МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

П.Г. КРУГ

НЕЙРОННЫЕ СЕТИ И НЕЙРОКОМПЬЮТЕРЫ

Учебное пособие
по курсу «Микропроцессоры»
для студентов, обучающихся по направлению
«Информатика и вычислительная техника»

МОСКВА ИЗДАТЕЛЬСТВО МЭИ 2002

УДК 621.398
К 84
УДК 621.398.724(072)

Утверждено учебным управлением МЭИ в качестве учебного пособия
Рецензенты:
проф., д-р. техн. наук Желбаков И.Н., проф., д-р. техн. наук Петров О.М.
Подготовлено на кафедре Информационно-измерительной техники

Круг П.Г.
Нейронные сети и нейрокомпьютеры: Учебное пособие по курсу «Микропроцессоры». – М.: Издательство МЭИ, 2002. – 176 с.
ISBN 5-7046-0832-9

Рассматриваются основы нейронных сетей, типовые решаемые задачи, области применения, программные продукты для моделирования, а также современные нейропроцессоры и нейрокомпьютеры.

Пособие включает практический курс, базирующийся на использовании программного симулятора TRAJAN, и отражает опыт подготовки российских и иностранных студентов в области искусственных нейронных сетей в Московском энергетическом институте (техническом университете).

Для студентов, обучающихся по направлению «Информатика и вычислительная техника».

ISBN 5-7046-0832-9

СОДЕРЖАНИЕ

СПИСОК АББРЕВИАТУР

АЛУ – арифметическо-логическое устройство
АЦП – аналого-цифровой преобразователь
БВЭ – базовые вычислительные элементы
БПФ – быстрое преобразование Фурье
ВСМП – вычислительные системы с массовым параллелизмом
ДПП – двухпортовая память
МП – микропроцессор
МСП – матричные сигнальные процессоры
ОЗУ (RAM) – оперативное запоминающее устройство
ПДП – прямой доступ к памяти
ПЗУ (ROM) – постоянное запоминающее устройство
ПЛИС – перепрограммируемые логические интегральные схемы
ПК – персональный компьютер
ПЦОС (DSP) – процессор цифровой обработки сигналов (Digital Signal Processor)
СКЗ – среднеквадратическое значение
СППР – система поддержки принятия решений
УП – управляющий процессор
ЦАП – цифроаналоговый преобразователь
ЦОС – цифровая обработка сигналов
ЦП (ЦПУ, CPU) – центральный процессор (центральное процессорное устройство, Cental Processor Unit)
ART – адаптивная резонансная теория (Adaptive Resonance Theory – вид нейронных сетей)
CIL – Chip-in-Loop (этап обучения нейронной сети)
CPLD – устройство со сложной программируемой логикой (Complex Programmable Logic Device)
CPS – число соединений в секунду (Connection per Second)
CUPS – число измененных значений весов соединений в секунду (Connection Update per Second)
DARAM – оперативное запоминающее устройство с двойным доступом
EPROM – постоянное запоминающее устройство с ультрафиолетовым стиранием
FPU – модуль, выполняющий операции с плавающей арифметикой (Floating-point Unit)
INNTS –Intel Neural Network Training System (программный продукт)
I/O – устройства ввода-вывода (Input/Output)
MFLOPS – миллионов операций с плавающей точкой в секунду (Million Floating-Point Operation per Second)
MIMD – вычислительная система с множественным потоком команд и данных (Multiple-Instruction and Multiple-Data)
MIPS – миллионов инструкций в секунду (Million Instruction per Second)
MMAC – миллионов операций умножения с накоплением в секунду
SBSRAM – синхронная пакетная статическая оперативная память
SDRAM – статическая оперативная память с динамическим рандомизированным доступом
SHARC – разновидность гарвардской архитектуры (Super Harvard ARChitecture, товарный знак компании Analog Devices)
SIMD – вычислительная система с одиночным потоком команд и множественный поток данных (Single-Instruction and Multiple-Data)
SNNS – Stuttgart Neural Network Simulator (разработка университета г. Штутгарт, ФРГ)

ВВЕДЕНИЕ

В качестве введения кратко рассмотрим историю создания и развития искусственных нейронных сетей.

Впервые о них заговорили в 1940-х годах. Считается, что теория нейронных сетей, как научное направление, была обозначена в классической работе *Мак Каллока* и *Питтса* в 1943 г., в которой утверждалось, что, в принципе, любую арифметическую или логическую функцию можно реализовать с помощью простой нейронной сети.

Среди основополагающих работ следует выделить модель *Д. Хэбба*, который в 1949 г. предложил закон обучения, явившийся стартовой точкой для алгоритмов обучения искусственных нейронных сетей, а также теоремы *М. Минского* и исследования им ряда типовых задач, в том числе, популярной задачи «Исключающего «ИЛИ» [6].

В 1958 г. *Ф. Розенблатт* предложил нейронную сеть, названную персептроном [7], и построил первый нейрокомпьютер «Марк-1». Персептрон был предназначен для классификации объектов и получал на этапе обучения от «учителя» сообщение, к какому классу принадлежит предъявляемый объект. Обученный персептрон был способен классифицировать объекты, в том числе, не использовавшиеся при обучении, делая при этом очень мало ошибок.

Затем, после разработок 1950-х и 1960-гг. наступил период затишья, длившийся с 1968 по 1985 гг.

В 1985-1986 гг. теория нейронных сетей получила «технологический импульс», который был вызван возможностью моделирования нейронных сетей на появившихся в то время доступных и высокопроизводительных персональных компьютерах [3].

Настольной книгой специалиста, моделирующего и применяющего нейронные сети, стала работа *Ф. Уоссермена* «Нейрокомпьютерная техника» (издана на русском языке в 1992 г.) [8].

В настоящее время, по оценкам специалистов, ожидается значительный технологический рост в области проектирования нейронных сетей и нейрокомпьютеров. За последние годы уже открыто немало новых возможностей нейронных сетей, а работы в данной области становятся важным вкладом в промышленность, науку и технологии, имеют большое экономическое значение [9].

Однако все попытки понять и моделировать объективные процессы обработки информации мозгом человека пока особого успеха не имели. Несмотря на то, что разработки по нейронному моделированию ведутся нейробиологами уже более 50 лет, нет ни одной области мозга, где процесс обработки информации был бы ясен до конца. Также ни для одного нейрона в мозге пока невозможно определить код, который он использует для передачи информации в виде последовательности импульсов.

Грядущий взрывной рост в области нейрокомпьютерных технологий по всей вероятности будет связан с новыми открытиями в области нейронного моделирования – как только мы разгадаем тайну функционирования хотя бы одной области мозга, так сразу, по-видимому, получим представление о работе многих других его областей.

Предполагается, что открытие биологических основ обработки информации вызовет существенную активизацию работ в построении искусственного мозга и инициацию беспрецедентного по своему размаху научного и технологического проекта. По сравнению с ним глобальные проекты прошлого и настоящего, такие как исследование космоса, открытия ядерной физики, молекулярная биология и генная инженерия покажутся весьма скромными. Ожидается, что новый проект будет способен достаточно быстро дать значительный экономический эффект и, наконец-то, появится возможность синтезировать «умные» машины и системы, способные вместо людей выполнять монотонные, скучные и опасные задания [8, 9].

Справедливости ради отметим, что для достижения этих целей, также важно развитие и ряда других областей информатики, микроэлектроники и искусственного интеллекта.

1. ОСНОВНЫЕ НАПРАВЛЕНИЯ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ И НЕЙРОКОМПЬЮТЕРОВ

1.1. Типовые решаемые задачи

Потенциальными областями применения искусственных нейронных сетей являются те, где человеческий интеллект малоэффективен, а традиционные вычисления трудоемки или физически неадекватны (т.е. не отражают или плохо отражают реальные физические процессы и объекты).

Действительно, **актуальность применения нейронных сетей многократно возрастает тогда, когда появляется необходимость решения *плохо формализованных* задач.**

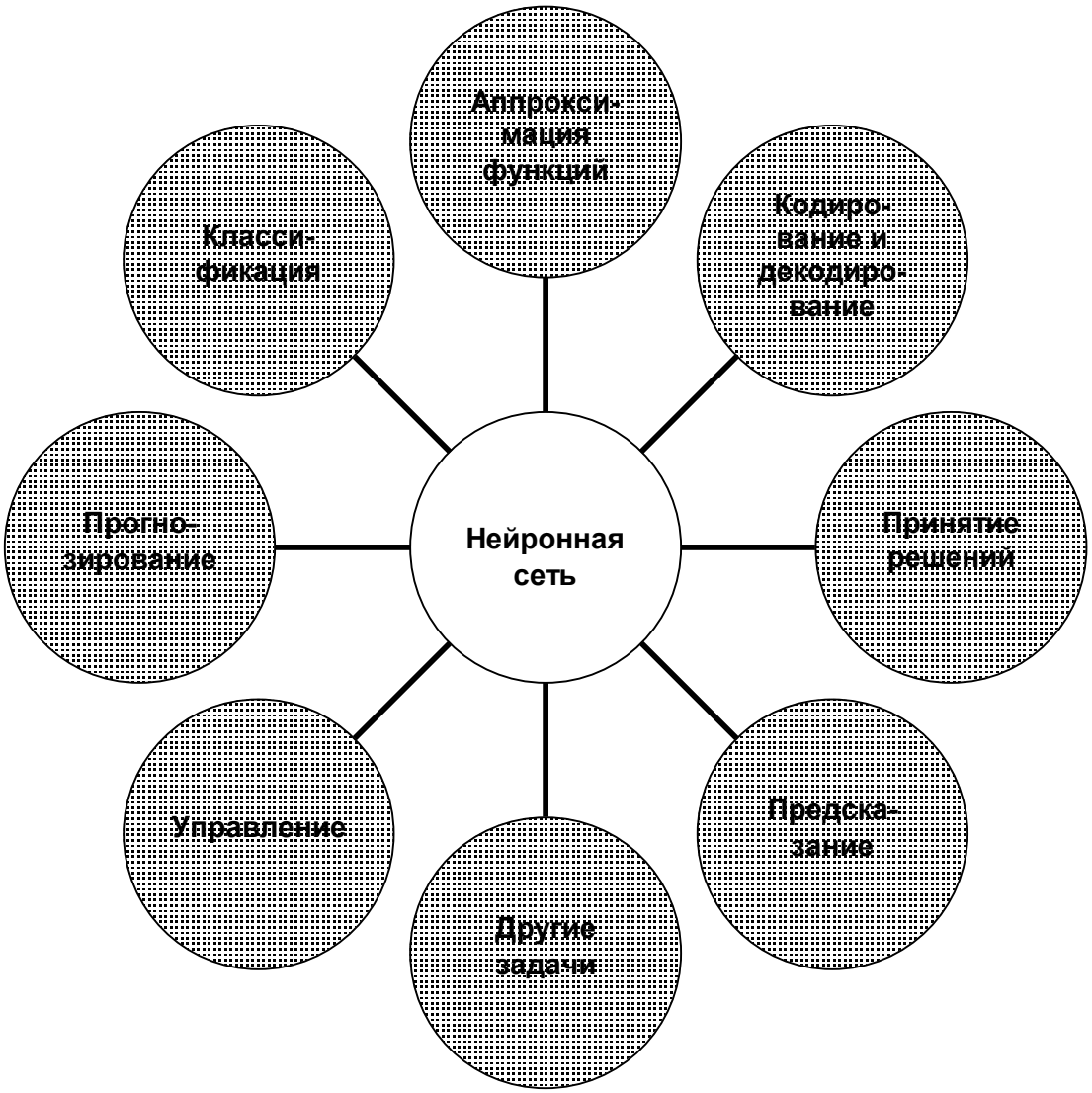


Рис. 1.1. Типовые задачи, решаемые с помощью нейронных сетей и нейрокомпьютеров

Таблица 1.1. Основные области применения нейронных сетей

Промышленность		
Управление технологическими процессами	Идентификация химических компонент	Контроль качества артезианских вод
Оценка экологической обстановки	Прогнозирование свойств синтезируемых полимеров	Управление водными ресурсами
Оптимальное планирование	Разработка нефти и газа	Управление работой пресса
Идентификация вида полимеров	Управление ценами и производством	Оптимизация работы моторов
Обнаружение повреждений	Оптимизация закупок сырья	Контроль качества изделий
Приложения аналитической химии	Анализ проблем функционирования заводов и магазинов	Прогнозирование потребления энергии
Высокие технологии		**Оборона**
Проектирование и оптимизация сетей связи	Идентификация и верификация говорящего	Анализ визуальной аэрокосмической информации
Анализ и сжатие изображений	Видеонаблюдение	Отбор целей
Распознавание печатных и рукописных символов	Автоматизированное распознавание речевых команд	Обнаружение наркотиков и взрывчатых веществ
Фальсификации в пищевой и парфюмерной пром-сти	Распознавание слитной речи с (и без) настройки на говорящего	Сличение изображений с криминальной базой данных
Обслуживание кредитных карт	Речевой ввод текста в компьютер	Предсказание целесообразности условного освобожден.
Наука и техника		**Здравоохранение**
Поиск неисправностей в научных приборах	Спектральный анализ и интерпретация спектров	Идентификация микробов и бактерий
Диагностика печатных плат	Интерпретация показаний сенсоров	Диагностика заболеваний
Идентификация продуктов	Моделирование физических систем	Интерпретация ЭКГ
Синтез новых видов стекла	Анализ данных в ботанике	Анализ качества лекарств
Автоматизированное проектирование	Планирование химических экспериментов	Обработка и анализ медицинских тестов
Оптимизация биологических экспериментов	Отбор сенсоров для контроля химических процессов	Прогнозирование результатов применения методов лечения
Геофизические и сейсмологические исследования	Прогноз температурного режима технологических процессов	Оптимизация атлетической подготовки
Распознавание ингредиентов	Диагностика сбоев сигнализации	Диагностика слуха
Бизнес и финансы		
Выбор сбытовой политики	Прогноз прибыли (Cash-flow)	Прогнозирование продаж
Принятие административных решений	Предсказание и расшивка «узких мест»	Анализ целей маркетинговой политики
Предсказания на фондовой бирже	Прогноз эффективности кредитования	Прогнозирование экономических индикаторов
Анализ финансового рынка	Прогнозирование валютного курса	Анализ страховых исков
Исследование фактора спроса	Прогнозирование и анализ цен	Отбор перспективных кадров
Моделирование бизнес-стратегии	Построение макро- и микроэкономических моделей	Стратегии в области юриспруденции
Предсказание наступления финансовых кризисов	Предсказание необходимых трудодней для реализации проекта	Оценка и прогнозирование стоимости недвижимости

Нейронные сети продемонстрировали свою способность решать сложные задачи. Они имеют уникальные потенциальные возможности, хотя не свободны от ограничений и вопросов, на которые до сих пор не существует ответа. Такая ситуация настраивает на умеренный оптимизм [8].

Типовые задачи, решаемые с помощью нейронных сетей и нейрокомпьютеров следующие (рис. 1.1):

- автоматизация процесса классификации;
- автоматизация прогнозирования;
- автоматизация процесса предсказания;
- автоматизация процесса принятия решений;
- управление;
- кодирование и декодирование информации;
- аппроксимация зависимостей и др.

1.2. Обзор областей применения

Примеры применения нейронных сетей и нейрокомпьютеров представлены в табл. 1.1. Безусловно, данный перечень не полон, однако он позволяет получить представление о характере востребованности нейросетевых и нейрокомпьютерных технологий.

В качестве иллюстрации успешного применения нейронных сетей рассмотрим следующие примеры [14].

1.2.1. Проектирование и оптимизация сетей связи

С помощью нейронных сетей успешно решается важная задача в области телекоммуникаций – нахождение оптимального пути трафика между узлами. Учитываются две особенности: во-первых, решение должно быть адаптивным, т. е. учитывать текущее состояние сети связи и наличие сбойных участков, а во-вторых, оптимальное решение необходимо находить в реальном времени.

Кроме управления маршрутизацией потоков, нейронные сети используются для получения эффективных решений в области проектирования новых телекоммуникационных сетей.

1.2.2. Распознавание речи

Распознавание речи – одна из наиболее популярных областей применений нейронных сетей.

Демонстрационная система для дикторо-независимого речевого управления встроенным калькулятором Windows (Российская компания Нейропроект) способна распознавать 36 команд, сказанных в стандартный микрофон. Для классификации слов используется двухкаскадная иерархическая нейронная сеть, где первый каскад состоит из одного

персептрона (1000 входов, 24 нейрона в скрытом слое, 6 выходов), а второй каскад – из 6 персептронов с различными параметрами слоев.

При этом первый персептрон осуществляет грубое распознавание слова, относя его к одному из 6 классов; роль второго каскада – точно классифицировать команду внутри класса. Для построения данной нейронной сети используется библиотека NeuroWindows, а также специальный алгоритм иерархического обучения. В обучении сети принимали участие 19 дикторов.

1.2.3. Управление ценами и производством

Часто недооцениваются потери от неоптимального планирования производства.

В связи с тем, что спрос и условия реализации продукции зависят от времени, сезона, курсов валют и многих других факторов, то и объем производства должен гибко варьироваться с целью оптимального использования ресурсов.

Нейросетевая система (компания Neural Innovation Ltd.), предназначенная для планирования затрат при издании газет, обнаруживает сложные зависимости между затратами на рекламу, объемами продаж, ценой, ценами конкурентов, днем недели, сезоном и т.д. В результате использования системы осуществляется выбор оптимальной стратегии издательства с точки зрения максимизации объема продаж или прибыли.

1.2.4. Анализ потребительского рынка

Один из популярных маркетинговых механизмов – распространение купонов, дающих право покупки определенного товара со скидкой. Так как затраты на рассылку купонов довольно велики, решающим фактором является *эффективность рассылки*, то есть повышение доли клиентов, воспользующихся скидкой. Для повышения эффективности купонной системы необходимо проведение предварительной сегментации рынка, а затем адресация клиентам каждого сегмента именно тех купонов, которыми они с большей вероятностью воспользуются.

Нейросетевая система (компания IBM Consulting), прогнозирующая свойства потребительского рынка пищевых продуктов, решает задачу *кластеризации* с помощью сетей Кохонена. На втором этапе для потребителей каждого из кластеров подбираются подходящие коммерческие предложения, а затем строится прогноз объема продаж для каждого сегмента.

Другой популярный маркетинговый механизм – распространение поощрительных товаров (когда, например, присылая 5 этикеток от кофе, клиент бесплатно получает кружку). Здесь, обычные методы прогнозирования отклика потребителей могут быть недостаточно точны:

иногда, спрос на кружки оказывается слишком велик и многие покупатели годами ждут получения приза.

Прогнозирующая нейросетевая система (компания GoalAssist Corp.) использует сеть с адаптивной архитектурой нейросимулятора NeuroShell Classifier (компании Ward Systems Group).

На входы данной нейронной сети, применяемой для классификации возможных откликов потребителей, подаются различные параметры товаров и рекламной политики для разделения входов на 4 вида откликов. Те же входы вместе с ответом первой сети подаются на вход сети нейросимулятора NeuroShell Predictor (компании Ward Systems Group), предназначенной для решения задачи количественного прогнозирования.

При этом средняя ошибка предсказаний эффекта от распространения поощрительных товаров составляет всего около 4,0 %.

1.2.5. Исследование спроса

Для сохранения бизнеса в условиях конкуренции компании приходится поддерживать постоянный контакт с потребителями – «обратную связь». Крупные компании проводят опросы потребителей, позволяющие выяснить, какие факторы являются для них решающими при покупке данного товара или услуги, почему в некоторых случаях предпочтение отдается конкурентам и какие товары потребитель хотел бы увидеть в будущем. Анализ результатов такого опроса – достаточно сложная задача, так как существует большое количество коррелированных параметров.

Нейросетевая система (компания Neural Technologies) позволяет выявлять сложные зависимости между факторами спроса, прогнозировать поведение потребителей при изменении маркетинговой политики, находить наиболее значимые факторы и оптимальные стратегии рекламы, а также очерчивать сегмент потребителей, наиболее перспективный для данного товара. В частности, система применяется для исследований предпочтений различных сортов пива в зависимости от возраста, дохода, семейного положения потребителя и других параметров.

1.2.6. Анализ страховых исков

Нейросетевая система Claim Fraud Analyser (компания Neural Innovation Ltd.) предназначена для выявления в реальном времени подозрительных страховых исков, поступающих в связи с повреждениями автомобилей. На входы системы подаются такие параметры, как возраст и опыт водителя, стоимость автомобиля, наличие подобных происшествий в прошлом и др.

В результате обработки такой информации нейронная сеть определяет вероятность того, что данный иск не связан с мошенничеством. Система позволяет не только обнаруживать фальсификации, но и улучшать

отношения с клиентами за счет более быстрого удовлетворения справедливых исков.

1.2.7. Обслуживание кредитных карт

Нейросетевая система Falcon (компания HNC), разработанная для отслеживания операций с краденными кредитными картами и поддельными чеками, позволяет по частоте сделок и характеру покупок выделить подозрительные сделки и сигнализировать об этом в контролирующие службы. Благодаря данной системе, отслеживающей более 260 миллионов счетов 16 крупнейших эмитентов кредитных карт, потери банков от таких операций заметно уменьшились.

Аналогичная система (компания ITC), используемая для обработки операций с кредитными картами VISA, предотвратила в 1995 г. нелегальные сделки на сумму более 100 млн долларов.

1.2.8. Медицинская диагностика

Система объективной диагностики слуха у грудных детей (Российская компания НейроПроект) обрабатывает зарегистрированные "вызванные потенциалы" (отклики мозга), проявляющиеся в виде всплесков на электроэнцефалограмме, в ответ на звуковой раздражитель, синтезируемый в процессе обследования.

Обычно, для уверенной диагностики слуха ребенка опытному эксперту-аудиологу необходимо провести около 2000 тестов, что занимает около часа. Система на основе нейронной сети способна с той же достоверностью определить уровень слуха уже по 200 наблюдениям в течение всего нескольких минут, причем без участия квалифицированного персонала.

1.2.9. Обнаружение фальсификаций

Подсчитано, что потери бюджета США от мошенничеств и фальсификаций в области здравоохранения составляют около 730 млн долларов в год. Тестирование системы обнаружения (стоимость – 2,5 млн долларов, компания ITC) показало, что нейронная сеть позволяет обнаруживать 38,0% мошеннических случаев, в то время как существовавшая ранее экспертная система – только 14,0 %.

1.2.10. Оценка недвижимости

Стоимость квартиры или дома зависит от большого числа факторов, таких как общая площадь, удаленность от центра, экологическая обстановка, престижность, тип дома, и т.д. Так как вид этих зависимостей неизвестен, то стандартные методы анализа неэффективны в задаче оценки стоимости. Как правило, эта задача решается экспертами-оценщиками, работающими в агентстве по недвижимости. Недостатком такого подхода является

субъективность оценщика, а также возможные разногласия между различными экспертами.

Система на основе нейронной сети (компания Attrasoft) способна эффективно решать широкий спектр задач объективной оценки стоимости недвижимости, в частности, с учетом 13 факторов при оценке стоимости домов в г. Бостон (США).

Группа исследователей из университета г. Портсмут (Великобритания) в системе на основе нейронной сети использовала данные по оценке недвижимости из обзоров риэлтеровских фирм и списков аукционных цен.

Результаты исследования показали, что система делает оценки стоимости близкие к оценкам лучших экспертов и специалистов данного профиля.

1.3. Распознавание символов

Распознавания букв и символов, с одной стороны – одна из наиболее разработанных и освещенных в специальной литературе проблем, а с другой – не смотря на кажущуюся простоту, чрезвычайно трудно реализуемая на практике задача.

Рассмотрим особенности применения нейронной сети (компания AT&T Bell Laboratories) при сортировке писем на почте в г. Буффало, США. Задача состоит в применении нейросетевых методов при разработке системы распознавания рукописных цифр, которые отправители писем указывали на конвертах в качестве индекса. Исследовались две строчки индексов: первые – написанные быстро и, как правило, неразборчиво, и вторые – написанные более тщательно печатными буквами.

Разработчики наполнили базу данных более 9000 символами, переведенных с конвертов, которые прошли через почтовую службу г. Буффало в 1988 г. На рис. 1.2 показаны некоторые почтовые индексы (вверху) и уже изолированные цифры, подготовленные для распознавания (внизу).

Видно, что индексы пишутся крайне неразборчиво, так что сотрудники почты считают, что некоторые отправители в действительности не желают, чтобы их письма доходили по назначению. В связи с этим, к сожалению, большинство подобных систем распознавания обладают точностью 95,0%, что является едва приемлемым показателем.

В целом ряде случаев, наиболее трудная проблема при распознавании символов – не собственно распознавание, а обнаружение символов и выяснение их местоположения, т.е. – верная интерпретация индекса, состоящего из известного количества позиций, изолирование и подготовка к распознаванию отдельных цифр индекса.

Поэтому для простоты будем полагать далее, что процесс распознавания начинается уже после изолирования цифры.

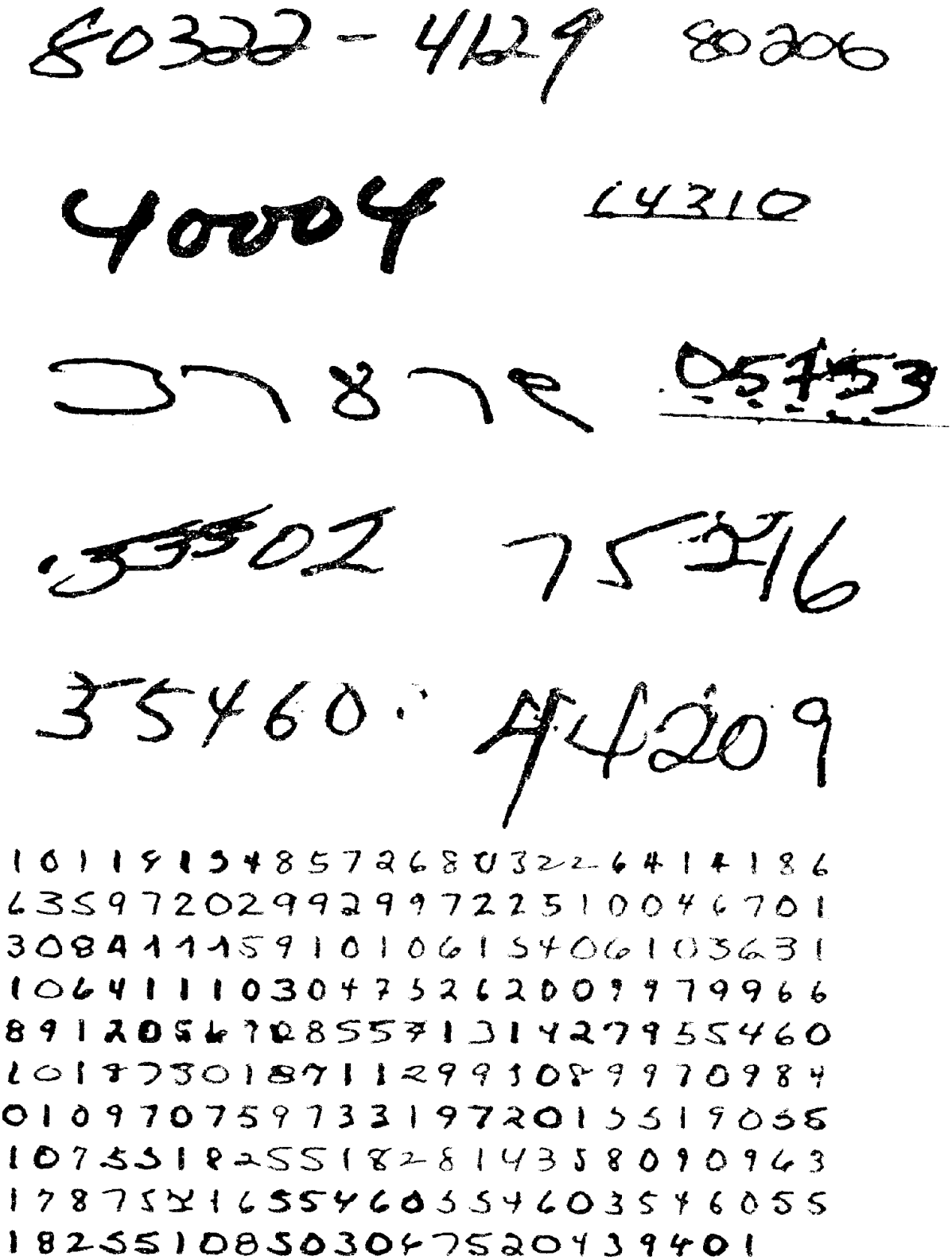


Рис. 1.2. Рукописные индексы, обрабатываемые почтовой службой г. Буффало (вверху) и взятые из них отдельные изолированные цифры (внизу)

Для каждой цифры разработчики строили решетку (сетку) размерностью 16 х 16 пикселей. В ходе исследований в 1988 г. было выяснено, что применение стандартного подхода, основанного на применении обученной нейронной сети обратного распространения к сырому массиву пикселей и чисел, приемлемого результата не дает.

В 1988-1990 гг. был предложен метод локализации информативных участков, вокруг которых строились решетки 5 х 5 и 7 х 7 пикселей, после чего на вход нейронной сети обратного распространения поступал 180-мерный вектор.

Конечная нейросетевая система распознавания представляет собой аппаратный модуль, реализованный на базе ПЦОС и соединенный с ПК. Обучение нейронной сети системы проводится единожды, однако достаточно медленно с использованием 167693 представительских выборок.

Ошибка системы в процессе распознавания символов – 0,14 % при предъявлении обучающей пары из набора представительских выборок, использованной при обучении, и 5,0 % при распознавании «новых» символов. Таким образом, разработчики и пользователи приняли решение о приемлемости результатов и необходимости использования системы для предварительной сортировки конвертов.

1.4. Искусственный нос

1.4.1. Принцип действия искусственного носа

Среди пяти чувств, чувство запаха наиболее загадочное.

Человеческий нос стал объектом исследования ученых и инженеров, специализирующихся в области высоких технологий и пытающихся понять, как нос функционирует. Такой повышенный интерес к обонятельной системе человека возник в связи с последними достижениями в области проектирования электронного (искусственного) носа.

В классическом понимании электронный нос представляет собой мультисенсорное цифровое устройство, предназначенное для анализа содержимого воздушной среды путем классификации запахов. Несмотря на то, что электронный нос сегодня не способен заменить человеческую обонятельную систему, сфера применения данной технологии достаточно широка.

В производственных целях возможности обонятельной системы человека широко используется во многих странах, например, для проверки различных продовольственных продуктов. Тренированный человеческий нос, детально изучив запахи продовольственных продуктов, таких, например, как зерно, сыр, вино, водка, рыба, способен в последствии определять их качество и свежесть. Аналогичным образом «нюхачи» оценивают перспективность того или иного парфюмерного запаха, обнаруживают фальсифицированные духи и дезодоранты. Запахи

учитываются также и докторами при выявлении общих заболеваний: такие болезни, как пневмония или диабет, вызывающие специфическое дыхание или жидкие выделения с характерными запахами, могут быть замечены квалифицированными врачами.

Если искусственный интеллект электронного носа окажется способен классифицировать запахи подобным образом, то тогда электронный нос смог бы справится с той же работой, причем гораздо лучшим образом.

Проблема в том, что человеческая обонятельная система чрезвычайно субъективная: зачастую, разные люди по-разному реагируют на одни и те же запахи. Электронный нос решает эту проблему, наверняка устанавливая «стандарт» для каждого требуемого запаха, например строго определяя запах испорченного зерна.

В настоящее время различные прототипы электронного носа уже широко используются в промышленности. В частности, в агрокомплексе Швеции электронный нос применяется для независимого контроля качества зерна путем автоматической классификации проб зерна на кондиционное и испорченное (достоверность – 90,0 %).

Другие проблемно-ориентированные разновидности электронного носа позволяют контролировать испарения вредных для здоровья химических, в т.ч. аллергических, веществ.

Привлекательность использования электронного носа в этих целях заключается в следующем:

- обнаружение тех или иных компонент носит объективный, а не субъективный характер;
- собственно процесс обнаружения токсичных веществ путем вдыхания воздуха через нос может быть вреден;
- некоторые химические вещества и комбинации веществ, которые легко обнаруживаются электронным носом, в традиционном понимании могут не иметь запаха;
- электронный нос способен функционировать в местах, где не может функционировать человеческий, например, в условиях крайне высоких и низких температур, внутри тела человека, в масляных или бензиновых резервуарах, в сточных трубах, на космических спутниках и т.д.

Для понимания принципа действия электронного носа, уточним, каким образом возникает запах: иногда его создает какое-либо одно химическое вещество, но чаще – комбинация множества различных химических компонент. Так, например, запах кофе формируют сотни различных молекул. Электронный нос должен реагировать на определенную концентрацию требуемых молекул и их комбинаций.

Существуют два принципиально разных подхода к реализации электронного носа:

- с использованием методов газовой хроматографии и масс-спектрометрии;
- с использованием нейросетевых методов и алгоритмов.

Однако, применение методов хроматографии и масс-спектрометрии, в отличие от нейросетевых методов, не позволяет регистрировать запахи, а лишь присутствие тех или иных отдельных химических компонент. Поэтому, хроматографы и масс-спектрометры могут называться электронным носом лишь условно.

Достоинства использования нейросетевого подхода следующие:

1. Применение нейросетевых методов позволяет отказаться от использования редких и дорогостоящих сенсоров. Используется, как правило, мультисенсорная комбинация («головка» или матрица), состоящая из набора (от 5 до 15) слабоселективных доступных по цене химических сенсоров.
2. Нейронные сети способны обнаруживать большее количество химических компонент, чем количество сенсоров нейросетевой системы.
3. Продолжительность измерений существенно короче. Следует отметить, что временные затраты меньше как на собственно регистрацию, так и на этапы подготовки и обработки измерений.

Говоря о *недостатках* использования нейросетевого подхода, можно упомянуть о необходимости предварительной настройки электронного носа на требуемые запахи (подготовки базы данных «стандартов» запахов), выражающейся, в частности, в предварительном обучении нейронной сети.

Далее, под электронным носом будем подразумевать интеллектуальные системы, реализованные на основе нейросетевых методов.

1.4.2. Аппаратура искусственного носа

Электронный нос состоит из двух основных функциональных частей (рис. 1.3): мультисенсорного измерительного модуля и интеллектуального цифрового блока.

Содержание кислорода в воздушной среде изменяется из-за присутствия в ней определенных химических веществ. Содержание кислорода, в свою очередь, изменяет выходное напряжение сенсора, которое измеряется как разность между текущим значением и нормальным (или стандартным) уровнем. Аналоговый сигнал затем преобразовывается АЦП в цифровой код, готовый для дальнейшей цифровой обработки. Измерительный модуль содержит аналоговый или цифровой препроцессор агрегирующий, а также усиливающий измерительный сигнал для уменьшения шума и «повышения» чувствительности сенсора.

Интеллектуальный цифровой блок (в частности ПК) регистрирует сигналы, поступившие от всех сенсоров, и формирует *исследуемый вектор*.

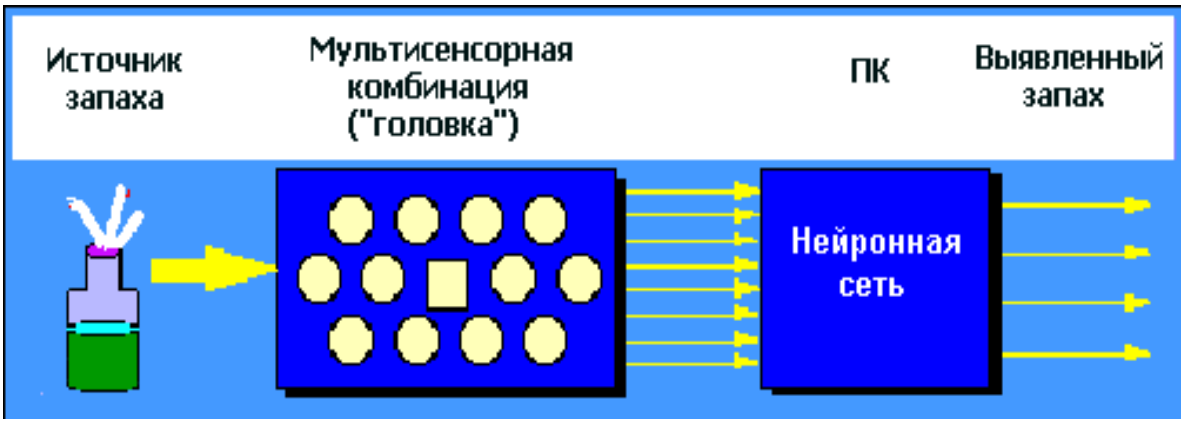


Рис. 1.3. Принцип действия искусственного носа

На *этапе предварительной настройки* электронного носа производится обучение нейронной сети путем установления «стандартов» для каждого требуемого запаха. Процесс обучения производится путем предъявления нейронной сети *представительских выборок*, состоящих из входного и выходного вектора нейронной сети. Роль входного вектора играет исследуемый вектор, сформированный из результатов измерений. Размерность выходного вектора соответствует количеству запахов, которые электронный нос должен распознать.

На *этапе тестирования* определяется насколько нейронная сеть качественно обучилась, а электронный нос способен достоверно распознавать предъявляемые ему запахи. Этапы предварительной настройки и тестирования, как правило, итерационные – если результат предварительной настройки не устраивает потребителя, то осуществляются необходимые изменения и настройка повторяется заново.

1.4.3. Пример реализации искусственного носа

Рассмотрим в качестве примера один из зарубежных прототипов электронного носа (рис. 1.4).

Данная система, реализованная в учебных целях, предназначена для автоматического определения запахов ряда изделий пищевой, бытовой и офисной химии, таких как ацетон, аммиак, изопропанол, белый «штрих» и уксус.

Мультисенсорная «головка» сформирована из девяти недорогих газовых сенсоров компании Figaro Co. Ltd. (сенсор 1 – TGS 109; сенсоры 2 и 3 – TGS 822; сенсор 4 – TGS 813; сенсор 5 – TGS 821; сенсор 6 – TGS 824; сенсор 7 – TGS 825; сенсор 8 – TGS 842; сенсор 9 – TGS 880), а также датчика влажности (сенсор 10 – NH-02) и двух датчиков температуры (сенсоры 11 и 12 – 5KD-5).

Датчики влажности и температуры встроены в «головку» для контроля условий проведения измерительных экспериментов, и их показания также включены в исследуемый вектор.

Хотя каждый из газовых сенсоров изначально был задуман и создан как моноселективный (т.е. для реагирования на вполне конкретный химический компонент), каждый из них, в силу своей конструкции, реагирует на широкий ряд химических веществ.

При этом, те или иные комбинации показаний всех сенсоров «головки» являются уникальными, и, следовательно, могут указывать на присутствие самых разных химических веществ и их комбинаций.

На этапе предварительной настройки нейронной сети предъявляются показания сенсоров (входной слой) и указываются химические вещества, которые этим показаниям соответствуют (выходной слой). Таким образом, на данных представительских выборках нейронная сеть обучается обнаруживать предъявленные настройщиком химические вещества.

Интеллектуальный цифровой блок данного прототипа реализован на основе ПК (рис. 1.4).

Для сравнения реализованы две разновидности нейронных сетей – стандартный многослойный персептрон со структурой: 11 – 6 – 6, обучаемый по алгоритму обратного распространения (рис. 1.5) и нечеткая ART-сеть.

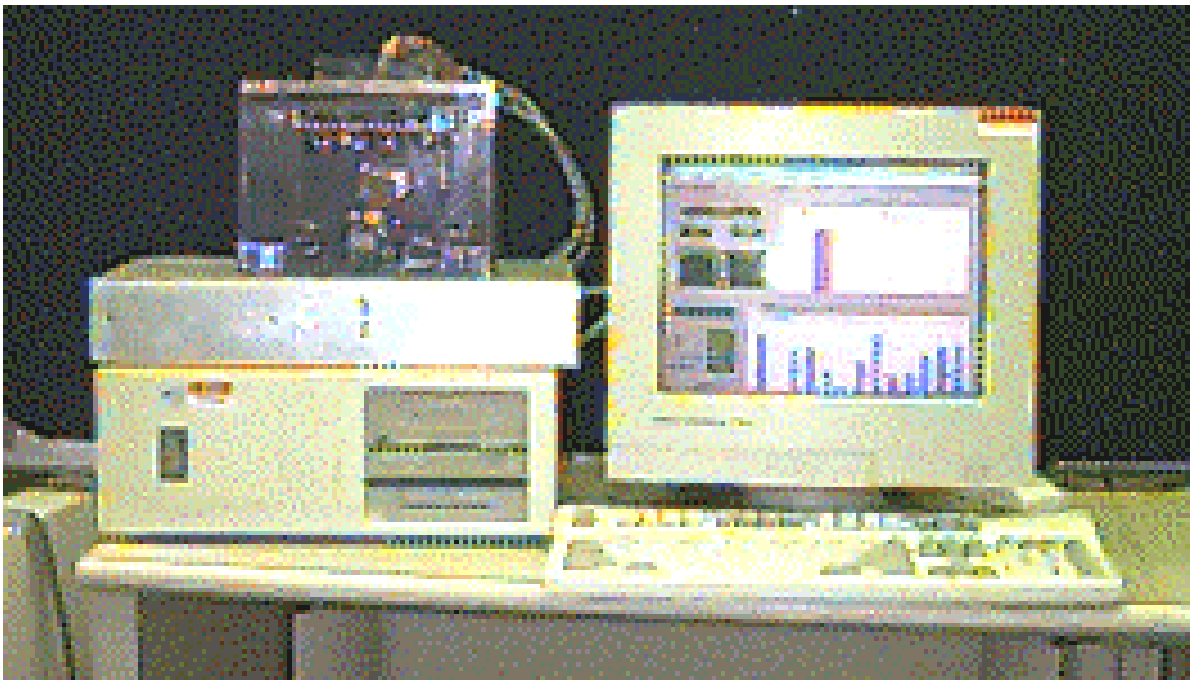


Рис. 1.4. Внешний вид прототипа искусственного носа

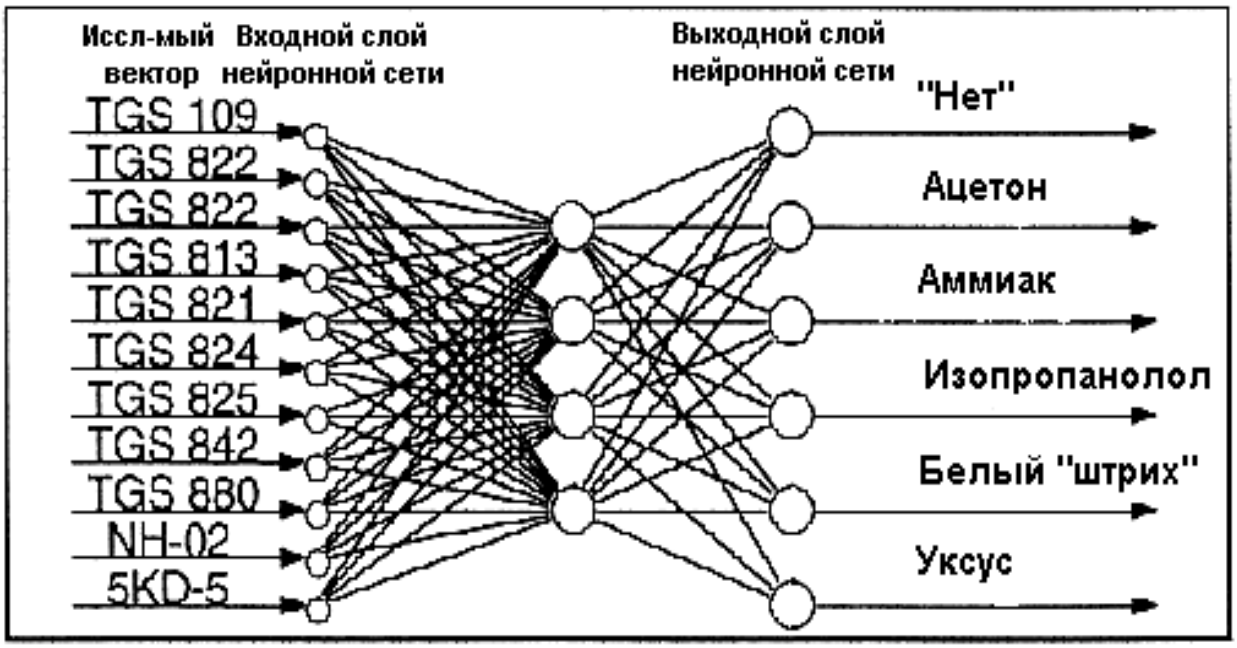


Рис. 1.5. Структура нейронной сети обратного распространения, используемой в искусственном носе

Таблица 1.2. Параметры нейронных сетей, применяемых в искусственном носе

Нейронная сеть с алгоритмом обратного распространения		
1	Структура нейронной сети	11 (вх. слой); 6 (внутр. слой); 11 (вых. слой)
2	Активационная функция	Сигмоидальная
3	Скорость обучения	0,10
4	Момент обучения	0,90
5	Количество итераций	1369
Нечеткая ART-сеть		
1	Чувствительность обучения	0,98
2	Чувствительность тестов	0,80
3	Количество итераций	3

Таблица 1.3. Результаты тестирования искусственного носа

	Количество обучающих пар	Количество тестов	Предъявляемое вещество	Вероятность идентификации, %	
				Алгоритм обратного распространения	Нечеткая ART-сеть
1	67	28	Нет	96,4	96,4
2	75	22	Ацетон	100,0	100,0
3	64	14	Аммиак	100,0	100,0
4	93	28	Изопропанол	92,9	100,0
5	5	3	Аммиак и изопропанол	0,0	66,7
6	106	25	Белый «штрих»	100,0	96,0
7	74	27	Аммиак и белый «штрих»	100,0	92,6
8	66	21	Уксус	81,0	95,2
9	68	26	Аммиак и уксус	92,3	76,9
10	1	2	Изопропанол и уксус	0,0	0,0
	619	196	В целом	92,9	93,4

Следует отметить, что быстродействие электронного носа ограничивается практически лишь временем отклика химических сенсоров, что соответствует единицам секунд.

Для настройки и тестирования прототипа электронного носа были использованы пробы пяти из вышеперечисленных веществ: ацетон, аммиак, изопропанол, белый «штрих» и уксус. Также добавлена категория «Нет» для обозначения ситуации, когда отсутствуют все перечисленные запахи. Параметры, используемые для обучения и тестирования нейронной сети с алгоритмом обратного распространения и нечеткой ART-сети, представлены в табл. 1.2.

Обе нейронные сети были обучены с использованием произвольно выбранных представительских выборок. Следует отметить, что при обучении во многих случаях не ставится задача, чтобы подготавливаемая нейронная сеть определяла уровни концентрации того или иного вещества. Наоборот, при обучении нейронной сети могут последовательно предъявляться одни и те же вещества, но в различной концентрации.

Благодаря этому нейронная сеть становится способной одинаково успешно определять как «густые» запахи, так и еле ощутимые.

Тестирование показало приблизительно одинаковые результаты для примененных видов нейронных сетей. При этом достоверность обнаружения запахов (ошибка идентификации) варьировалась на интервале от 89,7 % до 98,2 % в зависимости от используемых проб, которые выбирались произвольным образом. В табл. 1.3 приведены количественные результаты тестирования при обнаружении различных составов.

Показания сенсоров и результаты обнаружения запахов, осуществленных интеллектуальным цифровым блоком (рис. 1.6 и 1.7), иллюстрируют тот факт, что качественно обученная нейронная сеть способна правильно классифицировать предъявляемые запахи с приемлемым уровнем достоверности (выше 90,0 %).

Показания сенсоров свидетельствуют (рис. 1.6), что по отношению к классифицируемым запахам они не являются моноселективными и реагируют с разной степенью интенсивности на все предъявляемые им запахи. Обнаружение отдельного запаха при предъявлении нейронной сети состава из двух или нескольких запахов, в принципе, может вызвать некоторые затруднения электронного носа, однако, из рис. 1.7 c, d, e видно, что ошибка и этой идентификации также невелика.

1.4.4. Искусственный нос для контроля окружающей среды

В связи с объективной тенденцией распространения высоких технологий и высокотехнологических производств, существенно возрастет спрос на те или иные прототипы электронного носа. Ожидается, что в ближайшем будущем электронный нос будет востребован для реализации крупных экологических программ, направленных на защиту окружающей воздушной и водной среды.

Гигантские объемы опасных (ядерных, химических и комбинированных) отходов накоплены за более чем 40 лет производства оружия в США. По поручению отдела энергетического оружия Минобороны США Северо-западная атлантическая национальная лаборатория изучает технологии по воссозданию окружающей среды и рентабельной утилизации опасных отходов. Данная программа подразумевает, в том числе, разработку портативных, недорогих систем, таких как электронный нос, способных в реальном масштабе времени идентифицировать опасные загрязняющие вещества в воздушных и жидких средах.

Прототипы электронного носа могут быть широко использованы также при контроле и идентификации токсичных выбросов в атмосферу; при анализе топливных смесей; при обнаружении выбросов масляных смесей; при исследовании качества, в том числе и запаха, артезианских вод; для контроля качества воздуха в помещениях; обнаружения наркотиков и взрывчатых веществ и др.

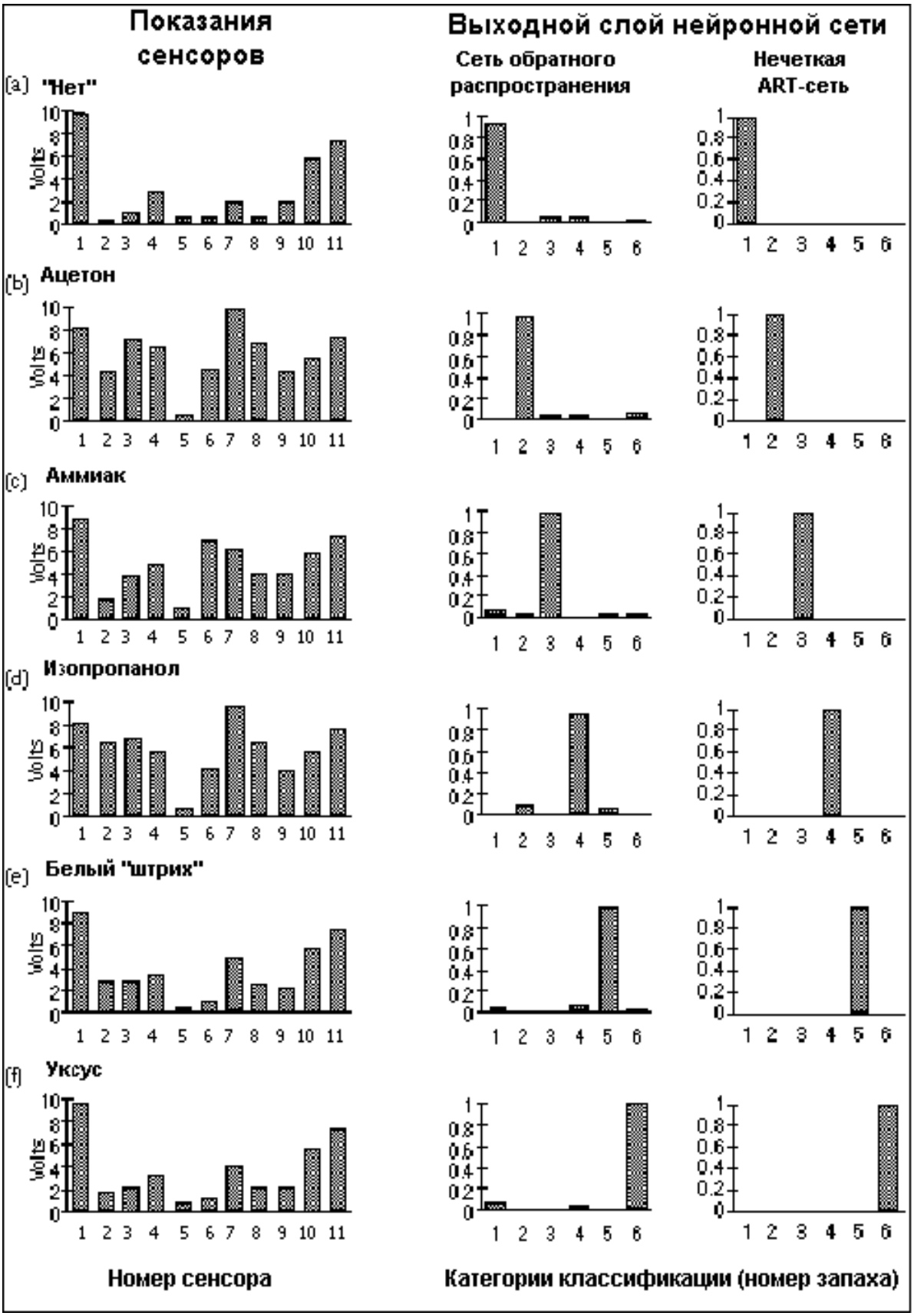


Рис. 1.6. Показания сенсоров и результаты автоматической классификации запаха при предъявлении одного запаха

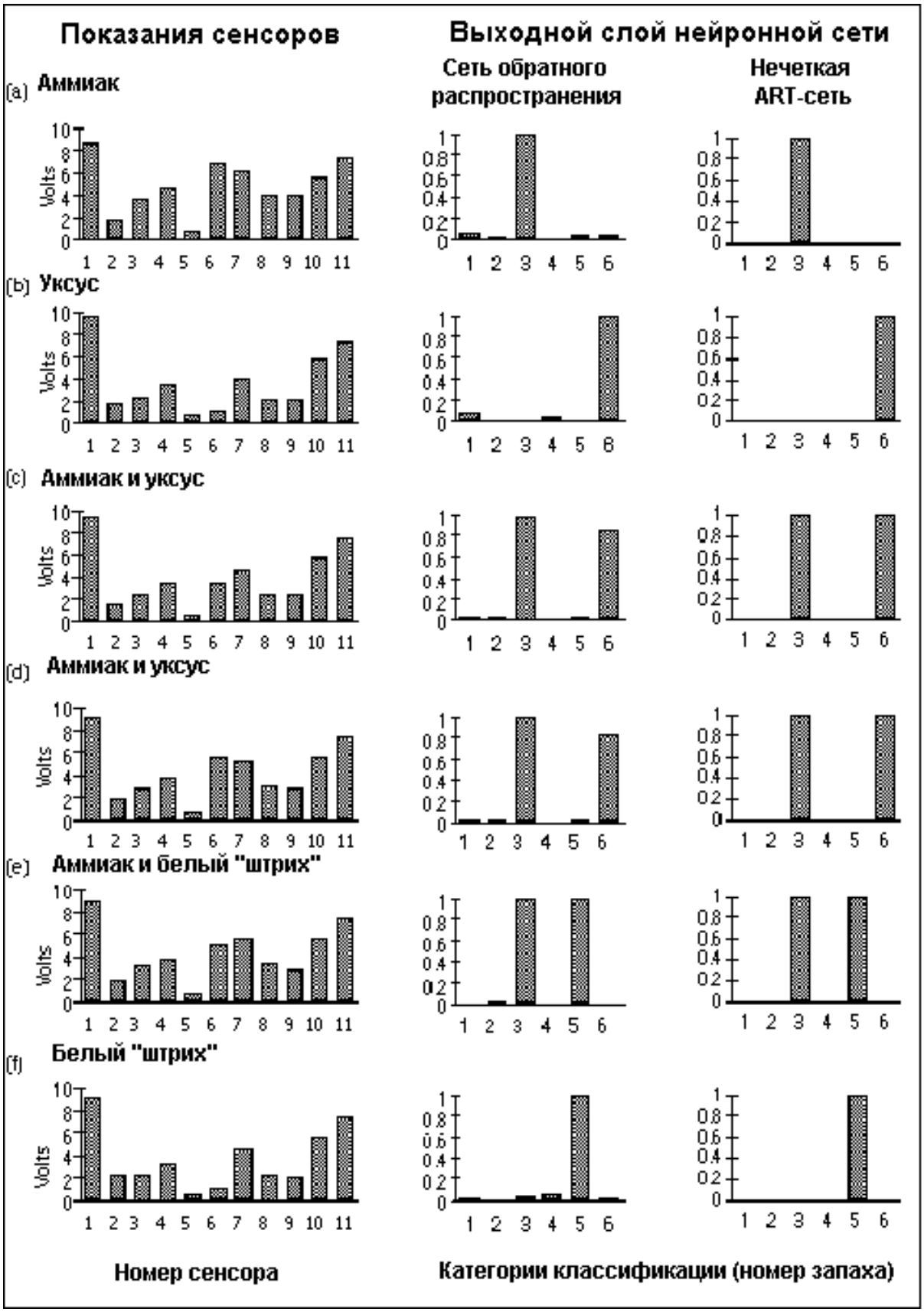


Рис. 1.7. Показания сенсоров и результаты автоматической классификации запаха при предъявлении состава из двух запахов

1.4.5. Искусственный нос в медицине

Применение электронного носа в качестве диагностического прибора обусловлено тем, что запах широко используется медиками при диагностике целого ряда болезней. Функциональные возможности электронного носа позволяют диагностировать те или иные отклонения организма при исследовании дыхания, запаха ран, различных физиологических и других жидкостей и т.п.

Так, запахи при дыхании могут свидетельствовать о желудочно-кишечных болезнях, диабете, болезнях печени и некоторых инфекционных заболеваниях.

Инфекционные раны и ткани также выделяют отчетливые запахи, характер которых может анализироваться с применением электронного носа. В настоящее время прототип такого электронного носа используется для исследований инфекционных ран в университетской больнице южного района г. Манчестер (Великобритания).

Анализ запахов жидкостей тела, может быть использован также для диагностики заболеваний печени и других органов.

Сотрудники медучреждений при возникновении подозрений с успехом используют электронный нос для обнаружения просроченных, недоброкачественных или фальсифицированных лекарств и препаратов.

Ожидается, что в будущем электронный нос будет востребован в дистанционной хирургии. В настоящее время в этих целях широко распространена передача визуальных, звуковых и механических сигналов, в то время как значение запаха игнорируется. Потенциально, электронный нос способен идентифицировать операционные запахи и предоставлять для передачи обонятельные сигналы, создавая для дистанционной хирургии полноту среды так называемой виртуальной реальности.

1.4.6. Искусственный нос в пищевой промышленности

В настоящее время наибольший рынок для электронного носа предоставляется в агропромышленном комплексе.

Прототипы электронного носа применяются как для оценки качества продуктов питания, так и контроля качества приготовления пищи.

Предприятия агрокомплекса расширяют применение данных технологий, в частности, для контроля свежести рыбы на промыслах и оптовых складах; контроля процессов брожения; обследования контейнеров, резервуаров и элеваторов; проверки натуральности апельсинового сока; проверки прогорклости майонеза; ограничения лукового запаха; установления сортности (выдержанности) виски и коньяков; автоматического управления вкусом и т.д.

Важнейшее направление применения электронного носа – обнаружения на оптовых складах и в торговле фальсифицированных продуктов питания, напитков (в частности, водки) и пищевых добавок.

В ряде случаях результаты работы электронного носа могут быть использованы в качестве дополнительной аналитической информации для экспертов – специалистов в области запахов. В других случаях, использование электронного носа при химическом анализе в агрокомплексе предпочтительнее приобретения аналитической аппаратуры, характеризующейся большей стоимостью и продолжительностью экспериментов. Это особенно заметно, когда требуется получение не количественных, а качественных результатов типа «свежее – не свежее», «натуральное – не натуральное», «качественное – не качественное», «настоящее – фальсифицированное» и т.д.

1.5. Прогнозирование

Прогнозирование – важнейший элемент современных информационных технологий принятия решений в управлении.

Эффективность того или иного управленческого решения оценивается по событиям, возникающим уже после его принятия. Поэтому прогноз неуправляемых аспектов таких событий перед принятием решения позволяет сделать наилучший выбор, который, без прогнозирования мог бы быть не таким удачным.

Прогнозирование – одна из самых востребованных, но при этом одна из самых сложных задач интеллектуального анализа данных. Проблемы прогнозирования связаны с недостаточным качеством и количеством исходных данных, изменениями среды, в которой протекает процесс, воздействием субъективных факторов. Прогноз всегда осуществляется с некоторой погрешностью, которая зависит от используемой модели прогноза и полноты исходных данных. При увеличении информационных ресурсов, используемых в модели, увеличивается точность прогноза, а убытки, связанные с неопределенностью при принятии решений, уменьшаются.

Характер затрат, связанных с прогнозированием, таков, что за определенным пределом дополнительные затраты не приведут к снижению потерь. Это связано с тем, что объективно невозможно снизить погрешность прогнозирования ниже определенного уровня, вне зависимости от того насколько хорош примененный метод прогнозирования. Поэтому определение погрешности прогноза, наряду с самим прогнозом, позволяет значительно снизить риск при принятии решений.

Известны и широко применяются различные методы прогнозирования: алгоритмы экстраполяции экспериментальных данных в несложных инженерных расчетах и программных продуктах, а также более громоздкие статистические методы, использующие параметрические модели.

В последние десятилетия для прогнозирования широко применяются другие подходы, и в частности, нейронные сети. Рассмотрим особенности применения нейронных сетей, которые показывают их *преимущества* по

сравнению с другими существующими методами при выборе модели прогноза (*ограничения* и *недостатки* применения нейронных сетей при прогнозировании см. в п. 1.5.4).

1. *Результативность при решении неформализованных или плохо формализованных задач*. Из общеизвестных преимуществ методов на основе нейронных сетей следует выделить одно самое привлекательное – отсутствие необходимости в строгой математической спецификации модели, что особенно ценно при прогнозировании плохо формализуемых процессов. Известно, что большинство финансовых, бизнес и других подобных задач плохо формализуется.
2. *Устойчивость к частым изменениям среды.* Достоинства нейронных сетей становятся заметными, когда часто изменяются «правила игры»: среда, в которой существует прогнозируемый процесс, а также характер воздействия влияющих факторов. Поэтому, нейронные сети наилучшим образом подходят для решений таких задач, как прогнозирование тенденций фондового рынка, характеризующихся влиянием целого набора постоянно изменяющихся факторов.
3. *Результативность при работе с большим объемом противоречивой информации*. Нейронные сети будут предпочтительнее там, где имеется очень много анализируемых данных, в которых скрыты закономерности. В этом случае автоматически учитываются также различные нелинейные взаимодействия между влияющими факторами. Это особенно важно, в частности, для предварительного анализа или отбора исходных данных, выявления «выпадающих фактов» или грубых ошибок при принятии решений.
4. *Результативность при работе с неполной информацией*. Целесообразно использование нейронных сетей в задачах с неполной или "зашумленной" информацией, а также в задачах, для которых характерны интуитивные решения.

1.5.1. Постановка задачи прогнозирования

Задача прогнозирования в общем случае сводится к получению оценки будущих значений упорядоченных во времени данных на основе анализа уже имеющихся, а также (при необходимости) тенденции изменения влияющих факторов. Прогнозируемой величиной являются значения временного ряда на интервале $[T(n+1), T(n+f)]$, где $T(n)$ – текущий момент времени, а f – интервал прогнозирования. Иногда возникает необходимость не в прогнозе значений временного ряда на заданном интервале, а в прогнозе вероятности того, что они будут вести себя тем или иным образом (возрастать, убывать, находиться в некоторых пределах и т.д.).

Рассмотрим типовой алгоритм прогнозирования, осуществляемого с использованием нейронных сетей (рис. 1.8).

Рис. 1.8. Типовой алгоритм прогнозирования, осуществляемого с использованием нейронных сетей

Отбор значащих факторов. На первом этапе выделяется максимальное число из значащих, влияющих на прогноз, факторов. Такие дополнительные факторы, влияющие на поведение прогнозируемой величины, называют экзогенными (внешними) или артефактами.

Здесь же выбирается интервал наблюдения (окно скольжения), т. е. выясняется, по какому количеству предшествующих значений временного ряда осуществляется прогноз.

Предобработка данных. На втором этапе устраняются несущественные, по мнению эксперта, и не влияющие на прогноз, данные. При необходимости, также, восстанавливается пропущенная информация, устраняются аномальные выбросы, убираются высокочастотные шумы. Умело проведенная предобработка данных позволяет значительно улучшить качество прогноза.

Построение модели. На следующем этапе для данного анализируемого процесса выбирается наиболее подходящая парадигма и структура нейронной сети, а также алгоритм и параметры ее обучения.

Собственно *прогнозирование (получение результата).* Эксперименты осуществляются по схеме, аналогичной той, при которой производилось обучение.

Рассмотрим так называемый метод скользящих окон. Он предполагает использование двух окон *Wi* и *Wo* с фиксированными размерами *n* и *m* соответственно. Эти окна перемещаются с некоторым шагом скольжения *s* по временной последовательности имеющихся данных, начиная с первого элемента. При этом первое окно *Wi* длиной *n* формирует входной вектор нейронной сети, а второе – *Wo* – выходной вектор размерностью *m*.

Последовательность обучающих выборок (обучающих пар)

$$Wi \rightarrow Wo$$

формирует т.н. блок обучающих или представительских выборок.

Пример 1.1. Простейшая модель прогнозирования продаж с помощью нейронной сети

Дано:

Информация об еженедельных продажах компьютеров (табл. 1.4.) за четыре месяца (число недель $k = 16$).

Таблица 1.4. Данные о еженедельных продажах компьютеров

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
100	94	90	96	91	94	95	99	95	98	100	97	99	98	96	98

Получить:

Многошаговый и одношаговый прогноз продаж.

Пример решения:

В данной временной последовательности предполагается наличие скрытых нелинейных зависимостей. Поэтому для построения модели прогноза применим нейронную сеть:

1. Синтезируем нейронную сеть со следующими параметрами:

- число нейронов входного слоя (ширина окна) $n = 4$;
- число нейронов выходного слоя $m = 1$;
- сдвиг (скольжения) $s = 1$.

2. С помощью метода скользящих окон для нейронной сети формируется блок обучающих (представительских) выборок (табл. 1.5).

Очередная обучающая выборка получается в результате сдвига окон *Wi* и *Wo* вправо на один элемент ($s = 1$). Нейронная сеть обучается на данных выборках, настраивая свои коэффициенты, и формирует в качестве результата требуемую функцию прогноза *P*.

Собственно процесс прогнозирования осуществляется после обучения нейронной сети, проводимого по тому же принципу, что и процесс формирования обучающей выборки.

Таблица 1.5. Блок обучающих выборок нейронной сети, применяемой для целей прогнозирования продаж

№ обучающих выборок	Входной слой				Выходной слой
	1	2	3	4	
1	100	94	90	96	91
2	94	90	96	91	94
3	90	96	91	94	95
4	96	91	94	95	99
5	91	94	95	99	95
и т.д.					

Нейронная сеть должна обучится на временной последовательности, взятой из табл. 1.4 с использованием блока обучающих выборок из табл. 1.5. Затем, при подаче на вход нейронной сети последней из известных выборок 99, 98, 96, 98 (см. конец табл. 1.4), прогнозируется (k+1)-й элемент последовательности, например, 95.

На данном простейшем примере рассмотрим действия при *многошаговом* и *одношаговом* прогнозировании.

Одношаговое прогнозирование. Применяется для краткосрочных прогнозов на один шаг вперед. На очередном текущем шаге в качестве исходной информации используются только объективные данные (результаты прогнозов, полученных на предыдущих шагах, не используются).

Если на шаге (k+1)-м для временной последовательности, взятой из табл. 1.4, прогнозируется 95, а на самом деле продается не 95, а 96 компьютеров, то на шаге (k+2)-м в качестве входного вектора будет использована выборка 98, 96, 98, 96.

Многошаговое прогнозирование. Применяется для осуществления долгосрочного прогноза и предназначено для определения основного тренда и главных точек изменения тренда для некоторого интервала времени. При этом прогнозирующая система использует результаты прогноза (выходные данные), полученные для моментов времени k+1, k+2 и т.д. в качестве входных данных для прогнозирования на моменты времени k+2, k+3, k+4 и т.д.

Многошаговое прогнозирование на (k+2)-м шаге продолжается при подаче на вход нейронной сети выборки 98, 96, 98, 95, в которой последний элемент является результатом прогноза на предыдущем шаге. И так далее.

1.5.2. Прогнозирование в сфере бизнеса и финансов

Подавляющее большинство задач прогнозирования на основе нейронных сетей так или иначе связаны со сферой бизнеса и финансов. Это – краткосрочные и долгосрочные прогнозы тенденций следующих финансовых рынков:

- рынков купонных и бескупонных облигаций,
- фондовых рынков (рынков акций),
- валютных рынков.

Сюда же можно отнести прогнозы:

- платежеспособного спроса;
- продаж и выручки;
- рисков кредитования;
- финансирования экономических и инновационных проектов;
- фьючерсных контрактов и ряд других.

По оценкам экспертов, бум вокруг систем искусственного интеллекта в финансовой индустрии пришелся на период 1984 - 1989 гг. В основном он затронул США и Великобританию, где создатели сложных систем для военных (таких как программа «Звездные войны») решили попытать счастья на Уолл-стрит.

Фондовая биржа в Нью-Йорке в 1987 году начала использовать программный продукт прогнозирования Stockwatch Alert Terminal (SWAT) II и вела переговоры о его внедрении с рядом бирж Европы и региона Юго-восточной Азии. В этот период времени на рынке появляются программные продукты моделирования и прогнозирования банкротства, анализа портфеля ценных бумаг, оптимальной торговли акциями, а также предназначенные для определения вероятности риска при выдаче кредита. Подразделение Mellon Bank в Питтсбурге (США) применило программную систему Neural-Works Professional II/Plus 5,0 компании Neural-Ware (Питтсбург, США) для

распределения фондов и специальной селекции акций, так как в ходе работы было обнаружено, что между влияющими факторами и прогнозируемыми параметрами существуют нелинейные связи, не поддающиеся точному учету с помощью стандартных статистических методов.

Департамент торговли и промышленности правительства Великобритании финансирует две программы, направленные на развитие нейронных вычислений в финансовой сфере. Первая – «Нейропрогнозирование», инициированная Лондонской школой бизнеса совместно с университетским колледжем Лондона (UCL). Вторая – «Нейронные сети для финансовых услуг» создана TBS Bank Technology совместно с UCL и Центром прогнозирования Henley. Среди финансовых институтов, использующих технологию нейронных сетей – Chemical Bank, Citibank, JP Morgan и др.

Специалисты программы «Нейропрогнозирование» разработали модель для выработки тактики распределения фондов на глобальных рынках облигаций. Модель охватывает семь географических регионов: Великобританию, Францию, Германию, Японию, США, Канаду, Австралию, каждый из которых моделируется с помощью нейронных сетей с различной структурой. Для получения краткосрочных прогнозов обучение нейронной сети производилось с использованием статистической информации, характеризующей ситуацию на этом рынке за каждый месяц. Далее, полученные локальные прогнозы объединяются в центре управления единым портфелем ценных бумаг. С ноября 1992 года данная программная система использовалась Североамериканской страховой компанией (г. Бостон, США). В результате использования капитал компании увеличился с 25,0 до 50,0 млн долларов, а портфель ценных бумаг повысил доходность на 25,0 % в первый год внедрения системы.

В качестве инструмента для оптимизации параметров нейронных сетей часто используются генетические алгоритмы. В частности, компанией Hill Samuel Investment Management разработана программная система для прогнозирования результатов контрактов по долгосрочным ценным бумагам повышенной надежности. При моделировании нескольких стратегий торгов в задаче прогнозирования направлений движения рынка она достигла точности 57,0 %.

В страховой компании TSB General Insurance (г. Ньюпорт, США) используется сходная методика для прогноза уровня риска при страховании частных кредитов. Эта нейронная сеть самообучается на статистических данных о состоянии безработицы в стране.

Прогнозирования краткосрочных и долгосрочных тенденций финансовых рынков

Задачей автоматизированной системы прогнозирования краткосрочных и долгосрочных тенденций финансовых рынков является

анализ некоторого набора влияющих факторов с последующим выводом о дальнейшем краткосрочном или долгосрочном поведении прогнозируемой величины.

Возможными прогнозируемыми величинами для подобных систем являются доходность и ценовые показатели: средневзвешенная цена, цены закрытия и открытия, максимальная и минимальная цены. Причем прогнозироваться могут как показатели, определенные для целой группы инструментов или некоторого рынка в целом, так и показатели, определенные только для одного инструмента финансового рынка. Как для совокупности инструментов, так и для каждого индивидуально может определяться доходность; ценовые показатели определяются для каждого конкретного инструмента. Целями прогноза (прогнозируемой величиной) в области финансовых рынков могут являться, например, средневзвешенная доходность бескупонных облигаций (для группы инструментов), средневзвешенная цена акции РАО «ЕЭС России», курс американского доллара к рублю и др.

В качестве исходных данных (влияющих факторов) для такого прогноза могут использоваться различные макро- и микроэкономические показатели, информация с торговых площадок, данные, предоставляемые информационно-торговыми агентствами, экспертные оценки специалистов. Количество влияющих на прогноз факторов зависит от рассматриваемого рыночного инструмента и конкретной рыночной ситуации (временного момента). То есть одни факторы оказывают влияние на все финансовые рынки, другие – только на определенные. Кроме того, влияние факторов на рынки может меняться с течением времени (меняются рыночные тенденции). Так как в определенные моменты времени прослеживается явная взаимосвязь между финансовыми рынками и инструментами рынков, целесообразно в качестве исходных данных для прогнозирования одного рынка или его инструментов использовать информацию о тенденциях других рынков. Например, при прогнозировании цены открытия «сегодня» для любых инструментов всех финансовых рынков, этот прогноз сильно зависит от цены закрытия «вчера» и обе эти величины могут выступать как прогнозируемые.

Прогнозирование краткосрочных и долгосрочных тенденций финансовых рынков включает следующие этапы.

1. Сбор и хранение статистических данных – возможной исходной информации для прогноза (либо в качестве исходных данных, либо в качестве прогнозируемой величины, либо как и то и другое);
2. Определение для рассматриваемого рынка или инструмента прогнозируемой величины и набора влияющих факторов (причем не всегда могут быть использованы данные, непосредственно хранящиеся в базе данных, зачастую требуется произвести

некоторые преобразования данных: например, часто в качестве таких факторов используются относительные изменения величин);
3. Выявление зависимости между прогнозируемой величиной и набором влияющих факторов в виде некоторой функции;
4. Вычисление интересующей величины в соответствии с определенной функцией, значениями влияющих факторов на прогнозируемый момент и видом прогноза (краткосрочный или долгосрочный).

Процедура выполнения краткосрочного прогноза отличается от процедуры долгосрочного на первом и четвертом этапах. В случае краткосрочного прогноза считается, что все участвующие в нем значимые влияющие факторы на прогнозируемую дату известны и хранятся в базе данных. Горизонт краткосрочного прогноза не превышает 3–4 дня. В случае долгосрочного прогноза считается, что значимые влияющие факторы на прогнозируемую дату неизвестны и должны быть указаны ожидаемые значения и погрешности. Соответственно погрешность определения прогнозируемой величины существенно увеличивается (чем дальше горизонт прогноза, тем больше погрешность определения влияющих факторов и вероятность ошибки аналитика). Горизонт долгосрочного прогноза, как правило, превышает 3–4 дня.

Кредитование

Характерный пример успешного применения нейронных сетей в финансовой сфере – управление кредитными рисками. Перед выдачей кредита для оценки вероятности собственных убытков от несвоевременного возврата финансовых средств крупные банки, как правило, предпринимают сложные статистические расчеты по определению финансовой надежности заемщика. Такие расчеты обычно базируются на оценке кредитной истории, динамики развития компании, стабильности ее основных финансовых показателей и многих других факторов. Так, Bank of New York, США опробовав метод нейронных вычислений и применив его для оценки 100 тыс. банковских счетов, выявил свыше 90,0 % потенциальных неплательщиков.

Прогнозирование тенденций фондового рынка (рынка акций)

Важная область применения нейронных сетей в сфере финансов – прогнозирование ситуации на фондовом рынке. Стандартный подход к решению этой задачи (не использующий нейронные сети) базируется на жестко фиксированном наборе «правил игры», который со временем теряет свою актуальность из-за изменения условий торгов на фондовой бирже. Помимо того, системы, построенные на основе такого стандартного подхода, оказываются слишком медленными для ситуаций, требующих от трейдера (участника торгов) мгновенного принятия решений.

Рассмотрим, некоторые особенности действий на фондовом рынке.

Трейдер, принимающий решения о купле–продаже акций, имеет доступ к одному или нескольким электронным источникам информации (Reuters, Dow Jones Telerate, Bloomberg, Tenfore). Он наблюдает текущие значения и графики интересующих его индексов на мировых фондовых биржах, основные кросс-курсы валют и другие показатели валютного, фондового и кредитного рынков в многооконной среде с различной степенью детализации.

На принятие его решения о купле–продаже акций, естественно, влияют макроэкономические и общественно-политические события, сообщения о которых через каждые 5–10 минут появляются в текстовом окне монитора и сопровождаются комментариями экспертов, озвучивающих разнообразные слухи и прогнозы. Трейдеру также доступна дополнительная информация, такая как сообщения из Центрального банка России и от других значащих источников об основных показателях рынков.

Обязательно учитывается психология конкурирующих трейдеров, для которых важную роль играют ожидания ряда влияющих событий. Например, в 16:00 многие московские трейдеры внутренне готовы к изменениям тенденции поведения индекса Доу-Джонса на Нью-Йоркской фондовой бирже, которая с учетом сдвига по часовым поясам открывается лишь в 17:30 по московскому времени.

Фондовый рынок характеризуется также следующими *особенностями*:

- рыночные процессы весьма неоднородны во времени: например, состояние рынка осенью существенно отличается от его состояния летом того же года; поэтому не всегда имеет смысл формировать обучающие выборки большого объема;
- «загрязнениями» данных и их неоднородностью;
- наличием малоинформативных показателей при относительно малом объеме статистики.

В целом, задача краткосрочного прогноза котировок акций пусть и с использованием нейронных сетей представляется достаточно сложной, особенно на стремительно изменяющемся российском фондовом рынке.

Примером прогнозирования тенденций фондового рынка может служить нейросетевая система (компания Alela Corp.), предназначенная для прогноза изменения биржевых индексов Dow Jones, S&P500 и Merval. На сайте компании можно бесплатно воспользоваться прогнозом изменения данных индексов и, используя его в качестве дополнительной информации, убедиться, что доля верных прогнозов составляет не менее 80,0 %.

Японские компании, оперирующие на рынке ценных бумаг, также широко применяют нейронные сети (компания Mitsubishi). Для входа нейронной сети использовалась информация о деловой активности нескольких организаций, полученная за 33 года, включая также оборот, предыдущую стоимость акций, уровни дохода и т.д. Данная нейронная сеть

самообучалась на реальных примерах и показала высокую точность прогнозирования, а также быстродействие. Общая результативность прогноза по сравнению с системами, использующими стандартные статистические подходы, улучшилась на 19,0 %.

Оптимальное распределение свободных средств банка между различными финансовыми рынками

Успешное прогнозирование поведения как финансовых рынков в целом, так и их отдельных инструментов позволяет банку эффективнее управлять имеющимися в его распоряжении средствами.

Задача оптимального распределения свободных средств между различными финансовыми рынками и их инструментами встает перед банком ежедневно. Любой банк имеет в своем распоряжении «портфель», куда могут входить различные ценные бумаги и валюта. Принцип формирования портфеля - получение прибыли с вложенного в финансовые инструменты капитала не ниже некоторого фиксированного уровня при минимальном для этого уровня риске.

Ежедневно могут происходить следующие взаимоисключающие процессы: поступление денежных средств для их вложения в финансовые инструменты и отток денежных средств для выполнения обязательств банка. То есть существуют следующие причины для изменения состава портфеля:

- с течением времени отдельные финансовые инструменты начинают терять свою привлекательность и необходимо выполнить оптимальное (доходность не ниже фиксированного уровня, риск - минимальный) перераспределение средств между финансовыми инструментами внутри портфеля;
- банку необходимо выполнить некие требования, для чего реализуется некоторое количество финансовых инструментов, входящих в портфель, на определенную сумму; естественно, что финансовые инструменты должны быть выбраны таким образом, чтобы характеристики портфеля по возможности не ухудшились;
- у банка увеличился объем свободных денежных средств и необходимо произвести их оптимальное распределение между различными финансовыми инструментами.

Независимо от причины и механизма изменения состава портфеля расчет выгодности этих изменений производится на фиксированную дату, называемую горизонтом портфеля.

Исходными данными для задачи оптимального распределения свободных средств между различными финансовыми рынками и их инструментами являются либо результаты долгосрочного прогноза для всех рассматриваемых инструментов, либо вероятностный анализ поведения рассматриваемых инструментов в сходных рыночных ситуациях.

Выбор финансового инструмента с максимальной предполагаемой доходностью не составляет большого труда, но задача усложняется необходимостью учитывать риск предполагаемых вложений, т.е. возможность инструмента не реализовать эту доходность. Как правило, чем выше доходность, тем выше риск, и снижение риска ведет к снижению доходности. Поэтому при планировании распределения средств банка рассматриваются две задачи:

- вложение средств с минимальным риском;
- вложение средств с доходностью не ниже фиксированного уровня и минимальным для этого уровня риском.

Классическим примером снижения риска портфеля в целом является сочетание в нем инструментов с отрицательным коэффициентом корреляции.

Поступление данных в систему. В программном продукте, применяемом в Промстройбанке, реализован автоматизированный ввод в нейронную сеть новой информации из следующих источников:

- информационно-торговые данные агентства REUTERS;
- торговые данные с площадок ММВБ и РТС;
- прочие данные с использованием ручного ввода.

Выбор и подготовка данных для участия в прогнозе. Задача данного этапа прогнозирования – выбор из более чем 200 видов информационно-торговых данных наиболее значимых влияющих факторов для прогноза интересующей стоимостной величины некоторого финансового инструмента или группы финансовых инструментов. Первичный выбор влияющих факторов осуществляется специалистом и зависит от его опыта и интуиции, в виду того, что автоматизация этого процесса, как правило, неэффективна. В помощь специалисту предоставляются инструменты технического анализа в виде графиков, анализируя которые можно уловить скрытые взаимосвязи. Специалист также может использовать доступные ему матрицы корреляции и ковариации для указанной выборки влияющих факторов и прогнозируемой величины, однако, с помощью матриц корреляции и ковариации не удается уловить нелинейную, редко возникающую зависимость, которая, тем не менее, может оказать существенное влияние на прогнозируемую величину.

После осуществления прогноза аналитик может определить значимость участвовавших в нем влияющих факторов по изменению функции оценки и выходных сигналов системы с целью окончательной коррекции участвующих в прогнозе влияющих факторов.

Достаточно часто возникает ситуация, когда в качестве влияющего фактора или прогнозируемой величины полезно использовать информационно-торговые данные в преобразованном с помощью некоторой функции виде. Например, в качестве значимого влияющего фактора при прогнозе цены часто используется та же самая цена, но с однодневным

сдвигом. Поэтому для преобразования влияющих факторов и прогнозируемых величин был определен ряд операций, которые можно применять в любой последовательности. Кроме того, с помощью соответствующих последовательностей данных операций реализуются все наиболее популярные инструменты технического анализа.

Процесс определения величин, участвующих в прогнозе, как в качестве значимых влияющих факторов, так и в качестве прогнозируемой величины, является наиболее субъективным и трудоемким. И, естественно, нет необходимости повторять его каждый день для всех интересующих аналитика финансовых инструментов. Существует возможность сохранения перечня выбранных влияющих факторов, участвующих в прогнозе, и выполненных с ними преобразований для некоторого финансового инструмента или группы инструментов.

1.5.3. Применение нейронных сетей для прогнозирования курсов валют

Пример прогнозирования валютных курсов швейцарского франка к доллару и швейцарского франка к немецкой марке

Такое моделирование с использованием нейронных сетей и технической базы Sun SPARCstation LX провели специалисты компании Logica по заказу банка Chemical Bank. Выбор именно этих валют в то время объяснялся высоким уровнем подвижности первого соотношения и малым – второго (до кризиса в 1993 году). Данные о динамике кросс-курсов этих валют собирались с 1 октября 1992 года по 1 октября 1993 года, при этом ценовые прогнозы характеризовались пятью категориями: большой рост, малый рост, без изменений, малый спад, большой спад. В итоге применяемая нейронная сеть позволила синтезировать прогноз за вышеупомянутый период 55,0 % данных, совпавших с реальными, по первому соотношению валют и 23,0 % – по второму.

Пример прогнозирования курса украинского карбованца к доллару

Следующий пример иллюстрирует результаты прогнозирования курса американского доллара по отношению к украинскому карбованцу (UKB/USD).

Исследования проводились на основе модели сети с обратным распространением. Целью экспериментов было прогнозирование курса UKB/USD. Для достижения данной цели было проведено исследование влияния представления исторических и прогнозируемых данных на ошибку прогнозирования. Также были рассмотрены вопросы влияния структуры нейронной сети на скорость обучения и ошибку прогнозирования. При этом ставились следующие задачи:

- поиск значимых влияющих факторов;
- поиск оптимального представления статистических данных о курсе;
- поиск оптимального представления результата прогнозирования;

- поиск оптимального размера окна «скольжения»;
- поиск оптимальной структуры сети.

Прогнозирование курса UKB/USD проводилось на основе временной последовательности ежедневных данных о курсе. Такой подход к прогнозированию основан на идее американских экономистов, что для прогнозирования некоторых экономических показателей вполне достаточно исследования истории их изменения. Успешное применение данного подхода другими исследователями для прогнозирования курсов DM/USD и SUR/USD позволяет надеяться на успех прогнозирования UKB/USD.

Исходными данными для экспериментов служили ежедневные измерения курса UKB/USD с 15.06.93 по 26.06.95 всего 842 измерений (данные взяты из архивов банка Porto-Franco). Прогнозировалось среднее значение курса за день (среднее арифметическое дневных курсов покупки и продажи).

Каждый из экспериментов состоял из несколько этапов:

1. Формирование обучающей выборки. На этом этапе определялся вид представления исторических и прогнозируемых данных, осуществлялось формирование блока представительских (обучающих) выборок. Большинство проведенных экспериментов было направлено на прогноз не фактического курса валют, а его относительного изменения:

$$\sigma_{Kt} = (K_{t+1} - K_t)/K_t \ . \qquad (1.1)$$

2. Обучение нейронной сети с использованием сформированного на первом этапе блока обучающих выборок. Качество обучения характеризовалось ошибкой обучения, определяемой как суммарное квадратичное отклонение значений на выходах нейронной сети в обучающей выборке от реальных значений, полученных на выходах нейронной сети. Критерий прекращения обучения – 1500 итераций или уменьшение ошибки на выходах сети на два порядка, по сравнению с первичной ошибкой. В том случае, если при описании опыта не указано, что произошло снижение ошибки на два порядка, обучение останавливается по первому критерию.
3. Третий этап – тестирование нейронной сети. Определяется качество прогнозирования при подаче на вход 4,0-5,0 % наборов из обучающей выборки. Эксперимент является успешным, если относительная достоверность не менее 80,0 %.
4. На четвертом этапе осуществляется пробное прогнозирование. На входе нейронной сети – наборы, которые не были внесены в обучающую выборку, но результат по которым (прогноз) известен.

1.5.4. Ограничения и недостатки, связанные с использованием нейронных сетей для прогнозирования

1. Для эффективного прогнозирования, как правило, необходим некоторый минимум наблюдений (более пятидесяти и даже ста). Однако существует много задач, когда такое количество статистических данных недоступно. Например, при производстве сезонного товара, статистики предыдущих сезонов недостаточно для прогноза на текущий сезон из-за изменения стиля продукта, политики продаж и т.д. Даже при прогнозировании потребностей в достаточно стабильном товаре на основе информации о ежемесячных продажах невозможно накопить статистику за период от 50 до 100 месяцев. Для сезонных процессов эта проблема еще более выражена: каждый сезон фактически представляет собой одно наблюдение. Следует отметить, что удовлетворительная модель прогноза с использованием нейронной сети все же может быть построена даже в условиях нехватки данных. При этом модель будет уточняться при поступлении в нее свежих данных.
2. Другим недостатком моделей на основе нейронных сетей являются значительные временные затраты для достижения удовлетворительного результата. Эта проблема не столь существенна, если исследуется небольшое число временных последовательностей, однако обычно прогнозирующая система в области управления производством включает от нескольких сотен до нескольких тысяч временных последовательностей. Отметим, что завышенные ожидания эффекта от внедрения нейронных сетей в ряд финансовых структур в США и Великобритании не оправдались. Так, один крупный инвестиционный банк на Уолл-стрит потратил более 1,0 млн долларов на разработку такой системы для оптимизации финансовых операций, однако, спустя некоторое время вынужден был вернуться к старой системе. Основной причиной неудачи стал недостаточный по сравнению с ожидаемым уровень производительности, полученный в результате внедрения системы.
3. Обучить и эксплуатировать нейронную сеть для решения многих задач, как правило, может и не специалист, но надежно интерпретировать результаты, а также численно оценивать значимость получаемых прогнозов способны специалисты, имеющие навыки в моделировании нейронных сетей.

1.5.5. Программные продукты прогнозирования на основе нейронных сетей

Импульсом для более широкого использования нейронных сетей в финансовых прогнозах стало появление в 1990 г. системы моделирования

нейронных сетей Brain Maker компании California Scientific Software. Данный программный продукт – наиболее продаваемый в своем классе – имеет следующие достоинства:

- используемая модель нейронной сети является надежной и удобной при прогнозировании в сфере бизнеса и финансов;
- для его освоения от аналитика не требуется углубленных знаний в области математики или программирования;
- эффективен при работе в случаях, когда правила, по которым изменяется прогнозируемая величина, неизвестны и трудновыявляемы.

Рассмотрим также ряд особенностей и затруднений, связанных с использованием данного и других подобных программных продуктов нейросетевого моделирования:

1. На фондовом рынке лишь немногие из специалистов успешно справляются с эффективной настройкой нейросимуляторов особенно в тех случаях, когда к прогнозированию приходится привлекать малозначимые влияющие факторы и требуется правильно интерпретировать результаты настройки нейронной сети. Для эффективного использования нейросимуляторов необходимо также хорошо понимать сущность моделируемого процесса.
2. При использовании нейронной сети необходимо учитывать влияние детерминированной периодической функции называемой в теории временных рядов «аддитивной сезонной компонентой» и определяемой методами спектрального анализа. Период сезонной компоненты составляет от 7 до 14 дней. Она может учитывать, например, то, что в первые два–три дня каждого месяца обычно наблюдается локальный подъем котировок акций, а в середине месяца существуют дни, когда на денежный рынок оказывают влияние обязательства по контрактам на куплю–продажу валюты по заранее оговоренной цене и т. д. На этапе прогноза сезонная компонента может автоматически добавляться в одну из колонок электронной таблицы с данными и, таким образом, учитываться в нейросимуляторе при оценке прогнозируемого приращения котировок.
3. Практика работы с нейросимуляторами на финансовом рынке свидетельствует о том, что создание и тщательное ведение обширной, постоянно обновляемой и хорошо структурированной базы финансовых, макроэкономических и политических данных крайне важно, поскольку они существенно влияют на ситуацию и качество прогноза. Так как ситуация на рынке непрерывно изменяется, то и набор значащих влияющих факторов (или их порядок внутри этого набора) также изменяется во времени. В связи

с этим, нейронную сеть необходимо время от времени настраивать и обучать заново.

4. Наличие подробной документации крайне важно при работе с нейросимулятором. Документация обычно включает подробное описание методов и примеров, индексный и предметный указатели, а также обучающий курс. Некоторые компании–разработчики нейросимуляторов поддерживают «горячую линию» по телефону и Интернет, а также проводят семинары пользователей по обучению приемам эффективной работы с нейросимуляторами.

1.5.6. Прогнозирование потребления электроэнергии

Система анализа данных о потреблении электроэнергии (компания ZSolutions) использует данные, полученные в результате обработки показаний счетчиков частных и корпоративных клиентов. Измерения проводятся каждые 15 минут, причем известно, что некоторые из них – неверные. С помощью нейронных сетей был построен алгоритм выявления неверных измерений, а также алгоритм прогнозирования потребления энергии в зимний период времени.

Использование данного прогноза позволило энергетической компании применить гибкую тарифную политику и сократить риск возникновения энергетического кризиса в регионе.

1.5.7. Прогнозирование свойств полимеров [9]

Реализована технология прогнозирования свойств материалов в химических полимерных производствах с помощью нейронных сетей (компании Aspen Technology и NeuralWare Inc., 1997 г.). Данный подход оказался более эффективным и дешевым, чем разработка теоретической модели полимеров.

Так, с помощью нейросимулятора NeuroShell разработан новый сорт безопасного стекла (компания DuPont).

В заключение раздела отметим, что прогнозирование цены нефти Urals – одна из наиболее интересных задач, связанная с применением нейронных сетей для целей прогноза. В виду того, что такой прогноз имеет для России принципиальное значение: цена существенно влияет на исполнение бюджета и уровень жизни – успехи в развитии нейросетевых технологий для прогнозировании в финансовой сфере трудно переоценить.

1.6. Проблемы развития нейронных сетей

Рассмотрим ряд проблем, стоящих сегодня на пути широкого распространения нейросетевых технологий [8].

1. Большинство применяемых нейронных сетей представляют сети обратного распространения – наиболее популярного современного алгоритма. В свою очередь, алгоритм обратного распространения не свободен от недостатков. Прежде всего не существует гарантии, что нейронная сеть может быть обучена за конечное время: зачастую усилия и затраты машинного времени на обучение, пропадают напрасно. Когда это происходит, обучение повторяется – без всякой уверенности, что результат окажется лучше.
2. Нет также уверенности, что сеть обучится наилучшим возможным образом. Алгоритм обучения может попасть в «ловушку» так называемого локального минимума ошибки, и наилучшее решение не будет получено.
3. Разработано много других алгоритмов обучения нейронных сетей, имеющих свои преимущества, однако, следует отметить, что все они не свободны от ограничений.
4. Разработчики склонны преувеличивать свои успехи и замалчивать неудачи, создавая зачастую о нейронных сетях и нейрокомпьютерах необъективное впечатление. Поэтому предприниматели, желающие основать новые компании в области нейросетевых технологий, должны предельно четко представлять пути развития того или иного проекта и пути получения прибыли.
5. Таким образом, существует опасность, что нейросетевые технологии начнут продаваться и покупаться раньше, чем придет их время, обещая потребительские и функциональные возможности, которые пока невозможно достигнуть. Если это произойдет, то технология в целом может пострадать от потери кредита доверия и вернется к периоду невостребованности семидесятых годов.
6. Существует проблема неспособности традиционных искусственных нейронных сетей «объяснить», как они решают задачу. Это напоминает нашу неспособность объяснить, как мы узнаем человека, несмотря на расстояние, освещение и прошедшие годы.
7. Технология требует улучшения существующих методов и расширения теоретических основ, для того чтобы нейронные сети полностью реализовали свои потенциальные возможности.
8. Прежде чем искусственные нейронные сети можно будет использовать для решения задач, где поставлены на карту человеческие жизни или важные народнохозяйственные объекты, должны быть решены вопросы надежности ИНС.

2. ОСНОВНЫЕ ПОНЯТИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

2.1. Модель искусственного нейрона

2.1.1. Биологический нейрон

Нервная система и мозг человека состоят из нейронов, соединенных между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от нашей кожи, ушей и глаз к мозгу, процессы мышления и управления действиями – все это реализовано в живом организме как передача электрических импульсов между нейронами.

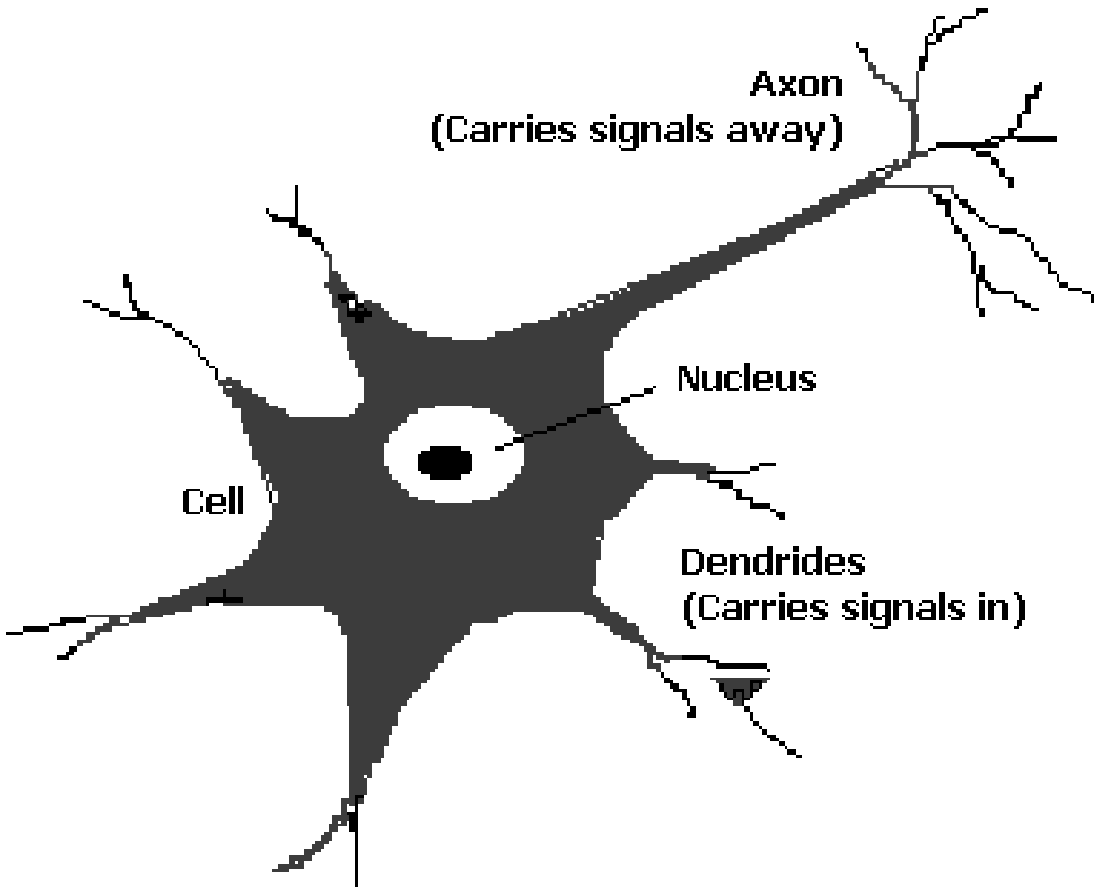


Рис. 2.1. Биологический (или естественный) нейрон

Биологический нейрон (Cell) имеет *ядро* (Nucleus), а также отростки нервных волокон двух типов (рис. 2.1) – *дендриты* (Dendrites), по которым принимаются импульсы (Carries signals in), и единственный *аксон* (Axon), по

которому нейрон может передавать импульс (Carries signals away). Аксон контактирует с дендритами других нейронов через специальные образования – *синапсы* (Synapses), которые влияют на силу передаваемого импульса.

Структура, состоящая из совокупности большого количества таких нейронов, получила название *биологической* (или *естественной*) нейронной сети.

2.1.2. Искусственный нейрон

Искусственный нейрон (далее – нейрон) является основой любой искусственной нейронной сети.

Нейроны представляют собой относительно простые, однотипные элементы, имитирующие работу нейронов мозга. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены и заторможены.

Искусственный нейрон, также как и его естественный прототип, имеет группу синапсов (входов), которые соединены с выходами других нейронов, а также аксон – выходную связь данного нейрона – откуда сигнал возбуждения или торможения поступает на синапсы других нейронов.

Общий вид нейрона представлен на рис 2.2.

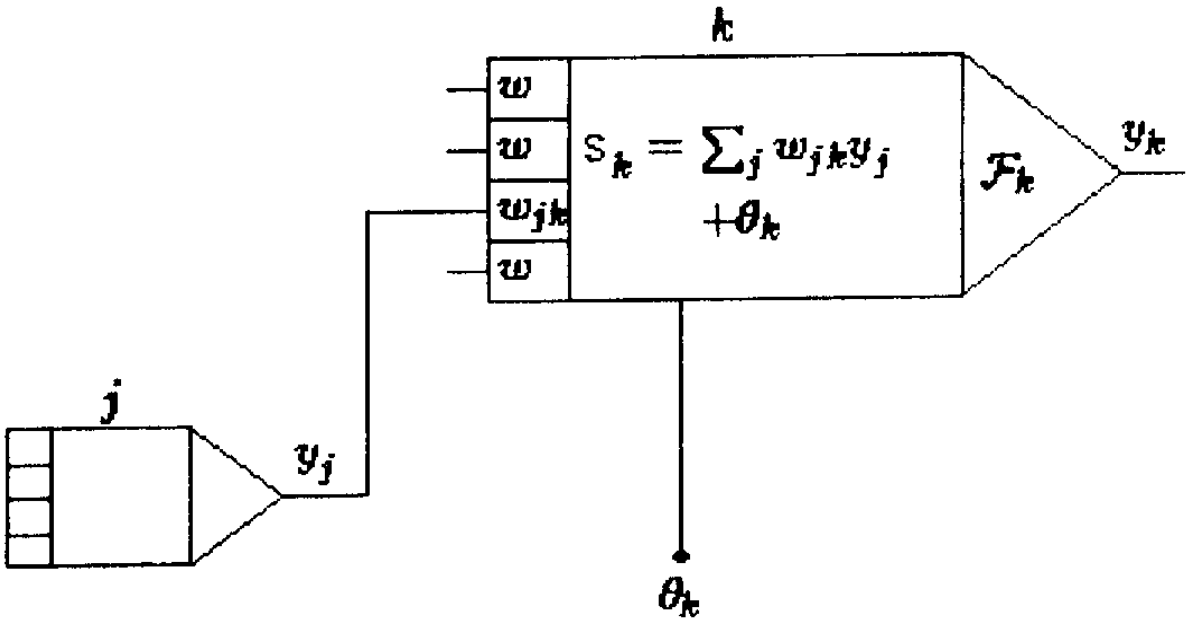


Рис. 2.2. Искусственный нейрон –
простейший элемент искусственной нейронной сети
y_j – сигнал, поступающий от нейрона *j*;
s_k – скалярное произведение вектора входных сигналов и вектора весов;
f_k – функция возбуждения; y_k – выходной сигнал нейрона

Каждый синапс характеризуется величиной *синаптической связи* или *весом* w_i, который по своему физическому смыслу эквивалентен электрической проводимости.

Текущее состояние нейрона определяется как взвешенная сумма его входов:

$$s = \sum_{i=1}^{n} x_i w_i \quad , \tag{2.1}$$

где x – вход нейрона, а w – соответствующий этому входу вес.

2.1.3. Активационная функция

Выход нейрона есть функция его состояния, т.е.

$$y = f(s) \quad . \tag{2.2}$$

Нелинейная функция $f(s)$ называется *активационной, сжимающей* функцией или функцией *возбуждения* нейрона.

Основные разновидности активационных функций, применяемых в нейронных сетях, представлены на рис. 2.3.

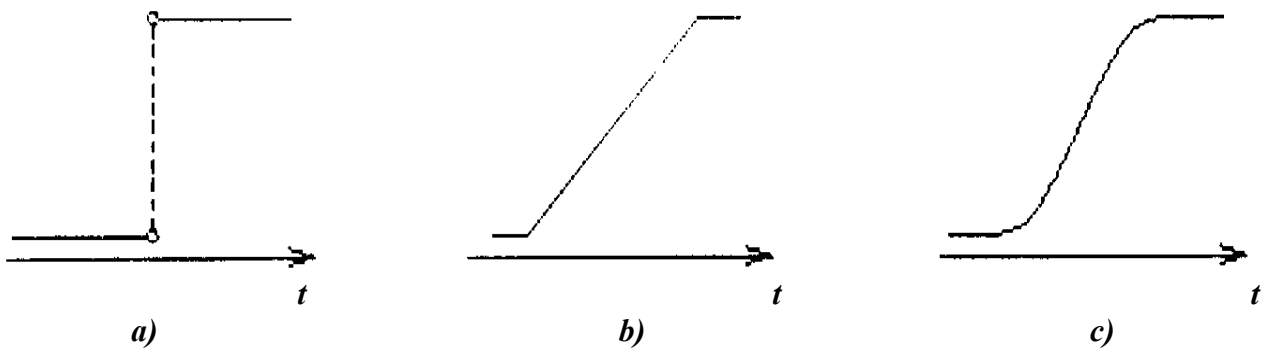


Рис. 2.3. Активационная функция
a) пороговая; b) полулинейная; c) сигмоидальная

В качестве активационной функции часто используется *сигмоидальная (s-образная* или *логистическая*) функция, показанная на рис. 2.3 *c*. Эта функция математически выражается по формуле

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad . \tag{2.3}$$

При уменьшении α сигмоидальная функция становится более пологой, в пределе при α=0 вырождаясь в горизонтальную линию на уровне 0,5; при увеличении α сигмоидальная функция приближается по внешнему виду к функции единичного скачка с порогом T в точке x=0. Из выражения для сигмоидальной функции видно, что выходное значение нейрона лежит в диапазоне [0,1]. Одно из полезных свойств сигмоидальной функции – простое выражение для ее производной, применение которого будет рассмотрено в дальнейшем:

$$f(x) = \alpha f(x)(1 - f(x)). \qquad (2.4)$$

Следует отметить, что сигмоидальная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, сигмоидальная функция обладает свойством усиливать малые сигналы лучше, чем большие, тем самым предотвращая насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоидальная функция имеет пологий наклон.

Выбор структуры нейронной сети осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные, на сегодняшний день конфигурации, описанные, например, в [6, 7, 8].

Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации.

Теоретически число слоев и число нейронов в каждом слое нейронной сети может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется нейронная сеть.

При этом, если в качестве активационной функции для всех нейронов сети используется функция единичного скачка, нейронная сеть называется *многослойным персептроном* (рис. 2.4).

В нейронных сетях, называемых персептронами, используется активационная функция единичного скачка.

2.2. Обучение нейронных сетей

Очевидно, что функционирование нейронной сети, т. е. действия, которые она способна выполнять, зависит от величин синоптических связей. Поэтому, задавшись структурой нейронной сети, отвечающей определенной задаче, разработчик должен найти оптимальные значения для всех весовых коэффициентов w.

Этот этап называется обучением нейронной сети, и от того, насколько качественно он будет выполнен, зависит способность сети решать во время

эксплуатации поставленные перед ней проблемы. Важнейшими параметрами обучения являются: качество подбора весовых коэффициентов и время, которое необходимо затратить на обучение. Как правило, два этих параметра связаны между собой обратной зависимостью и их приходится выбирать на основе компромисса.

В настоящее время все алгоритмы обучения нейронных сетей можно разделить на два больших класса: с учителем и без учителя.

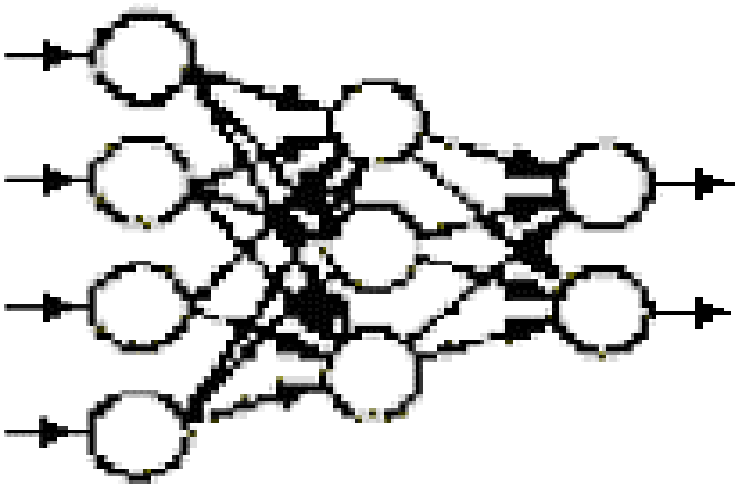


Рис. 2.4. Многослойный персептрон

2.2.1. Обучение с учителем

Нейронной сети предъявляются значения как входных, так и выходных параметров, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей.

Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются *представительской* или *обучающей выборкой*. Обычно нейронная сеть обучается на некотором числе таких выборок. Предъявляется выходной вектор, вычисляется выход нейронной сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в нейронную сеть, и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

2.2.2. Обучение без учителя

Нейронной сети предъявляются только входные сигналы, а выходы сети формируются самостоятельно с учетом только входных и производных от них сигналов.

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в естественном человеческом интеллекте, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Если допустить подобный механизм в человеческом мозге, то откуда тогда возникают желаемые выходы? Обучение без учителя является более правдоподобной моделью обучения в биологической системе. Развитая Кохоненом (п. 2.4) и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами.

Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса нейронной сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов.

Следовательно, выходы подобной сети должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является серьезной проблемой. Обычно не сложно идентифицировать связь между входом и выходом, установленную сетью.

2.3. Нейронные сети обратного распространения

Одним из наиболее распространенных видов нейронных сетей является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя, или в случае первого слоя со всеми входами нейронной сети.

Такие нейронные сети называются *полносвязанными*.

Алгоритм обратного распространения, применяемый для таких структур, заключается в распространение сигналов ошибки от выходов нейронной сети к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Эта процедура обучения нейронной сети и получила название алгоритма обратного распространения.

Согласно методу наименьших квадратов минимизируемой целевой функцией ошибки нейронной сети является

$$E(w) = \frac{1}{2}\sum_{j,p}(y_{j,p}^{(n)} - d_{j,p})^2 , \qquad (2.5)$$

где $y_{j,p}^{(n)}$ – реальное выходное состояние нейрона j выходного слоя n нейронной сети при подаче на ее входы p-го образа; d_{jp} – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем образам, обрабатываемым нейронной сетью. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}} . \qquad (2.6)$$

где w_{ij} – весовой коэффициент синаптической связи, соединяющей i-й нейрон слоя n–1 с j-м нейроном слоя n, η – коэффициент скорости обучения, $0 < \eta < 1$.

Как показано в [8],

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{dy_j}{ds_j}\frac{\partial s_j}{\partial w_{ij}} , \qquad (2.7)$$

где под y_j, подразумевается выход нейрона j, а под s_j – взвешенная сумма его входных сигналов, т. е. аргумент активационной функции.

Так как множитель dy_j/ds_j является производной этой функции по ее аргументу, следовательно, производная активационной функции должна быть определена на всей оси абсцисс. Поэтому функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых нейронных сетей. Как правило, применяются такие гладкие функции, как гиперболический тангенс или классическая сигмоидальная функция с экспонентой. В случае гиперболического тангенса

$$\frac{dy}{ds} = 1 - s^2 . \qquad (2.8)$$

Третий множитель $\partial s_j/\partial w_{ij}$ равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$.

Первый множитель (2.7) раскладывается следующим образом [8]:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{dy_k}{ds_k} \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{dy_k}{ds_k} w_{jk}^{(n+1)} \quad , \tag{2.9}$$

где суммирование по k выполняется среди нейронов слоя $n+1$.
Введем переменную

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \quad . \tag{2.10}$$

Тогда получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя $n+1$.

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{dy_j}{ds_j} \quad . \tag{2.11}$$

Для выходного слоя

$$\delta_l^{(n)} = (y_l^{(n)} - d_l) \frac{dy_l}{ds_l} \quad . \tag{2.12}$$

Запишем (2.6) в развернутом виде:

$$\Delta w_{ij}^{(n)} = -\eta \delta_j^{(n)} y_i^{(n-1)} \quad . \tag{2.13}$$

Для придания процессу коррекции весов инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (2.13) дополняется значением изменения веса на предыдущей итерации:

$$\Delta w_{ij}^{(n)}(t) = -\eta (\mu \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \delta_j^{(n)} y_i^{(n-1)}) \quad , \tag{2.14}$$

где μ – коэффициент инерционности, t – номер текущей итерации.

Таким образом, полный алгоритм обучения нейронной сети с помощью процедуры обратного распространения строится так:

1. При подаче на входы нейронной сети одного из возможных образов в режиме обычного функционирования нейронной сети, когда сигналы распространяются от входов к выходам, рассчитать значения сигналов

$$s_j^{(n)} = \sum_{i=0}^{m} y_i^{(n-1)} w_{ij}^{(n)} \quad , \tag{2.15}$$

где m – число нейронов в слое $n–1$ с учетом нейрона с постоянным выходным состоянием +1, задающего смещение; $y_i^{(n-1)}=x_{ij}^{(n)}$ – i-й вход нейрона j слоя n

$y_j^{(n)} = f(s_j^{(n)})$, где f – сигмоидальная функция; (2.16)

$y_q^{(0)}=I_q$, (2.17)

где I_q – q-я компонента вектора входного образа.

2. Рассчитать $\delta^{(n)}$ для выходного слоя по формуле (2.12), а также по формуле (2.13) или (2.14) изменения весов $\Delta w^{(n)}$ слоя n.

3. Рассчитать по формулам (2.11) и (2.13) (или (2.12) и (2.14)) соответственно $\delta^{(n)}$ и $\Delta w^{(n)}$ для всех остальных слоев, $n=N–1,...1$.

4. Скорректировать все веса в нейронной сети

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) . \quad (2.18)$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – завершение обучения.

Нейронной сети на шаге 1 попеременно в случайном порядке предъявляются все представительские выборки, чтобы нейронная сеть, образно говоря, не забывала одни по мере запоминания других (рис. 2.5).

Эффективность обучения заметно снижается когда выходное значение $y_i^{(n-1)}$ в (2.13) стремится к нулю. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться [8], поэтому область возможных значений выходов нейронов [0;1] желательно сдвинуть в пределы [–0,5;+0,5], что достигается простыми модификациями активационных функций, например: сигмоидальная функция с экспонентой преобразуется к виду

$$f(x) = -0{,}5 + \frac{1}{1+e^{-\alpha x}} . \quad (2.19)$$

Рассмотрим вопрос о числе образов, предъявляемых на входы нейронной сети, которые она способна научиться распознавать (емкость нейронной сети). Для нейронной сети с одним скрытым слоем, детерминистская емкость нейронной сети C_d оценивается как

$$N_w/N_y < C_d < N_w/N_y \log(N_w/N_y) , \quad (2.20)$$

где N_w – число подстраиваемых весов, N_y – число нейронов в выходном слое.

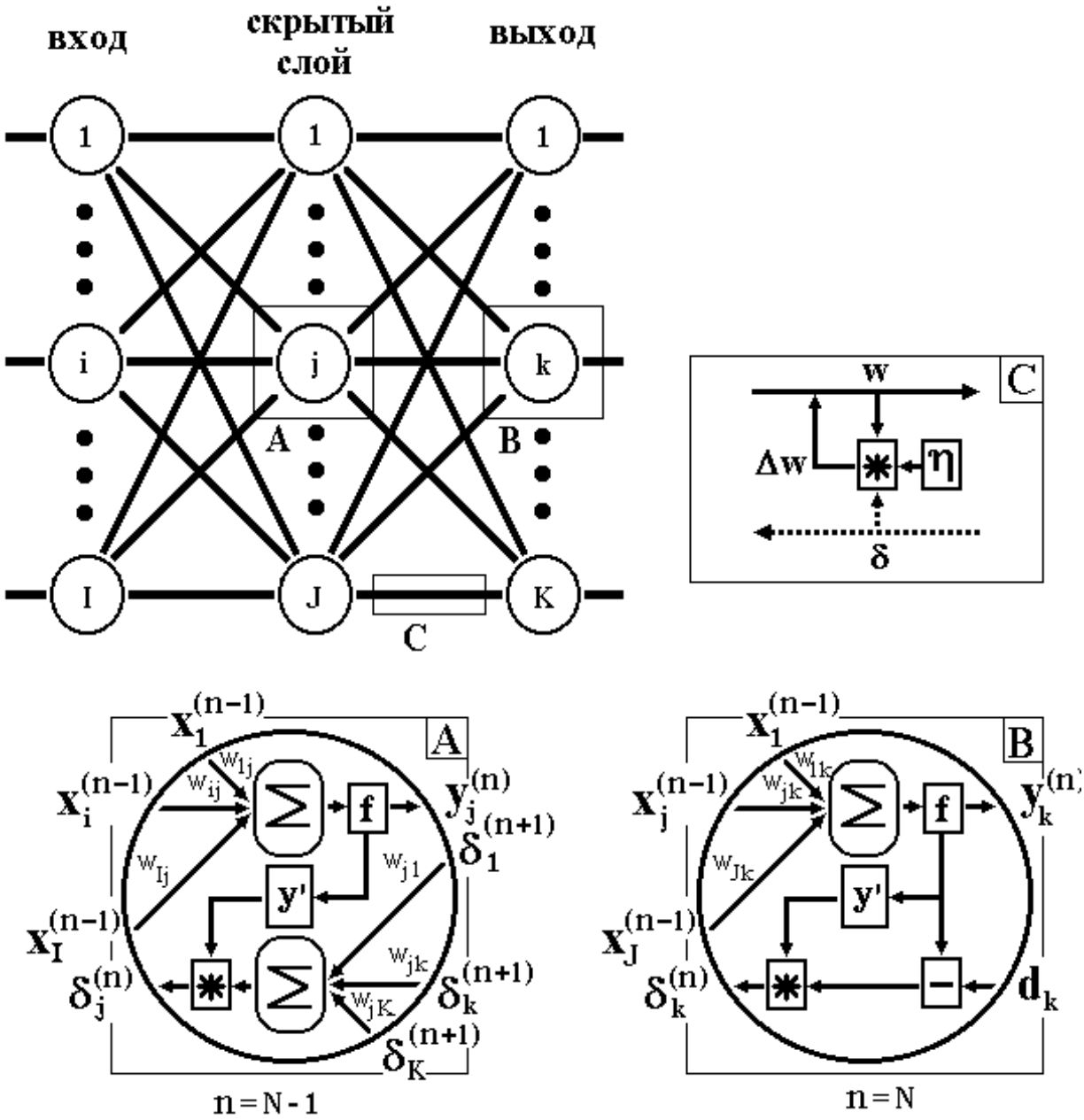


Рис. 2.5. Диаграмма сигналов при обучении нейронной сети по алгоритму обратного распространения

Следует отметить, что данное выражение получено с учетом некоторых ограничений.

Во-первых, число входов N_x и нейронов в скрытом слое N_h должно удовлетворять неравенству $N_x+N_h > N_y$.

Во-вторых, $N_w/N_y > 1000$. Однако вышеприведенная оценка выполнялась для нейронных сетей с активационными функциями нейронов в виде порога, а емкость сетей с гладкими активационными функциями,

например (2.19), обычно больше. Кроме того, фигурирующее в названии емкости прилагательное «детерминистский» означает, что полученная оценка емкости подходит абсолютно для всех возможных входных образов, которые могут быть представлены N_x входами. Распределение входных образов, как правило, обладает некоторой регулярностью, что позволяет нейронной сети проводить обобщение и, таким образом, увеличивать реальную емкость. Так как распределение образов, в общем случае, заранее не известно, можно говорить о такой емкости только предположительно, но обычно она раза в два превышает емкость детерминистскую.

Рассмотрим вопрос о размерности выходного слоя нейронной сети, выполняющего окончательную классификацию образов. Для разделения множества (классификации) входных образов, например, по двум классам достаточно всего одного выхода. При этом каждый логический уровень – «1» и «0» – будет обозначать отдельный класс. На двух выходах можно закодировать уже 4 класса и так далее. Однако результаты работы нейронной сети, организованной таким образом, «под завязку», не очень надежны. Для повышения достоверности классификации желательно ввести избыточность путем выделения каждому классу одного нейрона в выходном слое или, что еще лучше, нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности – высокой, средней или низкой, что позволит проводить классификацию входных образов, объединенных в нечеткие (размытые или пересекающиеся) множества. Это свойство приближает нейронные сети к естественному человеческому интеллекту.

Такая нейронная сеть имеет несколько ограничений. Во-первых, в процессе обучения может возникнуть ситуация, когда большие положительные или отрицательные значения весовых коэффициентов сместят рабочую точку на сигмоидальной функции многих нейронов в область насыщения. Малые величины производной от активационной функции в соответствии с (2.11) и (2.12) приведут к остановке обучения нейронной сети. Во-вторых, применение метода градиентного спуска не гарантирует, что будет найден глобальный, а не локальный минимум целевой функции. Эта проблема связана еще с одной, а именно – с выбором коэффициента скорости обучения. Доказательство сходимости обучения в процессе обратного распространения основано на производных, т. е. приращениях весов и, следовательно, скорость обучения должна быть бесконечно малой, однако в этом случае обучение будет происходить неприемлемо медленно. С другой стороны, слишком большие коррекции весов могут привести к постоянной неустойчивости процесса обучения. Поэтому, коэффициент η обычно выбирается меньше 1, но не очень малым, например, 0,1, и он может постепенно уменьшаться в процессе обучения. Кроме того, для исключения случайных попаданий в локальные минимумы иногда, после того как значения весовых коэффициентов стабилизируются, η

кратковременно можно значительно увеличить, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет алгоритм в одно и то же состояние нейронной сети, можно более или менее быть уверенным в том, что найден глобальный минимум ошибки.

2.4. Карты Кохонена

2.4.1. Определение

Нейронные сети Кохонена или *самоорганизующиеся карты Кохонена* (Kohonen's Self-Organizing Maps) предназначены для решения задач автоматической классификации, когда обучающая последовательность образов отсутствует [12]. Соответственно отсутствует и фиксация ошибки, на минимизации которой основаны алгоритмы обучения, например, алгоритм обратного распространения ошибки (Backpropagation).

Сеть Кохонена – это двухслойная нейронная сеть, содержащая *входной слой* (слой входных нейронов) и *слой Кохонена* (слой активных нейронов). Слой Кохонена может быть: одномерным, двумерным или трехмерным.

В первом случае активные нейроны расположены в цепочку. Во втором случае они образуют двухмерную сетку (обычно в форме квадрата или прямоугольника), а в третьем случае они образуют трехмерную конструкцию.

В силу отсутствия обучающей последовательности образов, для каждого из которых известна от учителя принадлежность к тому или иному классу, определение весов нейронов слоя Кохонена основано на использовании алгоритмов классической классификации (кластеризации или самообучения).

2.4.2. Принцип работы сети Кохонена

На рис. 2.6 приведен пример топологической карты сети Кохонена, содержащей входной слой и слой Кохонена. Нейроны входного слоя служат для ввода значений признаков распознаваемых образов. Активные нейроны слоя Кохонена предназначены для формирования областей (кластеров) различных классов образов.

На этом рисунке показаны связи всех входных нейронов лишь с одним нейроном слоя Кохонена. Каждый нейрон слоя Кохонена также соединен с соседними нейронами.

Поясним основной принцип работы сети Кохонена.

Введем следующие обозначения (рис. 2.6):

$$\mathbf{W}_j = (w_{j1}, w_{j2}, \ldots, w_{jn})^T, \quad j = \overline{1, m} \tag{2.21}$$

– вектор весовых коэффициентов j-го нейрона слоя Кохонена,

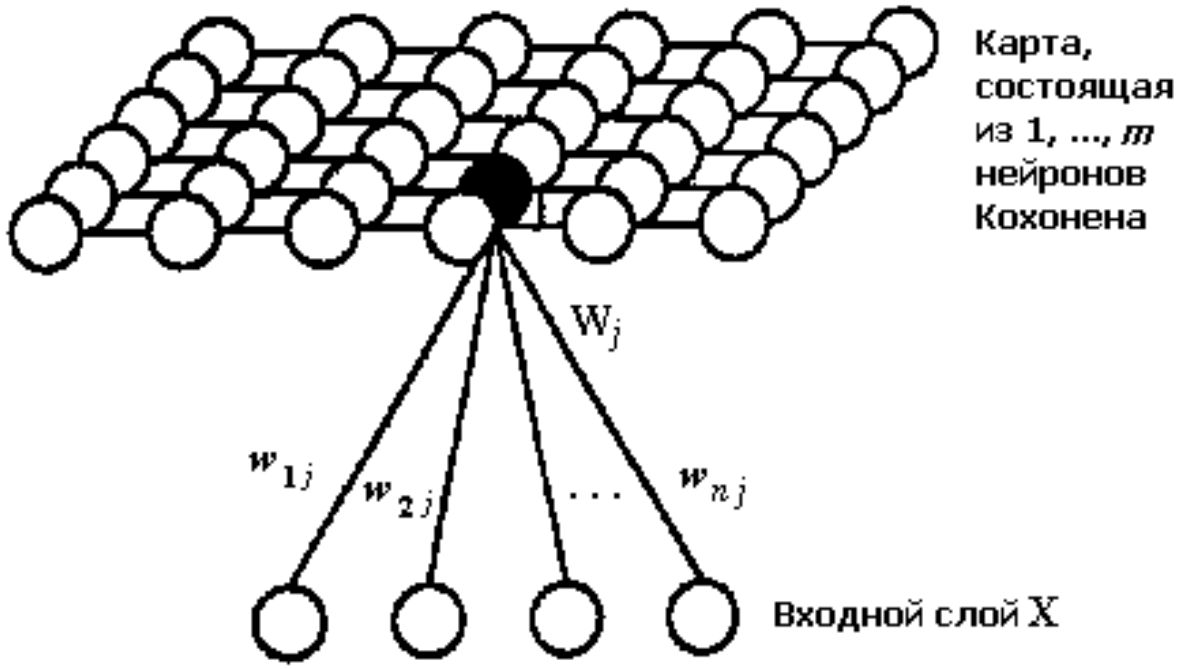


Рис. 2.6. Топологическая карта сети Кохонена

$$\mathbf{X} = (x_1, x_2, \ldots, x_n)^T \qquad (2.22)$$

– входной вектор или вектор значений признаков некоторого образца.

На стадии обучения (точнее самообучения) сети входной вектор $\mathbf{X}^c$ попарно сравнивается со всеми векторами $\mathbf{W}_j$ всех нейронов слоя Кохонена. Вводится некоторая функция близости (например, в виде эвклидова расстояния). Активный нейрон с номером слоя Кохонена, для которого значение функции близости $d\ (\mathbf{X},\ \mathbf{W}_c)$ между входным вектором $\mathbf{X}$, характеризующим некоторый образ, и вектором $\mathbf{W}_c$ максимально, объявляется «победителем». При этом образ, характеризующийся вектором $\mathbf{X}$, относится к классу, который представляется «нейроном-победителем». В результате осуществляется преобразование n-мерного входного пространства $\mathbf{R}^n$ на m-мерную сетку (слой Кохонена).

Следует подчеркнуть, что это отображение реализуется в результате рекуррентной (итеративной) процедуры самообучения (Unsupervised Learning). Отличительная особенность этого отображения – формирование кластеров (Cluster) или классов. По завершении процесса самообучения на стадии реального использования сети Кохонена неизвестные входные образы относятся к одному из выявленных кластеров (классов).

Возникает естественный вопрос: как возникает указанное отображение топологической карты? Для ответа на него рассмотрим алгоритм

самообучения сети Кохонена, полагая, что ее входной слой содержит n входных нейронов, а ее слой Кохонена – m активных нейронов.

Для определения расстояния между входным вектором $\mathbf{X}$ (2.22) и весовым вектором $\mathbf{W}_j$ (2.21) j-го нейрона слоя Кохонена можно использовать различные функции близости (обычно эвклидово расстояние).

При этом "выигрывает" тот нейрон c с весовым вектором $\mathbf{W}_c$, который наиболее близок к входному вектору $\mathbf{X}$:

$$\|\mathbf{X} - \mathbf{W}_c\| = \min_j \|\mathbf{X} - \mathbf{W}_j\| \qquad (2.23)$$

или при использовании функции *index*, определяющей номер минимального расстояния:

$$c = index \min_j \|\mathbf{X} - \mathbf{W}_j\|. \qquad (2.24)$$

При использовании скалярного произведения

$$\mathbf{X}^T\mathbf{W}_j = \sum_{i=1}^{n} x_i\, w_{ij} = net_j = z_j \qquad (2.25)$$

"выигрывает" нейрон с максимальным значением этого произведения.

На стадии самообучения сети Кохонена осуществляется коррекция весового вектора не только «нейрона-победителя», но и весовых векторов остальных активных нейронов слоя Кохонена, однако в существенно меньшей степени – в зависимости от удаления от «нейрона-победителя». При этом форма и величина окрестности вокруг «нейрона-победителя», весовые коэффициенты нейронов которой также корректируются, в процессе обучения изменяются. Сначала начинают с очень большой области – она, в частности, может включать все нейроны слоя Кохонена.

Изменение весовых векторов осуществляется по правилу:

$$w_j(t+1) = w_j(t) + \eta(t)\, d_{cj}(t)\, [x(t) - w_j(t)], \qquad \overline{j=1,m} \qquad (2.26)$$

где $w_j(t)$ – значение весового вектора на t-м шаге самообучения сети, $d_{cj}(t)$ – функция близости между нейронами слоя Кохонена (neighborhood Kernel) и $\eta(t)$ – изменяемый во времени коэффициент коррекции.

В качестве $\eta(t)$ обычно выбирается монотонно уменьшающаяся функция

$$0 < \eta(t) < 1,$$

т. е. алгоритм самообучения начинается сравнительно большими шагами адаптации и заканчивается относительно небольшими изменениями.

Обратим внимание, что в соответствии с (2.26) изменение того или иного весового вектора $\mathbf{W}_j$ пропорционально расстоянию между входным вектором $\mathbf{X}$ и этим весовым вектором $\mathbf{W}_j$.

В качестве примера рассмотрим сеть Кохонена с одномерным слоем Кохонена (рис. 2.7). На рис. 2.6 отображено движение весовых векторов нейронов слоя Кохонена. К входному вектору $\mathbf{X}$ ближе всех расположен весовой вектор $\mathbf{W}_5$ для нейрона c=5. Этот весовой вектор изменяется наиболее сильно: он в большей степени приближается к входному вектору $\mathbf{X}$. На втором месте по степени близости находятся весовые векторы $\mathbf{W}_4$ и $\mathbf{W}_6$. Изменение их – второе по силе (степени). Весовые векторы $\mathbf{W}_3$ и $\mathbf{W}_7$ также изменяются, однако в существенно меньшей степени.

Нейроны 1, 2, 8 и 9 расположены вне окрестности вокруг «нейрона-победителя» c=5, поэтому их весовые векторы оставляются без изменения после показа сети образца, характеризующегося вектором $\mathbf{X}$.

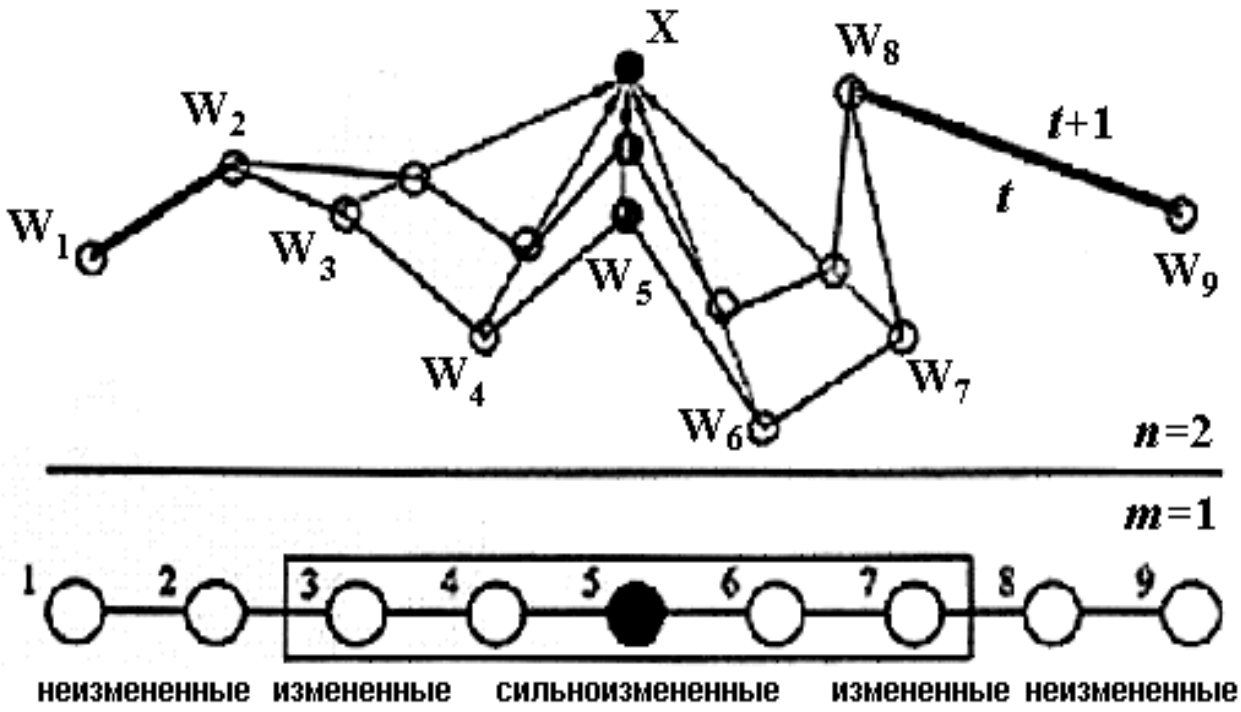


Рис. 2.7. Сеть Кохонена с одномерным слоем Кохонена

Приведем примеры функций близости (двумерного слоя Кохонена). Расстояние между нейронами i и j в двумерном пространстве:

$$z = \sqrt{(k_{i1}-k_{j1})^2 + (k_{i2}-k_{j2})^2}\ , \qquad (2.27)$$

где k_{i1} и k_{i2} – координаты по оси x и оси y нейрона i; k_{j1} и k_{j2} – аналогично для нейрона j. При этом можно использовать следующие функции близости:

$$d_{\text{Gauss}}(z) = e^{-z2}\ ; \qquad (2.28)$$

$$d_{\text{mexican-hat}}(z) = (1-z^2)\, e^{-z2}\ ; \qquad (2.29)$$

$$d_{\cos}(z) = \begin{cases} \cos(z\pi/2), \text{ для } z < 1 \\ 0\ . \end{cases} \qquad (2.30)$$

Как отмечено выше, изменение весовых векторов $\mathbf{W}_j$ осуществляется в направлении входного вектора $\mathbf{X}$ многократно. В процессе самообучения варьируется как коэффициент коррекции η, так и радиус d, задающий окрестность вокруг «нейрона-победителя».

2.4.3. Сходимость алгоритма самообучения

При рассмотрении проблемы сходимости ограничимся одномерным случаем, когда имеется лишь один вход. Пусть $[a, b]$ – область значений для входа (замкнутый интервал). Покажем, что алгоритм самообучения переводит вес x в середину интервала (рис. 2.8).

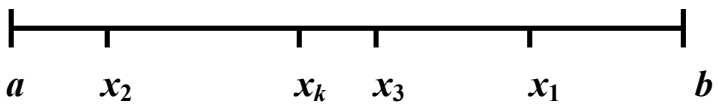


Рис. 2.8. Область значений для входа

Пусть x_1 – начальное значение веса одного активного нейрона слоя Кохонена. Это значение x_1 выбирается случайно: интервал $[a, b]$ разбивается на 2 подинтервала $[a, x_1]$ и $[x_1, b]$. Каждое изменение x определяется его расстоянием до a и до b:

$$dx/dt = \eta\,(b-x)/2 + \eta(a-x)/2 = \eta((a+b)/2-x)\ . \qquad (2.31)$$

Изменение веса x в точке x_1

$$\Delta x_1 = \eta\,((a+b)/2 - x_1)\ . \qquad (2.32)$$

Обозначим $y_i = x_i - (a+b)/2$, тогда соотношение (2.32) можно представить так:

$$\Delta x_1 = -\eta\, y_1\ . \qquad (2.33)$$

Определим математическое ожидание для значения веса x_2 на следующем шаге алгоритма самообучения:

$$x_2 = x_1 + \Delta x_1 = (a+b)/2 + y_1 - \eta y_1 = (a+b)/2 + y_1(1-\eta)\ . \qquad (2.34)$$

Аналогично можно определить и x_3:

$$x_3 = (a+b)/2 + y_1(1-\eta)^2 \qquad (2.35)$$

или в общем случае:

$$x_k = (a+b)/2 + y_1(1-\eta)^{k-1}. \qquad (2.36)$$

При $\eta \in [0, 1]$ значение x_k сходится к $(a+b)/2$.

Расширим рассмотренный одномерный случай и предположим, что одномерный слой Кохонена (линейка) содержит не один нейрон (как ранее), а m активных нейронов с весами $x_1, x_2, \ldots, x_m$. Предположим, что эти веса упорядочены

$$0 < x_1 < x_2 < \ldots < x_m < b$$

и равномерно распределены в интервале $[a, b]$. В этом случае в процессе самообучения весовые коэффициенты сходятся к значениям (рис. 2.9):

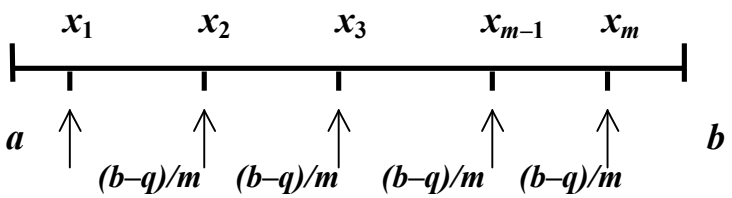


Рис. 2.9. Распределение весовых коэффициентов

$$x_i = a + (2i–1)\,(b–q)/2m, \qquad i = 1, 2, \ldots, m\,. \tag{2.37}$$

Обратим внимание, что точки (2.37) для весов x_i, $i = 1, 2, \ldots, m$ определяют наиболее устойчивые позиции, ибо

$$dx_i/dt = 0\,. \tag{2.38}$$

В двумерном случае слой Кохонена содержит m x m активных нейронов, а областью определения для входов является декартово произведение $[a, b]$ x $[c, d]$, т. е. входной слой содержит 2 нейрона. В этом случае весовой вектор каждого нейрона слоя Кохонена имеет две составляющие – по числу входов. Каждый нейрон слоя Кохонена также характеризуется двумя координатами – по оси абсцисс и по оси ординат.

Подобно одномерному случаю можно показать, что координаты весовых векторов нейронов слоя Кохонена на оси абсцисс в процессе самообучения равномерно распределяются в интервале $[a, b]$:

$$a < w_1^{\,1} < w_1^{\,2} < \ldots < w_1^{\,m} < b\,. \tag{2.39}$$

Аналогично для координат этих векторов по оси ординат:

$$c < w_2^{\,1} < w_2^{\,2} < \ldots < w_2^{\,m} < d\,. \tag{2.40}$$

В результате самообучения сети Кохонена весовые векторы нейронов слоя Кохонена равномерно распределяются во входном пространстве.

2.5. Сети Хопфилда

2.5.1. Определение

Американским исследователем Хопфилдом в 80-х годах предложен специальный тип нейронных сетей.

В отличие от сетей с прямыми связями (Feedforward Networks или FF - Nets), сети Хопфилда являются *рекуррентными* или *сетями с обратными связями* (Feedback Networks).

Сети Хопфилда обладают следующими свойствами [12]:

1. Симметрия дуг: сети содержат n нейронов, соединенных друг с другом. Каждая дуга (соединение) характеризуется весом w_{ij}, причем:

$$\forall i, j \in N : i \neq j : \exists_1 w_{ij}\ ,$$

где N – множество нейронов $N = \{1, 2, \ldots, n\}$.

2. Симметрия весов: вес соединения нейрона n_i с нейроном n_j равен весу обратного соединения:

$$w_{ij} = w_{ji} \ ; \qquad w_{ii} = 0 \ . \tag{2.41}$$

3. Бинарные входы: сеть Хопфилда обрабатывает бинарные входы {0,1} или {-1,1}. В литературе встречаются модели сетей как со значениями 0 и 1, так и –1, 1. Для структуры сети это безразлично. Однако формулы для распознавания образов (изображений) при использовании значений –1 и 1 для входов и выходов нейронов сети Хопфилда получаются нагляднее, поэтому эти значения и предполагаются ниже.

<u>Определение.</u> Бинарная сеть Хопфилда определяется симметричной матрицей с нулевыми диагональными элементами, вектором* T *порогов нейронов и знаковой функцией активации или выхода нейронов.

Каждый вектор **O** с компонентами –1 или 1, удовлетворяющий уравнению

$$\mathbf{O} = f(\mathbf{WO} - \mathbf{T}) \ , \tag{2.42}$$

называется образом для сети Хопфилда.

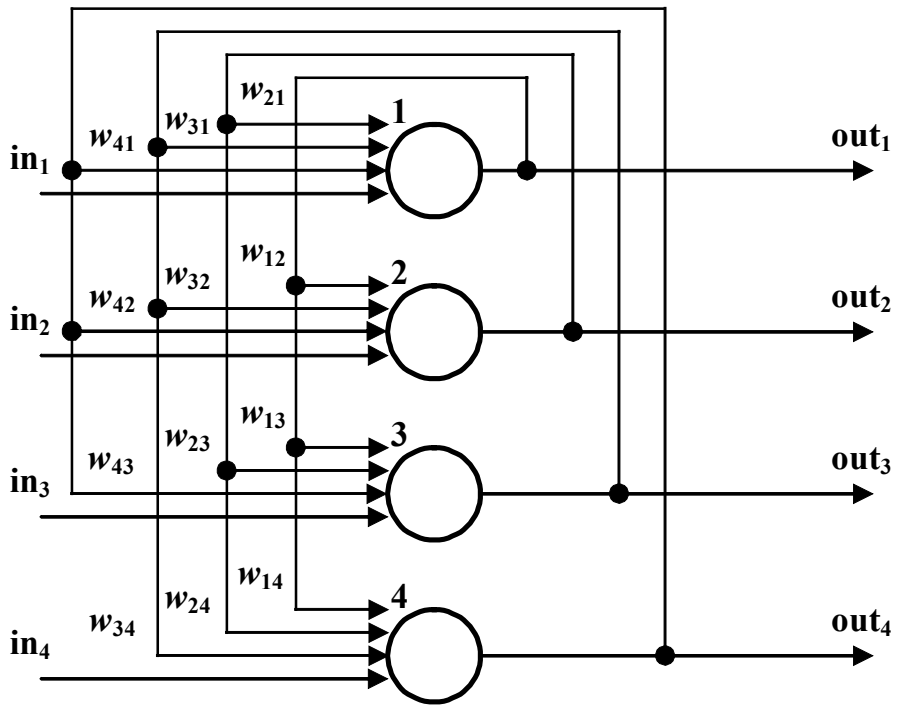


Рис. 2.10. Сеть Хопфилда, состоящая из 4 нейронов

Принцип функционирования сетей Хопфилда отличается от ранее рассмотренных. На так называемой Recall-стадии на входы сети подается некоторый образ.

Он остается на входах до тех пор, пока не завершатся изменения состояний сети. В этом случае говорят о сходимости сети [12].Наглядно это можно представить следующим образом. В начале сеть находится на высоком энергетическом уровне, из которого возможны переходы в различные состояния. Затем энергетический уровень нейронной сети уменьшается до тех пор, пока не достигается некоторое конечное состояние. Ниже мы рассмотрим, как обучаются сети Хопфилда и как считывается (извлекается) запомненная в них информация.

Класс сетей Хопфилда содержит только один слой нейронов, причем каждый нейрон соединен с остальными. Обратные связи с выхода нейрона на его же вход отсутствуют. На рис. 2.10 приведен конкретный пример сети Хопфилда из четырех нейронов.

2.5.2. Алгоритм Хопфилда

Обозначим через $\mathbf{X}^s$ образ s-го класса, а через x_{si} – i-ю составляющую вектора $\mathbf{X}^s$. При этом алгоритм Хопфилда может быть описан следующим образом [12].

1. Расчет весов

$$w_{ij} = \begin{cases} \sum_{s=0}^{k-1} x_{si} x_{sj}, i \neq j \\ 0 \qquad\quad , i = j \end{cases}, \tag{2.43}$$

где k – число классов образов.

2. Инициализация сети путем ввода

$$o_i(0) = x_i \ , \quad 1 \le i \le n \ , \tag{2.44}$$

где $o_i(0)$ – выход i-го нейрона в начальный (нулевой) момент времени.

3. Итерационное правило:

Repeat

$$o_i(t+1) = f_h\left(\sum_{i=0}^{n-1} w_{ij}\, o_j(t)\right) \tag{2.45}$$

Until $\forall \quad 1 \le i \le n : o_i(t+1) = o_i(t)$,

где $o_i(t)$ – выход i-го нейрона в момент времени t.

Соотношения (2.44), (2.45) представим в векторной форме:

$$\mathbf{O}(t+1) = f(\mathbf{WO}(t)) \ , \qquad t=1,2,\ldots \ ; \tag{2.46}$$

$$\mathbf{O}(0) = \mathbf{X}(0) \ , \tag{2.47}$$

где $\mathbf{X}$ – входной вектор, $\mathbf{W}$ – симметричная матрица, f – вектор-функция активации или выхода нейронов.

Взвешенная сумма входов j-го нейрона сети Хопфилда:

$$z_j = net_j = \sum_i w_{ij} o_i \ . \tag{2.48}$$

Функция активации или выхода имеет вид

$$o_j = f(z_i) = \begin{cases} 1 & ,для \quad z_j > 0 \\ -1, & для \quad z_j \le 0 \end{cases} \ . \tag{2.49}$$

Итак, пусть имеется некоторый образ, который следует запомнить в сети Хопфилда, тогда веса искомой сети могут быть рассчитаны, т.е. процесс обучения исключается. Если этот образ характеризуется n-мерным вектором $\mathbf{X} = (x_1, \ldots, x_n)^T$, то веса соединений определяются по формулам:

$$w_{ij} = \begin{cases} x_i x_j & ,для \quad i \ne j \\ 0 & ,для \quad i = j \end{cases} \ . \tag{2.50}$$

В этом случае сеть Хопфилда, характеризуемая матрицей $\mathbf{W}$ и порогами $T_i = 0$, i=1, 2, …, n, запоминает предъявленный образ.

Для доказательства определим

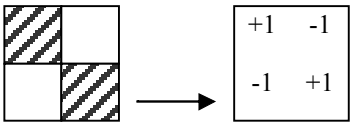
$$f(\mathbf{WX}) = \left(f \sum w_{ij} x_j\right) = \left(f \sum x_i x_j x_j\right) = \left(f\left(\left(\sum x_j^2\right) x_i\right)\right) = (x_i) = \mathbf{X} \ .$$

Веса w_{ij} можно умножить на некоторый положительный коэффициент (например, $1/n$).

Использование такого коэффициента особенно целесообразно в тех случаях, когда запоминаемые образы характеризуются большим числом признаков (например, видеоизображение отображается большим числом пикселей).

Пример 2.1. Воспроизведение корректного изображения с помощью сети Хопфилда

Имеется изображение из четырех пикселей:



Соответствующий вектор (образ) имеет вид: $\mathbf{X} = (1,-1,-1,1)^T$.

Стационарная сеть Хопфилда содержит при этом 4 элемента (нейрона). Ее весовая матрица рассчитывается на основе (2.50) и принимает вид

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{pmatrix}.$$

При этом легко можно убедиться в справедливости равенства $\mathbf{X} = f(\mathbf{WX})$.

Подадим на входы сети искаженный образ:

Можно легко установить, что нейронной сети потребуется лишь одна итерация, чтобы выдать корректное изображение.

2.5.3. Распознавание образов сетями Хопфилда

Обычно сети Хопфилда разрабатываются для запоминания и последующего распознавания большого количества образов (изображений). Обозначим через $\mathbf{X_1}, \ldots, \mathbf{X_K}$ эти образы, причем:

$$\mathbf{X}_j = (x_{j1}, \ldots, x_{jn})^T, \qquad j = 1, 2, \ldots, k, \tag{2.51}$$

где x_{ji} ($j = 1, 2, \ldots, k$; $i = 1, 2, \ldots, n$) – i-я составляющая j-го образа.

Подобно (2.50) введем матрицу $\mathbf{W}_s$ для s–го образа с весами

$$w_{ij}^{s} = \begin{cases} x_{si}\, x_{sj}, & s = \overline{1,k}, \quad i \neq j \\ 0, & s = \overline{1,k}, \quad i = j \end{cases} . \qquad (2.52)$$

Из этих матриц можно образовать результирующую матрицу сети:

$$\mathbf{W} = (\mathbf{W}_1 + \ldots + \mathbf{W}_k)/n \ , \qquad (2.53)$$

где n – размерность векторов (например, число пикселей в представлении изображения).

Если имеется немного изображений (т.е. k мало), то нейронная сеть с матрицей (2.52) запоминает k образов, естественно, что изображения не сильно коррелированны. Если же число k велико, то матрица (2.53) оказывается недостаточной для запоминания всех k изображений. С ростом k уменьшается вероятность воспроизведения образа. Приведем утверждение относительно числа k запоминаемых образов при использовании сети Хопфилда из n нейронов.

Пусть k – число образов, n – число признаков (пикселей), а образы, подлежащие запоминанию – некоррелированы, т.е. для двух изображений j и s сумма

$$\left| \sum_i x_{ji}\, x_{si} \right| \ . \qquad (2.54)$$

мала. При этом при подаче на вход сети с весовой матрицей (2.53) одного образа каждый пиксель корректно воспроизводится с вероятностью $p \cong 0{,}99$ при выполнении условия:

$$k \leq 0{,}15n \quad (n \to \infty) \ . \qquad (2.55)$$

Пример 2.2. Запоминание и корректное воспроизведение множества образов с помощью сети Хопфилда

Образ характеризуется тысячью признаками (n = 1000). В этом случае бинарная сеть Хопфилда в состоянии запомнить и корректно воспроизвести до 150 образов (например, видеоизображений).

В работе сети Хопфилда можно выделить следующие три стадии.

1. Инициализация сети: на этой стадии рассчитываются все веса сети для некоторого множества образов. Эти веса не определяются на основе рекуррентной процедуры, используемой во многих алгоритмах обучения с поощрением.
2. Ввод нового образа: нейроны сети устанавливаются в соответствующее начальное состояние по алгоритму (2.43) – (2.48).

3. Затухающий колебательный процесс: путем использования итеративной процедуры рассчитывается последовательность состояний сети до тех пор, пока не будет достигнуто стабильное состояние, т.е.

$$o_j(t+1) = o_j(t) \ , \qquad j = 1, 2, \ldots, n \tag{2.56}$$

или в качестве выходов сети используются значения (2.56), при которых сеть находится в динамическом равновесии:

$$\mathbf{O} = f(\mathbf{WO}) \ , \tag{2.57}$$

где **O** – выходной вектор сети.

Выход бинарной сети Хопфилда, содержащей n нейронов, может быть отображен бинарным вектором **O** состояния сети. Общее число таких состояний – 2^n (вершины n-мерного гиперкуба). При вводе нового входного вектора состояние сети изменяется от одной вершины к другой до достижения сетью устойчивого состояния.

Из теории систем с обратными связями известно: для обеспечения устойчивости системы ее изменения с течением времени должны уменьшатся. В противном случае возникают незатухающие колебания.

Для таких сетей Коуэном (Cohen) и Гроссбергом (Grossberg) (1983) доказана теорема, формулирующая достаточные условия устойчивости сетей с обратными связями.

Рекуррентные сети устойчивы, если весовая матрица $\mathbf{W} = (w_{ij})$ ***симметрична, а на ее главной диагонали – нули:***

$$1)\ w_{ij} = w_{ji} \quad \text{для всех } i \neq j \ ; \tag{2.58}$$
$$2)\ w_{ii} = 0 \quad \text{для всех } i \ .$$

Обратим внимание, что условия данной теоремы достаточны, но не необходимы. Для рекуррентных сетей отсюда следует, что возможны устойчивые сети, не удовлетворяющие приведенному критерию.

Для доказательства теоремы Коуэна и Гроссберга используем энергетическую функцию E (функцию Гамильтона), принимающую лишь положительные значения. При достижении сетью одного из своих устойчивых состояний эта функция принимает соответствующее минимальное значение (локальный минимум) для бинарных образов в результате конечного числа итераций:

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} x_i x_j + \sum_i x_i T_i \ , \tag{2.59}$$

где x_i – вход i-го нейрона, T_i – порог i-го нейрона.

Для каждого образа, вводимого в сеть, можно определить энергию E. Определив значение функции E для всех образов, можно получить поверхность энергии с максимумами (вершинами) и минимумами (низинами), причем минимумы соответствуют образам, запомненным сетью. Для сетей Хопфилда справедливо утверждение: *минимумы энергетической функции соответствуют образам, запомненным сетью*. В результате итераций сеть Хопфилда в соответствии с (2.43) – (2.48) сходится к запомненному образу.

Для запоминания каждого следующего образа в поверхность энергии, описываемой энергетической функцией E, необходимо ввести новую «низину». Однако при таком введении уже существующие «низины» не должны быть искажены.

Для некоторого образа $\mathbf{X}=(x_1, x_2, \ldots, x_n)^T$ следует минимизировать обе составляющие функции E (2.59). Для того чтобы второе слагаемое

$$\sum_i x_i T_i$$

было отрицательным, необходимо обеспечить различие знаков входов x_i и порогов T_i. При фиксированных порогах это невыполнимо, поэтому выберем пороги равными нулю. При этом второе слагаемое в (2.59) исключается, а остается лишь первое:

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} x_i x_j \quad . \tag{2.60}$$

Из общего числа k образов выделим некоторый образ с номером s. Тогда энергетическую функцию E (2.60) можно представить следующим образом:

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w'_{ij} x_i x_j - \frac{1}{2}\sum_i \sum_{j \neq i} w^s_{ij} x_{si} x_{sj} \quad , \tag{2.61}$$

где w^s_{ij} – составляющая весового коэффициента w_{ij}, вызванная s-м образом, w'_{ij} – составляющая весового коэффициента w_{ij}, вызванная остальными образами, запомненными сетью, x_{si} – значение i-го входа для s-го образа.

Выделенный образ с номером s определяет лишь второе слагаемое в (2.59):

$$E_s = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij}^s x_{si} x_{sj} \quad . \tag{2.62}$$

Задача минимизации величины E_s эквивалентна максимизации выражения

$$\sum_i \sum_{j \neq i} w_{ij}^s x_{si} x_{sj} \quad . \tag{2.63}$$

Входы сети x_{si} принимают значения из множества {-1, 1}, поэтому $(x_{si})^2$ всегда положительны. Следовательно, путем разумного выбора весовых коэффициентов w^s_{ij} можно максимизировать выражение (2.63):

$$\sum_i \sum_{j \neq i} w_{ij}^s x_{si} x_{sj} = \sum_i \sum_{j \neq i} (x_{si})^2 (x_{sj})^2 \quad , \tag{2.64}$$

где $w^s_{ij} = x_{si} x_{sj}$

Минимум энергетической функции E (2.60) достигается при выборе весового коэффициента

$$w_{ij}^s = x_{si} x_{sj} \, . \tag{2.65}$$

Это справедливо для s-го образа, подлежащего запоминанию сетью Хопфилда. Для всех k образов, запоминаемых нейронной сетью, получаем равенство

$$w_{ij} = \sum_s w_{ij}^s = \sum_s x_{si} x_{sj} \quad . \tag{2.66}$$

Пример 2.3. Определение выходного вектора за три шага с помощью сети Хопфилда

Дана сеть из трех нейронов и весами, равными 1 (рис. 2.11):

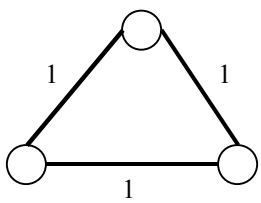


Рис. 2.11. Пример сети Хопфилда

Соответствующая весовая матрица имеет вид:

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} .$$

В качестве функции активации или выхода нейронов выберем знаковую функцию (sign) с нулевыми порогами. Пусть на вход такой сети подается вектор $\mathbf{X}=(1,-1,1)^T$.

Рассчитаем для него выходной вектор сети. При этом в соответствии с (2.43) – (2.48) получим:

$$\mathbf{O}(1) = f(\mathbf{WX}) = (-1,1,-1)^T ;$$
$$\mathbf{O}(2) = f(\mathbf{WO}(1)) = (-1,-1,-1)^T ;$$
$$\mathbf{O}(3) = f(\mathbf{WO}(2)) = (-1,-1,-1)^T .$$

Так как $\mathbf{O}(3)=\mathbf{O}(2)$, то после 3-го шага выходы сети не изменятся, т.е. выходной вектор определяется сетью после 3-го шага.

Основная область применения сетей Хопфилда – распознавание образов. Например, каждое черно-белое изображение, представляемое пикселями, можно отобразить вектором $\mathbf{X}=(x_1, \ldots, x_n)^T$, где x_i для i-го пикселя равен 1, если он черный, и x_i=–1, если – белый. При подаче на входы обученной сети Хопфилда искаженного изображения сеть после некоторого числа итераций выдает на выходы корректное изображение. На рис. 2.12 а приведены корректные образы, запомненные сетью, а на рис. 2.12 б – последовательность состояний сети Хопфилда при вводе искаженного образа (старт). После четвертой итерации нейронная сеть выдает корректный образ.

2.5.4. Непрерывные сети

В соответствии с предложением Хопфилда активация, функция активации и выходы сети Хопфилда могут быть непрерывными.

Для этого может быть использована, например, сигмоидальная логистическая функция с параметром λ:

$$f(z_j = net_j) = 1 \Big/ (1 + e^{-\lambda z_j}) \quad . \qquad (2.67)$$

Чем больше значения λ, тем лучше приближение сигмоидальной функции к бинарной пороговой функции.

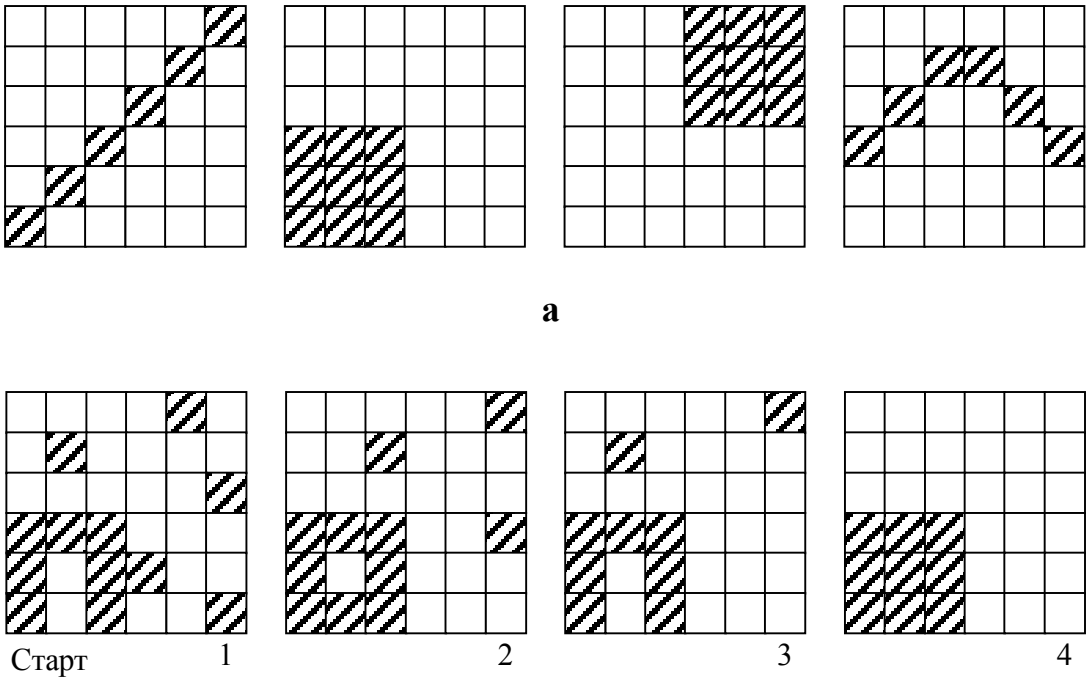


Рис. 2.12. Последовательность состояний сети Хопфилда
а – корректные образы, запомненные сетью
б – искаженные образы (старт)

Для непрерывных сетей Хопфилда справедлива модификация теоремы Коуэна и Гроссберга:

Сеть устойчива при выполнении следующих условий:
весовая матрица* W *симметрична: $w_{ij} = w_{ji}$ ***для*** $i \neq j$**;**
главная диагональ содержит нули: $w_{ii} = 0$ ***для всех*** i**.**

2.5.5. Применение сетей Хопфилда для оптимизации

Одной из известных «трудных» проблем оптимизации является задача о коммивояжере (Traveling Salesman Problem, TSP). Она состоит в определении кратчайшего пути, соединяющего некоторое множество городов, причем каждый город должен учитываться лишь один раз. Это пример NP – полной задачи.

Хопфилд и Танк показали подход к ее приближенному решению на основе сетей Хопфилда. Рассмотрим вкратце этот подход. Для описания возможных маршрутов авторы предложили специальный тип матрицы. В

ней города образуют строки, а столбцы отображают последовательность городов в маршруте. В позиции (x, i) матрицы стоит 1 в том случае, когда город x занимает i-е место в маршруте.

В табл. 2.1 отображен маршрут, в котором пять городов A, B, C, D, E посещаются в следующей последовательности: D ➔ B ➔ E ➔ A ➔ C.

В случае n городов существует $\frac{n!}{2n}$ различных маршрутов, среди которых необходимо найти кратчайший. Для получения решения задача отображается сетью Хопфилда.

Таблица 2.1. Маршрут посещения пяти городов в заданной последовательности (задача о коммивояжере)

Город	**Последовательность**				
	1	**2**	**3**	**4**	**5**
A	0	0	0	1	0
B	0	1	0	0	0
C	0	0	0	0	1
D	1	0	0	0	0
E	0	0	1	0	0

В ней каждый нейрон обозначается двумя индексами x и i, причем x отражает город, а i-ю позицию в маршруте, т.е. o_{xi} – это выход нейрона, в котором город x размещен на i-й позиции маршрута.

К энергетической функции E сети Хопфилда предъявляются следующие условия:

- должна быть минимальна только для допустимых решений, которые содержат одну единицу в каждой строке и в каждом столбце матрицы описания маршрутов;
- для решений с более короткими маршрутами должна принимать меньшие значения.

Энергетическая функция, удовлетворяющая этим условиям, может иметь вид:

$$E = \frac{A}{2}\sum_x\sum_i\sum_{j\neq i} o_{xi}\,o_{xj} + \frac{B}{2}\sum_x\sum_i\sum_{y\neq x} o_{yi}\,o_{xi} + \frac{C}{2}\left(\left(\sum_x\sum_i o_{xi}\right) - n\right)^2 + \\ + \frac{D}{2}\sum_x\sum_i\sum_y dist_{xy}\,o_{xi}\left(o_{y,i+1} + o_{y,i-1}\right) \qquad (2.68)$$

При ее выборе учтены следующие соображения:

- первое слагаемое равно нулю только в тех случаях, когда каждая строка матрицы описания маршрутов содержит лишь одну единицу;
- второе слагаемое равно нулю только тогда, когда каждый столбец матрицы содержит лишь одну единицу;
- третье слагаемое равно нулю лишь в тех случаях, когда в матрице описания маршрутов имеется n единиц, что означает: каждый город посещается лишь один раз;
- четвертое слагаемое отражает длину маршрута. Обратим внимание, что для каждого города x, расположенного на i-й позиции, определяется расстояние $dist_{xy}$ до его последователя y на позиции i+1 и его предшественника y на позиции i–1.

При такой энергетической функции необходимо определить веса $w_{xi,\,yi}$, причем вес $w_{xi,\,yi}$ отражает силу соединения нейрона x_i и нейрона y_j.

2.6. ART-сети

2.6.1. Определение

Сети ART (Adaptive Resonance Theory) образуют класс различных нейронных сетей, предложенных Карпентером и Гроссбергом (Бостонский университет) в период 1987-1991 гг. [12].

На практике данные, используемые для обучения или самообучения сети, часто нестабильны. Например, если на вход обычной нейронной сети с прямыми связями, обучаемую с помощью алгоритма с обратным распространением ошибки (Backpropagation), подать образ такого класса, который не был представлен в обучающей последовательности или во множестве образов, подлежащих автоматической классификации или кластеризации (при самообучении сети, unsupervised learning).

Здесь мы сталкиваемся с двумя противоречивыми требованиями или свойствами нейронной сети. С одной стороны очень важно, чтобы она была способна выявлять (обнаруживать) образы новых классов, ранее не представленных сети. Это свойство пластичности. С другой же стороны изученные классы образов должны сохраняться – свойство устойчивости нейронных сетей. Эти два свойства – пластичности и стабильности в известной мере противоречивы – дилемма пластичности-стабильности. Сети ART и были разработаны для разрешения этой дилеммы, а именно:

установление новых ассоциаций (классов) нейронной сетью без забывания старых ассоциаций (классов). Семейство ART-сетей включает:

- ART-1: для бинарных входных векторов, когда признаки распознаваемых образов принимают два значения 1 или 0;
- ART-2: расширение ART-1-сетей на непрерывные входные векторы;
- ART-2a: оптимальная версия ART-2-сетей, отличающаяся повышенной скоростью сходимости;
- ART-3: моделирование временных и химических процессов (биологических механизмов) на базе ART-2;
- ARTMAP: комбинация двух ART-сетей (например, ART-1 и ART-2);
- FuzzyART: гибридная сеть, объединяющая нечеткую логику (Fuzzy Logik) и ART сети.

2.6.2. Архитектура сети ART-1

Принцип работы ART-сетей сравнительно прост. При вводе значений признаков некоторого образа ART-1-сеть пытается сопоставить ему некоторый класс из числа уже изученных. Если такой класс удается найти, то производится сравнительно небольшая модификация прототипа (стереотипа, типичного представителя) этого класса для того, чтобы он хорошо отображал и новый образ. Классификация образа на этом заканчивается.

Если же такой класс найти не удается, то образуется (вводится) новый класс. При этом предъявленный образ несколько модифицируется и используется затем в качестве прототипа (стереотипа, типичного представителя) для нового класса. При этом уже изученные классы не изменяются.

На рис. 2.13 показаны основные компоненты ART-1-сетей:

- слой сравнения (Comparison Layer);
- слой распознавания (Recognition Layer);
- весовые матрицы;
- коэффициенты усиления (ключи) и Reset.

Обратим внимание, что обе весовые матрицы $\mathbf{W}_{ij}$ и $\mathbf{W}_{ji}$ отличаются обозначениями индексов: индекс i относится к элементу (входу) слоя сравнения F1, а индекс j – к нейрону (классу) слоя распознавания F2.

2.6.3. Слой сравнения и слой распознавания

Слой сравнения (Comparison Layer) осуществляет сравнение выхода слоя распознавания с текущим входом (входным образом). Для этого входной вектор **I** (Input) преобразуется сначала в вектор **S**, который затем передается на весовую матрицу действительных чисел (Bottom-up Matrix).

В начале расчета коэффициент усиления (Gain) g_1 равен 1. На выходе слоя распознавания рассчитывается так называемый ожидаемый вектор **V** или типичный представитель (прототип, стереотип) для класса образов, к

которому отнесен вектор **S**. В процессе обучения (точнее самообучения) сети вектор **S** определяется на основе правила [12]:

$$s_i = \begin{cases} 1, \textit{ если } I_i v_i \vee I_i g_1 \vee v_i g_1 = 1 \\ 0, \textit{ в противном случае} \end{cases}, \qquad (2.69)$$

или:

- *i*-я компонента вектора **S** принимает значение 1, если, по крайней мере, две из трех следующих переменных приняли значение 1;
- коэффициент усиления g_1 (для всех нейронов одинаков);
- *i*-я компонента I_i входного вектора **I**;
- *i*-я компонента v_i ожидаемого вектора **V** (взвешенная сумма выходов слоя распознавания).

Это правило кратко можно обозначить «2 из 3».

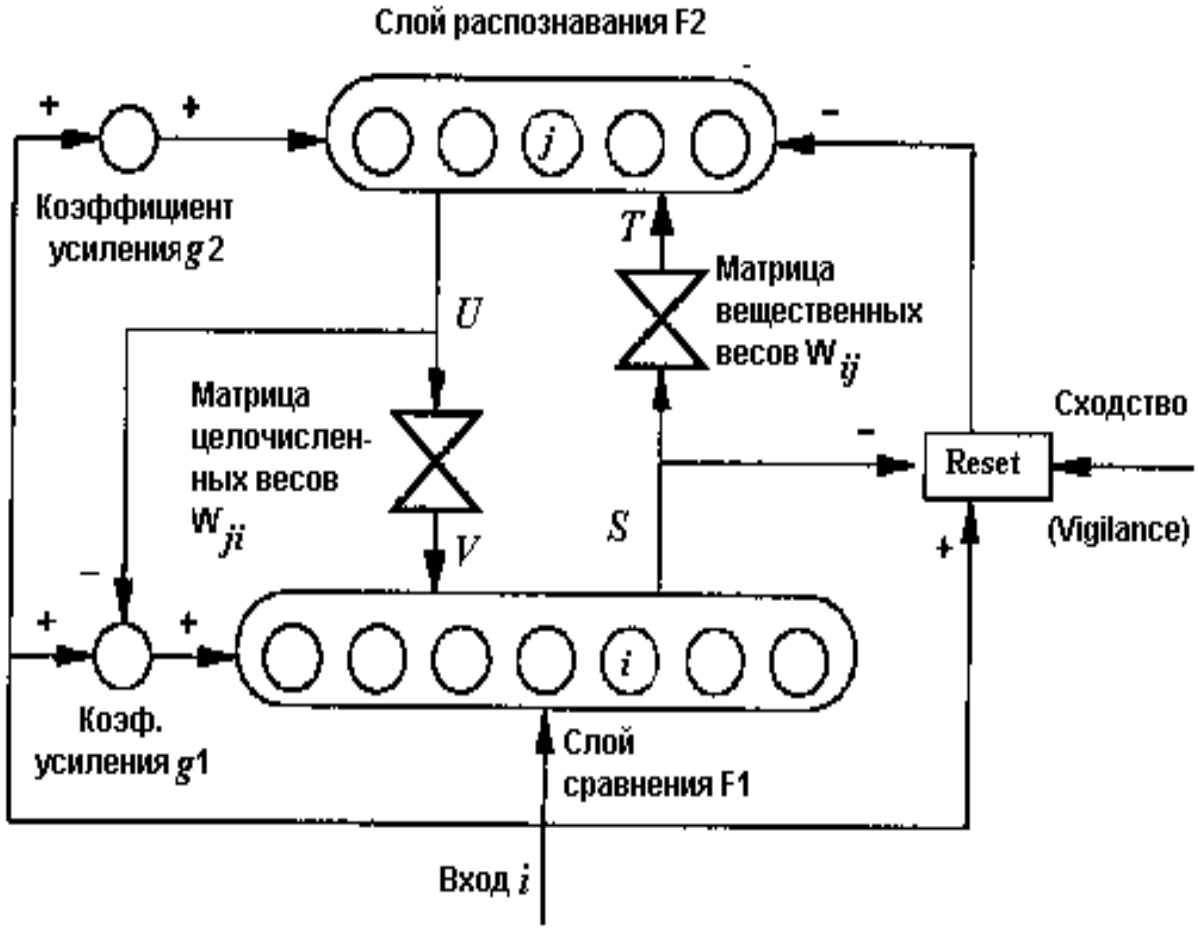


Рис. 2.13. Основные компоненты ART-1-сетей

Пример 2.4. Функционирование слоев сравнения и распознавания в случае, когда коэффициент усиления g_1 равен нулю

Дано (g_1=0):

I =	1	0	1	0	1	0	1	0	1	0
V =	1	1	1	0	0	0	1	0	0	0

Результат:

S =	1	0	1	0	0	0	1	0	0	0

Пример 2.5. Функционирование слоев сравнения и распознавания в случае, когда коэффициент усиления g_1 равен 1

Дано (g_1=1):

I =	1	0	1	0	1	0	1	0	1	0
V =	1	1	1	0	0	0	1	0	0	0

Результат:

S =	1	1	1	0	1	0	1	0	1	0

Определим вектор **S** при предъявлении сети первого образа, когда ожидаемый вектор нулевой: **V** = (0, 0, …, 0).

Пример 2.6. Функционирование слоев сравнения и распознавания в случае, когда ожидаемый вектор V нулевой

Дано (g_1=1):

I =	1	0	1	0	1	0	1	0	1	0
V =	0	0	0	0	0	0	0	0	0	0

Результат:

S =	1	0	1	0	1	0	1	0	1	0

т. е. вектор **S** на первом шаге обучения сети совпадает с входным вектором: **S** = **I**.

На рис. 2.14 представлена часть ART-1-сети с четырьмя нейронами в слое сравнения и тремя нейронами в слое распознавания.

Слой распознавания сопоставляет каждому входному вектору соответствующий класс. Если для входного вектора не удается найти достаточно близкий класс из числа уже выявленных, то открывается (образуется) новый класс.

Класс, представляемый *j*-м нейроном слоя распознавания и наиболее близкий к входному вектору **I** или вектору **S**, определяется так:

$$t_{j\max} = \max_k \left\{ t_k = SW_k = \sum_{i=1}^{m} s_i w_{ik} \right\}, \qquad (2.70)$$

где $\mathbf{SW}_k$ – скалярное произведение векторов **S** и $\mathbf{W}_k$. При этом сработает нейрон *j* слоя распознавания, действительный весовой вектор

$$\mathbf{W}_j = (w_{1j}, w_{2j}, \ldots, w_{mj})$$

которого имеет наибольшее сходство с вектором **S**. Компоненты вектора **U** на выходе слоя распознавания определяются при этом так:

$$u_j = \begin{cases} 1, если\_t_j = SW_j = \max \\ 0, в\_противном\_случае \end{cases}, \qquad (2.71)$$

т.е. $u_j = 1$, если скалярное произведение весового вектора $\mathbf{W}_j=(w_{1j}, w_{2j}, \ldots, w_{mj})$ и вектора $\mathbf{S}=(s_1, s_2, \ldots, s_m)$ максимально.

2.6.4. Весовые матрицы и коэффициенты усиления

В ART-1-сетях используются две *весовые матрицы*:

- действительная матрица $\mathbf{W}_{ij}$ (Bottom-up Matrix) служит для расчета степени сходства в фазе распознавания;
- бинарная матрица $\mathbf{W}_{ji}$ (Top-down Matrix) предназначена для перепроверки степени корректности классификации входного образа с помощью матрицы $\mathbf{W}_{ij}$ (Bottom-up Matrix).

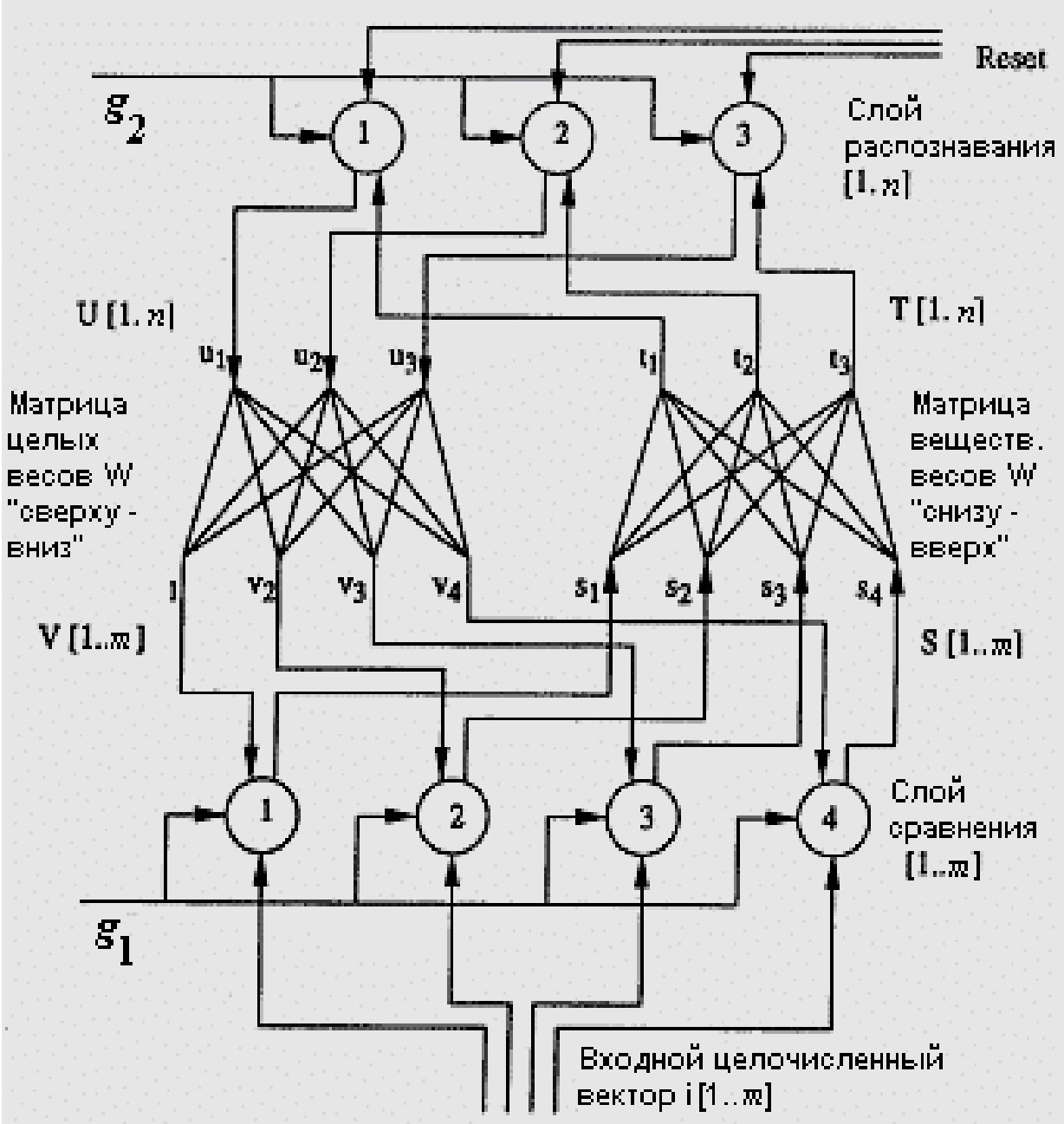


Рис. 2.14. Фрагмент ART-1-сети с четырьмя нейронами в слое сравнения и тремя нейронами в слое распознавания

В ART-1-сетях используются два *коэффициента усиления* (Gain) g_1 и g_2. Строго они не выполняют функции усиления, а используются лишь в качестве ключей для синхронизации работы нейронной сети.

g_1 принимает значение 1, если по меньшей мере одна составляющая входа **I** имеет значение 1 и одновременно ни один нейрон слоя распознавания не находится в состоянии 1:

$$g_1 = (I_1 \vee I_2 \vee ... \vee I_m) \wedge \neg(u_1 \vee u_2 \vee ... \vee u_m) \quad (2.72)$$

g_2 реализует логическое «ИЛИ» для входного вектора; g_2=1, если, по меньшей мере, одна составляющая входного вектора равна 1:

$$g_2 = (I_1 \vee I_2 \vee ... \vee I_m) \quad (2.73)$$

Для актуального нейрона («победителя») слоя распознавания reset-составляющая равна 1 (то есть функция Reset активируется), если различие входного вектора **I** и вектора **S** превышает некоторый порог (параметр толерантности).

2.6.5. Принцип работы

В ART-1-сетях различают следующие пять фаз внутренней обработки информации.

1. Инициализация сети: в начале инициализируются обе весовые матрицы (Bottom-up Matrix и Top-down Matrix), а также параметр толерантности.
2. Распознавание (Recognition): на этой фазе для входного вектора **I** или для вектора **S**, определенного на основе вектора **I**, определяется наиболее близкий класс.
3. Сравнение (Comparison): ожидаемый вектор **V** сравнивается со входным вектором **I** или вектором **S**. При слишком малом совпадении векторов **V** и **S** осуществляется повторное распознавание.
4. Поиск (Search): производится поиск альтернативного класса или же при необходимости открывается новый класс.
5. Адаптация весов (Training): на этой стадии осуществляется модификация обеих весовых матриц.

Рассмотрим приведенные фазы более подробно.

Инициализация. Пусть i = 1, 2, …, m – индекс нейронов слоя сравнения, а j = 1, 2, …, n – индекс нейронов слоя распознавания. Веса w_{ij} матрицы **W** (Bottom-up Matrix) сначала устанавливаются небольшими в соответствии с неравенством

$$w_{ij} < \frac{L}{L-1+m} \quad (2.74)$$

для всех нейронов i слоя сравнения и нейронов j слоя распознавания, где m – число составляющих входного вектора **I**, L – постоянная (L > 1, типичное значение L=2). При выборе весов w_{ij} слишком большими все входные вектора отображаются на один нейрон слоя распознавания.

Все веса бинарной матрицы (Top-down matrix) **W** вначале устанавливаются единицами: w_{ji}=1 для всех j слоя распознавания и слоя сравнения.

Параметр толерантности (сходства, Vigilance) p выбирается между 0 и 1 ($p \in [0;1]$) – в зависимости от желаемой степени совпадения. Небольшое значение p (близкое к 0) «не обращает внимания» на большие различия образов внутри одного класса. При выборе p=1 требуется абсолютное (точное) совпадение между входом и прототипом (стереотипом) класса образов.

Распознавание. В начале процесса обучения (самообучения) сети входной вектор нулевой: **I** = (0, 0,…, 0). Соответственно коэффициент усиления g_2=0. В результате все нейроны слоя распознавания отключены. Следствием этого является нулевой вектор **V** (Top-down Vector).

При подаче на вход сети ненулевого входного вектора **I** коэффициенты усиления (ключи) принимают значения: g_1=1 и g_2=1. Это приводит к срабатыванию по правилу “2 из 3” тех нейронов слоя сравнения, входы которых I_i=1. В результате формируется вектор **S**, который сначала является точной копией вектора **I**. Затем для каждого j слоя распознавания вычисляется скалярное произведение весового вектора $\mathbf{W}_j$ и вектора **S**:

$$t_j = net_j = \mathbf{W}_j \mathbf{S} \quad .$$

Скалярное произведение является мерой сходства между векторами $\mathbf{W}_j$ и **S**. Нейрон j – «победитель» – с максимальным значением скалярного произведения «выигрывает» сравнение («соревнование») и возбуждается (срабатывает), все же остальные нейроны не срабатывают.

Из (2.55) следует, что при этом лишь j-я составляющая вектора **U** (вход для Top-down матрицы) принимает значение 1. Все же остальные составляющие – нули.

Фаза сравнения. Единственный нейрон j–«победитель» слоя распознавания, весовой вектор которого наиболее близок к входному вектору, выдает единицу (1). Эта единица распространяется через бинарные веса w_{ji} матрицы Top-down. Остальные же нейроны слоя распознавания выдают нули. В результате каждый нейрон i слоя сравнения получает бинарный сигнал v_i, равный значению w_{ji} (1 или 0):

$$v_i = \sum_{j \in Recog} u_j w_{ji} = w_{Ji} \quad . \qquad (2.75)$$

Так как входной вектор **I** не нулевой и j-й нейрон слоя распознавания находится в состоянии 1, то из (2.72) следует g_1=0, ибо только одна компонента в фазе распознавания установлена в 1.

После этого рассчитывается новое значение для **S**. Этот расчет сводится по существу к покомпонентному сравнению векторов **V** и **I**. Если векторы **V** и **I** различаются в некоторой составляющей, то соответствующие составляющие вектора **S** считаются равными 0, т. е. $\mathbf{S} = \mathbf{V} \wedge \mathbf{I} = \mathbf{W}_J \wedge \mathbf{I}$.

Пример 2.7. Принцип работы ART-1-сети в фазе сравнения и распознавания в случае, когда ожидаемый вектор V нулевой

Дано: вектор **U**, т. е. входной образ отнесен к третьему классу (j=3):

U =	0	0	1	0	0	0

и бинарная матрица (Top-down Matrix) **W**

1	1	1	1	1	0
1	0	1	1	1	0
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	1	1	1
0	1	1	1	1	1

Результат:

V_3 =	1	0	1	0	1	0

Затем определяется степень сходства (Similarity) между бинарными векторами **I** и **S** для Reset-компоненты:

$$sim = \frac{|\mathbf{S}|}{|\mathbf{I}|} = \frac{|\mathbf{W}_j \wedge \mathbf{I}|}{|\mathbf{I}|} \geq p \ . \qquad (2.76)$$

Функция Reset запускается при невыполнении этого неравенства, а именно при $\frac{|\mathbf{S}|}{|\mathbf{I}|} < p$. Значение параметра сходства p выбирается между 0 и 1: значение p=1 требует полного совпадения, а при p=0 совершенно различные образы отображаются в один класс.

Или иначе: при выборе большого значения p образуется больше классов образов; при выборе же небольших значений p входные образы разделяются на меньшее число классов (кластеров). Обычно значение p выбирается между 0,7 и 0,95: $p \in [0{,}7;\ 0{,}95]$

Пример 2.8. Определение степени сходства между бинарными векторами I и S для Reset-компоненты (*sim* = 1)

Дано:

$\mathbf{W}_j$=	0	1	1	1	0	1	0
I=	0	1	0	1	0	1	0

Результат: $sim = \frac{3}{3} = 1$.

Пример 2.9. Определение степени сходства между бинарными векторами I и S для Reset-компоненты (*sim* = 0,75)

Дано:

$\mathbf{W}_j$=	0	1	0	1	0	1	0
I=	0	1	1	1	0	1	0

Результат: $sim = \frac{3}{4} = 0{,}75$.

Если степень сходства выше установленного порога толерантности (степени сходства) $sim > p$, то вход **I** считается опознанным. В противном случае включается функция Reset, и классификация начинается заново.

Поиск. При включении (активации) функции Reset (т.е. при недостаточной степени сходства векторов **I** и **S**) вектор **U** обнуляется

(устанавливается нулевой вектор **U**={0,0,...,0}), что является условием старта (запуска) процедуры классификации. При этом вектор **S** полагается равным вектору **I**: **S=I**, и классифицируется непосредственно входной образ, отображаемый вектором **I**. Процесс классификации опять-таки включает распознавание, и сравнение и продолжается до тех пор, пока не будет выявлен класс, обеспечивающий достаточную степень сходства исключающий запуск функции Reset. В противном случае, когда такой класс выявить не удается, в слое распознавания открывается новый класс.

Весовые векторы матрицы Bottom-up рассчитываются так, чтобы обеспечить максимальное значение скалярного произведения для входа **I**.

Адаптация весов. В ART-сетях различают два типа обучения (тренинга): медленное обучение или медленный тренинг (Slow Training) и быстрое обучение или быстрый тренинг (Fast Training). При медленном обучении входные векторы настолько кратковременно подаются на вход нейронной сети, что веса сети не успевают достигнуть своих асимптотических значений. Динамика нейронной сети (точнее динамика весов нейронной сети) описывается при этом дифференциальными уравнениями, рассмотрение которых опустим.

При быстром обучении входные векторы (образы) сохраняются на входе нейронной сети в течение времени, достаточного для достижения весовыми матрицами их стабильных значений. Ограничимся рассмотрением лишь быстрого обучения.

На стадии адаптации осуществляется юстирование (уточнение) весов обеих матриц. Для весов Bottom-up матрицы:

$$w_{iJ} = \frac{Ls_i}{L-1+\sum_{k=1}^{m} s_k} \quad , \tag{2.77}$$

где L – постоянная ($L > 1$, причем обычно L=2), s_i – i-я составляющая выходного вектора слоя сравнения, j – номер нейрона-«победителя» слоя распознавания и w_{ij} вес (точнее i-я составляющая) Bottom-up вектора **W** или иначе w_{ij} – это вес соединения i-го нейрона слоя сравнения и j-го нейрона слоя распознавания.

Веса бинарной Top-down матрицы изменяются в соответствии с правилом:

$$w_{ji}(t+1) = s_i \wedge w_{ji}(t) \quad , \tag{2.78}$$

т.е. соответствующий весовой вектор Top-down матрицы модифицируется таким образом, что он воспроизводит вектор **S**.

2.6.6. Поток информации в сети

Представим еще раз кратко поток информации внутри ART-1-нейросети. Для каждого входа *i* ART-1-сеть ищет адекватный класс образов в слое распознавания *F2*. Для этой цели на входе *i* в слое сравнения *F1* генерируется образ долговременной памяти или ДВП-образ **S**. При этом одновременно функция Reset отключается. Вектор **S** трансформируется затем в вектор **T**, который в свою очередь активирует некоторый класс **J** в слое *F2*. Слой распознавания *F2* генерирует затем вектор **U**, который преобразуется в так называемый ожидаемый вектор **V** или иначе в вектор **V**, ожидаемый для класса **J**.

Если степень совпадения между **V** и входом ниже установленного порога, то генерируется новый вектор **S**. В этом случае активируется функция Reset, а выявленный ранее класс **J** в слое распознавания *F2* забывается, ибо введенный образ не может быть отнесен к этому классу. Затем осуществляется новый поиск подходящего класса для *i*. Этот поиск заканчивается тогда, когда в слое сравнения не активируется функция Reset или же слой распознавания *F2* дополняется еще одним классом.

2.6.7. Другие ART-сети

Приведем краткий обзор сетей *ART-2*, *ART-2a*, *ART-3*, *ARTMAP* и *FUZZY-ART*.

ART-2 и ART-2a. Главное отличие от ART-1-сетей: они обрабатывают не бинарные, а действительные входные векторы. ART-2a-сеть представляет «более быстрый» вариант сети ART-2. В силу этого многими авторами ART-2a-сети рекомендуются для решения сложных задач. Сравнительными исследованиями установлено, что качество классификации (образование классов) сетями ART-2 и ART-2a почти во всех случаях одинаково. Скорость же обучения (сходимости) при использовании ART-2a-сетей значительно выше по сравнению с ART-2-сетями.

ART-3-сети. Они разработаны для моделирования биологических механизмов. Их достоинство: простота использования в каскадных архитектурах нейронных сетей.

ARTMAP. Они объединяют элементы обучения и самообучения (или обучения с поощрением и без поощрения, Supervised and Unsupervised Training). Для этого обычно формируется комбинация из двух ART-сетей.

FUZZY-ART-сети (нечеткие ART-сети). Они представляют собой прямое расширение ART-1-сетей средствами нечеткой логики (Fuzzy Logic). В них применяются следующие операторы:

- для определения класса образов в слое распознавания;
- для расчета степени сходства (Reset-критерий);
- для адаптации весов.

В результате ART-1-сети могут быть использованы для обработки не бинарных, а действительных входных векторов.

Основная особенность ART-1-сетей состоит в способности к формированию новых классов в процессе обучения или иначе в разрешении дилеммы стабильности-пластичности.

3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

3.1. Обзор программного обеспечения для моделирования

Различают универсальные и прикладные программные продукты для моделирования нейронных сетей (Neural Network Simulators).

Универсальные или ***объектно-инвариантные среды*** позволяют синтезировать оптимальные нейронные сети, применяемые для решения широкого класса задач, с предложением различных парадигм и алгоритмов обучения.

Прикладные среды моделирования ориентированы для синтеза нейронных сетей, применяемых в той или иной отрасли, прикладной области или специфичной задаче.

Среди важнейших свойств нейросетевых симуляторов – способность синтезировать код программы результирующей нейронной сети на алгоритмическом языке высокого уровня (чаще всего – Си и Паскаль). Такой код впоследствии легко интегрировать в пользовательскую программу.

В табл. П.1 Приложений представлены наиболее распространенные *универсальные* программные среды для моделирования нейронных сетей.

3.2. Краткое описание программного продукта TRAJAN

Программный продукт TRAJAN компании TRAJAN Software Co. (Великобритания) [1, 11, 18] является симулятором полного цикла и предназначен для моделирования в среде Windows различных видов нейронных сетей и алгоритмов обучения.

Симулятор включает широкие возможности для графического и статистического контроля в процессе моделирования параметров и характеристик эффективности синтезируемых нейронных сетей.

Рассмотрим основные функциональные и потребительские характеристики программного продукта TRAJAN [18].

3.2.1. Автоматизация процесса синтеза нейронной сети

Программное обеспечение TRAJAN позволяет автоматизировать следующие процессы:

- формирования представительских выборок и переменных;
- выбора вида нейронной сети и ее структуры;
- обучения нейронной сети;
- сохранения оптимальных параметров нейронных сетей;
- выдачи графической и статистической информации о характеристиках синтезируемой нейронной сети.

Эти возможности позволяют работать в программной среде пользователям незнакомым с нюансами теории нейронных сетей, в то время как специалисты в данной области могут существенно сокращать рутинные этапы синтеза оптимальных нейронных сетей и временные затраты.

При проектировании нейронной сети, направленной на решение особенно сложных задач, моделирование в среде программного продукта TRAJAN, как правило, занимает несколько дней.

3.2.2. Формирование представительской выборки

Важнейший этап моделирования – формирование репрезентативной представительской выборки нейронной сети для решения конкретной задачи. Сложность этой проблемы обусловлена возможной взаимной зависимостью (корреляцией) между входами нейронной сети. Программный продукт TRAJAN реализует ряд алгоритмов решения этой проблемы, включая форвардный и пошаговый выбор, генетические алгоритмы и ряд других.

3.2.3. Многоуровневые персептроны

TRAJAN поддерживает многоуровневые персептроны, обучающиеся по методу обратного распространения (гл. 2). Также могут быть использованы алгоритм быстрого распространения и Delta-Bar-Delta-алгоритм.

3.2.4. Карты Кохонена

TRAJAN позволяет моделировать нейронные сети, реализованные в виде самоорганизующихся карт Кохонена (2.5). При этом процесс моделирования сопровождается визуализацией графического окна наилучших частот и окна с топологией карты, что позволяет в реальном времени моделирования локализовать и пометить кластеры.

3.2.5. Гибридные нейронные сети

TRAJAN позволяет моделировать также гибридные нейронные сети, построенные с использованием комбинации двух или нескольких парадигм.

3.2.6. Бейесовские сети

TRAJAN поддерживает современные достижения в области теории Байесовских вероятностных и регрессионных нейронных сетей. Данные парадигмы подразумевают почти мгновенное обучение сети, что неоценимо при модельном экспериментировании.

3.2.7. Линейные модели

Большинство специалистов сравнивают результаты нелинейных моделей с результатами, получаемыми при использовании линейной модели

нейронов. Линейные модели зачастую способны эффективно достигать цели за меньшее время при использовании того же программного обеспечения.

3.2.8. Интерфейс пользователя

Удобный пользовательский интерфейс программного продукта позволяет обеспечивать простой доступ к большим объемам информации.

Представительские выборки и структура нейронной сети сохраняются в файлах с единым именем и различными расширениями, что обеспечивает легкость группировки исходных данных для моделирования.

Обновляющиеся в реальном времени графики и гистограммы позволяют наблюдать за обучением и исполнением нейронной сети, оперативно реагировать на ход моделирования. При решении задач классификации или аппроксимации автоматически вычисляются разнообразные статистические параметры и характеристики.

Специализированные топологические карты и кластерные диаграммы применяются при изучении и анализе результатов моделирования.

Фактически вся символьная и числовая информация доступна в ***электронных таблицах (Datasheets)***, т.е. может быть мгновенно импортирована и экспортирована через буфер обмена Windows. Графическая информация также может быть экспортирована, например, для составления отчета о модельных экспериментах.

3.2.9. Ограничения

TRAJAN поддерживает нейронные сети «глубиной» 128 слоев, хотя, в подавляющем большинстве случаев, требуемое количество слоев существенно меньше.

В TRAJAN первый слой является всегда слоем входа. Он используется только, чтобы вводить величины в нейронную сеть, так как нейроны входного слоя не подразумевают никакой обработки. Последний слой является выходным, и результаты выполнения нейронов этого слоя являются и выходом нейронной сети в целом.

3.3. Описание основных этапов работы в среде TRAJAN

3.3.1. Создание нейронной сети

Новая нейронная сеть создается в TRAJAN с помощью окна ***Network Creating (Создание сети),*** которое доступно из меню ***File/New/Network*** или по нажатию соответствующей кнопки на панели инструментов [1, 18].

Для создания новой нейронной сети после появления на экране окна ***Network Creating*** следует:

- выбрать тип сети;
- определить число слоев и их размерности.

3.3.2. Выбор типа сети

TRAJAN предлагает несколько видов нейронных сетей для моделирования. Нейронная сеть типа ***Многоуровневый персептрон*** выбрана в данном окне по умолчанию.

3.3.3. Определение числа слоев и их размерности

При задании количества слоев нейронной сети следует учитывать, что TRAJAN может поддерживать нейронные сети вплоть до 128 слоев по 128 нейронов в каждом, при этом первый из них всегда является входным и используется только для получения сетью исходных данных, а последний слой является выходным, и выходы его нейронов являются выходами всей сети в целом.

Задать количество нейронов в каждом слое позволяет матрица, представленная в окне ***Network Creating.*** Она выглядит как небольшая электронная таблица. Количество нейронов в каждом слое нейронной сети определяется с помощью первой ячейки этой матрицы, при этом любые слои с нулевым количеством нейронов будут проигнорированы.

После того, как задано количество нейронов в каждом слое, TRAJAN самостоятельно определит количество слоев в сети путем выбора из матрицы всех слоев, у которых количество нейронов отлично от нуля.

Примечание. ***Можно заметить, что матрица содержит строку для задания ширины каждого слоя. Данная функция редко используется в TRAJAN и необходима только для нейронных сетей, использующих карты Кохонена (п. 2.5).***

3.3.4. Подготовка нейронной сети к обучению

Одной из ключевых характеристик нейронных сетей является их способность обучаться решать поставленные перед ними задачи с использованием представительских выборок. То есть вместо того, чтобы выставлять нейронам их веса вручную (что в принципе конечно возможно), создается некоторый набор представительских выборок, определяющих решение поставленной задачи, для обучения предварительно подготовленной структуры нейронной сети (п. 3.3.3). При этом используется один из алгоритмов обучения нейронных сетей, наиболее популярным из которых является алгоритм *Обратного распространения*.

Обучающий алгоритм во время своей работы подбирает веса нейронов в нейронной сети, исходя из информации, содержащейся в представительской выборке.

В TRAJAN набор представительских выборок (обучающих пар) создается с помощью окна ***Pattern Set Creating,*** которое доступно из пункта меню ***File/New/Pattem***.

После выбора этой команды меню на экране появляется диалоговое окно, в котором необходимо задать количество входов и выходов формируемых обучающих пар.

Примечание. ***Для целей обучения количество входов и выходов обучающих пар должно совпадать с количеством нейронов, содержащихся во входных и выходных слоях нейронной сети, созданной в п. 3.3.3.***

После этого TRAJAN запустит редактор вводимых обучающих пар, в котором отобразит матрицу обучающих пар. Каждая обучающая пара представлена в матрице своей строкой, которая содержит значение входных и соответствующих им выходных сигналов нейронной сети.

После создания нейронной сети и набора обучающих пар для ее обучения исходные данные моделирования сохраняются в файле. Для сохранения нейронной сети необходимо воспользоваться пунктом меню ***File/Save/Network.*** После выбора команды TRAJAN запрашивает имя файла, в котором будет сохранена моделируемая нейронная сеть.

Для сохранения набора обучающих пар необходимо воспользоваться пунктом меню ***File/Save/Pattern.*** После выбора данной команды TRAJAN запрашивает имя файла, в котором будут сохранены введенные представительские выборки (обучающие пары).

3.3.5. Редактирование сети и представительских выборок

Средства редактирования нейронных сетей (меню: ***Edit/Network***, окно: ***Network Editor***) позволяют модифицировать следующие параметры определенной ранее сети:

- функцию ошибки нейронной сети;
- изменить активационную функцию нейронов;
- изменить значения весов связей;
- удалить и добавить слои;
- удалить и добавить нейроны в текущем слое.

Электронная таблица (матрица) представительских выборок позволяет изменять любые имеющиеся обучающие пары, а также их удалять и добавлять новые.

3.3.6. Обучение нейронной сети

В настоящее время программным продуктом TRAJAN поддерживаются наиболее распространенные алгоритмы обучения нейронных сетей.

Рассмотрим процедуру обучения в TRAJAN по алгоритму *обратного распространения*.

Алгоритм обратного распространения работает, подготавливая нейронную сеть, используя доступные ему данные, которые хранятся в

наборе представительских выборок и которые были подготовлены для обучения сети. На каждой итерации (в терминах нейронных сетей – эпохе), весь составленный набор обучающих пар предоставляется сети. Выходы, получаемые сетью, сравниваются с желаемыми результатами. Ошибка нейронной сети вычисляется как разность между желаемыми и фактическими результатами и используется для регулирования весов нейронов в сети.

Перед началом обучения нейронной сети выполняются следующие действия:

1. Используя меню ***Statistics/Training Graph,*** открывается окно ***Training error Graph***;

2. Используя меню ***Train/Backprop,*** открывается окно ***Back Propagation***;

3. Данные окна располагаются так, чтобы они не перекрывали друг друга;

4. Запускается алгоритм обучения нажатием кнопки ***Train*** в окне ***Back Propagation***. При этом зависимость среднеквадратической ошибки от числа итераций будет показана на графике, расположенном в окне;

5. Увеличивается максимальное число итераций и алгоритм запускается (с помощью кнопки ***Train***) до тех пор, пока среднеквадратическая ошибка не примет приемлемого малого значения.

Вначале моделирования при использовании небольшого числа итераций среднеквадратическая ошибка уменьшается, но ненамного. Данный факт обусловлен тем, что задача «исключающего «ИЛИ» для нейронной сети, как не парадоксально, гораздо сложнее в решении, чем многие более сложные задачи.

Окно ***Training Error Graph*** отображает общую ошибку обучения нейронной сети, однако, иногда бывает полезно пронаблюдать за работой нейронной сети при использовании отдельно взятой обучающей пары, с помощью окна ***Pattern Error.***

3.3.7. Запуск нейронной сети

После обучения нейронной сети, она готова к запуску. Запустить нейронную сеть на выполнение можно несколькими способами.

Запуск с текущим набором представительских выборок

Нейронная сеть может быть запущена с предъявлением либо полного набора представительских выборок, использованных ранее при обучении нейронной сети, либо с предъявлением одиночных выборок. При этом необходимо воспользоваться пунктом меню ***Run/Single Pattern***, чтобы получить информацию о работе нейронной сети при предъявлении одной отдельно взятой представительской выборки или целого набора представительских выборок.

Запуск индивидуальной представительской выборки, не входящей в набор обучающих пар

Часто бывает необходимо проверить работу нейронной сети на представительской выборке, которая не входила в набор обучающих пар, использованных ранее при обучении, при решении следующих задач:

1. Прогнозирование появления новых данных с заранее неизвестными нейронной сети выходами. Если выходы заранее известны, то можно оценить качество работы подготовленной нейронной сети. В противном случае, результаты, полученные при запуске нейронной сети, могут быть использованы в качестве прогноза. Данный тип задач для нейронных сетей будет рассмотрен в лабораторной работе № 5.
2. Распознавание образов (подробнее данная задача будет рассмотрена в лабораторной работе № 3). В этом случае, оценивается чувствительность нейронной сети к небольшому изменению параметров исследуемого вектора, с помощью которого проводилось обучение.

4. НЕЙРОПРОЦЕССОРЫ

4.1. Определение и классификация нейропроцессоров

Нейропроцессор – это кристалл, который обеспечивает выполнение нейросетевых алгоритмов в реальном масштабе времени.

Среди разновидностей кристаллов, используемых в качестве нейропроцессоров выделим следующие (рис. 4.1):

- специализированные нейрочипы;
- заказные кристаллы (ASIC);
- встраиваемые микроконтроллеры (mC);
- процессоры общего назначения (GPP);
- перепрограммируемые логические интегральные схемы (FPGA, ПЛИС);
- процессоры цифровой обработки сигналов (ПЦОС);
- транспьютеры.

Специализированные нейрочипы часто реализуются на основе *процессорных матриц (систолических процессоров)*. Такие нейрочипы близки к обычным RISC-процессорам, объединяют в своем составе некоторое число процессорных элементов, а управляющая и дополнительная логика, как правило, строится на базе дополнительных схем.

Различают также *нейросигнальные процессоры*, ядро которых представляет собой типовой ПЦОС, а реализованная на кристалле дополнительная логика обеспечивает выполнение характерных нейросетевых операций (например, дополнительный векторный процессор и т.п.).

Транспьютеры, в частности T414, T800, T9000, и транспьютероподобные элементы являются важным компонентом ВСМП, однако, их применение сдвигается в сторону коммутационных систем и сетей ЭВМ [2].

4.2. Параметры нейропроцессоров

Для оценки производительности устройств (реализованных на основе ПЦОС и ПЛИС), применяемых для ЦОС, контролируется время выполнения типовых операций ЦОС, таких как цифровая фильтрация, БПФ и др.

В свою очередь, для оценки производительности нейропроцессоров и нейрокомпьютеров применяется ряд специальных показателей (параметров):

- ***MMAC*** – миллионов умножений с накоплением в секунду;
- ***CUPS (Connections Update per Second)*** – число измененных значений весов в секунду (оценивает скорость обучения);
- ***CPS (Connections per Second)*** – число соединений (умножений с накоплением) в секунду (оценивает производительность);

Рис. 4.1. Кристаллы, используемые в качестве нейропроцессоров

- ***CPSPW*** = *CPS*/*Nw*, где *Nw* – число синапсов в нейроне;
- ***CPPS*** – число соединений примитивов в секунду:

$$CPPS = CPS \bullet Bw \bullet Bs, \qquad (4.1)$$

где *Bw*, *Bs* – разрядность чисел, отведенных под веса и синапсы.

Ориентация процессоров на выполнение нейросетевых операций обуславливает, с одной стороны, повышение скоростей обмена между памятью и параллельными арифметическими устройствами, а с другой стороны, уменьшение времени весового суммирования (умножения и

накопления) за счет применения фиксированного набора команд типа регистр-регистр.

4.3. Специализированные нейрочипы

Основное отличие нейрочипов от других процессоров - это обеспечение высокого параллелизма вычислений за счет применения специализированного нейросетевого логического базиса или конкретных архитектурных решений. Использование возможности представления нейросетевых алгоритмов для реализации на нейросетевом логическом базисе как раз и является основной предпосылкой резкого увеличения производительности нейрочипов.

Проведем классификацию нейропроцессоров по ряду основных признаков:

- *по типу логики* нейропроцессоры разделяют на цифровые, аналоговые и гибридные (рис. 4.2);
- *по типу реализации нейросетевых алгоритмов* – с полностью аппаратной реализаций и с программно-аппаратной реализацией (когда нейроалгоритмы хранятся в ПЗУ);
- *по характеру реализации нелинейных преобразований* – нейропроцессоры с жесткой структурой нейронов (аппаратно реализованных) и нейрокристаллы с настраиваемой структурой нейронов (перепрограммируемые);
- *по гибкости структуры нейронных сетей* – нейропроцессоры с жесткой и переменной нейросетевой структурой (т.е. топология нейронных сетей реализована жестко или гибко).

Производство специализированных нейрочипов ведется во многих странах мира.

Большинство из них ориентируются на закрытое использование (так как создаются для конкретных прикладных систем), однако среди нейрочипов достаточно и универсальных кристаллов (табл. 4.1) [10, 13].

Исследования в области разработки перспективных нейрочипов проводят многие лаборатории и университеты, среди которых можно выделить следующие [2]:

- В США – лаборатории Naval Lab и MIT Lab, Пенсельванский Университет, Колумбийский Университет, Аризонский Университет, Иллинойский Университет и др.
- В Европе – Берлинский Технический Университет, Технический Университет Карлсруе, Институт микроэлектроники г.Штутгарт и др.
- В России – МФТИ, Ульяновский Государственный Технический Университет, МГТУ им. Н.Э. Баумана, Красноярский Государственный технический университет, Ростовский Госуниверситет и др.

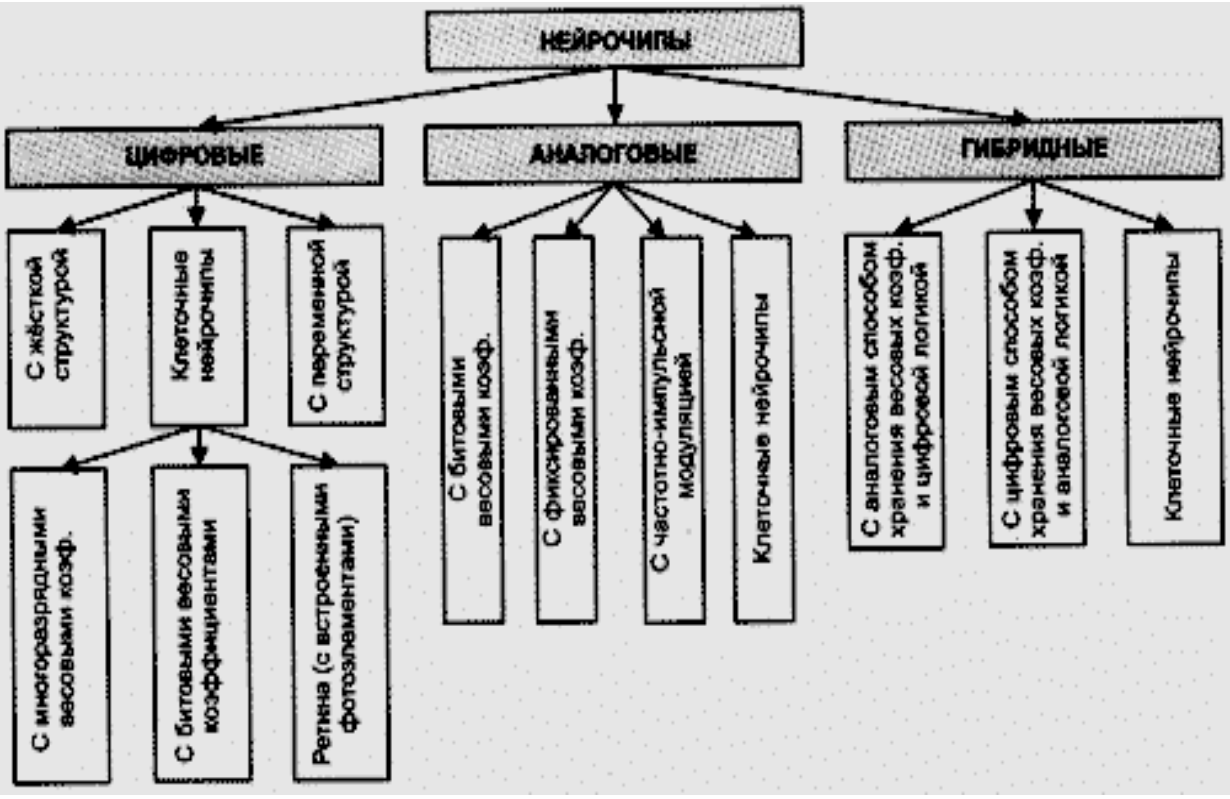


Рис. 4.2. Классификация нейрочипов

Так, в МГТУ им. Н.Э. Баумана нейросетевой и нейрокомпьютерной проблематикой занимаются более десятка лабораторий на четырех факультетах: «Информатики и систем управления», «Специального машиностроения», «Радиоэлектроники и лазерной техники», «Биомедицинских систем».

Для создания единого образовательного пространства в области нейроинформатики на кафедре «Конструирование и технология производства электронной аппаратуры» МГТУ им. Н.Э. Баумана проводятся работы по созданию интерактивной глобальной информационно-обучающей системы в области нейрокомпьютеров и нейроинформатики. Основными направлениями деятельности этой кафедры в области нейросетевых приложений являются: нейроадаптивные системы активного управления пространственными волновыми полями (акустика и вибрации); нейроадаптивные системы управления робототехническими технологическими комплексами; распознавание изображений; обработка сигналов в системах неразрушающего контроля; создание систем интеллектуального управления трафиком в телекоммуникационных системах; исследовательские работы в области нейроинформатики (экспертные системы, аналитические системы, контекстно-поисковые системы и т.п.) [13].

Таблица 4.1. Краткая характеристика специализированных нейрочипов

Наименование	Компания изготовитель	Разрядность, бит	Максимальное число синапсов*	Максимальное число слоев**	Примечание
1	2	3	4	5	6
MA16	Siemens	48	-	-	400 MMAC
NNP (Neural Networks Processor)	Accurate Automation	Nx16	-	-	MIMD, N – число процессоров
CNAPS-1064	Adaptive Solutions	16	128 Кбайт	64	
100 NAP Chip	HNC	32	512 Кбайт	4	4 процессорных элемента с плав. арифметикой
NeuroMatrix NM6403, Такт. частота 50 МГц	Модуль, Россия	64 (векторный процессор), 32 RISC ядро	4096 шт.	24	Совместим с портами TMS320C4x
NeuroMatrix NM6404, Такт. частота 133 МГц	Модуль, Россия	64 (векторный процессор), 32 RISC-ядро	4096 шт.	48	Совместим с портами TMS320C4x
CLNN 32 CLNN 64	Bellcore	32 64	496 1024	32 нейрона	10^8 перекл./с 2 x 10^8 перекл./с
NC 3001	NeuriGam	16	4096 шт.	32	
ZISC 036 (Zero Instruction Set Computer)	IBM	64-разрядные входные векторы	-	36 нейронов	Частота 20 МГц, Векторно-прототипный нейрочип
ETANN 80170NW	Intel	64 входа	Два банка весов 64x80	3 слоя по 64 нейрона	Аналоговая
MD-1220	Micro Dev.	16	64 шт.	8	8 нейронов
MT 19003 - Neural Instruction Set Processor	Micro Circuit Engineering (MCE)	16- разрядный умножитель; 35- разрядный сумматор	-	1	RISC-процессор с 7 специальными командами
NI 1000	Nestor	5-16 (одного нейрона)	-	1024 256-мерных векторов	Векторно-прототипный нейрочип
NLX420 (NLX 110, 230)	Adaptive Logic	16	1 Мбайт	16	16 процессорных элементов
OBL Chip	Oxford Computer	16	16 Мбайт	-	

Окончание таблицы 4.1

1	2	3	4	5	6
L-Neuro 1.0 **L-Neuro 2.3**	Philips	16 16	1536	16 нейронов (12x16)	26 МГц 60 МГц
Pram-256 Chip	UCLi Ltd.	8 (одного нейрона)	-	256 нейронов	33 МГц
SAND	Datafactory	16	-	4	200 MCPS
ACC		16	-	-	
Геркулес	Россия	16	1 Мбайт	64	
Neuro Classifier	Университет Твенте, DESY	70 входных нейронов	-	6 (внутр) 1 вход., 1 выход	2 x 1010 перекл./с
ANNA	AT&T	Число нейронов 16-256	4096 весов	-	Число входов у нейрона 256-16
WSC (Wafer Scale Integration)	Hitachi	-	64 связи на нейрон	576 нейронов	
SASLM2	Mitsubishi	2 (одного нейрона)	-	4096 (64x64) нейронов	50 МГц
TOTEM	Kent (Uni., UK), di Trento (Italy)	16 (одного нейрона)	-	64 нейрона	30 МГц
Neuron 3120, Neuron 3150	Echelon (США)	8 бит (шина данных)	-	-	Наличие параллельных, последовательных и коммуникациионных портов

*** Максимальное число синапсов определяет размер внутрикристалльной памяти весов;**
**** Максимальное число слоев определяется числом операций умножения с накоплением, выполняемых за один такт для операндов длиной 8 бит**

Показатели производительности некоторых нейрочипов приведены в табл. 4.2 [13].

Рассмотрим наиболее популярные специализированные нейрочипы более подробно [10].

Нейросигнальный процессор NeuroMatrix NM6403

Основа нейрочипа NeuroMatrix NM6403 (компания Модуль, Россия) – ЦП, который является высокопроизводительным ПЦОС. ЦП состоит из двух базовых блоков: 32-разрядного RISC-процессора и 64-разрядного векторного процессора, обеспечивающего выполнение векторных операций над данными переменной разрядности. Имеются два идентичных программируемых интерфейса для работы с внешней памятью различного типа и два коммуникационных порта, аппаратно совместимых с портами ПЦОС TMS320C4x, для возможности построения многопроцессорных систем [15].

Таблица 4.2. Производительность специализированных нейрочипов*

Наименование нейрочипа	Конфигурация	CPS	CPSPW	CPPS	CUPS
NLX420	32-16, 8 bit mode	10M	20K	640M	-
100 NAP	4 chips, 2 M wts, 16 bit mantissa	250M	125	256G	64M
WSI (Hitachi)	576 neuron Hopfield	138M	3.7	10G	-
N64000 (Inova)	64-64-1, 8 bit mode	871M	128K	56G	220M
MA16	1 chip, 25 MHz	400M	15M	103G	-
ZISC036	64 8 bit element inp. Vector	-	-	-	-
MT19003	4-4-1-, 32 MHz	32M	32M	6.8G	-
MD1220	8-8	9M	1M	142M	-
NI 1000	256 5 bit element inp. Vector	40 000 vec in sec.	-	-	-
L-neuro-1	1-chip, 8 bit mode	26M	26K	1.6G	32M
NM6403	8 bit mode, 50 MHz	1200M	150M	77G	-

*** В таблице приведены средние округленные показатели производительности**

Внешний вид нейрочипа представлен на рис. П.1 приложений.

Основные характеристики:

- тактовая частота – 50 МГц (20 нс – время выполнения всех инструкций);
- технология – КМОП 0,5 мкм;
- корпус – 256BGA;
- напряжение питания от 2,7 до 3,6 В;
- потребляемая мощность при 50 MHz около 1,3 Вт.

RISC-ядро:

- 5-ступенчатый 32-разрядный конвейер;
- 32- и 64-разрядные команды (обычно выполняется две операции в одной команде);
- два адресных генератора, адресное пространство – 16 Гбайт;
- два 64-разрядных программируемых интерфейса с SRAM/DRAM-разделяемой памятью;
- формат данных – 32-разрядные целые.
- восемь 32-разрядных регистров общего назначения;
- восемь 32-разрядных адресных регистров;
- специальные регистры управления и состояния;
- два высокоскоростных коммуникационных порта ввода/вывода, аппаратно-совместимых с портами TMS320C4x.

Векторный сопроцессор:

- переменная 1–64-разрядная длина векторных операндов и результатов;
- формат данных – целые числа, упакованные в 64-разрядные блоки, в форме слов переменной длины от 1 до 64 разрядов каждое;
- поддержка как векторно-матричных, так и матрично-матричных операций;
- два типа функций насыщения на кристалле;
- три внутренних 32 × 64-разрядных блока ОЗУ.

Производительность скалярных операций:

- 50 MIPS;
- 200 MOPS для 32-разрядных данных.

Производительность векторных операций:

- от 50 до 50000 MMAC.

I/O и интерфейсы с памятью:

- пропускная способность двух 64-разрядных интерфейсов с памятью – до 800 Мбайт/сек;
- I/O коммуникационные порты – до 20 Мбайт/сек каждый.

Базовыми для кристалла являются вычисления вида:

$$Z_i = f(Yi) = f\,(U_i + e(X_j W_{ij}))\ , \qquad (i = 1, .. , M; j = 1, .., N)\ , \qquad (4.2)$$

где Z_i – выходной сигнал i-го нейрона, X_j – j-й входной сигнал слоя, U_i – смещение i-го нейрона, W_{ij} – весовой коэффициент j-го входа 1-го

нейрона, Y_i – сумма взвешенных входов i-го нейрона, f – функция активации, N – количество входных сигналов слоя, M – количество нейронов в слое.

Операнды Z_i, X_i, U_i и W_{ij} представлены в дополнительном параллельном коде и могут иметь произвольную разрядность.

Особенностями данного кристалла являются:

- возможность работы с входными сигналами (синапсами) и весами переменной разрядности (от 1 до 64 бит), задаваемой программно, что обеспечивает уникальную способность нейрокристалла увеличивать производительность с уменьшением разрядности операндов;
- быстрая подкачка новых весов на фоне вычислений (24 операции умножения с накоплением за один такт при длине операндов 8 бит);
- V-аппаратная поддержка эмуляции нейронных сетей большой размерности;
- реализация функции активации в виде пороговой функции или функции ограничения;
- наличие двух широких шин (по 64 разряда) для работы с внешней памятью любого типа: до 4Мб SRAM и до 16 Гб DRAM;
- наличие двух байтовых коммуникационных портов ввода/вывода, аппаратно совместимых с коммуникационными портами TMS320C4x для реализации параллельных ВСМП большой производительности;
- возможность работы с данными переменной разрядности по различным алгоритмам, реализуемым программами, хранящимися во внешнем ОЗУ.

Технические характеристики:

- число вентилей на кристалле – 100000;
- размер кристалла – 10 мм × 10,5 мм при технологии 0,7 мкм;
- потребляемая мощность – не более 3 Вт;
- пиковая производительность для байтных операндов – 720 MCPS при тактовой частоте 30 МГц; для бинарных операций – 8640 MCPS.

Кристалл может применяться как базовый элемент нейрокомпьютеров, реализованных в виде карт и модулей для ПК (нейроускорителей), а также в конструктивно-автономных нейрокомпьютерах. Внешний вид нейрочипа представлен на рис. П.1 Приложений. Используется в нейрокомпьютерах компании «Модуль» [15].

Нейросигнальный процессор NeuroMatrixR NM6404

NeuroMatrixR NM6404 представляет собой высокопроизводительный ПЦОС-образный RISC-процессор. В его состав входят два основных блока: 32-разрядное RISC-ядро и 64-разрядный векторный сопроцессор для поддержки операций над векторами с элементами переменной разрядности. NM6404 по системе команд совместим с предыдущей версией NM6403. Имеются два программируемых интерфейса для работы с внешней памятью

различного типа и два коммуникационных порта, аппаратно совместимых с портами ПЦОС TMS320C4x, для возможности построения многопроцессорных систем.

Технические параметры [15]:

- тактовая частота – 133 МГц (8 нс – время выполнения любой команды);
- технология КМОП 0,25 мкм;
- корпус PQFP256;
- напряжение питания 2,5 В; 3,3 В; 5 В;
- потребляемая мощность – около 1,0 Вт;
- условия эксплуатации: –40...+80 С.

RISC-ядро:

- 5-ступенчатый 32-разрядный конвейер;
- 32- и 64-битовые команды (обычно выполняется две операции в одной команде);
- 2 Мбита внутреннее ОЗУ;
- доступ к внутренней памяти соседей;
- два адресных генератора с адресуемым пространством – 16 Гбайт;
- два 64-разрядных программируемых интерфейса с SDRAM/SRAM/DRAM/Flash разделяемой ПЗУ;
- четыре одновременных доступа к внутренней памяти; широковещательный режим доступа к внешней памяти;
- 64 К загружаемого ПЗУ;
- формат данных – 32-разрядные целые;
- четыре канала ПДП;
- два коммуникационных порта ввода/вывода, аппаратно совместимых с портами TMS320C4x; JTAG-совместимый отладочный интерфейс; система управления потребляемой мощностью;

Векторный сопроцессор:

- от 1 до 64-разрядная длина векторных операндов и результатов;
- формат данных – целые числа, упакованные в 64-разрядные блоки, в форме слов переменной длины от 1 до 64 разрядов каждое;
- поддержка векторно-матричных и матрично-матричных операций; 16 тактов на перезагрузку матрицы коэффициентов;
- реализация на кристалле двух типов активационной функции.

Производительность скалярных операций:

- 133 MIPS;
- 399 MOPS для 32-разрядных данных.

Производительность векторных операций – от 133 до 38000 ММАС.

I/O и интерфейсы с памятью:

- пропускная способность двух 64-разрядных интерфейсов с памятью – 2128 Мбайт/сек;
- I/O коммуникационные порты – до 20 Мбайт/сек каждый.

Внешний вид нейрочипа представлен на рис. П.2 Приложений.

Нейрочип NNP компании Accurate Automation Corp.

Состоит из нескольких миниатюрных процессоров, работающих параллельно. Каждый из них представляет собой быстрый 16-разрядный вычислитель с памятью для хранения синаптических весов. Процессор использует всего 9 команд. Процессоры на кристалле связаны друг с другом локальной шиной. NNP создан в коммерческих целях и доступен на рынке.

В комплект поставки процессора включены средства разработки программ, а также библиотека подпрограмм с реализованными нейросетевыми алгоритмами, такими как сети Хопфилда, сети Кохонена и другими. Нейрочип поставляется на платах под шины ISA, VME. Производительность – 140 MCPS для однопроцессорной системы и 1,4 GCPS для 10-процессорной системы.

Нейрочип MA16 компании Siemens

Изготовлен по технологии КМОП (1 мкм), состоит из 610 тыс. транзисторов и выполняет до 400 млн операций умножения и сложения в секунду. Используется в качестве элементной базы нейрокомпьютера Synapse 1 и нейроускорителей Synapse 2 и Synapse 3 (распространяемых сегодня на рынке французской фирмой Tiga Technologies).

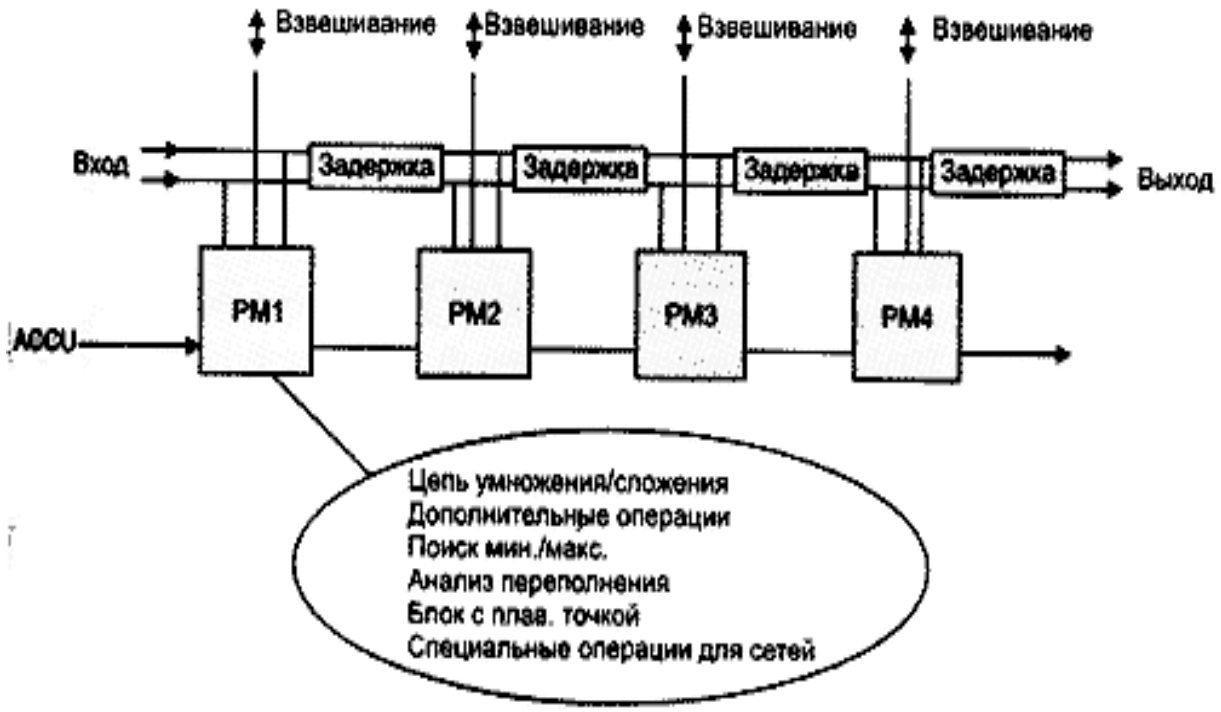


Рис. 4.2. Функциональная схема нейрочипа MA16

MA16 представляет собой программируемый каскадируемый процессор для векторных и матричных операций, поддерживающий на аппаратном уровне следующие операции:

- матричное умножение;
- матричное сложение/вычитание;
- нормировка результата;
- вычисление векторной нормы (метрики L1 и L2);
- вычисление векторного расстояния (мера Манхэттена, геометрическое расстояние).

Нейрочип содержит четыре идентичных процессорных элемента, работающих параллельно (рис. 4.2). Входные данные имеют точность 16 бит, тактовая частота 50 МГц. Для операций матричного умножения/сложения скорость вычислений достигает 8 × 108 операций/с. Программное обеспечение работает в среде UNIX/XWIND и реализовано на C++. Нейронная сеть тоже описывается на C++ или может вводится интерактивно с помощью графического интерфейса типа OSF/Motif, что позволяет визуализировать конфигурацию чипа после отображения на него структуры сети. Хорошо развиты средства тестирования и эмуляции. С 1995 года MA16 является коммерчески доступным.

Внешний вид нейрочипа представлен на рис. П.3 Приложений.

Нейрочип MD1220 компании MicroDevices

Содержит восемь нейронов с 8 связями и 16-разрядные сумматоры. Во внутрикристалльной памяти хранятся 16-разрядные веса. Входы имеют последовательные одноразрядные умножители с продолжительностью такта 7,2 мкс. Средняя производительность около 9 MCPS.

Нейрочип L-Neuro компании Philips

Один из первых нейрочипов. На сегодня широко известны две его модификации L-Neuro 1.0 и L-Neuro 2.3. Вторая версия имеет 12 слоев, а первая один слой из шестнадцати одноразрядных, или двух восьмиразрядных, или четырех 4-разрядных, или двух восьмиразрядных процессорных элементов, т.е. имеет возможность работать в мультиразрядном режиме. На кристалле реализован 1Кбайт памяти для хранения 1024 8-разрядных или 512 16-разрядных весов. Гибкая каскадируемая структура нейрочипа позволяет использовать его для различных нейросетевых парадигм. При реализации 64 восьмиразрядных процессорных элементов средняя производительность составляет 26 MCPS (32 MCUPS).

Нейрочип NLX-420 компании NeuroLogix

Каждый из 16 процессорных элементов нейрочипа содержит 32-разрядный сумматор, логику параллельного выполнения 16 умножений. Средняя производительность 300 MCPS. Также имеется возможность каскадирования и мультиразрядных вычислений.

Нейрочип ETANN 80170NX компании INTEL

Аналоговая СБИС ETANN 80170NX содержит 64 входа, 16 внутренних уровней и 64 нейрона (пороговый усилитель с сигмоидальной передаточной функцией). Каждый вход соединен с 64 синапсами. Передаточная функция нейрона близка к сигмоидальной функции.

Усиление передаточной функции определяет чувствительность нейрона. Низкое значение усиления позволяет интерпретировать выход нейрона как аналоговый, а высокое – как цифровой.

Нейрочип имеет следующие параметры:

- максимальное значение выхода нейрона определяется напряжением Vrefo;
- веса ограничены интервалом [-2,5, 2,5];
- скорость прохождения сигнала по одному слою зависит от усиления и примерно равна 1,5 мкс, что и определяет быстродействие;
- точность выполнения операций примерно эквивалентна 6 битам, быстродействие – 1,3 – 109 переключений/с.

Обучение выполняется методом обратного распространения с помощью INNTS. Применяемое системное окружение – специальная версия пакета DynaMind. Обучение выполняется до получения приемлемого уровня ошибки выхода сети, и после достижения удовлетворительной работы веса загружаются в СБИС. Для реальной работы такого обучения недостаточно, так как программа симуляции не может точно смоделировать аналоговую работу СБИС и, например, не отслеживает флуктуации в передаточной функции каждого нейрона. Поэтому следующий этап обучения представляет собой так называемый CIL Training, когда после каждого цикла веса записываются в СБИС, и выход сети непосредственно используется в процессе обучения.

Точность ETANN 5 – 6 разрядов для весов и выходов.

Поскольку ETANN представляет собой аналоговую СБИС, то для ее надежной работы важны стабильные внешние условия. Специально сконструированный для этого модуль обеспечивает:

- низкую пульсацию источника питания (до 5 мВ при напряжении питания 5 В);
- температурную стабильность до 1°С при 18°С (потребляемая мощность ETANN 5 Вт).

Нейрочип CLNN32/CLNN64 компании Bellcore

Гибридный нейрочип CLNN32 состоит из 32 нейронов и 496 двунаправленных адаптивных синапсов. CLNN64 содержит только 1024 адаптивных синапса. В наборе CLNN32/CLNN64 все нейроны взаимосвязаны, так что любая топология сети отображается подбором синапсов. Динамика нейронной сети полностью аналоговая, но значения синапсов хранятся/обновляются в цифровом виде с точностью 5 бит. На аппаратном уровне реализовано обучение сети – подбор весов происходит по алгоритму обучения машины Больцмана или Mean Field. Внутри также имеется некоррелированный генератор шума (32 канала), используемый при обучении по методу машины Больцмана. CLNN32 может быть использован независимо или совместно с CLNN64 для построения более сложной архитектуры нейронной сети.

Производительность СБИС достигает 108 переключений/с (при работе с CLNN64 удваивается). Для CLNN32 это означает, что примерно 105 32-битовых образцов/с или 32 аналоговых канала (с полосой пропускания 50 кГц) могут быть использованы для быстрого распознавания/обучения. Время распространения для одного слоя нейронов до 1 мкс. «Охлаждение» (по методу Больцмана) или MF обучении требует 10 – 20 мкс. По сравнению с ETANN СБИС CLNN32 имеет следующие преимущества:

- быстрое обучение (микросекунды по сравнению с часами при C1L процессе);
- эффективный алгоритм обучения Больцмана, обеспечивающий быстрое нахождение "почти оптимального" решения;
- простые и быстрые процедуры чтения/записи весов, выполняемые в цифровом виде, что значительно увеличивает скорость обмена между сетевым сервером и клиентами в сети;
- легкая каскадируемостъ.

Нейрочип ANNA компании AT&T

Другим примером реализации гибридного нейрочипа является СБИС ANNA. Логика нейрочипа – цифровая, хранение весов – аналоговое (на элементах динамической (конденсаторной) памяти). Нейрочип имеет следующие характеристики:

- содержит 4096 весов;
- максимальное число нейронов 256;
- точность весов – 6 разрядов
- для однослойной сети 64 × 64 производительность нейрочипа достигает 2,1 GCPS.

Нейрочип NeuroClassifier

Аналоговая СБИС NeuroClassifier создана в университете Твенте совместно с компанией DESY. Архитектура нейрочипа состоит из входного

слоя (70 входов, полоса пропускания до 4 Гбайт/с), шести внутренних слоев и одного выходного нейрона. Точность аналогового умножения – 5 бит, время решения – 20 нс, что позволяет использовать NeuroClassifier в триггере первого уровня.

Нейрочип SAND/1 компании Datafactory

Компания Datafactory (бывшая INCO) предложила на рынок нейрочип SAND/1 (Simple Applicable Neural Device). SAND/1 представляет собой каскадно-соединенные систолические процессоры, оптимизированные для быстрого решения задач в нейросетевом базисе. Производительность одного процессора составляет 200 MCPS (миллионов связей в секунду). Процессор имеет четыре 16-битных потока и 40-битный сумматор.

Нейрочип разработан Исследовательским центром в Карлсруе и Институтом микроэлектроники г. Штутгарта.

Нейрочип N64000 компании Inova

Относится к классу систолических нейропроцессоров. Содержит 80 процессорных элементов, из которых 64 образуют основную матрицу, а 16 являются резервом, 4 Кбайт памяти весов и 32 регистра общего назначения.

Арифметический модуль нейрочипа имеет девять параллельных 16-разрядных умножителя и один 32-разрядный сумматор.

Нейрочип 100 NAP компании Hecht-Nielson Computers

Содержит четыре 32-разрядных процессорных элемента с плавающей точкой. Средняя производительность около 150 MFLOPS, адресуемое адресное пространство внекристалльной памяти 512 Кбайт.

Нейрочип MT19003 компании Micro Circuit Engineering

Относится к классу систолических нейропроцессоров. Основой архитектуры является RISC-ядро с семью специальными командами, 16-разрядный векторный умножитель и 32-разрядный сумматор. Внутрикристалльная память для хранения весов отсутствует. Точность входов и весов 13 разрядов. Средняя производительность 50 MCPS.

Нейрочип NEURON компании Echelon

Ориентирован на создание кластерно-параллельных вычислительных систем. Программно-алгоритмическое обеспечение по управлению кластерной структурой реализовано внутри кристалла. Предложенная архитектура кристалла стала в настоящее время основой стандарта ANSI/EIA 709.1-1999 построения различных АСУ технологическими процессами [2].

В семействе нейрочипов NEURON выделяют: NEURON 3120 и NEURON 3150 (рис. 4.3).

Кристаллы содержат:

- 2К динамической памяти для хранения весов и данных,
- 512 байт (EEPROM), для размещения управляющих программ.

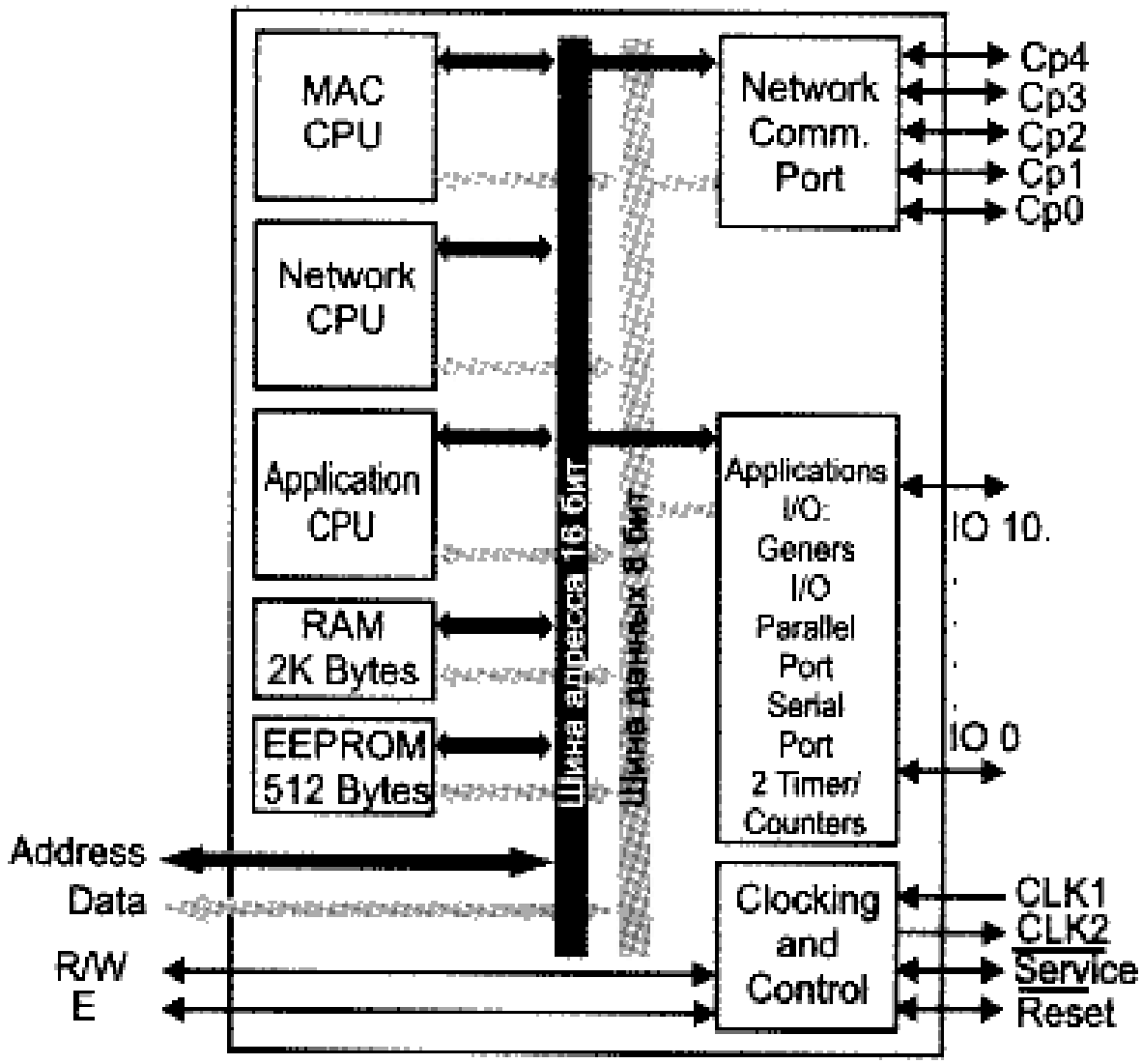


Рис. 4.3. Структурная схема нейрочипа NEURON 3150 компании Echelon

Для выполнения специализированных сетевых и управляющих операций в структуре кристалла имеется два спецвычислителя: Applications CPU, Network CPU. Также следует отметить широкие коммуникационные возможности, реализованные на кристалле.

Нейрочип ZISC036 компании IBM

Нейрочип ZISC036 (Zero Instructions Set Computer) относится к нейрочипам векторно-прототипной архитектуры, т.е. алгоритм обучения строится на соотношении входного вектора и запомненных прототипными векторами весов входов нейронов. Ориентирован на решение широкого круга задач распознавания образов и классификации. Каждый нейрон представляет собой независимый процессор (рис. 4.4).

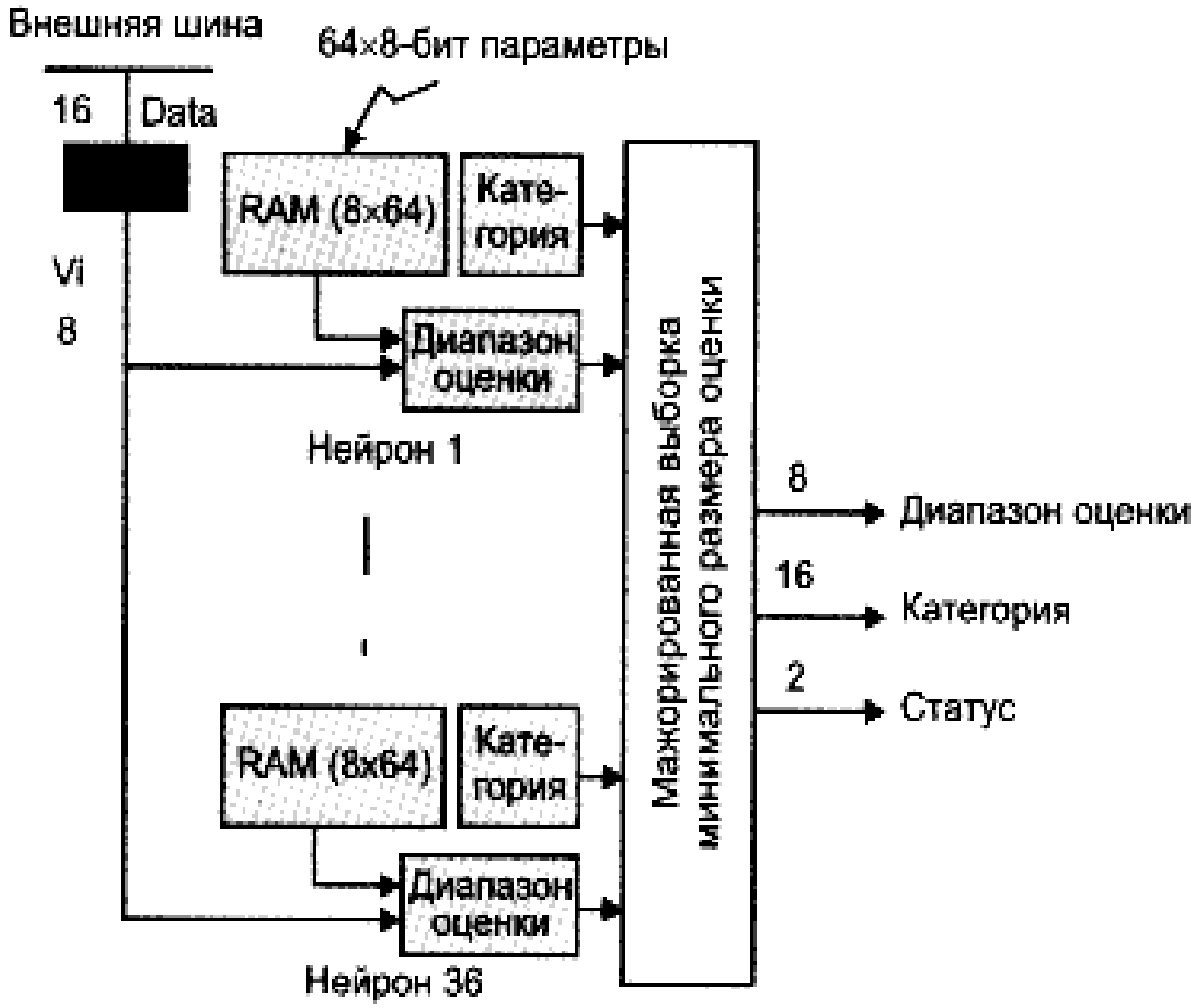


Рис. 4.4. Функциональная схема нейрочипа ZISC компании IBM

Характеристики нейрочипа ZISC036:

- 36 нейронов;
- возможность соединения нескольких процессоров (каскадирования);
- от 1 до 64 компонент во входном векторе;
- напряжение питания 5V;

- потребляемая мощность – 1 W при 16 MHz;
- частота – 0 – 20 MHz;
- CMOS-технология;
- входной вектор загружается через 3,5 мкс, результат появляется через 0,5 мкс.

Для увеличения производительности фирма IBM разработала ISA и PCI модули, включающие параллельно работающие ZISC-процессоры.

4.4. Нейропроцессоры на основе ПЦОС и ПЛИС

4.4.1. Основные понятия

Стремительный переход современных систем управления на цифровые стандарты привел к необходимости обрабатывать с высокой скоростью достаточно большие объемы информации. Сложная обработка и фильтрация сигналов, например распаковка сжатых аудио- и видеоданных, маршрутизация информационных потоков, требует применения достаточно производительных вычислительных систем. Подобные системы могут быть реализованы на различной элементной базе, но наибольшее распространение получили устройства с применением цифровых сигнальных процессоров и ПЛИС [10].

Программируемая логика способна работать на более высоких частотах, но поскольку управление реализовано аппаратно, то изменение алгоритмов работы требует перепрограммирования интегральной схемы. Низкая тактовая частота ПЦОС пока ограничивает максимальную частоту обрабатываемого аналогового сигнала до уровня в 10 – 20 МГц, но программное управление позволяет достаточно легко изменять не только режимы обработки, но и функции, выполняемые ПЦОС. Помимо обработки и фильтрации данных ПЦОС могут осуществлять маршрутизацию цифровых потоков, выработку управляющих сигналов и даже формирование сигналов системных шин ISA, PCI и др.

ПЛИС и ПЦОС получили значительное распространение и в качестве нейропроцессоров.

Особенностью использования ПЦОС и ПЛИС в качестве элементной базы нейрокомпьютеров является то, что ориентация в выполнении нейросетевых операций обуславливает с одной стороны повышение скоростей обмена между памятью и параллельными арифметическими устройствами, а с другой стороны уменьшение времени весового суммирования (умножения и накопления) за счет применения фиксированного набора команд типа регистр – регистр.

4.4.2. Нейропроцессоры, реализованные на основе ПЦОС

ПЦОС, обладая мощной вычислительной структурой, позволяют реализовать различные алгоритмы обработки информационных потоков.

Сравнительно невысокая цена, а также развитые средства разработки программного обеспечения позволяют легко применять их при построении вычислительных систем с массовым параллелизмом.

ПЦОС вот уже на протяжении двух десятилетий являются элементной базой нейрокомпьютеров, реализованных как в виде карт и модулей, так и в виде конструктивно-автономных систем.

Анализ сложившегося рынка показывает, что доминирующие позиции в ближайшем будущем будут занимать крупные компании-производители ПЦОС, такие как Texas Instruments Inc., Analog Devices, Motorola и ряд других, которые способны создавать не только ПЦОС низкой себестоимости, но и ежегодно инвестировать новые разработки и ноу-хау, создавать принципиально новые модели и платформы. Например, компания Texas Instruments Inc. только в 1998 году выделила на исследования и разработки ноу-хау для ПЦОС – 1,2 млрд долларов – величину, близкую к годовому обороту некоторых конкурентов в области производства ПЦОС.

Выбор процессоров той или иной компании для реализации конкретного проекта — многокритериальная задача, и сформулировать более или менее четкую методику выбора практически невозможно. Не отдавая предпочтение ПЦОС той или иной компании, отметим, что изделия Analog Devices и Texas Instruments Inc. годятся для приложений, требующих выполнения больших объемов математических вычислений (таких как цифровая фильтрация сигнала, вычисление корреляционных функций и т.п.). Между тем, для задач, требующих наряду с вычислениями выполнять интенсивный обмен с внешними устройствами (многопроцессорные системы, различного рода контроллеры), чаще используются ПЦОС Texas Instruments Inc. [16, 17], обладающие высокоскоростными интерфейсными подсистемами. Компания Motorola является лидером по объему производства дешевых и достаточно производительных контроллеров на основе 16- и 24-разрядных ПЦОС с фиксированной точкой. Расширенные коммуникационные возможности, наличие достаточных объемов внутрикристалльной памяти для данных и программы, возможность защиты программы от несанкционированного доступа, поддержка режима энергосбережения делают эти микропроцессоры привлекательными для использования не только в качестве специализированных вычислителей, но и в качестве контроллеров, в бытовых электронных приборах, в системах адаптивной фильтрации и т.д.

Высокая производительность, необходимая при обработке сигналов в реальном времени, обусловила широкое распространении транспьютероподобных ПЦОС серий TMS320C4x (компания Texas Instruments Inc.) и ADSP2106x (компания Analog Devices), ориентированных на использование в мультипроцессорных системах (потребительские и функциональные характеристики приведены в табл. 4.3).

Оценка производительности ряда процессоров при выполнении некоторых популярных алгоритмов ЦОС приведена в табл. 4.4.

Таблица 4.3. Сравнительные характеристики ПЦОС ADSP21061 и TMS320C40

Характеристика/процессор	ADSP21061	TMS320C40/ TMS320C44
Производительность		
Время выполнения команд, нс	20,0	33,0
Пиковое значение MFLOPS	150,0	60,0
Стоимость		
Цена, $	49	176 (‘C44 – 99)
Соотношение производительность - цена, MFLOPS/$	3,1	0,34 (‘C44 – 0,6)
Скорость выполнения типовых алгоритмов		
Комплексное БПФ для 1024 выборок, мс	0,37	0,97
Характеристики ЦП		
Количество регистров данных	32	12
Циклические буфера	32	1 (Fixed Length)
Возможности ввода-вывода		
Каналы ПДП, шт	6	6
Последовательные порты	2	-
Максимальная пропускная способность, МБайт/сек	300	60
Внутрикристалльная память, 32-разрядных слов	32K	2K
Общая внутрикристалльная память, Кбит	1024	64
Поддержка мультипроцессорных операций	6	6
Интерфейс	Параллельный	-

Таблица 4.4. Производительность процессоров при выполнении ряда известных алгоритмов

Наименование теста	Intel Pentium II, 300 МГц	Intel Pentium MMX, 200 МГц	Texas Instruments TMS320C40, 50 МГц	НТЦ "Модуль" NM6403, 40 МГц
Фильтр Собеля (размер кадра 384 × 288 байт), кадров/с.	-	21	6,8	68
Быстрое преобразование Фурье (256 точек, 32 разр.), мкс (тактов)	200	-	464 (11588)	102 (4070)
Преобразование Уолша-Адамара (21 шаг, входные данные – 5 бит), с	2,58	2,80	-	0,45

При создании нейрокомпьютеров на базе ПЦОС необходимо помнить, что они обладают высокой степенью специализации. В ПЦОС широко используются методы сокращения длительности командного цикла, характерные для универсальных RISC-процессоров, такие как конвейеризация на уровне отдельных микроинструкций и инструкций, размещение операндов большинства команд в регистрах, использование теневых регистров для сохранения состояния вычислений при переключении контекста, разделение шин команд и данных (гарвардская архитектура).

В то же время для сигнальных процессоров характерным является наличие аппаратного умножителя, позволяющего выполнять умножение как минимум двух чисел за один командный такт. Другой особенностью сигнальных процессоров является включение в систему команд таких операций, как умножение с накоплением MAC ($c = a \times b + c$) с указанным в команде числом выполнений в цикле и с правилом изменения индексов используемых элементов массивов A и B, т.е. уже реализованы прообразы базовых нейроопераций – взвешенное суммирование с накоплением), инверсия бит адреса, разнообразные битовые операции. В ПЦОС реализуется аппаратная поддержка программных циклов, кольцевых буферов. Один или несколько операндов извлекаются из памяти в цикле исполнения команды.

Реализация однотактного умножения и команд, использующих в качестве операндов содержимое ячеек памяти, обуславливает сравнительно низкие тактовые частоты работы сигнальных процессоров. Специализация не позволяет поднимать производительность за счет быстрого выполнения коротких команд типа R,R➔R, как это делается в универсальных процессорах. Этих команд просто нет в программах ЦОС.

Ведущие компании-производители выпускают ПЦОС двух разновидностей, существенно отличающихся по точности вычислений и по цене: более дешевые процессоры для обработки данных в формате с фиксированной точкой и процессоры, аппаратно поддерживающие операции над данными в формате с плавающей точкой.

Типичные ЦОС-операции требуют выполнения множества простых сложений и умножений.

Сложение и умножение требуют:

- произвести выборку двух операндов;
- выполнить сложение или умножение (обычно и то и другое);
- сохранить результат или удерживать его до повторения.

Для выборки двух операндов за один командный цикл необходимо осуществить два доступа к памяти одновременно. Но в действительности кроме выборки двух операндов необходимо еще сохранить результат и прочитать саму команду. Поэтому число доступов в память за один командный цикл будет больше двух и, следовательно, ПЦОС процессоры поддерживают множественный доступ к памяти за один и тот же командный цикл. Но невозможно осуществить доступ к двум различным адресам в памяти одновременно, используя для этого одну шину памяти. Существует два вида архитектур ПЦОС процессоров, позволяющих реализовать механизм множественного доступа к памяти:

- гарвардская архитектура;
- модифицированная архитектура фон Неймана.

Гарвардская архитектура подразумевает две физически разделенные шины данных. Это позволяет осуществить два доступа к памяти одновременно: гарвардская архитектура выделяет одну шину для выборки инструкций (шина адреса), а другую – для выборки операндов (шина данных). Но для выполнения ПЦОС операций этого недостаточно, так как в основном все они используют по два операнда. Поэтому гарвардская архитектура применительно к цифровой обработке сигналов использует шину адреса и для доступа к данным. Важно отметить, что часто необходимо произвести выборку трех компонентов – инструкции с двумя операндами, на что собственно гарвардская архитектура неспособна. В таком случае данная архитектура включает в себя кэш-память. Она может быть использована для хранения тех инструкций, которые будут использоваться вновь. При использовании кэш-памяти шина адреса и шина данных остаются свободными, что делает возможным выборку двух операндов. Такое расширение – гарвардская архитектура плюс кэш – называют расширенной гарвардской архитектурой или SHARC (аббревиатура введена компанией Analog Devices).

Гарвардская архитектура требует наличия двух шин памяти. Это значительно повышает стоимость производства чипа. Так, например, ПЦОС работающий с 32-битными словами и в 32-битном адресном пространстве

требует наличия, по крайней мере, 64 выводов для каждой шины памяти, а в сумме получается 128 выводов. Это приводит к увеличению размеров чипа и к трудностям при проектировании схемы.

Архитектура фон Неймана использует только одну шину памяти. В то же время, она обладает и рядом положительных черт:

- более дешевая при реализации;
- требует меньшего количества выводов шины;
- является более простой в использовании, так как программист может размещать и команды и данные в любом месте свободной памяти.

Рассмотрим некоторые наиболее перспективные ПЦОС с точки зрения реализации нейрокомпьютеров.

4.4.3. ПЦОС компании Analog Devices

Реализация нейровычислителей высокой пространственной размерности требует все более производительной элементной базы. Для преодоления возникающих трудностей разработчики используют два подхода: улучшение характеристик уже имеющихся процессоров и увеличение производительности путем разработки новых архитектур. Первый способ ограничен увеличением производительности в 5 – 8 раз. Второй способ предполагает разработку архитектур, которые были бы наиболее удобны в конечном приложении и оптимизированы для конкретного языка программирования.

Компания Analog Devices ведет разработки в обоих направлениях. Так, ядро первого 32-разрядного процессора ADSP-21020 производительностью 30 MFLOPS было усовершенствовано, что привело к созданию нового процессора ADSP-21065L с максимальной производительностью 198 MFLOPS, что соответствует ускорению в 6,6 раз. Работая над дальнейшим увеличением производительности, оптимизируя архитектуру существующих процессоров, был разработан новый сигнальный микропроцессор ADSP-2116x с тактовой частотой 100 МГц производительностью 600 MFLOPS.

Среди особенностей ПЦОС семейства ADSP-2116x можно отметить:

- Быстрые и гибкие модули арифметики. Все команды выполняются за один такт. Набор команд микропроцессора наряду с традиционными арифметическими операциями включает такие, как 1/x, 1/R(x), команды сдвига, циклического сдвига, комбинации операций сложения/вычитания с умножением.
- Независимые потоки данных в (из) вычислительные (х) модули (ей). За один такт процессор может одновременно считать (записать) два операнда в регистровый файл, загрузить два операнда в АЛУ, принять два операнда в умножитель, АЛУ и умножитель могут вырабатывать два результата (или три, если АЛУ выполняет операцию совместно со сложением/вычитанием). 48-битовое командное слово позволяет

задавать в одной инструкции параллельное выполнение арифметических операций и обмен данными.

- Повышенную точность и расширенный динамический диапазон выполняемых операций. Все представители микропроцессорного семейства оперируют с данными в 32-битовом формате с плавающей точкой, 32-битовыми целочисленными данными (в дополнительном коде и беззнаковыми) и 40-битовыми данными расширенной точности с плавающей точкой. Повышенная точность вычислений достигается благодаря уменьшению ошибки округления результата в вычислительных модулях. Аккумулятор для 32-разрядных данных с фиксированной точкой имеет 80 разрядов.
- Наличие двух генераторов адреса. Генераторы адреса обеспечивают пред- или постформирование прямого или косвенного адреса данных, выполняют над адресами модульные и бит-реверсные операции.
- Эффективные средства формирования последовательности команд и механизм организации программных циклов. Инициализация, возврат на начало и выход из программного цикла выполняется за один процессорный цикл для уровня вложенности до шести. Процессор аппаратно поддерживает выполнение команд перехода и перехода с задержкой.

Универсальное АЛУ ПЦОС, устройство барабанного сдвига и универсальный умножитель функционируют независимо, обеспечивая высокую степень внутреннего параллелизма операций. Регистровый файл общего назначения служит для обмена данными между вычислительными модулями и внутренней шиной, а также для запоминания промежуточных результатов. Регистровый файл содержит 32 регистра (16 – первичных и 16 – вторичных), имеет 10 портов и, совместно с гарвардской архитектурой, позволяет организовывать эффективный обмен между вычислительными модулями и памятью. Расширенная гарвардская архитектура процессора позволяет выбирать до двух операндов и команду из кэш-памяти команд за один цикл.

ПЦОС ADSP-210xx содержат высокопроизводительную кэш-память команд. Кэш-память работает избирательно: кэшируются только те команды, выборка которых конфликтует с выборкой данных из памяти программ.

Адресные генераторы (DAG1 и DAG2) обеспечивают аппаратную реализацию циклических (кольцевых) буферов, позволяющих эффективно выполнять фильтрацию и Фурье-преобразование, для которых требуется циклическое изменение адресов обрабатываемых данных. Физически циклический буфер может быть расположен, начиная с любого адреса памяти, а для ссылки на его содержимое используются регистровые указатели. Два DAG содержат 16 первичных и 16 вторичных регистров, что позволяет работать одновременно с 32 циклическими буферами.

4.4.4. ПЦОС компании Texas Instruments Inc.

Компания Texas Instruments Inc. на рубеже столетий оказалась в заметно обновленном виде [17]. Руководство компании приняло стратегическое решение сконцентрировать силы на упрочении лидирующего положения на рынке ПЦОС, а также других изделий, в первую очередь аналоговых, необходимых для системной интеграции процессоров в прикладные системы.

В период с 1998 по 2000 г. компанией Texas Instruments Inc. были проданы подразделения по производству компьютеров-ноутбуков, схем памяти, оборонной электроники и были приобретены известные фирмы, занимающиеся разработкой прикладного программного обеспечения для ПЦОС (GO DSP, TARTAN, AMATI, Spectron Microsystems). В результате в 1998 году доля Texas Instruments Inc. на рынке ПЦОС вплотную приблизилась к 50% (по результатам 1997 года – 45%). Кроме того, компания Texas Instruments Inc. вышла на первое место в мире по продажам аналоговых и аналого-цифровых схем. Этому способствовало также состоявшееся в 2001 году слияние с компанией Burr-Brown.

В области технических решений в компании Texas Instruments Inc. также произошел ряд существенных изменений. В 1999 году начато массовое производство кремния по запатентованной технологии TimeLine с разрешением 0,18 мкм.

ПЦОС с фиксированной точкой компании Texas Instruments Inc. представлены сериями (рис. 4.5) [16, 17]: TMS320C1x, TMS320C2x, TMS320C2xx, TMS320C5x и TMS320C62x. Класс ПЦОС с плавающей точкой включает ПЦОС TMS320C3x, TMS320C4x и TMS320C67x.

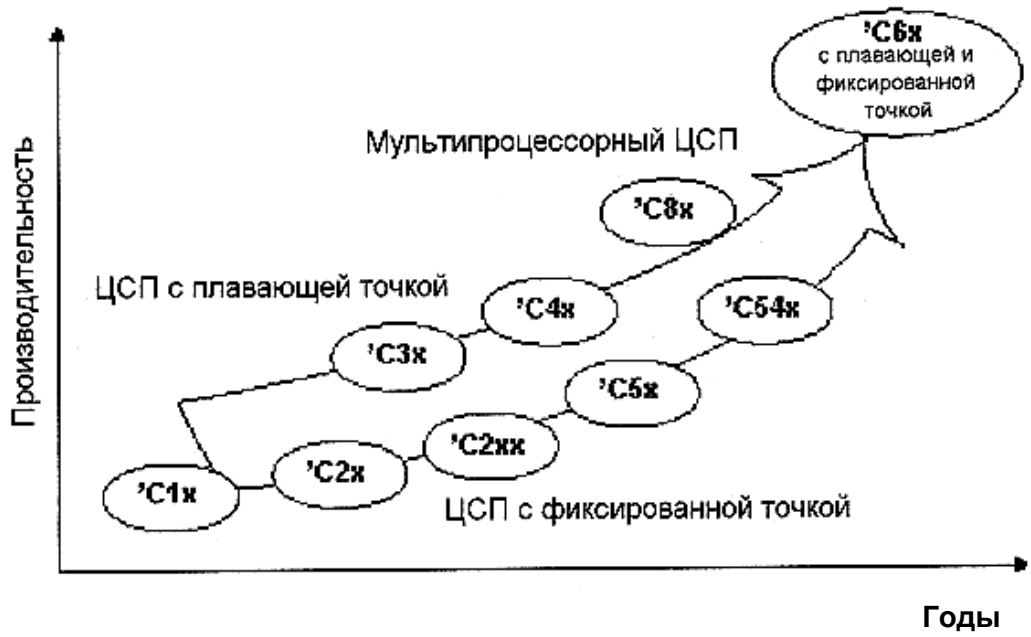


Рис 4.5. Серии ПЦОС TMS320 компании Texas Instruments Inc.

ПЦОС TMS320C8x также поддерживает операции с плавающей точкой и представляет собой мультипроцессорную систему, выполненную в одном кристалле.

Три серии – TMS320C2000, TMS320C5000 и TMS320C6000, по мнению экспертов компании, в ближайшем будущем должны покрыть весь диапазон возможных применений ПЦОС, предоставляя потребителю выбор ПЦОС по критерию "производительность / стоимость / потребляемая мощность".

ПЦОС серии TMS320C2000 предназначены для решения задач встроенных применений и управления; процессоры отличаются развитой периферией и невысокой стоимостью.

Данную серию представляют универсальные ПЦОС подсерии TMS320C20x и подсерии TMS320C24x для цифрового управления электродвигателями.

ПЦОС серии TMS320C5000 ориентированы на рынок малопотребляемых портативных устройств и мобильной связи. ПЦОС подсерии TMS320C54xx оптимизированы по быстродействию (до 200 MIPS) и минимальному энергопотреблению (до 32 mA/MIPS). При этом массовое использование технологии 0,18 мкм позволило снизить стоимость отдельных ПЦОС данной подсерии до 5 $ при производительности 100 MIPS.

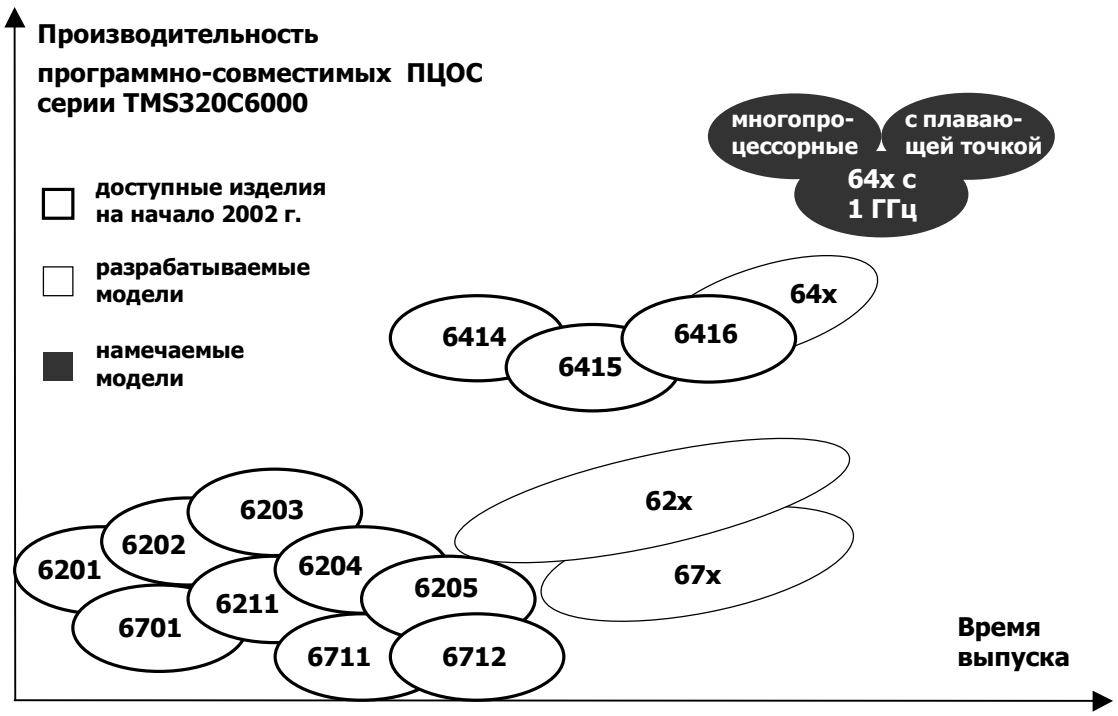


Рис. 4.6. Высокопроизводительные ПЦОС серии TMS320C6000

ПЦОС серии TMS320C6000

Данные ПЦОС характеризуются максимальной производительностью для применений, требующих предельных скоростей вычислений как с фиксированной, так и с плавающей точкой. Обе подсерии, TMS320C62x – ПЦОС с фиксированной точкой и быстродействием 1600 MIPS и TMS320C67x – ПЦОС с плавающей точкой и производительностью от 1 GFLOPS, программно совместимы.

Типовые области применений ПЦОС серии TMS320C6000 [16, 17] – многоканальные модемы, базовые станции, устройства обработки изображений и др. ПЦОС серии отличает относительно высокая производительность и стоимость (рис. 4.6).

Высокая производительность достигается за счет внедрения параллельной архитектуры VelociTI, реализованной на основе технологии VLIW (“Very Long Instruction Word” или «очень длинного командного слова»), а также за счет применения ряда других аппаратных решений и средств разработки.

По оценкам специалистов, применение данной архитектуры в будущем позволит, при сохранении совместимости по командам, достичь рубежей 8000 MIPS для ПЦОС с фиксированной точкой и 3 GFLOPS для ПЦОС с плавающей.

Ожидается также существенное удешевление ПЦОС данной серии как за счет общего снижения стоимости устройств при совершенствовании технологии, так и за счет выпуска новых моделей ПЦОС.

Изготовляются и широко применяются следующие три разновидности ПЦОС серии TMS320C6000 (табл. 4.5):

- подсерия ПЦОС TMS320C62x – устройства с фиксированной точкой и производительностью от 1200 до 2400 MIPS;

Таблица 4.5. Производительность подсерий ПЦОС TMS320C6000

Подсерии TMS320C6000	Производительность			
	Тактовая частота, МГц	MIPS/ MFLOPS	MMAC (16-разрядные слова)	MMAC (8-разрядные слова)
TMS320C62x	150-300	1200-2400 MIPS	300-600	300-600
TMS320C64x	400-600	3200-4800 MIPS	1600-2400	3200-4800
TMS320C67x	100-225	600-1350 MFLOPS	200-550	200-550

- подсерия ПЦОС TMS320C64x – устройства с фиксированной точкой и производительностью от 3200 до 4800 MIPS. Данные ПЦОС являются наиболее скоростными (табл. 4.5) и предназначены как для широкого применения (TMS320C6414), так и для использования в мультимедийных (TMS320C6415) и телекоммуникационных (TMS320C6416) приложениях;
- подсерия ПЦОС TMS320C67x – устройства с плавающей точкой и производительностью от 600 до 1350 MFLOPS.

Оценка продолжительности выполнения популярных алгоритмов в системах на основе ПЦОС серии TMS320C6000 приведена в табл. 4.6.

При проектировании ПЦОС серии TMS320C6000 особое внимание изготовителя уделялось снижению времени, которое понадобится пользователю для разработки и выпуска конечных систем. Сокращению этих сроков способствует свойство совместимости устройства с фиксированной точкой с соответствующим устройством с плавающей точкой.

ПЦОС TMS320C67x имеют совместимость по командам и по выводам микросхем с соответствующими ПЦОС TMS320C62x, что позволяет разработчику быстро выполнять прототипы, используя плавающую точку, и легко переходить к ПЦОС с фиксированной точкой для снижения стоимости изделия при производстве. То есть вначале разработчик может взять за основу ПЦОС с плавающей точкой, отработать все элементы устройства, определить оптимальные алгоритмы обработки данных. При этом большие запасы по производительности и по точности вычислений позволяют заниматься именно алгоритмами, а не экономией ресурсов. После, когда все параметры определены, наступает этап оптимизации системы с учетом наработанных решений и перевод ее на более дешевый ПЦОС с фиксированной точкой.

Таблица 4.6. Оценка продолжительности выполнения популярных алгоритмов

Подсерии TMS320C6000	**Быстрое преобразование Фурье (FFT)** **комплексный спектр, длительность N = 1024, Radix 4**		**Фильтрация сигналов (Digital Filtering)** **фильтр с КИХ, число выходных точек M = 100 (64 – для TMS320C67x)**	
	тактов процессора	**мкс**	**тактов процессора**	**мкс**
TMS320C62x	**13228**	**66,0**	**6410**	**23,0**
TMS320C64x	**6002**	**12,0**	**1019**	**2,0**
TMS320C67x	**18055**	**108,3**	**2216**	**13,3**

Данный подход, предопределил переход от аппаратно-ориентированной среды разработки к программным моделям, что делает процесс разработки более быстрым, дешевым и простым.

Изготовитель также производит широкий ассортимент аналоговых и аналого-цифровых устройств, ориентированных на применение совместно с ПЦОС серии TMS320C6000.

ПЦОС всех трех серий могут комплектоваться современными средствами разработки и отладки программ, объединенных единым пользовательским интерфейсом на базе программных средств Code Explorer и Code Composer Studio [16, 17].

ПЦОС компании Texas Instruments Inc. разделяются на два класса: это процессоры для обработки чисел с фиксированной точкой и процессоры для обработки чисел с плавающей точкой (рис. 4.5). Первый класс представлен тремя семействами процессоров, базовыми моделями которых являются соответственно TMS320C10, TMS320C20, TMS320C50. Второй класс включает процессоры TMS320C30, TMS320C40, TMS320C80, которые поддерживают операции с плавающей точкой и представляют собой мультипроцессорную систему, выполненную на одном кристалле, а семейство TMS320C6x включает как процессоры с фиксированной, так и с плавающей точкой.

ПЦОС более поздних серий TMS320 наследуют основные архитектурные особенности и совместимы "снизу вверх" по системе команд (чего нельзя сказать о процессорах, входящих в разные семейства). Процессоры компании Texas Instruments обладают высокоскоростными интерфейсными подсистемами и поэтому их предпочтительнее использовать для тех задач, в которых требуется выполнение интенсивного обмена с внешними устройствами (микропроцессорные системы, различного рода контроллеры).

ПЦОС серии TMS320C80

Данные ПЦОС, работающие с производительностью в 2 млрд операций в секунду, представляют собой комбинацию из пяти процессоров, реализованных по MIMD-архитектуре. На одном кристалле реализованы одновременно две технологии – ПЦОС и RISC, расположены один управляющий RISC-процессор и четыре 32-разрядных цифровых сигнальных процессора усовершенствованной архитектуры с фиксированной точкой (ADSP0-ADSP-3), обладающие высокой степенью конвейеризации и повышенной до 64 бит длиной слова инструкций, а это в свою очередь позволяет описывать сразу несколько параллельно выполняемых команд. Каждый из процессоров работает независимо друг от друга и может программироваться отдельно друг от друга и выполнять различные или одинаковые задачи, обмениваясь данными через общую внутрикристалльную кэш-память.

Суммарная производительность TMS320C80 на регистровых операциях составляет 2 млрд RISC-подобных команд в секунду. Благодаря столь высокой производительности TMS320C80 может заменить при реализации приложений более 10 высокопроизводительных ПЦОС или ЦП общего назначения. Пропускная способность шины ПЦОС TMS320C80 достигает 2,4 ГБайт/с – в потоке данных и 1,8 Гбайт/с в потоке команд.

ПЦОС TMS320C80 обеспечивает высокую степень гибкости и адаптивности системы, построенной на его базе, которая достигается за счет наличия на кристалле параллельно функционирующих ПЦОС процессоров и главного RISC-процессора. Входящие в состав ПЦОС TMS320C80 процессоры программируются независимо один от другого и могут выполнять как разные, так и одну общую задачу.

Обмен данными между процессорами осуществляется через общую внутрикристалльную память. Доступ к разделяемой внутрикристалльной памяти обеспечивает матричный коммутатор, выполняющий также функции контроллера при обращении к одному сегменту памяти несколькими процессорами.

Основные технические характеристики ПЦОС серии TMS320C8x:

- тактовая частота 40 или 50 МГц;
- производительность свыше 2 млрд операций в секунду;
- четыре 32-разрядных ADSP-процессора;
- 32-разрядный главный RISC-процессор с вычислителем с плавающей точкой;
- 50 Кбайт SRAM на кристалле (для TMS320C82 – 44 КБайт);
- 64-разрядный контроллер обмена с динамическим конфигурированием шины на обмен 64-, 32-, 16- и 8-разрядными словами;
- режим ПДП к 64-разрядному SRAM, DRAM, SDRAM, VRAM;
- 4 ГБайтный объем адресного пространства;
- видеоконтроллер;
- 4 внешних прерывания;
- встроенные средства внутрисхемной эмуляции;
- напряжение питания 3,3 В;
- около 4000000 транзисторов на кристалле;
- 0,5/0,6 КМОП-технология;
- 305-контактный корпус PGA.

Архитектура центрального процессора ПЦОС серии TMS320C8X

Центральный процессор (ЦП) – это вычислительное устройство с RISC-архитектурой и встроенным сопроцессором для выполнения операций с плавающей точкой. Подобно другим процессорам с RISC-архитектурой ЦП использует команды загрузки/сохранения для доступа к данным в памяти, а

также выполняет большинство целочисленных, битовых и логических команд над операндами в регистрах в течение одного такта.

Вычислитель с плавающей точкой (Floating-Point Unit) конвейеризирован и позволяет одновременно выполнять операции над данными как с одинарной, так и с двойной точностью. Производительность устройства составляет около 100 MFLOPS при внутренней тактовой частоте 50 МГц. FPU использует тот же регистровый файл, что и устройство целочисленной и логической обработки. Специальный механизм отметок (Scoreboard) фиксирует занятость регистров и обеспечивает их бесконфликтное использование.

Основными компонентами ЦП являются:

- регистровый файл, состоящий из 31 32-разрядного регистра;
- барабанное устройство сдвига (Barrel Rotator);
- генератор маски;
- таймер;
- целочисленное АЛУ;
- управляющий регистр;
- 4 аккумулятора с плавающей точкой двойной точности;
- умножитель с плавающей точкой;
- сумматор с плавающей точкой;
- контроллер кэш-памяти.

Архитектура ADSP-процессоров ПЦОС серии TMS320C8X

Архитектура ADSP-процессоров TMS320C80 ориентирована для применений, связанных с графикой и обработкой изображений (где использование нейропарадигм дает наибольший эффект). Она обеспечивает эффективное выполнение операций фильтрации и частотного преобразования, типичных для данных приложений. ADSP может выполнять за один такт одновременно операцию умножения, арифметико-логическую операцию (например, сдвиг-суммирование) и два обращения к памяти. Внутренний параллелизм ADSP позволяет обеспечить быстродействие свыше 500 млн операций в секунду на некоторых алгоритмах.

ADSP манипулирует 32-разрядными словами, а разрядность команд составляет 64 бита. ПЦОС использует прямую, непосредственную и 12 видов косвенной адресации.

Архитектура ADSP характеризуется следующими параметрами:

- 3-этапный конвейер;
- 44 доступных пользователю регистра (10 - адресных, 6 - индекса, 8 - данных, 20 - прочих);
- 32-разрядное 3-входовое АЛУ;
- репликатор битов;
- два адресных устройства;
- 32-разрядное устройство барабанного сдвига;

- генератор масок;
- блок условных операций для сокращения времени выполнения переходов.

Контроллер обмена управляет операциями обмена процессоров и памяти как внутри кристалла (через коммутатор), так и вне кристалла с использованием входящих в его состав интерфейсных схем, поддерживающих все распространенные стандарты памяти (DRAM, VRAM, SRAM) и обеспечивающих возможность динамического изменения разрядности шины от 8 до 64. Используя приоритетную дисциплину обслуживания запросов к памяти в режиме ПДП, контроллер обмена позволяет выполнить обмен данными, не прерывая вычислений со скоростью до 400 Мбайт/с. Контроллер обмена поддерживает линейную и координатную адресацию памяти для эффективного выполнения обмена при работе с 2- и 3-мерными графическими изображениями.

ПЦОС серии TMS320C4x

Многие нейрокомпьютеры проектируются на основе ПЦОС серии TMS320C4x. Благодаря своей уникальной структуре эти ПЦОС получили широкое распространение в мультипроцессорных системах и практически вытеснили ранее господствующие в этой области транспьютеры. ПЦОС TMS320C4x совместимы по системе команд с TMS320C3x, однако обладают большей производительностью и лучшими коммуникационными возможностями.

ЦП ПЦОС TMS320C4x имеет конвейерную регистро-ориентированную архитектуру. Компонентами ЦП являются:

- умножитель данных в формате с фиксированной и плавающей точкой;
- арифметико-логический модуль;
- 32-разрядное барабанное устройство сдвига;
- внутренние шины;
- дополнительные модули регистровой арифметики;
- регистровый файл ЦП.

Умножитель выполняет операции над 32-разрядными данными в формате с фиксированной точкой и 40-разрядными данными в формате с плавающей точкой, причем умножение производится за один такт (25 нс), независимо от типа данных и параллельно с обработкой данных в других функциональных блоках микропроцессора (например, АЛУ).

АЛУ выполняет за один такт операции над 32-разрядными целыми и логическими и 40-разрядными данными в формате с плавающей точкой, в том числе и операции преобразования форматов представления данных. ПЦОС аппаратно поддерживает операции деления и извлечения квадратного корня. Устройство барабанного сдвига за один такт выполняет сдвиг данных влево или вправо на число позиций от 1 до 32. Два дополнительных модуля регистровой арифметики (Address Generation 0 и Address Generation 1)

функционируют параллельно с умножителем и АЛУ и могут генерировать два адреса в одном такте. В ПЦОС поддерживается относительная базовая, базово-индексная, циклическая и бит-реверсная адресации.

Первичный регистровый файл представляет собой многовходовый файл из 32 регистров. Все регистры первичного регистрового файла могут использоваться умножителем, АЛУ и в качестве регистров общего назначения. Регистры имеют некоторые специальные функции; восемь дополнительных регистров могут использоваться для некоторых косвенных способов адресации, а также как целочисленные и логические регистры общего назначения. Остальные регистры обеспечивают функции системы такие, как адресация, управление стеком, прерывания, отображение статуса процессора, блочные повторы.

Регистры повышенной точности предназначены для хранения и обработки 32-разрядных целых чисел и 40-разрядных чисел с плавающей точкой. Дополнительные регистры доступны как для АЛУ, так и для двух модулей регистровой арифметики. Основная функция этих регистров – генерация 32-разрядных адресов. Они также могут использоваться как счетчик циклов или как регистры общего назначения.

Адресуемое пространство ПЦОС составляет 4Г 32-разрядных слов. На кристалле расположены два двухвходовых блока оперативной памяти RAM0 и RAM1, размером 4 Кбайт каждый, а также двухвходовой блок ROM, содержащий программу начальной загрузки.

Кэш команд процессора емкостью 128 32-разрядных слов содержит наиболее часто используемые участки кода, что позволяет сократить среднее время выборки команд. Высокая производительность ПЦОС TMS320C40 достигается благодаря внутреннему параллелизму процессов и многошинной организации процессора. Раздельные шины позволяют одновременно выполнять выборку команды, данных и прямой доступ в память.

4.4.5. ПЦОС компании Motorola

ПЦОС компании Motorola в меньшей степени, чем рассмотренные выше, используются для реализации нейропарадигм. Они подразделяются на семейства 16- и 24-разрядных ПЦОС с фиксированной точкой – DSP560xx, DSP561xx, DSP563xx, DSP566xx, DSP568xx и ПЦОС с плавающей точкой – DSP960xx. Линия 24-разрядных ПЦОС компании Motorola включает два семейства: DSP560xx и DSP563xx. Основные принципы, положенные в основу их архитектуры, были разработаны и воплощены в семействе DSP560xx. Дальнейшие работы по совершенствованию сигнальных процессоров проводились по трем направлениям:

- наращивание производительности 24-разрядных процессоров за счет конвейеризации функциональных модулей и повышения тактовой частоты;

- создание дешевых 16-разрядных микропроцессоров с расширенными средствами взаимодействия с периферией;
- разработка высокопроизводительных процессоров, включающих блок вычислений с плавающей точкой.

Компания Motorola является лидером по объему производства сигнальных микропроцессоров, большую часть которых составляют дешевые и достаточно высокопроизводительные 16- и 24-разрядные микропроцессоры с фиксированной точкой. Расширенные коммуникационные возможности, наличие достаточных объемов внутрикристалльной памяти для данных и программы, возможности защиты программы от несанкционированного доступа, поддержка режима энергосбережения делают эти микропроцессоры привлекательными для использования, в основном, в качестве специализированных вычислителей, контроллеров в промышленных роботах, бытовых электронных приборах, системах управления оружием, средствах беспроводной связи и др.

Примеры построения нейрокомпьютеров на основе ПЦОС компании Motorola довольно редки.

4.4.6. Нейропроцессоры, реализованные на основе ПЛИС

Отдельно следует рассмотреть возможность создания параллельных вычислителей (в том числе и нейропроцессоров) на базе ПЛИС (программируемых логических интегральных схем). На ПЛИС можно реализовывать распространенные в последнее время гибридные нейропроцессоры, когда блок обработки данных реализуется на ПЦОС, а логика управления на ПЛИС. В настоящее время множество фирм в мире занимается разработкой и выпуском различных ПЛИС, однако лидерство делят две фирмы Xilinx и ALTERA. Выделить продукцию какой-либо одной из этих фирм невозможно, так как по техническим характеристикам они различаются очень мало.

В настоящее время фирма ALTERA выпускает семь серий СБИС ПЛИС. Основные характеристики наиболее популярных из них приведены в табл. 4.7.

Компания Xilinx выпускает семь серий ПЛИС двух типов:

- FPGA – Field Programmable Gate Array;
- CPLD – Complex Programmable Logic Device.

Каждая серия содержит от одной до нескольких серий, в свою очередь состоящих из ряда кристаллов различной емкости, быстродействия и типов корпуса.

Таблица 4.7. Характеристики ПЛИС фирмы ALTERA

Характеристики	Семейства СБИС			
	MAX 7000E(S)	**MAX 9000**	**FLEX 8000A**	**FLEX 10K**
Архитектура	Матрицы И-ИЛИ	Матрицы И-ИЛИ	Таблицы перекодировки	Таблицы перекодировки
Логическая емкость	600-5000	6000-12000	2500-16000	10000-100000
Внутренняя память	Нет	Нет	Нет	6-24 Кбит
Число пользовательских выводов	36-164	60-216	68-208	59-406

Основные свойства и параметры ПЛИС Xilinx:

- значительный объем ресурсов – до 4 млн системных вентилей на кристалл;
- высокая производительность с системными частотами до 300 МГц;
- технологические нормы – до 0,18 мкм на шести слоях металла;
- высокая гибкость архитектуры с множеством системных особенностей: внутреннее распределенное и блочное ОЗУ, логика ускоренного переноса, внутренние буферы с третьим состоянием и т. д.;
- низкое энергопотребление;
- короткий цикл проектирования и быстрое время компиляции;
- развитые и недорогие средства проектирования;
- возможность перевода проектов в заказные схемы Xilinx.

При изготовлении ПЛИС фирмой Xilinx используются три основные технологии:

1. На основе SRAM (тип FPGA) – конфигурация ПЛИС хранится во внутреннем "теневом" ОЗУ, а инициализация осуществляется из внешнего массива памяти. По данной технологии выполнены серии: Spartan, Virtex, XC3000, XC4000, XC5200.
2. На основе FLASH (тип CPLD) – конфигурация сохраняется во внутренней энергонезависимой FLASH - памяти и в любой момент времени может быть перегружена непосредственно из ПК. По данной технологии выполнена серия XC9500.
3. На основе EEPROM (тип CPLD) – конфигурация сохраняется во внутренней энергонезависимой EEPROM - памяти и в любой момент времени может быть перегружена непосредственно из ПК. По данной технологии выполнена серия CoolRunner.

Таблица 4.8. Особенности реализации нейропроцессоров на ПЛИС

№	Тип ПЛИС	Производитель	Сложность кристалла, количество макроячеек (CLB)	Максимальное число нейронов
1	**XC4005E/XL**	XILINX	196 (14x14)	6
2	**XC4013XLA**	XILINX	576 (24x24)	18
3	**XC4020XLA**	XILINX	784 (28x28)	24
4	**XC4044XLA**	XILINX	1600 (40x40)	50
5	**XC4062XLA**	XILINX	2304 (42x42)	72
6	**XC4085XL**	XILINX	3136 (56x56)	97
7	**XC40250XV**	XILINX	8000	200
8	**EPF10K2**	ALTERA	144	4
9	**EPF10K50E**	ALTERA	360	11
10	**EPF10K100E**	ALTERA	624	19
11	**EPF10K250E**	ALTERA	1520	50
12	**M4LV-96/48**	AMD	966	3
13	**M4LV-192/96**	AMD	192	6
14	**M5LV-256**	AMD	256	8
15	**M5LV-512**	AMD	512	16

Реализация нейропроцессоров на основе ПЛИС требует участия эксперта на топологической стадии проектирования. Это обусловлено тем, что автоматизированный режим разводки пока не позволяет достигать 60-100% использования ресурсов кристалла по разводке, а это является принципиальным для сильносвязанных схем, к которым относятся и нейросетевые вычислители. Характеристики ПЛИС с точки зрения реализации нейросетевых парадигм представлены в табл. 4.8.

Построение нейрокомпьютеров на их основе хотя и дает высокую гибкость создаваемых структур, но пока еще проигрывает другим решениям по производительности.

5. НЕЙРОКОМПЬЮТЕРЫ

5.1. Основные понятия

Нейрокомпьютер – это вычислительная система, построенная на основе нейропроцессора (ов), использующая архитектуру, предусматривающую параллельные потоки однотипных команд и множественные потоки данных, и предназначенная для реализации нейросетевых алгоритмов в *реальном масштабе времени.*

Нейроэмулятор – система, построенная на базе каскадного соединения универсальных SISD-, SIMD- или MISD-процессоров (Intel, AMD, Sparc, Alpha, Power PC и др.) и реализующая типовые нейрооперации (взвешенное суммирование и нелинейное преобразование) на программном уровне.

Нейроускоритель – это нейрокомпьютер, реализованный в виде карты или модуля с распараллеливанием операций на аппаратном уровне. Нейроускорители могут быть построены на основе ПЦОС (Texas Instruments Inc., Analog Devices, Motorola), ПЛИС и (или) на основе специализированного (ых) нейрочипа (ов).

Нейрокомпьютеры, с точки зрения конструктивной реализации, можно подразделить на изделия, реализованные в виде карт и модулей, и конструктивно-автономные системы.

Нейрокомпьютеры, изготовленные в виде карт (*виртуальные нейрокомпьютеры*), как правило, предназначены для установки в слот расширения стандартного ПК. С другой стороны – нейрокомпьютеры в виде модулей соединяются с управляющей Host-ЭВМ по стандартному интерфейсу или шине [4].

5.2. Нейрокомпьютеры, выпускаемые в виде карт и модулей

Нейрокомпьютеры в виде карт и модулей представляют мощное средство, которое обеспечивает современному инженеру-исследователю и аналитику реализацию нейросетевых алгоритмов в реальном масштабе времени.

Если раньше большая часть времени уходила на подготовку и проверку одной-единственной гипотезы, то теперь, с использованием нейроускорителей, система обрабатывает данные и выдает заключение практически в реальном времени. Несмотря на свои впечатляющие возможности, такие нейрокомпьютеры не очень распространены на компьютерном рынке из-за высокой цены (от единиц до десятков тысяч

долларов для карт и от десятков до сотен тысяч долларов для модулей) и специфики и сложности освоения.

Все же основная причина отсутствия полнофункциональных нейрокомпьютеров на рынке – закрытость разработок. Из сотен фирм, производящих специализированные нейрокомпьютеры и нейропроцессоры, лишь единицы поставляют свою продукцию на массовый рынок. При этом, многие из них создают единичные продукты для спецприложений или обслуживают оборонный комплекс.

Рассмотрим некоторые возможности приобретения современных нейрокомпьютеров в РФ.

На рынке высокопараллельных вычислителей можно выделить следующие отечественные компании: АОЗТ "Инструментальные системы" и НТЦ "Модуль" [15], производящие несколько моделей нейрокомпьютеров, это также компании AUTEX Ltd. и L-Card Ltd. – производящие нейрокомпьютеры под заказ на базе микропроцессоров Analog Devices, и компания ScanTI-Rus [16] – представляющая высокопараллельные вычислители на базе ПЦОС фирмы Texas Instruments Inc.

5.2.1. Нейрокомпьютеры, реализованные на базе ПЦОС и ПЛИС

В основе построения нейрокомпьютеров данного типа лежит использование ПЦОС или ПЛИС, объединенных между собой в соответствии с архитектурой, которая обеспечивает параллельность выполнения вычислительных операций.

Как правило, такие нейрокомпьютеры строятся на основе гибкой модульной архитектуры, которая обеспечивает простоту конфигурации системы и наращиваемость вычислительной мощности путем увеличения числа процессорных модулей или применения более производительных ПЦОС (рис.5.1). Системы реализуются в основном на базе несущих модулей стандартов ISA, PCI, VME.

Основные функциональные элементы данных нейрокомпьютеров:

- модуль матричных ПЦОС,
- рабочая память,
- память программ,
- модуль обеспечения ввода/вывода сигналов (включающий АЦП, ЦАП и TTL линии),
- модуль управления, который может быть реализован на основе специализированного управляющего ПЦОС (УП), на основе ПЛИС или иметь распределенную структуру, при которой функции общего управления распределены между матричными ПЦОС.

Реализация нейрокомпьютеров и специализированных вычислителей с массовым параллелизмом на базе ПЦОС и ПЛИС является весьма эффективным решением задач цифровой обработки сигналов, обработки видео- и аудиоданных и построения технических систем управления.

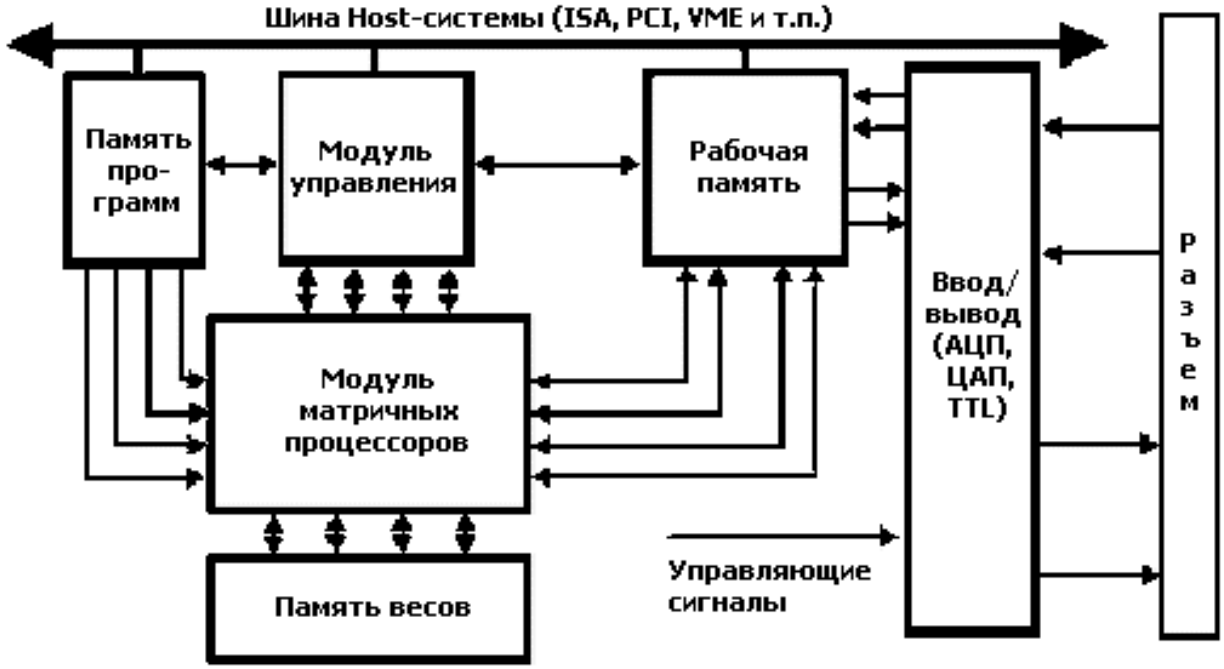


Рис. 5.1. Обобщенная функциональная схема нейрокомпьютера, реализованного на основе ПЦОС или (и) ПЛИС

При создании нейрокомпьютеров используется гибридная схема – блок матричных вычислений реализуется на базе кластерного соединения ПЦОС, а логика управления – на основе ПЛИС. В качестве элементной базы матричного кластера может использоваться ПЦОС ADSP21060 и TMS320C44, в ближайшее время им на смену придут ADSP2106x и TMS320C67xx. По оценке специалистов, в ближайшем будущем матричное ядро чаще будет реализовываться на базе нейропроцессоров, а ПЦОС и ПЛИС останутся основой для построения логики управления (например, Synapse 3).

Для построения нейрокомпьютеров данного типа наиболее перспективным является использование сигнальных процессоров с плавающей точкой ADSP2106x, TMS320C4x,8x, DSP96002 и др.

Типовая структурная схема реализации нейрокомпьютеров на основе ПЦОС ADSP2106x приведена на рис. 5.2. В ее состав включены один управляющий ПЦОС для осуществления функций общего управления, и до восьми процессоров осуществляющих параллельные вычисления согласно заложенным алгоритмам (матричные ПЦОС).

Управляющий и матричные процессоры образуют кластер процессоров с общей шиной и ресурсами разделяемой памяти. Обмен информацией между управляющим процессором, матричными процессорами, Host-ЭВМ и внешней средой осуществляется посредством

портов ввода/вывода. Для тестирования и отладки предназначен отладочный JTAG-порт. Так, в случае использования четырех матричных ПЦОС, обмен информацией между ними и УП осуществляется посредством четырех связанных портов ADSP2106x, по два связанных порта УП и модуля матричных ПЦОС выводятся на внешние разъемы для обеспечения связи с внешними устройствами. Имеется 12 внешних линков; по 3 линка каждого из матричных ПЦОС предназначены для внутримодульного межпроцессорного обмена.

Синхронизация работы системы может осуществляться как от внутренних кварцевых, так и от внешних генераторов. Активизация вычислений – программная или внешняя.

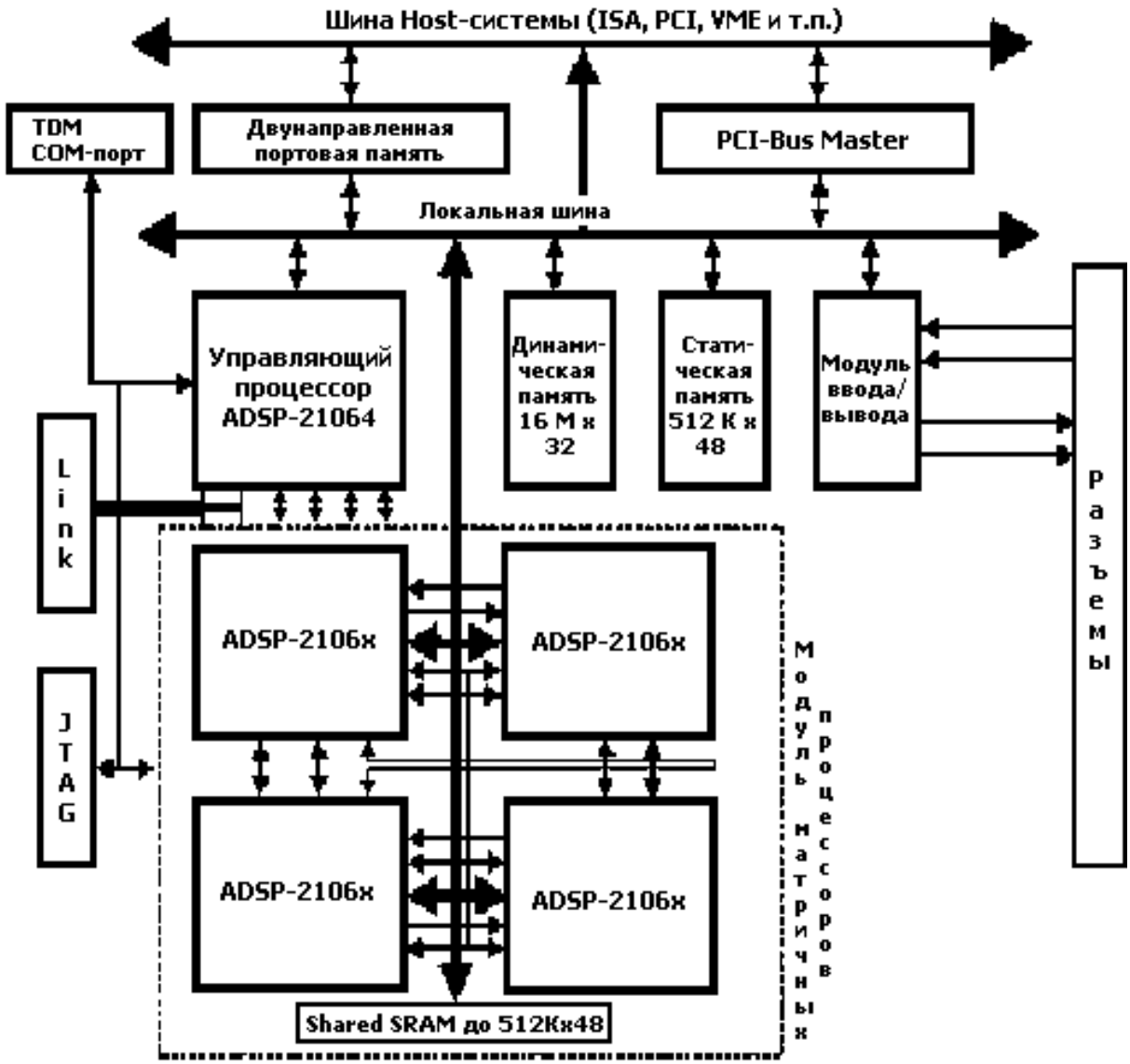


Рис. 5.2. Пример реализации нейрокомпьютера на основе ПЦОС ADSP2106x

Для ввода/вывода и АЦ/ЦА преобразований сигналов предназначен специализированный модуль, который включает: универсальный цифровой TTL порт, АЦП, ЦАП, узел программируемых напряжений для смещения шкал АПЦ и установки порога срабатывания стартовых компараторов, узел фильтрации выходных аналоговых сигналов, подсистему тестирования, узел синхронизации и управления, буферную память FIFO.

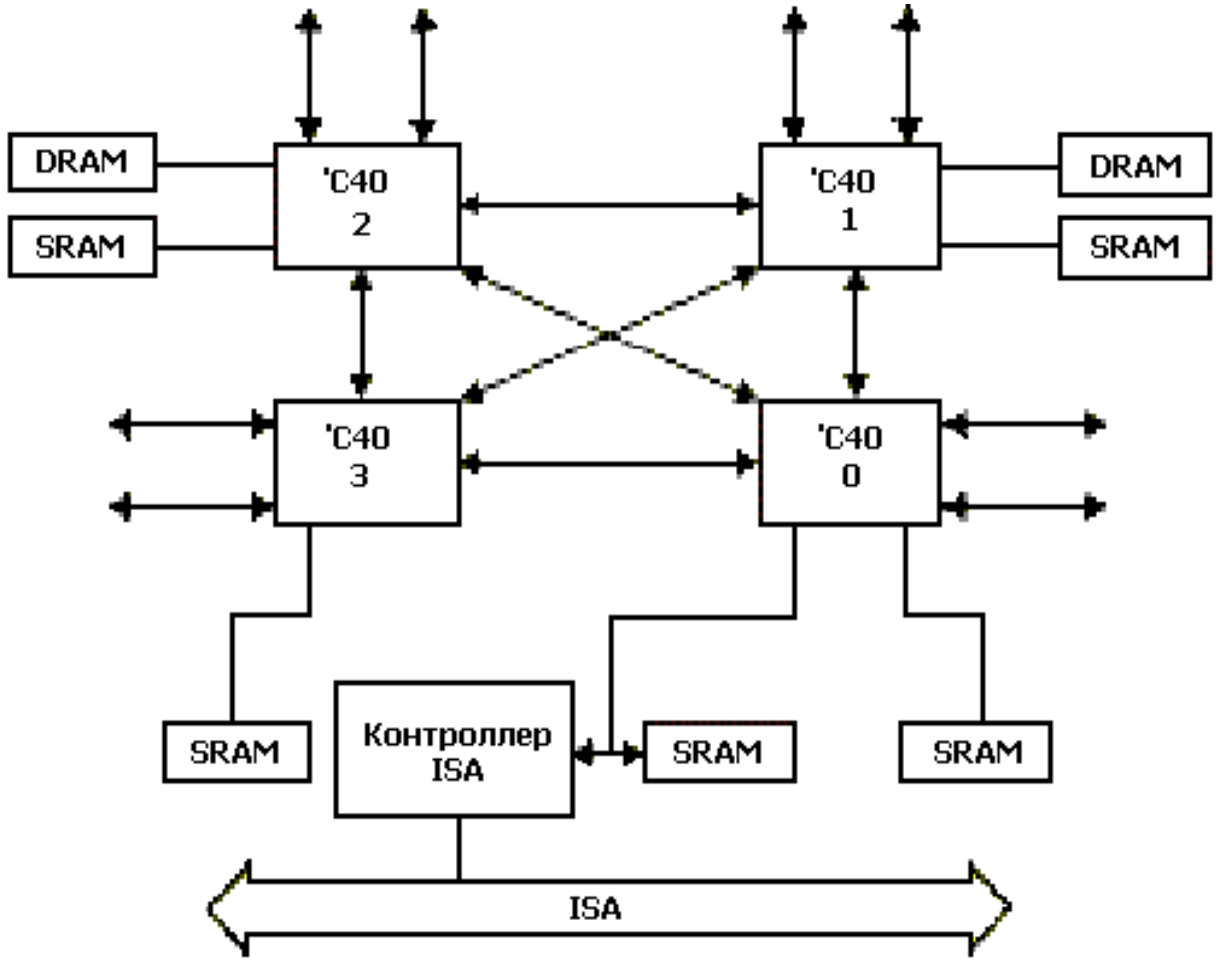


Рис. 5.3. Пример реализации нейрокомпьютера на основе ПЦОС TMS320C4x

Первоначальная загрузка осуществляется по Host-интерфейсу или по линкам. Управляющий интерфейс любого матричного ПЦОС позволяет управлять процессорным сбросом и прерываниями, его идентификационным номером и т.п.

Такая архитектура нейрокомпьютеров обеспечивает выполнение операций ЦОС в реальном времени, ускорение векторных вычислений,

возможность реализации нейросетевых алгоритмов с высоким параллелизмом выполнения векторных и матричных операций.

Структурная схема нейрокомпьютера на основе ПЦОС TMS320C4x представлена на рис. 5.3. Несколько ПЦОС, входящих в структуру нейрокомпьютера, образуют распределенную вычислительную структуру из процессорных модулей, соединенных между собой высокоскоростными портами. Данный вариант нейрокомпьютера может быть реализован с использованием ПЦОС в количестве от двух до восьми.

При использовании двух параллельных 32-разрядных ПЦОС TMS320C40 обмен информацией при реализации нейросетевых алгоритмов осуществляется с помощью шести связанных портов с пропускной способностью в 30 МБайт/с и каналов ПДП каждого из ПЦОС. Поддерживая параллельную независимую работу, подсистемы ПДП и ПЦОС обеспечивают параллельный обмен информацией со скоростями до 560 Мб/с. При помощи высокоскоростных портов возможна реализация таких архитектур, как кольца, иерархические деревья, гиперкуб и т.п. Каждая из локальных шин TMS320C40 обеспечивает обмен информации на скоростях до 120 Мбайт/с.

Процессорные модули функционируют независимо и при необходимости объединяются посредством связанных портов. Функции обмена, управления процессорными модулями, прерываниями и каналами DMA реализуют ПЛИС, например фирмы Xilinx. Применение в нейрокомпьютерах динамических реконфигурируемых структур (нейронной сети со структурной адаптацией) и использование современных ПЛИС семейств xC2xxx-xC4xxx (фирмы Xilinx) или аналогичных, требует минимизации времени на реконфигурацию ПЛИС, которые чаще всего программируются в режимах Master Serial и Peripherial. Основной недостаток при использовании данных режимов перепрограммирования заключается в зависимости процесса переконфигурации ПЛИС от встроенного тактового генератора. Минимальные потери времени можно получить при проведении переконфигурации ПЛИС в режиме Slave Serial, в котором внутренний тактовый генератор отключен, а синхронизация осуществляется посредством внешних синхросигналов. Реконфигуратор ПЛИС выполняется в виде специализированной микросхемы (например, XC2018-84pin-50MHz, XC3020-68pin-50MHz).

Построение нейрокомпьютеров на базе ПЛИС с одной стороны позволяет гибко реализовать различные нейросетевые парадигмы, а с другой – сопряжено с большими проблемами разводки всех необходимых соединений. Выпускаемые в настоящее время ПЛИС имеют различные функциональные возможности (с числом вентилей от 5 до 100 тысяч). Нейрокомпьютеры на базе ПЛИС, как правило, считаются гибкими, идеально приспособленными для научно-исследовательских целей и

мелкосерийного производства. Для построения более производительных и эффективных нейрокомпьютеров применение ПЦОС предпочтительнее.

Вопросам создания нейрокомпьютеров на ПЛИС посвящено большое число работ, ежегодно представляемых на конференциях и выставках «Нейрокомпьютеры и их применение», проходящих в Институте проблем управления (ИПУ РАН) им. В.А. Трапезникова.

Нейрокомпьютер ППК НИИ Системных исследований РАН

Рассмотрим в качестве примера удачной реализации нейрокомпьютеров, созданный в НИИ Системных исследований РАН параллельный перепрограммируемый компьютер (ППК).

Данный нейрокомпьютер разработан в стандарте VME и реализован на базе перепрограммируемых микросхем семейства 10К фирмы Altera. Предназначается для работы в качестве аппаратного ускорителя и является ведомым устройством на шине VME. Он должен включаться в систему как подчиненное устройство основной управляющей ЭВМ (host-машины) с универсальным процессором. Тактовая частота нейрокомпьютера 33 МГц.

Таблица 5.1. Оценка производительности ППК *

Название алгоритма	Pentium-100, с	PentiumII-333, с	Ultra SPARC, с	ППК, с
Свертка с ядром 4x4 3)	0,65	0,11	0,76	0,02
Медианный фильтр	1,97	0,49	0,75	0,001
Повышение контрастности	0,51	0,13	1,31	0,004
Прямое поточечное сравнение с маской 32x32 4)	43,78	7,14	58,89	0,142
Поиск локальных неоднородностей 32x32	0,120	0,028	0,146	0,032
Умножение матрицы на матрицу	8,61	0,60	12,31	0,011

*** - Оценки приведены для:**
Pentium-100 на частоте 100 МГц, объем ОЗУ 16 Мбайт;
Pentium-333 на частоте 350 МГц, объем ОЗУ 128 Мбайт;
UltraSPARC при частоте 200 МГц, объем ОЗУ 64 Мбайт;
ППК на частоте 33 МГц.

ППК используется для построения систем распознавания образов на основе обработки телевизионной, тепловизионной и другой информации, а также систем, основанных на реализации алгоритмов с пороговыми функциями и простейшими арифметическими операциями, и позволяет добиться при этом значительной скорости вычислений (табл. 5.1).

ППК состоит из следующих функциональных блоков (рис. 5.4):

- схема управления;
- базовые вычислительные элементы (БВЭ1 - БВЭ6);
- контроллер внешней шины (контроллер E-bus);
- контроллер системной шины (контроллер VME);
- два массива статической памяти (ОЗУ 0, ОЗУ 1);
- блок высокоскоростных приемников/передатчиков (блок ВПП).

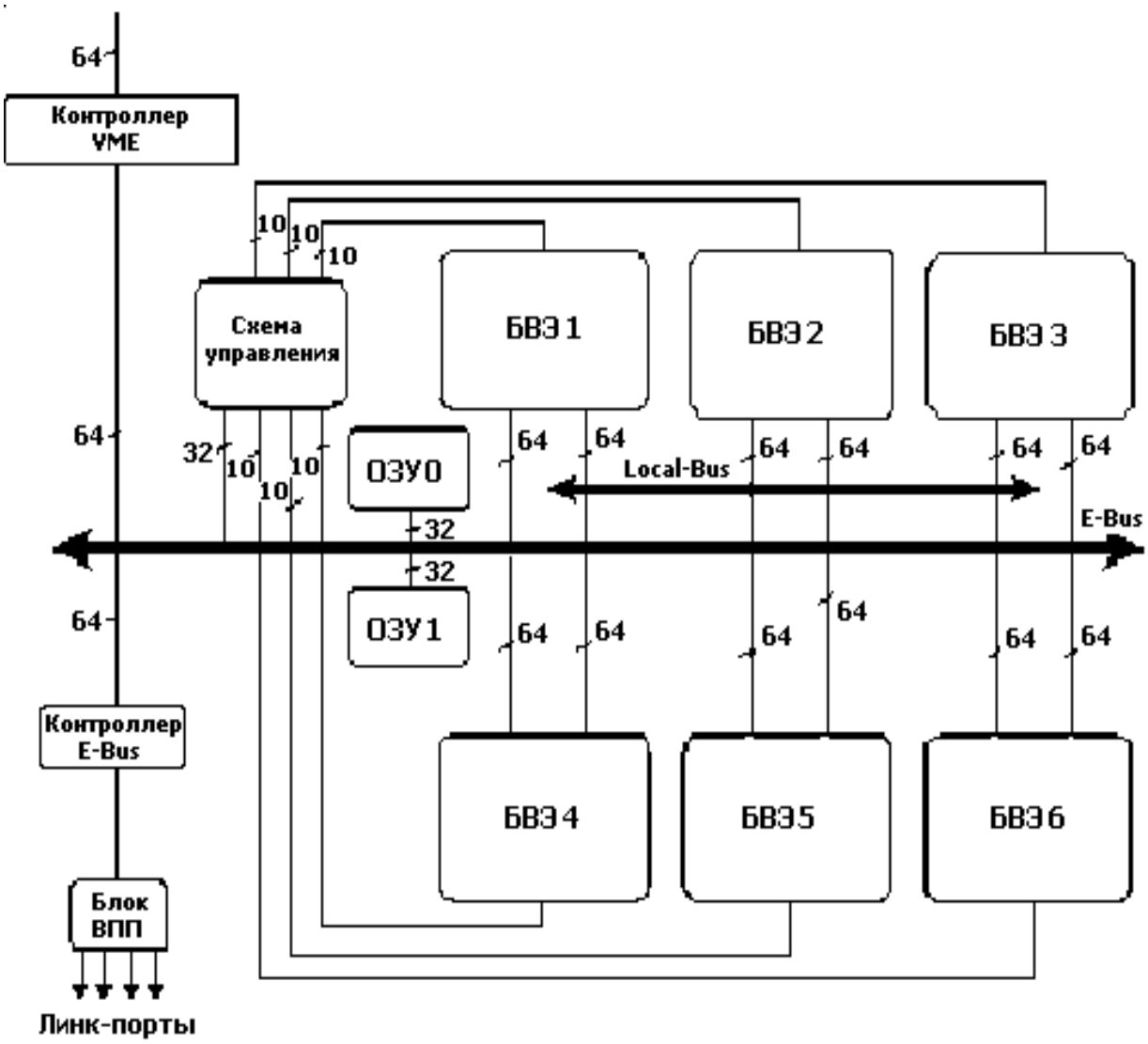


Рис. 5.4. Структурная схема ППК

Схема управления используется для управления БВЭ и потоками данных в вычислителе и представляет собой простейший RISC-процессор. Структура и набор команд процессора могут изменяться в зависимости от типа решаемой задачи.

БВЭ используются для выполнения простейших арифметических операций суммирования, вычитания, умножения и вычисления пороговых функций. Так как БВЭ реализованы на перепрограммируемых микросхемах, их архитектура может изменяться.

Архитектура БВЭ для различных алгоритмов может отличаться, но они легко реализуются путем комбинации библиотечных функций, компиляции их при помощи САПР (типа MaxPlus) и загрузки файла конфигурации в выбранный БВЭ.

Два массива локальной статической памяти, собранные из 8 микросхем статической памяти емкостью 0,5 Мбайт, имеют размер 4Мбайт и организованы как массив 512К 8-байтовых слов. Массивы памяти связаны со схемой управления отдельными адресными шинами и могут функционировать независимо друг от друга. Память предназначена для хранения общих коэффициентов, а также промежуточных или окончательных результатов вычислений, подготовленных к передаче через контроллер системной шины в центральный процессор или через контроллер E-bus на линк-порты. Связь нескольких вычислителей между собой или вычислителя с устройством оцифровки изображения при наличии у устройства оцифровки соответствующего интерфейса осуществляется посредством последовательного канала приемников/передатчиков HOTLink фирмы CYPRESS. Управление передачей данных выполняет контроллер внешней шины, который представляет собой набор четырех стандартных FIFO и регистров управления и данных. Контроллер шины VME выполняет функцию интерфейса с центральным процессором и является стандартным устройством. С точки зрения программиста вычислитель можно представить как RISC-процессор (схема управления или управляющий процессор) и шесть векторных процессоров (вычислительных элементов), отрабатывающих SIMD-команды (одна команда для многих данных). Большое количество шин данных, возможность одновременной работы всех БВЭ и выполнение арифметических операций умножения и сложения за один такт позволяет эффективно распараллеливать процесс обработки информации. Особенностью схемы управления перепрограммируемого вычислителя для систем обработки информации является наличие рабочей команды, управляющей шестью базовыми вычислительными элементами. Команда позволяет одновременно за один такт задавать различные режимы функционирования шести базовым вычислительным элементам и инкрементировать адреса обоих массивов памяти на любое число от 0 до 255, хранимое в регистрах инкремента, причем каждому массиву соответствует свой регистр. Команда может повторяться любое количество

раз в соответствии со значением, хранимым в специальном регистре. Это позволяет выполнять основную команду без потерь на организацию циклов и переходов. Рабочая команда позволяет одновременно запускать оба контроллера локальной памяти, инкрементировать адресные регистры на требуемое значение, выставлять на адресные шины адреса из соответствующих регистров адреса, выставлять на шины управления БВЭ команды из соответствующих регистров БВЭ. Кроме того, рабочая команда осуществляет организацию обмена данными между контроллером внешней шины и локальной памятью.

Внешний вид ППК показан на рис. П.4. Приложений.

Основные тенденции в проектировании нейрокомпьютеров на ПЛИС – это увеличение плотности компоновки нейропроцессоров за счет уменьшения площади соединений и функциональных узлов цифровых нейронов. Методика решения этой задачи применяется:

- в оптических связях для передачи информации между нейронами;
- для модификации программно-аппаратной реализации функциональных элементов нейровычислителей;
- для оптимизации представления промежуточных данных в слоях нейронов – нейронной сети со сжатой формой внутренних данных.

Рассмотренные варианты нейрокомпьютеров обеспечивают выполнение ЦОС и нейроалгоритмов в реальном масштабе времени, ускорение векторных и матричных вычислений, по сравнению с традиционными вычислительными средствами, в несколько раз и позволяют реализовывать нейронную сеть с числом синапсов до нескольких миллионов.

Еще больше повысить производительность нейрокомпьютеров данного типа можно при использовании одного из самых мощных на сегодня сигнальных процессоров – TMS320C80, TMS320C6xxx фирмы Texas Instruments.

Нейрокомпьютер Neuro-Turbo фирмы Fujitsu

Примером реализации нейрокомпьютеров на ПЦОС фирмы Motorola является нейрокомпьютер Neuro-Turbo фирмы Fujitsu. Он выпускается на основе четырех связанных кольцом 24-разрядных ПЦОС с плавающей точкой MB86220 (основные параметры: внутренняя точность 30 разрядов, машинный цикл 150 нс, память программ – 25 Кслов х 2 (внутренняя), 64 Кслов х 4 (внешняя), технология изготовления КМОП 1,2 мкм). Активационная функция нейронов ограничивается в диапазоне от 0 до 1, а возможные значения входов не превышают 16 разрядов, что обуславливает достаточную точность при 24-разрядной архитектуре. Построение нейрокомпьютера на основе кольцевой структуры объединения ПЦОС позволяет снизить аппаратные затраты на реализацию подсистемы централизованного арбитража межпроцессорного взаимодействия.

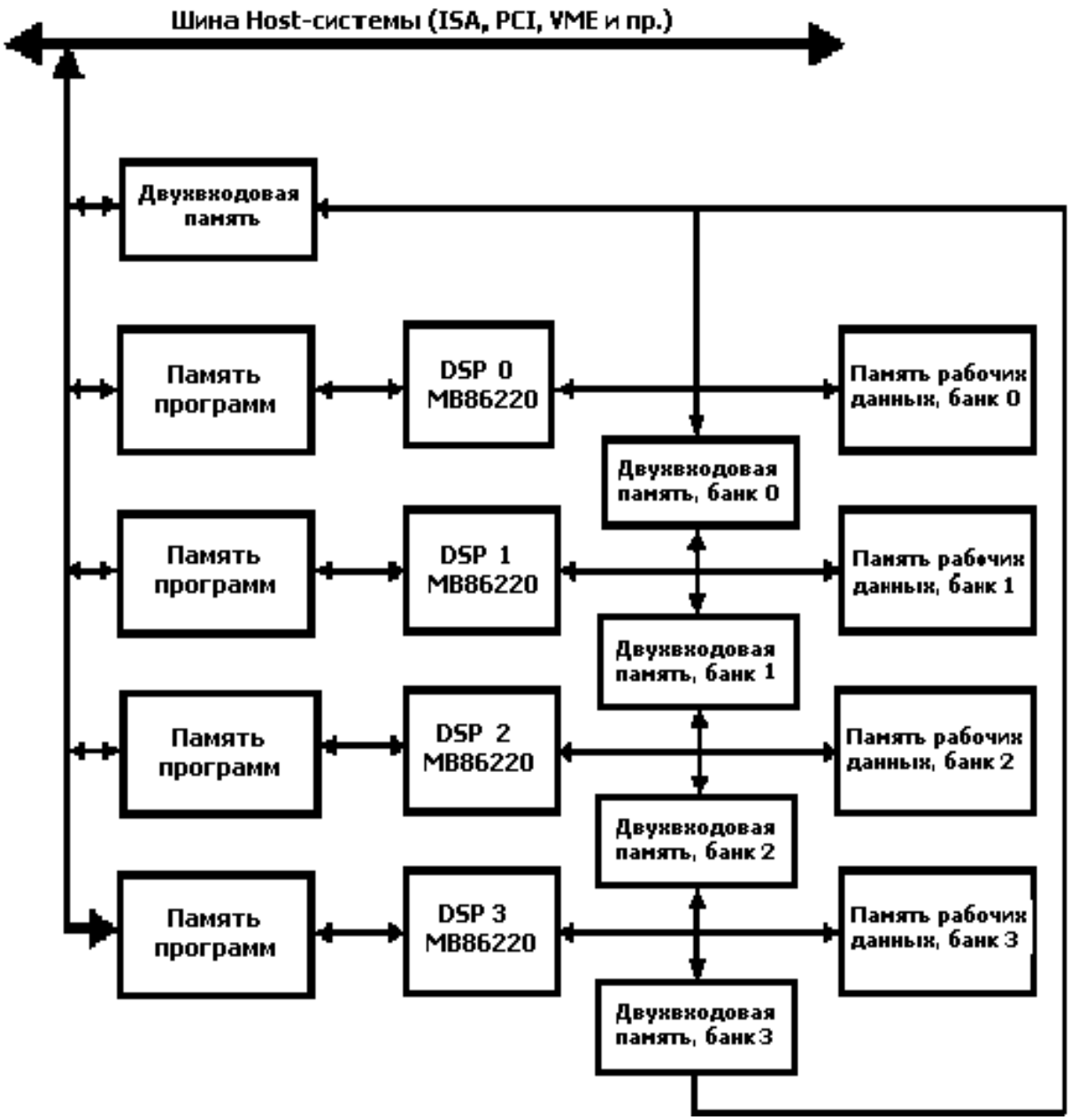


Рис. 5.5. Структура нейрокомпьютера Neuro-Turbo фирмы Fujitsu

Нейрокомпьютер Neuro-Turbo (рис. 5.5) состоит из четырех ПЦОС, связанных друг с другом двухпортовой памятью (ДПП). Каждый из ПЦОС может обращаться к двум модулям такой памяти (емкостью 2К слов каждая) и к рабочей памяти (емкостью 64К слов х 4 банка) в своем адресном пространстве. Вследствие того, что доступ к двухпортовой памяти осуществляется случайным образом одним из соседних ПЦОС, то передача данных между ними происходит в асинхронном режиме. Рабочая память используется для хранения весовых коэффициентов, данных и вспомогательной информации. Для успешной работы нейронной сети

необходимо получение сверток во всех элементарных нейронных узлах. Кольцевая структура объединения ПЦОС обеспечивает конвейерную архитектуру свертки, причем передача данных по конвейеру осуществляется посредством ДПП. После того как ПЦОС загружает данные из одной ДПП, он записывает результаты своей работы в смежную ДПП, следовательно, кольцевая архитектура параллельной обработки обеспечивает высокую скорость операции с использованием относительно простых аппаратных решений.

Для выполнения функций общего управления используется Host-ЭВМ на основе обычной вычислительной системы. Обмен данными между нейроплатой и Host-ЭВМ происходит через центральный модуль ДПП. Загрузка программ в ПЦОС осуществляется посредством памяти команд для каждого ПЦОС. Следовательно, архитектура нейрокомпьютера полностью соответствует параллельной распределенной архитектуре типа MIMD. Пиковая производительность системы 24 MFLOPS.

Для реализации модели нейронной сети иерархического типа фирмой Fujitsu выпущена нейроплата на основе ПЦОС MB86232 с собственной памятью до 4 Мб, что позволяет осуществлять моделирование нейронной сети, содержащей более 1000 нейронов. Структура нейронной сети включает в себя входной, промежуточный и выходной уровни (наибольшее число скрытых слоев – два (ограничение по памяти)). Для обучения нейрокомпьютера используются оригинальные алгоритмы:

- виртуального импеданса;
- скорректированного обучения;
- и расширенного обучения.

Каждая из рассмотренных типовых структур нейронной сети может моделироваться на основе приведенных выше вариантов построения мультипроцессорных нейрокомпьютеров. Так, для нейрокомпьютера на основе ПЦОС TMS320C4x при реализации какой-либо из рассмотренных схем (кольцо, иерархическое дерево, гиперкуб и т.п.) достаточно только изменить назначения коммуникационных портов чтобы обеспечить гибкость и масштабируемость при синтезе нейросетевых систем различной архитектуры.

Нейрокомпьютер ADP6701PCI компании Инструментальные системы

Карты данной серии ориентированы на применение в телекоммуникационных и навигационных системах, включая базовые станции, в системах медицинской диагностики, позиционирования, мультимедиа, где требуется сверхвысокая вычислительная мощность.

Структурная схема мультипроцессорной карты ADP6701PCI компании «Инструментальные системы» реализована на базе ПЦОС TMS320C6701 производительностью 1 GFLOPS. Внешний вид карты представлен на рис. П.5 Приложений.

Нейрокомпьютер имеет восемь параллельных вычислительных блоков и обеспечивает выполнение БПФ на 1024 отсчета за 109 мкс.

Отличительные особенности карты:

- буферная память тракта ввода 64 К × 32;
- буферная память тракта вывода 64 К × 16;
- двухпортовая статическая память 64 К × 32;
- синхронная динамическая память до 16 МБайт.

Благодаря гибкому аппаратному интерфейсу на ПЛИС, обеспечивается программная совместимость различных субмодулей.

Нейрокомпьютер DSP60V6 компании Инструментальные системы

DSP60V6 - высокопроизводительный мультипроцессорный модуль сбора и цифровой обработки сигналов, основанная на ПЦОС ADSP-21060/62 SHARC (рис. П.6 Приложений). Он позволяет выполнять программы SHARC во взаимодействии с устройствами, размещенными на дочерней плате ADM, в качестве которой могут использоваться модули АЦП и ЦАП. ADP60V5 устанавливается в промышленные крейты с размером плат 6U. Нейрокомпьютер может работать как автономно, так и с компьютером, имеющим шину VME. Программы ADSP-21060/62 и данные загружаются через шину VME и/или через пользовательские выводы разъема J2/P2 (X2). Через данные интерфейсы осуществляется сброс ПЦОС, просмотр памяти и инициирование выполнения программ.

Нейрокомпьютер построен на процессорном кластере из шести ПЦОС ADSP2106x компании Analog Devices производительностью 120 MFLOPS каждый. В процессорном кластере устанавливается до 1M x 48 бит оперативной статической памяти и до 16M x 32 бит оперативной динамической памяти. Кластер имеет в своем адресном пространстве VME интерфейс и FLASH память 4M x 8 бит. Нейрокомпьютер может работать независимо от шины VME – в этом случае прием и передача данных производится по 6 коммуникационным портам.

Нейрокомпьютер M1 компании Модуль

Модуль M1 выполнен на базе ПЦОС TMS320C40 компании Texas Instruments, связанных по высокоскоростным линкам. Имеется возможность каскадирования – подключения к модулю аналогичных плат [15].

Основные характеристики:

- ISA-интерфейс;
- до четырех TMS320C40 с частотой 50 МГц;
- пиковая производительность 100 MIPS, 200 MFLOPS, 1100 MOPS;
- объем SRAM 5 Мб (по 1 Мб на ПЦОС + 1 Мб разделяемый с ПК);
- время выборки 20 нс;
- объем DRAM – до 32 Мб;
- 8 внешних связей (скорость – 20 Мб/с).

Нейрокомпьютер М2 компании Модуль

Многопроцессорный модуль М2 для ЦОС выполнен на основе ПЦОС TMS320C40 фирмы Texas Instruments Inc. и представляет собой одноплатную многопроцессорную вычислительную систему.

Нейрокомпьютер предназначен как для автономной работы, так и для функционирования в составе ПК с системной шиной VME-bus, в том числе состоящей из нескольких таких же модулей. Конструктивно блок выполнен в соответствии с механическим стандартом на интерфейс VME-bus IEEE 1014 (6U).

Нейрокомпьютер М2 содержит:

- VME-bus контроллер;
- Master/Slave интерфейс;
- до шести TMS320C40 с частотой 50 МГц;
- до 2 Мб SRAM на процессор;
- до 64 Мб DRAM на плате;
- Flash-EPROM до 0,5 Мб;
- JTAG-интерфейс;
- RS-232-интерфейс.

Общая производительность – до 300 MFLOPS.

5.2.2. Нейрокомпьютеры, реализованные на базе нейрочипов

Наряду с нейроускорителями на базе ПЛИС и ПЦОС в последнее время широкое распространение получают нейроускорители на базе специализированных нейрочипов.

Проанализируем особенности их реализации на конкретных примерах.

Нейрокомпьютер МЦ4.01 (NM1) компании Модуль

Встраиваемый модуль МЦ4.01 (NM1) производства компании Модуль [15] предназначен для решения различных задач нейронными и нейроподобными алгоритмами, а также задач ЦОС и ускорения векторно-матричных вычислений. Модуль (рис. 5.6) выполнен на основе нейрочипа NeuroMatrixR NM6403 и представляет собой одноплатный нейроускоритель конструктивно выполненный в виде карты, вставляемой в стандартный слот шины PCI ПК. Он содержит:

- два нейрочипа NM6403;
- от 2 до 8 МБайт статической памяти (SRAM);
- 64 МБайт динамической памяти (EDO DRAM);
- четыре внешних COM-порта с темпом обмена 20 МБайт/с каждый.

Производительность:

- векторные операции – 1,9 млрд операций в секунду;
- скалярные операции – до 320 млн операций в секунду.

Конструктивное исполнение – стандарт PCI (версия 2-1) с темпом обмена да 132 МБайт/сек.

Нейроускоритель МЦ4.02 содержит один процессор NM6403, обладает производительностью от 40 до 11,500 ММАС и обеспечивает обработку данных переменной разрядности от 1 до 64 бит. Модуль предназначен для работы в составе комплекса с системной шиной PCI, блок статической памяти модуля доступен для записи и чтения как со стороны ПЦОС, так и со стороны шины PCI. На внешние разъемы модуля выведены два коммуникационных порта, аппаратно совестимых с портами TMS320C4x.

Соединение коммуникационных портов нескольких модулей позволяет создавать мультипроцессорные системы различной конфигурации.

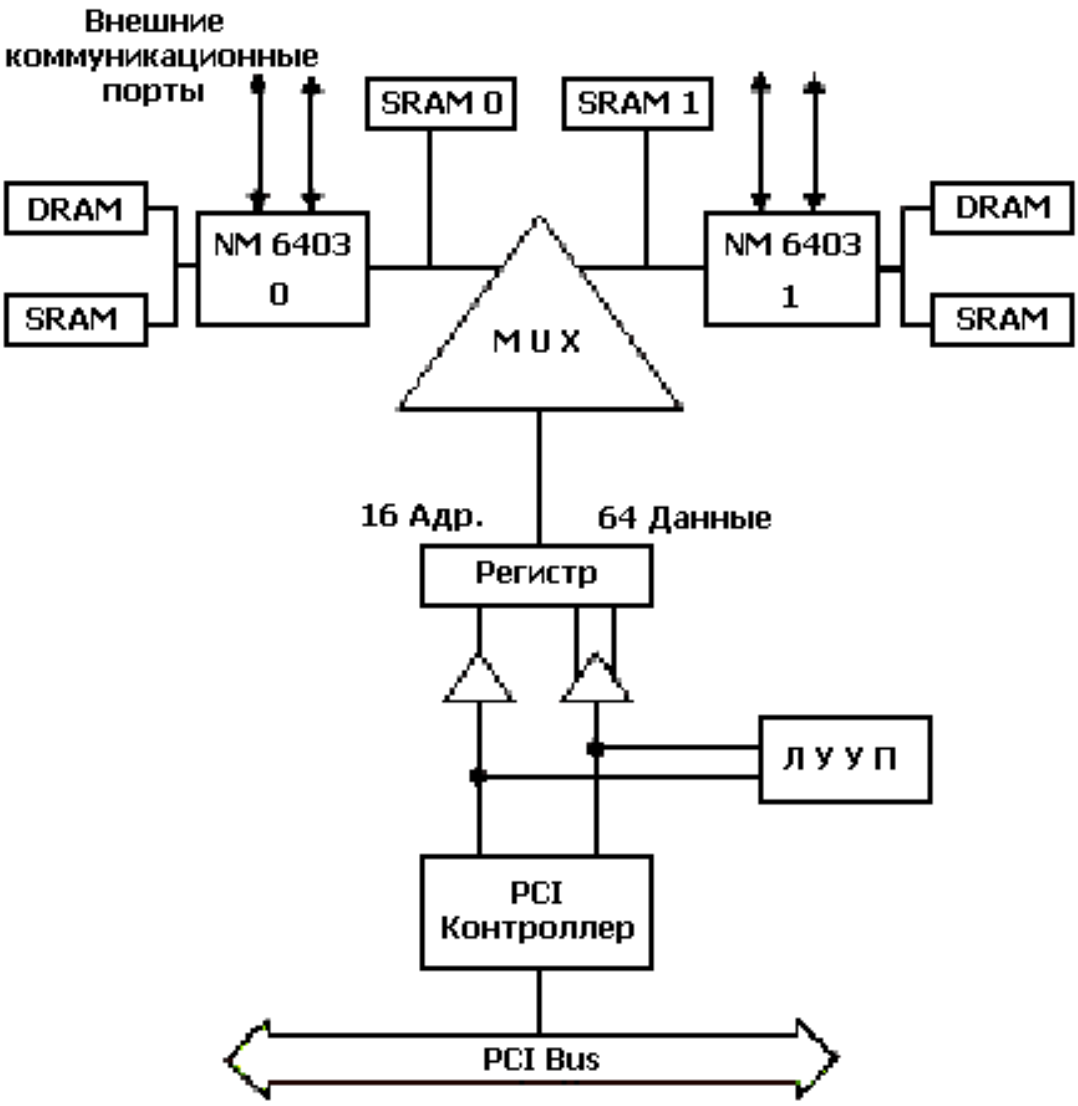


Рис. 5.6. Структурная схема модуля МЦ4.01

Нейрокомпьютер CNAPS/PC-128 компании Adaptive Solutions

Карта CNAPS/PC была выпущена в США в ноябре 1995 года и обладает рядом характеристик, которые на первый взгляд кажутся недостижимыми. При обучении на сложных данных нейрокомпьютер превосходит ПК на базе Pentium по быстродействию в несколько сотен (на отдельных тестах – в тысячу) раз. Нейрокомпьютер позволяет прогнозировать заведомо нереализуемые задачи – текущую ситуацию на мировом валютном рынке, динамику политических событий в регионах и даже исход футбольных матчей.

Старшая модель серии имеет пиковое быстродействие 2,27 млрд соединений/с, что позволяет сократить время аналитической обработки данных. Для сравнения – при решении нейросетевых задач стандартные ПК показывают следующее быстродействие:

- 486/ 50 MHz – 750 тыс. соед. /с;
- Pentium / 90 MHz – 1980 тыс. соед. /с.

Конструктивно нейрокомпьютер CNAPS/PC представляет собой полноразмерную карту, вставляемую в слот расширения PC (поддерживаются шины ISA и PCI). На плате размещены 2 либо 4 нейро-СБИС, реализующих 64 либо 128 нейропроцессоров соответственно. Кроме того, карта содержит 512 КБайт быстродействующей кэш-памяти и стандартный SIMM ОЗУ – 4 МБайт (расширяемый до 36 МБайт).

Нейрокомпьютер ZISC/ISA компании IBM

ZISC/ISA предназначен для IBM PC совместимых ПК. Нейроускоритель построен на 16 нейрочипах ZISC036 и имеет 576 нейронов. Возможна установка нескольких карт, одна из которых работает в режиме Master, а другие – Slave. Внешний вид ISA-нейроускорителя представлен на рис П.7 Приложений.

Нейрокомпьютер ZISC/PCI компании IBM

Данный высокопроизводительный нейроускоритель предназначен для работы в PCI слоте ПК (внешний вид представлен на рис. П.8 приложений).

Основные параметры:

- рабочая частота – 33 MHz;
- производительность – 165000 операций в секунду;
- ускоритель может содержать – 1,7,13 или 19 нейрочипов ZISC036.

Нейрокомпьютеры Synapse компании Siemens

Компания Siemens Nixdorf Informationssysteme (SNI) – дочернее предприятие концерна Siemens в сотрудничестве с Мангеймским университетом создала нейрокомпьютер под названием Synapse 1 (см. п. 5.3), который появился на рынке в середине 1994 г. В дальнейшем были выпущены нейроускорители Synapse 2 и Synapse 3.

Таким образом, компания Siemens стала первой европейской фирмой, выпустившей нейрокомпьютеры (в настоящее время они распространяются французской фирмой Tiga Technologies).

Сфера применения нейрокомпьютеров Siemens – распознавание речи, изображений, образов, ускорение работы программных эмуляторов.

Сложность моделирования на рабочей станции процесса самообучения для нейронных сетей до сих пор тормозит разработку нейронных применений, поскольку каждый шаг в обучении требует много времени. Что касается нейрокомпьютера Synapse, то за один час самообучения он достигает таких же результатов, что и нейронные сети в обычном компьютере за целый год. Эти системы обладают скалярной многопроцессорной архитектурой и наращиваемой памятью.

Нейрокомпьютер Synapse 2 компании Siemens

В состав Synapse 2 входит: один нейрочип MA16 (40 гц), управляющий ПЦОС TMS320C50 (55 МГц), модуль целочисленной обработки на базе ПЦОС TMS320C50 (55 мгц), память образцов (Y-Memory), память весов (W-Memory). Структурная схема нейроускорителя Synapse 2 представлена на рис. 5.7, а внешний вид на рис П.9 Приложений.

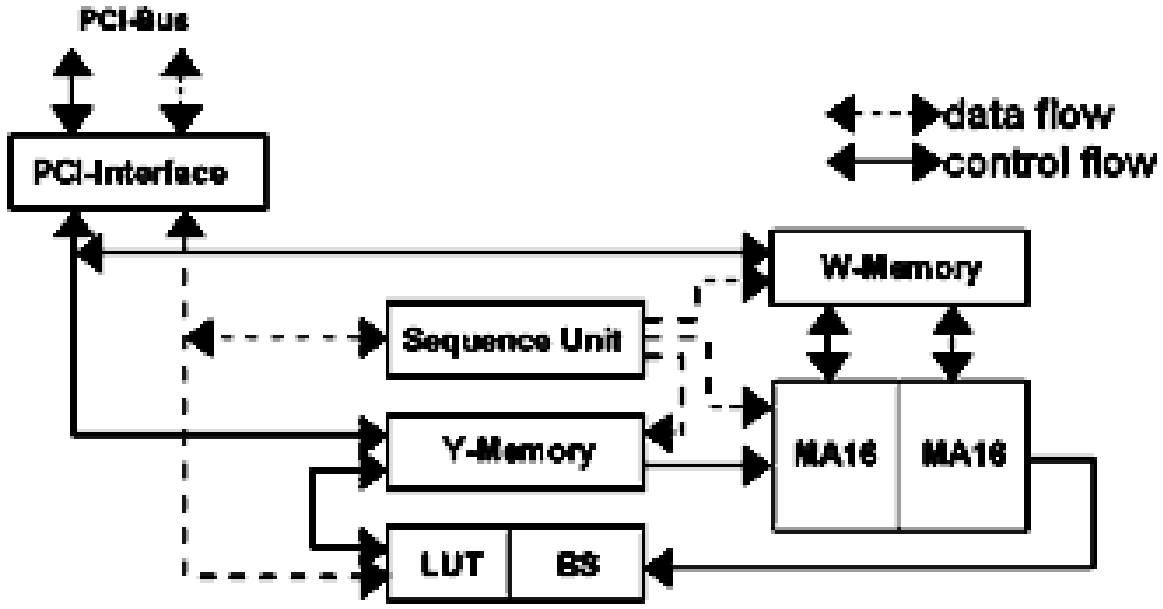


Рис. 5.7. Структурная схема Synapse 2

Нейрокомпьютер Synapse 3 компании Siemens

Серийно выпускаемые нейроускорители Synapse 3 поставляются с двумя нейрочипами MA 16. Пиковая эффективность одной карты Synapse 3 –

2,4 млрд операций/сек. Программное обеспечение работает в среде UNIX/XWIND и реализовано на C++. Нейронная сеть тоже описывается на C++ или может вводится интерактивно с помощью графического интерфейса типа OSF/Motif, что позволяет визуализировать конфигурацию СБИС после отображения на нее структуры нейронной сети.

Сравнительная диаграмма производительности нейрокомпьютеров Synapse 2 и Synapse 3, а также процессора Pentium-200 на матричных операциях приведена на рис. 5.8.

Рис. 5.8. Сравнительная диаграмма производительности

5.3. Нейрокомпьютеры, выпускаемые в виде конструктивно-автономных систем

Рассматривая подходы к аппаратной реализации нейрокомпьютеров необходимо отметить, что, несмотря на широкое распространение различных высокопараллельных ускорителей для различных задач, число моделей полнофункциональных нейрокомпьютеров невелико, а коммерчески доступны из них единицы. Это и понятно, так как большинство из них реализованы для спецприменений.

В качестве примеров рассмотрим ряд прототипов, реализованных в виде конструктивно-автономных нейрокомпьютеров [10]:

- нейрокомпьютер Synapse 1 компании Siemens;

- нейрокомпьютер Силиконовый мозг, созданный в США по программе "Электронный мозг" и предназначенный для обработки аэрокосмических изображений с производительностью 80 MFLOPS (80 × 1015 операций в секунду), с объемом данных, равным среднему объему информации, хранящейся в мозге человека;
- нейрокомпьютер Эмбрион (Россия).

Нейрокомпьютер Synapse 1 компании Siemens

Базовый комплект Synapse N110 предполагает наличие главной ЭВМ – рабочей станции Sun Sparcstation 5 модели TX1 в качестве вспомогательного консолидирующего устройства, облегчающего процессы программирования, проектирования нейронных сетей, тестирования, управления внешними устройствами, вывода результатов и т.п. Главная ЭВМ сопрягается с аппаратурой Synapse 1 через шину VME.

В архитектуре Synapse 1 можно выделить четыре основные компоненты: матричный процессор, память весов, устройство управления и устройство данных со следующими характеристиками:

- процессорная плата с матрицей из восьми ПЦОС MA 16 с производительностью 3,2 миллиарда операций умножения (16 × 16 бит) и сложения (48 бит) в секунду;
- память весов 128 МБайт;
- устройство управления на базе Motorola MC68040;
- устройство данных на базе Motorola MC68040;
- все аппаратные средства размещаются в небольшом корпусе 667 × 398 × 680 мм.

Проводимые исследования показали, что производительность выполнения нейросетевых операций на нейрокомпьютере Synapse 1, по крайней мере, на три порядка выше производительности традиционных вычислительных систем.

Нейрокомпьютер позволяет моделировать нейронные сети с количество синапсов до 64000000, а гибкость архитектуры практически не ограничивает разнообразность нейросетевых парадигм.

Нейрокомпьютер Эмбрион

Нейрокомпьютер Эмбрион разработан с участием члена-корреспондента МАИ В.Д. Цыганкова (рис. П.5 приложений). Было реализовано несколько модификаций данного нейрокомпьютера для различных приложений [10]:

- измерение случайных многомерных управляемых импульсных потоков Эмбрион-1;
- интерсенсорный перенос «глаз» - «рука»;
- техническая диагностика неисправностей энергогенератора самолетной электростанции (Эмбрион-2);

- управление нестационарным объектом в реальном масштабе времени (Эмбрион-3 и Эмбрион-4);
- реализация технического зрения (Эмбрион-5);
- управление адаптивным промышленным роботом Универсал-5А, обслуживающим карусельную плавильную печь стекольного завода;
- управление адаптивным промышленным роботом Р-2 с искусственными мышцами при сборке и покраске;
- управление мобильным автономным роботом Краб-1 при взаимодействии с неориентированными предметами и др.

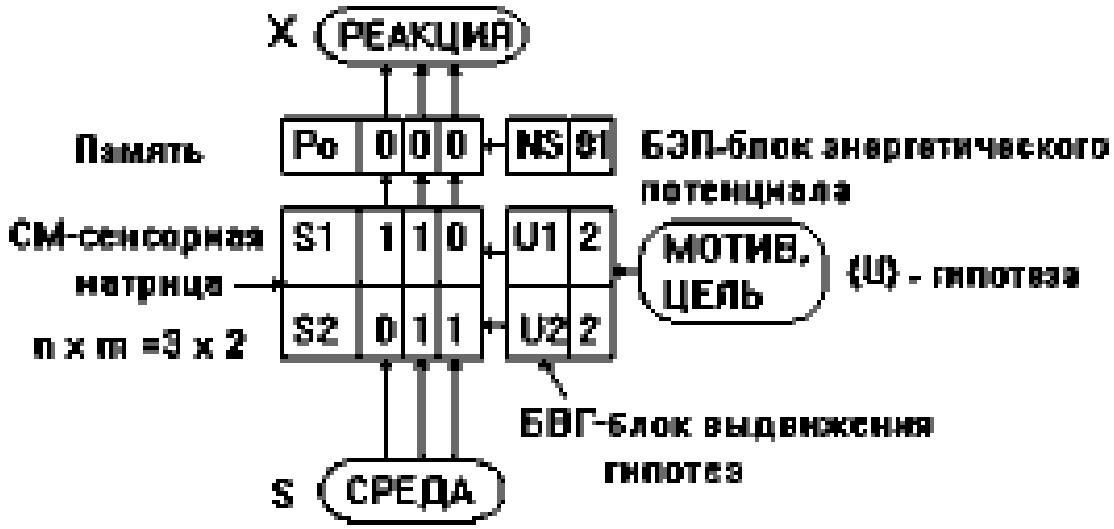


Рис. 5.9. Процесс генерации нейронной сети в нейрокомпьютере Эмбрион

На рис. 5.9 представлена структурная схема процесса генерации нейронной сети в нейрокомпьютере Эмбрион. Сигналы из внешней среды (S) проецируются на сенсорную матрицу. Под воздействием активирующего потока импульсов из Блока выдвижения гипотез информация из сенсорной матрицы переносится в регистр внутренней памяти, а его меняющиеся во времени коды-состояния формируют виртуальную квазинейронную сеть и ее выходную реакцию.

6. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Цель – предоставить обучаемому возможность самостоятельно решить несколько несложных, но весьма интересных задач, используя программный продукт для моделирования нейронных сетей TRAJAN компании Trajan Software Co. (Великобритания) или любую другую аналогичную программу на выбор (см. табл. П.1 приложения).

Лабораторный практикум отражает опыт подготовки российских и иностранных студентов в области применения нейронных сетей на кафедре Информационно-измерительной техники Московского энергетического института (технического университета) и на кафедре ИТ-7 Московской государственной академии приборостроения и информатики.

Позволяет закрепить теоретический материал и получить некоторые практические навыки в синтезе и обучении нейронных сетей.

Лабораторный практикум построен на одной из простых, и в то же время наиболее часто применяемой, модели многослойного персептрона и алгоритма обратного распространения.

Выполнение практикума допускает использование свободно-распространяемой демонстрационной версии пакета TRAJAN.

6.1. Лабораторная работа № 1
Создание и обучение простейшей нейронной сети

Цель – освоение основных приемов работы с демонстрационной версией программного продукта TRAJAN в ходе создания и обучения простейшей нейронной сети [1].

Задание

1. Повторить соответствующий теоретический материал (глава 2: п. 2.1 – 2.3 и глава 3).
2. Создать и обучить нейронную сеть, которая будет способна решать логическую задачу исключающего «ИЛИ». Таблица истинности для весьма полезной логической функции приведена в табл. 6.1.
3. Проверить работоспособность нейронной сети.
4. Ответить на вопросы для самопроверки № 1 – 4.

Создание нейронной сети

Новая нейронная сеть создается в Trajan с помощью окна ***Network Creating (Создание сети),*** которое доступно из меню ***File/New/Network*** или по нажатию соответствующей кнопки на панели инструментов.

После того, как на экране появится окно ***Network Creating*** для создания новой нейронной сети, следует произвести следующие действия.

Выбор типа нейронной сети

Демонстрационная версия TRAJAN предлагает два типа нейронных сетей. Для решения задач, представленных в данном лабораторном практикуме, рекомендуется использовать нейронную сеть типа ***многослойный персептрон,*** которая выбрана в данном окне по умолчанию.

Таблица 6.1. Таблица истинности для логической функции исключающего «ИЛИ»

Вход 1	Вход 2	Истина?
0	0	1
0	1	0
1	0	0
1	1	1

Определение количества слоев в нейронной сети и их размерностей

При задании количества слоев вашей нейронной сети, следует учитывать следующие особенности пакета TRAJAN:

1. Программный продукт поддерживает максимальный размер нейронные сети: 128 слоев по 128 нейронов в каждом, при этом первый слой всегда является входным и используется только для получения сетью исходных данных, а последний – выходным, и выходы его нейронов являются выходами всей сети в целом.
2. Для решения поставленной задачи рекомендуется использовать простейшую структуру нейронной сети, состоящую из трех слоев: входной слой с двумя нейронами, скрытый слой с двумя нейронами и выходной слой с одним нейроном (2-2-1).

Для задания количество нейронов в каждом слое используется матрица, представленная в окне ***Network Creating.*** Она выглядит как небольшая электронная таблица.

Необходимо определить количество нейронов в каждом слое сети с помощью первой ячейки этой матрицы, при этом любые слои с нулевым количеством нейронов будут проигнорированы.

После задания количества нейронов в каждом слое нейронной сети, TRAJAN самостоятельно определит количество слоев путем выбора из матрицы тех слоев, у которых количество нейронов отлично от нуля.

Примечание. Можно заметить, что матрица содержит строку для задания «ширины» каждого слоя. Строка редко используется в TRAJAN для карт Кохонена, хотя с помощью нее можно задавать и ширину слоев для нейронных сетей некоторых других типов.

Обучение нейронной сети

Алгоритм обратного распространения обучает нейронную сеть, используя доступные ему данные, которые хранятся в наборе представительских выборок для обучения. На каждой итерации (в терминах программного продукта TRAJAN – «эпохе»), нейронной сети предоставляется весь подготовленный набор обучающих пар. Выходы, получаемые нейронной сетью, сравниваются с желаемыми результатами. При этом ошибка нейронной сети вычисляется как разность между желаемыми и фактическими результатами и используется для регулирования весов нейронов в сети.

Для обучения нейронной сети необходимо:

- открыть окно ***Training Error Graph***, используя позицию меню ***Statistics/Training Graph***;
- открыть окно ***Back Propagation***, используя позицию меню ***Training/Backprop***;
- расположить на экране окна так, чтобы они были оба видны и не перекрывали друг друга;
- запустить алгоритм обучения путем нажатия на кнопку ***Train (Обучение)*** в окне ***Back Propagation***. При этом зависимость среднеквадратической ошибки обучения нейронной сети от числа используемых итераций будет вычерчиваться на графике в окне ***Training Error Graph***;
- увеличить число итераций в окне ***Back Propagation*** и обучить нейронную сеть вновь, нажав кнопку ***Train***.

Вначале моделирования при использовании небольшого числа итераций, среднеквадратическая ошибка уменьшается, но ненамного. Это обусловлено тем, что задача «исключающего «ИЛИ» для нейронной сети, как не парадоксально, гораздо сложнее в решении, чем многие более сложные задачи.

Окно ***Training Error Graph*** отображает общую ошибку обучения нейронной сети, однако иногда бывает полезно пронаблюдать за работой сети при использовании отдельно взятой обучающей пары.

Данный режим реализуется в TRAJAN с помощью окна ***Pattern Error.***

Запуск сети

После обучения нейронная сеть готова к запуску, причем запустить ее на выполнение можно несколькими способами.

Запуск, используя текущий набор представительских выборок

Нейронная сеть может быть запущена с предъявлением полного набора представительских выборок, использованных ранее при ее обучении, или выполняемыми наборами по одиночке. При этом необходимо воспользоваться пунктом меню ***Run/Single Pattern***, чтобы получить информацию о работе нейронной сети при предъявлении одной отдельно взятой представительской выборки или целого набора представительских выборок.

Запуск индивидуальной представительской выборки, не входящей в набор обучающих пар

При решении целого ряда задач необходимо проверять работу нейронной сети на представительской выборке, которая не входила в набор обучающих пар, использованных ранее при обучении. Например:

1. Прогнозирование появления новых данных с заранее неизвестными нейронной сети выходами. Если выходы заранее известны, то можно оценить качество работы подготовленной нейронной сети. В противном случае, результаты, полученные при запуске, могут быть использованы в качестве прогноза. Данный тип задач для нейронных сетей будет рассмотрен в лабораторной работе № 5.
2. Распознавание образов (задача будет рассмотрена в работе № 3). В этом случае оценивается чувствительность нейронной сети к небольшому изменению параметров исследуемого вектора, с помощью которого проводилось обучение.

Замечание

Нейронная сеть, подготовленная в данной лабораторной работе, была обучена с использованием всех возможных для нее обучающих пар, поэтому она может быть запущена на выполнение с использованием каждой из четырех представительских выборок. Следовательно, можно будет оценить работу нейронной сети на каждой из них.

При запуске, также возможно единовременное использование всего набора представительских выборок для оценивания общих параметров работы нейронной сети.

6.2. Лабораторная работа № 2
Определение направления двоичного сдвига

Цель – построение, обучение и тестирование нейронной сети, предназначенной для определения направления сдвига двоичного кода [1].

Задание

1. Повторить соответствующий теоретический материал (глава 2: п. 2.1 – 2.3 и глава 3).

2. Создать и обучить нейронную сеть, которая будет способна определять направление циклического сдвига четырехпозиционного двоичного кода.

3. Проверить работоспособность нейронной сети.

4. Ответить на вопросы для самопроверки № 5 – 8.

О применении нейронных сетей для решения задачи классификации

Типовая задача нейронных сетей – классификация того или иного исследуемого вектора (объекта). Получив в процессе обучения исходные данные об объекте, нейронная сеть определяет, к какому из множества классов принадлежат исследуемые векторы.

Проблема исключающего ИЛИ, рассмотренная в предыдущей лабораторной работе, является примером решения именно такой задачи. Если исследуемый вектор может принадлежать только к одному из двух классов, то задача называется двухклассной. Задача, поставленная в данной работе, сводится к двухклассной.

Простейший путь решения задачи двухклассной классификации при помощи нейронных сетей – формирование у сети единственного выхода, который получает значение 1 для одного класса и 0 – для другого. Значения, лежащие внутри данного диапазона, характеризуют степень принадлежности объекта к тому или иному классу.

Действительно, на том или ином выходе многоуровневого персептрона практически невозможно получить значения равные точно 0 или 1, хотя к этим значениям иногда можно подойти довольно близко.

Таким образом, для решения двухклассных задач с использованием одного выхода необходимо задаваться *уровнем доверия*, например: если значения выхода выше 0,95 – считать, что объект (исследуемый вектор) принадлежит к одному классу, а если ниже 0,05 – к другому.

Решение задачи классификации в TRAJAN

Нейронная сеть с предъявлением единственной представительской выборки запускается в окне ***Run Single Pattern (Запустить единственный образец)*** или в окне ***Run One-off Pattern (Запустить одиночный образец не входящий в представительскую выборку)***.

TRAJAN сравнивает выходную величину сети с пределами доверия и определяет:

- если выход выше установленного верхнего порога, исследуемый вектор (объект) классифицируется положительно;
- если выход ниже установленного нижнего порога, то сообщается о негативной классификации;

- если значение выхода находится между порогами, то сообщается о том, что исследуемый вектор (объект) классифицировать не удалось.

Общая статистика результатов классификации осуществляется при нажатии кнопки ***Run*** в окне ***Statistic/Classification*****,** которое открывается из меню ***Statistic/Classification*****.** Статистика в этом окне отображается в виде матрицы, содержащей один столбец для каждого класса. Каждый столбец содержит две секции: «Общая статистика» и «Статистика процесса классификации», разделенные широкой горизонтальной чертой.

Секция «Общая статистика» содержит следующую информацию:

- ***Total (Всего)*** – количество образцов данного класса в наборе.
- ***Correct (Правильные)*** – количество образцов данного класса правильно классифицированных нейронной сетью.
- ***Wrong (Неправильные)*** – количество образцов неправильно классифицированных сетью (как принадлежащих к другому классу).
- ***Unknown (Неизвестные)*** – количество образцов данного класса, которые нейронная сеть не смогла классифицировать.

Секция «Статистика процесса классификации» показывает, сколько исследуемых векторов (представительских выборок) было отнесено к каждому классу. При этом неклассифицированные векторы в данной секции не отображаются.

Нейронная сеть для определения направления двоичного сдвига

Для решения поставленной задачи, следует построить и обучить нейронную сеть, которая должна будет определять направление двоичного сдвига.

Операция двоичного сдвига является типичной для многих языков программирования. Сущность ее заключается в том, что число представляется в двоичном коде, а затем с полученной последовательностью производится операция циклического сдвига вправо или влево. Если производится сдвиг влево, у числа самая первая (левая) цифра переставляется в конец, а если сдвиг производится вправо, то последняя (правая) цифра переставляется в начало. Для построения нейронной сети представим в четырехпозиционном двоичном коде числа от 0 до 15. Далее, следует определить количество входов и выходов нейронной сети, необходимой для решения поставленной задачи. Очевидно, что для определения направления сдвига на входы нейронной сети необходимо представить исходную четырехпозиционную двоичную последовательность и четырехпозиционную двоичную последовательность, которая получилась в результате сдвига. Выходной слой нейронной сети может состоять из одного нейрона. Его значение будет равно 0, если сдвиг произведен влево, и 1 – если сдвиг произведен вправо. Таким образом, для решения данной двухклассной задачи необходима нейронная сеть с *восемью* входами и *одним* выходом.

Таблица 6.2. Результаты сдвига влево и вправо четырехпозиционного кода

Число	Двоичный код	Сдвиг влево	Сдвиг вправо
1	0001	0010	1000
2	0010	0100	0001
3	0011	0110	1001
4	0100	1000	0010
5	0101	1010	1010
6	0110	1100	0011
7	0111	1110	1011
8	1000	0001	0100
9	1001	0011	1100

Обучение нейронной сети

Определив количество входов в сети, приступим к созданию набора обучающих пар для обучения сети. Для этого выберем шесть четырехпозиционных двоичных кодов и выполним с ними операции сдвига вправо и влево (табл. 6.2). Следует заметить, что в качестве представительских выборок нельзя выбирать числа с двоичным представлением **0000, 1111, 1010, 0101,** поскольку в независимости от направления сдвига (влево или вправо) для этих чисел будет получен один и тот же результат.

После подготовки набора представительских выборок (обучающих пар) и обучения, следует протестировать получившуюся нейронную сеть (проверить качество ее обучения). Тестирование проводится на оставшихся трех четырехпозиционных двоичных кодах табл. 6.2, которые не вошли в набор представительских выборок, использованный при обучении.

6.3. Лабораторная работа № 3
Распознавание символов

Цель – разработать и исследовать нейронную сеть обратного распространения, предназначенную для распознавания образов [1].

Задание

1. Повторить соответствующий теоретический материал (п. 1.3).

2. Построить и обучить нейронную сеть, которая могла бы решать задачу распознавания символов.

3. Произвести тестирование нейронной сети при добавлении шума.

Описание работы

На качество решения поставленной задачи в сильной степени влияют ограничения, которые накладываются производителями на демонстрационные версии своих программных продуктов. Так, в демо-версии программного продукта TRAJAN количество нейронов в слое не может превышать 9, поэтому при распознавании символов будем оперировать матрицей 3 × 3.

Определение структуры нейронной сети

Представим в виде матрицы 3 × 3 четыре латинские буквы *X, Y, L, I* и обучим нейронную сеть распознавать их матричное представление (см. табл. 6.3).

Таблица 6.3. Матричное представление для букв *X, Y, I* и *L*

X			***Y***			***I***			***L***		
1	0	1	1	0	1	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0	1	0	0
1	0	1	0	1	0	0	1	0	1	1	1

В соответствии с табл. 6.3 входной сигнал для нейронной сети может быть представлен в виде развернутого растра – вектора длинной 9. Например, для буквы *X* это

101010101

Теперь определимся с выходами нейронной сети. Очевидно, что для распознавания образов нейронная сеть должна иметь возможность формировать столько выходных сигналов, сколько образов она должна уметь распознавать.

В нашем случае таких образов четыре, поэтому возможны два варианта представления выходных данных нейронной сети:

- выходной слой с двумя нейронами (выходами), т.е. каждому символу ставится в соответствие двухпозиционный двоичный код;
- выходной слой с четырьмя нейронами (выходами), т.е. каждому символу свой выход.

Предлагается выбрать любой вариант.

Обучение нейронной сети

Набор обучающих пар, используемых для обучения нейронной сети, составляется с учетом того, какой вариант формирования выходного слоя выбран в предыдущем разделе. Если выбран вариант с двумя выходами – каждой букве ставится в соответствие двухпозиционный двоичный код, то выходной слой выглядит следующим образом:

***X* – 00** ***Y* – 01** ***I* – 10** ***L* – 11**

Если выбран вариант с четырьмя выходами, то выходной слой такой:

***X* – 0001** ***Y* – 0010** **I – 0100** **L – 1000**

После того, как набор представительских выборок (обучающих пар) создан, необходимо обучить нейронную сеть и проверить, насколько корректно она решает поставленную задачу.

Проверка работы нейронной сети

После качественного обучения нейронной сети, следует внести в исходные данные некоторый шум (хотя это сделать непросто, так как в матрицу 3 × 3 очень трудно добавлять шум).

Например, вместо растра буквы ***I*** –

010010010

попробуйте подать

010110010

и посмотреть: удастся ли нейронной сети распознать символ, несмотря на внесенные в данные шум.

6.4. Лабораторная работа № 4 Искусственный нос

Цель – разработать и исследовать нейронную сеть обратного распространения для искусственного носа, предназначенного для химического анализа воздушной среды [11].

Задание

1. Повторить соответствующий теоретический материал (гл. 1: п. 1.4; гл. 2: п. 2.1 – 2.3; и гл. 3: п. 3.2).

2. Исследовать и проанализировать имеющиеся экспериментальные данные (табл. 6.4), и определить количество вводов и выводов, требуемых для полносвязанной нейронной сети обратного распространения.

3. Создать и обучить нейронную сеть, которая будет способна указывать наличие определенных примесей в воздухе при анализе показаний химических датчиков.

4. Обучить нейронную сеть, расшив количество представительских выборок (обучающих пар), применяемых для обучения нейронной сети (табл. 6.5).

5. Определить оптимальную структуру нейронной сети с точки зрения минимизации среднеквадратической ошибки обучения.

6. Обучить нейронную сеть, изменив параметры алгоритма обратного распространения.

7. Для пп. 3, 4, 6 построить графические зависимости среднеквадратической ошибки обучения от количества нейронов, используемых в скрытых слоях, и от количества итераций, используемых для обучения.

8. Сравнить результаты обучения в пп. 3, 4, 6.

9. Ответить на вопросы лабораторной работы.

Определение количества вводов и выводов нейронной сети

Сформируйте представительские выборки (обучающие пары) для проведения обучения.

Для этого используйте следующие рекомендации.

1. Количество входов нейронной сети должно соответствовать количеству химических датчиков (рис. 6.1).

2. Существует два основных метода кодировки выхода нейронной сети. Первый – это использование бинарного вектора: для каждой примеси только один единственный выход принимает значение 1. В этом случае количество выходов равно количеству примесей, определяемых системой. В другом случае все номинальные добавки пронумерованы и их числа перенесены в бинарную систему.

В этом лабораторном исследовании следует использовать первый метод.

3. Исходные данные:

- начальные экспериментальные данные, в виде показаний химических сенсоров, представлены в табл. 6.4;
- рекомендуемое число входов сети 11;
- рекомендуемое число выходов 6;
- рекомендуемое начальное число нейронов скрытого (внутреннего) слоя 4.

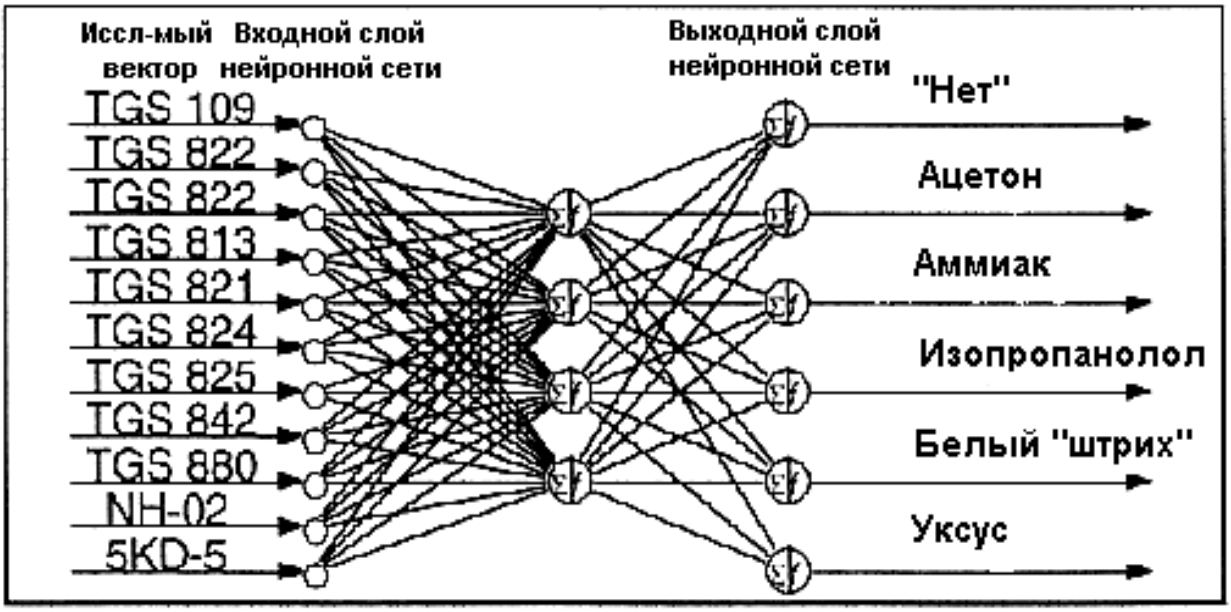


Рис. 6.1. Топология нейронной сети прототипа искусственного носа

Таблица 6.4. Первичная обучающая пара

Исследуемый вектор, составленный по показаниям сенсоров и формирующий вход нейронной сети	**Выходы нейронной сети**
{1; 0,05; 0,1; 0,3; 0,07; 0,08; 0,2; 0,05; 0,2; 0,6; 0,8}	{0, 0, 0, 0, 0, 1}

Таблица 6.5. Набор вторичных обучающих пар

Исследуемые векторы, составленные по показаниям сенсоров и формирующие вход нейронной сети	**Выходной вектор нейронной сети**
{1; 0,05; 0,1; 0,3; 0,07; 0,08; 0,2; 0,05; 0,2; 0,6; 0,8}	«Нет»
{0,8; 0,4; 0,7; 0,6; 0,1; 0,5; 1,0; 0,75; 0,5; 0,7; 0,8}	Ацетон
{0,9; 0,2; 0,4; 0,5; 0,1; 0,7; 0,6; 0,5; 0,5; 0,7; 0,8}	Аммиак
{0,85; 0,7; 0,8; 0,65; 0,1; 0,4; 1,0; 0,7; 0,4; 0,6; 0,7}	Изопропанол
{0,9; 0,3; 0,3; 0,4; 0,04; 0,1; 0,5; 0,3; 0,2; 0,7; 0,8}	Белый «штрих»
{0,95; 0,18; 0,21; 0,3; 0,05; 0,1; 0,3; 0,2; 0,2; 0,5; 0,7}	Уксус

4. Вторичные экспериментальные данные, в виде показаний химических сенсоров, представлены в табл. 6.5: набор представительских выборок
включает шесть обучающих пар, которые, естественно, подготовлены в формате первой обучающей пары (табл. 6.4).

Определение топологии нейронной сети

Практические исследования показывают, что одного внутреннего (скрытого) слоя достаточно.

Обучение нейронной сети

Следует проводить при помощи алгоритма обратного распространения. Используйте любой доступный программный продукт для моделирования нейронных сетей (см. табл. П.1 приложений), который поддерживает данный алгоритм обучения. Как результат, получите надежную и быстродействующую нейронную сеть искусственного носа, применяемого для контроля атмосферы в воздушной среде промышленных, офисных и домашних помещениях и в других задачах.

Вопросы для проверки

1. Назовите основное свойство многослойных нейронных сетей прямого распространения.
2. Какие существуют модификации алгоритма обратного распространения?
3. Назовите распространенные направления применения искусственного носа.

6.5. Лабораторная работа № 5
Прогнозирование

Цель – разработать и исследовать нейронную сеть обратного распространения, предназначенную для прогнозирования временных серий, а также для анализа качества генератора случайных чисел [11].

Задание

1. Повторить соответствующий теоретический материал (гл. 1: п. 1.5; гл. 2: п. 2.1 – 2.3 и гл. 3).
2. Создать и обучить нейронную сеть, предназначенную для анализа временных серий заданной размерности и отражающую структуру данных серий.
3. Осуществить прогноз значений будущих элементов временных серий.

Таблица 6.5. Числа в диапазоне от 0 до 10, полученные с использованием генератора случайных чисел Турбо-паскаль

Номера чисел	Числа, полученные с использованием функции Random(10) Турбо-паскаль									
1 – 10	0	0	8	2	2	6	3	1	3	4
11 – 20	0	4	0	8	0	2	9	3	7	3
21 – 30	6	8	7	3	1	3	4	2	8	2
31 – 40	4	1	8	2	7	9	4	8	8	0
41 – 50	1	1	5	0	5	0	7	6	7	7
51 – 60	5	2	6	5	9	6	9	2	6	2
61 – 70	0	7	4	8	5	5	9	6	3	0
71 – 80	7	9	7	7	1	1	9	7	5	8
81 – 90	6	2	0	6	2	8	1	2	5	9
91 – 100	1	1	2	1	5	4	2	1	6	7

Исследуемые временные серии

Объектом исследования являются временные серии, полученные с помощью генератора случайных чисел, формирующего равномерно распределенные числовые значения.

За основу рекомендуется взять сто элементов (чисел) временной серии, значения которых лежат в диапазоне от 0 до 9 и сто элементов – в диапазоне от 0 до 99. Данные временные серии следует получить с помощью функции генерации случайных чисел любого программного продукта для математического моделирования (MathCAD, Matlab или др.), среды программирования на языке высокого уровня (Delphi, C Builder или др.) или взять из табл. 6.5 и 6.6.

Выясним, есть ли какая-либо закономерность в появлении элементов данных временных серий и тем самым определим качество генератора случайных чисел, который по идее должен обладать свойством некоррелированности значений числовых последовательностей.

Таблица 6.6. Числа в диапазоне от 0 до 100, полученные с использованием генератора случайных чисел Турбо-паскаль

Номера чисел	Числа, полученные с использованием функции Random(100) Турбо-паскаль									
1 – 10	0	3	86	20	27	67	31	16	37	42
11 – 20	8	47	7	84	5	29	91	36	77	32
21 – 30	69	84	71	30	16	32	46	24	82	27
31 – 40	48	14	87	28	77	97	49	88	82	2
41 – 50	14	14	50	2	59	0	77	65	77	70
51 – 60	55	20	68	59	95	64	99	24	67	29
61 – 70	8	77	49	88	50	57	95	68	33	0
71 – 80	70	98	77	74	19	14	91	78	58	86
81 – 90	68	28	9	62	28	87	16	27	54	96
91 – 100	17	15	26	17	57	49	28	15	60	73

Определение начальной структуры нейронной сети

Чтобы синтезировать оптимальную структуру, необходимо подготовить и обучить несколько нейронных сетей и проверить качество выполнения требуемых операций. Ниже приведем основные этапы синтеза такой структуры (п. 1.5.1).

1. В качестве нейронной сети предлагается воспользоваться сетью типа многослойный персептрон.
2. Количество входов соответствует количеству выбираемых элементов из временных серий (в нашем случае – N).
3. Число нейронов во внутренних (скрытых) слоях и число таких слоев зависит от сложности задачи анализа или прогнозирования временных серий.
4. Выходной слой нейронной сети следует составить из одного нейрона, значение которого будет соответствовать прогнозируемому элементу временных серий.

Обучение нейронной сети

Для обучения нейронной сети подготовим блок обучающих выборок следующим образом.

Выберем кадр из N+1 числа элементов, идущих от конца к началу временной серии, где первые N элементов – формируют вектор входного слоя, а последний (N+1)-й – элемент выходного слоя нейронной сети. Следующую обучающую выборку получаем, передвигаясь окном на один элемент влево и т.д.

Таким образом, имея временную серию, состоящую из 100 элементов, можно подготовить 100–N–1 обучающих пар.

Оптимизация структуры нейронной сети

Чтобы оценить качество прогнозирования, получите в соответствии с разделом «Исследуемые временные серии» еще 3 элемента, следующих за 100 элементами временной серии, которые уже использовались при обучении. Далее, запустите нейронную сеть, предъявив ей 3 новых исследуемых вектора, и сравните результаты нейронной сети с числами, синтезированными генератором.

Разницу между значениями, синтезированными генератором, и результатами работы нейронной сети используйте как качественный показатель сети. Уменьшите эту разницу, модифицируя структуру нейронной сети [5].

Замечания

Подход, используемый в данной лабораторной работе для проверки качества генератора случайных чисел, наверное, не самый популярный и эффективный.

В то же время, разработанная в лабораторной работе нейронная сеть является аналогичной нейронным сетям, применяемым в интеллектуальных системах прогнозирования различного назначения, например систем прогноза знаков изменения биржевых индексов, систем прогноза цены, систем расчета оптимального использования ресурсов и т.п (см. п.1.5).

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Для решения каких задач применяются нейронные сети?
2. Искусственный нейрон и его основные свойства.
3. Виды активационных функций.
4. Какое основное отличие искусственных нейронов, которые используются для построения нейронных сетей, получивших название персептронов?
5. К какому типу алгоритмов обучения относится алгоритм обратного распространения, и в чем отличительная черта этих алгоритмов.
6. Дайте свое определение «Многослойному персептрону».
7. В чем заключается задача классификации?
8. Почему, по Вашему мнению, наиболее распространенной топологией сети является модель прямого распространения информации?
9. Каков смысл параметра α – скорости обучения (или, по-другому, коэффициента корреляции)?
10. В чем заключается явление перетренированности сети?
11. Что такое обобщение?
12. Для решения каких задач применяются самоорганизующиеся карты Кохонена?
13. Каков смысл параметров, передаваемых функции обучения Кохонена?
14. Что означают термины «компонентная карта» и «карта расстояний»?
15. Объясните построение бинарной сети Хопфилда. Какими свойствами обладает ее весовая матрица?
16. Объясните способ функционирования сети Хопфилда. Какие стадии работы она имеет?
17. Одно из важнейших свойств сетей Хопфилда состоит в восстановлении корректных образов. Для каких приложений оно может быть использовано?
18. Опишите поведение одного нейрона сети Хопфилда.
19. В случае сетей Хопфилда также возникает проблема локальных минимумов. Поясните различие локальных минимумов в сетях Хопфилда и сетях, использующих алгоритм обратного распространения ошибок (Backpropagation).
20. Какие входные векторы используют ART1-сети?
21. Какую работу осуществляют слои сравнения и распознавания ART-сети?
22. В чем заключается основной принцип решения задачи распознавания образов?
23. В чем разница между нейропроцессорами и нейрокомпьютерами?
24. Приведите классификацию нейрокомпьютеров по типу применяемых процессоров и конструктивной реализации.

ЛИТЕРАТУРА

1. **Алексеев А.В., Круг П.Г., Петров О.М.** Нейросетевые и нейрокомпьютерные технологии: Методические указания к проведению лабораторных работ по курсу «Информационные технологии». М.: МГАПИ, 1999 г.
2. **Галушкин А.И.** Некоторые исторические аспекты развития элементной базы вычислительных систем с массовым параллелизмом (80- и 90- годы) // Нейрокомпьютер. № 1. 2000. С. 68-82.
3. **Горбань А., Россиев Д.** Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
4. **Кирсанов Э.Ю.** Цифровые нейрокомпьютеры: Архитектура и схемотехника / Под ред. А.И.Галушкина. Казань: Казанский Госуниверситет, 1995. 131 с.
5. **Круг П.Г., Филатенков Ю.В., Шилин А.В.** Оптимизация структуры нейронной сети, применяемой для автоматизированной классификации результатов моделирования. IV Всероссийская научн.-техн. конф. «Новые информационные технологии». М.: МГАПИ, 2001. С. 114-118.
6. **Минский М., Пейперт С.** Персептроны. М.: Мир, 1971.
7. **Розенблатт Ф.** Принципы нейродинамики. Персептроны и теория механизмов мозга. М.: Мир, 1965.
8. **Уоссермен Ф.** Нейрокомпьютерная техника. М.: Мир, 1992.
9. **Хехт-Нильсен Р.** Нейрокомпьютинг: история, состояние, перспективы // Открытые системы. № 4. 1998.
10. **Шахнов В.А., Власов А.И., Кузнецов А.С., Поляков Ю.А.** Нейрокомпьютеры: архитектура и схемотехника. М.: Изд-во Машиностроение, 2000. 64 с.
11. **Alekseev A., Krug P., Shahidur R.** The Neural Networks. Teaching Edition. Moscow. Publishing House of MPEI. 2000. 64 pp.
12. SNNS. User Manual. (http://www.informatik.uni-stutgart.de/ipvr/bv/projekte/snns)
13. http://neurnews.iu4.bmstu.ru
14. http://neuronets.chat.ru
15. http://www.module.ru
16. http://www.scanti.ru
17. http://www.ti.com
18. http://www.trajan-software.demon.co.uk
19. http://www.basegroup.ru/tasks/forecast.htm

ПРИЛОЖЕНИЯ

Таблица П.1. Программные продукты моделирования нейронных сетей

Название	Разработчик/ производитель	Платформа	Поддерживаемые парадигмы и алгоритмы обучения	Интерфейс	Цена	Комментарии
1	2	3	4	5	6	7
Matlab Neural Network Toolbox 3.0 http://www.mathworks.com	MathWorks, США	Win 95, 98, NT 4.0	Поддерживаемые парадигмы: персептрон, обратное распространение, радиальный базис, сети Эльмана, сети Хопфильда, вероятностная и обобщенная регрессия. Неподдерживаемые парадигмы: Хебб, Кохонена, карты свойств, самоорганизующиеся карты	GUI для Ms Windows	----	Генерирует ANSI-совместимый код
SNNS http://www.informatik.uni-stutgart.de/ipvr/bv/projekte/snns/announce.html	Институт параллельных и распределенных систем (IPVR) при Штуттгардском университете	Unix	Обратное распространение, радиальный базис, ART1, ART2, карты Кохонена, сети Джордана, сети Эльмана, ассоциативная память	GUI для X-Windows	Бесплатно	Один из лучших симуляторов. Может работать с MS Windows при использовании эмулятора X-Windows. С программой поставляются исходные коды на C++
Trajan http://www.trajan-software.demon.co.uk/commerce.htm	Trajan Software Ltd., Великобритания	Win 3.x, 9x, NT	Многоуровневый персептрон, Обратное распространение, радиальный базис, карты Кохонена, вероятностная и обобщенная регрессия	GUI для MS Windows	620$	Демо-версия программы доступна на сайте компании-производителя

Продолжение табл. П.1

1	2	3	4	5	6	7
Delta	Artificial Intelligence Group, Франция (Департамент компьютерных наук)	HP-UX Sun OS 4.1	Многоуровневый персептрон, обратное распространение, сети Джордана, сети Ельцина, карты Кохонена	GUI для X-Windows	Беспла тно	Демо-версия
http://www-inf.enst.fr/~milc/dnns/dnns.us.html						
X-Sim	IIC, Испания (Мадрид)	Unix Linux	Многослойный персептрон, обратное распространение, карты Кохонена	Режим командной строки для ОС типа Unix, GUI для X-Windows	Беспла тно	Демо-версия
http://www.iic.uam.es/xsim/Welcome.html						
Brain Wave	Университет Куинсланд, США	Любая	Многослойный персептрон, обратное распространение, сети Хеббиана, карты Кохонена	Internet-броузер с поддержкой Java	Беспла тно	Реализован в виде Java-апплета - может работать с любой операционной системы
http://www2.psy.uq.edu.au/~brainwav/						
VieNet2	Австрийский институт исследования проблем искусственного интеллекта	DOS Win Unix Linux	Многослойный персептрон, обратное распространение, сети Джордана, сети Эльмана, карты Кохонена, Ассоциативная память	Формируетс я пользовател ем	Беспла тно	Распространяется в форме исходных кодов, что позволяет активно использовать его для написания собственных программ
www.ai.univie.ac.at/oefai/nn/tool.html						
NeuroWindows	НейроПроект, Россия	Win	Обратное распространение, ассоциативная память, карты Кохонена	Формируетс я пользовател ем	450$	Библиотека динамической компоновки Visual Basic, C++ и Delphi
http://www.neuroproject.ru/						

Продолжение табл. П.1

1	2	3	4	5	6	7
Aspirin/ MIGRAINES	Mitre Corp.	Unix	Обратное распространение	GUI для X-Windows	Беспла тно	Сохраняет веса и вектор узлов нейронной сети на диске в доступном формате
Atree	Билл Армстронг, Университет г. Альберта, США	Dos, Unix	Адаптивные логические деревья	Режим командной строки для Unix-подобных ОС, окна для DOS	Беспла тно	Демо-версия
alnl@cs.ualberta.ca						
Cnaps	Adaptive Solutions Inc.	SunOS	Обратное распространение, карты Кохонена (одномерные и двумерные), LVQ2 и частотно-чувствительное конкурентное обучение	GUI для X-Windows	68,75 $	Производительность при обучении по алгоритму обратного распространения 1 биллион CUPS
ICSIM	Международный институт компьютерных наук, Беркли, Калифорния, США	Unix	Предопределенные сети	Shell, GUI для X-Windows	Беспла тно	Демо-версия
Neural Shell	Лаборатория SPANN, Департамент инженерной энергетики, Университет Огайо, США	Unix	Сети Хопфильда, сети Хемминга, обратное распространение, карты Кохонена, адаптивное медленное обратное распространение, частотно-чувствительное конкурентное обучение	Режим командной строки, GUI для X-Windows и SUNTOOLS	Беспла тно	Демо-версия
ftp://ftp.quanta.eng.ohio-state.edu/						
Neuron	Университет Дьюка, США	Unix	Трехмерная реконструированная пирамидальная ячейка, диффузия	GUI для X-Windows	Беспла тно	Демо-версия

Окончание табл. П.1

1	2	3	4	5	6	7
Sankom	Дортмундский университет, Германия	Unix	Карты Кохонена	Режим командной стоки Shell	Беспла тно	Демо-версия
SOMPAK	SOM, Лаборатория компьютерных и информационных наук, Хельсинкский университет технологий	Unix, DOS	Самоорганизующиеся карты	Формируетс я пользовател ем	Беспла тно	Демо-версия
Xerion	Университет Торонто, Департамент компьютерных наук, США	Unix	Обратное распространение, рекуррентное обратное распространение, машина Больцмана, теория среднего поля, манипуляция свободной энергией, жесткое и мягкое конкурентное обучение, карты Кохонена	GUI для X-Windows	Беспла тно	Демо-версия
ftp://ftp.cs.toronto.edu/pub/xerion						
NETS	COSMIC, Университет Джорджии, США	DOS, UNIX	Обратное распространение	Режим командной строки	------	Демо-версия
service@cossack.cosmic.uga.edu						

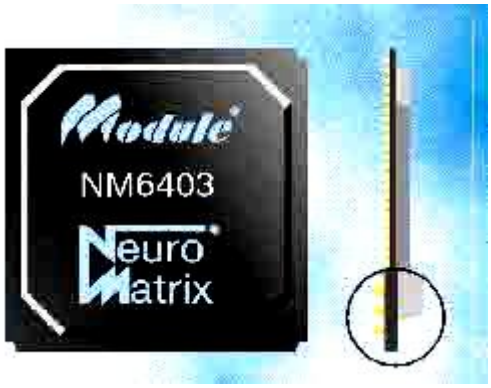


Рис. П.1. Нейрочип NeuroMatrix NM6403 компании Модуль

Рис. П.2. Нейрочип NeuroMatrixR NM6404 компании Модуль

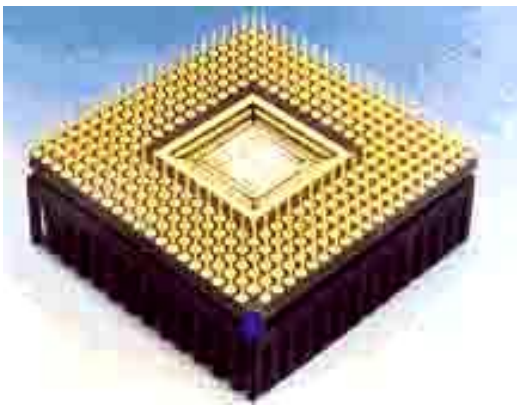


Рис. П.3. Корпус нейрочипа MA16 компании Siemens

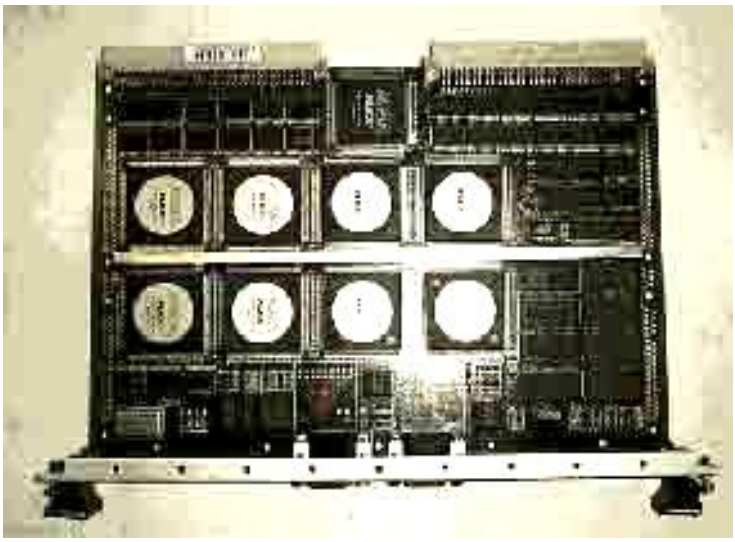


Рис. П.4. Нейрокомпьютер ППК

Рис. П.5. Процессорный модуль ADP6701PCI компании Инструментальные системы на базе ПЦОС TMS320C6701

Рис. П.6. Нейрокомпьютер DSP60V6 компании Инструментальные системы

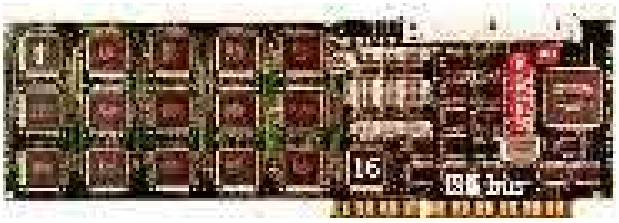


Рис. П.7. ISA-нейроускоритель ZISC 036 компании IBM

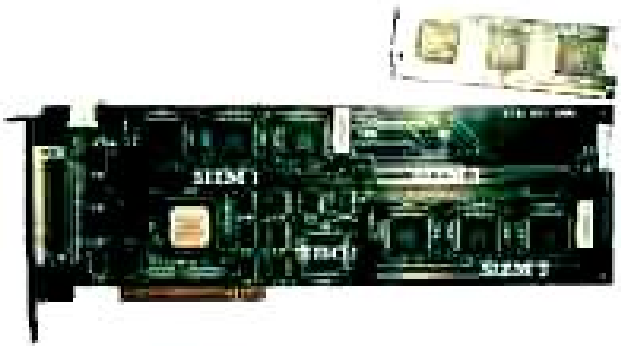


Рис. П.8. PCI-нейроускоритель ZISC 036 компании IBM

Рис. П.9. Нейрокомпьютер Synapse 2 компании Siemens

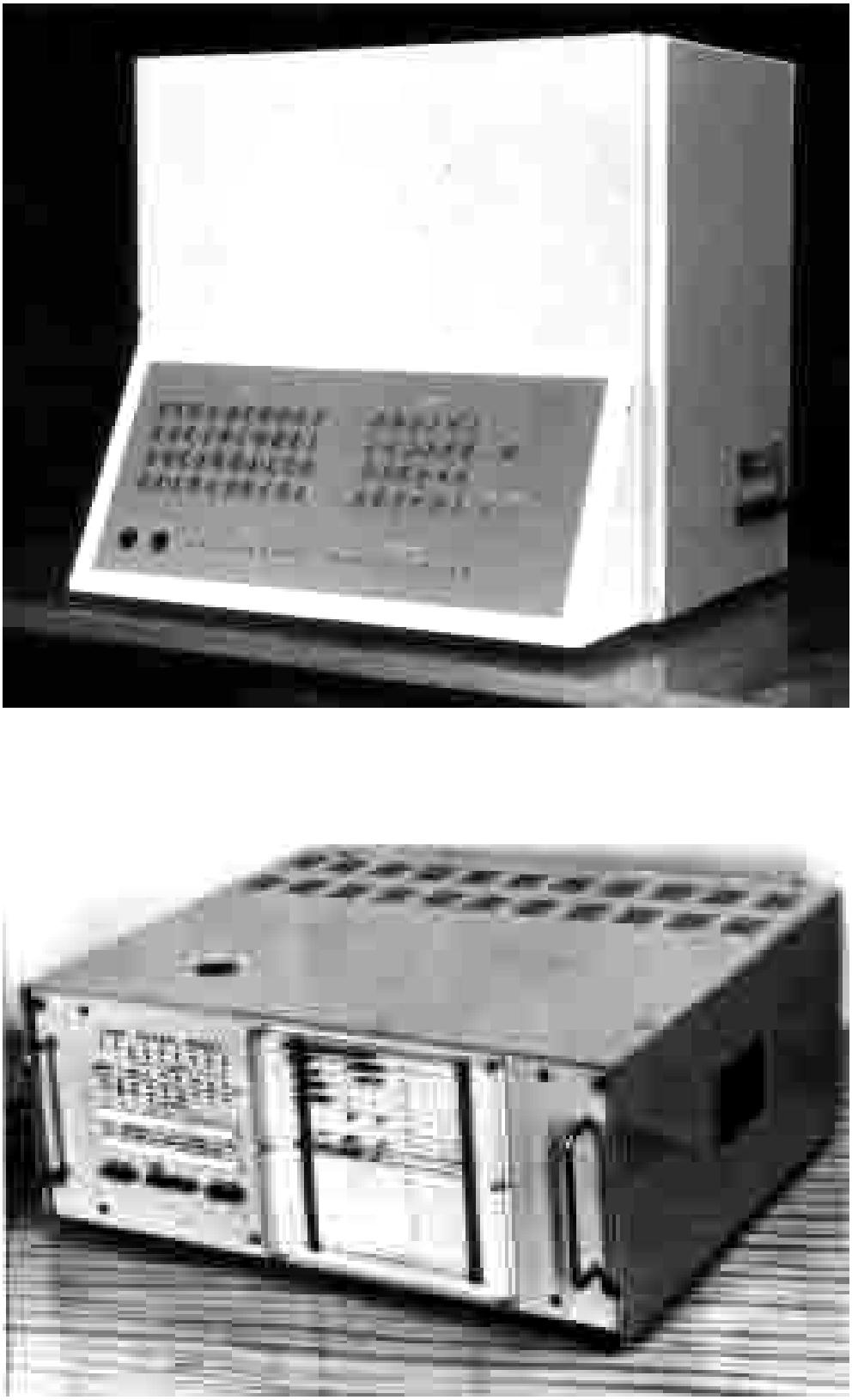


Рис. П.10. Прикладные нейрокомпьютеры «Эмбрион»

P.G. Krug
The Neural Networks and The Neural Computers. Moscow: Publishing House of MPEI. 2002 – 176 pp.
ISBN 5-7046-0832-9

The basics of the Neural Networks, popular solving problems, applications, simulation software products, and also the modern neural processors and neural computers are considered.
The book contains the practical course, which based on Trajan simulator.
The book reflects the experience of teaching foreign students in Neural Networks in English at the Moscow Power Engineering Institute (Technical University).
For students dealing with Computer Sciences.

Учебное издание

Круг Петр Германович

НЕЙРОННЫЕ СЕТИ И НЕЙРОКОМПЬЮТЕРЫ

Учебное пособие
по курсу “Микропроцессоры” для студентов, обучающихся
по направлению “Информатика и вычислительная техника”

Редактор издательства Черныш Н.Л.
ЛР № 020528 от 05.06.97 г.

Темплан издания МЭИ 2002 (I), учебн.
Подписано к печати 10.08.02. Формат 60x84/16
Печ. л. 11,0 Тираж 100 Заказ Изд. № 15 Цена 33 р.

Издательство МЭИ, 111250 Москва, Красноказарменная, д. 14
Типография ЦНИИ «Электроника»,
117415, Москва, просп. Вернадского, д. 39