов – нейронов с «жесткой» активационной характеристикой. Студенты экспериментально исследуют возможность обучения элементарного персептрана по правилу Уидроу – Хоффа и проверяют справедливость теоремы Розенблatta о сходимости этой процедуры при выполнении условия усиленной разделимости классов. В лабораторной работе студенты знакомятся с двумя архитектурами, методами обучения и функциональными возможностями нейронных сетей, решающих задачу классификации данных для n попарно линейно разделимых и двух линейно неразделимых классов.

Третья лабораторная работа посвящена изучению самой распространенной в практических приложениях архитектурой нейронной сети – многослойного персептрана, обладающего универсальной способностью к аппроксимации непрерывных функций многих переменных. На учебном примере, который генерируется каждым студентом в соответствии с индивидуальным заданием, рассматриваются особенности организации про-

* Мишулина О.А., Лабинская А.А., Щербинина М.В. Лабораторный практикум по курсу «Введение в теорию нейронных сетей». – М.: МИФИ, 2000.

цесса обучения многослойного персептрона методом обратного распространения ошибки.

В четвертой лабораторной работе полученные студентами навыки по работе с многослойным персептроном применяются для решения содержательных задач аппроксимации функций, прогноза временных рядов или классификации данных в соответствии с индивидуальным заданием.

В пятой лабораторной работе выполняются экспериментальные исследования свойств и возможностей практического применения рекуррентной нейронной сети, предложенной Хопфилдом и рассматриваемой как модель ассоциативной памяти. На практических примерах студенты изучают способность предварительно обученной по эталонным образцам нейронной сети Хопфилда восстанавливать объекты, поступающие на сеть со значительными искажениями.

Заключительная лабораторная работа знакомит студентов с самообучающимися нейронными сетями. Соревновательный принцип, используемый при самообучении, изучается на ряде примеров кластеризации данных. Эксперименты выполняются на простейшем соревновательном слое нейронов и самоорганизующейся карте Кохонена.

Выполнение каждой лабораторной работы требует предварительного ознакомления с теорией по учебным пособиям или монографиям, представленным в списке рекомендуемой литературы. Авторы умышленно не включили в пособие теоретические разделы, предложив студентам только серии контрольных вопросов по всем изучаемым темам.

Результаты экспериментальных исследований при выполнении лабораторных работ представляются в форме отчета. Пособие содержит указания по структуре отчета, графическому и табличному представлению результатов лабораторных исследований. Для контроля понимания студентом существа проводимых исследований в отчет вводятся текстовые фрагменты, в которых студент должен изложить выводы по полученным экспериментальным результатам и проанализировать их соответствие теоретическим положениям.

Пособие фактически является лабораторным журналом, в котором представлена полная информация для самостоятельной работы студента, изучающего теорию нейронных сетей.

Программные разработки, используемые в лабораторном практикуме, выполнены А.А. Лабинской, И.А. Кругловым и А.Г. Трофимовым.

Лабораторная работа 1

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ. СЕТЬ ХЕММИНГА

1.1. Цель и задачи лабораторной работы

Цель: изучение особенностей функционирования технического нейрона и математических моделей нейронных сетей различных архитектур; исследование нейросетевых способов реализации соревновательной активационной характеристики; анализ решения задачи классификации векторов биполярных признаков с помощью нейронной сети Хемминга на примерах.

В работе подробно исследуется задача отнесения биполярного входного вектора (возможные значения +1 или -1) размерности 30 к одному из трех классов, заданных аналогичными эталонными векторами. Вектора в лабораторной работе преобразованы к прямоугольной матрице 6x5 и представлены графически. Клетки матрицы, соответствующие признакам со значением +1, представлены черным цветом, а (-1) – белым. Подобная матрица может интерпретироваться как «клеточное» изображение, а задача классификации – как задача распознавания образов.

В качестве классификатора используется сеть Хемминга. Сеть Хемминга содержит слой нейронов, к которому применяется соревновательная активационная характеристика. Каждый нейрон слоя Хемминга соответствует одному эталонному представителю класса. На вход сети Хемминга поступает биполярный вектор, который требуется классифицировать. На выходе сеть Хемминга формирует выходное значение 1 для нейрона, соответствующему классу принадлежности входного вектора, и 0 – для всех остальных нейронов.

В работе рассматриваются две реализации соревновательной активационной характеристики: MAXNET прямого распространения и рекуррентная MAXNET.

Эксперимент 1 посвящен изучению постановки задачи классификации объектов, представленных векторами биполярных признаков.

В эксперименте 2 на примере сети MAXNET прямого распространения на три входа изучаются особенности функционирования технического нейрона с различными активационными характеристиками и свойства сети прямого распространения.

В эксперименте 3 студент знакомится с рекуррентной сетью MAXNET и изучает ее функциональные возможности.

В заключительном эксперименте исследуются особенности решения задачи классификации на нейронной сети Хемминга.

Далее в описании для синаптических коэффициентов нейронов используются следующие обозначения: w_{ij} – синаптический коэффициент передачи значения выхода j -го нейрона к i -му нейрону в нейронной сети; v_{ij} – синаптический коэффициент от j -го входа к i -му нейрону.

Программное обеспечение для лабораторной работы подготовлено А.А. Лабинской.

1.2. Варианты индивидуальных заданий

В каждом варианте в графическом виде заданы эталонные представители трех классов (рис. 1.1). При выполнении лабораторной работы эти данные используются для определения архитектуры и расчета синаптических коэффициентов сети Хемминга. Эталонные представители не меняются в ходе лабораторной работы. Входной вектор варьируется в зависимости от эксперимента.

1.3. Руководство пользователя программы Hamming

Программа позволяет определять класс принадлежности объектов, представленных вектором признаков длиной 30. Число классов фиксировано и равно трем. Каждый класс задается своим эталонным представителем. Для классификации используется сеть Хемминга. В программе предоставлена возможность выбора соревновательной активационной характеристики: MAXNET *прямого распространения* или *рекуррентная* MAXNET.

1.3.1. Запуск программы

Программа запускается из командной строки системы MATLAB по команде `hamming`. Вызывается стартовое окно программы, которое содержит краткие сведения о решаемой задаче (рис. 1.2).

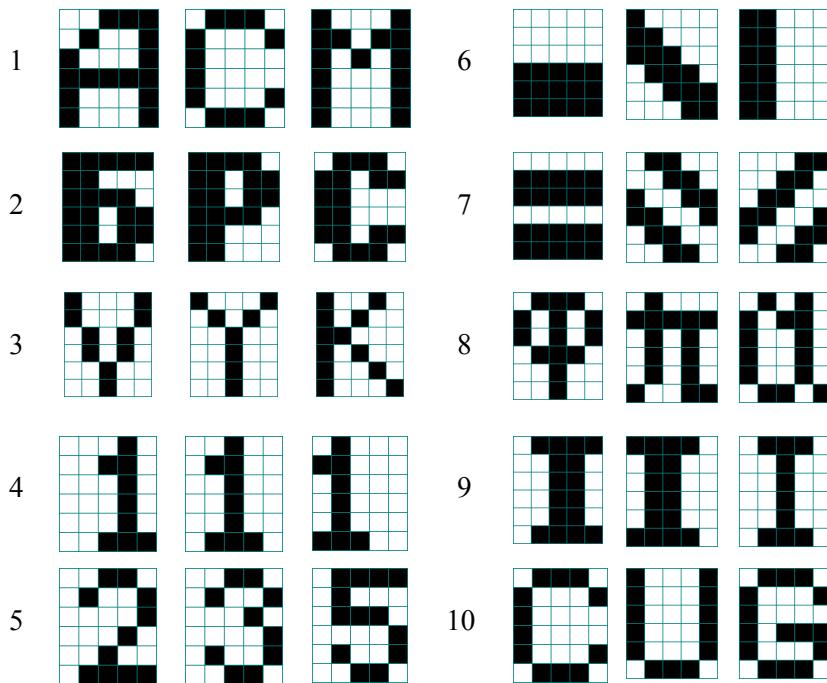


Рис. 1.1. Варианты для проведения экспериментов

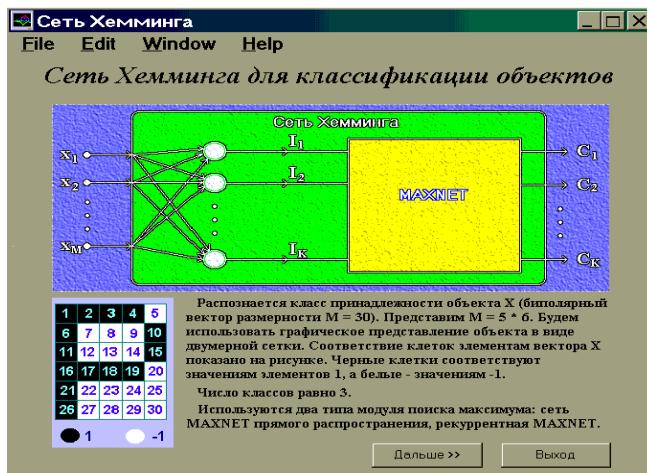


Рис. 1.2. Стартовое окно

1.3.2. Ввод входного объекта и эталонов классов

Управление классификатором осуществляется из главного окна программы, которое показано на рис. 1.3. Для удобства ввода и визуального восприятия векторов признаков объекта и эталонов классов предусмотрены специальные панели с матрицами черных и белых клеток размером 6×5 . Панель ввода для объекта находится в левой части экрана и обозначена «объект X». Панели эталонных представителей классов расположены в правой части экрана, пронумерованы и обозначены как «эталоны классов». Для ввода векторов признаков требуется нажатиями мыши на клетки матриц (цвет клетки после однократного нажатия инвертируется) установить нужные изображения объекта и эталонов, после чего включить флаг «Готово» внизу окна. В результате устанавливаются новые значения весов связей нейронов рабочего слоя Хемминга.

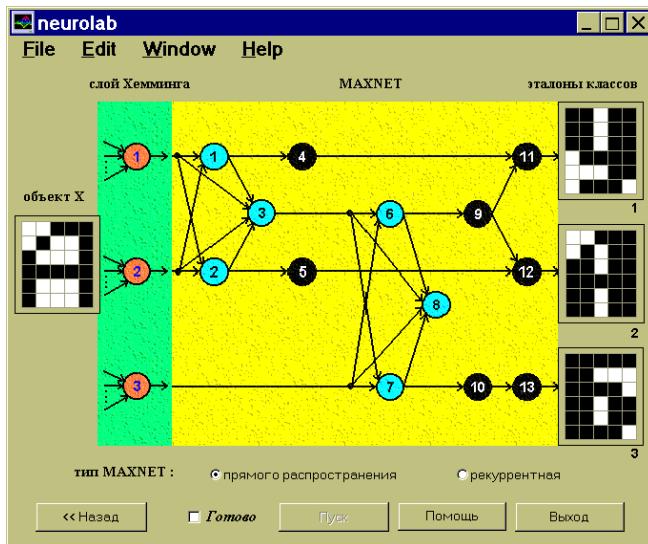


Рис. 1.3. Главное окно. Выбрана MAXNET прямого распространения

1.3.3. Выбор способа реализации соревновательной активационной характеристики

Способ реализации соревновательной активационной характеристики (тип модуля MAXNET: прямого распространения или ре-

куррентный) устанавливается переключателями в нижней части экрана. Соответственно меняется изображение MAXNET в центре экрана (рис. 1.3 и 1.4).

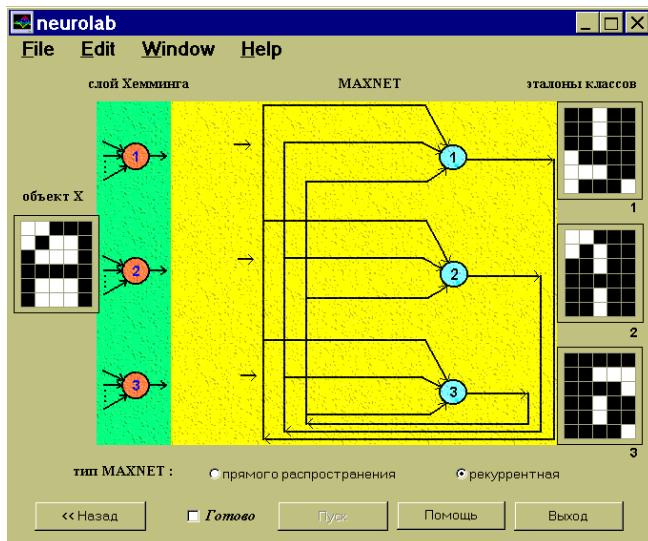


Рис. 1.4. Главное окно. Выбрана рекуррентная MAXNET

1.3.4. Ввод синаптических коэффициентов и смещений рабочего слоя сети Хемминга

Изменение значений синаптических коэффициентов может производиться автоматически или вручную.

Автоматическая установка коэффициентов происходит после внесения изменений в графические изображения эталонных представителей и нажатия кнопки «Готово» внизу окна.

Для ручного ввода весов связей и смещения каждого нейрона рабочего слоя Хемминга предусмотрено специальное диалоговое окно (рис. 1.5), которое вызывается нажатием мыши на изображение *соответствующего нейрона* слоя. Левая часть этого окна предназначена для ввода весов связей и содержит 30 редактируемых элементов, помеченных как x_1, \dots, x_{30} в соответствии с элементом входного вектора, подаваемого на данный вход нейрона.

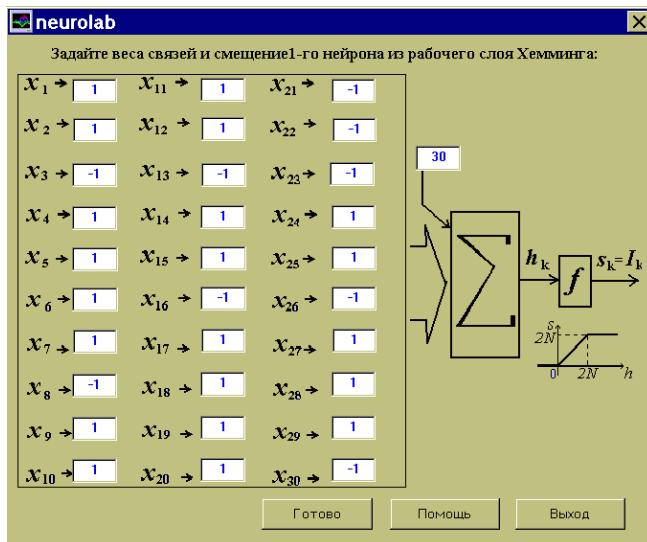


Рис. 1.5. Окно ввода весов и смещения первого нейрона рабочего слоя сети Хемминга

1.3.5. Ввод коэффициентов MAXNET

Для того чтобы просмотреть установленные или ввести новые значения весов связей сети MAXNET, нужно нажать мышью на изображение MAXNET в главном окне. В зависимости от того, какой тип MAXNET установлен в данный момент, появится окно ввода весов связей либо сети прямого распространения (рис. 1.6), либо рекуррентной сети (рис. 1.7). Следует отметить, что после внесения изменений в коэффициенты MAXNET не предусмотрена возможность их автоматического восстановления.

1.3.6. Проведение классификации

После установки или проверки всех изменяемых параметров (с помощью описанных средств) можно запускать классификатор. Для этого нужно установить флаг «Готово». В результате кнопка «Пуск» внизу экрана становится доступной. Классификатор запускается по нажатии кнопки «Пуск».

В процессе классификации на экран в области изображения сети Хемминга рядом с изображениями нейронов выводятся их выход-

ные значения. На экране также отображается текущий номер такта функционирования MAXNET. По окончании процесса классификации рамка панели эталонного представителя класса, к которому был отнесен предъявленный объект, выделяется цветом.

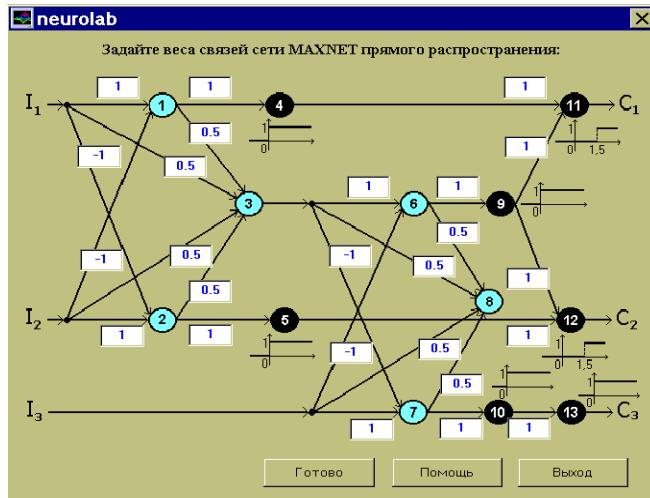


Рис. 1.6. Окно ввода весов связей в сети MAXNET прямого распространения

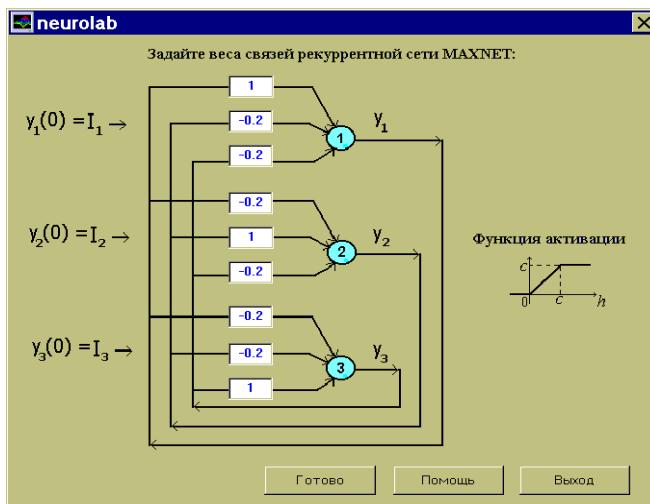


Рис. 1.7. Окно ввода весов связей в рекуррентной сети MAXNET
12

Для повторного запуска классификатора достаточно снова установить флаг «Готово» (в результате информация о предыдущей работе классификатора пропадает) и нажать кнопку «Пуск».

1.3.7. Некоторые особенности работы программы

Программно реализована возможность проведения классификации при двух одинаковых максимальных значениях входов в MAXNET прямого распространения. При этом первый по номеру из одинаковых входов несколько увеличивается, а второй уменьшается.

Определение результата классификации объекта при работе с MAXNET прямого распространения осуществляется не только по выделенной рамке эталонного представителя, но и по выходам MAXNET. При этом возможны следующие ситуации:

- 1) на всех выходах MAXNET наблюдается значение 0;
- 2) на одном выходе MAXNET наблюдается значение 1;
- 3) на двух или трех выходах MAXNET наблюдается значение 1.

В первой ситуации на экране появится сообщение «Cannot classify!», а во второй и третьей – сообщений не возникает, происходит выделение рамкой соответствующих эталонных представителей.

1.3.8. Как классифицировать объект

Последовательность действий, необходимых для классификации объекта в программе «Классификатор Хемминга», проиллюстрирована на рис. 1.8.

1. Ввести вектора признаков эталонных представителей классов, инвертируя нажатиями мыши цвет клеток на панелях, обозначенных «эталоны классов» и помеченных соответствующими номерами.

2. Аналогичным способом ввести вектор признаков объекта, подлежащего классификации. Соответствующая панель с черно-белыми клетками помечена «объект X».

3. Для каждого из трех нейронов рабочего слоя Хемминга проанализировать автоматически установленные веса связей и смещения (либо установить другие значения). Для этого нажатием на изображение соответствующего нейрона вызвать диалоговое окно настройки параметров этого нейрона.

4. Выбрать тип MAXNET, установив соответствующий переключатель внизу окна.
5. Проконтролировать автоматически установленные веса связей (либо установить другие значения) сети MAXNET выбранного типа. Для этого нажатием на изображение MAXNET вызвать диалоговое окно установки весов.
6. Установить флаг «Готово».
7. Нажать кнопку «Пуск».

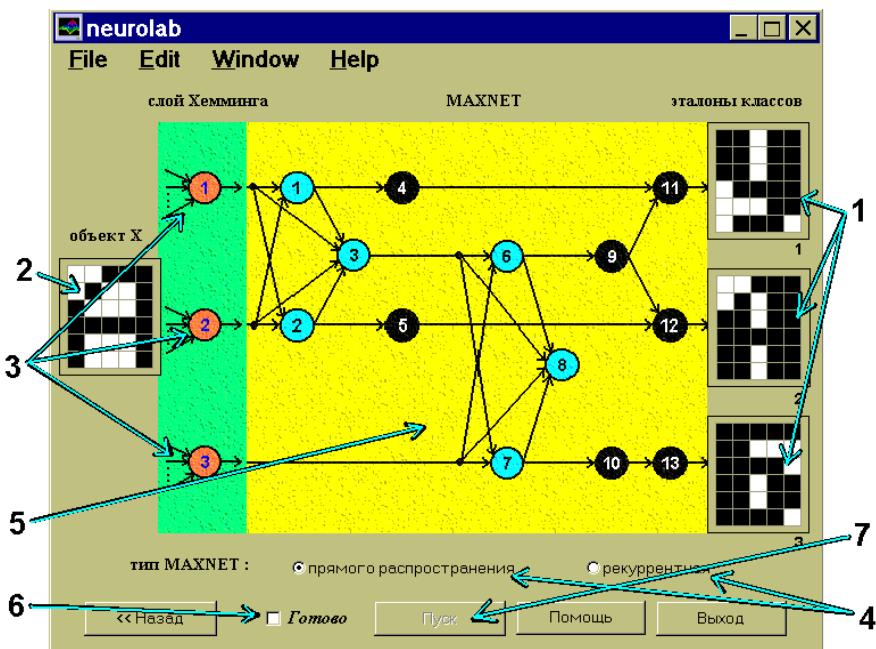


Рис. 1.8. Последовательность действий при классификации объекта

1.4. Содержание лабораторных исследований

Перед началом исследований студент запускает программу, вводит эталонные векторы классов по своему варианту (см. разд. 1.3.1 и 1.3.2) и зарисовывает графические образы в отчет.

1.4.1. Исследование возможности классификации объекта в зависимости от степени отличия его признаков от эталонного представителя класса (эксперимент 1)

Особенностью постановки задачи является то, что входной объект X будет отнесен к тому классу, для которого расстояние по Хеммингу (число несовпадающих разрядов) до входного объекта минимально. В данном эксперименте используется этот критерий. Начиная с входного объекта, близкого по расстоянию к некоторому эталону, предлагается «удалять» входной объект до тех пор, пока другой эталон не станет более близким.

Эксперимент проводится с MAXNET прямого распространения (см. разд. 1.3.3). В качестве входного объекта X рекомендуется вначале задать вектор, равный некоторому эталонному представителю. Учитывая первую из особенностей программной реализации, указанную в разд. 1.3.7, рекомендуется использовать не первый по счету эталонный представитель. Сначала следует убедиться, что объект классифицируется правильно. Далее требуется искажать объект X, внося каждый раз 1 – 2 произвольных изменения клеток матрицы до тех пор, пока объект не будет распознаваться как принадлежащий некоторому другому классу. Необходимо зафиксировать в отчете в п. 1 максимальное расстояние по Хеммингу (число несовпадающих разрядов с первоначальным эталоном) и записовать соответствующий объект для случая, когда классификация еще происходит правильно, т. е. объект X до последнего искажения. Следует отметить, что не требуется искать глобального максимума расстояния по Хеммингу, а взять первое подходящее решение.

1.4.2. Исследование свойств MAXNET прямого распространения (эксперимент 2)

Для изучения особенностей функционирования технического нейрона и сети прямого распространения студенту предлагается воздействовать на работу сети, изменяя синаптические коэффициенты нейронов. Первоначально, когда изменения еще не внесены, сеть верно выполняет свою задачу по классификации входного объекта. Студенту предлагается «сломать» работу сети, т. е. добиться «неправильной» классификации.

Под «неправильной» классификацией следует понимать результат отнесения объекта X к любому классу, кроме первоначального. При этом должна наблюдаться ситуация 2, отмеченная в особенностях работы программы (см. разд. 1.3.7). Ситуации 1 и 3 под «неправильной» классификацией не понимаются.

Предлагается «сломать» верную работу сети тремя способами: путем изменения коэффициентов только нейронов «сравнения» (нейроны № 1, 2, 6, 7), только нейронов, «хранящих максимум» (нейроны № 3, 8), и только нейронов – «индикаторов» (нейроны № 4, 5, 9, 10, 11, 12, 13).

Для выполнения эксперимента в качестве входного объекта X необходимо задать некоторый эталон (не первый по счету). Внести 2 – 3 искажения объекта. Зафиксировать в отчете номер класса, к которому принадлежит входной объект без искажения сети.

Добиться «неправильной» классификации входного объекта путем изменения синаптических коэффициентов нейронов «сравнения» (нейроны № 1, 2, 6, 7), не изменяя коэффициентов других нейронов (см. разд. 1.3.5). Зафиксировать в отчете значения *только измененных* коэффициентов согласно указанию к записи коэффициентов (см. далее). *Восстановить измененные коэффициенты, путем введения коэффициентов нейросетевого компаратора.*

Повторить эксперимент для коэффициентов нейронов, «хранящих максимум» (нейроны № 3, 8), и зафиксировать результат в отчете, а затем для коэффициентов нейронов – «индикаторов» (нейроны № 4, 5, 9, 10, 11, 12, 13), также фиксируя результат в отчете. Перед каждым экспериментом искаженные коэффициенты восстанавливаются.

1.4.3. Исследование сходимости рекуррентной сети MAXNET (эксперимент 3)

На примере реализации соревновательной активационной характеристики в виде рекуррентной сети изучаются особенности функционирования динамической сети MAXNET. Рекуррентная сеть после получения начального импульса функционирует некоторое число тактов до устойчивого состояния – аттрактора. При фиксированном числе нейронов на число тактов функционирования динамической MAXNET оказывают влияние значения входов

(т. е. выходов слоя Хемминга) и значения синаптических коэффициентов, определяемые параметром ϵ . Предлагается исследовать зависимость числа тактов сходимости от параметра ϵ при фиксированных значениях входов.

В качестве входного объекта X следует взять некоторый эталон (не первый по счету). Внести 2 – 3 искажения объекта. Выбрать реализацию соревновательной активационной характеристики в виде рекуррентной MAXNET (см. разд. 1.3.3). Зафиксировать в отчете число нейронов рекуррентной сети. Изменить значение параметра ϵ синаптических коэффициентов сети (см. разд. 1.3.5). Запустить сеть (см. разд. 1.3.6) и зафиксировать в отчете число тактов сходимости сети. Эксперимент необходимо провести три раза, каждый раз меняя значение ϵ . Оформить вывод по эксперименту 3 в отчете.

1.4.4. Исследование свойств нейронов рабочего слоя сети Хемминга (эксперимент 4)

В эксперименте рассматриваются особенности решения задачи классификации на нейронной сети Хемминга. В первой части эксперимента изучается архитектура слоя Хемминга (число нейронов и значения синаптических коэффициентов). Во второй части исследуется робастность сети (способность сети верно выполнять свою задачу при зашумлении синаптических коэффициентов).

Нахождение минимального и максимального выходов нейронов слоя Хемминга. При проведении эксперимента необходимо учесть, что каждый нейрон при формировании потенциала выполняет скалярное произведение входного вектора и вектора синаптических коэффициентов. Зная вектор синаптических коэффициентов и значение смещения, можно легко подобрать входной вектор так, чтобы потенциал нейрона был максимальен или минимальен. Значения синаптических коэффициентов нейронов и смещений остаются неизменными в этом эксперименте, меняются только значения входного вектора – объекта X .

Следует выбрать произвольный нейрон слоя Хемминга, зарегистрировать значения его синаптических коэффициентов и смещения (см. разд. 1.3.4). Далее необходимо менять входной объект X так, чтобы выход *выбранного* нейрона был максимальен. Значения выходов других нейронов и результат классификации в данном

эксперименте значения не имеют. В отчете необходимо зафиксировать номер выбранного нейрона, максимальное значение выхода и соответствующий входной объект X.

Аналогичный эксперимент проводится для поиска минимального значения выхода.

Исследование рабочности слоя Хемминга. При проведении эксперимента также необходимо учесть принцип расчета потенциала нейрона, использованный в первой части эксперимента. Зная координаты векторов синаптических коэффициентов нейронов и входного вектора, можно предсказать, как изменятся значения потенциалов нейронов слоя Хемминга при изменении координат векторов синаптических коэффициентов.

Вначале следует ввести входной объект X как некоторый эталон (не первый по счету) с 2 – 3 искажениями. Чтобы ограничить число тактов функционирования сети Хемминга, в качестве реализации соревновательной активационной характеристики следует взять MAXNET прямого распространения. Далее нужно убедиться, что сеть верно классифицирует входной объект. Зафиксировать объект X и номер класса, к которому он отнесен, в отчете. Входной объект X в эксперименте не меняется.

В эксперименте А проводится имитация разрыва канала. Считается, что сигнал не проходит от входного объекта X до рабочих нейронов сети Хемминга. Под j -м каналом следует понимать связь между j -й координатой входного вектора и j -м входом сети. Чтобы имитировать разрыв j -го канала необходимо изменить j -й синаптический коэффициент *всех* нейронов рабочего слоя сети Хемминга (см. разд. 1.3.4). Далее следует запустить сеть и проверить, сохраняет ли сеть возможность верной классификации. Если сеть продолжает правильно работать, то номер разорванного канала следует зафиксировать в отчете и (не восстанавливая предыдущий разрыв) добавить разрыв другого канала. Если же сеть исказила результат классификации, то эксперимент А окончен. В данном эксперименте не требуется искать максимально возможное число разорванных каналов. Необходимо зафиксировать первый подходящий результат. Студентам предлагается заранее продумывать, какой канал следует разорвать, чтобы быстрее справиться с заданием. Следует отметить, что следить за тем, как разрыв канала повлиял на классификацию, нужно по значениям потенциалов рабочих ней-

ронов слоя Хемминга. В случае, если будут наблюдаться два равных и наибольших значения, можно считать, что объект X классифицируется неверно.

Перед проведением эксперимента Б *необходимо восстановить все синаптические коэффициенты*, воспользовавшись автоматической установкой (см. разд. 1.3.4).

В эксперименте Б моделируется ситуация, когда в одном канале наблюдается сильная помеха. Предполагается, что вносимая помеха не меняет знак коэффициента усиления, а лишь меняет значение его модуля. Учитывая результаты эксперимента А и значение входного вектора, студент выбирает один произвольный канал и увеличивает в нем значения модулей синаптических коэффициентов у *всех* нейронов (не меняя знак). Амплитуда шума выбирается произвольно (рекомендуется задавать значение более 100). Далее проверяется правильность классификации. Если объект X классифицируется правильно, то значения синаптических коэффициентов восстанавливаются, и эксперимент проводится с другим каналом. Если же наблюдается неверная классификация, то ее результат фиксируется в отчете. Далее амплитуда помехи постепенно снижается, пока не найдется пороговое значение (с точностью до целых), при котором входной объект классифицируется правильно. Коэффициент k соответствует отношению порогового значения синаптического коэффициента к его первоначальному значению. Значение k следует записать в отчет.

В эксперименте В необходимо обнаружить все каналы, по которым можно увеличить значение модулей синаптических коэффициентов в k раз и при этом результат классификации останется прежним. Это означает, что необходимо вносить мультипликативную помеху с коэффициентом усиления k для разных каналов, каждый раз восстанавливая значения синаптических коэффициентов. В отчете следует фиксировать номера каналов, искажение которых не повлияло на результат классификации. В целях экономии времени при проведении этого эксперимента желательно учитывать результат, полученный в эксперименте А. Кроме того, необходимо указать особенности эталонных представителей классов и входного вектора, которые объясняют нечувствительность результата относительно некоторых каналов. Соответствующий вывод следует записать в отчет.

1.5. Рекомендации к оформлению отчета

Отчет по работе должен содержать описания исходных данных и условий проведения каждого эксперимента, а также полученные результаты в числовой, графической или табличной форме и выводы.

Ниже приводится содержание отчета с краткими комментариями.

1. Исходные данные для лабораторных исследований

Указывается номер варианта, приводятся заданные эталонные представители трех классов в форме матричных черно-белых изображений.

2. Исследование возможности классификации объекта сетью Хемминга в зависимости от степени отличия признаков объекта от эталонного представителя класса (эксперимент 1)

Приводится изображение полученного в эксперименте максимально искаженного эталона, для которого еще наблюдается правильное распознавание сетью Хемминга. Фиксируется расстояние по Хеммингу между эталоном и его искаженным представлением. Делается вывод относительно надежности распознавания образов сетью Хемминга.

3. Исследование свойств MAXNET прямого распространения (эксперимент 2)

Изображается выбранный для проведения исследования эталон и подтверждается его правильное распознавание сетью Хемминга.

Приводятся измененные значения синаптических коэффициентов нейронов, при которых MAXNET выдает ошибочный результат о принадлежности объекта классу. Рассматриваются следующие постановки экспериментов:

- изменяются синаптические коэффициенты нейронов «сравнения»;
- изменяются синаптические коэффициенты нейронов, «хранящих максимум»;
- изменяются синаптические коэффициенты нейронов – «индикаторов».

4. Исследование свойств сходимости рекуррентной сети MAXNET (эксперимент 3)

Приводятся результаты трех моделирований для разных значений параметра ε . В таблице для каждого моделирования указывается значение параметра ε и полученное в эксперименте число тактов сходимости сети. Делается вывод относительно зависимости процесса сходимости от параметра ε .

5. Исследование свойств нейронов рабочего слоя сети Хемминга (эксперимент 4)

Указываются параметры архитектуры сети Хемминга (число нейронов рабочего слоя и число входов сети).

Приводятся результаты двух экспериментов *по нахождению максимального и минимального потенциалов нейронов рабочего слоя сети Хемминга*. Для каждого эксперимента указывается входное изображение и наблюдаемый максимальный или минимальный потенциал (регистрируется номер соответствующего нейрона).

В экспериментах с целью *исследования рабочности слоя Хемминга* приводятся условия моделирования: изображение входного образа и номер класса его принадлежности.

По результату эксперимента А (*имитация разрыва канала*) указываются значения измененных синаптических коэффициентов и результат работы сети в соответствии с методикой, изложенной в разд. 1.4.4.

По результату эксперимента Б (*анализ действия помехи в каналах передачи входных данных в сети Хемминга*) приводятся максимальное значение коэффициента усиления сигнала по одному каналу, при котором слой Хемминга сохраняет работоспособность, и минимальное значение, при котором наблюдается нарушение работоспособности сети Хемминга.

Для эксперимента В (*исследование чувствительности выхода сети к изменению синаптических коэффициентов нейронов рабочего слоя сети Хемминга*) приводятся номера нейронов, изменение синаптических коэффициентов которых не влияет на результат ра-

боты сети Хемминга. Даётся логическое обоснование наблюдаемого эффекта.

6. Общие выводы по проведенным исследованиям

Излагаются особенности архитектуры сети Хемминга и область ее практического применения. Приводятся свойства сети, которые наблюдались при выполнении экспериментов.

Контрольные вопросы

1. Напишите уравнение функционирования технического нейрона.
2. Какое действие над векторами входов и синаптических коэффициентов производится в процессе формирования потенциала нейрона?
3. Какой особенностью функционирования технического нейрона объясняется его действие как нелинейного преобразователя?
4. Перечислите основные виды активационных характеристик. В чем состоят особенности каждого вида?
5. Может ли в качестве активационной характеристики нейрона использоваться любая нелинейная функция? Объясните свой ответ.
6. Какая нейронная сеть называется рекуррентной?
7. Объясните особенности функционирования нейронной сети прямого распространения.
8. Напишите уравнения, характеризующие динамику рекуррентной нейронной сети в дискретном времени (скалярная форма).
9. Напишите векторно-матричные уравнения, характеризующие динамику рекуррентной нейронной сети в дискретном времени.
10. В чем состоит задача распознавания образов? Какая информация должна быть представлена об объекте при решении задачи распознавания образов?
11. Какую задачу решает сеть Хемминга?
12. В какой форме представлены характеристики объектов при решении задачи распознавания образов с помощью сети Хемминга?
13. Каким способом заданы классы при решении задачи распознавания образов с помощью сети Хемминга?

14. Сформулируйте критерий оптимальности отнесения объекта к одному из классов при применении сети Хемминга.
15. Какое содержание имеют выходные значения нейронов рабочего слоя сети Хемминга?
16. Сколько нейронов содержит рабочий слой сети Хемминга? Каковы их синаптические коэффициенты?
17. Как связаны расстояние по Хеммингу между эталонным представителем и классифицируемым входным вектором и потенциал соответствующего нейрона сети Хемминга?
18. Какая активационная характеристика используется в сети Хемминга? Как она может быть реализована?
19. Нарисуйте схему нейросетевого компаратора на два входа и объясните его работу. Какие активационные характеристики нейронов используются в схеме нейросетевого компаратора?
20. Какие ограничения накладываются на значения входов сети MAXNET прямого распространения?
21. Объясните принцип построения сети MAXNET прямого распространения при произвольном числе входов с использованием нейросетевого компаратора на два входа. Как в этом случае формируется выход сети?
22. Чем определяется число тактов функционирования сети MAXNET прямого распространения?
23. Нарисуйте схему рекуррентной сети MAXNET. Какие начальные условия устанавливаются на сети?
24. Постройте матрицу синаптических коэффициентов рекуррентной сети MAXNET на n входов.
25. Объясните принцип работы рекуррентной сети MAXNET.
26. Какие ограничения накладываются на значения входов рекуррентной сети MAXNET?
27. Чем определяется число тактов функционирования рекуррентной сети MAXNET?

Лабораторная работа 2

РАСПОЗНАВАНИЕ ОБРАЗОВ. ПЕРСЕПТРОН РОЗЕНБЛATTA

2.1. Цель и задачи лабораторной работы

Цель: исследование трех нейросетевых архитектур для решения задач классификации данных в трех различных постановках: классификация данных для двух линейно разделимых классов (задача 1); классификация данных для нескольких ($n > 2$) попарно линейно разделимых классов (задача 2); классификация при наличии двух линейно неразделимых классов (задача 3).

Все исследуемые в лабораторной работе архитектуры нейронных сетей построены на нейронах с bipolarными активационными характеристиками. Такие нейроны в дальнейшем называются элементарными персепtronами.

Первая из перечисленных выше задач решается с использованием единственного элементарного персептрана, обучаемого по аддитивному δ -правилу Уидроу – Хоффа.

Решение второй задачи достигается на двухслойной нейронной сети, в которой синаптические коэффициенты первого слоя настраиваются по правилу Уидроу – Хоффа, а второго слоя устанавливаются в соответствии с выполняемыми ими логическими функциями.

Для решения третьей задачи применяется метод Мезарда и Надала, названный ими «черепичным» алгоритмом. Он предполагает конструирование многослойного персептрана путем наращивания слоев и нейронов в слоях, стартуя с единственного нейрона. Процесс роста сети завершается, когда достигается разделимость классов на данных обучающей выборки.

В лабораторной работе исследуется сходимость процедур обучения нейросетевых классификаторов, изучается влияние параметров постановки задачи на скорость сходимости, на многочисленных примерах качественно анализируется результат классификации, полученный на обученной нейронной сети. Графические возможности используемых при выполнении лабораторной работы программных средств позволяют наглядно представить характеристики нейронных сетей, особенности процесса их обучения и результаты обработки данных.

Программное обеспечение для выполнения лабораторной работы спроектировано и разработано И.А. Кругловым.

2.2. Варианты индивидуальных заданий

№ п/п	Границы рабочей области	Задача 1				Задача 2			Задача 3		
		N_{cl}	N_p	d_x d_y	$a; b;$ c	N_{cl}	d_x	d_y	N_{cl}	d_x	d_y
1	2	3	4	5	6	7	8	9	10	11	12
1	(-100, 100) (-100, 100)	25	120	25, 30	-10; 20; 10	5 6	25	30	5 6	15	20
2	(-50, 100) (-100, 50)	25	80	25, 25	-1; -1; 25	5 6	20	20	5 6	10	12
3	(-130, 200) (-130, 200)	25	150	40, 36	-1; 3; 20	7 4	20	20	7 4	15	15
4	(-150, 100) (-100, 150)	25	75	24, 35	0,3; -0,4; 35	5 4	12	16	5 4	12	10
5	(-110, 210) (-110, 210)	20	200	35, 40	-1; 1; 10	6 4	16	20	6 4	16	9
6	(-125, 85) (-85, 125)	20	180	18, 25	-0,3; 0,5; -15	4 5	12	10	4 5	8	10
7	(-200, 100) (-100, 200)	20	130	36, 36	0,9; 0,2; 0	7 5	15	15	7 5	9	12

1	2	3	4	5	6	7	8	9	10	11	12
8	(-90, 90) (-90, 90)	20	130	12, 10	1; 0; 12	5 6	10	10	5 6	5	5
9	(-50, 250) (-100, 100)	30	110	5, 40	0,5; -0,3; -4,99	6 4	20	10	6 4	10	10
10	(-100, 100) (-50, 250)	30	120	25, 36	0,5; 0,35; -34	6 4	10	15	6 4	8	12
11	(-130, 130) (-200, 200)	50	300	40, 40	-1; 1; 50	7 5	9	20	7 5	9	15
12	(-90, 90) (-100, 100)	30	150	5, 25	-3; 4; 50	4 5	10	12	4 5	5	8

2.3. Задача классификации для двух линейно разделимых классов (задача 1)

2.3.1. Общее содержание задачи и программной модели

Сформулируем постановку решаемой задачи.

В пространстве признаков размерности M заданы *два линейно разделимых класса*. Известна обучающая выборка, содержащая некоторое количество примеров объектов первого и второго классов. Требуется построить нейронную сеть, которая произвольный объект, характеризующийся вектором признаков размерности M , отнесет к одному из двух заданных классов.

В лабораторной работе задача решается для случая, когда каждый объект характеризуется двумя признаками ($M = 2$) и потому может быть представлен точкой на плоскости. Одна часть точек относится к первому классу, а другая – ко второму ($n = 2$). В соответствии с постановкой задачи существует такая прямая, которая разделяет рассматриваемые два класса точек.

Для выполнения лабораторной работы необходимо сгенерировать обучающую выборку, т. е. создать множество точек в плоскости (X , Y) и присвоить каждой метку +1 или -1 в зависимости от принадлежности первому или второму классу. Эта задача решается в два этапа.

На первом этапе генерируются точки без указания их принадлежности классам. Точки располагаются в прямоугольнике с границами по горизонтали и вертикали (X_{\min} , X_{\max}) и (Y_{\min} , Y_{\max}) соответственно. Этот прямоугольник будет в дальнейшем называться *рабочей областью*. Множество точек (классифицируемых объектов обучающей выборки) генерируется как совокупность эллипсоидальных кластеров (число кластеров N_{cl}), центры которых случайны и равномерно распределены в рабочей области. Внутри каждого кластера число точек фиксировано и равно N_p . Координаты точек кластера распределены по нормальному закону с дисперсиями d_x и d_y по осям X и Y соответственно.

Вариант задания на лабораторное исследование (см. разд. 2.2) содержит значения всех перечисленных выше параметров для генерации случайной выборки точек.

Для разделения точек на классы Ω_1 и Ω_2 (второй этап построения обучающей выборки) проводится прямая, уравнение которой $ax + by + c = 0$ (параметры a , b , c указаны в варианте задания). Для выполнения условия усиленной разделимости классов около прямой $ax + by + c = 0$ строится коридор ширины ε , внутри которого удаляются все сгенерированные ранее точки. Далее в программе реализуется правило, согласно которому точки, удовлетворяющие условию $ax + by + c > 0$, относятся к первому классу (область Ω_1), а условию $ax + by + c < 0$ – ко второму (область Ω_2).

Сформированная обучающая выборка используется для настройки синаптических коэффициентов элементарного персептрона по *правилу Уидроу – Хоффа*.

Если $\varepsilon > 0$, поставленная выше задача классификации может быть решена на элементарном персептроне и процедура Уидроу – Хоффа завершается за конечное число тактов.

Во всех вариантах задания сначала устанавливается значение $\varepsilon = 12$.

Если при генерации обучающей выборки не удовлетворено требование линейной разделимости классов, задача классификации не

может быть решена на элементарном персептроне и процедура Уидроу – Хоффа должна быть принудительно остановлена путем задания конечного числа тактов обучения. Эта ситуация программно реализуется заданием параметра $\epsilon = 0$.

Описанные выше функциональные возможности реализованы в программе Perceptron.

2.3.2. Руководство пользователя программы Perceptron

Программа Perceptron реализует модель элементарного персептрана – простейшую форму нейронной сети, предназначенную для решения задачи классификации объектов для двух линейно разделимых классов. Описание возможностей программы приводится в следующем порядке.

1. Генерация данных. Программа позволяет произвести генерацию точек на плоскости (X, Y) в виде совокупности кластеров. Кластеры случайным образом располагаются в рабочей области и включают в себя одинаковое число точек. Пользователю предоставляется возможность задавать границы допустимой области (X_{\min}, X_{\max}) и (Y_{\min}, Y_{\max}), количество кластеров в ней (N_{cl}), количество точек в кластерах (N_p), а также значения дисперсий d_x и d_y внутри кластеров по осям X и Y соответственно.

Для изменения всех перечисленных параметров следует выбрать команду Tools>Properties в главном меню программы.

Допустимые значения варьируемых параметров представлены в табл. 2.1.

Таблица 2.1

**Пределы изменения параметров для генерации точек
в программе Perceptron**

Параметр	X_{\min}	X_{\max}	Y_{\min}	Y_{\max}	N_{cl}	N_p	d_x	d_y
Минимальное значение	-300	50	-300	300	2	1	1	1
Максимальное значение	-50	300	-50	50	50	500	40	40

Генерация данных производится автоматически при каждом запуске программы со значениями параметров, заданными по умол-

чанию, а также при каждом их изменении пользователем. В остальных случаях генерация выполняется выбором команды меню File>New.

2. Разделение данных на два класса (идентификация сгенерированных точек). Разделение данных на два класса осуществляется с помощью прямой, которая определяется уравнением $ax + by + c = 0$, где a , b и c – задаваемые пользователем параметры. Далее идентификация точек происходит по двум возможным вариантам.

- В первом варианте точки, находящиеся внутри полосы ширины ϵ относительно разделяющей прямой, удаляются. Остальным точкам присваивается номер класса соответственно положению относительно прямой. Если (x_0, y_0) – координаты точки, то ей присваивается номер класса 1, если $ax_0 + by_0 + c > 0$, и -1, если $ax_0 + by_0 + c < 0$. Номер класса точки соответствует желаемому выходу элементарного персептрона. Точки с номером класса 1 окрашиваются в красный цвет, а с номером класса -1 – в синий. Пример данных (примеров обучающей выборки), разделённых на два класса таким способом, показан на рис. 2.1.

- Второй вариант идентификации точек приводит к формированию двух линейно неразделимых классов. В отличие от первого варианта при данном способе ширина разделяющей полосы ϵ , внутри которой происходит удаление точек, устанавливается равной нулю. Точке, принадлежащей кластеру с координатами центра (x_c, y_c) , присваивается номер класса 1, если $ax_c + by_c + c > 0$, и -1, если $ax_c + by_c + c < 0$. Таким образом, к одному и тому же классу относятся все точки каждого кластера. Пример данных, разделенных на два класса таким способом, дан на рис. 2.2.

Первый вариант идентификации точек, устанавливаемый по умолчанию, гарантирует линейную разделимость классов и конечность процесса обучения персептрона. При втором варианте разделимость классов не гарантируется. Поэтому при выборе второго варианта разделения точек пользователю предоставляется возможность задать максимальное число тактов T_{\max} обучения персептрона во избежание зацикливания программы.

Допустимые диапазоны изменений задаваемых пользователем параметров для распределения точек по двум классам представлены в табл. 2.2.

Таблица 2.2

Пределы изменения параметров, характеризующих разделимость классов

Параметр	a	b	c	ε	T_{\max}
Минимальное значение	-50	-50	-100	0,5	100
Максимальное значение	50	50	100	20	60 000

Выбор способа разделения точек, задание значений параметров разделяющей прямой, а также ширины разделяющей полосы (при первом варианте идентификации точек) или максимального числа тактов обучения персептрона (при втором варианте идентификации точек) с последующим разделением классов осуществляется командой меню Tools>Separate classes. Объединение классов – Tools>Unify classes.

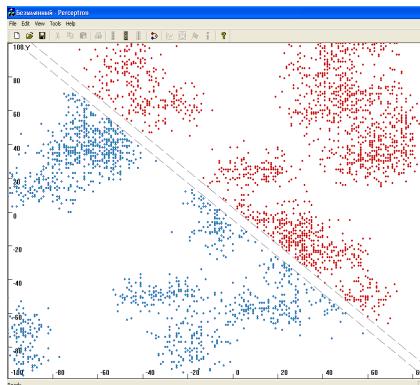


Рис. 2.1. Иллюстрация к процессу разделения данных на два класса при $\varepsilon > 0$

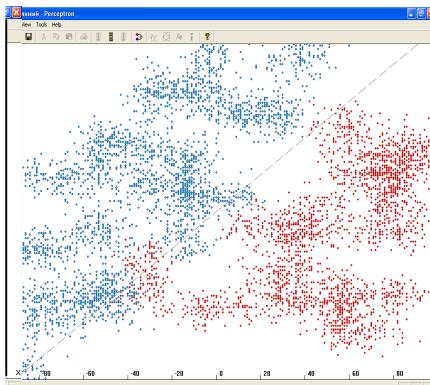


Рис. 2.2. Иллюстрация к процессу разделения данных на два класса при $\varepsilon = 0$

3. Обучение персептрана. При выборе команды главного меню Tools>Teach производится обучение персептрана по подготовленной обучающей выборке. Начальные значения синаптических коэффициентов и смещения элементарного персептрана регулируются пользователем в пределах, приведенных в табл. 2.3.

Таблица 2.3

Допустимые начальные значения синаптических коэффициентов

Параметр	$w1(0)$	$w2(0)$	$b(0)$
Минимальное значение		-50	
Максимальное значение		50	

Каждый выборочный пример обучающей выборки представляет собой двумерный вектор (x, y) и желаемый выход персептрона σ , равный +1 или -1 в зависимости от класса принадлежности вектора. Структура обучающей выборки для персептрона представлена в табл. 2.4.

Если точки были разделены на классы способом, соответствующим значению параметра $\varepsilon = 0$, то даже если на такте T_{\max} обучения не все примеры обучающей выборки распознаются персептроном верно, процесс обучения останавливается.

Таблица 2.4

Структура обучающей выборки для элементарного персептрона

№ п/п	x	y	σ
1	65	-42	1
2	-56	-95	-1
...

4. Сходимость процесса обучения персептрона. Сходимость процесса обучения персептрона в случае двух линейно разделимых классов гарантируется теоремой Розенблатта. Согласно этой теореме максимальное число тактов обучения обратно пропорционально значению θ^2 , где θ – гарантированная ширина разделяющего «коридора» между классами в расширенном пространстве признаков. Необходимо отметить, что ширина полосы ε , разделяющей классы в плоскости (X, Y) , не отождествляется с величиной θ .

Описанный способ моделирования выборочных примеров и разделяющего «коридора» ε не противоречит ситуации, представленной на рис. 2.3.

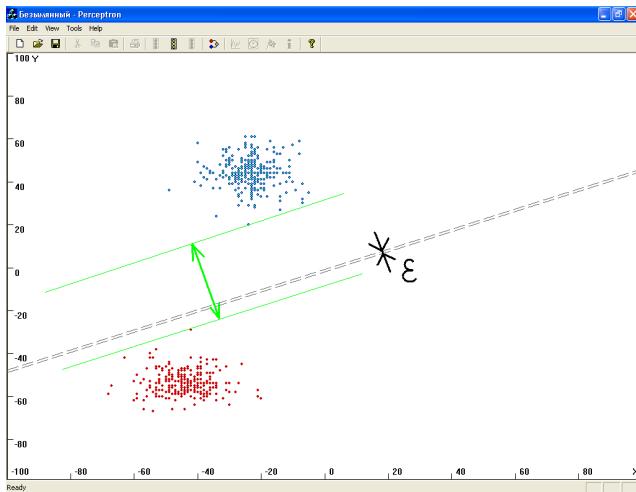


Рис. 2.3. Пример возможного расположения разделяющего «коридора» между классами

5. Просмотр процесса обучения. Информация о полном количестве тактов обучения отображается в левом верхнем углу рабочей области.

Просмотреть ход обучения персептрана можно выбором команды меню View>Learning steps. В качестве параметров ей передаётся задержка после отображения очередного такта обучения (в пределах 0,1 – 3 с) и шаг отображения (от 1 до 10000 тактов).

Визуализация разделяющей классы прямой, сформированной обученным персептраном, осуществляется выбором команды меню View>Last step.

При помощи команды View>History можно посмотреть график зависимости числа верно/неверно распознанных примеров от такта обучения. Переключение между графиками осуществляется щелчком левой кнопки мыши. Для скрытия графика необходимо ещё раз выбрать данную команду.

Численные значения количества верно/неверно распознанных примеров по тактам обучения, автоматически рассчитанное значение ширины разделяющего «коридора» между классами θ , а также обучающую выборку можно посмотреть при помощи команды View>Information.

Очистка рабочей области от изменений, внесённых при тестировании персептрона или просмотре шагов его обучения, осуществляется командой View>Clean up.

6. Тестирование персептрона. Программа предоставляет возможность просмотра реакции обученного персептрона на подаваемые ему тестовые образцы. Для этого пользователю необходимо выбрать команду Tools>Test главного меню. Предъявление тестового образца осуществляется щелчком левой кнопки мыши по рабочей области. В выбранном месте появляется новая точка, которая в зависимости от реакции персептрона, относится к первому или второму классу. Результат распознавания отображается на экране соответствующим цветом введенной точки.

2.3.3. Содержание лабораторных исследований

Лабораторные исследования по задаче 1 проводятся в следующем порядке.

1. Генерация данных. Студент вводит параметры и генерирует случайные точки на плоскости (X, Y) в соответствии с индивидуальным заданием (см. разд. 2.2).

2. Обучение элементарного персептрона. Введением разделяющей прямой в пространстве признаков (X, Y) и «коридора» между классами реализуется идентификация сгенерированных точек и гарантируется линейная разделимость классов. На этом завершается построение обучающей выборки.

Далее выполняется обучение элементарного персептрона для различных начальных значений вектора синаптических коэффициентов и смещения. Расширенный вектор синаптических коэффициентов $\vec{w}(0)$ при выполнении эксперимента следует варьировать в соответствии с программными ограничениями, представленными в табл. 2.3. Результаты записываются в отчет.

По результатам обучения анализируется, наблюдается ли зависимость финального вектора $\vec{w}(T)$ от начального значения $\vec{w}(0)$.

В отчете приводится иллюстрация, содержащая контуры сгенерированных кластеров данных, заданный разделяющий «коридор» между классами и несколько разделяющих прямых, наблюдавшихся в процессе обучения, включая финальную.

3. Исследование зависимости числа тактов обучения от ширины разделяющего «коридора» ε . Проводится анализ числа коррекций синаптических коэффициентов элементарного персептрана в процессе обучения от ширины разделяющего «коридора» ε между классами. Результат оформляется в отчете в виде графика.

4. Графики зависимости числа верно/неверно распознанных примеров от времени обучения. Анализируется процесс настройки синаптических коэффициентов элементарного персептрана с точки зрения числа верно/неверно распознанных примеров для случая, когда классы линейно разделимы ($\varepsilon > 0$). Соответствующие графики приводятся в отчете.

5. Применимость элементарного персептрана к решению задачи классификации в случае, когда классы не являются линейно разделимыми. Изучаются особенности процесса обучения элементарного персептрана в случае линейно неразделимых классов, когда решение задачи классификации на элементарном персептране не существует. В отчете приводится график зависимости числа верно распознанных примеров от времени обучения для рассматриваемого случая, а также последовательность разделяющих прямых с некоторой скважностью ΔT по числу T коррекций синаптических коэффициентов (значение ΔT следует подобрать экспериментально).

2.4. Задача классификации для нескольких попарно линейно разделимых классов (задача 2)

2.4.1. Общее содержание задачи и программной модели

Сформулируем постановку решаемой задачи.

В пространстве признаков размерности M заданы n классов. Каждый класс представлен областью Ω_i , $i = \overline{1, n}$. Предполагается, что рассматриваемые области попарно линейно разделимы в пространстве признаков. Более того имеется в виду выполнение «усиленного» свойства разделимости классов, необходимого для конечности времени обучения нейросетевого классификатора. Известна обучающая выборка, содержащая некоторое количество примеров объектов каждого из n классов. Требуется построить нейронную

сеть, которая произвольный объект, характеризующийся вектором признаков размерности M , отнесет к одному из n заданных классов.

В лабораторной работе задача решается для случая, когда каждый объект характеризуется двумя признаками ($M = 2$) и потому может быть представлен точкой на плоскости. Каждый класс представлен в лабораторной работе одним кластером точек, которые генерируются по тем правилам, которые описаны в разд. 2.3.1. Параметры N_{cl} , d_x и d_y для генерации данных указаны в разд. 2.2. На рис. 2.4 приведен пример расположения четырех классов Ω_i , $i = \overline{1, 4}$, которые попарно линейно разделимы и имеют гарантированные «коридоры» для построения разделяющих прямых.

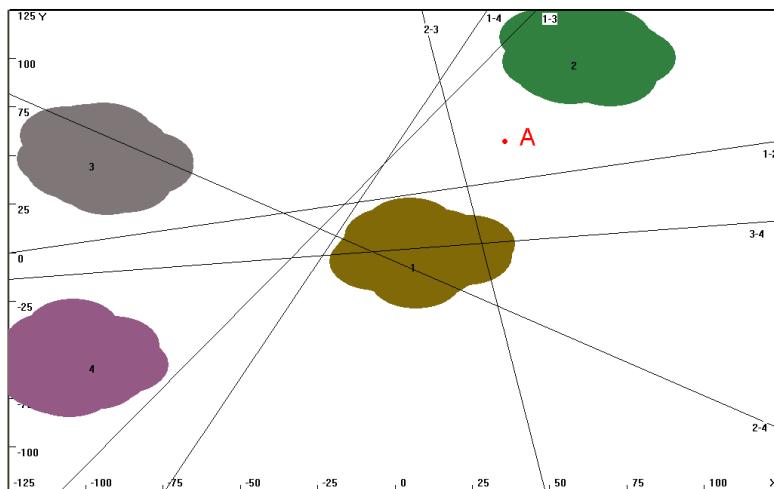


Рис. 2.4. Пример четырех попарно линейно разделимых классов в плоскости (X, Y)

Для решения поставленной задачи в лабораторной работе строится двухслойная нейронная сеть на элементарных персептранах с биполярной активационной характеристикой. Первый слой содержит $\frac{n(n - 1)}{2}$ нейронов по числу разделяющих прямых между всеми парами классов. Второй слой, содержащий n нейронов, выполняет логическую функцию. Выходной нейрон с номером i принимает значение 1 только тогда, когда входной вектор признаков

принадлежит классу i (области Ω_i). В противном случае на рассматриваемом выходе наблюдается значение -1 . Таким образом, выходной вектор размерности n всегда имеет единственную равную 1 координату по тому выходу, который соответствует классу принадлежности входного вектора признаков.

Нейроны первого слоя обучаются последовательно и независимо друг от друга, создавая разделяющие прямые между парами классов. Для обучения нейронов первого слоя используется процедура Уидроу – Хоффа, изученная при проведении исследований по задаче 1. На рис. 2.4 показан пример разделяющих прямых, получившихся после завершения процесса обучения нейронов первого слоя (число прямых – 6). Этот результат определяет области Ω_i , $i = \overline{1, 4}$, которые формируются двухслойным нейросетевым классификатором. При выполнении лабораторной работы эти области закрашиваются соответствующим цветом.

Лабораторные исследования по задаче 2 выполняются в программе TwinClasses.

2.4.2. Руководство пользователя программы TwinClasses

Программа TwinClasses реализует модель двухслойного нейросетевого классификатора, предназначенного для решения задачи классификации объектов для n ($n \geq 3$) попарно линейно разделимых классов.

Описание возможностей программы приводится в следующей последовательности.

1. Генерация данных. Программа позволяет произвести генерацию данных в виде совокупности кластеров точек на плоскости (X, Y) для задачи классификации образов на n ($n \geq 3$) попарно линейно разделимых классов. Программа реализует процедуру, которая относит к кластеру не только сгенерированные точки, но и их круговые окрестности радиусом R .

Кластеры случайным образом, не накладываясь друг на друга, располагаются в рабочей области и включают в себя по 100 точек.

Пользователю предоставляется возможность задавать границы допустимой области ($X_{\min}, X_{\max}, Y_{\min}, Y_{\max}$), количество кластеров в ней (N_{cl}), величину окрестности точек (R), а также значения дисперсий внутри кластеров по осям X и Y (d_x и d_y соответственно).

Для изменения всех вышеперечисленных параметров следует выбрать команду Tools>Properties в главном меню программы.

Допустимые значения варьируемых параметров представлены в табл. 2.5.

Таблица 2.5

Пределы изменения параметров в программе TwinClasses

Параметр	X_{\min}	X_{\max}	Y_{\min}	Y_{\max}	N_{cl}	d_x	d_y	R
Минимальное значение	-200	50	-200	50	3	10	10	5
Максимальное значение	-50	200	-50	200	10	30	30	15

Генерация данных производится автоматически при каждом запуске программы, со значениями параметров, заданными по умолчанию, а также при каждом изменении последних. В остальных случаях генерация выполняется выбором команды меню Tools>Generate new classes.

Создание новой рабочей области производится с помощью команды File>New.

2. Идентификация данных. Число классов совпадает с заданным числом кластеров. Каждый кластер – отдельный класс. Все классы помечаются собственными цветами и номером.

3. Просмотр информации о строении сети и корректировка значений ее синаптических коэффициентов. Пользователю предоставляется возможность коррекции значений синаптических коэффициентов сети. Изменение синаптических коэффициентов до начала процесса обучения соответствует заданию начальных условий.

Вышеописанные действия осуществляются при выборе команды меню View>Network structure. Вид окна редактирования приведен на рис. 2.5.

Например, для изменения значения коэффициента $w_{10,1}$ (коэффициент усиления сигнала от 1-го входа к 10-му нейрону первого слоя) необходимо выбрать номер нейрона из соответствующего списка вверху окна, а затем синаптический коэффициент из списка

под нейроном. В окне справа от индекса коэффициента отображается его значение, доступное для изменения (см. рис. 2.5).

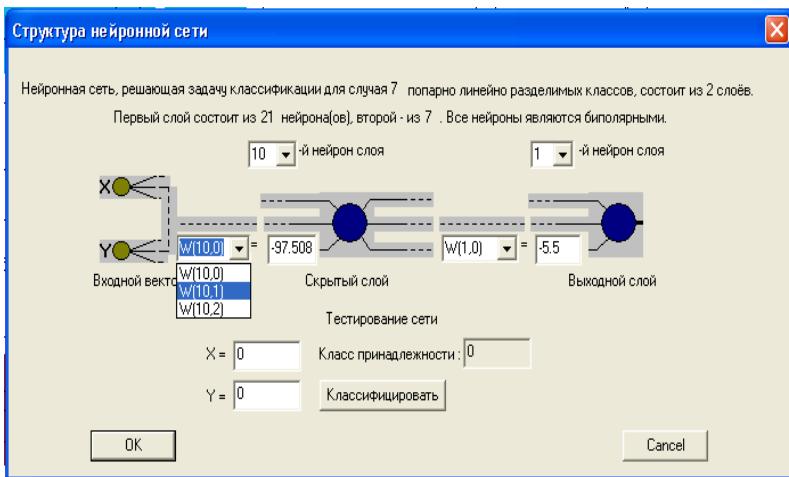


Рис. 2.5. Информационное окно при просмотре и редактировании синаптических коэффициентов нейронов первого слоя

4. Обучение сети. Обучение нейронной сети решению задачи классификации основано на методике обучения персептрона Розенблатта. По ней настраиваются синаптические коэффициенты нейронов первого слоя сети. Второй слой не обучается – синаптические коэффициенты устанавливаются по определенному логическому правилу.

Обучение осуществляется выбором команды меню Tools>Teach.

5. Тестирование нейронной сети. Программа предоставляет возможность просмотра реакции обученной нейросети. Для этого пользователю необходимо выбрать команду Tools>Test главного меню. В результате тестирования каждая точка рабочей области помечается цветом согласно принадлежности её тому или иному классу по реакции сети. Необходимо отметить, что цвет исходного кластера данных не совпадает с цветом точек области, также относимых к нему в результате тестирования. Такая реализация объяс-

няется требованием, чтобы кластер был заметен на фоне области, отнесенной сетью к соответствующему классу.

Для того, чтобы скрыть/просмотреть разделяющие прямые между классами, сформированные нейронной сетью, необходимо снять/поставить метку напротив пункта меню View>Separation lines.

Тестирование сети осуществляется в информационном окне, отображающемся при выборе команды меню View>Network structure. В этом случае необходимо задать координаты классифицируемого образа и нажать кнопку «Классифицировать». Этой опцией можно воспользоваться только после обучения сети. В противном случае режим тестирования недоступен.

2.4.3. Содержание лабораторных исследований

Лабораторные исследования по задаче 2 проводятся в следующем порядке.

1. Генерация данных. Студент вводит параметры и генерирует данные, принадлежащие n классам.

2. Обучение нейросетевого классификатора. Студент запускает процедуру обучения нейронной сети. В отчете записываются сведения по структуре и способу обучения двухслойного классификатора.

3. Анализ разделяющих прямых между классами и результата классификации. В отчете на рисунках показываются результаты классификации для двух вариантов генерации данных с параметрами, указанными в задании. Студент выделяет на рисунке построенные классификатором границы классов и визуально контролирует правильность решения задачи классификации.

4. Тестирование нейросетевого классификатора. Ставится эксперимент, который позволяет оценить чувствительность результата классификации произвольного входного вектора к изменению синаптических коэффициентов нейронов первого слоя.

2.5. Задача классификации при наличии двух линейно неразделимых классов (задача 3)

2.5.1. Общее содержание задачи и программной модели

Сформулируем постановку решаемой задачи.

В пространстве признаков размерности M заданы два линейно неразделимых класса. Известна обучающая выборка, содержащая некоторое количество примеров объектов первого и второго классов. Требуется построить нейронную сеть на элементарных персептронах с bipolarной активационной характеристикой, которая произвольный объект, характеризующийся вектором признаков размерности M , отнесет к одному из двух заданных классов.

В лабораторной работе каждый из двух классов генерируется как совокупность нескольких кластеров точек. Программно обеспечивается непересечение кластеров. Из-за того, что центры кластеров случайно расположены в рабочем поле, в подавляющем большинстве случаев сгенерированные кластеры оказываются линейно неразделимыми.

Для построения нейросетевого классификатора в лабораторной работе применяется метод Мезарда и Надала («черепичный» алгоритм). Его применение приводит к многослойной нейронной сети, в которой нейроны обучаются последовательно по мере роста конструируемой нейронной сети. В лабораторной работе предоставляется возможность анализа результата обучения первого слоя. Построенные первым слоем разделяющие прямые позволяют выделить области в пространстве признаков, которые относятся к каждому из двух классов.

Лабораторное исследование по задаче 3 выполняется с помощью программы Tile.

2.5.2. Руководство пользователя программы Tile

Программа Tile реализует модель многослойного нейросетевого классификатора, предназначенную для решения задачи классификации объектов в случае двух линейно неразделимых классов.

Далее приводится описание функциональных возможностей программы.

1. Генерация данных. Программа позволяет произвести генерацию данных в виде совокупности кластеров точек на плоскости (X , Y). Как и в программе TwinClasses, помимо самой точки, кластеру также принадлежит ее круговая окрестность радиусом R .

Кластеры случайным образом, не накладываясь друг на друга, располагаются в рабочей области и включают в себя по 100 точек.

Пользователю предоставляется возможность задавать границы допустимой области (X_{\min} , X_{\max} , Y_{\min} , Y_{\max}), количество кластеров в ней (N_{cl}), величину окрестности точек (R), а также значения дисперсий внутри кластеров по осям X и Y (d_x и d_y соответственно).

Для изменения всех вышеперечисленных параметров следует выбрать команду Tools>Properties в главном меню программы.

Допустимые значения варьируемых параметров представлены в табл. 2.6.

Таблица 2.6

Пределы изменения параметров в программе Tile

Параметр	X_{\min}	X_{\max}	Y_{\min}	Y_{\max}	N_{cl}	d_x	d_y	R
Минимальное значение	-200	100	-200	100	3	10	10	5
Максимальное значение	-100	200	-100	200	7	30	30	15

Генерация данных производится автоматически при каждом запуске программы со значениями параметров, заданными по умолчанию, а также при каждом изменении последних. В остальных случаях генерация выполняется выбором команды меню Tools>Generate new classes.

Создание новой рабочей области производится с помощью команды File>New.

2. Идентификация данных. Каждый из двух классов состоит из одного или более кластеров. Точки, принадлежащие первому классу, помечаются красным цветом, ко второму – синим (рис. 2.6).

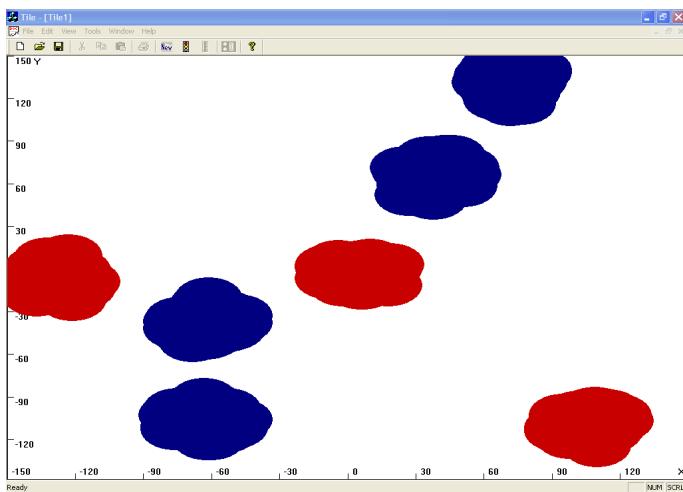


Рис. 2.6. Пример двух сгенерированных классов данных

3. Обучение сети. Конструирование нейронной сети, решающей задачу распознавания объектов, осуществляется по «черепичному» алгоритму, предложенному Мезардом и Надалом. Хотя сам этот метод предусматривает построение многослойной нейронной сети, в программе строится сеть, состоящая только из одного слоя нейронов. Это связано с тем, что всю необходимую информацию можно получить, используя выходные значения нейронов только первого слоя. Поскольку сам «черепичный» алгоритм не является оптимальным, такой подход к неполному построению сети позволяет существенно упростить программную реализацию классификатора.

Обучение каждого нейрона осуществляется по правилу Уидроу – Хоффа. Необходимо отметить, что поскольку этот алгоритм предназначен для обучения при линейно разделимых классах, в рассматриваемом случае он может не сходиться. Это проявляется в возможном превышении числа нейронов первого слоя по сравнению с другими нейросетевыми методами.

Конструирование и обучение нейронной сети осуществляется выбором команды меню Tools>Teach.

4. Просмотр информации о строении сети и корректировка значений синаптических коэффициентов нейронов. Пользователю предоставляется возможность коррекции значений синаптиче-

ских коэффициентов сети, осуществляющей при выборе команды меню View>Network structure. Вид окна редактирования приведен на рис. 2.7.

Например, для изменения значения синаптического коэффициента w_1 четвёртого нейрона первого слоя необходимо выбрать номер нейрона из соответствующего списка вверху окна. В соответствующем окне справа отображается значение коэффициента w_1 , доступное для изменения (см. рис. 2.7).

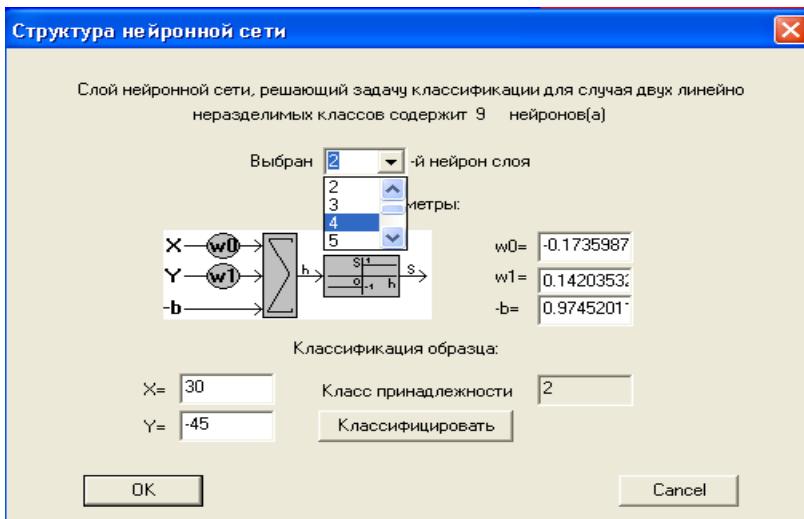


Рис. 2.7. Информационное окно при контроле и коррекции синаптических коэффициентов нейронов первого слоя

5. Тестирование нейронной сети. Программа предоставляет возможность просмотра реакции обученной нейросети. Для этого пользователю необходимо выбрать команду Tools>Test главного меню. В результате тестирования каждая точка рабочей области помечается цветом класса, к которому она относится согласно реакции нейронной сети. На рис. 2.8 исходные образцы, используемые в качестве обучающей выборки, отображаются более темным цветом.

Для того чтобы скрыть/просмотреть разделяющие прямые между классами, сформированные нейронной сетью, необходимо

снять/поставить метку напротив пункта меню View>Separation lines.

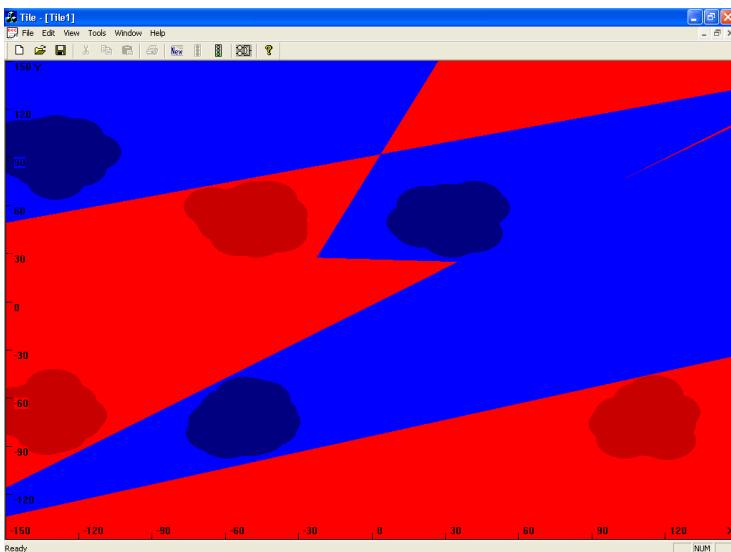


Рис. 2.8. Пример решения задачи классификации программой Tile

Тестирование сети возможно также в информационном окне, отображающемся при выборе команды меню View>Network structure. В этом случае необходимо задать координаты классифицируемого образца и нажать кнопку «Классифицировать».

2.5.3. Содержание лабораторных исследований

Лабораторные исследования по задаче 3 проводятся в следующем порядке.

1. Генерация данных. Студент вводит параметры и генерирует данные, принадлежащие двум линейно неразделимым классам.

2. Обучение нейросетевого классификатора. В отчет записываются краткие сведения о применяемом подходе к построению многослойного классификатора и методе обучения нейронов, после чего запускается процедура обучения сети.

3. Анализ результата классификации. Для двух экспериментов с разным числом кластеров в отчете строятся разделяющие прямые и выделяются горизонтальной и вертикальной штриховкой построенные нейронной сетью классы. Визуально контролируется правильность решения задачи классификации.

4. Тестирование нейросетевого классификатора. Ставится эксперимент, который позволяет оценить чувствительность результата классификации произвольного входного вектора к изменению синаптических коэффициентов нейронов первого слоя.

2.6. Рекомендации к оформлению отчета

В отчете фиксируются исходные данные и основные результаты лабораторных исследований, а также необходимые сведения по архитектуре и способам обучения используемых нейронных сетей. Рекомендуется составить отчет в следующей форме.

1. Задача классификации для случая двух линейно разделимых классов (задача 1)

Генерация данных. Приводится номер варианта индивидуального задания. В таблице указываются значения параметров для генерации данных: границы области по осям X и Y , число кластеров, количество точек в кластерах, внутрикластерные дисперсии d_x , d_y по координатным осям X и Y соответственно.

Обучение элементарного персептрона. Указываются параметры a , b и c разделяющей прямой $ax + by + c = 0$. Даётся пояснение к свойству усиленной разделимости классов и содержанию параметра ε . По результатам исследования заполняется таблица, в которой для пяти разных экспериментов указываются начальные и финальные значения синаптических коэффициентов, а также число тактов обучения персептрона и число выполненных коррекций синаптических коэффициентов. Делается вывод относительно зависимости финального вектора синаптических коэффициентов от их начальных значений.

На рисунке приводится рабочая область и прореженная по времени последовательность разделяющих прямых, сформированных персептроном в процессе обучения, для заданного начального состояния синаптических коэффициентов.

Исследование зависимости числа тактов обучения от ширины разделяющего коридора ε . В таблице фиксируются результаты экспериментов, в которых варьируется параметр ε . Для пяти значений параметра ε указывается число тактов обучения персептрона. Строится график $T(\varepsilon)$.

По результатам исследования делается вывод о соответствии/несоответствии экспериментальных данных теореме Розенблатта.

Зависимость числа верно/неверно распознанных примеров от такта обучения. Приводятся полученные в эксперименте графики зависимости числа верно/неверно распознанных примеров от такта обучения. Анализируется характер зависимости.

Применимость персептрона Розенблатта к решению задачи классификации данных в случае, если классы не являются линейно разделимыми. Приводится результат соответствующего эксперимента в форме графиков зависимости числа верно/неверно распознанных примеров от номера такта обучения. Делается вывод относительно соответствия/несоответствия экспериментальных данных теореме Розенблатта.

2. Задача классификации для случая несольких попарно линейно разделимых классов (задача 2)

Генерация данных. Приводятся в форме таблицы исходные данные индивидуального задания для проведения лабораторного исследования по задаче 2. В таблице указываются границы области по координатным осям, число классов для двух вариантов, дисперсии координат точек класса по координатным осям.

Архитектура нейросетевого классификатора. Даются краткие сведения по архитектуре классификатора, типу нейронов и способу обучения.

Разделяющие прямые между классами, сформированные нейронной сетью (для двух случаев с разным количеством классов). На рисунке изображается расположение классов и разделяющих прямых, сформированных в процессе обучения классификатора. Штриховкой выделяются области, соответствующие разным классам.

Тестирование нейросетевого классификатора. Приводится несколько примеров тестирования обученного классификатора. Те же

примеры рассматриваются для случая измененных значений синаптических коэффициентов нейронов.

3. Задача классификации для случая двух линейно неразделимых классов (задача 3)

Генерация данных. Приводятся в форме таблицы исходные данные индивидуального задания для проведения лабораторного исследования по задаче 3. В таблице указываются границы области по координатным осям, число кластеров в классе для двух вариантов, дисперсии координат точек кластеров по координатным осям.

Архитектура нейросетевого классификатора. Кратко излагается принцип формирования архитектуры классификатора по методу Мезарда и Надала. Указываются свойства нейронов и правило их обучения.

Разделяющие прямые между классами, сформированные нейронной сетью (для двух случаев с разным числом кластеров в классах). На рисунке изображается расположение классов и разделяющих прямых, сформированных в процессе обучения классификатора. Штриховкой выделяются области, соответствующие разным классам.

Тестирование нейросетевого классификатора. Приводится несколько примеров тестирования обученного классификатора. Те же примеры рассматриваются для случая измененных значений синаптических коэффициентов нейронов.

Контрольные вопросы

1. В чем состоит задача распознавания образов?
2. Каким образом описывается каждый объект, подлежащий классификации?
3. В чем состоит условие *простой разделимости классов* при решении задачи классификации данных?
4. Почему при решении практических задач требуется выполнение *усиленного условия разделимости классов*?
5. Как на единственном нейроне реализуется *линейная граница* между двумя классами? Дайте графическую иллюстрацию.
6. Что называется элементарным персептроном?

7. Сформулируйте математическую модель элементарного персептрона с бинарной (биполярной) активационной характеристикой.
8. Запишите модель функционирования элементарного персептрона в расширенном пространстве признаков. Объясните состав координат расширенного вектора признаков и расширенного вектора синаптических коэффициентов.
9. Какова структура обучающей выборки для настройки параметров элементарного персептрона?
10. Сформулируйте правило Хебба обучения элементарного персептрона с бинарной активационной характеристикой.
11. В чем состоит негативная особенность правила Хебба обучения элементарного персептрона?
12. Сформулируйте правило Уидроу – Хоффа обучения элементарного персептрона. Какая активационная характеристика нейрона предполагается в правиле Уидроу – Хоффа?
13. Что означает параметр δ в аддитивном δ -правиле Уидроу – Хоффа?
14. Что означает параметр ϵ при выполнении лабораторных исследований?
15. Сформулируйте используемую в теореме Розенблatta процедуру обучения элементарного персептрона по примерам обучающей выборки.
16. Сформулируйте *простое условие разделимости* классов в теореме Розенблatta.
17. Сформулируйте *усиленное условие разделимости* классов в теореме Розенблatta.
18. Какой смысл имеет параметр θ в формулировке усиленного условия разделимости классов в теореме Розенблatta? Какое можно дать этому параметру геометрическое толкование? Приведите графическую иллюстрацию к ответу.
19. Что понимается под числом шагов обучения персептрона в теореме Розенблatta?
20. Какой факт устанавливает теорема Розенблatta относительно сходимости процедуры обучения элементарного персептрона?
21. Какова архитектура нейронной сети, используемой для распознавания нескольких *попарно линейно разделимых классов*?
22. Допустим, вектор признаков распознаваемых объектов имеет размерность M , а число *попарно линейно разделимых классов*

равно N . Сколько нейронов содержит *первый* слой нейронной сети, используемой для решения задачи распознавания? Каковы их активационные характеристики?

23. Какой метод может быть использован для настройки синаптических коэффициентов нейронов первого слоя в нейронной сети, применяемой для классификации данных в случае *попарно линейно разделимых классов*?

24. Какой принцип используется при построении второго слоя нейронной сети, применяемой для классификации данных в случае *попарно линейно разделимых классов*?

25. Допустим, вектор признаков распознаваемых объектов имеет размерность M , а число *попарно разделимых классов* равно N . Сколько нейронов содержит *второй* слой нейронной сети, используемой для решения задачи распознавания? Каковы их активационные характеристики?

26. В чем состоит принцип решения задачи классификации для двух классов, если они *не являются линейно разделимыми*? Приведите графическую иллюстрацию.

27. В чем состоит процедура Мезарда и Надала построения многослойного нейросетевого классификатора для двух *линейно неразделимых классов*?

28. Почему «черепичный» алгоритм Мезарда и Надала называют *конструктивным* алгоритмом построения нейронной сети для решения задачи распознавания?

29. Чем определяется число *нейронов первого слоя* при построении многослойного нейросетевого классификатора методом Мезарда и Надала? Является ли это число определенным заранее по исходным данным задачи классификации?

30. Чем определяется число *слоев* многослойного нейросетевого классификатора, построенного методом Мезарда и Надала? Является ли это число определенным заранее по исходным данным задачи классификации?

31. Сколько нейронов должен содержать последний слой многослойного нейросетевого классификатора, построенного методом Мезарда и Надала?

32. Чем обеспечивается сходимость процедуры Мезарда и Надала синтеза нейросетевого классификатора?

Лабораторная работа 3

МНОГОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ. МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

3.1. Цель и задачи лабораторной работы

Цель: ознакомление с архитектурой многослойных нейронных сетей (МНС) с непрерывной активационной характеристикой нейронов, математической моделью МНС и постановкой задачи ее обучения; изучение метода обратного распространения ошибки для настройки параметров сети в процессе обучения; исследование способностей МНС по аппроксимации непрерывных функций; анализ точности нейросетевой аппроксимации.

Широкое использование многослойных сетей (многослойных персепtronов с непрерывными активационными характеристиками) в практических приложениях объясняется не только возможностью реализации на них произвольных функциональных преобразований векторного входного сигнала, но и разработкой эффективной вычислительной процедуры обучения на основе метода обратного распространения ошибки.

Исследовательская работа проводится на данных, которые генерируются студентом в соответствии с индивидуальным заданием. В лабораторной работе студент осваивает следующие технологические процедуры, используемые для решения задачи аппроксимации функций многих переменных на многослойном персептроне: выбор архитектуры МНС, предварительная и постобработка данных, формирование обучающей и тестовой выборок, инициализация синаптических коэффициентов перед их обучением, обучение нейронной сети с контролем условия завершения этой процедуры, проверка способности сети к обобщению на тест-примерах, контрастирование сети.

Ряд специальных исследований позволяет изучить особенности организации процесса обучения (режимы «по примерам» и «по эпохам»), условия возникновения «паралича» нейронов, эффект возможного «переобучения» сети, влияние помех измерения входных данных на точность обучения сети.

Исследования, выполненные в данной лабораторной работе на учебном примере, помогут студенту решить с применением многослойного персептрана содержательную задачу, предусмотренную в лабораторной работе 4.

3.2. Варианты индивидуальных заданий

Рассмотрим точку в плоскости, которая может характеризоваться как декартовыми координатами x, y , так и полярными – ρ, φ , $\varphi \in [-\pi, \pi]$. В зависимости от устройства измерительного прибора или метода измерения могут регистрироваться различные характеристики текущего координатного положения объекта: $x \sin \varphi$, $x^2 + y^2 = \rho^2$, $|\varphi|$, $x + y$, $x - y$, $\cos \varphi$, $\sin \varphi$, $\operatorname{tg} \varphi$, $x \cos \varphi$. Задание для исследовательской работы содержит доступные для измерения характеристики координатного положения объекта (входные сигналы многослойного персептрана), желаемые характеристики на выходе персептрана и описание границы области в плоскости, в которой должно быть реализовано требуемое функциональное преобразование.

Таблица 3.1

Варианты к лабораторной работе

Номер варианта	Входные переменные	Желаемые выходы МНС	Область моделирования функции	Объем выборки	Начальная архитектура МНС
1	x, y	ρ, φ	$x \in [-5, 5], y \in [0, 5]$	150	3:4:2
2	x, ρ	y, φ	$x \in [-5, 5], y \in [0, 5]$	200	4:3:2
3	ρ, φ	x, y	$x \in [-5, 5], y \in [0, 5]$	250	4:3:2
4	$x, y \cos \varphi, x \sin \varphi$	ρ, φ	$x \in [-10, 10], y \in [-5, 0]$	250	5:4:2
5	$x + y, x - y$	ρ^2, φ	$x \in [-10, 10], y \in [-5, 0]$	200	5:3:2

3.3. Руководство пользователя

Нейроэмулатор, предназначенный для работы с многослойными сетями, предоставляет функциональные возможности по предварительной и постобработке данных, используемых при построении нейросетевой модели, реализует метод обратного распространения ошибки для обучения МНС, некоторые методы оптимизации критерия обучения МНС, а также предоставляет интерфейс для визуализации исходных данных и анализа результатов обучения.

3.3.1. Общее описание программы

Главное окно программы разделено на две панели: «Выборки» и «Нейросети». Панель «Выборки» предназначена для генерации, просмотра и предобработки данных, используемых для построения нейросетевой модели. В этой панели производится подготовка выборок для обучения, указание входных и желаемых выходных данных, разбиение выборок на обучающую, валидационную и тестовую. Панель «Нейросети» содержит опции, связанные с построением нейросетевой модели. В этой панели проводится создание многослойных персептронов, выбор архитектуры сети, обучение, валидация и тестирование сети.

Панель «Выборки» содержит список созданных пользователем выборок данных, а панель «Нейросети» – список созданных пользователем нейросетей. Для работы с конкретной выборкой или с конкретной нейросетью необходимо выделить их имена на соответствующих панелях и использовать предоставляемые в этих панелях опции.

3.3.2. Создание выборок данных

Создание выборки данных в системе возможно двумя способами. Первый способ предполагает загрузку данных из файла формата .txt. Каждый пример выборки представляет собой строку файла. В каждой строке файла должно быть одинаковое количество чисел, разделенных пробелами или знаками табуляции. При необходимости в загружаемом файле могут быть указаны названия столбцов данных. Эти названия перечисляются в начале файла, каждое название на новой строке. После перечисления названий в файле че-

рез пробельную строку указывается выборка данных. Для загрузки выборки из файла используется опция «Открыть файл...» на панели «Выборки».

Второй способ создания выборки предполагает автоматическую генерацию данных, используемых в лабораторной работе в соответствии с вариантом задания. Для автоматической генерации выборки используется опция «Данные для ЛР...» на панели «Выборки». В появившемся окне «Генерация данных» студент указывает имя выборки, общее число полей и число примеров в выборке, для каждого поля помечается тип – входное или желаемое выходное. Для выходных полей необходимо выбрать формулу расчета (например, $x + y$ или $y \cos \phi$). После создания выборки в списке на панели «Выборки» появится имя созданной выборки.

При построении нейросетевой модели исходная выборка данных обычно разбивается на три выборки: обучающую, валидационную и тестовую в заданном соотношении. Для разбиения выборки данных необходимо выделить ее имя в списке созданных выборок и использовать опцию «Разбить...» на панели «Выборки». В появившемся окне «Разбиение выборки» следует указать имена создаваемых выборок и процентное соотношение примеров в обучающей, валидационной и тестовой выборках. После разбиения выборки в списке выборок на панели «Выборки» появятся имена созданных выборок.

3.3.3. Предобработка данных

Для предобработки данных выборки используется опция «Предобработка...» на панели «Выборки». Перед нажатием на кнопку «Предобработка...» необходимо выделить в списке созданных выборок имя выборки, примеры которой будут подвергнуты предобработке. В появившемся окне «Предобработка данных» указать метод предобработки и его параметры.

В программе реализованы следующие методы предобработки:

- масштабирование;
- линейное преобразование.

Масштабирование данных выборки предполагает отображение ее поля данных с учетом реальных минимального и максимального значений на заданный интервал, например $[-1; 1]$. Это отображение

осуществляется для каждого поля данных выборки, независимо от других полей.

3.3.4. Создание МНС

Создание сети осуществляется с помощью опции «Создать...» на панели «Нейросети». При нажатии на кнопку «Создать...» появляется окно «Создание сети», в котором необходимо указать имя и архитектуру создаваемой сети. Архитектура сети включает число входов и число слоев сети, распределение нейронов по слоям и тип активационной характеристики нейронов слоя. После создания сети в списке на панели «Нейросети» появится имя созданной сети.

Для просмотра синаптических коэффициентов сети необходимо в списке созданных сетей на панели «Нейросети» в главном окне программы выбрать имя многослойного персептрона и нажать кнопку «Синапсы...». В появившемся окне «Синаптические коэффициенты» в таблице приводятся матрицы синаптических коэффициентов сети. При создании сети синаптические коэффициенты инициализируются случайным образом в соответствии с законом распределения и параметрами, установленными по умолчанию. Для повторной инициализации синаптических коэффициентов необходимо в таблице выделить мышью инициализируемые синаптические коэффициенты, после чего нажать на кнопку «Инициализировать...». В появившемся окне «Инициализация» указать закон распределения синаптических коэффициентов и его параметры.

3.3.5. Обучение МНС

Для обучения МНС используется опция «Обучить...» на панели «Нейросети». Перед нажатием на кнопку «Обучить...» необходимо выделить в списке созданных нейросетей имя обучаемой сети. В появившемся окне «Обучение сети» необходимо указать выборки, используемые для обучения, метод оптимизации синаптических коэффициентов сети и его параметры, критерий останова процедуры обучения.

В программе для обучения сети реализованы следующие градиентные методы:

- простой градиентный метод;
- градиентный метод с моментом;

- метод Левенберга – Маркардта;
- градиентный метод с адаптивной настройкой шага delta-bar-delta.

Расчет градиента проводится по методу обратного распространения ошибки.

Запуск процедуры обучения сети осуществляется при нажатии на кнопку «Обучить» после указания имен выборок и метода обучения. В окне «Процесс обучения» через заданное число эпох происходит обновление информации о значениях среднеквадратичной ошибки и максимального модуля ошибки на обучающей и валидационной выборках.

3.3.6. Зашумление выборок данных и параметров МНС

Для зашумления данных используется опция «Зашумить...» на панели «Выборки». Перед нажатием на кнопку «Зашумить...» необходимо выделить в списке созданных выборок имя выборки, данные которой будут подвергнуты зашумлению. В появившемся окне «Зашумление» указать уровень шума (в процентах).

Для зашумления синаптических коэффициентов сети используется опция «Зашумить...» на панели «Нейросети». Перед нажатием на кнопку «Зашумить...» необходимо выделить в списке созданных нейросетей имя сети, синаптические коэффициенты которой будут подвергнуты зашумлению. В появившемся окне «Зашумление» указать уровень шума (в процентах).

3.3.7. Визуализация и анализ результатов обучения МНС

Для визуализации данных и анализа результатов обучения МНС используется опция «Анализ...». При нажатии на кнопку «Анализ...» появляется окно «Визуализация и анализ результатов», с помощью которого могут быть рассчитаны следующие показатели и построены следующие таблицы и графики:

- диаграмма рассеяния данных обучающей/валидационной выборки;
- значения среднеквадратичной ошибки на обучающей/валидационной/тестовой выборках до и после процедуры обучения;

- таблица значений и график динамики критерия на обучающей/валидационной выборке в процессе обучения сети;
- таблица значений и гистограмма средних абсолютных ошибок аппроксимации выходной переменной обученной сети на обучающей/валидационной выборке;
- таблица показателей вариабельности нейронов сети на обучающей/валидационной выборке.

Для построения этих таблиц и графиков необходимо использовать соответствующую опцию в окне «Визуализация и анализ результатов».

3.4. Содержание лабораторных исследований

В лабораторной работе исследуются важные аспекты обучения и функционирования многослойного персептрона с непрерывными активационными характеристиками:

- исследование процедуры обучения многослойного персептрона;
- анализ информационной активности нейронов и контрастирование сети;
- экспериментальный поиск архитектуры многослойного персептрона с улучшенными свойствами по точности;
- сравнительный анализ различных вычислительных процедур обучения МНС;
- исследование влияния на сходимость процесса обучения параметра скорости обучения ε ;
- влияние помех измерения входных/выходных данных обучающей выборки на точность аппроксимации функции с помощью МНС;
- влияние ошибок реализации синаптических коэффициентов и смещений нейронов МНС на точность аппроксимации функции.

3.4.1. Исследование процесса настройки параметров многослойного персептрона (эксперимент 1)

Цель эксперимента – обучение многослойного персептрона заданной архитектуры методом обратного распространения ошибки.

Перед началом процесса обучения студент создает выборку данных в соответствии с вариантом задания, осуществляет предобработку данных выборки и делит ее на три части: обучающую, валидационную и тестовую в выбранном соотношении. После создания этих выборок студент создает многослойный персептрон, для обучения, валидации и тестирования которого использует генерированные выборки.

Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Создание выборки данных (опция «Данные для ЛР...», см. разд. 3.3.2).
2. Предобработка данных (опция «Предобработка...», см. разд. 3.3.3).
3. Разбиение выборки на обучающую, тестовую и валидационную (опция «Разбить...», см. разд. 3.3.3).
4. Создание многослойного персептрана (опция «Создать...», см. разд. 3.3.4).
5. Обучение, валидация и тестирование многослойного персептрана (опция «Обучить...», см. разд. 3.3.5).
6. Анализ результатов обучения (опция «Анализ...», см. разд. 3.3.7).

Объем создаваемой выборки данных указан в варианте задания к лабораторной работе. Объемы обучающей, валидационной и тестовой выборок рекомендуется выбирать в соотношении 60, 30 и 10 % от общего объема данных соответственно. При выполнении предобработки исходных данных лучше использовать масштабирование на интервал $[-0,9; 0,9]$.

При создании многослойного персептрана в последнем слое рекомендуется выбирать линейную активационную характеристику, а в скрытых – гауссиану или сигмоиду.

В качестве метода оптимизации параметров многослойного персептрана выбрать простой градиентный метод с параметром скорости обучения 0,6 – 0,8. При медленной сходимости процедуры обучения увеличить это значение, при возникновении колебательного характера процедуры обучения – уменьшить.

3.4.2. Анализ информационной активности нейронов и контрастирование сети (эксперимент 2)

Цель эксперимента – исследование возможности упрощения архитектуры многослойного персептрона, построенного в разд. 3.4.1. Анализ проводится на основе значений показателей, характеризующих влияние каждого нейрона на формирование реакции сети: минимального и максимального значений выхода нейрона на обучающей/валидационной выборке, а также его среднего значения и значения стандартного отклонения. Для сравнения вариабельности выходов нейронов сети они упорядочиваются по значению стандартного отклонения выхода на обучающей/валидационной выборке, что позволяет построить график зависимости стандартного отклонения от номера нейрона в вариационном ряду.

Для данного исследования используется сеть, обученная в разд. 3.4.1. Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Используя опцию «Анализ...» программы, вывести на экран таблицу показателей, характеризующих вариабельность нейронов. Из окна «Анализ вариабельности нейронов» переписать в отчет всю необходимую информацию и построить по имеющимся данным необходимые графики.

2. Исключить из сети нейроны, оказывающие малое влияние на выход сети на примерах обучающей/валидационной выборки, используя опцию «Исключить» в окне «Анализ вариабельности нейронов».

3. Дообучить полученную сеть, используя опцию «Обучить...» в панели «Нейросети» главного окна программы.

4. Сравнить значения среднеквадратической ошибки на примерах обучающей/валидационной выборки для обученной сети и сети, обученной в разд. 3.4.1. Сделать выводы.

3.4.3. Экспериментальный поиск архитектуры многослойного персептрона с улучшенными свойствами по точности (эксперимент 3)

Цель эксперимента – исследование и поиск архитектуры нейронной сети, показывающей наилучшие значения по качеству ап-

проксимации данных с точки зрения достигнутой точности на обучающей, валидационной и тестовой выборках.

Порядок проведения данного исследования аналогичен изложенному в разд. 3.4.1. Перед проведением исследования создать несколько МНС различных архитектур и последовательно их обучить.

3.4.4. Сравнительный анализ различных вычислительных процедур обучения МНС (эксперимент 4)

Цель эксперимента – сравнение по скорости сходимости и особенностям динамики следующих вычислительных процедур обучения МНС: простой градиентный метод, метод Левенберга – Маркардта и градиентный метод с адаптивной настройкой шага delta-bar-delta. При этом каждый раз для обучения различными методами используется одна и та же нейронная сеть с одними и теми же начальными значениями синаптических коэффициентов. Исследования проводятся как для пакетного, так и для стохастического режимов обучения.

Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Создать МНС (опция «Создать...», см. разд. 3.3.4).
2. Дублировать созданную МНС, используя опцию «Дублировать...».
3. Обучить дублированную МНС одним из указанных выше методов (опция «Обучить...», см. разд. 3.3.5). Для обучения выбрать пакетный режим.
4. Провести анализ качества обучения сети выбранным методом, используя обучающую/валидационную/тестовую выборки.
5. Повторить пп. 2 – 4 для другого метода обучения.
6. Провести обучение исходной сети указанными выше методами в стохастическом режиме обучения.
7. Сравнить различные методы обучения в пакетном и стохастическом режимах, а также сами режимы обучения. Сделать выводы.

Так как для обучения каждый раз использовалась одна и та же сеть, то серия графиков, на которых представлены зависимости среднеквадратической ошибки на обучающей/валидационной/тестовой выборках от номера такта обучения для указанных выше ме-

тодов, характеризуется общим значением в начале процесса обучения.

3.4.5. Исследование влияния на сходимость процесса обучения параметра скорости обучения ϵ (эксперимент 5)

Цель эксперимента – анализ зависимости сходимости процедуры обучения МНС от значения параметра скорости обучения ϵ простого градиентного метода. Исследование проводится при использовании пакетного режима обучения. При этом каждый раз для обучения простым градиентным методом с различными значениями ϵ используется одна и та же нейронная сеть с одинаковыми начальными значениями синаптических коэффициентов.

Данный эксперимент организуется по той же схеме, что и эксперимент 4. Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Создать МНС (опция «Создать...», см. разд. 3.3.4).
2. Дублировать созданную МНС, используя опцию «Дублировать...».
3. Обучить дублированную МНС простым градиентным методом с выбранным значением параметра ϵ (опция «Обучить...», см. разд. 3.3.5). Для обучения выбрать пакетный режим.
4. Провести анализ качества обучения сети на основе обучающей, валидационной и тестовой выборок.
5. Повторить пп. 2 – 4 для других значений параметра ϵ .
6. Сравнить процессы обучения и качество обучения сети. Провести анализ длительности процесса обучения и особенности динамики процессов изменения среднеквадратической ошибки обучения. Сделать выводы.

Так как для обучения каждый раз использовалась одна и та же сеть, то серия графиков, на которых представлены зависимости среднеквадратической ошибки на обучающей/валидационной/тестовой выборках от номера такта обучения, для всех значений параметра ϵ , характеризуется общим значением в начале процесса обучения.

3.4.6. Влияние помех измерения входных/выходных данных обучающей выборки на точность аппроксимации функции с помощью МНС (эксперимент 6)

Цель эксперимента – проверка робастности сети при зашумлении входных данных. Исследуются качество аппроксимации функции многих переменных при наличии аддитивной помехи в выборке данных и способность обученной сети к обобщению. Зашумлению подвергаются как входные, так и желаемые выходные данные сети.

В качестве исходных данных использовать обучающую, валидационную и тестовую выборки, созданные при выполнении эксперимента 1. Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Создать МНС («Создать…», см. разд. 3.3.4).
2. Дублировать созданную МНС, используя опцию «Дублировать…».
3. Обучить дублированную сеть (опция «Обучить…», см. разд. 3.3.5) на незашумленных данных.
4. Зашумить обучающую, валидационную и тестовую выборки с указанным уровнем помехи (опция «Зашумить…» на панели «Выборки», см. разд. 3.3.6).
5. Дублировать исходную МНС, используя опцию «Дублировать…».
6. Обучить дублированную сеть (опция «Обучить…», см. разд. 3.3.5) на зашумленных данных.
7. Повторить пп. 4 – 6 для других значений уровня аддитивной помехи.
8. Провести сравнительный анализ качества обучения сети при различных уровнях помехи и ее способности к обобщению. Сделать выводы.

В процессе экспериментальных исследований рекомендуется варьировать уровень помехи от 5 до 20 %.

3.4.7. Влияние ошибок реализации синаптических коэффициентов и смещений нейронов МНС на точность аппроксимации функции (эксперимент 7)

Цель эксперимента – проверка робастности сети при зашумлении ее синаптических коэффициентов. Исследуются способность

сети с зашумленными коэффициентами аппроксимировать исходные данные и способность обученной сети к обобщению. Зашумлению подвергаются как синаптические коэффициенты, так и смещения нейронов.

В качестве исходных данных использовать обучающую, валидационную и тестовую выборки, созданные при выполнении эксперимента 1. Последовательность этапов при выполнении данного экспериментального исследования приведена ниже.

1. Создать МНС (опция «Создать...», см. разд. 3.3.4).
2. Дублировать созданную МНС, используя опцию «Дублировать...».
3. Обучить дублированную сеть (опция «Обучить...», см. разд. 3.3.5).
4. Провести анализ качества обучения сети, используя обучающую, валидационную и тестовую выборки.
5. Дублировать обученную МНС, используя опцию «Дублировать...».
6. Зашумить синаптические коэффициенты и смещения дублированной сети (опция «Зашумить...» на панели «Нейросети», см. разд. 3.3.6).
7. Повторить пп. 5 – 7 для других значений уровня аддитивной помехи синаптических коэффициентов и смещений.
8. Провести анализ качества аппроксимации данных сетью с зашумленными параметрами, сравнить значения среднеквадратичной ошибки на обучающей, валидационной и тестовой выборках при различных уровнях помехи. Сделать выводы.

В процессе экспериментальных исследований рекомендуется варьировать уровень помехи от 5 до 20 %.

3.5. Рекомендации к оформлению отчета

Отчет по работе должен содержать описания исходных данных и условий проведения каждого эксперимента, а также полученные результаты в числовой, графической или табличной форме и выводы.

Ниже приводится содержание отчета с краткими комментариями.

1. Исследование процесса настройки параметров многослойного персептрона

Приводятся обозначения входных и желаемых выходных переменных, область моделирования функции, объем выборки данных. По результатам визуального анализа расположения выборочных точек делается вывод о равномерности их распределения в пространстве признаков и в случае неравномерного распределения указываются подобласти с высокой и низкой плотностью.

Указывается начальная архитектура МНС, используемой для аппроксимации функции, общее число настраиваемых параметров сети и рассчитывается среднее число примеров обучающей выборки, приходящееся на один настраиваемый параметр.

Выполняются подготовительные операции перед запуском процедуры обучения МНС простым градиентным методом. Выбирается фиксированное значение параметра скорости обучения ϵ . После завершения процедуры обучения в пакетном режиме на графике изображается зависимость критериальной функции на обучающей и валидационной выборках от номера эпохи обучения. Проводится анализ этих зависимостей, в результате которого делается вывод о возможном переобучении сети. Фиксируются начальное и финальное значения среднеквадратического отклонения, и проводится их сравнение.

На отдельном графике изображаются ошибки обработки сетью тестовых примеров, на графике отмечается минимальное и максимальное значения ошибки по одному из выходов сети. Приводятся гистограммы распределения средних абсолютных ошибок аппроксимации одной из выходных переменных на обучающей и валидационной выборках. По результатам анализа построенных графиков и гистограмм делается вывод о способности обученной нейронной сети к обобщению.

2. Анализ информационной активности нейронов и контрастирование сети

Приводится график вариабельности нейронов сети на обучающей и тестовой выборках. По результатам анализа вариабельности делается вывод о том, какие нейроны могут быть исключены из сети. Для каждого них указывается номер слоя и номер нейрона в

пределах слоя в обученной сети. Для сети с сокращенным числом нейронов рассчитывается среднее число примеров обучающей выборки, приходящееся на один настраиваемый параметр.

После дообучения сети фиксируются значения критериальной функции на обучающей, валидационной и тестовой выборках, проводится сравнение этих значений с полученными в эксперименте 1. По результатам обучения делается вывод о способности обученной сети к обобщению.

3. Экспериментальный поиск архитектуры многослойного персептрона с улучшенными свойствами по точности

Приводятся архитектуры нейронных сетей, для которых проводилось сравнение точности аппроксимации. Для каждой архитектуры указываются значения критериальной функции на обучающей, валидационной и тестовой выборках.

4. Сравнительный анализ различных вычислительных процедур обучения МНС

Приводятся архитектура нейронной сети и объем обучающей выборки. На одном графике для всех методов обучения изображается зависимость среднеквадратического отклонения на примерах обучающей выборки от номера эпохи обучения при обучении в пакетном режиме. На другом графике – зависимость среднеквадратического отклонения на примерах тестовой выборки.

По результатам обучения различными методами проводится сравнительный анализ методов, указывается метод, с помощью которого было достигнуто минимальное значение среднеквадратического отклонения на обучающей выборке.

Аналогичные результаты приводятся для стохастического режима обучения.

5. Исследование влияния на сходимость процесса обучения параметра скорости обучения ε

Приводятся архитектура нейронной сети и объем обучающей выборки. На одном графике для всех значений параметра ε простого градиентного метода изображается зависимость среднеквадра-

тического отклонения на примерах обучающей выборки от номера эпохи обучения при обучении в пакетном режиме. Делается вывод о влиянии параметра ϵ на процесс обучения.

6. Влияние помех измерения входных/выходных данных обучающей выборки на точность аппроксимации функции с помощью МНС

Приводится архитектура нейронной сети и объем обучающей выборки. Указываются метод обучения сети и различные значения уровня шума. После завершения процедуры обучения сети на зашумленных обучающих выборках фиксируется достигнутое значение среднеквадратического отклонения на примерах обучающей и тестовой выборках. Делается вывод о влиянии уровня помех измерения входных/выходных данных на качество аппроксимации.

7. Влияние ошибок реализации синаптических коэффициентов и смещений нейронов МНС на точность аппроксимации функции

Приводится архитектура нейронной сети и объем обучающей выборки. Указываются метод обучения сети и различные значения уровня шума. После завершения процедуры обучения сети фиксируется достигнутое значение среднеквадратического отклонения на примерах обучающей и тестовой выборках. Для выбранных значений уровня случайной помехи приводится значение среднеквадратического отклонения сети с зашумленными синаптическими коэффициентами. Делается вывод о влиянии уровня зашумления параметров сети на качество аппроксимации.

8. Общие выводы по проведенным исследованиям

Излагаются особенности процесса обучения многослойной сети при использовании различных методов и особенности применения МНС для аппроксимации функций. Делается вывод о способности многослойной сети решать задачу аппроксимации в условиях помех измерения входных/выходных данных и при зашумленных синаптических коэффициентах.

Контрольные вопросы

1. Напишите уравнения функционирования многослойной нейронной сети.
2. Объясните состав данных таблицы обучающей выборки, используемой для настройки параметров нейронной сети.
3. Какие параметры многослойной нейронной сети настраиваются в процессе ее обучения?
4. Какой критерий используется для организации обучения многослойной нейронной сети?
5. Какой метод применяется для обучения многослойной нейронной сети в используемом в работе нейроэмулаторе?
6. Чем характеризуется эффект «паралича» при обучении многослойной нейронной сети?
7. В чем состоит и как реализуется процесс инициализации при обучении нейронной сети?
8. В чем состоит и как проверяется эффект «генерализации данных» в нейронной сети?
9. Какие параметры режима обучения многослойной нейронной сети доступны пользователю для настройки?
10. В чем состоит процедура валидации многослойной нейронной сети?
11. В чем состоит процедура тестирования обученной многослойной нейронной сети?
12. Какие переменные вычисляются с помощью метода обратного распространения ошибки и как они используются в процессе обучения многослойной нейронной сети?
13. Приведите примеры активационных характеристик нейронов, используемых в многослойных нейронных сетях.
14. Что называется «эпохой» в процессе обучения нейронной сети?
15. Чем отличаются пакетный и стохастический режимы обучения многослойной нейронной сети?
16. Почему разные реализации процесса обучения многослойной нейронной сети из разных начальных условий не приводят к одному и тому же финальному результату?
17. По какому правилу производится модификация значений параметров многослойной нейронной сети при ее обучении методом обратного распространения ошибки?

Лабораторная работа 4

ПРИМЕНЕНИЕ МНОГОСЛОЙНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ ОБРАБОТКИ ДАННЫХ

4.1. Цель и задачи лабораторной работы

Цель: построение многослойной нейронной сети для решения прикладной задачи обработки данных и исследование ее точностных характеристик.

Области практических приложений многослойных нейронных сетей чрезвычайно разнообразны: медицинская диагностика и назначение для больных индивидуальной лекарственной терапии, выявление объекта или некоторого качественного факта в зашумленных данных, прогнозирование значений определенных количественных характеристик объекта по доступным измерениям других характеристик, построение математических моделей для слабо формализованных явлений по выборочным примерам, прогноз финансовых временных рядов и др.

С формальной точки зрения прикладные задачи в подавляющем большинстве случаев относятся к одному из трех типов:

- аппроксимация функции многих переменных;
- прогноз значений временного ряда;
- классификация векторных данных.

Каждый студент получает индивидуальное задание по применению многослойной нейронной сети для решения задачи обработки данных одного из трех перечисленных выше типов. Для построения нейросетевой модели и проведения экспериментальных исследований применяется тот же нейроэмулатор, что и в лабораторной работе 3.

4.2. Варианты индивидуальных заданий

Вариант индивидуального задания выдается студенту непосредственно на лабораторной работе. Каждый вариант задания представлен совокупностью файлов с обучающей, валидационной и тестовой выборками, используемыми для построения нейросетевой

модели, а также файлом с контрольной выборкой, на данных которой применяется обученная сеть.

4.3. Содержание лабораторных исследований

В лабораторной работе предлагается решение практической задачи аппроксимации функции/прогноза временного ряда/классификации данных с использованием нейросетевого подхода. Лабораторная работа включает выполнение следующих заданий и исследований:

- визуализация данных;
- построение и исследование многослойного персептрона;
- экспериментальный поиск архитектуры многослойного персептрона с улучшенными свойствами по точности;
- анализ результатов обработки данных обученной нейросетью.

4.3.1. Визуализация данных. Построение и исследование нейросетевой модели (эксперимент 1)

В данном исследовании проводится визуальный анализ данных, используемых для построения нейросетевой модели. Цель исследования – оценка сложности аппроксимируемой функции/динамики временного ряда/расположения классов и оценка числа слоев и нейронов в многослойном персептроне, предназначенном для решения соответствующей задачи; построение нейросетевой модели и анализ ее точности.

В качестве исходных данных для лабораторной работы используются обучающая, валидационная, тестовая и контрольная выборки, загружаемые из соответствующих файлов. Имена файлов определяются в соответствии с вариантом задания к лабораторной работе.

Для визуального анализа данных используется опция «Анализ...» на панели «Выборки» главного окна программы. При нажатии на кнопку «Анализ...» появляется окно «Визуализация и анализ данных», в котором необходимо указать тип решаемой задачи – аппроксимация функции, прогноз временного ряда или классификация данных, и, используя опции в данном окне, построить необходимые графики.

После анализа сложности данных студент создает многослойный персептрон, для обучения, валидации и тестирования которого используются загруженные выборки данных.

Последовательность этапов при выполнении данного экспериментального исследования приведена ниже. Описание используемых опций нейроэмулатора приведено в лабораторной работе 3.

1. Предобработка данных (опция «Предобработка...»).
2. Создание многослойного персептрана (опция «Создать...»).
3. Обучение, валидация и тестирование многослойного персептрана (опция «Обучить...»).
4. Анализ результатов обучения (опция «Анализ...»).

При выполнении предобработки исходных данных рекомендуется провести их масштабирование на интервал $[-0,9; 0,9]$. В качестве метода оптимизации параметров многослойного персептрана выбрать простой градиентный метод с параметром скорости обучения $0,6 - 0,8$. При медленной сходимости процедуры обучения увеличить это значение, при возникновении колебательного характера процедуры обучения – уменьшить.

4.3.2. Экспериментальный поиск архитектуры многослойного персептрана с улучшенными свойствами по точности (эксперимент 2)

Цель эксперимента – исследование и поиск архитектуры нейронной сети, показывающей наилучшие значения по качеству обработки сетью загруженных данных с точки зрения достигнутой точности на обучающей, валидационной и тестовой выборках.

Порядок проведения данного исследования аналогичен изложенному в разд. 4.3.1. Перед проведением исследования создать несколько МНС различных архитектур и последовательно их обучить.

4.3.3. Анализ результатов обработки данных обученной нейросетью (эксперимент 3)

В данном исследовании проводится визуальный анализ результатов обработки обученным многослойным персептраном данных обучающей, валидационной, тестовой и контрольной выборок. Цель эксперимента – оценка качества построенной нейросетевой

модели и ее способности к обобщению на основе визуального представления результатов ее работы.

Для визуального анализа данных используется опция «Анализ...» на панели «Нейросети» главного окна программы. При нажатии на кнопку «Анализ...» появляется окно «Визуализация и анализ результатов», в котором необходимо указать тип решаемой задачи – аппроксимация функции, прогноз временного ряда или классификация данных, и, используя опции в данном окне, построить необходимые графики.

4.4. Рекомендации к оформлению отчета

Отчет по работе должен содержать описания исходных данных и условий проведения каждого эксперимента, а также полученные результаты в числовой, графической или табличной форме и выводы.

Ниже приводится содержание отчета с краткими комментариями.

1. Визуализация данных

Приводятся имена файлов с исходными данными для построения и применения нейросетевой модели: обучающей, валидационной, тестовой и контрольной выборками. Для каждой выборки приводится ее объем. На графиках изображается расположение выборочных данных в пространстве признаков. По результатам визуального анализа делается вывод о равномерности их распределения в пространстве признаков и в случае неравномерного распределения указываются подобласти с высокой и низкой плотностью.

2. Построение и исследование нейросетевой модели

Указывается начальная архитектура МНС, используемой для аппроксимации функции, общее число настраиваемых параметров сети и рассчитывается среднее число примеров обучающей выборки, приходящееся на один настраиваемый параметр.

Выполняются подготовительные операции перед запуском процедуры обучения МНС простым градиентным методом. Выбирается фиксированное значение параметра скорости обучения ϵ . После

завершения процедуры обучения в пакетном режиме на графике изображается зависимость критериальной функции на обучающей и валидационной выборках от номера эпохи обучения. Проводится анализ этих зависимостей, в результате которого делается вывод о возможном переобучении сети. Фиксируются начальное и финальное значения среднеквадратического отклонения, и проводится их сравнение.

На отдельном графике изображаются ошибки обработки сетью тестовых примеров, на графике отмечается минимальное и максимальное значения ошибки по одному из выходов сети. Приводятся гистограммы распределения средних абсолютных ошибок аппроксимации одной из выходных переменных на обучающей и валидационной выборках. По результатам анализа построенных графиков и гистограмм делается вывод о способности обученной нейронной сети к обобщению.

3. Экспериментальный поиск архитектуры многослойного персептрона с улучшенными свойствами по точности

Приводятся архитектуры нейронных сетей, для которых проводилось сравнение точности аппроксимации. Для каждой архитектуры указываются значения критериальной функции на обучающей, валидационной и тестовой выборках.

4. Анализ результатов обработки данных обученной нейросетью

По результатам обработки сетью обучающей выборки приводятся:

- в задаче аппроксимации – график зависимости аппроксимируемой функции от аргумента;
- в задаче прогноза временных рядов – график зависимости значений ряда от времени;
- в задаче классификации – визуальное расположение точек в пространстве признаков с раскраской каждой точки цветом, соответствующим классу.

Аналогичные графики строятся по результатам обработки сетью тестовой выборки. Результаты работы сети на данных контрольной выборки приводятся в отдельной таблице. Визуально делается оценка качества построенной нейросетевой модели.

5. Общие выводы по проведенным исследованиям

Излагаются особенности применения многослойных нейронных сетей для решения задач обработки данных. По результатам проведенных исследований делается вывод о способности многослойной сети решать поставленную в лабораторной работе задачу.

Контрольные вопросы

1. Объясните содержание задачи аппроксимации функции многих переменных на многослойной нейронной сети.
2. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче аппроксимации функции многих переменных?
3. Объясните содержание задачи прогноза временных рядов на многослойной нейронной сети.
4. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче прогноза временных рядов?
5. Объясните содержание задачи классификации данных на многослойной нейронной сети.
6. В чем состоит этап подготовки данных для обучения многослойной нейронной сети в задаче классификации данных?
7. Какой критерий используется при решении задачи классификации данных с помощью многослойной нейронной сети?
8. Какие правила останова процесса обучения МНС применяются в практических приложениях?
9. В чем состоит функциональная задача дополнительного блока принятия решения на выходе МНС при использовании ее в качестве классификатора данных?
10. Почему чрезмерное увеличение числа нейронов скрытых слоев в МНС при решении задачи аппроксимации может ухудшить точностные показатели сети? Какое свойство нейронной сети имеется в виду?
11. Почему на задачи классификации данных и прогноза временного ряда распространяются те же рекомендации по выбору архитектуры МНС, что и в задаче аппроксимации функции многих переменных?

12. Какие вы можете предложить нейросетевые решения задачи прогноза временного ряда одновременно на несколько последовательных временных тактов?

13. Предложите вариант построения нейросетевого классификатора при числе классов 4 с использованием многослойных нейросетевых классификаторов на 2 класса.

Лабораторная работа 5

АССОЦИАТИВНАЯ ПАМЯТЬ. СЕТЬ ХОПФИЛДА

5.1. Цель и задачи лабораторной работы

Цель: изучение математической модели и исследование свойств сети Хопфилда; анализ динамики переходного процесса сети из различных начальных состояний и способности сети хранить и ассоциативно восстанавливать хранящиеся в памяти образы; проверка устойчивости функционирования ассоциативной памяти при зашумлении параметров сети.

В лабораторной работе сеть Хопфилда рассматривается как модель ассоциативной памяти. В качестве запоминаемых образов используется набор биполярных векторов (элементы векторов могут иметь одно из двух возможных значений: $\{-1; +1\}$) одинаковой заданной длины $N = 88$. Для удобства зрительного восприятия эти векторы визуализируются в виде прямоугольной матрицы с заданным числом строк $n_1 = 11$ и столбцов $n_2 = 8$ ($n_1 \cdot n_2 = N$). Элементам вектора, равным -1 , ставится в соответствие белая клетка, равным $+1$ – черная (рис. 5.1). Для создания и редактирования подобных наборов образов в программном обеспечении лабораторного занятия предусмотрен графический редактор, который сохраняет вектора и параметры n_1 и n_2 в текстовых файлах специального формата с расширением `pat`. Выполнение экспериментов осуществляется с помощью нейроэмулатора, реализованного в среде MATLAB.

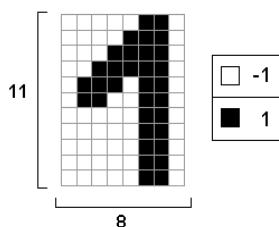


Рис. 5.1. Графическая интерпретация образа. Вектор из 88 элементов представлен в виде матрицы с 11 строками и 8 столбцами

Эксперимент 1 посвящен исследованию процесса эволюции сети, настроенной на один или несколько образцов, от различных начальных состояний к финальному устойчивому состоянию.

В эксперименте 2 осуществляется поиск множества устойчивых состояний сети и оценка размера бассейнов аттракторов.

В эксперименте 3 проводится анализ способности сети сохранять в качестве аттракторов некоторое множество образов и релаксировать к одному из них при подаче на вход образа с искажениями, т.е. являться моделью ассоциативной памяти.

В эксперименте 4 проверяется способность сети сохранять свойство ассоциативной памяти после искажения матрицы синаптических связей.

В лабораторной работе используется классическая модель сети Хопфилда, построенная на bipolarных нейронах и функционирующая в асинхронном режиме.

Программное обеспечение для выполнения лабораторной работы написано А.А. Лабинской.

5.2. Варианты индивидуальных заданий

Индивидуальный вариант задания и файл с образами для экспериментов выдаются непосредственно на лабораторном занятии. Ниже приведен пример индивидуального варианта:

Создать сети Хопфилда для каждой из 3-х групп образов, выполнить для каждой из них эксперименты, указанные в описании лабораторной работы.

1. Цифра «3».
2. Цифры «1», «3», «4».
3. Цифры «1», «2», «3», «4», «5», «6», «7», «8», «9».

Для сетей 1 – 3 использовать стандартное множество образов.

На рис. 5.2 показано стандартное множество образов для этого варианта задания.

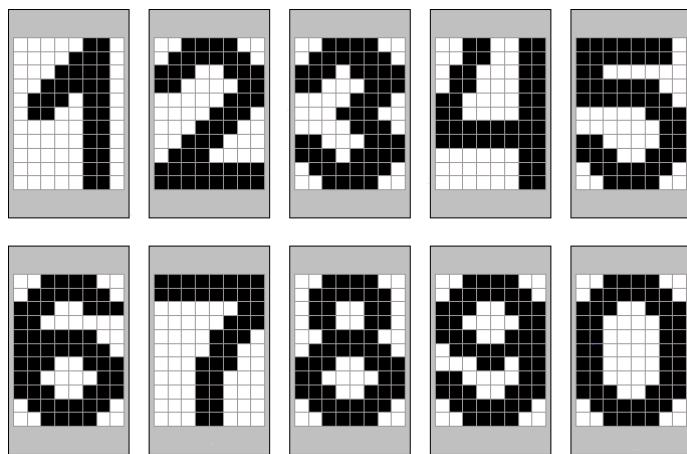


Рис. 5.2. Стандартное множество образов,
упомянутое в приведенном примере варианта задания

5.3. Руководство пользователя

Нейроэмулатор позволяет исследовать четыре важных аспекта классической модели сети Хопфилда:

- динамику (процесс эволюции сети от начального к финальному устойчивому состоянию);
- атTRACTоры (множество устойчивых состояний сети, размер бассейнов атTRACTоров);
- ассоциативную память (способность сети сохранять в качестве атTRACTоров некоторое множество образов и релаксировать к сохраненному образу при подаче на вход сети этого же образа с искажениями);
- робастность (способность сети сохранять свойство ассоциативной памяти после искажения матрицы синаптических связей, т. е. образ с одинаковым уровнем шума правильно восстанавливается сетью до и после внесения искажений в синаптическую матрицу).

Для редактирования запоминаемых сетью и входных образов в нейроэмулаторе реализован графический редактор образов.

5.3.1. Общее описание эмулятора

Программа запускается по команде `hopfnet` из командной строки системы MATLAB. При запуске программы появляется окно, показанное на рис. 5.3. В выпадающем меню в нижней части стартового окна требуется выбрать номер индивидуального варианта, после чего нажать кнопку «OK».

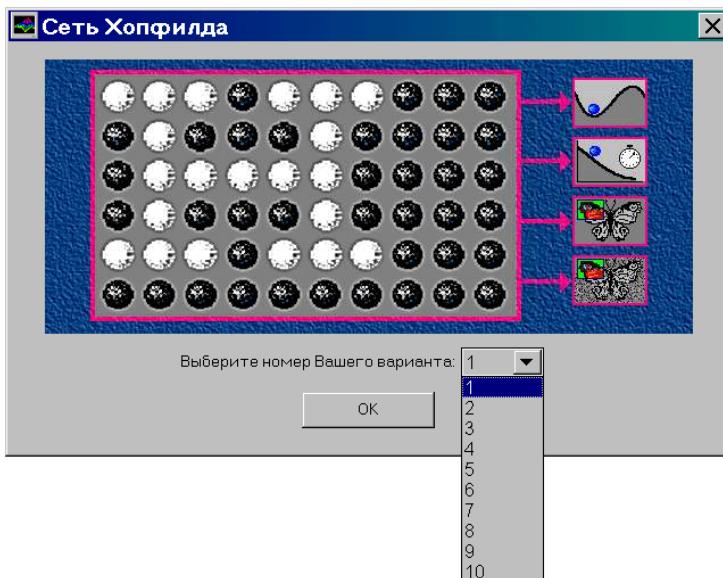


Рис. 5.3. Окно, которое появляется при запуске программы «Сеть Хопфилда»

В результате появляется главное окно программы, вид которого показан на рис. 5.4. Непосредственно после запуска программы недоступен пункт «Эксперименты» главного меню и четыре кнопки в правой части окна (дублирующие подпункты меню «Эксперименты»), поскольку нейронная сеть Хопфилда пока не рассчитана.

Кратко остановимся на содержании пунктов главного меню. Меню «Файл» позволяет сохранить результаты работы и выйти из программы. Меню «Параметры» включает два подпункта: «Образы» и «Сеть», которые дублируют одноименные кнопки в левой части окна. Меню «Помощь» помимо информации о программе включает пункт «Задание», которому соответствует одноименная кнопка в нижней части главного окна.

При запуске программы «Сеть Хопфилда» загружается текст индивидуального задания с указанным номером и соответствующий варианту стандартный набор образов. Для того, чтобы просмотреть текст задания, нужно нажать на кнопку «Задание». Появляется окно, показанное на рис. 5.5 (текст в окне отличается для разных вариантов).

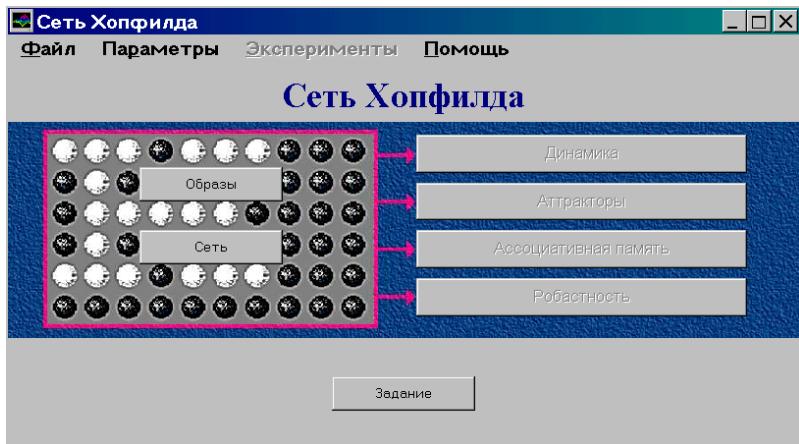


Рис. 5.4. Главное окно программы «Сеть Хопфилда» непосредственно после запуска

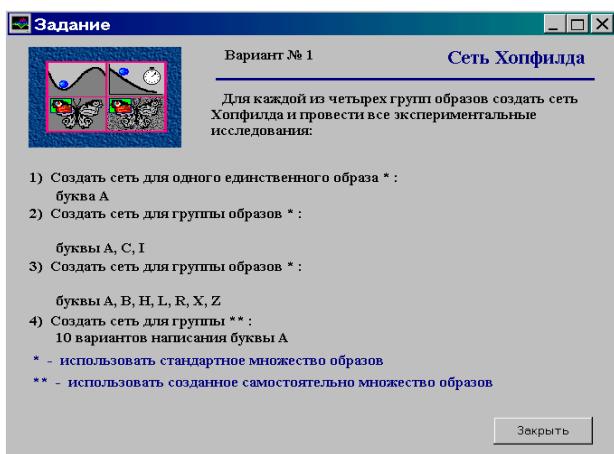


Рис. 5.5. Окно «Задание»

Для того чтобы внести изменения в текущее множество образов или загрузить новое, нужно нажать на кнопку «Образы» главного окна или выбрать пункт «Образы» меню «Параметры». Появляется окно, показанное на рис. 5.6. Центральная часть окна занята изображениями образов из текущего набора (в виде черно-белых диаграмм). Под каждым изображением выведено имя соответствующего образа. Однократное нажатие мыши на диаграмме устанавливает соответствующий образ текущим, пространство вокруг образа выделяется цветом. В нижней части окна «Образы» выводится информация о размерности образов и количестве образов в текущем наборе. Кнопка «Открыть файл» предназначена для загрузки нового множества образов из файла с расширением `pat`. Для того, чтобы вернуться к главному окну, загрузив новое множество образов, требуется нажать кнопку «Применить».

При нажатии на кнопку «Редактор» в окне «Образы», запускается программа «Редактор образов», вид главного окна которой показан на рис. 5.6.

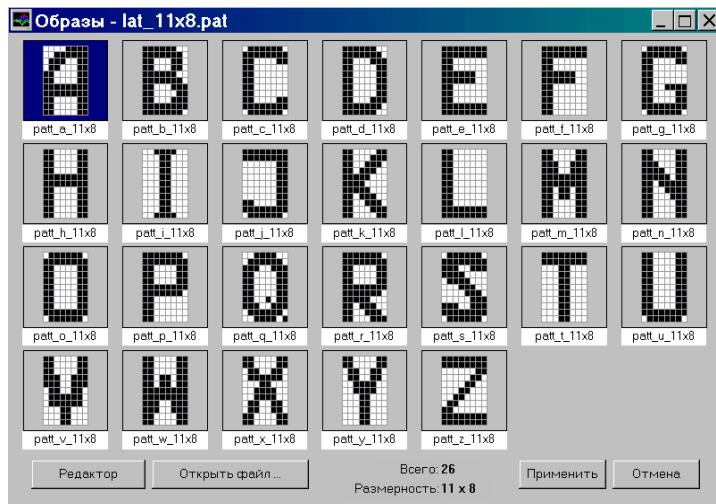


Рис. 5.6. Окно «Образы»

5.3.2. Описание программы «Редактор образов»

Программа «Редактор образов» является частью программного пакета «Сеть Хопфилда» и работает в среде MATLAB 5.0. Программа предназначена для создания и редактирования наборов биполярных образов, которые сохраняются в текстовых файлах специального формата. Все образы набора (множества) должны иметь одинаковую размерность, т. е. одинаковую длину вектора признаков и одинаковое число строк и столбцов для графического представления в виде черно-белой диаграммы.

Как видно на рис. 5.7, один из образов набора изображен в центре экрана. Этот образ называется *текущим*. В результате однократного щелчка кнопки мыши на клетке диаграммы текущего образа цвет этой клетки устанавливается выбранным из «палитры» (белый или черный) в левом нижнем углу главного окна «Редактора образов». На рис. 5.7 видно, что текущим цветом является черный. Для выбора цвета рисования нужно щелкнуть кнопкой мыши на черном или белом прямоугольнике палитры. Операции редактирования применяются к текущему образу. Для того чтобы сделать текущим другой образ из набора, используйте список в правой части окна. Заголовок окна начинается с имени файла, из которого загружено множество образов. В нижней части экрана выводится информация о размерности и количестве образов в наборе.

Рассмотрим главное меню программы «Редактор образов». Меню «Файл» включает команды «Новый», «Открыть ...», «Сохранить», «Сохранить как ...» и «Выход». Команда «Новый» предназначена для создания нового множества образов. При этом требуется в специальном диалоговом окне задать размерность (число строк и столбцов) нового множества образов. Команда «Открыть ...» используется для того, чтобы открыть имеющийся файл с множеством образов. Команды «Сохранить» и «Сохранить как ...» предназначены для сохранения текущего множества образов под текущим и новым именем соответственно.

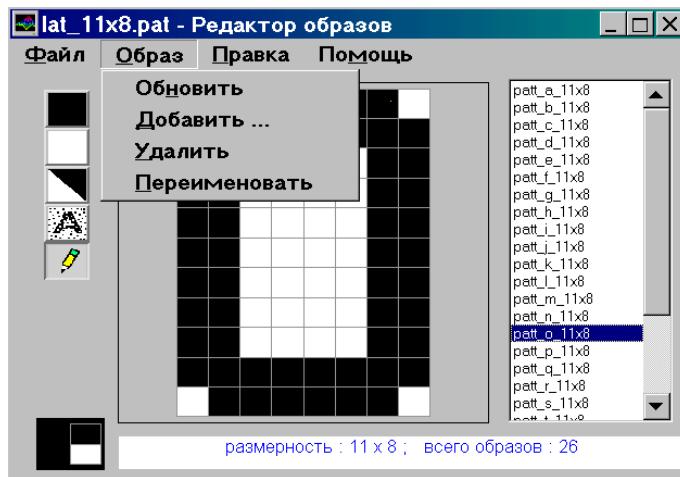


Рис. 5.7. Главное окно программы «Редактор образов»



Рис. 5.8. Окно «Зашумить»

Меню «Образ» (см. рис. 5.7) главного окна программы «Редактор образов» содержит подпункты «Обновить», «Добавить ...», «Удалить» и «Переименовать». Команда «Обновить» должна быть

выполнена для сохранения результата редактирования перед сменой текущего образа. Команда «Добавить ...» предназначена для добавления нового образа в набор. При этом в появляющемся специальном окне нужно задать имя добавляемого образа. Команда «Удалить» используется для удаления текущего образа из набора. С помощью команды «Переименовать» можно изменить имя текущего образа.

Меню «Правка» включает команды редактирования «Инвертировать», «Закрасить черным», «Закрасить белым», «Зашумить» (дублируются кнопками главного окна «Редактора образов», расположеннымми слева) и команды «Отменить», «Повторить» (результат последней выполненной по отношению к текущему образу команды редактирования). Команды редактирования меняют цвет клеток диаграммы с белого на черный или наоборот (и, соответственно, значения элементов вектора признаков с -1 на $+1$ или с $+1$ на -1). Команда инвертировать меняет цвет всех клеток диаграммы. Команды «Закрасить черным» и «Закрасить белым» устанавливают один и тот же цвет для всех клеток. По команде «Зашумить» появляется специальное окно, показанное на рис. 5.8.

Окно «Зашумить» предназначено для автоматического внесения искажений в текущий образ. В левой верхней части окна изображается искаженный результат, на который будет заменен текущий образ «Редактора образов» после нажатия кнопки «Применить». В центре имеется шкала с ползунком, который управляется мышью, для указания уровня зашумления в процентах. Установленный процент искажения выводится слева от шкалы (на рис. 5.8 уровень шума равен 25 %). С помощью переключателя выбирается один из трех режимов искажения: изменить цвет, закрасить белым, закрасить черным (на рис. 5.8 выбран режим «Закрасить белым»). Автоматическое искажение выполняется по такому закону: каждая клетка диаграммы изменяется в соответствии с выбранным режимом с вероятностью, равной заданному уровню шума.

При выходе из «Редактора образов» появляются диалоговые окна с предложением сохранить в файле отредактированное множество образов и загрузить это множество в программу «Сеть Хопфилда».

5.3.3. Создание сети Хопфилда

Для создания (расчета) сети Хопфилда, над которой можно проводить экспериментальные исследования, предназначено окно «Сеть» (рис. 5.9), которое вызывается при нажатии одноименной кнопки главного окна программы «Сеть Хопфилда» или по команде «Сеть» меню «Параметры». Основную часть окна занимают черно-белые изображения образов, слева имеется список имен всех образов текущего множества. Образ выделяется цветом фона при нажатии мыши на изображении или при указании в списке. Общее число и число выделенных образов выводятся внизу окна.

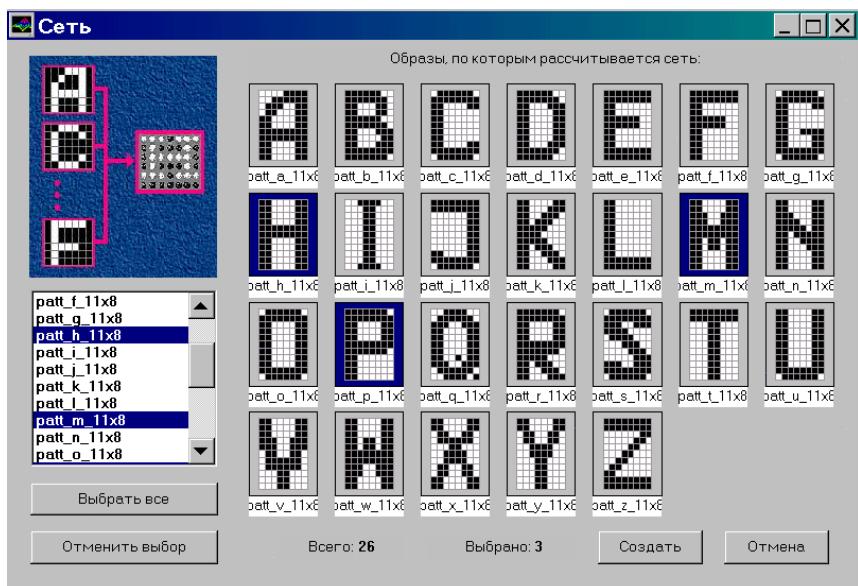


Рис. 5.9. Окно «Сеть»

Для расчета сети необходимо выделить нужные образы и нажать кнопку «Создать». В окне, изображенном на рис. 5.9, выбраны три образа, по которым будет рассчитана синаптическая матрица сети. Последующие иллюстрации к описанию программы «Сеть Хопфилда» соответствуют именно этой сети.

5.3.4. Описание рабочих окон

На рис. 5.10 показано окно эксперимента «Динамика» сразу после открытия. Для выполнения эксперимента нужно задать начальное состояние сети и выбрать режим просмотра, после чего нажать кнопку «Пуск». Начальное состояние может быть либо случайным вектором, сгенерированным автоматически, либо одним из тех векторов, по которым рассчитана сеть (способ задания начального состояния нужно указать с помощью переключателя). Предусмотрено три режима просмотра: непрерывный, пауза после каждого такта, пауза после обработки каждого нейрона. Вектор начального состояния сети можно редактировать щелчком мыши на клетках диаграммы (в режиме инвертирования). Как видно на рисунке, заданному начальному состоянию соответствует значение энергии сети 0,15909. На рис. 5.11 показано окно «Динамика» после завершения переходного процесса в сети Хопфилда. В центре изображено конечное состояние сети, число тактов (равно трем), за которое оно было достигнуто, соответствующие значение энергетического функционала ($-80,5$) и график зависимости энергии сети от времени. На графике по вертикальной оси отложено значение энергии, а по горизонтальной – номер обработанного нейрона. Для того, чтобы снова наблюдать переходный процесс, нужно нажать кнопку «Повторить эксперимент».

На рис. 5.12 представлено окно эксперимента «Аттракторы». Поскольку сеть рассчитана более чем по одному образу (по трем образам, соответствующим буквам латинского алфавита «Н», «М» и «Р»), имеет смысл при анализе результатов эксперимента учитывать корреляцию образов. В левой части окна для пары образов, выбранной из списка, отображается коэффициент корреляции. Единственным начальным условием эксперимента «Аттракторы» является число начальных состояний, из которых осуществляется поиск. Поиск начинается при нажатии кнопки «Пуск». В программе предусмотрено средство для принудительной остановки процесса поиска.

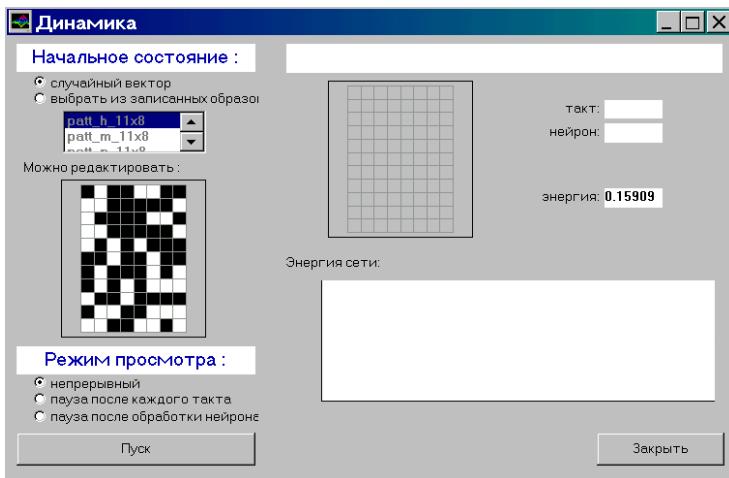


Рис. 5.10. Окно «Динамика»

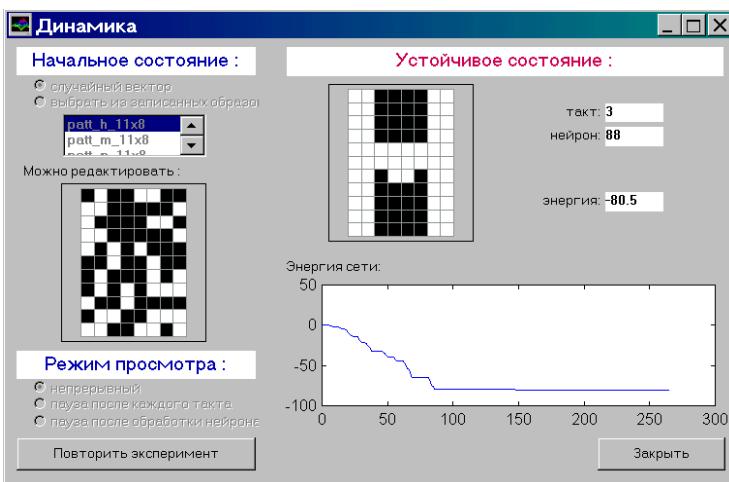


Рис. 5.11. Окно «Динамика» после завершения переходного процесса в сети Хопфилда

На рис. 5.13 показан результат поиска аттракторов. Вверху окна отображено число найденных устойчивых состояний и фактическое число начальных состояний, которые проверялись во время поиска. В центре окна находится круговая диаграмма оценки отно-

сительных размеров бассейнов. Бассейн аттрактора, выделенного в списке справа от круговой диаграммы, на экране монитора будет отображен синим цветом. Под изображением аттрактора выведена оценка относительного размера бассейна с точностью до двух знаков после запятой. Для того чтобы просмотреть другой найденный аттрактор, нужно либо один раз щелкнуть мышью на изображении одного из секторов круговой диаграммы, закрашенных серым цветом, либо выбрать другой пункт в списке найденных аттракторов. Найденные аттракторы можно сохранить в текстовом файле на диске в формате набора образов с расширением *rat*. Для этого предназначена кнопка «Сохранить аттракторы ...». В появляющемся диалоговом окне можно выбрать один из трех вариантов сохранения: добавить к текущему множеству образов, добавить к другому множеству образов, сохранить в отдельном файле.

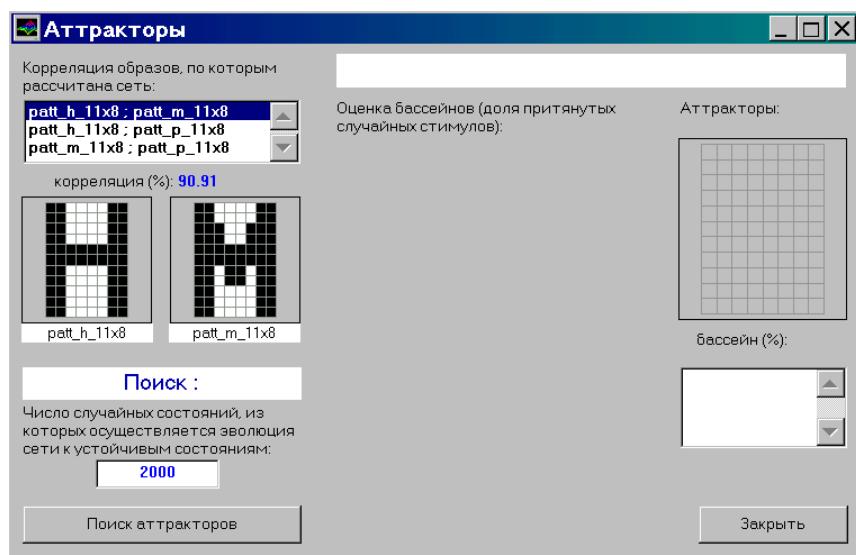


Рис. 5.12. Окно «Аттракторы»

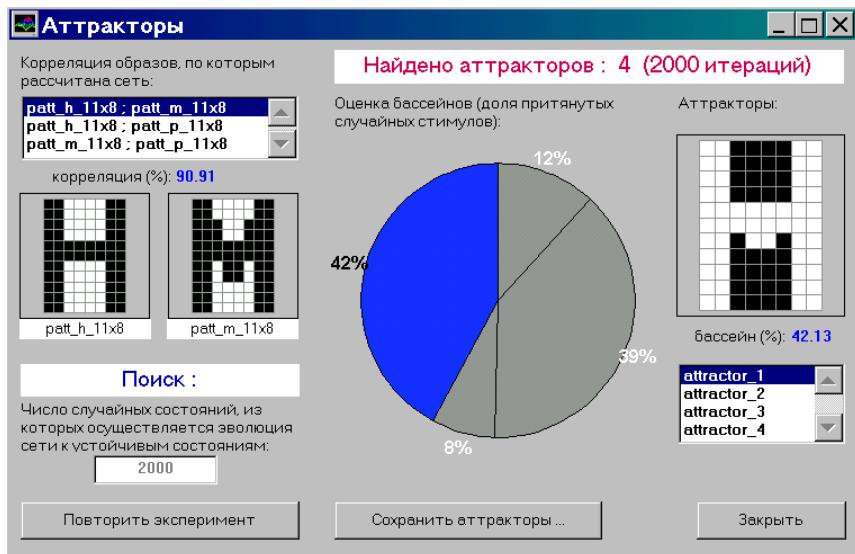


Рис. 5.13. Окно «АтTRACTоры». Результаты поиска

Окно эксперимента «Ассоциативная память» показано на рис. 5.14. Начальное состояние сети задается следующим образом. Сначала нужно один из образов выбрать в списке записанных в сеть (в левой части окна). Этот образ изображается на диаграмме в центре окна. Выбранное изображение можно редактировать мышью в режиме инвертирования и с использованием кнопок, расположенных над диаграммой. Эти кнопки функционально эквивалентны кнопкам «Редактора образов». По мере изменения исходного образа пересчитывается и выводится в нижней части окна расстояние Хемминга между исходным вектором и текущим состоянием (на рис. 5.14 оно равно 16). Когда начальное состояние задано, следует нажать кнопку «Пуск». На рис. 5.15 показан вид окна после срабатывания ассоциативной памяти. В правой части окна изображено конечное состояние сети. Видно, что оно совпадает с исходным образом. Это считается правильным распознаванием, и диаграмма конечного состояния выделяется зеленым цветом фона. На рис. 5.16 показан результат другого эксперимента с тем же исходным образом. Видно, что начальное состояние сильнее отличается от исходного образа (расстояние Хемминга равно 28). Конечное состояние не совпадает с исходным вектором. Неправильность

распознавания образа подчеркивается красным цветом фона диаграммы конечного состояния сети.

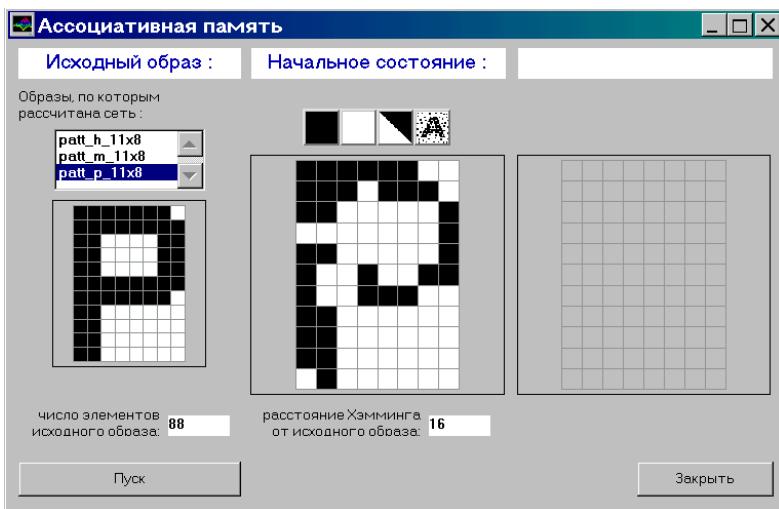


Рис. 5.14. Окно «Ассоциативная память»

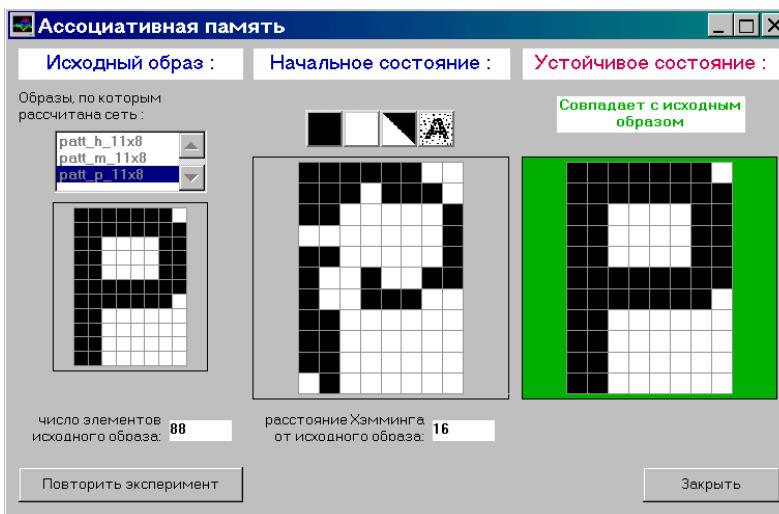


Рис. 5.15. Окно «Ассоциативная память». Правильное распознавание

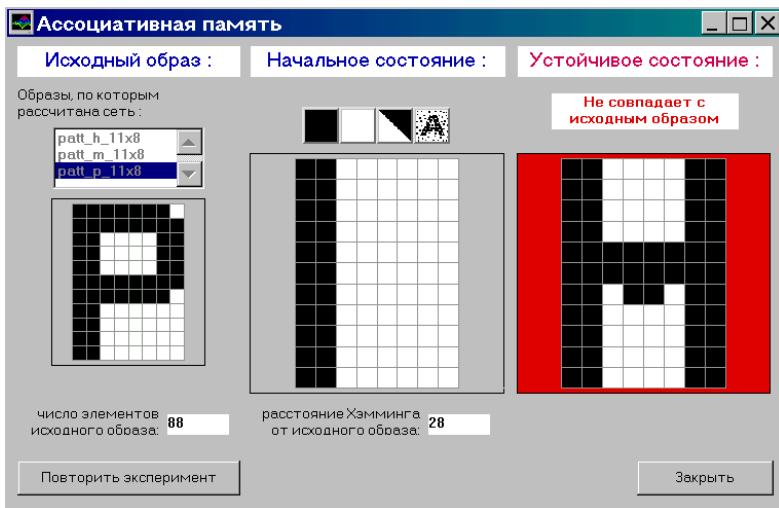


Рис. 5.16. Окно «Ассоциативная память». Неправильное распознавание

На рис. 5.17 показано окно эксперимента «Робастность». Правая часть окна почти совпадает функционально с элементами окна «Ассоциативная память». Точно так же, как и в предыдущем эксперименте, нужно выбрать исходный образ из списка и задать начальное состояние сети, зашумив исходный образ. Левая и центральная части окна предназначены для задания параметров автоматического искажения синаптической матрицы нейронной сети. Итак, начальными условиями эксперимента являются входной образ и текущая матрица коэффициентов связей. Для искажения синаптической матрицы нужно выполнить следующие действия: задать амплитуду шума (в редактируемой строке), выставить находящийся рядом флаг «Применить», выставить желаемым образом флаги «Сохранять симметричность» и «Сохранять нули по главной диагонали», нажать кнопку «Зашумить исходную». Результаты этих действий изображаются на условной диаграмме в виде прямоугольников. Вид окна «Робастность» с заданными условиями эксперимента приведен на рис. 5.18. Теперь можно нажать кнопку «Пуск». Результат срабатывания сети выводится в правом нижнем углу окна и интерпретируется аналогично эксперименту «Ассоциативная память». На рис. 5.19 и 5.20 показаны результаты распознавания для различных начальных состояний.

Начальные условия и результаты всех экспериментов для текущей сети сохраняются в процессе работы с программой. Эти данные могут быть записаны в текстовый файл на диске по команде «Сохранить последний результат» или «Сохранить все результаты» меню «Файл» главного окна. В первом случае в файл записывается последний результат каждого из четырех экспериментов.

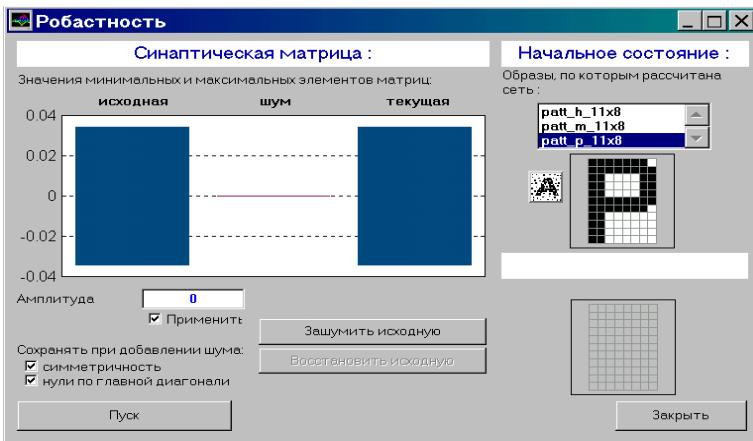


Рис. 5.17. Окно «Робастность»

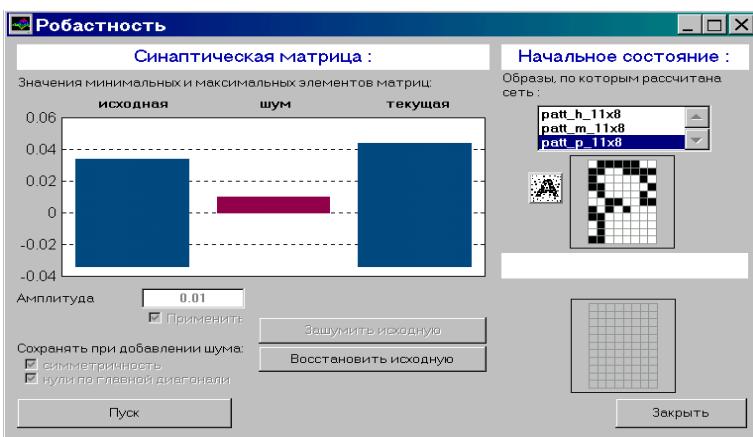


Рис. 5.18. Окно «Робастность». Начальные условия эксперимента

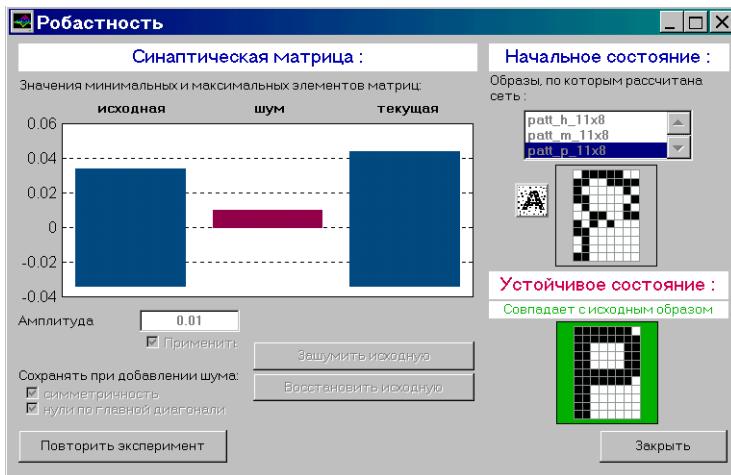


Рис. 5.19. Окно «Робастность». Правильное распознавание

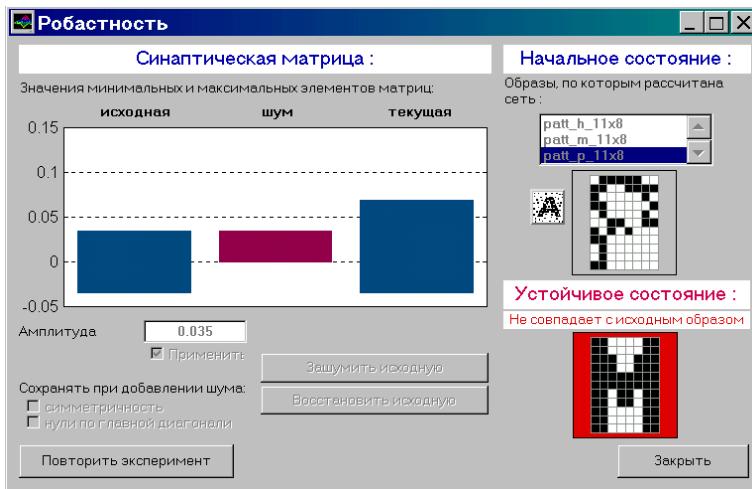


Рис. 5.20. Окно «Робастность». Неправильное распознавание

5.4. Содержание лабораторных исследований

Перед началом исследований студент запускает программу и зарисовывает образы, соответствующие варианту, в отчет. Группа экспериментов 1 – 4, описанных ниже, выполняется последова-

тельно, сначала для сети, настроенной на один образец, затем для сети, настроенной на три и семь образцов.

5.4.1. Исследование динамики сети Хопфилда, настроенной на один образец (эксперимент 1)

Цель эксперимента – определение средней длительности процесса эволюции сети Хопфилда к устойчивому состоянию, длительности переходного процесса в зависимости от числа различий между входным вектором сети и устойчивым состоянием, анализ свойств энергетического функционала.

На вход сети в качестве начального состояния нейронов подается некоторый образ. После этого запускается функционирование сети в соответствии с уравнениями динамики модели Хопфилда (в асинхронном режиме). При этом за один такт (цикл) *последовательно* в случайном порядке вычисляется новое состояние каждого из N нейронов сети. Порядок обработки нейронов выбирается перед каждым тактом с помощью генератора случайных чисел по равномерному закону. После каждого такта выполняется проверка идентичности достигнутого состояния сети состоянию сети после предыдущего такта. Когда эта проверка устанавливает факт достижения устойчивого состояния, процесс останавливается.

Эволюция сети сопровождается изменением энергетического функционала. Текущее значение энергии вычисляется для начального состояния сети и далее пересчитывается после обработки каждого нейрона. По результатам вычислений автоматически строится график зависимости энергии сети от времени асинхронного функционирования.

В начале проведения эксперимента необходимо указать, на какой образец будет настроена сеть. Для этого в главном окне программы (см. рис. 5.4) необходимо нажать кнопку «Сеть». В появившемся списке образов выбрать образ, указанный в варианте. После выбора образа становится активной кнопка «Динамика» в главном окне программы, при нажатии на которую появляется окно «Динамика» (см. рис. 5.10).

При выполнении экспериментов в качестве начального состояния сети необходимо последовательно выбирать позитив, негатив и случайный вектор. Указание случайного вектора производится пу-

тем выбора опции «Случайный вектор», указание негатива проводится вручную.

5.4.2. Исследование бассейнов аттракторов сети Хопфилда, настроенной на один образец (эксперимент 2)

В данном эксперименте исследуются устойчивые состояния сети Хопфилда, определяется их тип (истинная/ложная память), оцениваются их количество и размеры бассейнов.

Число всех возможных состояний сети Хопфилда, содержащей N нейронов, равно 2^N . Некоторые из них являются устойчивыми, причем множество аттракторов могут составлять как те образы, по которым рассчитана синаптическая матрица сети, так и другие состояния. Каждый аттрактор характеризуется *размером бассейна*. Бассейном аттрактора называется множество таких состояний сети, при подаче любого из которых на вход сеть Хопфилда в результате динамического процесса сходится к этому аттрактору. Узнать точное число аттракторов сети и размеры соответствующих бассейнов можно, только перебрав множество всех возможных состояний сети (подавать каждое в качестве начального состояния и фиксировать аттрактор – достигнутое конечное состояние). Такой полный перебор потребовал бы слишком много времени на вычисления.

В лабораторном исследовании используется метод поиска аттракторов и оценивания относительного размера бассейнов, основанный на случайном переборе вариантов. Выбирается заданное число начальных состояний (случайным образом), и из каждого из них осуществляется релаксация к аттрактору. По результатам определяется число различных найденных аттракторов и относительное число попаданий в каждый из аттракторов (от общего числа начальных состояний), которое является оценкой относительного размера бассейна. Аттракторы, характеризующиеся большим значением относительного размера бассейна, считаются «более мощными» по сравнению с аттракторами, имеющими меньший бассейн («более слабые»).

Известно, что когда сеть Хопфилда настроена более чем на один образ, объем памяти сети, в том числе ложной, зависит от того, насколько коррелированы образы, по которым рассчитана сеть. Поэтому для удобства анализа результатов поиска аттракторов автоматически вычисляются коэффициенты корреляции образов.

В начале проведения эксперимента необходимо указать количество запусков сети Хопфилда из случайных состояний, на основе которых будут оцениваться объемы бассейнов аттракторов. По умолчанию в программе заложено 2000 запусков. После этого нажать на кнопку «Поиск аттракторов» (см. рис. 5.12).

5.4.3. Исследование ассоциативной памяти сети Хопфилда, настроенной на один образец (эксперимент 3)

В данном эксперименте определяется средний уровень искажения образа, подаваемого на вход сети, допускающий правильное восстановление исходного образа на выходе. Под искажением понимается инвертирование элементов вектора (замена -1 на 1 или наоборот), мерой искажения является расстояние Хемминга (число отличающихся элементов в исходном и искаженном векторах). Поскольку для ассоциативного восстановления разных векторов, по которым рассчитана сеть, допустимый уровень шума может быть существенно различным, требуется определить среднее допустимое искажение для образа, являющегося мощным аттрактором, и для образа – слабого аттрактора.

Исследуемым образом в данном эксперименте является тот образ, на который настроена сеть Хопфилда (указан слева в окне «Ассоциативная память», см. рис. 5.14). В начале эксперимента в качестве начального состояния сети указать образ, в частности равный исследуемому образу. Убедиться, что расстояние по Хеммингу между этими образами равно нулю, а аттрактором сети является позитив (выводится справа в окне «Ассоциативная память»). Далее, постепенно внося искажения в начальное состояние, увеличивая тем самым расстояние по Хеммингу между исследуемым образом и вектором начального сети, найти такое начальное состояние сети, стартуя из которого сеть все еще приходит к исследуемому образу, а при увеличении этого расстояния на 1 – к любому другому образу, например негативу.

5.4.4. Исследование робастности сети Хопфилда, настроенной на один образец (эксперимент 4)

В эксперименте проверяется свойство робастности сети и оценивается максимальный уровень искажения синаптической матрицы

цы (максимальная амплитуда шума), допускающий правильную работу ассоциативной памяти. Для определенности под правильной работой ассоциативной памяти в рамках данного эксперимента будем понимать способность сети правильно восстанавливать записанный образ с приблизительно 50 % искажений от найденного максимально допустимого уровня в предыдущем эксперименте.

Искажение синаптической матрицы сети состоит в данном эксперименте во внесении аддитивной помехи ξ :

$$W = W + \xi,$$

где W – матрица коэффициентов связей (размерности $N \times N$); ξ_{ij} – число, распределенное по равномерному закону на отрезке $[0, A]$, $i, j = 1, 2, \dots, N$. В таком случае уровень искажения синаптической матрицы характеризуется параметром A , который далее в лабораторной работе будем называть *амплитудой* шума. При таких условиях в результате добавления помехи нарушается симметричность синаптической матрицы, и не выполняется условие $w_{ii} = 0$, $i = 1, 2, \dots, N$ (равенство нулю элементов матрицы связей по главной диагонали). Программное обеспечение позволяет искусственно поддерживать выполнение этих условий одновременно или по отдельности.

Поскольку уровень аддитивной помехи задается ее амплитудой, в программе при визуализации искажения синаптической матрицы для общности матрица характеризуется так же, как и шум, своими минимальным и максимальным элементами. Разумеется, приведенная форма задания шума и условий эксперимента являются одними из возможных для наблюдения проявлений свойства робастности.

Для зашумления матрицы синаптических коэффициентов в поле «Амплитуда» окна «Робастность» (см. рис. 5.17) необходимо указать амплитуду шума A . В качестве начального состояния сети выбирается образ, в точности равный тому, на который сеть была изначально настроена.

Эксперимент предполагает выполнение четырех исследований, соответствующих различным способам зашумления матрицы синаптических коэффициентов: зашумление с сохранением симметрии матрицы и нулей на главной диагонали, зашумление с сохранением симметрии матрицы, зашумление с сохранением нулей на главной диагонали, зашумление всех коэффициентов матрицы независимо друг от друга. В каждом исследовании необходимо найти такой уровень шума, при котором аттрактором сети все еще явля-

ется образ, на который она была настроена, а при увеличении уровня шума на 0,01 – уже другой образ.

5.5. Рекомендации к оформлению отчета

Отчет по работе должен содержать описания исходных данных и условий проведения каждого эксперимента, а также полученные результаты в числовой, графической или табличной форме и выводы.

Ниже приводится содержание отчета с краткими комментариями. Пункты отчета соответствуют экспериментам 1 – 4, описанным в разд. 5.4, в которых исследуется сеть Хопфилда, настроенная на один образец. При исследовании сети Хопфилда, настроенной на три или семь образцов, рекомендации к оформлению отчета аналогичны приведенным ниже.

1. Исследование динамики сети Хопфилда

Приводится образец, на который должна быть настроена сеть. По результатам запусков сети Хопфилда из трех различных начальных состояний (позитива, негатива и случайного состояния) указывается тип аттрактора, к которому приходит сеть (позитив, негатив или ложная память). Фиксируется число тактов, за которое сеть приходит к аттрактору, начальное и конечное значения энергии сети. На основе графика изменения энергии сети в процессе функционирования проверяется теорема о конечности переходного процесса сети.

2. Исследование бассейна аттракторов сети Хопфилда

Указывается число случайных состояний, из которых будет запускаться сеть, и общее число состояний сети. После нахождения аттракторов с использованием нейроэмулатора фиксируется их число, размеры бассейнов позитива, негатива и ложной памяти. Сравниваются полученные результаты с теоретическими. На основе полученных результатов делается вывод о размере ложной памяти.

3. Исследование ассоциативной памяти, реализованной сетью Хопфилда

Указывается исследуемый образ и фиксируется максимальное расстояние по Хеммингу между исследуемым образом и вектором начального состояния сети, стартую из которого аттрактором сети все еще является исследуемый образ. Указывается аттрактор, к которому приходит сеть при увеличении этого расстояния на 1.

4. Исследование робастности сети Хопфилда

Приводится входной вектор сети, используемый в эксперименте. Фиксируется максимальный уровень шума, при котором сеть Хопфилда правильно восстанавливает аттрактор, соответствующий входному вектору. Указывается тип аттрактора, к которому приходит сеть при малом увеличении уровня шума.

Рассматриваются следующие постановки экспериментов:

- зашумляется матрица синаптических коэффициентов сети с сохранением симметричности и нулей на главной диагонали;
- зашумляется матрица синаптических коэффициентов сети с сохранением симметричности;
- зашумляется матрица синаптических коэффициентов сети с сохранением нулей на главной диагонали;
- матрица синаптических коэффициентов зашумляется произвольным образом.

5. Общие выводы по проведенным исследованиям

Излагаются свойства сети Хопфилда, приводятся особенности ее функционирования, наблюдаемые в процессе проведения исследований. Делается вывод о способности сети Хопфилда выступать в качестве модели ассоциативной памяти.

Контрольные вопросы

1. Что называется автоассоциативной памятью? Приведите пример.
2. Что называется аттрактором динамической системы? Объясните принцип применения динамических систем с множеством аттракторов для построения ассоциативной памяти.

3. Напишите уравнения динамики сети Хопфилда.
4. Какова активационная характеристика нейронов сети Хопфилда?
5. Чему равно начальное состояние нейронов сети Хопфилда?
6. Объясните различие синхронного и асинхронного режимов функционирования рекуррентной нейронной сети. Какой из режимов функционирования используется в сети Хопфилда?
7. Каким выражением определен энергетический функционал в процессе работы сети Хопфилда?
8. Почему время достижения сетью Хопфилда одного из атTRACTоров из произвольного начального состояния конечно?
9. Как рассчитывается матрица синаптических коэффициентов сети Хопфилда? Какими свойствами она обладает?
10. Как приближенно оценивается объем памяти сети Хопфилда?
11. На какие типы можно разделить множество атTRACTоров сети Хопфилда? Что такое «ложная память»?
12. Как применяется сеть Хопфилда для кластеризации данных? Как интерпретируются кластеры?
13. Как устанавливается класс принадлежности вектора признаков при использовании нескольких (по числу классов) сетей Хопфилда, «настроенных» на разные классы?
14. Как организован в лабораторной работе процесс оценивания размеров бассейнов атTRACTоров сети Хопфилда?

Лабораторная работа 6

САМООБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ. КЛАСТЕРИЗАЦИЯ ДАННЫХ

6.1. Цель и задачи лабораторной работы

Цель: освоение теоретического принципа кластеризации данных на самообучающейся нейронной сети и анализ практических возможностей использования этого принципа на модельных примерах.

В процессе анализа больших информационных массивов данных, характеризующих новую предметную область, неизменно возникают задачи исследования топологической структуры данных, их группирования, способов визуализации. Это могут быть экономические, финансовые, научно-технические, медицинские и любые иные данные, внутренняя структура которых неизвестна и подлежит исследованию. Знание топологической структуры данных необходимо для решения практических задач сжатия данных, выбора способа хранения и поиска данных, прогноза характеристик объекта по ограниченному набору признаков.

Задачи рассматриваемого класса могут быть успешно решены с применением самоорганизующихся нейронных сетей, предложенных финским ученым Т. Кохоненом. В лабораторной работе исследуется соревновательный слой нейронов, изучаются способы его самонастройки и анализируется качество решения задачи кластеризации данных. Исследуется топографическая карта Кохонена, позволяющая построить удобную для визуального анализа многомерных данных графическую модель в плоскости. В ходе выполнения лабораторной работы студент ознакомится с демонстрационными примерами решения задач анализа данных с применением самообучающихся нейронных сетей.

Студентам предоставляется возможность варьировать пространственное распределение данных, значения параметров алгоритма обучения сети Кохонена и анализировать результат кластеризации с помощью графических средств визуализации.

6.2. Варианты индивидуальных заданий

Номер	Количество эллипсов рассеяния	Количество точек в кластере	Внутрикластерная дисперсия
1	5	4	0,1
2	4	5	0,08
3	3	15	0,12
4	6	10	0,2
5	6	6	0,16
6	4	5	0,14
7	5	9	0,05
8	3	7	0,15
9	7	5	0,11
10	5	10	0,22

6.3. Содержание лабораторных исследований

Лабораторная работа выполняется с применением пакета прикладных программ «Нейронные сети» в программной среде MATLAB и включает следующие исследования:

- анализ режима самообучения соревновательного слоя нейронов и качества кластеризации двумерных данных;
- изучение на демонстрационном примере особенностей кластеризации одномерных данных с неравномерной плотностью распределения точек на прямой;
- ознакомление с демонстрационным примером построения самоорганизующейся карты Кохонена;
- построение одномерной карты Кохонена для совокупности точек в плоскости и анализ качества полученного отображения.

Каждая из перечисленных задач поддерживается специальной программой, которая запускается из командной строки системы MATLAB. Кроме графического представления выборочных данных и результатов расчетов, для анализа привлекаются числовые данные, выводимые на экран в процессе интерпретации программных операторов в командном окне MATLAB.

6.3.1. Кластеризация двумерных данных (эксперимент 1)

Содержание эксперимента. В данном эксперименте исследуется качество кластеризации двумерных данных с помощью сети Кохонена. Все обучающие данные расположены в фиксированном

прямоугольнике с заданными диапазонами изменения координат x и y : $x \in [0,5; 1,5]$, $y \in [-1; 5]$. В этом прямоугольнике обучающие данные генерируются случайным образом из нормально распределенной на плоскости генеральной совокупности. Центры генерируемых кластеров также случайным образом располагаются в пределах прямоугольника, а степень рассеяния точек внутри кластера около центра характеризуется параметром «стандартное отклонение», указанным в варианте задания к выполнению лабораторной работы. Цель эксперимента состоит в построении и обучении соревновательного слоя нейронов, с помощью которого автоматически решается задача кластеризации данных. При этом число нейронов соревновательного слоя необязательно совпадает с числом кластеров в выборке.

На каждом такте самообучения на вход сети подается очередной пример обучающей выборки, определяется нейрон-победитель и осуществляется подстройка его синаптических коэффициентов. Визуально изменение синаптических коэффициентов нейрона можно представить как перемещение нейрона-победителя вдоль прямой, соединяющей его координаты с координатами обучающего примера.

Процесс самообучения сети характеризуется параметром скорости самообучения λ . Обычно начальное значение λ устанавливается равным $0,7 - 0,8$ и уменьшается в процессе самообучения до $0 - 0,2$. В программе, используемой в лабораторной работе, реализован один из возможных критериев останова процедуры самообучения – останов по заданному числу эпох.

После завершения процедуры самообучения построенная сеть тестируется. Для этого на ее вход подаются данные – координаты x, y точек тестовой выборки. Каждая из точек «захватывается» одним из нейронов обученного соревновательного слоя, так что ей сопоставляется номер этого нейрона (кластера).

Указания к программному выполнению эксперимента. Для проведения лабораторного исследования при решении задачи кластеризации двумерных данных используется программа `task1.m`, для запуска которой в командной строке MATLAB необходимо набрать

`task1`

Первым шагом программы `task1.m` является генерация в плоскости x, y нескольких групп точек (кластеров). Для генерации выборочных точек требуется указать следующие параметры:

`clusters` – число генерируемых кластеров;

`points` – число точек в каждом кластере;

`std_dev` – стандартное отклонение точек в кластере.

В программе `task1.m` для числа нейронов используется идентификатор `nneur`. Группа параметров `clusters`, `points`, `std_dev` и `nneur` содержится в текстовом файле `par.dat` и редактируется при выполнении задания. В табл. 6.1 приведен пример значений параметров этой группы.

Таблица 6.1

Значения параметров в файле `par.dat`

<code>clusters</code>	<code>points</code>	<code>std dev</code>	<code>nneur</code>
5	10	0,5	8

Следующая группа параметров определяет режим самообучения и его визуализацию. В состав группы входят параметры:

`df` – число «эпох», через которое обновляются данные на графике в процессе самообучения;

`me` – число «эпох» самообучения;

`lr` – параметр скорости самообучения.

Перечисленные параметры располагаются в текстовом файле `tp.dat` и редактируются при проведении лабораторных исследований. В табл. 6.2 приведен пример значений параметров этой группы.

Таблица 6.2

Значения параметров в файле `tp.dat`

<code>df</code>	<code>me</code>	<code>lr</code>
20	500	0,1

Заметим, что начальное значение синаптических коэффициентов всех нейронов, устанавливаемое процедурой инициализации, одинаково. Это значение соответствует центру тяжести выборки. Именно с этим обстоятельством связано то, что перед началом самообучения все белые кружочки, характеризующие вектора синаптических коэффициентов нейронов, на графике сливаются в один.

В ходе выполнения данного эксперимента необходимо подобрать значения параметра скорости самообучения и числа эпох, достаточных для успешной кластеризации данных

6.3.2. Негативные эффекты процедуры самообучения сети Кохонена (эксперимент 2)

Содержание эксперимента. В данном эксперименте исследуются особенности применения сети Кохонена для кластеризации данных. Проводится анализ качества кластеризации данных в зависимости от числа нейронов сети. Так, в случае неточного выбора числа нейронов сети, отличающегося от предполагаемого числа кластеров, в результате кластеризации, как правило, возникают негативные эффекты. При выборе числа нейронов слоя Кохонена меньшем, чем фактическое число кластеров, образуемых данными, возникает явление, называемое укрупнением кластеров. В этом случае один нейрон «притягивает» данные нескольких кластеров. В противном случае, когда число нейронов слоя Кохонена больше фактического числа кластеров, возникает детализация кластеров. В этом случае в кластере данных располагается несколько нейронов.

Другой негативный эффект состоит в появлении «мертвых» нейронов. Такие нейроны остаются неподвижными в течение всего времени обучения. В программном комплексе, используемом в лабораторной работе, реализован алгоритм устранения «мертвых» нейронов, основанный на поощрении или наказании нейронов сети. В связи с этим возникновение «мертвых» нейронов в процессе обучения крайне маловероятно и может наблюдаться лишь при небольших значениях шага обучения и числа эпох.

При выборе слишком большого значения параметра скорости обучения может возникать колебательный характер процедуры обучения, особенно заметно проявляющийся в случае укрупненных кластеров. При уменьшении в процессе самообучения параметра скорости колебательный эффект становится все менее заметным.

Указания к программному выполнению эксперимента. Для проведения исследования также используется программа task1.m, для запуска которой в командной строке MATLAB необходимо набрать

```
task1
```

В данном эксперименте требуется подобрать такие значения числа нейронов сети, начального значения параметра скорости самообучения и числа эпох, при которых наблюдаются перечисленные выше негативные эффекты – укрупнение и детализация кластеров, возникновение «мертвых» нейронов и колебательный характер процедуры самообучения. Перечисленные параметры редактируются в текстовых файлах par.dat и tp.dat.

Параметры расположения обучающих данных такие же, как и в предыдущем исследовании.

6.3.3. Проверка качества самообучения сети Кохонена (эксперимент 3)

Содержание эксперимента. В данном эксперименте осуществляется прогон через обученную сеть контрольной выборки примеров. При предъявлении сети каждого из примеров контрольной выборки происходит его отнесение к одному из сформированных сетью кластеров.

Указания к программному выполнению эксперимента. Даные контрольной выборки располагаются в текстовом файле test.dat. Они оформляются и вводятся в программу как матрица размерности $[2, N_{test}]$, где N_{test} – объем контрольной выборки. Первая строка содержит значения координаты x , а вторая – y . В табл. 6.3 приведен пример контрольной выборки из пяти точек.

Таблица 6.3

Пример контрольной выборки

0,5	0,1	1,5	0,75	1,1
3,2	-0,5	4,2	2	3,8

Результат отнесения контрольных точек к одному из кластеров анализируется визуально в графическом окне, на котором контрольные точки отмечаются зелеными крестиками, а центр соответствующего кластера (значения настроенных в результате самообучения синаптических коэффициентов «победившего» нейрона) – зелеными кружочками.

После завершения процедуры самообучения в командном окне системы MATLAB визуализируется также матрица результатов работы сети. Эта матрица имеет N_{test} столбцов по числу примеров контрольной выборки и пять строк, в которых располагаются координаты x, y контрольной точки, номера кластеров («победивших» нейронов соревновательного слоя) n_{class} и центры кластеров w_x, w_y . В табл. 6.4 приведен пример подобной матрицы:

Таблица 6.4

Матрица результатов работы сети

x	0,5	0,1	1,5	0,75	1,1
y	3,2	-0,5	4,2	2	3,8
nclass	2	5	4	1	4
wx	0,6	0,25	1,22	0,65	1,22
wy	2,9	-0,43	4,05	1,8	4,05

6.3.4. Кластеризация одномерных данных (эксперимент 4)

Содержание эксперимента. В данном эксперименте исследуется качество кластеризации одномерных данных с помощью сети Кохонена. Обучающие данные расположены на отрезке $[0; 1]$ с неравномерной плотностью. Плотность точек вблизи нуля максимальна и убывает по направлению к единице. Для кластеризации используется соревновательный слой из шести нейронов.

После 3000 «эпох» обучения с параметром скорости обучения 0,02 оказывается, что центры образовавшихся кластеров (координаты точек, соответствующих синаптическим коэффициентам обученных нейронов) также располагаются вдоль оси неравномерно, но таким образом, что все они «притягивают» примерно равное число выборочных точек. Это означает, что если случайным образом на оси x генерируется точка, закон распределения которой соответствует распределению обучающей выборки, то примерно с

равной вероятностью может «победить» любой из нейронов слоя Кохонена.

Указания к программному выполнению эксперимента. Для проведения исследования используется программа task2.m, для запуска которой в командной строке MATLAB необходимо набрать

```
task2
```

Сгенерированные точки располагаются вдоль горизонтальной прямой с неравномерной плотностью. В начале самообучения все нейроны располагаются в одной точке. В процессе самообучения на графике осуществляется визуальное отображение движения нейронов вдоль прямой. По горизонтальной оси отложены координаты обучающих примеров и нейронов, по вертикальной – номер эпохи самообучения. Обновление графика осуществляется через фиксированное число тактов самообучения.

6.3.5. Самообучение двумерной карты Кохонена на двумерных данных (эксперимент 5)

Содержание эксперимента. Для проведения исследований используются двумерные данные, равномерно распределенные на плоскости в прямоугольнике. Нейроны топографической карты находятся в узлах прямоугольной сетки.

Как указывалось ранее, каждому нейрону соответствует вектор, составленный из его синаптических коэффициентов, который может быть отмечен точкой в плоскости с данными. Таким образом, сетка нейронов отображается на плоскость данных. В процессе обучения узлы сетки перемещаются в соответствии с динамикой настройки синаптических коэффициентов, что и наблюдается на графике. Перед началом обучения сетка сжата в точку, так как синаптические коэффициенты всех нейронов полагаются равными и соответствуют положению центра тяжести выборочных данных.

На заключительном графике видно, что узлы топографической карты (центры кластеров) равномерно накрывают выборочные данные, а близким кластерам данных соответствуют близко расположенные на сетке нейроны.

Указания к программному выполнению эксперимента. Для проведения исследования используется программа `task3.m`, для запуска которой в командной строке MATLAB необходимо набрать

```
task3
```

Сгенерированные точки случайным образом располагаются в прямоугольной области на плоскости. В начале самообучения все нейроны (их образы в пространстве признаков, представленные векторами синаптических коэффициентов) располагаются в одной точке. В процессе самообучения на графике осуществляется визуальное отображение движения векторов синаптических коэффициентов нейронов в плоскости. Соседние на карте нейроны отображаются соединением отрезками.

6.3.6. Самообучение одномерной карты Кохонена на двумерных данных (эксперимент 6)

Содержание эксперимента. Для проведения исследований используются двумерные данные, распределенные равномерно вдоль дуги единичной окружности в угловом растворе от 0 до $\pi/2$. Нейроны на одномерной карте Кохонена упорядочены вдоль прямой, а расстояние между ними рассчитывается как модуль разности между их номерами.

Каждому нейрону соответствует двумерный вектор, составленный из его синаптических коэффициентов, который может быть отмечен точкой в плоскости с данными. Соседние нейроны на карте визуально отображаются как соединенные отрезком. После окончания процедуры самообучения видно, что узлы топографической карты (центры кластеров) равномерно накрывают выборочные данные, а близким кластерам данных соответствуют близко расположенные на сетке нейроны.

Указания к программному выполнению эксперимента. Для проведения исследования используется программа `task4.m`, для запуска которой в командной строке MATLAB необходимо набрать

```
task4
```

В начале самообучения все нейроны располагаются в одной точке. В процессе самообучения на графике осуществляется визу-

альное отображение движения нейронов в плоскости и отношения соседства между нейронами на карте.

Процессом самообучения управляют те же параметры, которые упоминались ранее: df , me , lr . Значения этих параметров, а также числа нейронов $nneur$ в одномерной карте Кохонена указываются в файле `p_task4.dat`. Для выполнения эксперимента необходимы данные, пример которых представлен в табл. 6.5.

Таблица 6.5

Значения параметров в файле `tp.dat`

df	me	lr	$nneur$
50	500	1	10

Заметим, что указанное в таблице значение параметра lr является начальным, которое убывает в процессе обучения. Фактические значения параметра lr выводятся на экран в командном окне MATLAB.

6.4. Рекомендации к оформлению отчета

Отчет по работе должен содержать описания исходных данных и условий проведения каждого эксперимента, а также полученные результаты в числовой, графической или табличной форме и выводы.

Ниже приводится содержание отчета с краткими комментариями.

1. Кластеризация двумерных данных

Приводятся исходные данные для выполнения исследования: количество эллипсов рассеяния, количество точек в каждом эллипсе и значение внутрикластерной дисперсии. Приводится архитектура сети Кохонена, указывается число входов, число выходов сети, число нейронов в слое Кохонена и тип активационной характеристики нейронов. Указывается начальное значение параметра скорости самообучения.

После завершения процедуры самообучения сети на графике изображается расположение исходных данных и центров кластеров. В отдельной таблице указываются координаты найденных центров

кластеров. По результатам кластеризации делается вывод о соответствии близких по номерам нейронов сети близким кластерам.

2. Негативные эффекты процедуры самообучения сети Кохонена

Приводятся число нейронов сети Кохонена, начальное значение параметра скорости самообучения и число эпох самообучения, при которых наблюдаются следующие негативные эффекты процедуры самообучения сети:

- укрупнение кластеров;
- детализация кластеров;
- возникновение «мертвых» нейронов;
- колебательный характер процедуры обучения.

Для каждого из эффектов на графике изображается визуальное представление исходных данных, кластеры данных и их центры, определяемые по результатам обучения сети. По результатам исследования сделать выводы о том, при каких значениях числа нейронов, параметра скорости самообучения и числа эпох самообучения наблюдаются перечисленные негативные эффекты, а также, какие параметры влияют на появление этих эффектов.

3. Проверка качества самообучения сети Кохонена

Для тестирования сети произвольно выбираются 5 – 7 точек, расположенных в рабочей области и визуально относящихся к разным кластерам. Указать координаты этих точек, а также номера кластеров, к которым их относит обученная сеть Кохонена. Выбранные тестовые точки изображаются на графике, построенном в результате выполнения эксперимента 1.

4. Кластеризация одномерных данных

Приводится архитектура сети Кохонена, указываются число входов, число выходов сети, число нейронов в слое Кохонена. На графиках изображаются расположение исходных данных и функция плотности их распределения. На графике с исходными данными изображается график зависимости синаптических коэффициентов нейронов сети от номера эпохи обучения.

5. Самообучение двумерной карты Кохонена на двумерных данных

Приводится архитектура сети Кохонена, указываются число входов, число выходов сети, число нейронов в слое Кохонена. После завершения процедуры самообучения карты Кохонена на графике изображается расположение исходных данных и центров кластеров. На отдельном графике изображается карта Кохонена. По результатам кластеризации делается вывод о соответствии близких нейронов на карте близким кластерам.

6. Самообучение одномерной карты Кохонена на двумерных данных

Приводится архитектура сети Кохонена, указываются число входов, число выходов сети, число нейронов в слое Кохонена. Указываются число эпох самообучения и начальное значение параметра скорости самообучения. После завершения процедуры самообучения карты Кохонена на графике изображается расположение исходных данных и центров кластеров *с указанием номеров нейронов*. На отдельном графике изображается карта Кохонена. По результатам кластеризации делается вывод о соответствии близких нейронов на карте близким кластерам.

7. Общие выводы по проведенным исследованиям

Излагаются особенности решения задач кластеризации с использованием сети Кохонена и карты Кохонена. Анализируются условия возникновения негативных эффектов при кластеризации и способы их устранения. Проводится сравнение качества кластеризации с использованием сети Кохонена и карты Кохонена.

Контрольные вопросы

1. В чем состоит проблема кластеризации данных? Какие выборочные данные необходимы для решения этой задачи?
2. Где в соревновательном слое нейронов хранится информация о прототипах кластеров?
3. Дайте описание работы нейронов соревновательного слоя.

4. В чем состоит избирательность нейронов соревновательного слоя по отношению к данным обучающей выборки?
5. Как оценивается расстояние между выборочным примером и вектором синаптических коэффициентов нейрона?
6. Почему именно потенциалы нейронов соревновательного слоя используются для выявления нейрона-победителя?
7. Объясните правило самообучения нейронов соревновательного слоя.
8. Что называется «эпохой» самообучения соревновательного слоя нейронов?
9. Как влияет выбор параметра скорости самообучения на качество процесса настройки синаптических коэффициентов? Какие практические рекомендации можно дать для выбора этого параметра?
10. Что такое «соревновательная» активационная характеристика?
11. Какие негативные особенности можно ожидать при реализации самообучения соревновательного слоя нейронов?
12. Каким способом можно избежать появления «мертвых» нейронов соревновательного слоя?
13. По каким причинам может возникать укрупнение кластеров при обучении соревновательного слоя нейронов?
14. Как практически можно оценить качество обучения соревновательного слоя нейронов? Каким свойством обладает оптимально обученный соревновательный слой?
15. Какое основное качество отличает карту Кохонена от соревновательного слоя нейронов?
16. Приведите примеры оценки расстояния между нейронами в топографической карте. Какими в этих примерах являются линии точек, равноудаленных от фиксированного нейрона?
17. В чем состоит правило обучения нейронов карты Кохонена? Каково его отличие от правила обучения нейронов соревновательного слоя?
18. В чем состоит «раскраска» топографической карты отдельными признаками?
19. Что представляет собой одномерная карта Кохонена?
20. Какой вы можете привести пример практического применения карты Кохонена?

Рекомендуемая литература

1. Хайкин С. Нейронные сети: Полный курс. – М.: Издательский дом «Вильямс», 2006.
2. Розенблatt Ф. Принципы нейродинамики. Перцепtron и теория механизмов мозга. – М.: Мир, 1965.
3. Минский М., Пейперт С. Перцептроны. – М.: Мир, 1971.
4. Ежов А.А., Шумской С.А. Нейрокомпьютинг и его приложения в экономике и бизнесе. – М.: МИФИ, 1998.
5. Галушкин А.И. Теория нейронных сетей. – М.: ИПРЖР, 2000.
6. Аксенов С.В. Новосельцев В.Б. Организация и использование нейронных сетей. Методы и технологии. – Томск: НТЛ, 2006.
7. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002.
8. Горбань А.Н. Обучение нейронных сетей. – М.: СП «Параграф», 1990.
9. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996.
10. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука, 1998.
11. Тархов Д.А. Нейронные сети. Модели и алгоритмы. – М.: Радиотехника, 2005.
12. Бэстенс Д.-Э., Ван Ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки. – М.: ТВП, 1997.
13. Головко В.А. Нейронные сети: обучение, организация и применение. – М.: ИПРЖР, 2001.
14. Назаров А.В., Лоскутов А.Н. Нейросетевые алгоритмы прогнозирования и оптимизации систем. – С.-Пб.: Наука и техника, 2003.
15. Радченко А.Н. Ассоциативная память. Нейронные сети. Оптимизация нейропроцессоров. – С.-Пб.: Наука, 1998.