ВВЕДЕНИЕ

В современных системах автоматизированного проектирования широко применяются методы математического моделирования, позволяющие на основе достаточно точных математических моделей проводить исследование свойств технических объектов, проводить их полный расчёт и оптимизацию. Процесс автоматизированного проектирования, начиная с ранних стадий разработки, позволяет накапливать информацию, уточнять модель и в результате разрабатывать проект изделия с заданными потребительскими свойствами.

Технология нейронных вычислительных сетей показала свою эффективность при решении задач распознавания образов, кластеризации данных, ассоциативного поиска информации в базах данных и в ряде других применений.

Традиционно нейронные сети реализуются в форме программ на универсальных компьютерах, или в форме электронных схем, выполненных на микропроцессорах или на специализированных нейронных процессорах (нейрочипах).

Нейроны (нейрочипы) сети выполняют операцию умножения входного сигнала на число (вес входа), складывают сигналы и вычисляют выходной сигнал, на основе заложенной в нейрон функции активации.

Промышленность выпускает цифровые, аналоговые и гибридные нейрочипы, которые работают соответственно с цифровыми, аналоговыми или одновременно аналоговыми и цифровыми сигналами.

Нейрочип – специализированный микропроцессор, оптимизированный для массового выполнения нейронных операций: скалярного умножения и нелинейного преобразования сигналов, изготовленный по технологии микроэлектроники. Для создания реально работающих нейронных сетей на основе существующих нейрочипов необходимы десятки и сотни

микросхем, что делает проекты достаточно дорогими и поэтому не находящими широкого спроса.

Работы по исследованию и разработке нейронных сетей проводятся с середины прошлого века. Теоретически показано, что наиболее универсальны многослойные нейронные сети с пространственной организацией, в которой входы и выходы каждого нейрочипа могут быть подключены к входам и выходам любого другого нейрочипа в сети. Аппаратная реализация таких сетей на кристалле в рамках традиционной планарной технологии микроэлектроники очень сложна, что не позволило до настоящего времени создать дешевые нейронные сети.

Нейронная сеть содержит большое количество одинаковых элементов – нейронов, и относится к классу вычислительных сетей с распределенными ресурсами. Повысить производительность нейронной сети можно, или уменьшая число каналов обмена информацией между нейронами, или увеличивая степень интеграции элементов в нейрочипе. Наиболее перспективным направлением реализации нейрочипов для нейронных сетей следует признать развитие КМОП технологии, применяемой для изготовления современных программируемых логических интегральных микросхем (ПЛИС). Архитектура ПЛИС содержит блоки элементов памяти и конфигурируемые логические блоки (КЛБ). Локальная связь между этими элементами осуществляется при помощи проводников трассировочных матриц, подключаемых при помощи двунаправленных транзисторных программируемых переключателей. Связь с внешней платой осуществляется при помощи двунаправленных программируемых блоков ввода-вывода.

Нейронная сеть может быть аппаратно реализована на системе ПЛИС. Однако по своим функциональным возможностям и цене ПЛИС слишком сложны, дороги и плохо согласуются с алгоритмами работы нейронной сети.

Развитие КМОП технологии привело к созданию элементов энерго-

независимой памяти на базе нанокристаллов Si, Ge, в пленке SiO_2. Такой нанокристалл совместно с подведенными к нему электродами образует элемент, близкий по свойствам к униполярному МОП транзистору с плавающим затвором. Комплементарные пары таких транзисторов служат элементной базой при создании гигабайтных микросхем «флэш-памяти» новой архитектуры.

Эти два технологических направления открывают перспективы создания гигабайтных нейронных сетей на одном кристалле. Группа элементов памяти и логических элементов, созданных на базе нанокристаллов кремния и германия, образуют нейроны нейронной сети. Связь между нейронами осуществляется при помощи проводников трассировочных матриц. В отличие от ПЛИС, структура нейронной сети достаточно проста и регулярна, что приводит к резкому уменьшению числа программируемых переключателей и объема предназначенной для управления ими теневой памяти.

В результате на одном кристалле в рамках стандартной технологии можно разместить значительно больше нейронов, чем в ПЛИС. Это технологическое направление перспективно для создания массовых и дешевых гигабайтных нейрочипов, позволяющих аппаратно реализовывать сложные нейронные сети.

Проблемам моделирования и функционирования нейронных сетей на основе твердотельных объектов посвящены работы многих российских, советских и зарубежных учёных.

Развитие технологии производства интегральных микросхем нового поколения ставит задачу изучения алгоритмов обработки информации в нейронных сетях, оптимизации структуры нейронной сети и структуры нейрона, соответствующих возможностям технологии. Это делает задачу разработки элементов информационной технологии в проектировании нейронных сетей на основе твёрдотельных объектов актуальной и своевременной.

Особенностью процессов распознавания образов в кластерных системах обработки информации является их длительность и большой объём входных данных.

Проектирование процессов распознавания образов невозможно без понимания принципов построения кластерных и распределённых систем обработки информации. Современные средства моделирования позволяют создавать предварительные проекты подобных систем. В связи с этим остро встаёт вопрос о производительности систем распознавания образов для автоматизации проектирования устройств и систем микро- и наноэлектроники, электронного машино- и приборостроения.

Высокая степень автоматизации современных процессов порождает риск снижения их безопасности (личной, информационной, государственной и т.п.). Доступность и широкое распространение информационных технологий делает их чрезвычайно уязвимыми по отношению к деструктивным воздействиям, в том числе и информационным.

Таким образом, чтобы быть защищённой, система должна успешно противостоять многочисленным и разнообразным угрозам безопасности, действующим в пространстве современных информационных технологий.

С возникновением нанотехнологий появилась техническая возможность сдвинуть ограничения на пространственное разрешение измерительных и исполнительных инструментов в нанометровую и субнанометровую область размеров.

Книга предназначена для инженерно-технических и научных работников, занимающихся информационными технологиями в проектировании объектов электронного машино- и приборостроения, в том числе проектированием нейронных сетей на основе твёрдотельных объектов и процессов распознавания образов в кластерных системах обработки информации. Издание может быть рекомендовано аспирантам вузов и студентам, обучающимся по специальности 210107 – «Электронное машиностроение» и направлению 210100 – «Электроника и наноэлектроника».

ГЛАВА 1. ОБЗОР И АНАЛИЗ В ОБЛАСТИ ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ НЕЙРОННЫХ СЕТЕЙ

1.1. Особенности автоматизированного проектирования искусственных нейронных сетей

Персептрон, перцептрон (англ. perceptron, нем. Perzeptron, от лат. perceptio – понимание, познавание, восприятие), математическая модель процесса восприятия.

Исследования в области нейронных сетей (НС) пережили три периода активизации. Первый пик в 40-х годах обусловлен пионерской работой МакКаллока и Питтса. Второй возник в 60-х – благодаря теореме сходимости перцептрона Розенблатта и работе Минского и Пейперта, указавшей ограниченные возможности простейшего перцептрона. Результаты Минского и Пейперта погасили энтузиазм большинства исследователей, особенно тех, кто работал в области вычислительных наук. Возникшее в исследованиях по нейронным сетям затишье продлилось почти 20 лет. С начала 80-х годов НС вновь привлекли интерес исследователей, что связано с энергетическим подходом Хопфилда и алгоритмом обратного распространения для обучения многослойного перцептрона (многослойные сети прямого распространения), впервые предложенного Вербесом и независимо разработанного рядом других авторов. Алгоритм получил известность благодаря Румельхарту в 1986 году. Андерсон и Розенфельд подготовили подробную историческую справку о развитии НС [1÷7].

Нейрон (нервная клетка) является особой биологической клеткой, которая обрабатывает информацию (рис. 1.1). Она состоит из тела клетки (cell body), или сомы (soma), и двух типов внешних древоподобных ветвей:

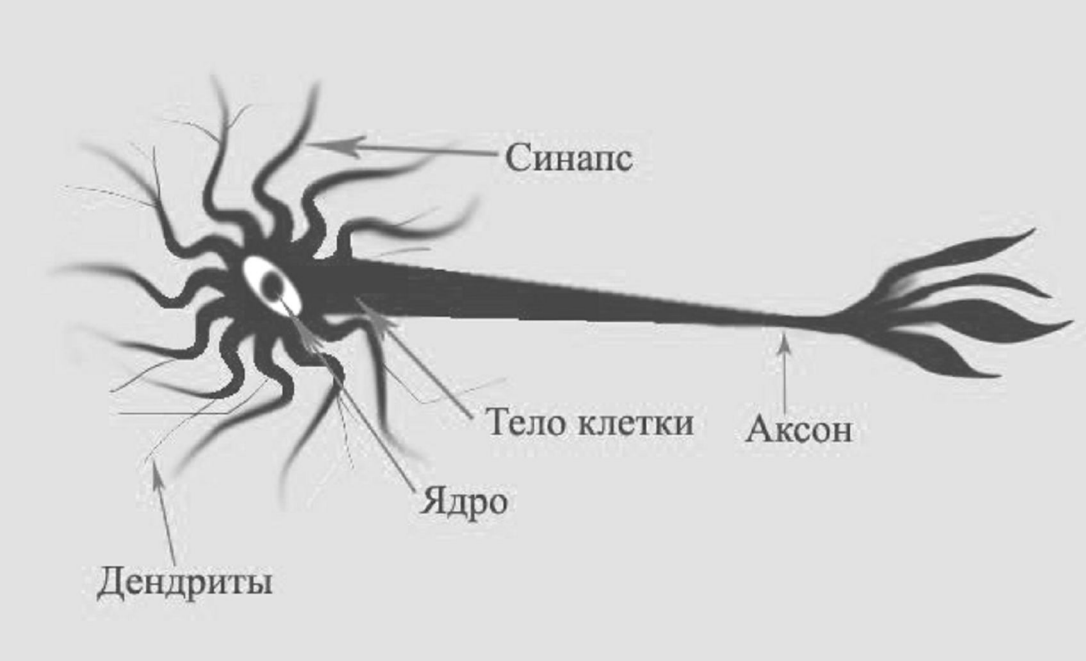


Рис. 1.1. Схема биологического нейрона

аксона (axon) и дендритов (dendrites). Тело клетки включает ядро (nucleus), которое содержит информацию о наследственных свойствах, и плазму, обладающую молекулярными средствами необходимых нейрону материалов. Нейрон получает сигналы (импульсы) от других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль аксона (передатчик), который в конце разветвляется на волокна (strands). На окончаниях этих волокон находятся синапсы (synapses).

Синапс является элементарной структурой и функциональным узлом между двумя нейронами (волокно аксона одного нейрона и дендрит другого). Когда импульс достигает синаптического окончания, высвобождаются определенные химические вещества, которые называются нейротрансмиттерами.

Нейротрансмиттеры диффундируют через синаптическую щель, возбуждая или затормаживая, в зависимости от типа синапса, способность нейрона-приёмника генерировать электрические импульсы. Результатив-

ность синапса может настраиваться проходящими через него сигналами, так что синапсы могут обучаться в зависимости от активности процессов, в которых они участвуют. Эта зависимость от предыстории действует на память, которая, возможно, ответственна за память человека.

Кора головного мозга человека является протяжённой, образованной нейронами поверхностью толщиной от 2 до 3 мм с площадью около 2200 см , что вдвое превышает площадь поверхности стандартной клавиатуры. Кора головного мозга содержит около 1011 нейронов, что приблизительно равно числу звезд Млечного пути. Каждый нейрон связан с 103 ÷ 104 другими нейронами. В целом мозг человека содержит от 1014 до 1015 взаимосвязей [8÷11].

Нейроны взаимодействуют посредством короткой серии импульсов, как правило, продолжительностью несколько мс. Сообщение передается посредством частотно-импульсной модуляции. Частота может изменяться от нескольких единиц до сотен Гц, что в миллион раз медленнее, чем самые быстродействующие переключательные электронные схемы. Тем не менее сложные решения по восприятию информации, как например, распознавание лица, человек принимает за несколько сотен мс. Эти решения контролируются сетью нейронов, которые имеют скорость выполнения операций всего несколько мс. Это означает, что вычисления требуют не более 100 последовательных стадий. Другими словами, для таких сложных задач мозг «запускает» параллельные программы, содержащие около 100 шагов. Это известно как правило 100 шагов [12]. Рассуждая аналогичным образом, можно обнаружить, что количество информации, посылаемое от одного нейрона другому, должно быть очень маленьким (несколько бит). Отсюда следует, что основная информация не передается непосредственно, а захватывается и распределяется в связях между нейронами. Этим объясняется такое название, как коннекционистская модель, применяемое к НС.

1.1.1. От биологических сетей к нейронным

Современные цифровые вычислительные машины превосходят человека по способности производить числовые и символьные вычисления. В свою очередь, НС могут без усилий решать сложные задачи восприятия внешних данных (например, узнавание человека в толпе только по его промелькнувшему лицу) с такой скоростью и точностью, что мощнейший в мире компьютер по сравнению с ними кажется безнадёжным тугодумом. В чем причина столь значительного различия в их производительности? Архитектура биологической нейронной системы совершенно не похожа на архитектуру машины фон Неймана (табл. 1.1) и существенно влияет на типы функций, которые более эффективно исполняются каждой из существующих моделей.

Подобно биологической нейронной системе НС является вычислительной системой с огромным числом параллельно функционирующих простых процессоров с множеством связей. Модели НС в некоторой степени воспроизводят «организованные» принципы, свойственные мозгу человека. Моделирование биологической нейронной системы с использованием НС также может способствовать лучшему понимаю биологических функций. Такие технологии производства, как VLSI (сверхвысокий уровень интеграции) и оптические аппаратные средства, делают возможным подобное моделирование [13].

Глубокое изучение НС требует знания нейрофизиологии, науки о познании, психологии, физики (статической механики), теории управления, теории вычислений, проблем искусственного интеллекта, статики/математики, распознавания образов, компьютерного зрения, параллельных вычислений и аппаратных средств (цифровых / аналоговых / VLSI / оптических). С другой стороны, НС также стимулируют эти дисциплины, обеспечивая их новыми инструментами и представлениями. Этот симбиоз

Таблица 1.1

Машина фон Неймана по сравнению с биологической нейронной системой

	Машина фон Неймана	**Биологическая нейронная система**
Процессор	Сложный	Простой
	Высокоскоростной	Низкоскоростной
	Один или несколько	Большое количество
Память	Отделена от процессора	Интегрирована в процессор
	Локализована	Распределенная
	Адресация не по содержанию	Адресация по содержанию
Вычисления	Централизованные	Распределенные
	Последовательные	Параллельные
	Хранимые программы	Самообучение
Надежность	Высокая уязвимость	Живучесть
Специализация	Численные и символьные операции	Проблемы восприятия
Среда функционирования	Строго определенная	Плохо определенная
	Строго ограниченная	Без ограничений

жизненно необходим для проведения исследований по нейронным сетям.

Представим некоторые проблемы, решаемые в контексте НС и представляющие интерес для учёных и инженеров.

Классификация образов. Задача состоит в указании принадлежно-

сти входного образа, представленного вектором признаков, одному или нескольким предварительно определённым классом.

Кластеризация/категоризация. При решении задачи кластеризации, которая известна также как классификация образов «без учителя», отсутствует обучающая выборка с метками классов. Алгоритм кластеризации основан на подобии образов и размещает близкие образы в один кластер. Известны случаи применения кластеризации для извлечения знаний, сжатия данных и исследования свойств данных.

Аппроксимация функций. Предположим что имеется обучающая выборка ((x_1, y_1), (x_2, y_2)..., (x_n, y_n)) (пары данных вход-выход), которая генерируется неизвестной функцией (x), искаженной шумом. Задача аппроксимации состоит в нахождении оценки известной функции (x). Аппроксимация функций необходима при решении многочисленных инженерных и научных задач моделирования.

Предсказание/прогноз. Пусть заданы n дискретных отсчетов $\{y(t_1), y(t_2)..., y(t_n)\}$ в последовательные моменты времени t_1, t_2,..., t_n. Задача состоит в предсказании значения $y(t_{n+1})$ в некоторый будущий момент времени t_{n+1}. Предсказание/прогноз имеют значительное влияние на принятие решений в бизнесе, науке и технике. Предсказание цен на фондовой бирже и прогноз погоды являются типичным приложениями техники предсказания/прогноза.

Оптимизация. Многочисленные проблемы в математике, статике, технике, науки, медицине и экономики могут рассматриваться как проблемы оптимизации. Задачей алгоритма оптимизации является нахождение такого решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию. Задача коммивояжера, относящаяся к классу NP-полных, является классическим примером задачи оптимизации.

Память адресуемая по содержанию. В модели вычислений фон

Неймана обращение к памяти доступно только посредством адреса, который не зависит от содержания памяти. Более того, если допущена ошибка в вычислении адреса, то может быть найдена совершенно другая информация. Ассоциативная память, или память, адресуемая по содержанию, доступна по указанию заданного содержания. Содержимое памяти может быть вызвано даже по частичному входу или искаженному содержанию. Ассоциативная память желательна при создании мультимедийных баз данных.

Управление. Рассмотрим динамическую систему, заданную совокупностью $\{u(t), y(t)\}$, где $u(t)$ является входным управляющим воздействием, а $y(t)$ – выходом системы в момент времени t. В системах управления с эталонной моделью целью управления является расчёт такого входного воздействия $u(t)$, при котором система следует по желаемой траектории, диктуемой эталонной моделью. Примером является оптимальное управление двигателем [9].

1.2. Модель нейрона

МакКаллок и Питтс предложили использовать бинарный пороговый элемент в качестве модели искусственного нейрона. Этот математический нейрон вычисляет взвешенную сумму n входных сигналов x_j, $j = 1, 2... n$, и формирует на выходе сигнал величины 1, если эта сумма превышает определенный порог u, и 0 – в противном случае.

Часто удобно рассматривать u как весовой коэффициент, связанный с постоянных входом $x_0 = 1$. Положительные веса соответствуют возбуждающим связям, а отрицательные – тормозным. МакКаллок и Питтс доказали, что при соответствующим образом подобранных весах, совокупность параллельно функционирующих нейронов подобного типа способна выполнять универсальные вычисления. Здесь наблюдается определенная ана-

логия с биологическим нейроном: передачу сигнала и взаимосвязи имитируют аксоны и дендриты, веса связей соответствуют синапсам, а пороговая функция отражает активность сомы [1÷5].

1.2.1. Архитектура нейронной сети

НС может рассматриваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей НС могут быть сгруппированы в два класса: сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями [1÷10].

В наиболее распространенном семействе сетей первого класса, называемых многослойным персептроном, нейроны расположены слоями и имеют однонаправленные связи между слоями. На рис. 1.2 представлены типовые сети каждого класса. Сети прямого распространения являются

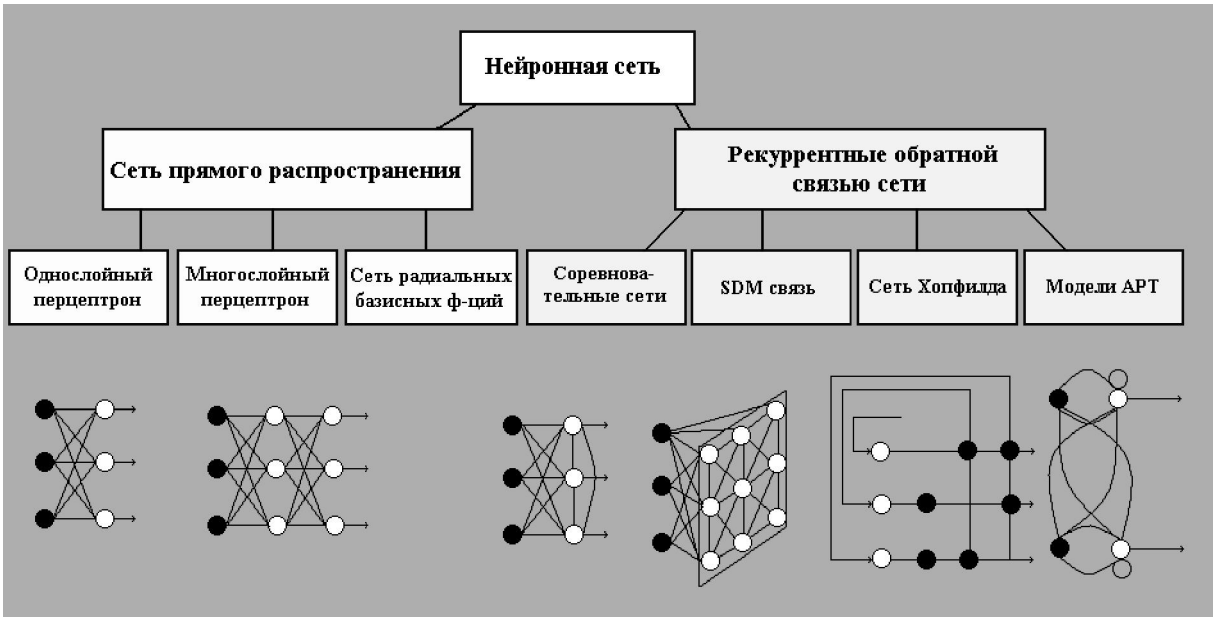


Рис. 1.2. Систематизация архитектур сетей прямого распространения и рекуррентных (с обратной связью)

статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети. Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

1.2.2. Обучение

Способность к обучению является фундаментальным свойством мозга. В контексте НС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с системами, которые следуют определённой системе правил функционирования, сформулированной экспертами.

Для конструирования процесса обучения, прежде всего, необходимо иметь модель внешней среды, в которой функционирует нейронная сеть – знать доступную для сети информацию. Эта модель определяет парадигму обучения [3]. Во-вторых, необходимо понять, как модифицировать весовые параметры сети – какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существуют три парадигмы обучения: "с учителем", "без учителя" (самообучение) и смешанная. В первом случае нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учи-

телем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода. Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям. При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: ёмкость, сложность образцов и вычислительная сложность. Под ёмкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать "переобученность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо – на тестовых примерах, подчинённых тому же статистическому распределению. Известны четыре основных типа правил обучения: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования.

Правило коррекции по ошибке. При обучении с учителем для каждого входного примера задан желаемый выход d. Реальный выход сети y может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала $(d - y)$ для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда персептрон ошибается. Известны различные модификации этого алгоритма обучения [2].

Обучение Больцмана. Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термоди-

намических принципов [10]. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

Правило Хебба. Самым старым обучающим правилом является постулат обучения Хебба [13]. Хебб опирался на следующие нейрофизиологические наблюдения: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом. Это существенно упрощает цепи обучения в реализации VLSI.

Обучение методом соревнования. В отличие от обучения Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно как правило "победитель берет все". Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования позволяет кластеризовать входные данные: подобные примеры группируются сетью в соответствии с корреляциями и представляются одним элементом.

При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счёт такого изменения сохраненного в сети образца (вектора весов связей победившего нейрона), при котором он становится чуть ближе ко входному примеру. На рис. 1.3 дана геометрическая иллюстрация обучения методом соревнования. Входные векторы нормализованы и представлены точками на поверхности сферы. Векторы весов для трёх нейронов инициализированы случайными значениями. Их

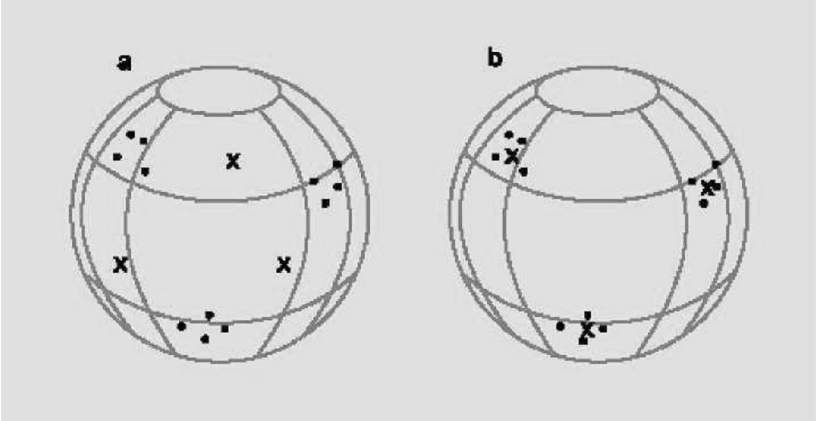


Рис. 1.3. Пример обучения методом соревнования:
а – перед обучением;
б – после обучения

начальные и конечные значения после обучения отмечены *X* на рис. 1.3,а и 1.3,б соответственно. Каждая из трёх групп примеров обнаружена одним из выходных нейронов, чей весовой вектор настроился на центр тяжести обнаруженной группы.

Можно заметить, что сеть никогда не перестанет обучаться, если параметр скорости обучения не равен 0. Некоторый входной образец может активизировать другой выходной нейрон на последующих итерациях в процессе обучения. Это ставит вопрос об устойчивости обучающей системы. Система считается устойчивой, если ни один из примеров обучающей выборки не изменяет своей принадлежности к категории после конечного числа итераций обучающего процесса. Один из способов достижения стабильности состоит в постепенном уменьшении до 0 параметра скорости обучения. Однако это искусственное торможение обучения вызывает другую проблему, называемую пластичностью и связанную со способностью к адаптации к новым данным. Эти особенности обучения методом соревнования известны под названием дилеммы стабильности-пластичности Гроссберга.

В табл. 1.2 представлены различные алгоритмы обучения и связанные с ними архитектуры сетей (список не является исчерпывающим). В последней колонке перечислены задачи, для которых может быть применен каждый алгоритм. Каждый алгоритм обучения ориентирован на сеть определенной архитектуры и предназначен для ограниченного класса задач. Кроме рассмотренных, следует упомянуть некоторые другие алгоритмы: Adaline и Madaline, линейный дискриминантный анализ, проекции Саммона, анализ главных компонентов [10÷12].

1.3. Многослойные сети прямого распространения

Стандартная L-слойная сеть прямого распространения состоит из слоя входных узлов (будем придерживаться утверждения, что он не включается в сеть в качестве самостоятельного слоя), (L-1) скрытых слоёв и выходного слоя, соединённых последовательно в прямом направлении и не содержащих связей между элементами внутри слоя и обратных связей между слоями. На рис. 1.4 приведена структура трёхслойной сети.

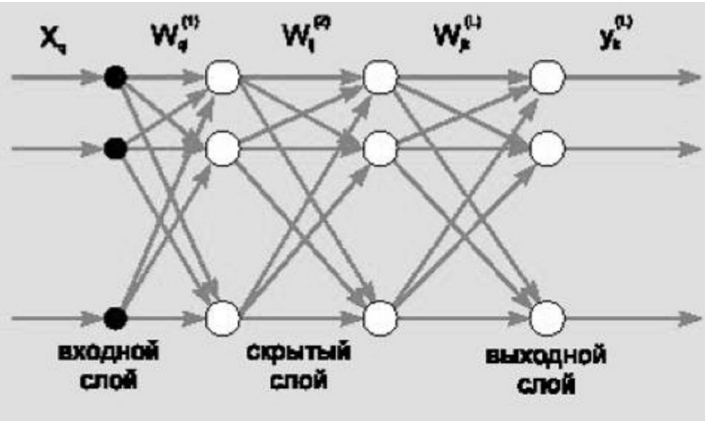


Рис. 1.4. Типовая архитектура трехслойной сети прямого распространения

Таблица 1.2

Известные алгоритмы обучения

Парадигма	Обучающее правило	Архитектура	Алгоритм обучения	Задача
С Учителем	Коррекция ошибки	Однослойный и многослойный персептрон	Алгоритмы обучения персептрона Обратное распространение Adaline и Madaline	Классификация образов Аппроксимация функций Предсказание, управление
	Больцман	Рекуррентная	Алгоритм обучения Больцмана	Классификация образов
	Хебб	Многослойная прямого распространения	Линейный дискриминантный анализ	Анализ данных Классификация образов
	Соревнование	Соревнование	Векторное квантование	Категоризация внутри класса Сжатие данных
		Сеть ART	ARTMap	Классификация образов
Без учителя	Коррекция ошибки	Многослойная прямого распространения	Проекция Саммона	Категоризация внутри класса Анализ данных
	Хебб	Прямого распространения или соревнование	Анализ главных компонентов	Анализ данных Сжатие данных
		Сеть Хопфилда	Обучение ассоциативной памяти	Ассоциативная память

Без учителя	Соревнование	Соревнование	Векторное квантование	Категоризация Сжатие данных
		SOM Кохонена	SOM Кохонена	Категоризация Анализ данных
		Сети ART	ART1, ART2	Категоризация
Смешанная	Коррекция ошибки и соревнование	Сеть RBF	Алгоритм обучения RBF	Классификация образов Аппроксимация функций Предсказание, управление

1.3.1. Многослойный персептрон

Наиболее популярный класс многослойных сетей прямого распространения образуют многослойные персептроны, в которых каждый вычислительный элемент использует пороговую или сигмоидальную функцию активации. Многослойный персептрон может формировать сколь угодно сложные границы принятия решения и реализовывать произвольные булевы функции [6]. Разработка алгоритма обратного распространения для определения весов в многослойном персептроне сделала эти сети наиболее популярными у исследователей и пользователей нейронных сетей.

Геометрическая интерпретация объясняет роль элементов скрытых слоёв (используется пороговая активационная функция) [13, 14].

1.3.2. RBF-сети

Сети, использующие радиальные базисные функции (RBF-сети), являются частным случаем двухслойной сети прямого распространения. Каждый элемент скрытого слоя использует в качестве активационной функции радиальную базисную функцию типа гауссовой. Радиальная базисная функция (функция ядра) центрируется в точке, которая определяется весовым вектором, связанным с нейроном. Как позиция, так и ширина функции ядра должны быть обучены по выборочным образцам. Обычно ядер гораздо меньше, чем создаётся обучающих примеров. Каждый выходной элемент вычисляет линейную комбинацию этих радиальных базисных функций. С точки зрения задачи аппроксимации скрытые элементы формируют совокупность функций, которые образуют базисную систему для представления входных примеров в построенном на ней информационном пространстве.

Существуют различные алгоритмы обучения RBF-сетей [3]. Основной алгоритм использует двушаговую стратегию обучения, или смешанное обучение. Он оценивает позицию и ширину ядра с использованием алгоритма кластеризации "без учителя", а затем алгоритм минимизации среднеквадратической ошибки "с учителем" для определения весов связей между скрытым и выходным слоями. Поскольку выходные элементы линейны, применяется неитерационный алгоритм. После получения этого начального приближения используется градиентный спуск для уточнения параметров сети.

Этот смешанный алгоритм обучения RBF-сети сходится гораздо быстрее, чем алгоритм обратного распространения для обучения многослойных персептронов. Однако RBF-сеть часто содержит слишкомбольшое число скрытых элементов. Это влечёт более медленное функционирование RBF-сети, чем многослойного персептрона. Эффективность (ошибка в за-

висимости от размера сети) RBF-сети и многослойного персептрона зависят от решаемой задачи [13,17,45].

1.3.3. Нерешённые проблемы

Существует множество спорных вопросов при проектировании сетей прямого распространения – например, сколько слоёв необходимы для данной задачи, сколько следует выбрать элементов в каждом слое, как сеть будет реагировать на данные, не включенные в обучающую выборку (какова способность сети к обобщению), и какой размер обучающей выборки необходим для достижения "хорошей" способности сети к обобщению.

Хотя многослойные сети прямого распространения широко применяются для классификации и аппроксимации функций [2], многие параметры ещё должны быть определены путём проб и ошибок. Существующие теоретические результаты дают лишь слабые ориентиры для выбора этих параметров в практических приложениях.

1.3.4. Самоорганизующиеся карты Кохонена

Самоорганизующиеся карты Кохонена [16] обладают благоприятным свойством сохранения топологии, которое воспроизводит важный аспект карт признаков в коре головного мозга высокоорганизованных животных. В отображении с сохранением топологии близкие входные примеры возбуждают близкие выходные элементы. По существу, основная архитектура сети Кохонена представляет собой двумерный массив элементов, причём каждый элемент связан со всеми n входными узлами.

Такая сеть является специальным случаем сети, обучающейся методом соревнования, в которой определяется пространственная окрестность для каждого выходного элемента. Локальная окрестность может быть

квадратом, прямоугольником или окружностью. Начальный размер окрестности часто устанавливается в пределах от 1/2 до 2/3 размера сети и сокращается согласно определённому закону (например, по экспоненциально убывающей зависимости). Во время обучения модифицируются все веса, связанные с победителем и его соседними элементами.

Самоорганизующиеся карты Кохонена могут быть использованы для проектирования многомерных данных, аппроксимации плотности и кластеризации. Эта сеть успешно применялась для распознавания речи, обработки изображений, в робототехнике и в задачах управления [2]. Параметры сети включают в себя размерность массива нейронов, число нейронов в каждом измерении, форму окрестности, закон сжатия окрестности и скорость обучения.

1.3.5. Модели теории адаптивного резонанса

Дилемма стабильности-пластичности является важной особенностью обучения методом соревнования. Как обучать новым явлениям (пластичность) и в то же время сохранить стабильность, чтобы существующие знания не были стерты или разрушены?

Карпентер и Гроссберг, разработавшие модели теории адаптивного резонанса (ART1, ART2 и ARTMAP) [17], сделали попытку решить эту дилемму. Сеть имеет достаточное число выходных элементов, но они не используются до тех пор, пока не возникнет в этом необходимость. Будем говорить, что элемент распределен (не распределен), если он используется (не используется). Обучающий алгоритм корректирует имеющийся прототип категории, только если входной вектор в достаточной степени ему подобен.

В этом случае они резонируют. Степень подобия контролируется параметром сходства k, $0 < k < 1$, который связан также с числом категорий.

Когда входной вектор недостаточно подобен ни одному существующему прототипу сети, создается новая категория, и с ней связывается нераспределённый элемент со входным вектором в качестве начального значения прототипа. Если не находится нераспределённого элемента, то новый вектор не вызывает реакции сети.

Чтобы проиллюстрировать модель, рассмотрим сеть ART1, которая рассчитана на бинарный (0/1) вход. Упрощённая схема архитектуры ART1 [2] представлена на рис. 1.5. Она содержит два слоя элементов с полными связями.

Направленный сверху вниз весовой вектор *Wj* соответствует элементу *j* входного слоя, а направленный снизу вверх весовой вектор *i* связан с выходным элементом *i*. *i* является нормализованной версией *Wi* . Векторы *W*, сохраняют прототипы кластеров. Роль нормализации состоит в том, чтобы предотвратить доминирование векторов с большой длиной над векторами с малой длиной. Сигнал сброса *R* генерируется только тогда, когда подобие ниже заданного уровня.

Модель ART1 может создать новые категории и отбросить входные примеры, когда сеть исчерпала свою ёмкость [1÷5,35,36,41,42,45]. Однако число обнаруженных сетью категорий чувствительно к параметру сходства.

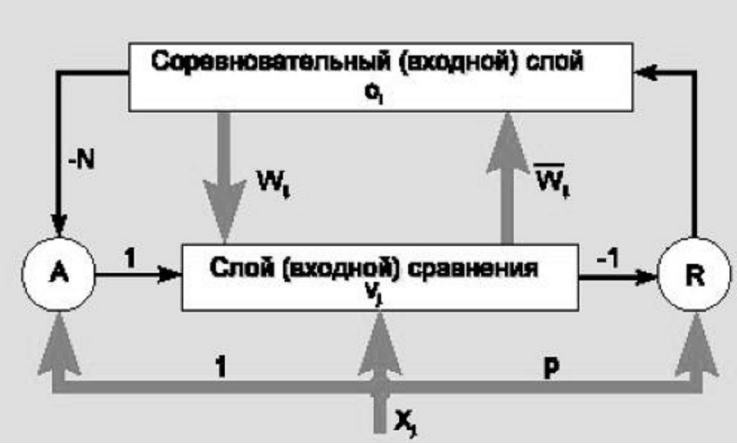


Рис. 1.5. Сеть ART1

1.4. Твёрдотельные объекты

Твёрдотельные объекты входят во многие наноструктуры в виде основных единиц, формирующих твёрдое тело. Процессы ведущие к образованию таких нанокластеров, весьма распространены в природе, например, кристаллизация из раствора или расплава, спекания, различного рода мартенситные превращения, кристаллизация из аморфных систем, образование магнитных и сегнетоэлектрических доменов, спинодальный распад. Все эти процессы подчиняются законам термодинамики и сопровождаются явлением упорядочения и самоорганизации [49].

Образование и организация кластеров в твёрдотельную наносистему во многом определяются способами их получения. При этом формирование наноструктуры возможно из отдельных кластеров, или путём наноструктурирования массивного твёрдого тела. Все эти способы уже имеют большое значение для создания наноматериалов на основе металлов, сплавов, оксидов, керамик и т.д. кроме формирования наноструктур, важным аспектом является их структурные механические и тепловые свойства, определяющие качество и назначение многих материалов.

Молекулярные кластеры металлов – это многоядерные комплексные соединения, в основе молекулярной структуры которых находится окруженный лигандами остов из атомов металлов. Кластером считается ядро, включающее более двух атомов. Металлический остов представляет собой цепи различной длинны, разветвленные циклы, полиэдры и их комбинации.

Молекулярные лигандные кластеры металлов образуются из металлокомплексных соединений в результате проведения химических реакций в растворе. Наибольшее распространение среди методов синтеза больших кластеров получили методы конденсации многоатомных кластеров и восстановление комплексов металлов. В качестве стабилизирующих лигандов используются органические фосфины, особенно PPh_3, или фенантролины.

Таким путём были синтезированы кластеры палладия, обладающие икосаэдрическим ядром и кластерные анионы молибдена. Безлигандные получают в основном тремя основными способами: с помощью сверхзвукового сопла, с помощью газовой агрегации и с помощью испарения с поверхности твёрдого тела или жидкости. Однако от момента получения кластера до момента его фиксации, когда, так сказать, его можно подержать в руках, путь гораздо более длинный, чем для молекулярных кластеров, синтезированных из раствора. Кластеры генерируются с помощью звукового сопла, проходят через диафрагму, ионизируются с помощью электронных или фононных столкновений, разделяются по массам (по отношению m/e на масс-спектрометре) и регистрируются детектором. Такая схема уже даёт основные элементы получения кластеров: это источники кластеров, масс-спектрометры и детекторы [48÷49].

Весьма удобной, можно сказать близкой к модельной, оказалась реакция термического разложения оксалата железа – $Fe_2(C_2)_3 \cdot 5H_2O$. При температуре T_d = 200 ^{0}C происходит дегидратация и разложение оксалата железа с выделением CO и CO_2, формируется та активная среда, в которой начинается нуклеация и образуются нанокластеры оксида железа. Второй минимум температуры T_d = 260 ^{0}C связан с дальнейшим выделением CO и CO_2, началом спекания и образованием наноструктуры, включающей нанокластеры оксида железа. Размеры кластеров увеличиваются с 1 до 6÷7 нм с увеличением температуры разложения и времени выдерживания при данной температуре (увеличение времени способствует гомогенизации кластеров по размерам) [40,41,49].

Термическое разложение оксалатов, цитратов и формиатов железа, кобальта, никеля, меди при температуре 200÷260 °C в вакууме или инертной атмосфере приводит к получению кластеров металлов с размерами 100÷300 нм. Нанокластеры карбидов и нитридов кремния можно синтезировать с помощью высокотемпературного пиролиза при 1300 °C полисила-

занов, поликарбосиланов и поликарбосилаксанов.

Нанокластеры боридов переходных металлов получаются пиролизом борогидридов при более низких температурах 300÷400 ^{0}C, иногда с помощью лазерного воздействия [42÷49].

1.5. Схема нейрона

Рассмотрим классический вариант эксперимента Юнга, в котором имеется лишь один барьер между источником частиц и детектором. Допустим, что можно помещать (или не помещать) в пространство до и/или после барьера пластинку с показателем преломления *n*. Таким образом, оказывается возможным сформировать четыре варианта входа в систему: (1,1), (1, *n),* (*n*,1), (*n,n*). Значения показателя преломления 1 (пластинка отсутствует) и *n* (пластинка присутствует) могут быть использованы для представления бинарных значений входа (например, 1 и 0) [13÷24].

Попытаемся подобрать параметры системы таким образом, чтобы ее выход давал значения функции XOR, а именно

x_1	x_2	y
1	1	0
1	*n*	1
n	1	1
n	*n*	0

Выберем в качестве значений показателя преломления $n = 5/3$, положим $h_0 = h_1 = h$, $r^{(2)}(D) = 0$, и предположим, что $\lambda \rangle\rangle h$ [14].

Выберем также положения двух щелей таким образом, чтобы длины обоих звеньев траектории частицы в мире 1 $\left(l_1^1, l_1^2\right)$ превосходили длины ее

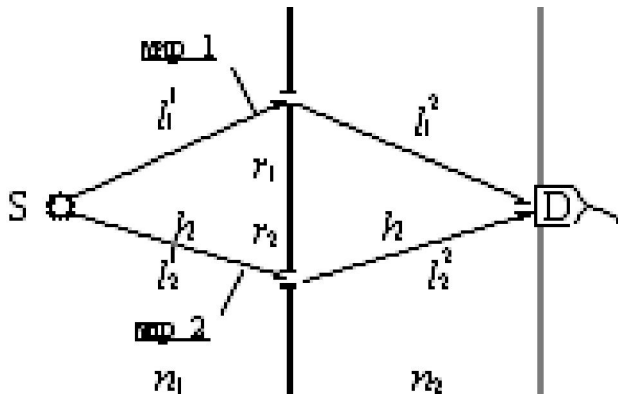


Рис. 1.6. Экспериментальная реализация функции XOR с помощью интерферирующих нейронов в двух мирах

звеньев в мире 2 $\left(l_2^1, l_2^2\right)$ на $\frac{3}{4}\lambda$ (рис. 1.6). Этого можно добиться, полагая [16,18÷20]

$$r_1 = r_2 + \sqrt{1+\frac{h^2}{r^2}}\cdot\tfrac{3}{4}\lambda\ . \tag{1.1}$$

Для таких значений параметров системы мы получаем:

$$\left\langle D\middle|s_1^1\right\rangle\left\langle s_1^1\middle|S\right\rangle = \frac{1}{l_1^1 l_1^2}\exp\left(\frac{2\pi i}{\lambda}\left(n_1 l_1^1 + n_2 l_1^2\right)\right); \tag{1.2}$$

$$\left\langle D\middle|s_2^1\right\rangle\left\langle s_2^1\middle|S\right\rangle = \frac{1}{l_2^1 l_2^2}\exp\left(\frac{2\pi i}{\lambda}\left(n_1 l_2^1 + n_2 l_2^2\right)\right). \tag{1.3}$$

Учитывая, что $\lambda \rangle\rangle h$ можно приближенно положить, что вероятность обнаружения частицы в детекторе, которая определяется интерференцией выходов нейронов из разных миров, составляет

$$\begin{aligned} P(n_1,n_2) &= \left|\left\langle D\middle|s_1^1\right\rangle\left\langle s_1^1(S)\right\rangle + \left\langle D\middle|s_2^1\right\rangle\left\langle s_2^1(S)\right\rangle\right|^2 \\ &\cong \frac{1}{\left(r^2+h^2\right)^2}\left|1+\cos\frac{2\pi}{\lambda}(n_1+n_2)\Delta l\right|^2\ , \end{aligned} \tag{1.4}$$

где $$\Delta l = l_1^1 + l_1^2 - l_2^1 - l_2^2 = \tfrac{3}{2}\lambda \ . \qquad (1.5)$$

Тогда, из выражения (1.7) следует, что [14]

$$P(1,1) = A\left|1 + \cos 3\pi\right|^2 = 0 \ ;$$

$$P(1,n) = A\left|1 + \cos 4\pi\right|^2 = 4A \ ;$$

$$P(n,1) = A\left|1 + \cos 4\pi\right|^2 = 4A \ ; \qquad (1.6)$$

$$P(n,n) = A\left|1 + \cos 5\pi\right|^2 = 0 \ .$$

Таким образом, выход системы совпадает со значением функции XOR с точностью до нормировочного фактора $4A = 4(r^2+h^2)^{-2}$ [15÷17,32].

Заметим, что *единственный квантовый нейрон* оказался способным выполнить функцию XOR, реализация которой на классических нейронах требует построения двухслойной *сети нейронов.*

Обобщённая модель

Можно обобщить описанную выше схему квантового нейрона. В фейнмановском представлении, эволюция любой квантовой системы может быть определена если мы сможем определить все пути, по которым она может достичь данного состояния. Каждый из этих путей определяет отдельный мир. Далее необходимо вычислить комплексную амплитуду соответствующего перехода из начального r_0 в конечное положение r в каждом из этих миров [12÷18]

$$\varphi \ \propto \ \exp\left(\frac{i}{\hbar} S(r_0, r)\right) . \qquad (1.8)$$

Здесь значение действия

$$S(r_0,r)=\int_{t_0}^{t}L(r,\dot{r},t)\,dt=\int_{t_0}^{t}\left(T(\dot{r},t)-U(r)\right)dt\quad,\qquad(1.9)$$

в котором $L(r,\ \dot{r},\ t)$ – функция Лагранжа,

$T(\dot{r},\ t)$ – кинетическая энергия,

$U(r)$ – потенциальная энергия,

является аддитивным и может быть рассмотрено как аналог аддитивной активации квантового нейрона.

Хотя такое рассмотрение в некотором смысле аналогично данному Э. Берман и др. [21]. Оно также ясно выявляет нейроноподобное нелинейное преобразование активации, даваемое выражением (1.8) и отличается тем, что не квантовое состояние определяет вход нейрона, а распределение потенциала $U(r)$. Выходом же нейрона является комплекснозначная амплитуда φ. А соотнесённая с решением задачи регрессии вероятность детектирования частицы является результатом интерференции комплексных выходов нейронов (интерференции миров).

Главное отличие предлагаемого рассмотрения от ранее предложенных подходов к построению квантовых нейронных систем заключается в том, что оно по сути указывает на необязательность использования квантовых нейронных сетей как таковых и, возможно, на достаточность создания единственного квантового нейрона для решения достаточно общих проблем нейрокомпьютинга [15÷19].

Фактически, структура квантовой множественной Вселенной (по Эверетту) представляет в наше распоряжение множество нейронов, существующих в различных мирах, которые, кооперируя друг с другом, и дают требуемый ответ в результате интерференции этих миров [17].

Важно подчеркнуть, что понимание предложенного здесь подхода возможно лишь в рамках Эвереттовской интерпретацией квантовой механики. Это также согласуется с мнением Д. Дойча [18], согласно которому

такая интерпретация существенна для понимания квантовых вычислений вообще. Аналогичную точку зрения поддерживают А. Нараянан и Т. Меннеер [22].

Другое важное свойство предложенной системы состоит в отсутствии необходимости решать очень сложную проблему удержания когерентности квантового состояния, которая является ключевой для реализации алгоритмических (не нейронных) квантовых вычислений. Причиной этого является то, что квантовый нейрон обрабатывает аналоговые входы. В действительности, классический нейрокомпьютинг имеет дело, в основном, с обработкой *аналоговых* сигналов. Обработка образов, кодируемых векторами с дискретными (в частности, бинарными) компонентами, конечно, возможна, но не она составляет главное поле приложения искусственных нейронных сетей. С другой стороны фон Неймановские компьютеры в общем случае являются *цифровыми.* Это приводит к необходимости оперировать с кубитами (вместо битов) в их квантовых аналогах [23,24].

Поскольку нейрокомпьютеры не обязаны обрабатывать биты, их квантовые аналоги не обязаны работать с кубитами. Поэтому, все трудности связанные с необходимостью обрабатывать кубитовый регистр для них не возникают [25].

Конечно, предложенный подход к построению системы квантовой нейронной обработки данных требует использования ансамбля квантов, поскольку вероятность детектирования частицы (градуальный выход системы) может быть оценена лишь статистически. Таким образом, эта система по сути реализует стохастические вычисления [22].

Обучение квантового нейрона в общем случае может быть реализовано стандартным методом градиентного спуска. При этом минимизация функции ошибки

$$E(r) = \tfrac{1}{2}\sum_{u}^{P}\left|y^u - t^u\right|^2 , \qquad (1.10)$$

где обучающее множество содержит P примеров, может быть достигнута путём коррекции положений щелей согласно выражению [23]

$$\Delta\, r_k^j = -\frac{\partial E}{\partial\, r_k^j}\ , \tag{1.11}$$

описали лишь простейшую физическую реализацию нейронной системы.

Для оценки перспективности её использования, необходимо исследовать такие характеристики как:

- универсальность – возможность аппроксимации с необходимой точностью любой функции многих переменных. Очевидно, что увеличение числа миров (определяемого числом щелей в барьерах) увеличивает гибкость системы [22];
- способность к обобщению – необходимо исследовать способность системы обрабатывать новые образы [29].

Одной из задач процесса проектирования является синтез конструктивного варианта объекта, в наибольшей степени удовлетворяющего требованиям технического задания. На начальных стадиях проектирования требования технического задания конкретизируются в виде системы ограничений, которым должны удовлетворять характеристики объекта проектирования, обеспечивающие успешное решение проектной задачи.

ГЛАВА 2. ИССЛЕДОВАНИЕ ПРОЦЕССА СОЗДАНИЯ ЭЛЕМЕНТОВ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПРОЕКТИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

2.1. Элементы нейронных сетей

Для описания алгоритмов и устройств в нейроинформатике выработана специальная "схемотехника", в которой элементарные устройства – сумматоры, синапсы, нейроны и т.п. объединяются в сети, предназначенные для решения задач.

Интересен статус этой схемотехники – ни в аппаратной реализации нейронных сетей, ни в профессиональном программном обеспечении все эти элементы вовсе не обязательно реализуются как отдельные части или блоки. Используемая в нейроинформатике идеальная схемотехника представляет собой особый язык для представления нейронных сетей. При программной и аппаратной реализации, описания, выполненные на этом языке, переводятся на языки другого уровня, более пригодные для применения [35].

Важнейший элемент нейросистем – это *адаптивный сумматор.* Он вычисляет скалярное произведение вектора входного сигнала x на вектор параметров a. На схемах будем обозначать его так, как показано на рис. 2.1. Адаптивным называем его из-за наличия вектора настраиваемых параметров a [36].

Для многих задач полезно иметь линейную неоднородную функцию выходных сигналов. Её вычисление также можно представить с помощью адаптивного сумматора, имеющего $n + 1$ вход и получающего на 0-й вход

постоянный единичный сигнал (рис. 2.2).

Нелинейный преобразователь сигнала изображен на рис. 2.3. Он получает скалярный входной сигнал x и переводит его в $\varphi(x)$.

Точка ветвления служит для рассылки одного сигнала по нескольким адресам (рис. 2.4). Она получает скалярный входной сигнал x и передает его всем своим выходам [34].

Стандартный формальный нейрон составлен из входного сумматора, нелинейного преобразователя и точки ветвления на выходе (рис. 2.5).

Линейная связь – синапс – отдельно от сумматоров не встречается, од-

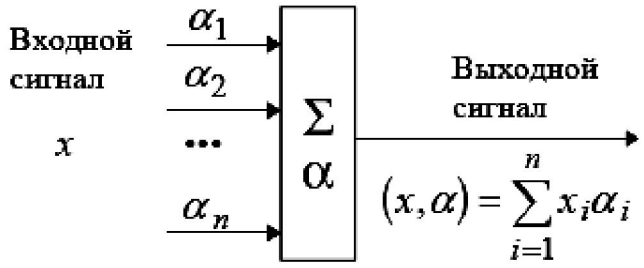


Рис. 2.1. Адаптивный сумматор

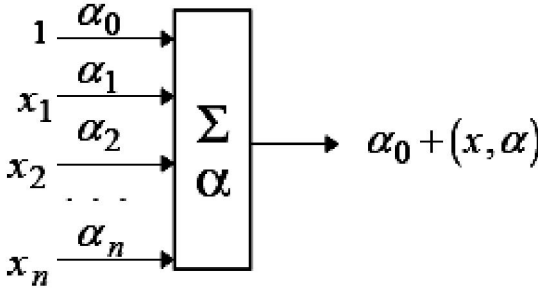


Рис. 2.2. Неоднородный адаптивный сумматор

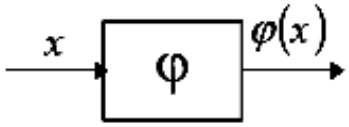


Рис. 2.3. Нелинейный преобразователь сигнала

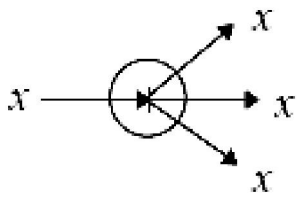


Рис. 2.4. Точка ветвления

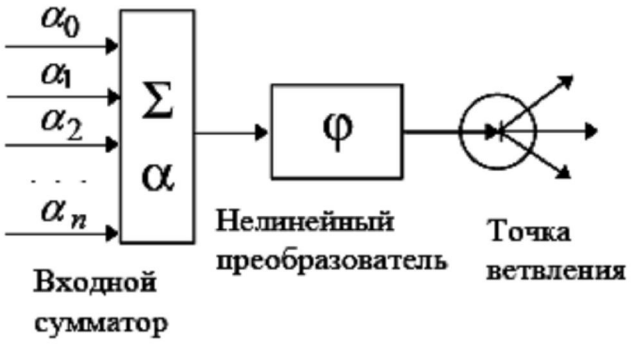


Рис. 2.5. Формальный нейрон

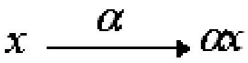


Рис. 2.6. Синапс

нако для некоторых рассуждений бывает удобно выделить этот элемент (рис. 2.6). Он умножает входной сигнал *x* на «вес синапса» *a*.

Также бывает полезно «присоединить» связи не ко входному сумматору, а к точке ветвления. В результате получаем элемент, двойственный адаптивному сумматору и называемый *«выходная звезда»*. Его выходные связи производят умножение сигнала на свои веса.

Итак, дано описание основных элементов, из которых составляются нейронные сети [35÷45].

Теория архитектуры нейронной сети

Как можно составлять сети из элементов? Строго говоря, как угодно, лишь бы входы получали какие-нибудь сигналы. Но такой произвол слишком необозрим, поэтому используют несколько стандартных архитектур, из которых путём вырезания лишнего или (реже) добавления строят большинство используемых сетей.

Сначала следует решить вопрос о том, как будет согласована работа различных нейронов во времени – вопрос о синхронности функционирования. Здесь и далее рассматриваются только нейронные сети, синхронно функционирующие в дискретные моменты времени.

Выделяется две базовых архитектуры нейронных сетей – *слоистые и полносвязные сети*.

Слоистые сети: нейроны расположены в несколько слоёв (рис. 2.7). Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до *k*-го слоя, который выдаёт выходные сигналы для интерпретатора и пользователя. Если не оговорено противоположное, то каждый

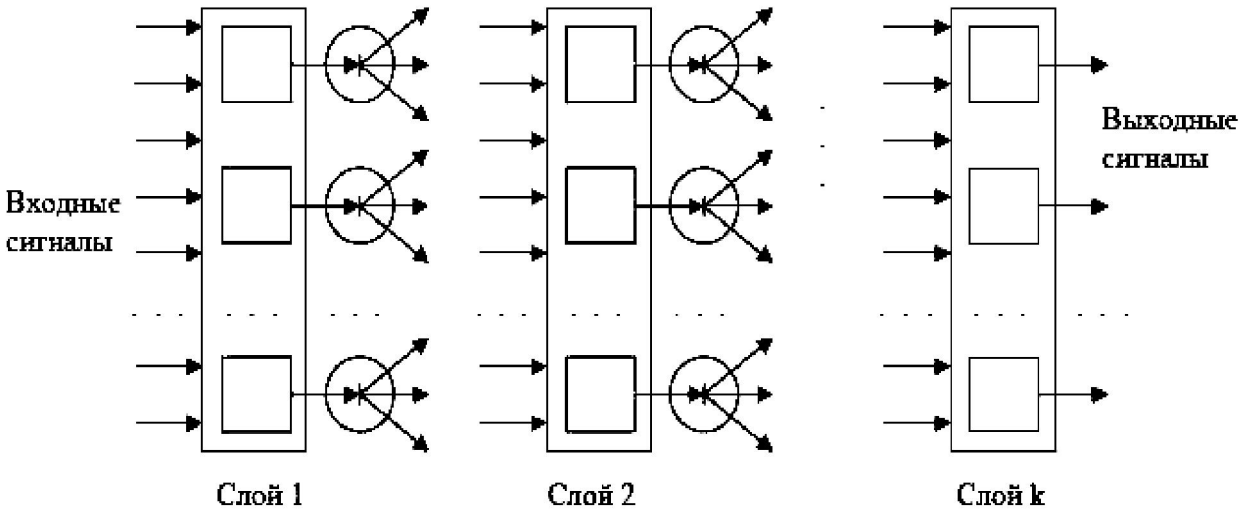


Рис. 2.7. Слоистая сеть

выходной сигнал *i*-го слоя подается на вход всех нейронов *i* +1-го. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: все нейроны первого слоя получают каждый входной сигнал. Особое распространение получили трёхслойные сети, в которых каждый слой имеет своё наименование: первый – входной, второй – скрытый, третий – выходной [41÷46].

Полносвязные сети: каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.

Элементы слоистых и полносвязных сетей могут выбираться поразному. Существует, стандартный выбор – нейрон с адаптивным неоднородным линейным сумматором на входе (см. рис. 2.5).

Для полносвязной сети входной сумматор нейрона фактически распадается на два: первый вычисляет линейную функцию от входных сигналов сети, второй – линейную функцию от выходных сигналов других нейронов, полученных на предыдущем шаге.

Функция активации нейронов (характеристическая функция) φ – нелинейный преобразователь, преобразующий выходной сигнал сумматора (см. рис. 2.5) – может быть одной и той же для всех нейронов сети. В этом случае сеть называют *однородной (гомогенной).* Если же φ зависит ещё от одного или нескольких параметров, значения которых меняются от нейрона к нейрону, то сеть называют *неоднородной (гетерогенной).*

Составление сети из нейронов стандартного вида (см. рис. 2.5) не является обязательным. Слоистая или полносвязная архитектуры не налагают существенных ограничений на участвующие в них элементы. Единственное жесткое требование, предъявляемое архитектурой к элементам се-

ти, это соответствие размерности вектора входных сигналов элемента (она определяется архитектурой) числу его входов.

Если полносвязная сеть функционирует до получения ответа (заданное число тактов *k*), то её можно представить как частный случай *L*-слойной сети, все слои которой одинаковы и каждый из них соответствует такту функционирования полносвязной сети [47].

Существенное различие между полносвязной и слоистой сетями возникает тогда, когда число тактов функционирования заранее не ограничено – слоистая сеть так работать не может.

Элемент нейронной сети (формальный нейрон) (рис. 2.8.) реализуется в составе интегральной микросхемы и в общем случае содержит входной сумматор 1, последовательно связанный с нелинейным преобразователем сигналов 4 и точкой ветвления 5, предназначенной для передачи выходного сигнала по нескольким адресам. Каждый вход A_i входного сумматора 1 называется синапсом. В состав сумматора входит схема умножения входного сигнала на вес синапса 2 и блок суммирования сигналов 3.

Формальный нейрон работает следующим образом. Входной сумматор 1 получает на входы A_i вектор входного сигнала $\{a_0, a_1, \ldots a_n\}$. Каждая

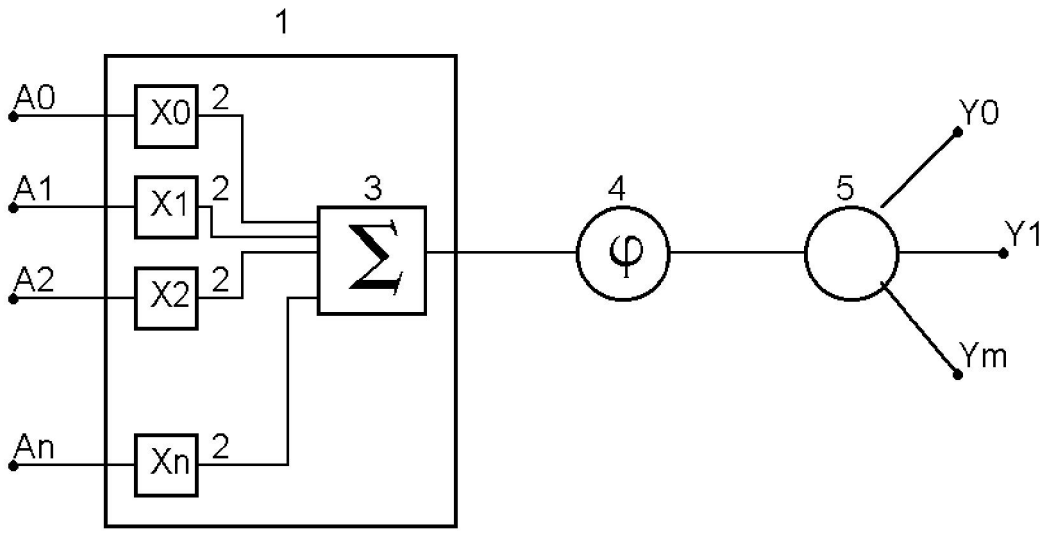


Рис. 2.8. Схема формального нейрона

компонента входного вектора умножается в блоке 2 на вес входа (вес синапса) x_i, а затем результаты суммируются в блоке 3 и передаются на вход нелинейного преобразователя 4. Нелинейный преобразователь 4 на основе выходного сигнала сумматора вырабатывает выходной сигнал нейрона в соответствии с заложенной в него функцией активации $y = \varphi(x)$. Выходной сигнал нейрона через точку ветвления 5 передаётся на входы других формальных нейронов.

Не нарушая общности, поставим задачу создания нейронной сети, предназначенной для распознавания графических образов методом сравнения с эталонами. Графические образы образуются, например, в видеокамере или цифровом фотоаппарате. Эталоны также задаются, например, в форме цифровых фотографий. Алгоритм работы нейронной сети при распознавании изображений состоит из двух обобщенных этапов.

1. Предварительный этап, состоящий в наведении видеокамеры на объект, автофокусировки изображения, масштабирования и центрирования изображения на фотоприемнике видеокамеры.
2. Собственно распознавание изображения методом сравнения с эталонами.

Первый этап решается стандартным набором функций видеокамер, предназначенных для работы в системах видеонаблюдения и обеспечения безопасности. Это, например, видеокамера PANASONIC WV-CS570. В результате получается стандартизованное изображение, грубо расположенное в центре экрана и имеющее стандартный масштаб.

Второй этап, состоящий собственно в распознавании и идентификации изображения, в системах обеспечения безопасности обычно решается оператором. В работе сделана попытка автоматизировать процесс распознавания графических изображений.

На основе теоретического анализа предложен алгоритм распознавания, состоящий из двух этапов:

1. Для каждого графического объекта формируется множество эталонов, отличающихся масштабом, сдвигом, контрастностью и т.д. То, что все эти эталоны относятся к заданному объекту, определяет оператор. На первом этапе система распознавания должна определить эталон, наиболее похожий на графический объект, выбирая его в каждой группе эталонов.
2. На втором этапе из группы отобранных эталонов выбирается один, наиболее похожий на заданный графический объект.

В процессе распознавания каждому эталону назначается число (вес), определяющее близость эталона и графического объекта. В результате анализа выбирается образ (группа эталонов), к которому относится графический объект.

Понятно, что получение однозначного результата при распознавании графических изображений возможно с определенной вероятностью узнавания. На экране видеокамеры может появиться изображение, для которого нет эталона. Тогда результат распознавания может быть неоднозначным, например, система обнаружит два или три похожих эталона.

Рассмотренный алгоритм соответствует алгоритму работы двухслойной нейронной сети с прямым распространением сигналов.

Рассмотрим технологические ограничения на архитектуру нейронной сети. Характерным элементом конструкции ПЛИС является наличие трассировочных матриц, состоящих из параллельных проводников. Две трассировочные матрицы с взаимно пересекающимися проводниками выполняется в двух изолированных друг от друга технологических слоях. Связь проводников с другими элементами микросхемы осуществляется через колодцы, выполненные в слоях изоляции микросхемы.

Эти особенности технологии делают обоснованным выбор регулярной плоской архитектуры нейронной сети, согласованной с технологией её изготовления.

Первый слой нейронной сети работает по следующему алгоритму. Графическое изображение, не уменьшая общности, можно представить в виде вектора X_i , заданного компонентами $\{x_0, x_1, x_2, \ldots x_n\}$ в n-мерном евклидовом пространстве. Каждая координата вектора соответствует пикселю графического изображения. Величина x_i соответствует яркости пикселя изображения.

Изображение-эталон j также представлено в виде группы из k, вектор Y_{jk}, компоненты которого $\{y_{1k}, y_{2\,k}, y_{3\,k}, \ldots y_{nk}\}$ соответствуют яркости соответствующего пикселя в эталонном изображении.

Результат сравнения изображения и эталона вычисляется как скалярное произведение

$$B_j = (X,\ Y_j) = (x_1 y_{1j} + \ldots + x_n y_{nj}) \ . \qquad (2.1)$$

Величина скалярного произведения B_j зависит от выбранного эталона. Поэтому в качестве меры близости двух векторов необходимо взять нормированное на единицу скалярное произведение (или косинус угла между векторами X_i Y_{ij}).

$$b_j = \frac{B_j}{\left|X\right|\left|Y_j\right|} = \frac{x_1 y_{1\,j} + x_2 y_{2\,j} + \ldots + x_n y_{nj}}{\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}\ \sqrt{y_{1j}^2 + y_{2\,j}^2 + \ldots + y_{nj}^2}} \ . \qquad (2.2)$$

Максимальное значение нормированного скалярного произведения b_j (2.2) равно единице, если на вход системы подан вектор-эталон. Для произвольного вектора X величина b_j меньше единицы.

Нормированное скалярное произведение принимается за меру близости вектора-изображения $\boldsymbol{X}$ *и вектора-эталона* $\boldsymbol{Y_j}$.

Вычисление скалярного произведения и нормировка проводится аппаратно в плоском модуле сети (матрице нейронов), представленном на рис. 2.9. Вектор-изображение $X = \{X_i\}$ подается на входы сети. В строках сети расположены формальные нейроны, в которые записаны компоненты

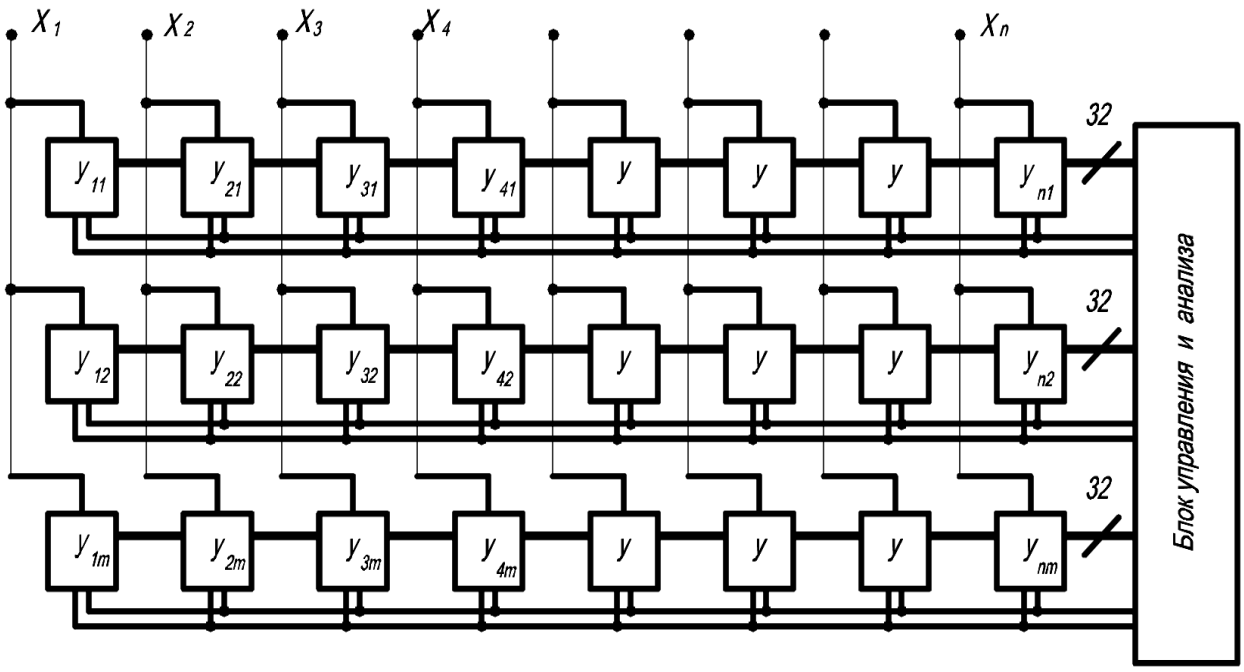


Рис. 2.9. Матрица нейронов – модуль плоской нейронной сети

векторов $Y_j = \{y_{ij}\}$ (весовые коэффициенты). Таким образом, число строк матрицы нейронов m соответствует максимальному числу эталонов для одного изображения.

Каждый формальный нейрон проводит умножение $x_i\, y_{ij}$, складывает полученную величину с значением $x_{i-1}\, y_{(i-1)\,j}$, полученным слева и передает сумму нейрону, расположенному справа. Таким образом, строка нейронов параллельно вычисляет скалярное произведение B_j (2.1) для всех строк и тем самым, для всех эталонов.

Скалярные произведения параллельно передаются в блок управления и анализа, в котором проводится нормировка (2.2) и выбор максимального значения b_j. Это максимальное значение определяет меру соответствия эталону, наиболее похожему на изображение X.

Матрицы формальных нейронов (см. рис. 2.9) образуют первый слой нейронной сети (рис. 2.10). Число матриц (модулей) соответствует числу групп изображений-эталонов M_i .

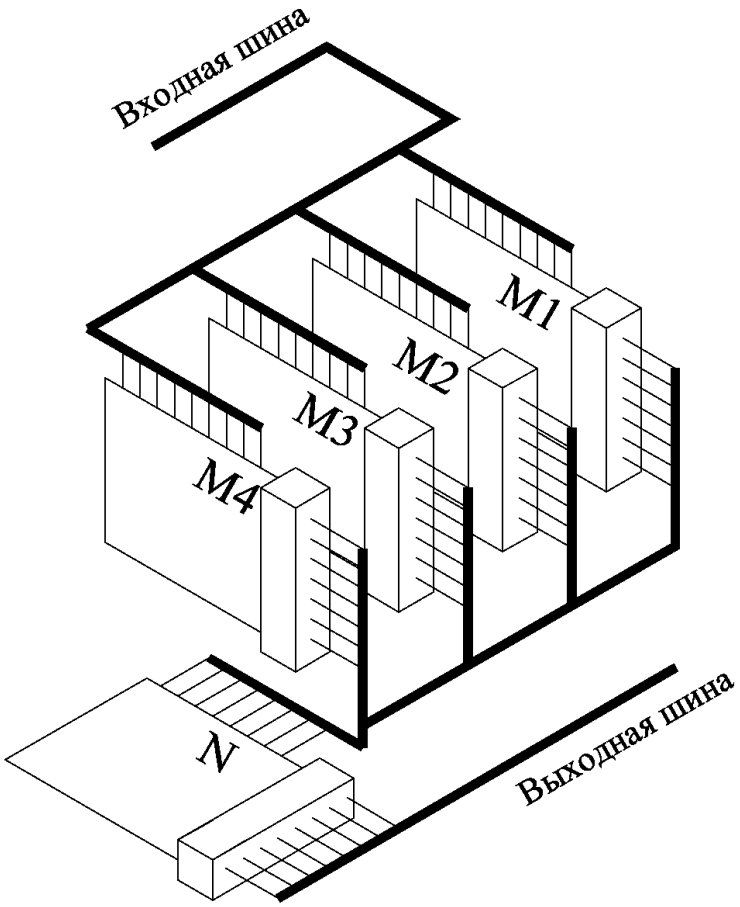


Рис. 2.10. Схема двухслойной нейронной сети

Значения нормированных скалярных произведений b_j с выходов модулей M_j передаются на входы выходного модуля N, который проводит окончательное распознавание изображения.

Компьютерное моделирование алгоритма распознавания изображения в нейронной сети показало его эффективность. В качестве эталонов Y_j принимались однотипные графические объекты разного масштаба (кресты, треугольники, квадраты и т.д.). В качестве входного вектора X выбирался или один из эталонов, или графическое изображение, близкое к одному из эталонов.

Предложенная структура нейронной сети хорошо соответствует современной планарной полупроводниковой технологии. Использование в

качестве базового элемента транзисторов на основе нанокристаллов Si, Ge в пленке SiO_2 позволяет разместить на одном кристалле плоскую нейронную сеть гигабайтного объёма. Параллельный алгоритм вычислений, реализованный в плоской нейронной сети, позволяет проводить распознавание изображений в реальном масштабе времени.

2.2. Твёрдотельные объекты

Наиболее часто, говоря о квантовых точках, имеют в виду спонтанно сформировавшиеся в процессе роста массивы островков-включений одного полупроводникового материала (с меньшей шириной запрещённой зоны) в матрице другого (с большей шириной запрещённой зоны) – рис. 2.11,а. Из-за различия ширины запрещённых зон электроны и дырки ока-

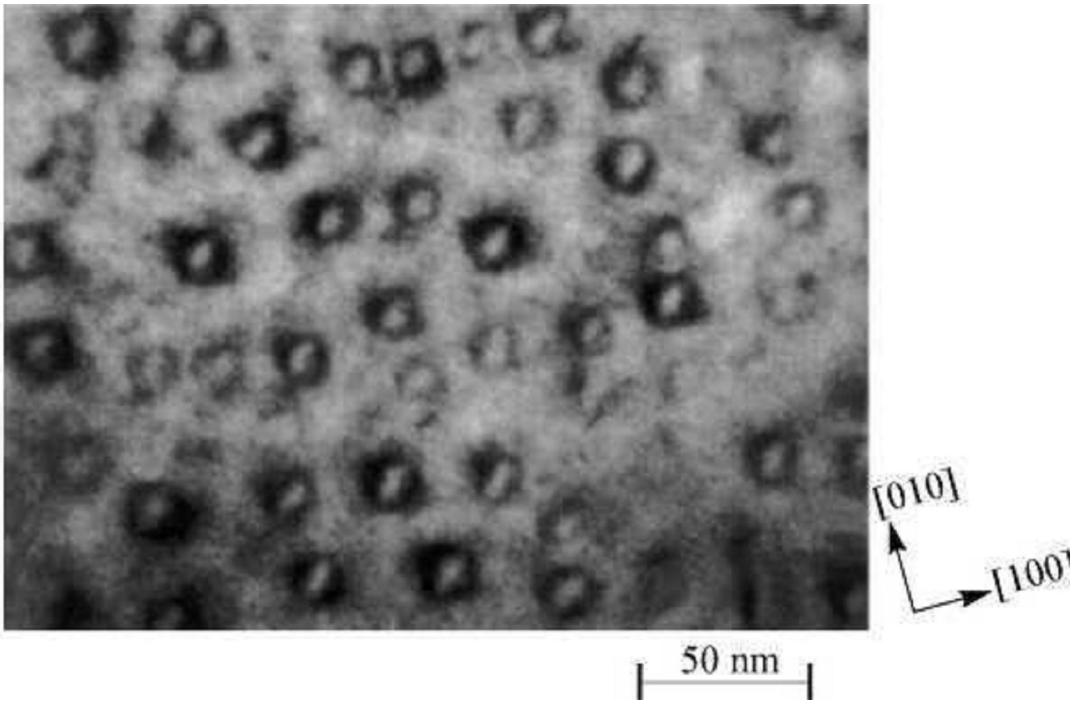


Рис. 2.11,а. Изображение квантовых точек InAs в матрице GaAs (вид сверху), полученное с помощью просвечивающей электронной микроскопии

зываются локализованы во всех трёх направлениях, как бы "заперты" в квантовой точке, следствием чего и является квазиатомный спектр.

На первом рисунке представлен "вид сверху", однако для более детального изучения структуры квантовых точек требуются изображения с большим разрешением и, конечно, исследователи имеют возможность получать такие изображения. В работе [1] с помощью сканирующей туннельной микроскопии (СТМ) были получены изображения квантовых точек с атомным разрешением. Непосредственно после формирования массива квантовых точек в результате выращивания ультратонкого слоя InAs на поверхности GaAs (массив квантовых точек, естественно, не заращивался сверху слоем GaAs) образец был в условиях высокого вакуума перемещён в специальную аналитическую камеру. На рисунках 2.11,б и 2.11,в показаны полученные с помощью СТМ трёхмерные изображения одиночной квантовой точки [41÷43].

Конечно, надо иметь в виду, что даже в случае тех же соединений InAs и GaAs размеры и форма квантовой точки зависят от ростовых условий. В случае других соединений квантовые точки могут формироваться и в форме плоских блинчиков, и в форме микросфер (в диэлектрической матрице).

Квантовые точки, их иногда ещё называют искусственными атомами, представляют собой специальным образом выращенные наноразмерные островки-включения одного полупроводникового материала (с меньшей шириной запрещённой зоны) в матрице другого (с большей шириной запрещённой зоны). Из-за различия ширины запрещённых зон носители заряда оказываются локализованы в пределах островка, следствием чего является квазиатомный (представляющий собой набор отдельных уровней) энергетический спектр [42].

Отдельная квантовая точка представляет собой специальным образом полученный наноразмерный объект, обладающий дискретным энерге

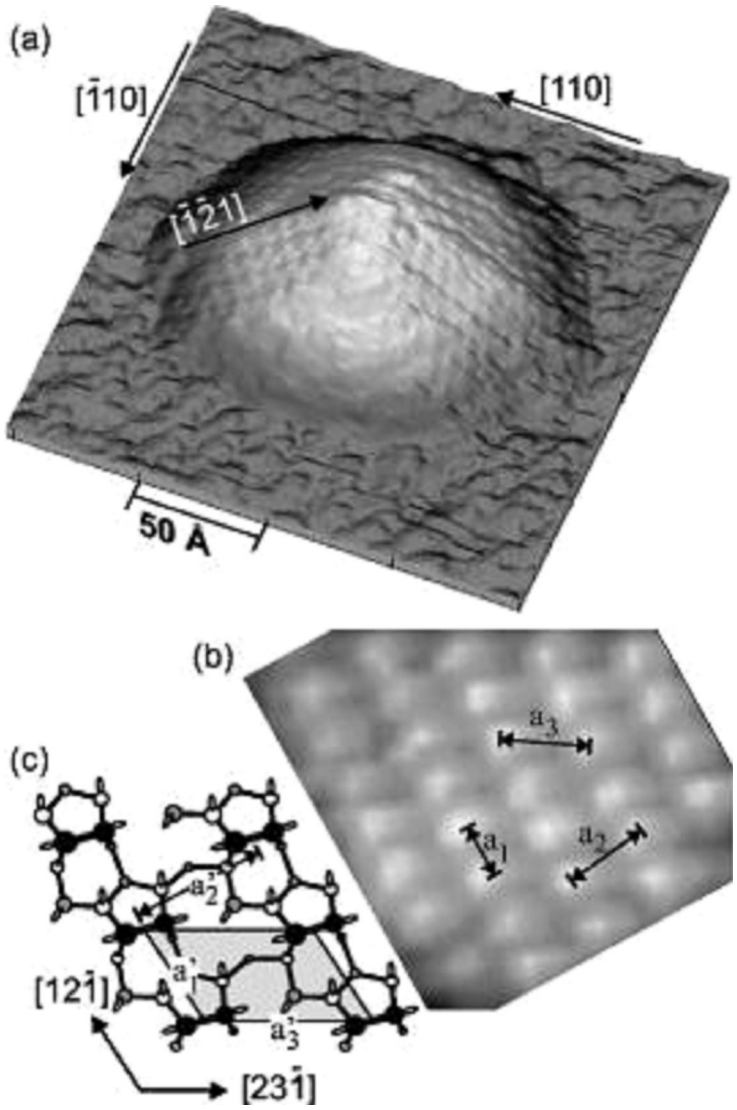


Рис. 2.11,б. Трёхмерное изображение квантовой точки. Видно, что она имеет пирамидальную форму с достаточно острой вершиной:

Стрелки и цифры в квадратных скобках обозначают различные кристаллографические направления;

b –более детальное изображение одной из граней;

c –модель реконструированной поверхности грани;

чёрные и серые шарики – атомы мышьяка (As);

светлые шарики – атомы индия (In)

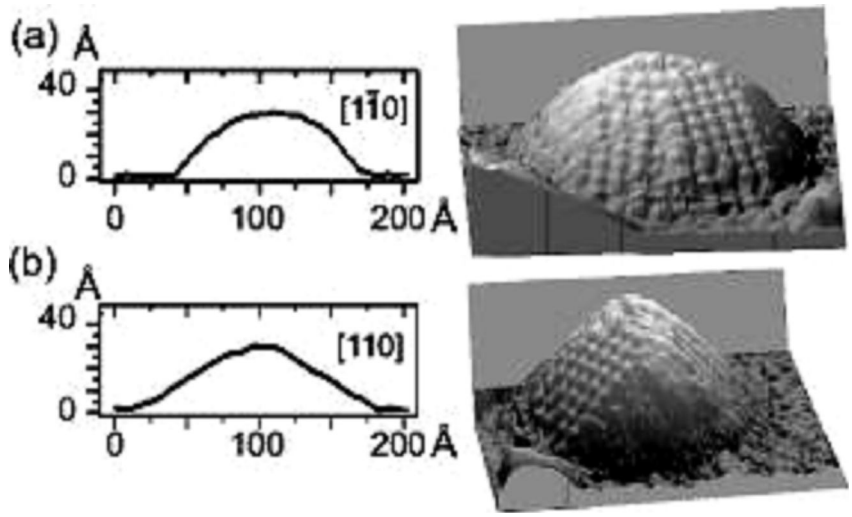


Рис. 2.11,в. Профили (по высоте) квантовой точки и соответствующие трехмерные СТМ-изображения, вид с разных направлений

тическим спектром. Способы получения полупроводниковых квантовых точек весьма различны: они могут создаваться из планарных полупроводниковых гетероструктур с помощью литографии, могут получаться химическими методами. Наиболее широко распространенным способом получения квантовых точек является спонтанное формирование наноразмерных островков-включений одного полупроводникового материала (с меньшей шириной запрещённой зоны) в матрице другого (с большей шириной запрещённой зоны) [44÷47]. Из-за различия ширины запрещённых зон носители заряда оказываются, локализованы в пределах островка, следствием чего и является квазиатомный (представляющий собой набор отдельных уровней) энергетический спектр.

2.3. Схема образования двумерных электронов в гетероструктуре

Прежде чем перейти к методам формирования полупроводниковых элементов нейрочипов, рассмотрим схему образования двумерных струк-

тур. Двумерные электроны образуются на плоской границе контакта двух полупроводников с разной шириной запрещённой зоны – так называемой гетероструктуре. Рассмотрим образование гетероперехода на примере двух полупроводников – GaAs и $Ga_{1-x}Al_xAs$ (ширина запрещённой зоны E_{g2} увеличивается при увеличении x). На рис. 3.12,а представлены зонные диаграммы двух разделённых в пространстве полупроводников разного состава, причём энергия электрона в вакууме выбрана в качестве точки отсчёта. Таким образом, внутри полупроводника энергия электрона понижается, то есть для того, чтобы электрон удалить из полупроводника, необходимо затратить определённую энергию [41÷43, 49].

Когда два различных полупроводника соединяются, у границы их раздела происходит перераспределение электрического заряда и образуется так называемый гетеропереход (рис. 3.12,б). Электрическое поле, создаваемое электронами в арсениде галлия и ионизированными примесями в твёрдом растворе арсенида галлия с алюминием (показаны на рис. 3.12,б светлыми кружками), приводит к изгибу зон, и в возникающей квантовой яме образуются несколько уровней энергии.

Характерный размер потенциальной ямы в GaAs в направлении, перпендикулярном гетерогранице, порядка или меньше длины волны де Бройля для электронов в данном полупроводнике, поэтому движение электронов в этом направлении квантовано. При этом электроны могут свободно двигаться вдоль границы раздела материалов, то есть ведут себя как двумерные. Типичной является гетероструктура $GaAs/Ga_{1-x}Al_xAs$. Ширина запрещённой зоны E_g в GaAs составляет 1,52 эВ. При добавлении Al величина E_g растёт. Для стандартной гетероструктуры при концентрации алюминия x = 0,3 разность запрещённых зон составляет – 0,4 эВ. На границе возникает скачок потенциала, – 60% которого приходится на зону проводимости и – 40% — на валентную зону [46].

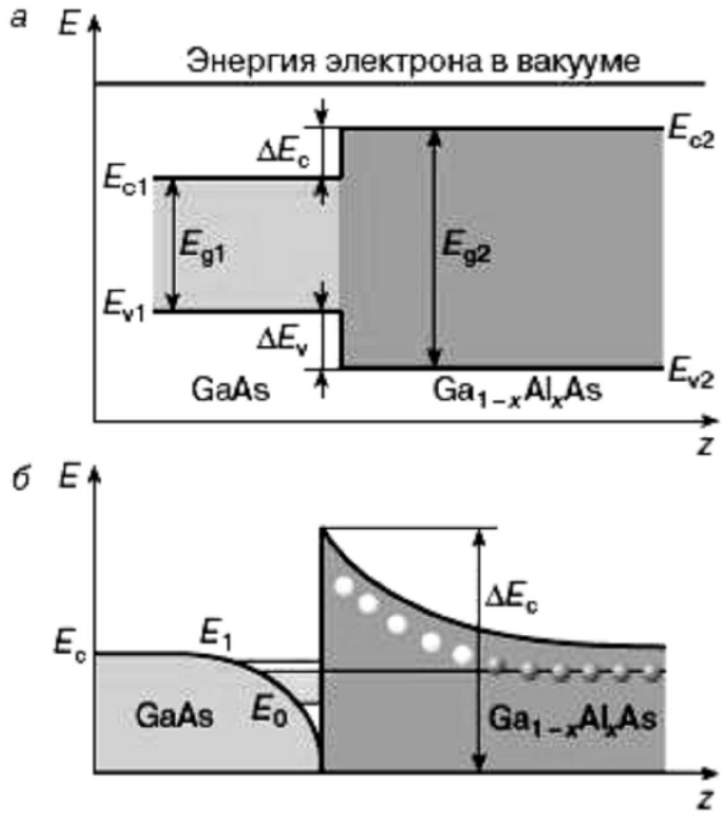


Рис. 2.12. Зонная диаграмма двух различных полупроводниковых материалов и профиль дна зоны проводимости гетероперехода. Индексы 1 и 2 относятся к GaAs и $Ga_{1-x}Al_xAs$ соответственно. Все энергии отсчитываются от уровня энергии электрона в вакууме. Двумерные электроны в гетеропереходе заштрихованы; светлые кружки – ионизированные, тёмные – неионизированные примеси:

а – зонная диаграмма двух различных полупроводниковых материалов (GaAs и $Ga_{1-x}Al_xAs$)
E_0 – дно зоны проводимости, E_v – потолок валентной зоны,
E_g – ширина запрещённой зоны;

б – профиль дна зоны проводимости E_0 – гетероперехода
ΔE_0 – разрыв зоны проводимости,
E_0 и E_1 – уровни размерного квантования

В настоящее время гетероструктуры созданы в самых различных полупроводниках и полупроводниковых соединениях Ge/Si, InAs/GaAs и т.д.

2.4. Теоретический подход к росту твёрдотельных объектов как элементов нейронной сети

Рассмотрим результаты исследований свойств самоорганизованных квантовых точек $Si_{1-x}Ge_x$ (x = 0;3), сформулированных методом ионного синтеза.

В пластины кристаллического кремния ориентации (111) имплантировали ионы германия $^{74}Ge^{+}$ на сильноточном ускорителе SCI-218 «BALZERS». Дозы имплантации составили $D = 5\cdot10^{16}$, $1\cdot10^{17}$ см$^{-2}$, энергия ионов 50 кэВ. Для предотвращения эффектов каналирования падающий на кремниевую подложку ионный поток направляли с отклонением 7º от нормального падения. После имплантации образцы подвергались фотонному импульсному отжигу при температуре 900ºС в атмосфере азота в течении 3 с. В результате подобного воздействия в тонком слое твёрдого раствора SiGe удалось сформировать области с повышенной концентрацией атомов Ge, протяжённость которых составляла несколько десятков нм и высота до 10 нм (наноразмерные структуры) [38÷42].

На электронном оже-спектрометре (ЭОС) PHI-680 фирмы Physical Electronics (США) проводились исследования локального элементного состава структур, а так же оценивались геометрические размеры структур и пространственное расположение квантовых точек в приповерхностной области. Ускоряющее напряжение первичных электронов составляло 10 кэВ, ток – 10 нА, диаметр первичного пучка 15÷20 нм, а глубина анализа не более 5 нм. Топографию поверхности изучали на атомно-силовом микроскопе (АСМ) Solver-47 фирмы НТ-МДТ. Для исследования формы наноразмерных структур и элементного состава поверхность образцов обрабаты-

вали раствором КОН (33%) в течении 25 с при 100º С, что позволило выявить области с максимальным содержанием германия [49].

Имплантация проводилась ионами Ge^+ с дозой $D = 10^{16}$ см $^{-2}$ в двух режимах:

1) с энергией ионов 50 кэВ, проецированный пробег при этом был равен R_p = 35,5 нм, а толщина скрытого слоя ΔR_p = 13 нм;

2) с энергией ионов 150 кэВ, R_p = 89 нм, ΔR_p = 30,6 нм.

Отжиг проводился при ≈1000° С в течении 15 минут [45÷47,49].

Ионы Si с энергией 140 кэВ имплантировали в слои SiO_2 толщиной 0,6 мкм, выращенные термически на кремниевых подложках. Плотность ионного тока не превышала 5мкА/см^2. Ионный синтез проводился в трёх вариантах так, чтобы во всех случаях сохранить одну и ту же дозу и сопоставимые термические бюджеты отжига. Таким образом, имелись образцы трёх типов, полученные в следующих режимах:

1) доза 10^{17} см $^{-2}$ с последующим однократным отжигом при 1100º С в течение 2 ч;

2) доза $5\cdot10^{16}$ см $^{-2}$ с последующим отжигом при 1100º С в течение 1 ч, и затем эта процедура повторялась ещё раз;

3) доза $3{,}3\cdot10^{16}$ см $^{-2}$ с последующим отжигом при 1100º С в течение 40 мин, и затем эта процедура повторялась дважды.

Все отжиги проводились в атмосфере азота. Согласно расчетам пробегов ионов по программе TRIM-95 для дозы 10^{17} см $^{-2}$ в максимуме распределения концентрация избыточных атомов Si составляла 10 ат% .

Образцы исследовались методами фотолюминесценции (ФЛ), рамановского рассеяния и высокоразрешающей электронной микроскопии на поперечных срезах. Для возбуждения ФЛ использовался азотный лазер с длиной волны излучения λ = 337 нм, а регистрация проводилась с помощью фотоумножителя ФЭУ-79. Все спектры нормировались на спектральную чувствительность аппаратуры. Рамановское рассеяние возбуждалось

излучением аргонового лазера с λ = 514 нм. Для снижения сигнала от кремниевой подложки была выбрана квазиобратная геометрия рассеяния $Z(XX)Z$, где Z – направление (001), X – направление (100). Спектры как рамановского рассеяния, так и ФЛ снимались при комнатной температуре. Поперечные срезы готовили по стандартной методике, а электронно-микроскопические исследования были проведены на микроскопе JEM-4000EX фирмы JEOL [49].

На рис. 2.13 показаны спектры рамановского рассеяния от образцов, полученных при трёх режимах ионно-лучевого синтеза нанопреципитатов. После имплантации полной дозы ионов Si и отжига вблизи полосы 520 см $^{-1}$, обусловленной рассеянием от кристаллической кремниевой подложки, появлялся чётко выраженный дополнительный пик с максимумом около 510 см $^{-1}$. Он свидетельствует об образовании нанокристаллов Si. Кроме того, просматривается слабая широкая полоса рассеяния в области с центром вблизи 480 см $^{-1}$, где рассеивают связи Si–Si аморфного кремния. Переход на режим имплантации с одним промежуточным отжигом существенно меняет спектр. Интенсивность дополнительного пика сильно понижается, а его максимум смещается в длинноволновую область $k \sim 507$ см $^{-1}$. Отмеченные тенденции в ещё большей степени проявились после ионного синтеза с двумя промежуточными отжигами. Как видно из рис. 2.13, дополнительное рассеяние, присущее кремниевым квантово-размерным кристаллам, практически полностью исчезает. Существует лишь некоторый намёк на дополнительное рассеяние около 504 см $^{-1}$, но его интенсивность сопоставима с шумами [38÷40,49].

По данным высокоразрешающей электронной микроскопии, на поперечном срезе однократное введение дозы 10^{17} см $^{-2}$ приводит после отжига к образованию кремниевых нанопреципитатов, у которых выявляется кристаллическая структура (рис. 2.14,а). Размеры нанокристаллов составляют 4÷5 нм, а плотность 10^{11}÷10^{12} см $^{-2}$. Если доза набиралась с промежу-

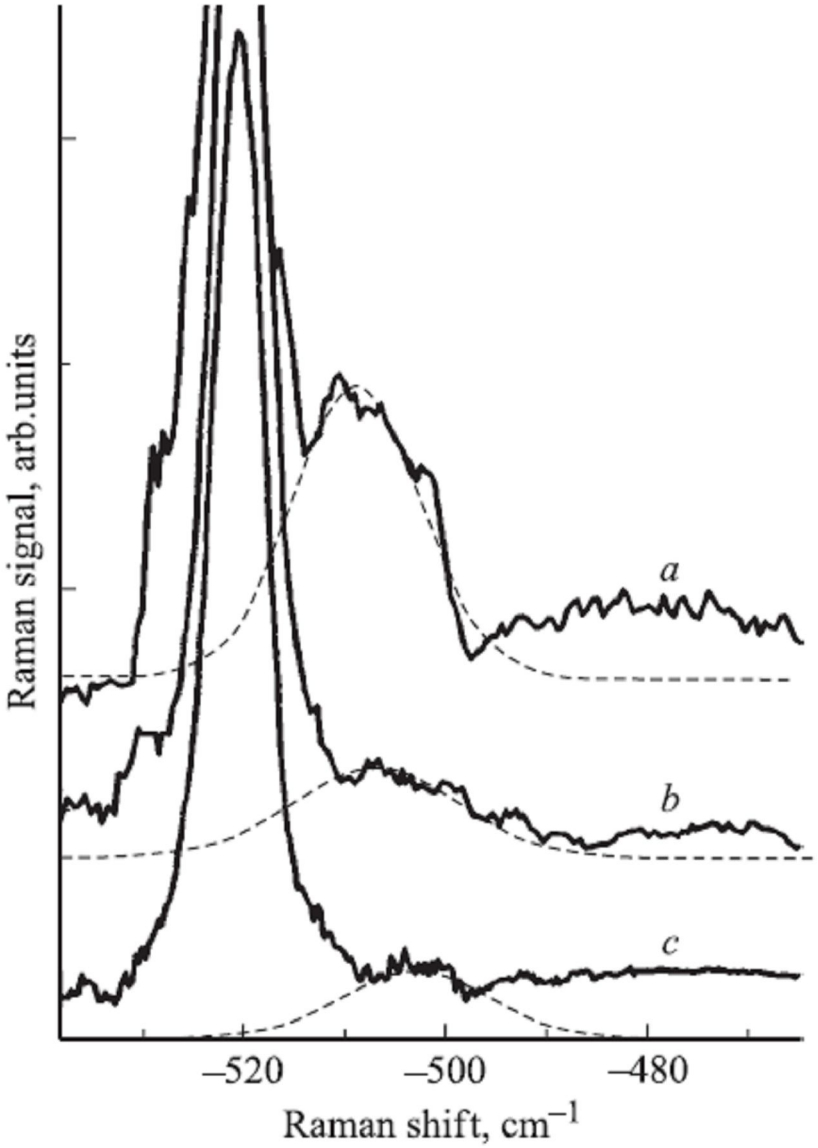


Рис. 2.13. Спектры рамановского рассеяния образцов, полученные в режимах 1 (*a*), 2 (*b*) и 3 (*c*)

точными отжигами, в SiO_2 были видны нанопреципитаты в виде тёмных пятен на изображении скола. Выявить в них признаки кристаллической структуры не удается. Подобные пятна наблюдались ранее неоднократно разными исследователями, когда условия синтеза оказывались недостаточными для формирования различимых нанокристаллов. Промежуточные отжиги приводят к уменьшению средних размеров преципитатов до 3÷4 нм и к некоторому снижению их концентрации (рис. 2.14,b, c). Делать здесь

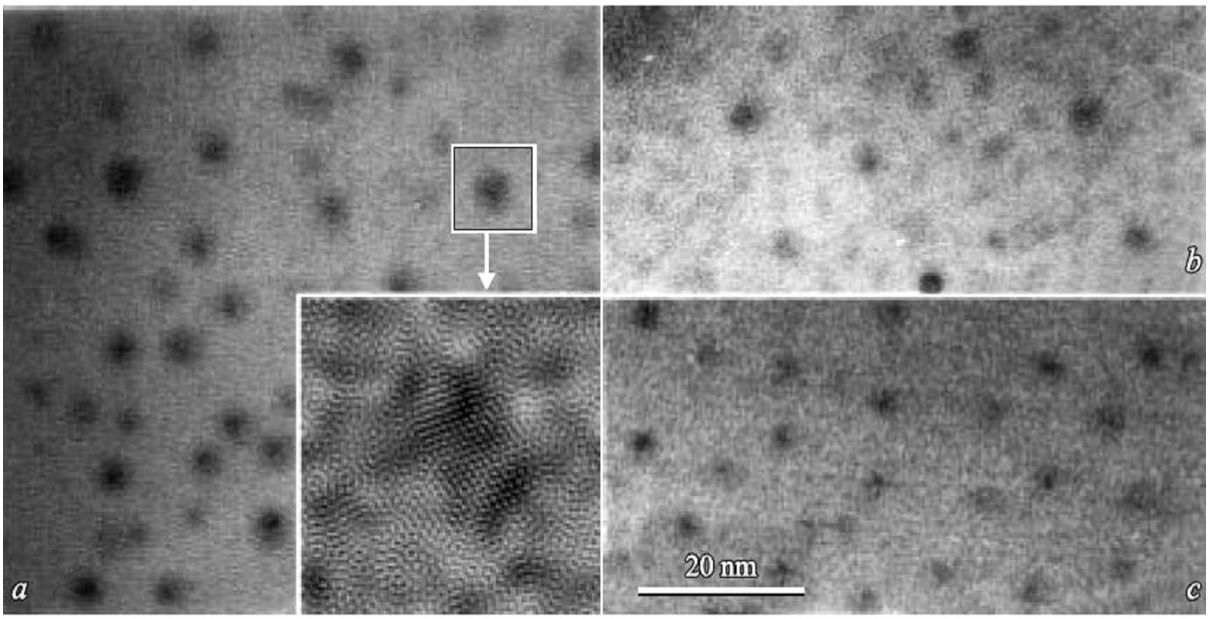


Рис. 2.14. Электронная микроскопия высокого разрешения на поперечных срезах образцов, полученных в режимах 1 (*a*), 2 (*b*) и 3 (*c*). Режим *a* – после Фурье-фильтрации на выделенном участке выявляется кристалличность включений

какие-либо количественные сравнения затруднительно из-за малости площади обзора.

Спектры ФЛ после ионно-лучевого синтеза в каждом из трёх режимов представлены на рис. 2.15.

В отличие от данных по рамановскому рассеянию и электронной микроскопии, где при использовании промежуточных отжигов существенно ослаблялись признаки присутствия кремниевых нанокристаллов, их люминесценция оказалась затронута в меньшей степени. В случае имплантации полной дозы с последующим отжигом в спектре возникала интенсивная полоса с максимумом вблизи 795 нм. В настоящее время практически все исследователи связывают её с излучательной рекомбинацией в образующихся квантово-размерных кристаллах кремния. Имплантация с

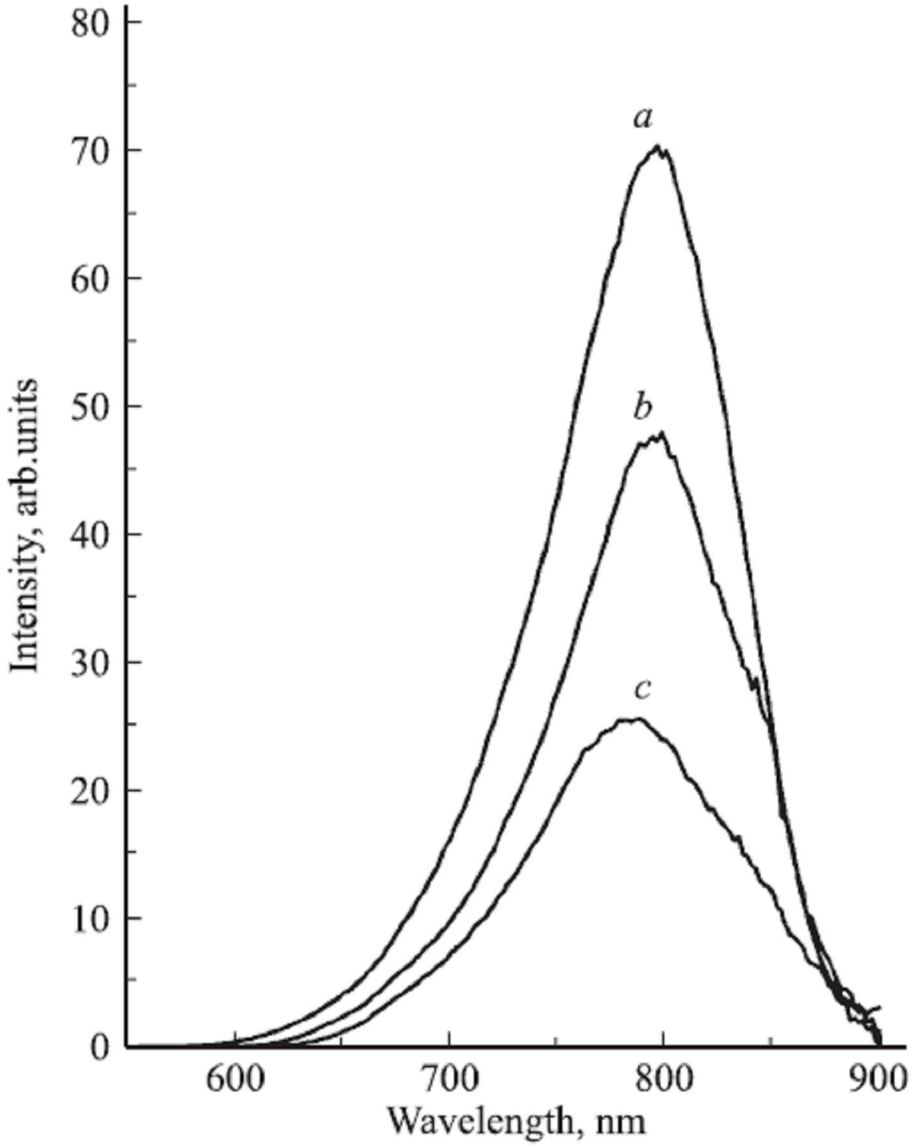


Рис. 2.15. Спектры фотолюминесценции образцов, полученных в режимах 1 (*a*), 2 (*b*) и 3 (*c*)

одним промежуточным отжигом приводила к некоторому снижению интенсивности свечения, причём заметного смещения максимума не происходило. Набор дозы10^{17} см $^{-2}$ в три приёма ведет к дальнейшему понижению интенсивности ФЛ, и здесь уже становится заметным коротковолновое смещение максимума полосы к ~ 785 нм. Таким образом, использование двух промежуточных отжигов снижало интенсивность ФЛ всего в 2 с небольшим раза, в то время как возможности обнаружения признаков на-

нокристаллов по рамановскому рассеянию или с помощью высокоразрешающей электронной микроскопии практически исчерпывались.

Данный механизм создания области затвор-диэлектрик, состоящий из управляющего оксида (верхний слой), слоя оксида с захороненными нанокристаллами германия (средний слой) и туннельного оксида (нижний слой) позволяет очень точно контролировать толщину оксидов окружающих захороненный слой нанокристаллов. В первую очередь на поверхности кремниевой подложки создается тонкий эпитаксиальный слой $Si_{1-x}Ge_x$ с последующим высокотемпературным влажным окислением, которое заставляет имплантированный Ge скапливаться на границе раздела Si/SiO_2 (вследствие отделения растущего оксида в данных условиях окисления). После удаления слоя высокотемпературного оксида верхний слой оксида выращивается при 800º С в атмосфере сухого O_2. В таких условиях Ge не взаимодействует с растущим оксидом. Затем слой $Si_{1-x}Ge_xO_2$ выращивается при 800º С во влажной среде таким образом, что Ge проникает в область затвор-диэлектрик. На заключительном этапе выращивается туннельный оксид при 800º С в атмосфере сухого O_2, а затем структура отжигается при температуре 900º С, что приводит к образованию преципитатов Ge и формированию захороненного слоя нанокристаллов.

Схема эксперимента и основные результаты представлены на рис. 2.16 [48,49].

Элементарные ячейки нейронной сети могут хранить один бит информации и состоят из одного полевого нанотранзистора с электрически изолированной областью (плавающим затвором – floating gate), способного хранить заряд многие годы. Ячейка нейронной сети представляет собой одиночный многоходовой МОП транзистор (рис. 2.17). Проводящий поликремниевый слой находится между внешне доступным затвором (обозначенный как управляющий затвор) и плавающим затвором. Диэлектрики между плавающим затвором и подложкой (термический оксид кремния), а

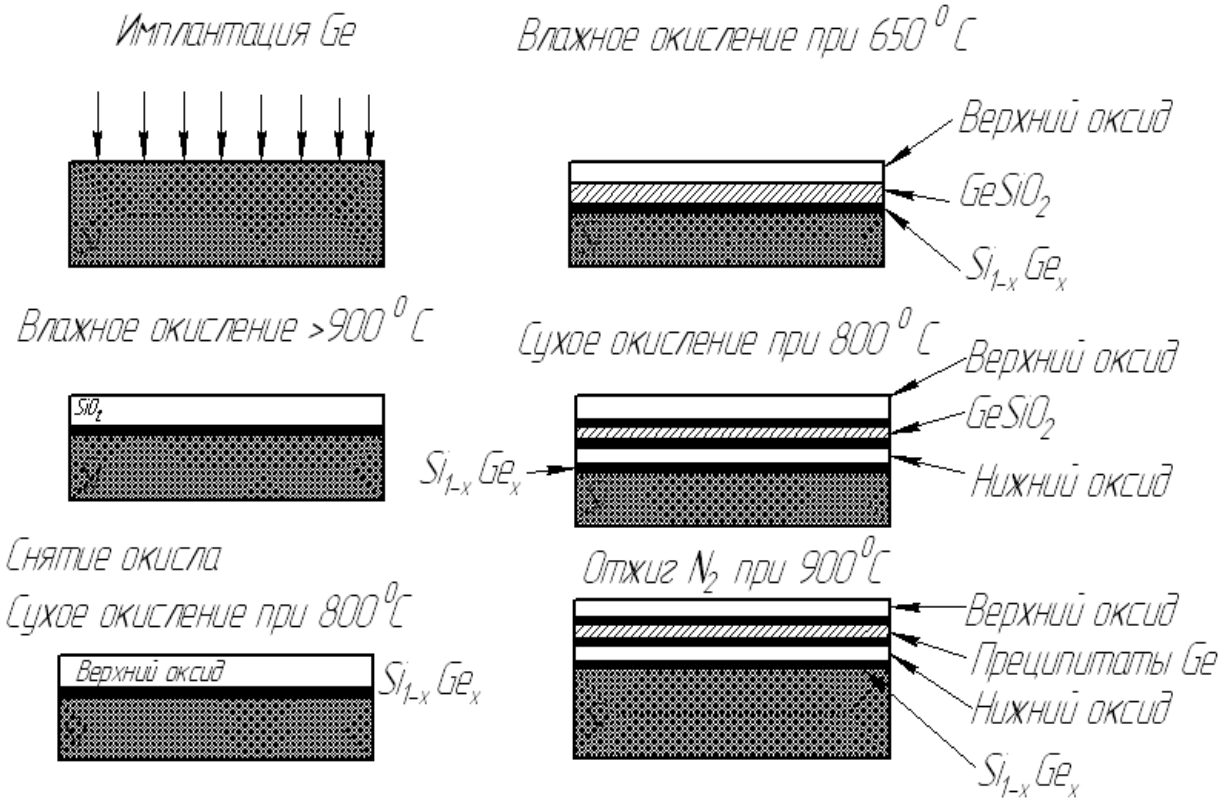


Рис. 2.16. Формирование слоя нанокристаллов Ge в подзатворном диэлектрике для хранения заряда в элементах нейронной сети

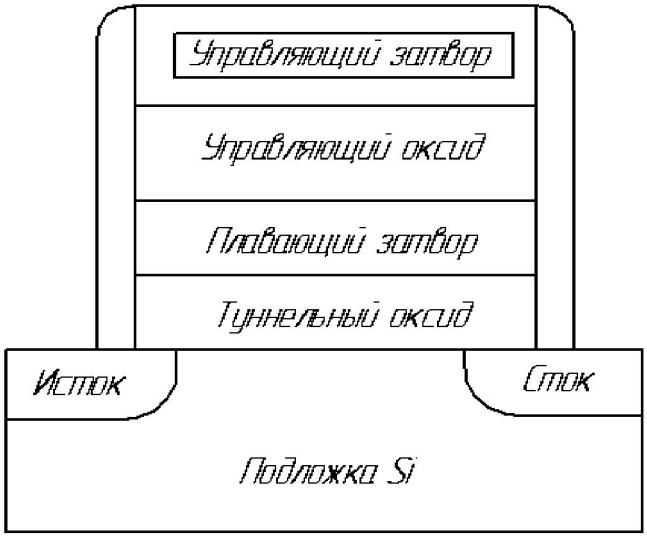


Рис. 2.17. Схематическое сечение МОП транзистора с плавающим затвором

так же между плавающим затвором и контрольным входом являются туннельными и управляющими диэлектриками соответственно.

Наличие или отсутствие заряда на плавающем затворе кодирует один бит информации. При записи заряд помещается на плавающий затвор одним из двух методов (зависит от типа ячейки): методом инжекции электронов или методом туннелирования электронов. Стирание содержимого ячейки (снятие заряда с плавающего затвора) производится методом туннелирования. Как правило, наличие заряда на транзисторе понимается как логический «ноль», а его отсутствие – как логическая «единица».

Изменение порогового напряжения ΔF_{TH} , вызванное хранением заряда Q_{FG} определяется как

$$\Delta F_{TH} = \frac{-Q_{FG}}{C_{CG}} \ ,$$

где $C_{C\,G}$ – емкость между контрольным и изолированным входом и задается формулой

$$C_{CG} = \varepsilon \cdot \frac{A}{t} \ ,$$

где A – площадь конденсатора, а ε и t – диэлектрическая константа и толщина управляющего диэлектрика соответственно (рис. 2.18).

На рис. 2.19 представлена схематическая диаграмма энергетических зон для кремниевой нанокристаллической памяти в процессе: (а) записи (инжекция электронов), и (б) стирания (экстракция электронов). В процессе записи (стирания) электроны инжектируются (экстрагируются) в/из нанокристаллы за счёт приложенного положительного (отрицательного) напряжения смещения на затвор по отношению к истоку и стоку. Толщина управляющего оксида должна быть относительно мала для выбора приемлемого низкого напряжения, но не такой маленькой, чтобы привести к утечке заряда в управляющий затвор. Эти ограничения приводят к выбору

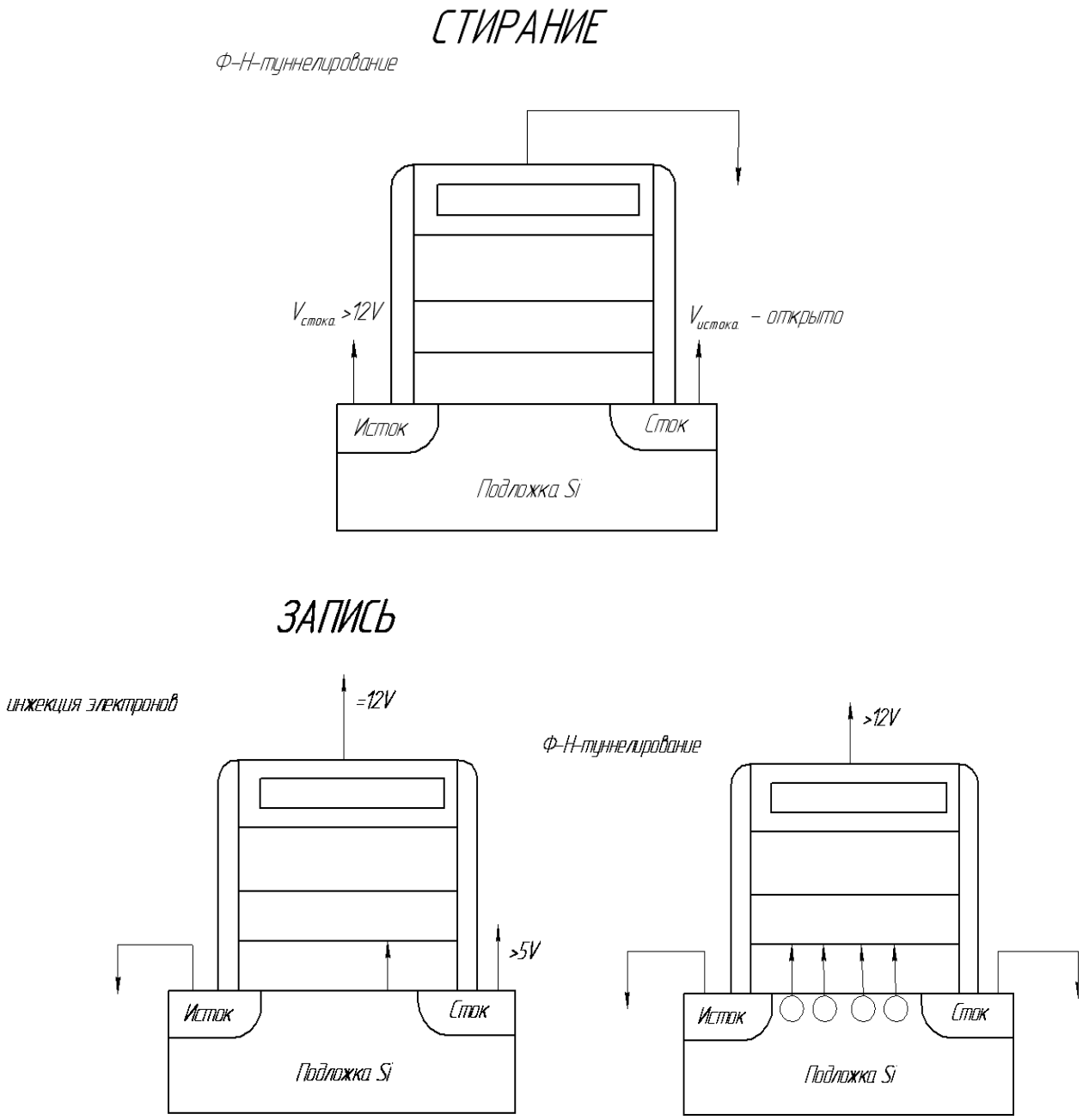


Рис. 2.18. Схематическое описание операций стирания/записи для ячейки элемента нейронной сети

оптимальной толщины управляющего оксида в 5÷15 нм. В случае тонкого туннельного оксида толщиной менее 3 нм передача заряда осуществляется через прямое туннелирование (потока электронов через весь оксид) вместо туннелирования по механизму Фаулера-Нордгейма. Генерация носителей с

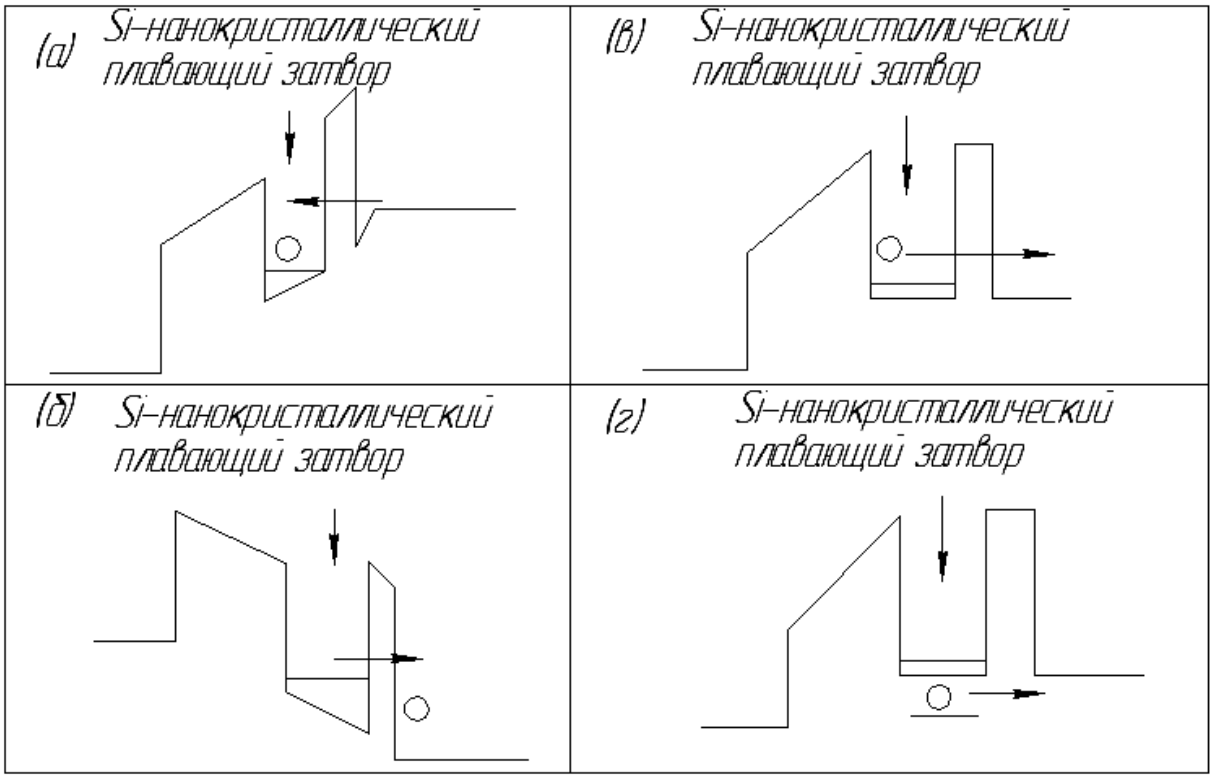


Рис. 2.19. Диаграммы потенциала зоны проводимости для элементов нейронной сети на основе кремниевых нанокристаллов во время:
(а) записи (инжекции электронов в нанокристалл);
(б) стирания (экстракция электронов из нанокристалла);
(в) хранение электронов на квантовыхуровнях энергии нанокристалла;
(г) хранение электронов на низких энергетических уровнях в нанокристалле и/или на границе раздела SiO/нанокристалл

низкими энергиями (с энергиями много меньше чем 3 эВ, что является порогом для некоторых главных механизмов деградации оксидов от горячих носителей заряда) во время операции программирования, понижает деградацию оксида во время Ф-Н-инжекции, в результате улучшая износоустойчивость и характеристику заряд/пробой [49], что может быть реализовано, например, с помощью баллистического транзистора (рис. 2.20).

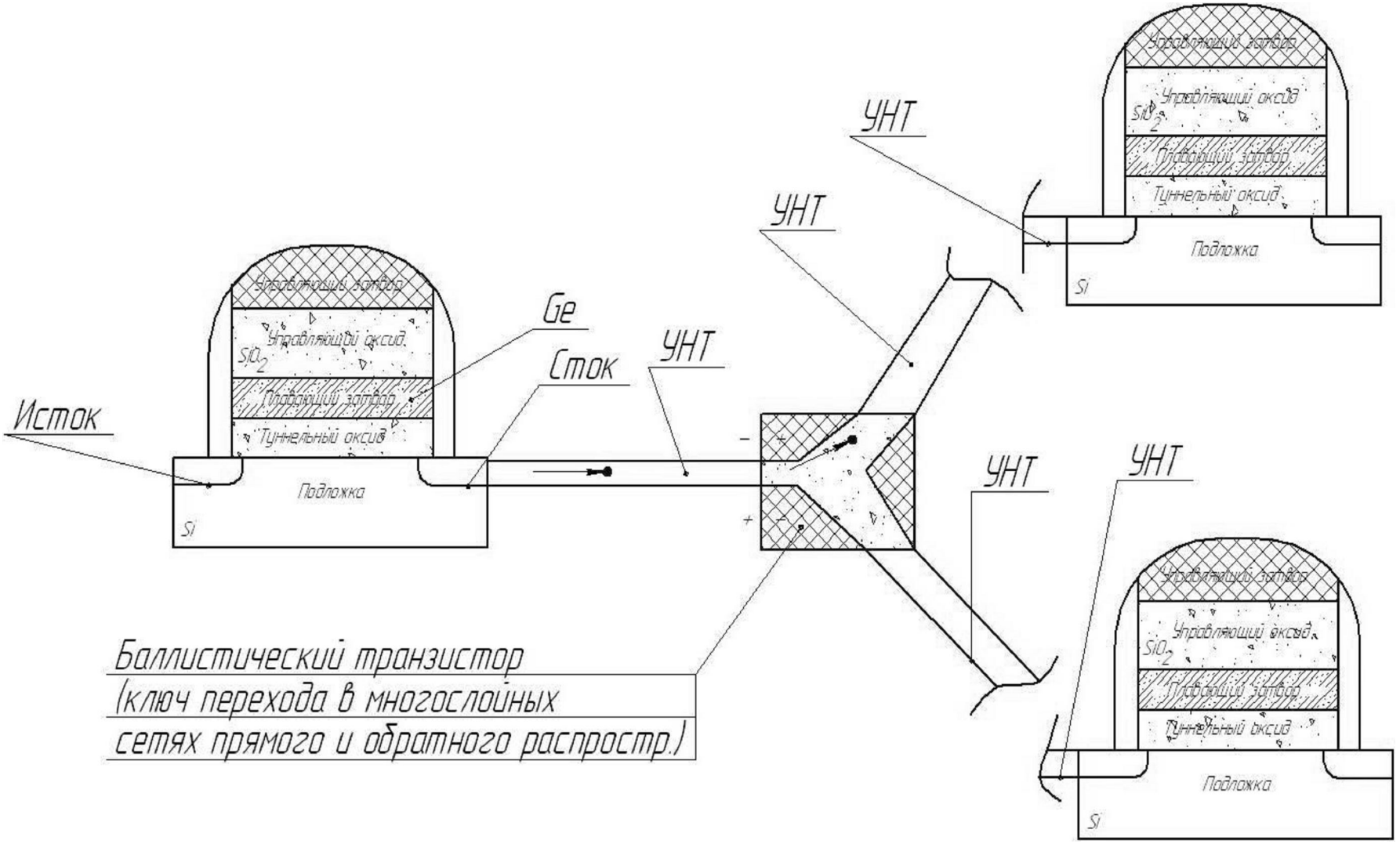


Рис. 2.20. Наноячейки памяти с архитектурой нейронных сетей реализованной с помощью баллистического транзистора

Одно из главных преимуществ ячеек нейронной сети на нанокристаллах, по сравнению с обычными устройствами на основе плавающих затворов, состоит в использовании взаимно изолированных узлов хранения заряда, вместо непрерывного поликремниевого слоя. Такой неоднородный плавающий затвор уменьшает потерю зарядов через дефекты в окисле, где происходит туннелирование, позволяя все более уменьшать толщину туннельного окисла.

ГЛАВА 3. ПОСТРОЕНИЕ ФИЗИКО-МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПРИ ПРОЕКТИРОВАНИИ ЭЛЕМЕНТОВ НЕЙРОННЫХ СЕТЕЙ

3.1. Модели ускоренного обучения нейронных сетей

Искусственные нейронные сети находят все более широкое применение в решении задач искусственного интеллекта. Они доказали свою способность решать такие задачи как ассоциативный поиск, кластирование и распознавание образов.

Схема нейронной сети представлена на рис. 3.1.

Каждый нейрон сети (рис. 3.2) умножает первый входной сигнал x_i на вес входа k_{ij}, складывает полученную величину с сигналом, поступаю-

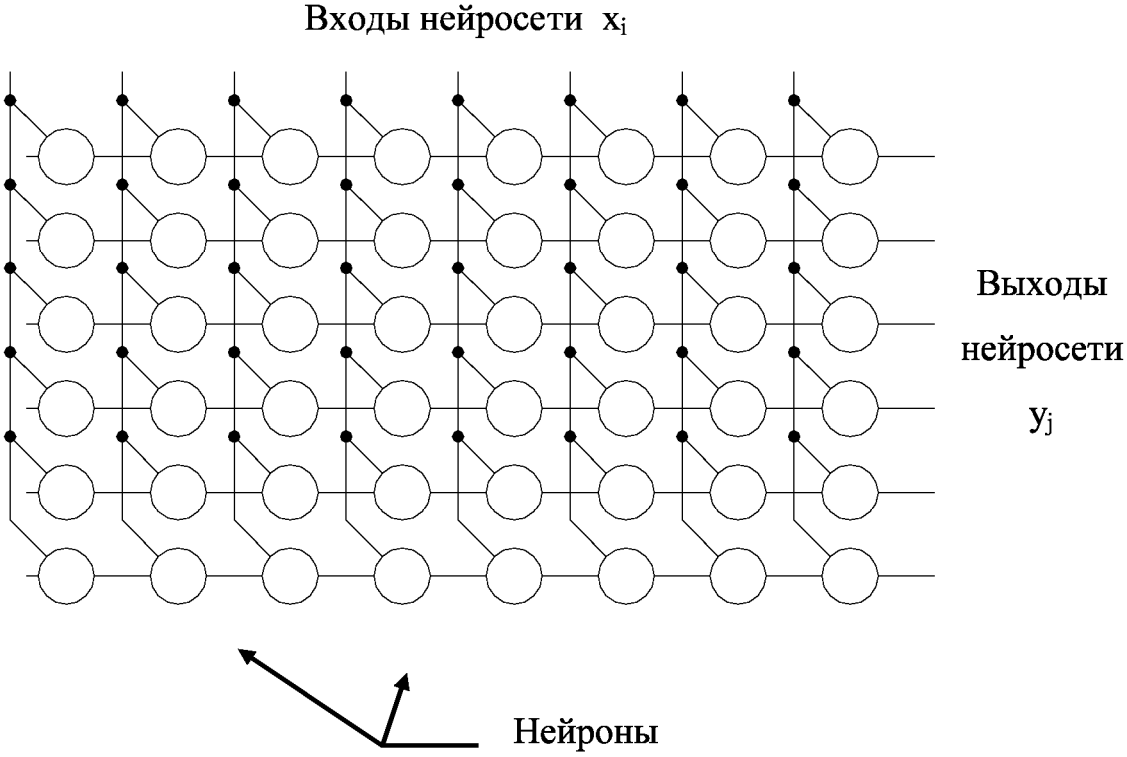


Рис. 3.1. Схема плоской нейронной сети

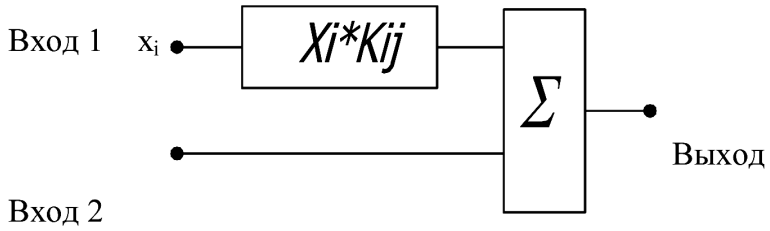


Рис. 3.2. Структурная схема нейрона

щим на второй вход нейрона, и передает результат на выход нейрона y_j .

Взаимодействие между нейронами происходит за счёт связей, реализованных в двух дополнительных слоях металлизации, что позволяет изготавливать нейронные сети как цифровые микросхемы высокой степени интеграции в рамках существующей планарной технологии.

Входные сигналы x_i могут иметь как бинарный формат (0 или 1), так и формат целого числа. Выходные сигналы y_j имеют формат целого числа. Веса входов k_{ij} так же имеют формат целого числа.

В результате на каждом выходе y_j получается взвешенная сумма входных сигналов

$$y_j = \sum_{i=1}^{n} k_{ij} x_i \ .$$

Веса входов k_{ij} могут быть установлены на основе априорной информации об алгоритме решаемой задачи или получены в процессе обучения.

Электрическая принципиальная схема нейрона представлена на рис. 3.3.

Схема содержит следующие функциональные узлы, представленные моделями в стандарте PSPICE:

1. DD1, DD2 – логические элементы "И-НЕ";

2. DD3, DD4 – четырехразрядные реверсивные двоичные счетчики;

3. DD5 ÷ DD12 – логические элементы "И";

4. DD13 ÷ DD20 – четырёхразрядные сусмматоры.

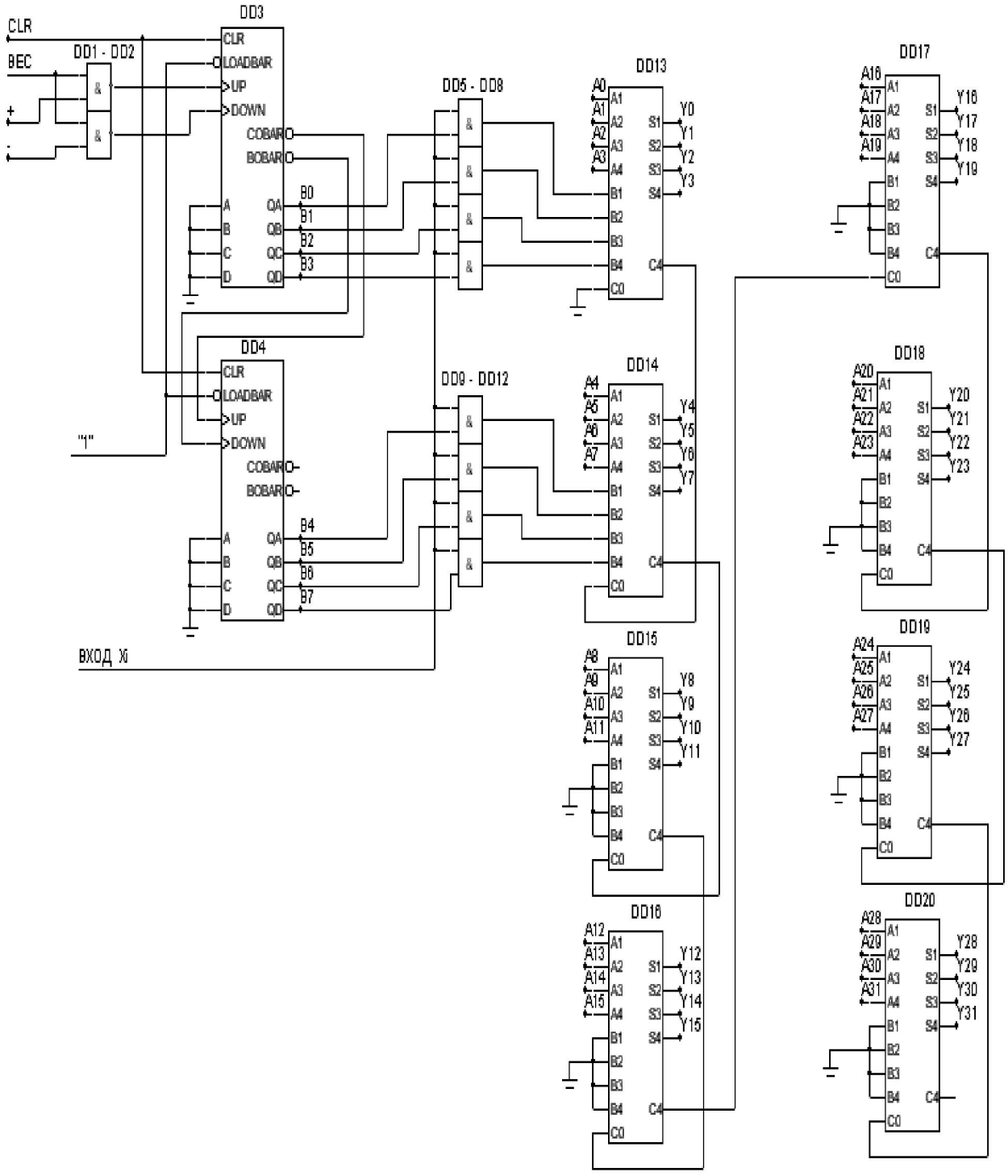


Рис. 3.3. Электрическая принципиальная схема нейрона

Разрядность сумматора – 32 двоичных разряда.

Максимальное число $N = 2^{32} = 4294967296$.

Разрядность весового элемента – 8 двоичных разрядов.

Максимальный вес $M = 2^8 = 256$.

На рисунке:

CLR – вход сброса счётчиков;

ВЕС – вход импульсов изменения веса нейрона;

Число, записанное в счетчик – вес нейрона Q

– – вход управления для уменьшения веса;

+ – вход управления для увеличения веса;

ВХОД Xi – первый вход нейрона (0 или 1);

A0 ÷ A31 – второй суммирующий вход нейрона (32-х разрядная шина);

Y0 ÷ Y31 – выход нейрона (32-х разрядная шина);

В табл. 3.1 представлены основные характеристики процесса функционирования нейрона.

Временная диаграмма работы нейрона показана на рис. 3.4.

На ВХОД подаются 9 импульсов (Вход + = 1, Вход – = 0). В результате в счётчик записывается вес $Q = 9_{10}$. На входы {Aj} подается число $A = 041_{10}$. После подачи на вход Xi числа 1 на выходах {Yj) появляется число $Y = A + Q = 3041 + 9 = 3050_{10}$.

Таблица 3.1

Таблица функционирования нейрона

+	–	X_i	$\{A_j\}$	$\{Y_j\}$
1	0	0	Увеличение веса	
0	1	0	Уменьшение веса	
0	0	0	$\{A_j\}$	$\{Y_j\} = \{A_j\}$
0	0	1	$\{A_j\}$	$\{Y_j\} = \{A_j\} + Q$

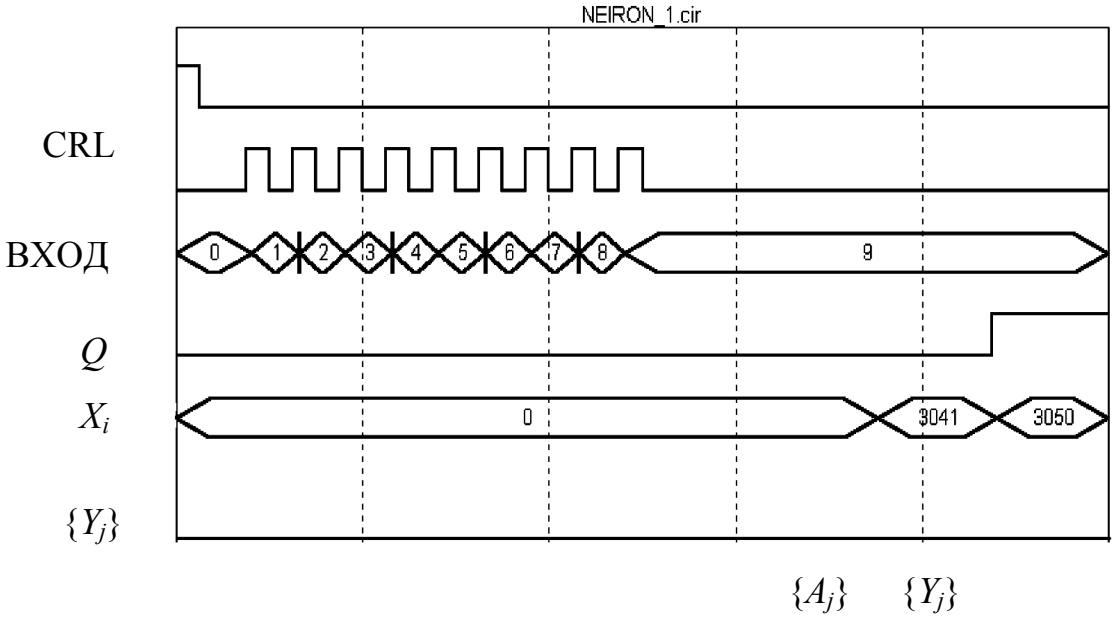


Рис. 3.4. Временная диаграмма работы нейрона

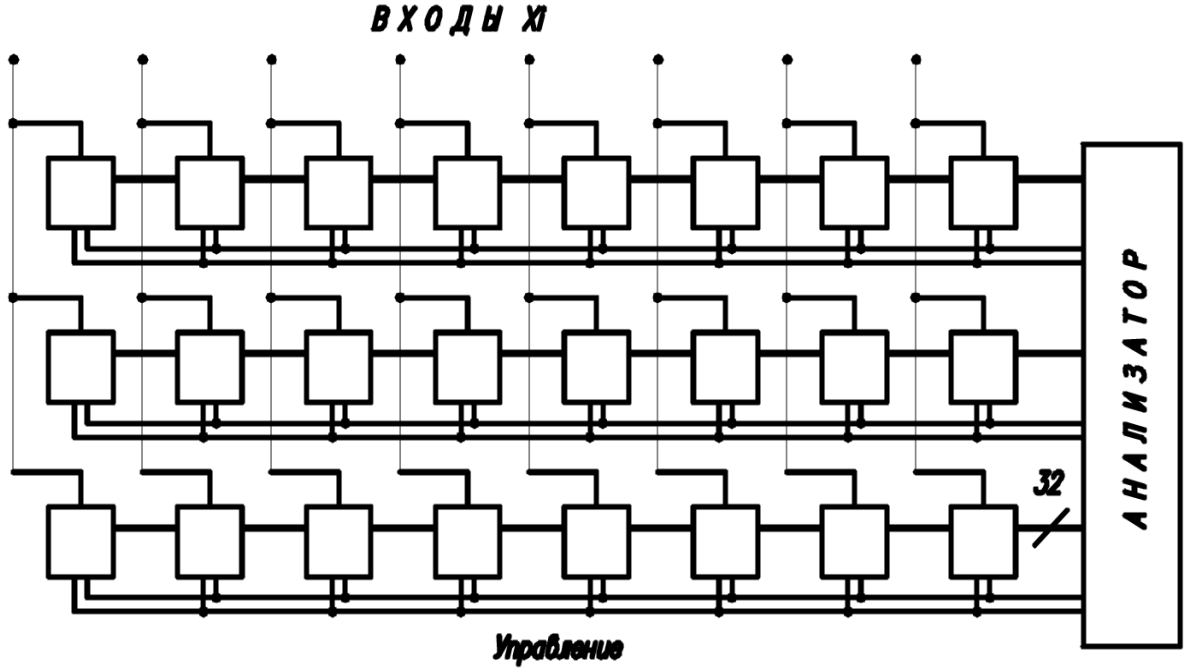


Рис. 3.5. Схема нейросети

Работа нейросети представленной на схеме (рис. 3.5):

■ *Режим обучения.* Изменение весов нейрочипов проводится построчно. В режиме увеличения весов на вход управления одной из строк + пода-

ется лог. 1, на вход подается лог. 0. На входы управления остальных строк подается лог. 0. На часть входов X_i подаются импульсы, увеличивающие вес нейрочипов. В режиме уменьшения весов на вход управления одной из строк подается лог. 1, на вход + подается лог. 0. На входы управления остальных строк подается лог. 0. На часть входов X_i подаются импульсы, уменьшающие вес нейрочипов.

- *Режим работы.* На входы X_i , соответствующие наличию признаков, подается Лог. 1, на остальные входы – Лог. 0. В результате на шине входа анализатора появится число, равное сумме весов нейрочипов строки, на которые подали лог. 1. Анализатор выделяет строку с максимальным числом. Это наиболее похожий образ.

Замечание 1. Сеть является универсальной, так как входы X_i можно группировать в группы и организовывать сложные сигналы на входе.

Замечание 2. В сети все входы соединены с каждым выходом. Это отличие от стандартного персептрона, в котором только часть входов соединена с каждым выходом.

Замечание 3. Можно сеть каскадировать, то есть входы одной сети можно соединить с выходами другой. Так можно построить многослойную сеть.

Результаты моделирования работы нейронной сети проводились на алгоритме написанным с использованием Open Source библиотеки OpenCV 0.99 на языке C++. Смоделированная нейронная сеть работает в режиме ассоциативной памяти и распознавания образов графических изображений в реальном времени, основанная на алгоритме распознавания Хаарта [36].

Моделирование нейронной сети проводилось методом распознавания графических изображений, а также в реальном времени при съёмке видеокамерой. В качестве тестовых изображений были взяты фотографии людей, а так же съёмка людей видеокамерой, где нейронная сеть распознавала лица, и сохраняла данные изображения в отдельно взятой ветви. Рас-

познавание образов (лиц) проводилось после одного тестового обучения, в качестве обучающего примера были указаны характерные размеры взаимного расположения объектов (в данном случае элементы лиц).

При обучении нейронной сети, желательно, чтобы она полностью повторяла обучающую выборку (ОВ), то есть её глобальная ошибка стремилась к нулю. Для построения нейронной сети нужно, чтобы каждый параметр входной выборки был представлен как минимум пятью значениями, равномерно распределёнными по диапазону допустимых для данного параметра значений. Размер выборки определяется количеством входных и выходных значений, характером этих величин, а так же сложностью математической модели, которой описывается данная задача. Изменение числа элементов в промежуточном слое в пределах 10% влияет только на грубость (чувствительность) обучения конечного обученного аппарата. Так же наблюдается незначительное изменение скорости обучения.

На сегодняшний день в распоряжение разработчика предоставлено большое количество различных моделей нейронных сетей и алгоритмов их обучения [45÷47]. И хотя постоянно ведутся научные исследования в области совершенствования существующих и создания новых моделей и обучающих алгоритмов, теория нейронных сетей пока остается слабо формализованной. Однако уже на данном этапе чётко прорисовываются два основных этапа создания нейронного вычислителя: структурный и параметрический синтез. В рамках первого этапа перед разработчиком ставятся задачи: определения модели сети, определение её структуры, выбор алгоритма обучения.

Параметрический синтез включает в себя процессы обучения нейронной сети и верификации полученных результатов. Причём в зависимости от результатов верификации возникает необходимость возврата на различные стадии структурного или параметрического синтеза, таким образом становится очевидной итеративность процесса проектирования ней-

ронного вычислителя [46÷50].

Слабая формализованность этих этапов приводит к тому, что разработчику, проектирующему нейронный вычислитель, приходится сталкиваться с решением некоторых проблем. Например, на этапе структурного синтеза, при проектировании нейровычислителя, решающего нестандартную задачу, приходится прилагать значительные усилия для выбора модели сети, её внутренней структуры и способа обучения. Проблемой параметрического синтеза сетей является трудоёмкость их обучения. Если решать реальные задачи, учитывая все возможные факторы, то время обучения нейронной сети для такой задачи оказывается достаточно продолжительным. Но при решении некоторых задач требуется затратить как можно меньше времени на этот процесс, например, такой как работа в реальном масштабе времени.

Данная работа ставит своей задачей предложить возможные методы уменьшения времени, затрачиваемого на обучение многослойных нейронных сетей с обратным распространением ошибки. В качестве таких методов предлагаются: управление процедурами изменения и вычисления весовых коэффициентов, реорганизация объектов в распознаваемых классах. Были предложены два возможных пути решения этой задачи. Первый основывался на выборе определённого функционального базиса нейронной сети. Второй метод управлял значением шага изменения весов сети, рассматривая его с точки зрения центробежной силы и, корректируя его таким образом, чтобы его вектор всегда был направлен на оптимум множества весовых коэффициентов [48÷54].

Рассмотрим поставленную задачу с точки зрения переобучения нейронной сети.

В большинстве случаев нейронную сеть обучают, пока её ошибка не станет равной нулю. Это приводит порой к неоправданным затратам драгоценных ресурсов времени, хотя для решения большей части задач *доста-*

точно, чтобы эта ошибка не превышала определённого значения [36÷52].

Иногда степень *достаточности* определяется исходя из условий задачи и искомого результата. Однако, в большинстве случаев этот процесс протекает на интуитивном уровне и руководствующий принцип не фиксируется сознанием в достаточной мере. На самом деле, этот момент является одним из самых важных в решении задач подобного типа, и оптимальное значение варьируемого параметра может зависеть от многих исходных величин и ограничений накладываемых на решение задачи. Таким образом, появляется необходимость в формализации данного принципа, в дальнейшем – *ПринципаДостаточности (ПД).*

Обучение с учётом ПРИНЦИПА ДОСТАТОЧНОСТИ

Нейронные сети используются для решения ряда задач искусственного интеллекта. Рассмотрим обучение многослойной нейронной сети с обратным распространением ошибки в рамках решения задач классификации [49].

В процессе обучения нейронных сетей, в числе прочих, можно выделить два вида ошибки, назовём их глобальной и локальной.

Выражение для вычисления локальной ошибки имеет следующий вид

$$E_i = \sqrt{\frac{\sum_{1}^{m} e_k^2}{m}} \; ; \qquad (3.1)$$

$$e_k = Y_k - A_k \; ,$$

где e_k – элементарная ошибка k-го нейрона выходного слоя сети; m – число нейронов в выходном слое сети.

Глобальная ошибка сети вычисляется по следующей формуле

$$E = \sqrt{\frac{\sum_{1}^{n} E_i^2}{n}}\ , \qquad (3.2)$$

где E_i – локальная ошибка нейронной сети на i-м наборе; n – число обучающих наборов.

Идеально обученной считается такая сеть, которая полностью повторяет ОВ [50], то есть её глобальная ошибка равна нулю. Но обучение нейронной сети до такой степени представляет собой очень трудоёмкую задачу, а нередко и вовсе неразрешимую. Эти трудности обычно связаны с тем, что разные классы имеют похожие объекты, и чем таких объектов больше и чем более они похожи, тем труднее будет обучить нейронную сеть.

Суть ПД заключается в отказе от обязательного стремления к Идеалу при поиске решения конкретной задачи. Рассматривая эту проблему с точки зрения ПД в рамках глобальной и локальной ошибки, можно сказать, что далеко не всегда необходима 100%-ая точность распознавания. Иногда, для того чтобы отнести исследуемый объект к заданному классу, достаточно, чтобы ошибка сети на данном наборе не превышала некоторого δ.

Минимальное значение δ зависит от характера обучающей выборки. В качестве параметров характеризующих ОВ рассмотрим её полноту, равномерность и противоречивость.

Полнота выборки характеризуется обеспеченностью классов обучающими наборами. Количество обучающих наборов для класса должно быть в 3÷5 раз больше, чем используемое в наборе число признаков класса.

Пусть величина, характеризующая полноту выборки, вычисляется следующим образом:

$$F_{OB} = \frac{N_F}{N} \cdot 100\%\ , \qquad (3.3)$$

где N_F – число классов удовлетворяющих указанному условию; N – общее

число классов.

Равномерность ОВ показывает, насколько равномерно распределены обучающие наборы по классам. Для её вычисления рассмотрим величину $[C_i]$ – количество обучающих наборов для класса i. Тогда среднее отклонение этой величины по выборке для данного класса будет вычисляться по формуле

$$\overline{\Delta}_{C_i} = \sqrt{\frac{\sum_{1}^{k}\left([C_i]-[C_k]\right)^2}{k-1}}\ ; \qquad k \neq i\ , \tag{3.4}$$

где k – количество классов.

Вычислим математические ожидания для величин $\overline{\Delta}_{C_i}$ и $[C_i]$ при условии, что они равновероятны и назовём их соответственно R_Δ и N_{cp}:

$$\begin{aligned} R_\Delta &= \frac{\sum_{1}^{k}\overline{\Delta}_{C_i}}{k}\ ; \\ N_{cp} &= \frac{\sum_{1}^{k}[C_k]}{k}\ . \end{aligned} \tag{3.5}$$

Тогда равномерность выборки будет вычисляться по формуле

$$R_{OB} = 1 - \frac{R_\Delta}{N_{cp}}\ . \tag{3.6}$$

Противоречивость – как процент противоречивых наборов в ОВ может быть представлена в виде

$$C_{OB} = \frac{N_{ПР}}{N}\ , \tag{3.7}$$

где $N_{пр}$ – число противоречивых наборов; N – общее число наборов в ОВ.

Очевидно, что чем меньше противоречивость ОВ и выше её равномерность, тем уже может быть интервал δ.

Однако, в процессе обучения объекты классов, попадая в интервал δ, ложатся неравноудалённо от Эталона класса (рис. 3.6,а). Дифференцирование этих ситуаций позволит улучшить качество обучения сети, поскольку позволит корректировать веса с учётом удаленности реакции сети от эталонной. В данном случае, расстояние до эталона будет определять величину градиента изменения веса. Для этого необходимо либо разбить область δ на отрезки и каждому из них поставить в соответствие значение градиента (рис. 3.6,б), либо задать на этом интервале функцию $a(t) = F(x)$ (рис. 3.6,в) [51÷56].

Таким образом, предполагалось уменьшить число итераций обучения нейронной сети при заданной точности распознавания элементов выборки.

Результатом применения этого метода стало то, что функция ошибки сети E из колебательной становилась фактически монотонно убывающей.

В оригинальном варианте алгоритма обратного распространения ошибки [50] изменения весовых коэффициентов, для пары нейронов i, j (рис. 3.7), выглядят следующим образом:

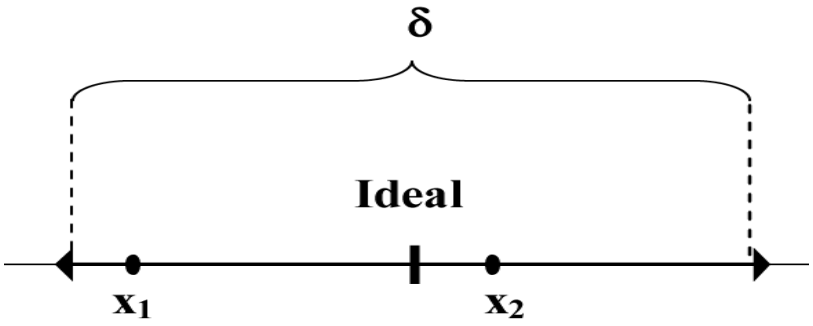
$$W_{ij}^{t+1} = W_{ij}^{t} + \alpha \cdot E_j \cdot A_i^t \ ,$$

где E_j – ошибка j-го нейрона;

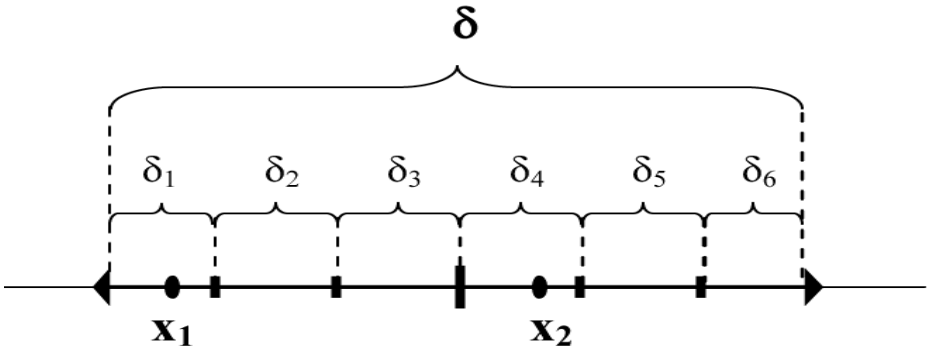
A_i – уровень активации i-го нейрона;

α – шаг изменения веса.

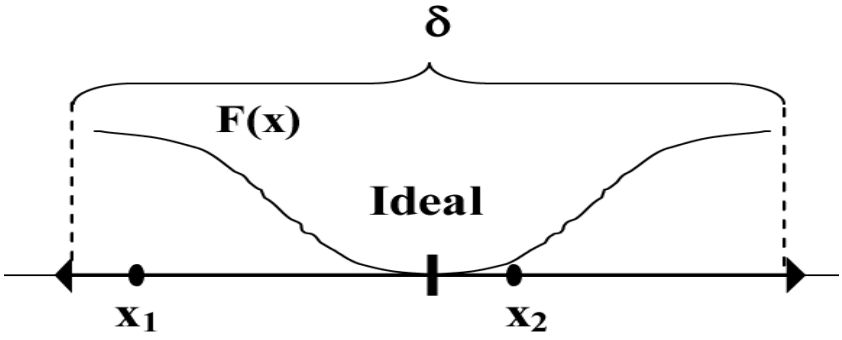
Здесь α – величина постоянная. Очевидно, что если шаг будет слишком мал, то обучение будет проходить очень медленно. Если же α велик – то, в момент достижения точки минимума (глобального или локального) функции ошибки $E = f(W)$ (E – глобальная ошибка сети; W – множество весовых коэффициентов сети) (рис. 3.8), сеть не сможет в неё попасть и



а.



б.



в.

Рис. 3.6. Удаление объектов от Эталона класса

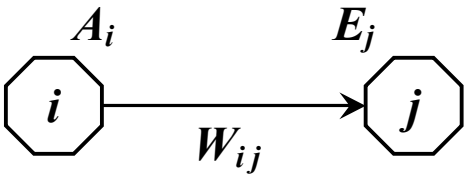


Рис. 3.7. Изменения весовых коэффициентов, для пары нейронов ***i***, ***j***

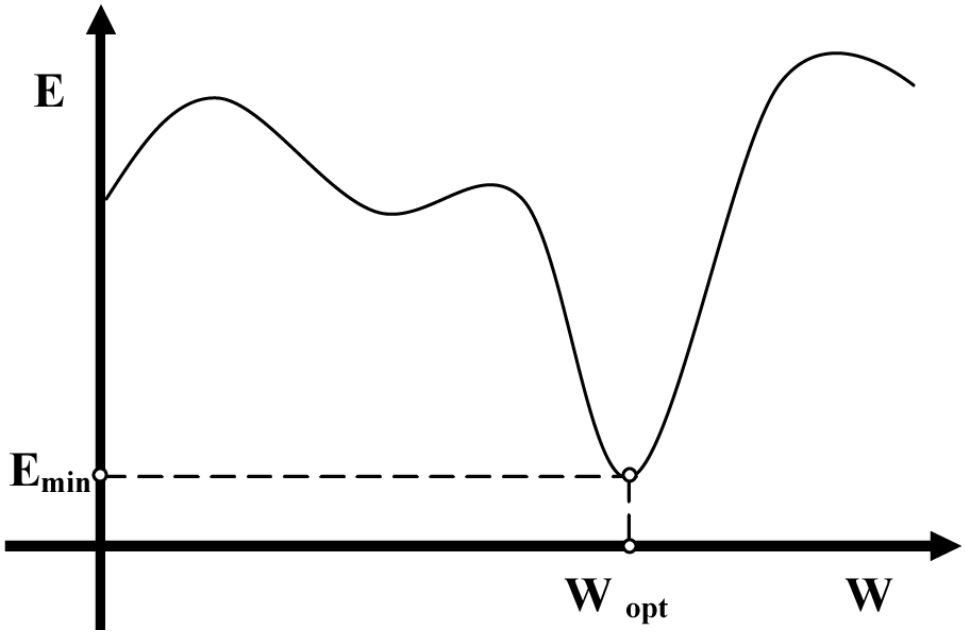


Рис. 3.8. Выбор оптимального множества весовых коэффициентов сети для минимальной глобальной ошибки

будет бесконечно долго колебаться вокруг неё, производя бесконечные пересчёты множества весов и *ухудшая свои показатели*.

Таким образом, возникает необходимость управлять величиной α. Очевидно, что если необходимо достигнуть оптимального множества весовых коэффициентов за минимальное количество итераций, то выбор некоторого среднего значения шага не является приемлемым. Целесообразно в момент начала обучения нейронной сети установить некоторое его максимальное значение, обеспечив, таким образом, быстрое приближение к области, где находится W_{opt} и, затем, постепенно уменьшать его по приближению к самой точке оптимума: $a_0 = 1$; $a_{t+1} = a_t - \partial\alpha$, где $\partial\alpha$ – декремент шага изменения весов сети.

Определение момента в процессе обучения нейронной сети, когда необходимо уменьшить величину шага, а также выбор значения $\partial\alpha$ для каждой задачи индивидуальны. Например, в решённой задаче прогнозирова-

ния остатков на банковском счёте, опытным путем было установлено, что уменьшение α целесообразно производить, если $\Delta E < 0{,}1 \cdot 10^{-6}$, а $\partial\alpha = 0{,}001$. Использование такого условия позволило в течение всего процесса обучения сохранить *достаточную скорость* уменьшения ошибки.

Предлагаемый метод динамического управления шагом изменения весовых коэффициентов нейронной сети в процессе её обучения [54] позволяет обеспечить максимальную скорость уменьшения ошибки сети E на всех участках графика $E = f(W)$.

Реорганизация распознаваемых классов

Ещё одним методом ускорения обучения нейронных сетей является реорганизация множества распознаваемых классов. В зависимости от того, насколько ОВ противоречива и неравномерна, мы можем объединять классы между собой либо образовывать новые.

Рассмотрим ситуацию с неравномерной ОВ. Здесь параметрами ПД выступают точность интерпретации исходных данных и размерность самой нейронной сети. Точность представления данных обуславливается формируемыми классами.

Для многослойных нейронных сетей типа Back Propayation зависимость этих характеристик заключается в следующем: количество распознаваемых классов однозначно определяет число нейронов в выходном слое сети, что косвенно определяет и количество нейронов в её скрытых слоях. Таким образом, сокращение числа классов ведёт к уменьшению размерности нейронной сети, а чем меньше сеть, тем быстрее она обучается.

Однако большинство реальных задач, решаемых на нейронных сетях, не допускают таких потерь в точности классификации. Таким образом, этот метод может быть применён только для задач, не обладающих жёсткими ограничениями по точности [53].

Уменьшать количество классов предлагается путём их объединения. Для выявления классов подлежащих объединению необходимо проанализировать построенную ОВ на предмет её равномерности и полноты. Если число обучающих наборов для некоторого класса не удовлетворяет условию полноты ОВ, либо оно значительно меньше, чем число наборов в других классах, то распознавание сетью этого класса будет затруднено.

Примером результатов, полученных после анализа, может быть ситуация с классическим нормальным распределением, представленная на рис. 3.9, где в качестве классов выбраны изменения некоторой величины в процентах. Для повышения равномерности ОВ выберем классы, обеспеченные обучающими наборами менее определённого N_{min} и склеим близлежащие классы. Число обучающих наборов полученных классов преодолеет порог N_{min} , и сеть сможет нормально и быстро обучиться. Однако общее число распознаваемых классов уменьшится, что приведёт к снижению точности нейронной сети в решении данной задачи. Таким образом, необходимо согласовать, используя понятие ПД, число классов, распознаваемых нейронной сетью, с её размерностью [55].

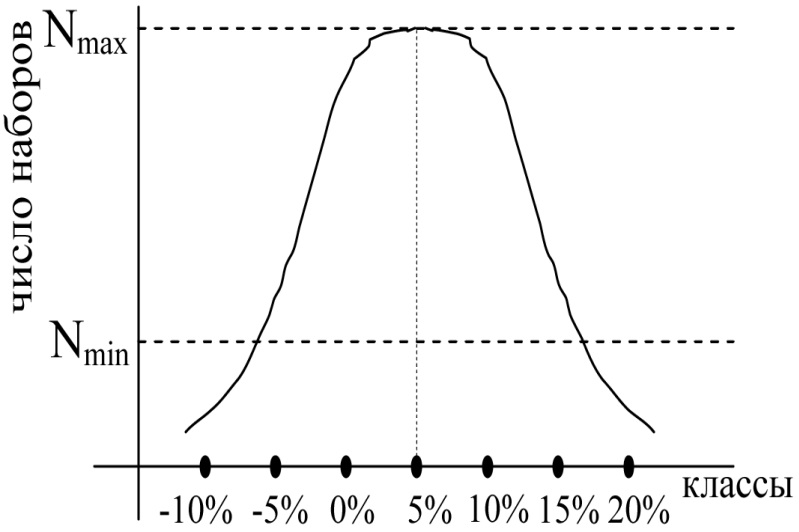


Рис. 3.9. Нормальное распределение изменения заданной величины

Формирование новых классов производится в следующих случаях:

1. Если в классе кроме объектов, приближенных к эталону класса и имеющих низкую дисперсию – *Правила*, существуют объекты удалённые от эталона и (разрозненно) расположенные вблизи его границ – *Исключения*.
2. Если дисперсия внутри класса велика и невозможно чётко определить эталон.

Исключения и нечёткие классы повышают противоречивость ОВ, их наличие может говорить о неверном разбиении пространства объектов на классы. Решением данной проблемы может быть перемещение *Исключений* в другие классы либо образование новых классов с меньшей величиной дисперсии.

Таким образом, мы можем повысить скорость обучения нейронной сети либо за счёт сокращения числа распознаваемых классов, либо, понижая противоречивость ОВ путём перемещения объектов между классами и образуя новые классы. В первом случае скорость обучения увеличивается из-за уменьшения размерности нейронной сети, во втором – из-за повышения качества ОВ.

Итак, предложено три возможных способа увеличения скорости обучения многослойных нейронных сетей. Отметим, что способ, основанный на выборе функционального базиса нейронной сети, рассматривает эту проблему с точки зрения увеличения скорости вычислений. Описанные же в статье [48] первые два метода рассматривают задачу ускорения обучения, как задачу уменьшения числа итераций обучения, а последний предполагает оба варианта. Исходя из этих предпосылок, можно сделать вывод о том, что в рамках быстрого развития современного аппаратного обеспечения, предложенные методы являются более перспективными в проблеме ускорения обучения нейронных сетей [50÷57].

3.2. Модель нейросетевой структуры для оптимизации функционирования

Предлагается алгоритм, основанный на процессе обратного воспроизводства, который динамически развивает структуру наращиваемых многослойных нейронных сетей и демонстрирует их потенциал в плане применения для управления. Данный алгоритм содержит процесс "генерирования и испытания" и оценивает действие используемой структуры, а также изменяет её в соответствии с используемыми альтернативами и отбирает наиболее перспективную. Алгоритм изменяет структуру нейронных сетей путём добавления или удаления нейронов или слоёв. Эффективность алгоритма продемонстрирована при испытании в нескольких опытах с многообещающими результатами.

Представленная нейронная сеть является многослойной наращиваемой нейронная сетью, основанной на алгоритме обратного распространения (Румельхарт, Хинтон, Уильяме, 1986 г.).

Функция активации нейрона *j* может быть выражена

$$f(x) = \frac{2}{1 + e^{-x} - 1} \tag{3.8}$$

и

$$x = \sum_j \left(W_{ji} \mathrm{O}_i + \theta_j \right) , \tag{3.9}$$

где O*i* – выход части *i*;

$W_{j\,i}$ – вес между частью *i* и частью *j*;

θ_j – пороговая величина части *j*.

Производная нейронной сети представлена в формуле

$$O_{NN} = N(W, I) . \tag{3.10}$$

Формулы (3.11÷3.13) отображают *n*-слойную нейронную сеть с сиг-

мовидной функцией:

$$N(W, I) = F(W_n, Z_{n-1}) ; \tag{3.11}$$

$$Z_{n-1} = F(W_{n-1}, Z_{n-2}) ; \tag{3.12}$$

$$Z_1 = F(W_1, I) , \tag{3.13}$$

где I – входной набор;

W_1 – матрица весовых коэффициентов между входным и выходным слоями;

W_{n-1} – матрица весовых коэффициентов между $n-2$ и $n-1$ слоями;

$W_{n\ ;\ n-1}$ – матрица весовых коэффициентов между $n-1$ и выходным слоем.

Обобщённый алгоритм правила дельта (Румельхарт, 1986 г.) используется для обновления веса нейронные сети для того, чтобы минимизировать функцию стоимости формулы (3.14) и (3.15).

$$E = \tfrac{1}{2}\sum_k \left(t_{pk} + o_{pk} \right)^2 , \tag{3.14}$$

где t_{pk} и $o_{pк}$ – желаемая и полученная величины, соответственно, выхода k,

$$E = \tfrac{1}{2}\sum_k \left(t_k + o_k \right)^2 . \tag{3.15}$$

Полученная конвергенция (сходимость) выражается формулами (3.16÷3.19).

$$w_{ji}\left(t+1\right) = w_{ji}\left(t\right) + \nabla w_{ji}\left(t+1\right) ; \tag{3.16}$$

$$\nabla w_{ji}\left(t+1\right) = \eta \delta_j O_i ; \tag{3.17}$$

$$\delta_j = \left(t_k - o_k\right) O_j \left(1 - O_j\right) , \tag{3.18}$$

если j – выходной слой, и

$$\delta_j = O_j\left(1-O_j\right)\sum_k \delta_k w_{kj} \quad , \tag{3.19}$$

если j – внутренний слой (k – преуспевающий уровень).

Вышеупомянутые уравнения показываются, что BP перемещает веса в направлении максимального уменьшения, пока E_{min} не достигнут, когда значение $\left\|\nabla_W E\right\| = 0$, где

$$\left\|\nabla_W E\right\| = \sqrt{\sum_i \left(\Delta w_i\right)^2} \quad , \tag{3.20}$$

и индекс i охватывает все веса в сети.

Алгоритм

Представленный алгоритм является конструктивным алгоритмом: изучение начинается с малой структуры, действие оценивается после каждой итерации и структура изменяется, если она не удовлетворяет показателям и критериям.

Функция оценки определена в x:

$$\left\|\nabla_W E_{av}\right\| = \sqrt{\sum_i \left(\Delta w_i\right)^2} \quad ; \tag{3.21}$$

$$\left(\Delta w_i\right)_{av} = \frac{\sum_i p\left(\Delta w_i\right)_p}{P} \quad , \tag{3.22}$$

где P – число представлений в обучающем наборе; (Δw_i) – инкрементное изменение веса k при обучении представления p.

В BP алгоритме изменение весов выполняются после каждого обучающего набора [32, 36÷51].

Изменение структуры

Как сказано выше, динамические алгоритмы не ограничиваются изменением значений синоптических весов. Они также изменяют структуру сети. В стратегии структура может быть изменена добавлением или удалением нейронов/слоёв. В начале нейрон добавляется к нейросети, путём вживления его после последнего нейрона в последний скрытый слой и соединеняется с нейронами в предыдущем слое и последующем слое через новые синоптические веса. Далее, нейрон удаляется смещением последнего нейрона из последнего скрытого слоя и разъединением всех его весов. Затем, слой добавляется, путём размещения его последним скрытым слоем и выходным слоем, перемещая все старые связи между ними, но соединяя новый слой с двумя другими через новые связи.

Несмотря на то, что вживление слишком большого количества нейронов в этот слой может привести к перенасыщению, использование очень малого числа нейронов может закончится "эффектом бутылочного горлышка" (Хект-Нильсен, 1989 г.). Количество нейронов в новом слое выражается формулой

$$N_{nev} = \mathrm{int}\left(\frac{N_0 + N_h}{2} \right), \tag{3.23}$$

где N_0 и N_h число нейронов в выходном и последнем скрытом слоях соответственно [22÷27, 58].

И, наконец, чтобы удалить слой, последний скрытый слой удаляется со всеми его связями, а новые веса соединяются между предыдущим и последующим слоями.

Стратегия

Модель предполагает постоянную оценку действия сети и генерирование наиболее перспективных структур после обнаружения деградации поведения. Операция продолжается до наступления конвергенции, которую можно выразить:

Начать обучение с малой структуры.
Repeat
    Подсчитать $\|\nabla_W E_{av}\|$ после каждой итерации
    If $\|\nabla_W E\| \geq \|\nabla_W E\|_{t-1}$ then
        генерировать новую более перспективную структуру.
    Продолжить процесс обучения.
Until ошибочная конвергенция.

Создание структуры

Чтобы создать новую структуру, нужно использовать несколько структур, изменяя структуру одной рабочей сети и выбирая самую перспективную. Этот процесс состоит из двух фаз: фаза удаления и фаза добавления. Сначала структура изменяется удалением нейрона или слоя. Если полученная структура перспективна, отметить её. В противном случае выполняется фаза добавления, а в ней используются две структуры: первая получается добавлением нейрона, вторая – добавлением слоя. Алгоритм выбирает наиболее перспективную структуру. Фаза удаления производится сначала, если преобладают малые структуры, которые уменьшают

сложность нейронной сети. Выбор удаления нейрона или слоя зависит от подвижности изменённой структуры. Обычно удаляют нейрон, но, если это приводит к малой эффективности структуры, то удаляют слой, так как дальнейшее удаление нейронов все равно приведёт к малой подвижности структуры. В результате алгоритм отмечает все малоэффективные и тупиковые структуры как неперспективные. Эти структуры будут объяснены позже.

Процесс генерации можно выразить:

```
Генерация структуры:
    Создать структуру удалением нейрона
    If nonviable then Создать структуру удалением слоя.
    If созданная структура перспективная, отметить её И выйти.
    Создать структуру добавлением нейрона.
    Создать структуру добавлением слоя.
    Отметить наиболее перспективную структуру.
```

Чтобы создать новую структуру, старая изменяется как указано выше, а затем алгоритм вступает в "стадию сна", где производится ряд повторений, когда новая структура обучается без оценки действия до обнаружения улучшения поведения или его ухудшения. Новая структура не действует, если достигнуто максимальное количество "циклов сна". Если структура не действует, она отмечается как неперспективная, в противном случае оценивается её потенциал. Создание новой структуры может быть обобщено:

Создание новой структуры:

If новая структура неподвижна или из разряда тупиковых отметьте её как неперспективную.

Изменить старую структуру.

Вступить в стадию сна.

Repeat

Выполните бездействующую обучающуюся итерацию.

If $E_{\text{new}} \le E_{\text{old}}$ Выйдите с успехом.

If $E_{st} \le E_{st-1}$ Выйдите с отказом.

Until число итераций фазы сна меньше максимального числа.

If достигнут максимум итераций фазы сна – структура ошибочна.

Выход из фазы сна.

Если замечена неудача, структура считается неперспективной.

Иначе оцените перспективность структуры.

Вновь произведённая структура считается перспективной, если структуре удается достичь значительного уменьшения ошибок по сравнению со старой. Формально, перспективная структура может быть определена по формуле

$$\left\|\nabla_W E_{av}\right\|_t \le \left\|\nabla_W E_{av}\right\|_{t-1} \text{ or } \left(E_{old} - E_{nev}\right) \ge \left(DR \times E_{old}\right) \ ,$$

и

$$0 < DR < 1 \ ,$$

где t-1 – первая итерация после фазы сна;

E_{new} – ошибка новой структуры после итерации t;

DR – коэффициент уменьшения ошибки.

Процесс добавления предполагает выбор лучшей из двух самых перспективных структур. Первая структура создается добавлением нейрона к старой, а во второй добавляется слой. Метод, используемый здесь, главным образом основан на уровне снижения функции оценки каждой структуры.

Однако, если две структуры являются перспективными, более предпочтительна структура с добавленным нейроном, потому что менее сложна.

Если только одна структура перспективна, она отмечается сразу без дальнейших сравнений с другой структурой [28÷35]. Как поступать в тупиковой ситуации (в случае, если нет перспективных структур) будет указано ниже.

Неподвижные структуры

Модель, используемая здесь, представляет собой вариант алгоритма "создания и проверки" в своей самой систематичной форме (Рич, 1983 г.). Практический способ найти решение в разумно короткое время – не рассматривать некоторые образования, если они кажутся неперспективными (Рич, 1983 г.). Гуо и Гелфанд (1991 г.) отмечали, что структуры в форме песочных часов, редко дают результаты на опыте. Поэтому и здесь алгоритм уменьшает место для поиска, относя эти структуры в разряд неподвижных [37, 38].

Самый прямолинейный путь к воплощению систематического процесса "создания и проверки" лежит через использование исследовательского дерева с возвращением к отдельным отросткам (Рич, 1983 г.). Этот алгоритм может привести к положению, которое не является требуемым решением, но из которого и которым нельзя достичь лучшей позиции (Рич, 1983 г.). Эти позиции называются "тупиками". Один из способов выхода из тупиков – использование техники возвращения: вернуться к последнему сделанному выбору и его последствиям, выбрать альтернативу в этой точке выбора, и снова двигаться вперёд (Уинстон, 1984 г.). В этой работе тупиковая ситуация случается, если структура не срабатывает и алгоритм не может произвести другую перспективную структуру. Применяя в данном случае вышеупомянутую технику возвращения, необходимо сохранить

список всех произведённых структур и их ассоциированных весов. Но, так как глубина произведённого дерева и количество структур практически неограниченны, эта техника требует большой компьютерной базы данных и сложных ресурсов. Другой способ выхода из тупиковых ситуаций предлагает возвращение к некоторым предыдущим отросткам и движение в другом направлении. Стратегия, применяемая в этом алгоритме, предлагает сохранение только одного отростка на пути: отростка, на котором алгоритм решает добавить второй скрытый слой. Этот выбор главным образом основан на экспериментах показывающих, что на этом отростке, который называют "умным" отростком, алгоритм имеет хорошую возможность выбрать перспективное направление. Хотя алгоритм приносит хорошие результаты в экспериментах, нельзя говорить о его универсальной полезности [43, 44].

Кроме того, все многослойные тупиковые структуры заносятся в список тупиковых структур. Любая структура в этом списке отмечается как неперспективная, чтобы она не производилась снова. Но если структура имеет только один слой, в самой неперспективной структуре выбирается та, у которой наименьший уровень ошибок. Процесс возвращения можно выразить:

```
IF структура многослойна then
    Begin
        Сохранить структуру в списке тупиковых:
        Возврат к "умному" отростку;
    End
    else
        Выбрать из неперспективных структур структуру у которой
        наименьший уровень ошибок.
```

Воплощение

Методы "производства и проверки" используют два основных модуля. Один модуль – "генератор" считает возможные решения. Второй модуль – "тестер" оценивает каждое предложенное решение путём принятия его или отвержения (Уинстон, 1984 г.). Однако сложность данного алгоритма потребовала использование и других модулей. Модуль обратного воспроизводства управляет процессом обучения нейросети. Он может быть как в режиме действия, так и в режиме бездействия. В активном режиме он оценивает полезное поведение и связан с экспериментатором. В стадии бездействия вызывается модуль конвергенции и проверки. Последний проверяет конвергенцию созданной структуры, посылает сообщение модулю обратного воспроизводства о выходе из стадии бездействия в случае достижения конвергенции, или вызывает экспериментатора в случаях обнаружения сбоя. Экспериментатор, принимающий решения, управляет общей работой программы. Когда необходимо изменение, он испытывает различные альтернативы и выбирает одну для дальнейших действий. Кроме того, он должен обнаружить и решить, как выйти из тупиковой ситуации. Модуль действий, который представляет производящего, изменяет структуру нейросети согласно командам экспериментатора путём добавления или удаления нейронов или слоев. Он посылает экспериментатору сообщение, если новая структура неподвижна или тупиковая, а также вызывает изучающий модуль для начала фазы бездействия. Тот факт, что алгоритм использует много тестов в различных ситуациях не позволяет иметь отдельный модуль-тестер. Вместо этого он размещён между различными модулями системы. Для воплощения модуля экспериментатора используется система, основанная на правилах, потому что это хороший способ моделировать интеллектуальные действия сильного характера, движимого данными (Рич, 1983г.).

Правила принимают следующую форму:

IF условия THEN действия.

Например:

IF (структура принята) && (статус = неудача) THEN удаление нейрона.

IF (структура = структура с удаленным нейроном) AND (статус = успех) THEN принять структуру.

Здесь, в части условия, используются переменные (структура, статус и т.д.), которые устанавливаются другими модулями системы. Например, переменная структура определяется модулем действий в соответствии с выполняемым действием [41÷49]. На нижнем уровне нейронная сеть представлена в форме ориентированной на объекты, где каждый нейрон, его веса и все его локальные процессы привязываются к одному предмету, чтобы уменьшить сложность программы.

3.3. Теоретический подход к возможности ускоренного обучения нейронных сетей за счёт адаптивного упрощения обучающей выборки

С увеличением размерности задач, возлагаемых на интеллектуальные системы, повышение скорости и качества обучения НС приобретает всё большее значение. Подтверждением этому является значительное количество исследований, направленных на ускорение обучения НС. Во многих предложенных методах увеличение скорости достигается за счёт модификации метода градиентного спуска, основанной на предположении о том или ином характере исходных данных [15]. Также часто используется динамическое изменение шага обучения [16, 17], дополнительные методы вывода НС из локального минимума [18]. Однако перечисленные методы не учитывают то обстоятельство, что дополнительное повышение скорости

и качества обучения может быть достигнуто и за счёт обработки данных, на которых это обучение происходит.

Обратившись к естественным обучающимся системам, можно заметить, что чаще всего обучение происходит не сразу на всём обучающем множестве (которым для естественных систем являются объекты реального мира), а на его упрощенной модели, отражающей лишь некоторые примеры и закономерности. По мере усвоения более простого материала модель становится всё более подробной и адекватной. То есть обучение происходит как бы «от простого к сложному». Исследование применимости такого подхода для повышения качества и скорости обучения НС и является данной целью.

Сложность обучающей выборки и способы её снижения

Под сложностью ОВ подразумевается сложность её аппроксимации нейронной сетью, которую для пары наборов $(X;Y)_i$, $(X;Y)_j$ можно охарактеризовать следующим образом [35]:

$$L = \frac{\left\| Y_i - Y_j \right\|}{\left\| X_i - X_j \right\|}, \tag{3.24}$$

где X и Y – соответственно входные и выходные вектора.

Сложность воспроизведения всей ОВ может быть получена расчётом среднего или максимального и минимального значений L_{ij} для всех пар наборов. Применение соотношения (3.24), в теории непрерывных функций называемого константой Липшица, с целью оценки обучающей возможности ОВ неоднократно обсуждалось в литературе и показало свою практическую применимость [19, 20]. Одним из способов снижения сложности

ОВ является искусственное сближение выходных векторов для наборов, входные вектора которых находятся близко друг к другу. При этом выходной вектор набора k упрощённой выборки ОВ' рассчитывается как среднее выходных векторов наборов исходной выборки ОВ, взвешенное по функции от расстояния до входного вектора k-го набора

$$Y'_k = \frac{\sum_i Y_i \cdot c_{ik}}{\sum_i c_{ik}} \ . \tag{3.25}$$

Роль взвешивающей функции может выполнять функция от расстояния между входными векторами, удовлетворяющая следующим условиям:

1. Существовать и быть неотрицательной на всём множестве возможных значений расстояния.
2. Убывать с увеличением расстояния.
3. В зависимости от некоторого параметра α изменять скорость убывания.

Таким образом, параметр α определяет «крутизну» взвешивающей функции и задаёт степень упрощения исходной выборки. Одной из наиболее известных и широко применяемых функций, удовлетворяющих перечисленным условиям, является функция Гаусса [32÷35], которую и предлагается использовать в качестве взвешивающей:

$$c_{ij} = e^{-\left(\frac{\|X_k - X_i\|}{\alpha}\right)^2} . \tag{3.26}$$

Ниже приведён пример упрощения функции одной переменной при различных значениях параметра α (рис. 3.10).

Для количественной оценки упрощения ОВ в процессе обучения НС рассмотрим следующие величины:

δ(ОВ; ОВ) – отклонение упрощенной выборки от исходной;

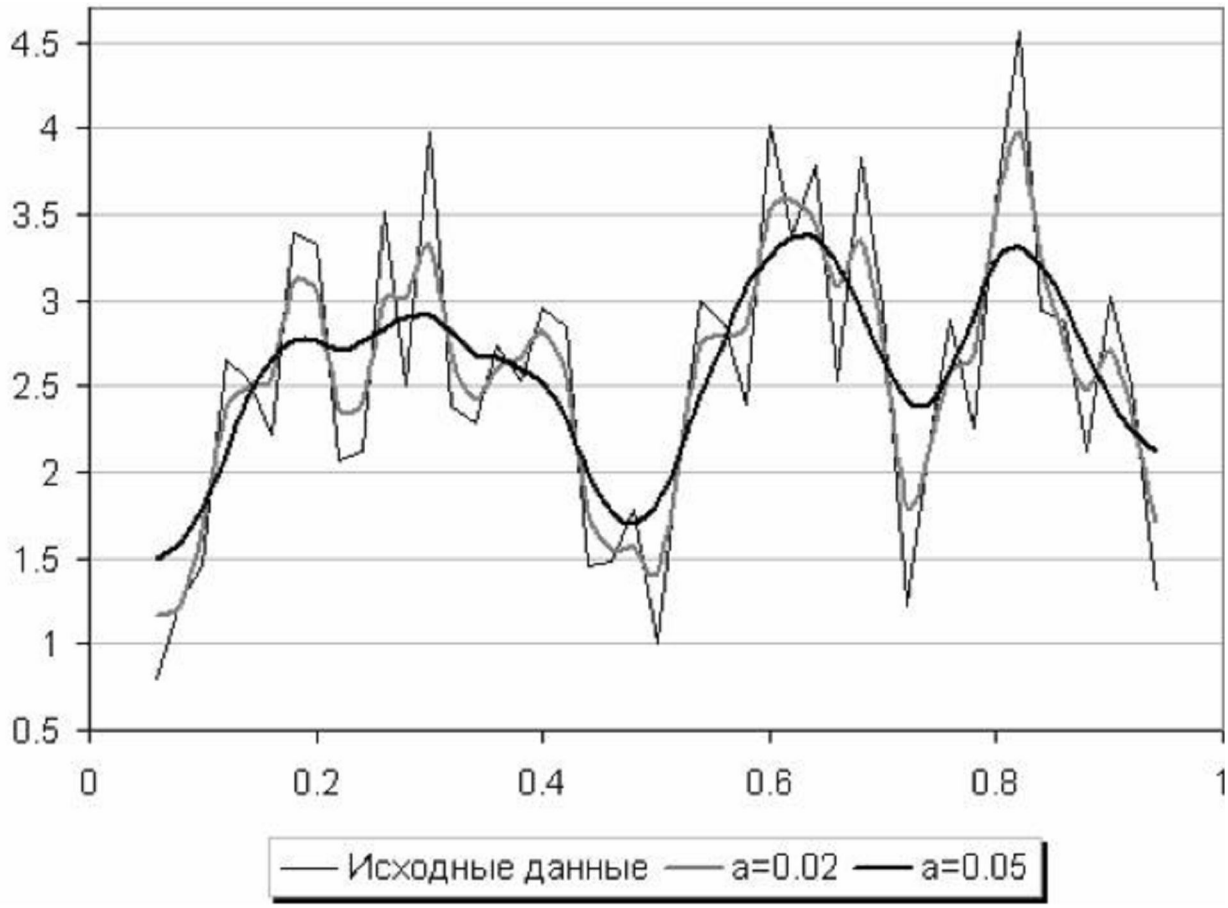


Рис. 3.10. Пример упрощения исходной функции

δ(НС; ОВ') – ошибка НС на упрощённой выборке ОВ';

δ(НС; ОВ) – ошибка НС, обученной на упрощённой выборке, рассчитанная для исходной выборки ОВ.

Пусть эти величины определены как среднее расстояние между выходными векторами в выбранной метрике. Тогда имеет место неравенство:

$$\delta(\text{ОВ}';\text{ОВ}) + \delta(\text{НС};\text{ОВ}') \geq \delta(\text{НС};\text{ОВ}) . \qquad (3.27)$$

Это позволяет воспользоваться левой частью неравенства в качестве критерия остановки обучения, а не тратить время на дополнительный расчет δ(НС; ОВ). Вместе с тем, нет необходимости обучаться на ОВ' с точностью, большей точности самой ОВ'. Следовательно, должно выполняться соотношение:

$$\delta(\text{ОВ}';\text{ОВ}) \geq \delta(\text{НС};\text{ОВ}') . \qquad (3.28)$$

Учитывая (3.27) и (3.28), можно предложить следующий алгоритм обучения НС:

1. Задается начальное значение параметра упрощения.
2. Формируется упрощенная выборка ОВ' и рассчитывается δ(ОВ'; ОВ).
3. Производится обучение НС выборке ОВ' до тех пор, пока не выполнится одно из условий:

   а) $\delta(\text{ОВ}';\text{ОВ}) + \delta(\text{НС};\text{ОВ}') \geq \delta_{доп}$,

   где $\delta_{доп}$ – допустимая ошибка, определяемая требуемой точностью решения задачи.

   Выполнение условия означает окончание обучения.

   б) $\delta(\text{ОВ}';\text{ОВ}) \geq \delta(\text{НС};\text{ОВ}')$.

   При выполнении обучения – уменьшение параметра α и переход на шаг 2.

Данный алгоритм позволяет изменить процесс обучения так, что в начале НС будет обучаться основным тенденциям и закономерностям, несколько теряя в точности, но зато не повторяя, возможно присутствующий, в исходной выборке шум. По мере усложнения выборка ОВ' будет приближаться к исходной и, в конечном итоге, либо повторит её, либо обеспечит достаточную точность решения задачи, что для НС будет означать финальный этап обучения [33÷38].

Практические результаты обучающей выборки

В ходе предварительных экспериментов производилось сравнение ошибок обучения МНС с применением предложенного подхода и без него. Исходными данными служила функция 2-х переменных с шумовой составляющей, взятая из набора функций, сформированных для сравнения обучающих алгоритмов [21]. Обучающая выборка содержала 225 наборов, а контрольная – 10000. Сеть обучалась в течение 5000 эпох, затем рассчиты-

валась ошибка на контрольной выборке (КВ) [37].

За время обучения было произведено 4 итерации упрощения ОВ. К моменту окончания обучения среднеквадратичное отклонение упрощенной выборки от исходной составляло 0,13, среднеквадратичная ошибка НС на упрощённой выборке – 0,14, на исходной – 0,17. Ошибка на КВ составила 0,48.

В случае с не изменяющейся ОВ, к окончанию обучения ошибка на ОВ составила 0,18, на КВ – 0,56.

В результате для случая с упрощением ОВ было отмечено снижение ошибок как на ОВ (6%), так и на КВ (15%).

Таким образом, использование адаптивного упрощения ОВ позволяет снизить время и, что более важно, повысить качество обучения НС. Это достигается в основном за счёт снижения избыточной подробности обучающего множества на ранних этапах обучения, что вполне характерно для естественных обучающихся систем.

Используемые в подходе преобразования относятся только к исходным данным и не затрагивают алгоритма настройки весовых коэффициентов НС. Это делает подход совместимым со многими известными методами ускоренного обучения НС, тем самым давая дополнительный выигрыш во времени и качестве обучения [61÷69].

В отношении дальнейшего развития подхода можно отметить исследование неравномерного упрощения ОВ, когда коэффициент упрощения различен для каждого набора и определяется с учётом ошибки НС на данном наборе, а не на всей выборке в среднем.

3.4. Обучение персептрона

Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. Допустим, что входные образы нанесены на демон-

страционные карты. Каждая карта разбита на квадраты и от каждого квадрата на персептрон подается вход. Если в квадрате имеется линия, то от него подается единица, в противном случае – ноль. Множество квадратов на карте задает, таким образом, множество нулей и единиц, которое и подаётся на входы персептрона. Цель состоит в том, чтобы научить персептрон включать индикатор при подаче на него множества входов, задающих нечётное число, и не включать в случае чётного [45].

На рис. 3.11 показана такая персептронная конфигурация. Допустим, что вектор $\boldsymbol{X}$ является образом распознаваемой демонстрационной карты.

Каждая компонента (квадрат) $\boldsymbol{X}$ – (x_1, x_2, x_n) – умножается на соответствующую компоненту вектора весов $\boldsymbol{W}$ – (w_1, w_2, w_n). Эти произведения суммируются. Если сумма превышает порог Θ, то выход нейрона Y равен единице (индикатор зажигается), в противном случае он – ноль. Эта операция компактно записывается в векторной форме как $Y = \boldsymbol{XW}$, а после неё следует пороговая операция.

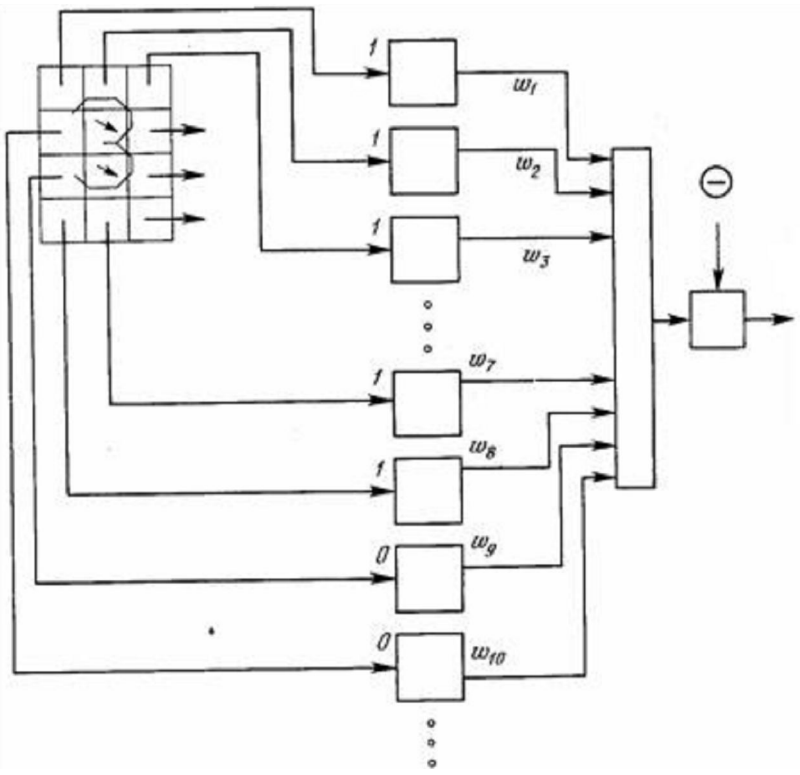


Рис. 3.11. Персептронная система распознавания изображений

Для обучения сети образ ***X*** подается на вход и вычисляется выход *Y*. Если *Y* правилен, то ничего не меняется. Однако, если выход неправилен, то веса, присоединённые к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку.

Чтобы увидеть осуществление данного процесса, допустим, что демонстрационная карта с цифрой 3 подана на вход, и выход *Y* равен 1 (показывая нечётность). Так как это правильный ответ, то веса не изменяются. Если на вход подается карта с номером 4 и выход *Y* равен единице (нечётный), то веса, присоединённые к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если карта с номером 3 дает нулевой выход, то веса, присоединённые к единичным входам, должны быть увеличены, чтобы скорректировать ошибку.

Этот метод обучения может быть подытожен следующим образом:

1. Подать входной образ и вычислить *Y*.
2. а. Если выход правильный, то перейти на **шаг 1**;
   б. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам;
   в. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса
3. Перейти на **шаг 1**.

Серьезные вопросы имеются относительно эффективности запоминания информации в персептроне (или любых других нейронных сетях) по сравнению с обычной компьютерной памятью и методами поиска информации в ней. Например, в компьютерной памяти можно хранить все входные образы вместе с классифицирующими битами. Компьютер должен найти требуемый образ и дать его классификацию. Различные, хорошо известные, методы могли бы быть использованы для ускорения поиска. Если точное соответствие не найдено, то для ответа может быть использовано правило ближайшего соседа [45].

Число битов, необходимое для хранения этой же информации в весах персептрона, может быть значительно меньшим по сравнению с методом обычной компьютерной памяти, если образы допускают экономичную запись. Однако, Минский [45] построил патологические примеры, в которых число битов, требуемых для представления весов, растёт с размерностью задачи быстрее, чем экспоненциально. В этих случаях требования к памяти с ростом размерности задачи быстро становятся невыполнимыми. Если, как он предположил, эта ситуация не является исключением, то персептроны часто могут быть ограничены только малыми задачами. Насколько общими являются такие неподатливые множества образов? Это остается открытым вопросом, относящимся ко всем нейронным сетям. Поиски ответа чрезвычайно важны для исследований по нейронным сетям.

Способность искусственных нейронных сетей обучаться является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами моделируют себя в результате попыток достичь лучшей модели поведения [1, 32,]

Используя критерий линейной разделимости, можно решить, способна ли однослойная нейронная сеть реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесёт мало пользы, если нет способа найти нужные значения для весов и порогов. Чтобы сеть представляла практическую ценность, нужен систематический метод (алгоритм) для вычисления этих значений. Розенблатт [4] сделал это в своём алгоритме обучения персептрона вместе с доказательством того, что персептрон может быть обучен всему, что он может реализовывать. Обучение может быть с учителем или без него. Для обучения с учителем нужен «внешний» учитель, который оценивал бы поведение системы и управлял её последующими модификациями. При обучении без учителя, рассматриваемого ниже, сеть путём самоорганизации делает требуемые изменения. Обучение персептрона является обучением с учителем.

Слой Гроссберга [45] функционирует в сходной манере. Его выход NET является взвешенной суммой выходов k_1, k_2,…, k_n слоя Кохонена, образующих вектор ***K*.** Вектор соединяющих весов, обозначенный через ***V*,** состоит из весов v_{11}, v_{21},…, v_{np}. Тогда выход NET каждого нейрона Гроссберга есть

$$\mathrm{NET}_j = \sum_i k_i w_{ij} \ , \tag{3.29}$$

где NET_j – выход j-го нейрона Гроссберга, или в векторной форме

$$\boldsymbol{Y} = \boldsymbol{V}\,\boldsymbol{K} \ , \tag{3.30}$$

где ***Y*** – выходной вектор слоя Гроссберга;

K – выходной вектор слоя Кохонена;

V – матрица весов слоя Гроссберга.

Если слой Кохонена функционирует таким образом, что лишь у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора ***K*** отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и сеть становится, в определенном смысле, самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается». Доказательство Розенблатта стало основной вехой и дало мощный импульс исследованиям в этой области. Сегодня в той или иной форме элементы алгоритма обучения персептрона встречаются во многих сетевых парадигмах.

Моделирование работы нейронной сети проводилось на алгоритме написанным с использованием Open Source библиотеки OpenCV 0.99 на

языке C++. Смоделированная нейронная сеть работает в режиме ассоциативной памяти и распознавания образов графических изображений в реальном времени, основанная на алгоритме распознавания Хаарта.

Моделирование нейронной сети проводилось методом распознавания графических изображений, а также в реальном времени при съёмке видеокамерой.

В качестве внешних данных для нейронной сети служило изображение с видеокамеры, далее в качестве единичного входного сигнала на один вход бралась область изображения с *x* точками. Нейронная сеть, обработав значения *x*, распознавала взаимное расположение изображений объектов в пространстве и их движение.

В качестве программной оболочки для визуализации расчётов использовался интерфейс с использованием Open Source библиотеки OpenCV 0.99 на языке C++ (рис. 3.12).

Параметр xsens отвечает за чувствительность перемещения объекта по координате *x* (1÷100) ysens – за чувствительность перемещений объекта по оси *y* (1÷100). Параметр imntorafr (1÷100) – определяет количество входных данных по точкам для обработки движения объекта. Чем больше точек, тем больше данных для решения задачи распознавания, и тем больше глобальная ошибка сети при следующем распознавании стремится к нулю, но тем выше вероятность «шумов», то есть увеличивается диапазон допустимых значений параметров координат движения [1, 4, 84÷88].

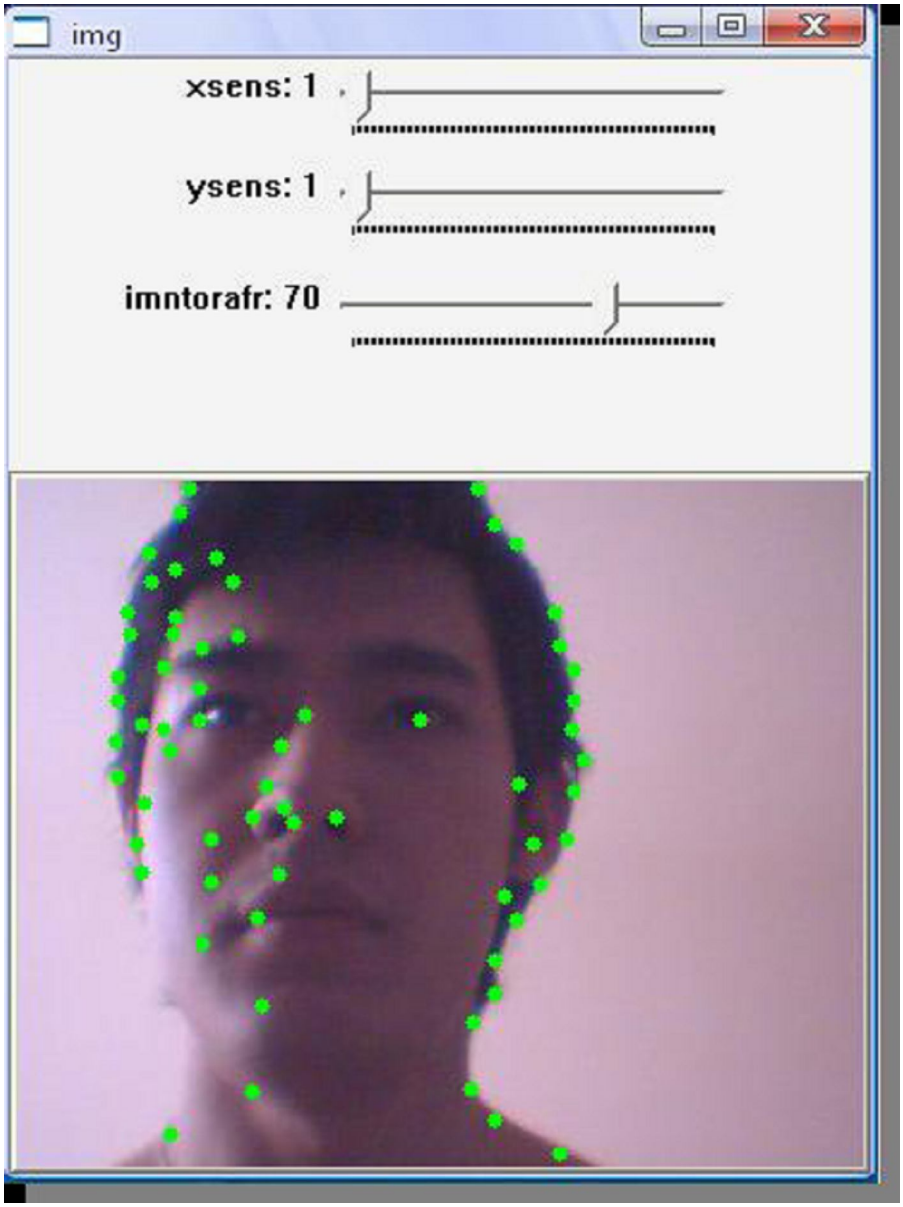


Рис. 3.12. Интерфейс программы распознавания образов с видеокамеры

ГЛАВА 4. РЕЗУЛЬТАТЫ ПРИМЕНЕНИЯ АЛГОРИТМА ПОИСКА ТЕХНИЧЕСКИХ РЕШЕНИЙ ПО УСТРОЙСТВАМ ДЛЯ ПРОИЗВОДСТВА ЭЛЕМЕНТОВ НЕЙРОННЫХ СЕТЕЙ В ТУННЕЛЬНО-ЗОНДОВОЙ НАНОТЕХНОЛОГИИ.

4.1. Нейронная сеть

Рассмотрим несколько вариантов установок элементов искусственных нейронных сетей. Показанный, на рис. 4.1 граф определяет топологию связей нейронной сети.

Для искусственной нейронной сети достаточным структурным описанием может служить ориентированный граф, в котором каждая вершина соответствует одному нейронному ядру, а дуги отвечают операторам

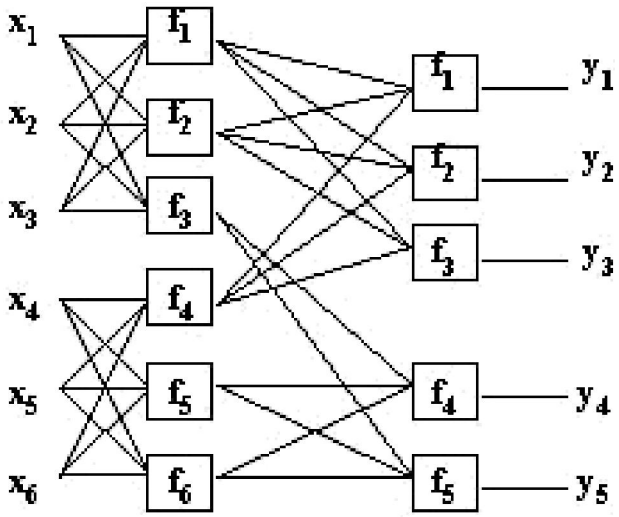


Рис. 4.1. Двухслойная искусственная нейронная сеть
(каждая вершина соответствует формальному нейрону)

межъядерных связей. Такой граф называется структурной моделью искусственной нейронной сети. На рис.4.2 показан пример структурной модели для топологической реализации. На графе структурной модели показаны веса вершин и веса дуг. Вес вершины i определяется структурной характеристикой ядра (A_i, B_i), а вес дуги (r_{ij}) равен рангу проектирующего оператора межъядерной связи [93÷100].

Обратный переход от структурной модели к топологической реализации неоднозначен, что можно трактовать как свойство топологической пластичности ядерной нейронной сети. Степень топологической пластичности будем оценивать мощностью множества топологических реализаций отвечающих данной структурной модели.

Искусственная нейронная сеть содержит входной сумматор, последовательно связанный с нелинейным преобразователем сигналов и точкой ветвления для рассылки одного сигнала по нескольким адресам. Также сеть содержит линейную связь – синапс, для умножения входного сигнала на вес синапса. В качестве входного сумматора используются нанотранзисторы с электрически изолированной областью (плавающим затвором – floating gate), способные хранить заряд многие годы. Нелинейным преобразователем в свою очередь, являются квантовые точки. Точкой ветвления является узел, выполненный из углеродных нанотрубок, а синапс представляет собой наличие или отсутствие заряда на плавающем затворе, которое кодирует один бит информации. При записи заряд помещается на плавающий

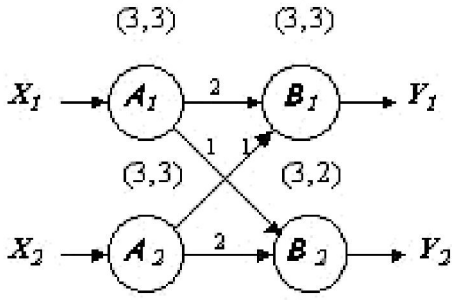


Рис. 4.2. Структурная модель

затвор одним из двух методов (зависит от типа ячейки): методом инжекции электронов или методом туннелирования электронов. Стирание содержимого ячейки (снятие заряда с плавающего затвора) производится методом туннелирования. Как правило, наличие заряда на транзисторе понимается как логический «ноль», а его отсутствие – как логическая «единица».

Искусственная нейронная сеть работает следующим образом (рис. 4.3). Адаптивный сумматор 1, имеющий $n + 1$ вход и получающий на 0 вход постоянный сигнал, вычисляет скалярное произведение вектора входного сигнала χ_5 на вектор параметров α . Нелинейный преобразователь сигнала 2 получает скалярный входной сигнал χ_5, и переводит его в $\varphi(\chi)$. Точка ветвления 3 служит для рассылки одного сигнала по нескольким адресам, она получает скалярный входной сигнал χ_5, и передает его всем своим выходам. Стандартный формальный нейрон составлен из входного сумматора1, нелинейного преобразователя 2 и точки ветвления 3 на выходе. Линейная связь – синапс 4 умножает входной сигнал на вес синапса α.

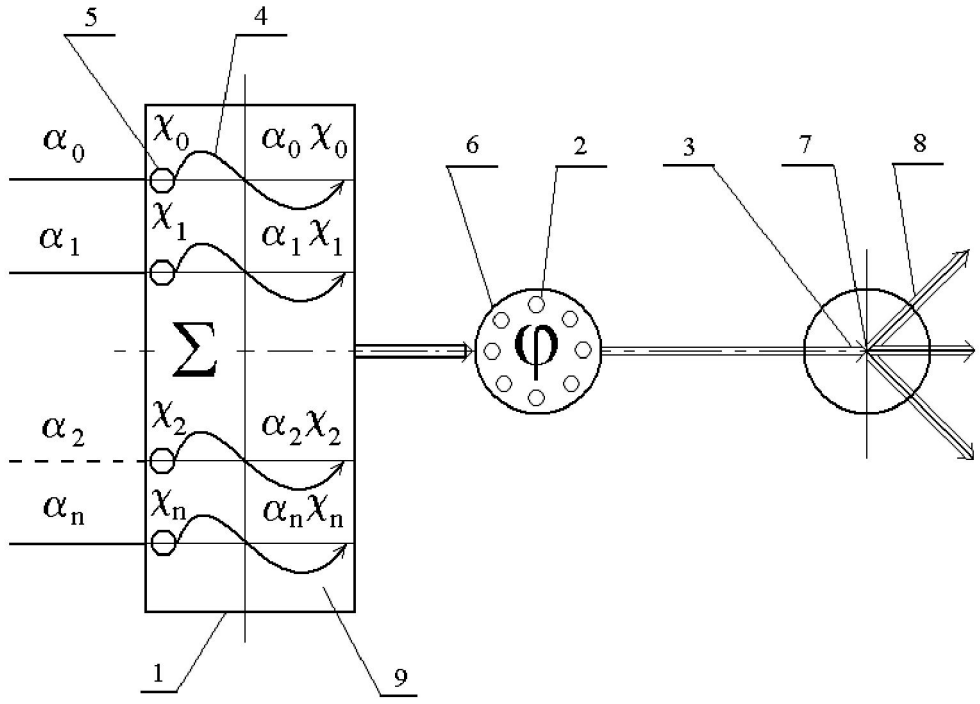


Рис. 4.3. Искусственная нейронная сеть

Нелинейным преобразователем 2 являются квантовые точки 6. Точкой ветвления 3 является узел 7 выполненный из углеродных нанотрубок 8. Синапс представляет собой отдельно взятый переход одного наличествующего или отсутствующего заряда на плавающем затворе, который кодирует один бит информации 5 под действием туннелирования электронов 9.

Применение предложенной модели технического решения искусственной нейронной сети позволяет снизить физические габариты при реализации наноэлектронных схем и компонентов, что связанно с повышением производительности системы [102].

4.2. Устройство для получения углеродных плёнок

В основу устройства положена задача снизить толщину, при реализации получаемых углеродных плёнок на наноуровне.

Эта задача решается тем, что между молекулярным источником и подложкой установлена система подачи инертного газа, выполненная в виде тепловой трубы, связанной с криопанелью.

Введение в устройство для получения углеродных плёнок молекулярного источника и подложки, а также введение установленной системы подачи инертного газа, выполненного в виде тепловой трубы, связанной с криопанелью обеспечивает возможность снизить толщину, при реализации получаемых углеродных плёнок на наноуровне.

Устройство для получения углеродных пленок представлено на рис. 4.4. Оно содержит молекулярный источник 1, установленный с возможностью воздействия на подложку 2, выполненную из полупроводникового материала. Между молекулярным источником 1 и подложкой установлена система подачи инертного газа 3, выполненная в виде тепловой трубы 4, связанной с криопанелью 5.

При подаче напряжения на молекулярный источник 1, инертный газ

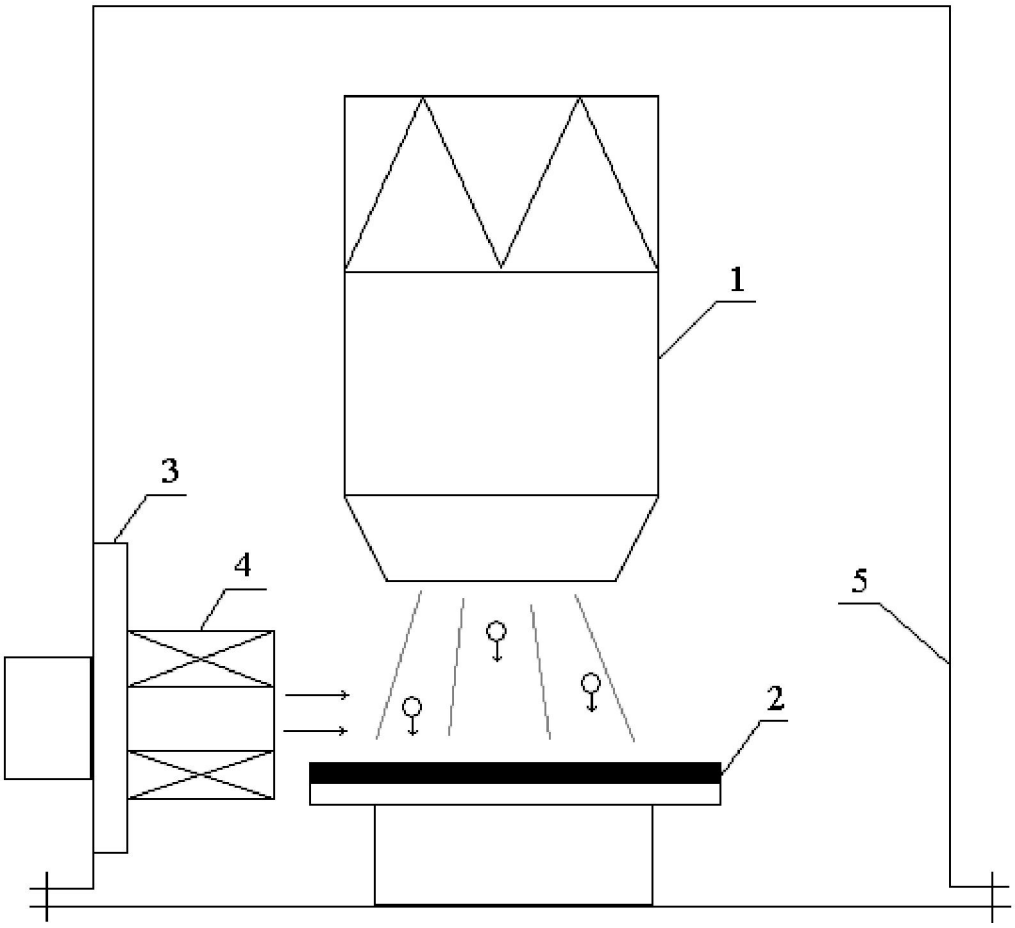


Рис. 4.4. Устройство для получения углеродных плёнок

напускаемый системой подачи 3 ионизируется и осаждается на подложке 2, выполненной из полупроводникового материала, в виде плёнки наноразмерной толщины Технологическая зона осаждения изолирована от воздействия внешней среды криопанелью 5, а рабочая температура поддерживается с помощью тепловой трубы 4 [100].

Применение предложенного устройства для получения углеродных плёнок позволяет получать углеродные пленки наноразмерной толщины.

4.3. Устройство для получения нанодорожек

В основу устройства, показанного на рис. 4.5, положена задача обеспечения возможности оперативного управления процессом получения раз-

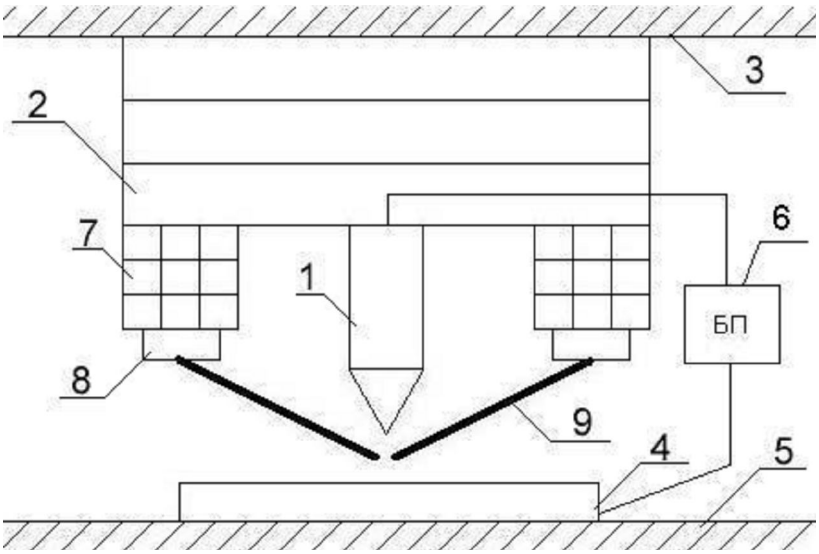


Рис. 4.5. Устройство для получения нанодорожек

личных нанодорожек.

Эта задача решается тем, что устройство дополнительно снабжено двухкоординатными пьезоприводами, жёстко закреплёнными на торце основного. На торцах дополнительных двухкоординатных пьезоприводов установлены ёмкости с рабочим веществом. С каждой ёмкостью герметично связана капиллярная трубка, свободный торец которой расположен вблизи острия зонда.

Введение в устройство для получения нанодорожек дополнительных двухкоординатных пьезоприводов, жёстко закреплённых на торце основного, ёмкостей с рабочим веществом, установленных на торцах двухкоординатных пьезоприводов, а так же капиллярных трубок, свободные торцы которых расположены вблизи острия зонда, позволяет наносить мозаичную структуру на поверхность подложки.

Устройство для получения нанодорожек содержит зонд 1, жестко закреплённый на основном пьезоприводе 2, установленный на платформе 3, подложку 4, установленную на основании 5 и блок питания 6, электрически связанный с зондом 1 и подложкой 4. Так же оно оснащено двумя до-

полнительными двухкоординатными пьезоприводами 7, жёстко закреплёнными на торце основного пьезопривода 2, ёмкостями 8, с рабочим веществом, установленными на торцах двухкоординатных пьезоприводов 7, и капилярными трубками 9, свободные торцы которых расположены вблизи острия зонда.

Устройства для получения нанодорожек работает следующим образом.

Посредством основного пьезопривода устанавливается туннельный зазор между зондом и подложкой. С помощью дополнительных двухкоординатных пьезоприводов сопла с рабочим веществом устанавливаются в исходное положение. В туннельный зазор подаётся рабочее вещество из сопл 9. Количество сопл, а следовательно, и рабочих веществ не ограничено. При подаче разности потенциалов между зондом и подложкой, атомы рабочего вещества проникают в глубь подложки. В результате чего появляется возможность наносить мозаичную структуру на поверхность подложки.

Применение предложенного устройства для получения нанодорожек позволяет значительно упростить нанесение мозаичной структуры на поверхность подложки [100].

4.4. Устройство наноперемещений

В основу устройства положена задача повысить быстродействие за счёт нитевидного зонда и линий напряженности между обкладками конденсатора, выполняющих роль направляющих.

Эта задача решается тем, что периферийная часть зонда выполнена нитевидной, диаметром (0,05÷0,1) мм, **с** крестообразными закреплёнными на ней электропроводящими пластинами таким образом, что острие зонда расположено ниже электропроводящих пластин на (0,5÷1,0) мм. Пластины,

в свою очередь, установлены между обкладками двух конденсаторов, расположенными взаимноперпендикулярно, с возможностью независимой подачи электрического напряжения на их обкладки.

Введение в устройство наноперемещений неподвижного основания, установленного на нём пьезопривода, на торце которого закреплён зонд с возможностью электрического взаимодействия с подложкой, установленный на платформу, а также периферийную часть зонда выполненную нитевидной, диаметром (0,5÷1,0) мкм, с крестообразными закреплёнными на ней электропроводящими пластинами таким образом, что острие зонда расположено ниже электропроводящих пластин на (0,5÷1,0) мкм, которые, в свою очередь установлены между обкладками двух конденсаторов, расположенными взаимноперпендикулярно, с возможностью независимой подачи электрического напряжения на их обкладки, позволяет повысить быстродействие за счёт нитевидного зонда и линий напряженности между обкладками конденсатора, выполняющих роль направляющих.

Устройство наноперемещений показано на рис. 4.6 и содержит неподвижное основание 1, установленный на нём пьезопривод 2, на торце которого закреплён зонд 3 с возможностью электрического взаимодействия с подложкой 4, установленной на платформу 5. Переферийная часть 6 зонда 3 выполнена нитевидной, диаметром (0,5÷1,0) мкм, с крестообразными закреплёнными на ней электропроводящими пластинами 7 таким образом, что острие зонда 3 расположенно ниже электропроводящих пластин 7 на (0,5÷1,0) мкм. Пластины 7, в свою очередь установлены между обкладками 8 двух конденсаторов 9, 10, расположенными взаимноперпендикулярно, с возможностью независимой подачи электрического напряжения на их обкладки 8.

Устройство наноперемещений работает следующим образом. С помощью пьезопривода периферийная часть зонда выполненная нитевидной, диаметром (0,5÷1,0) мкм, устанавливается в зазор между электропроводя-

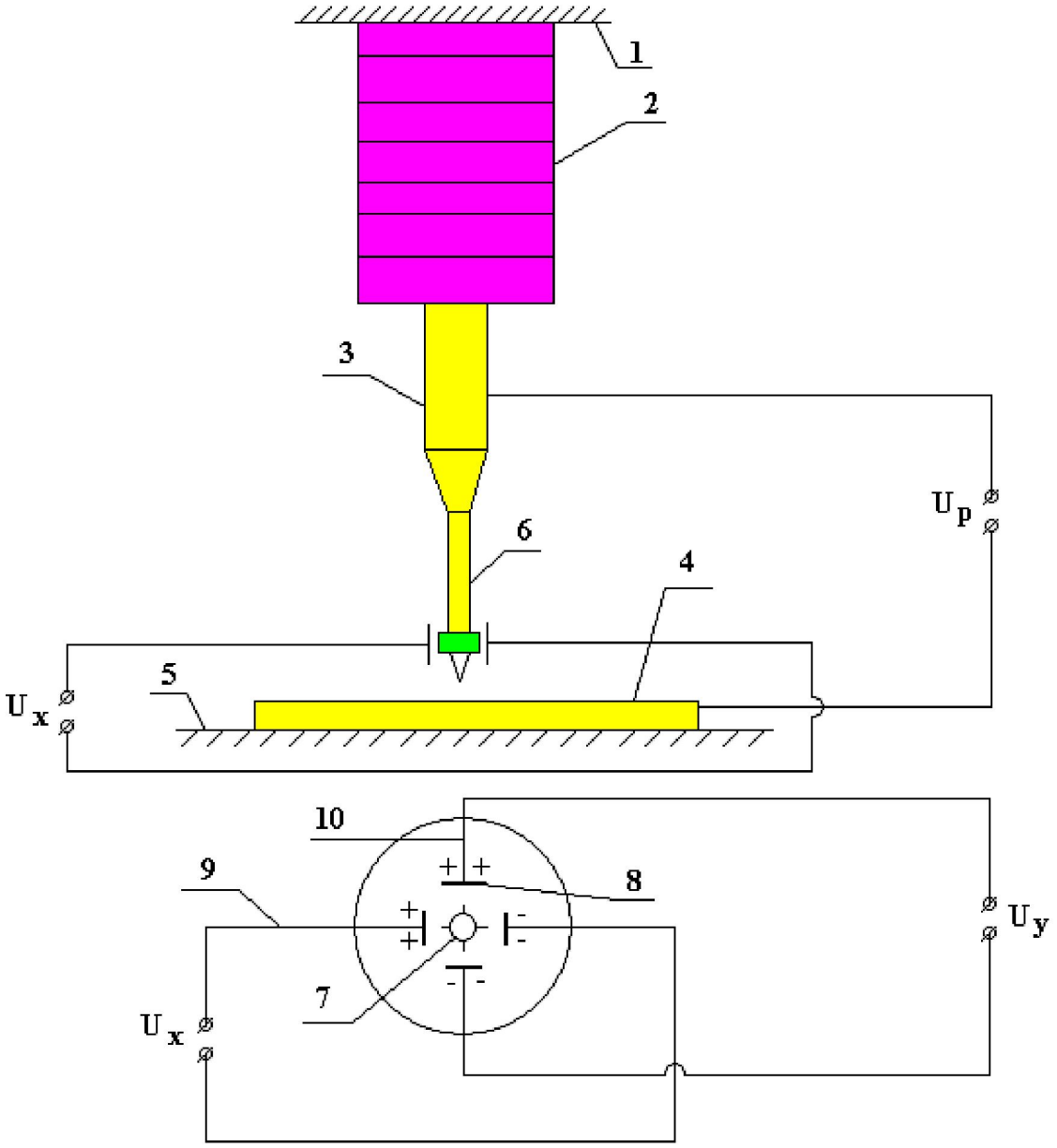


Рис. 4.6. Устройство наноперемещений

щими пластинами, до возникновения туннельного тока, таким образом что острие зонда расположено ниже электропроводящих пластин 7 на (0,5÷1,0) мкм. Пластины 7 установлены между обкладками 8 двух конденсаторов 9, 10, расположенными взаимноперпендикулярно, с возможностью независимой подачи электрического напряжения на их обкладки. При подаче разности потенциалов между зондом и подложкой, атомы рабочего вещества проникают в глубь подложки, а зонд имеющий нитевидную форму, диа-

метром (0,5÷1,0) мкм, позволяет существенно повысить быстродействие за счёт своих малых габаритов.

Применение предложенного устройства наноперемещений позволяет повысить быстродействие за счёт нитевидного зонда и линий напряженности между обкладками конденсатора, выполняющих роль направляющих, при изготовлении наносхем [98].

4.5. Устройство флэш-памяти

В основу разработки положена задача снижения габаритных размеров и увеличения объёма памяти при реализации матрицы с элементами логической памяти выполненными на наноразмерном уровне.

Введение в устройство флэш-памяти матрицы с закрепленным на ней элементами логической памяти типа 0-1, и элементов логической памяти выполненных в одиночных многоходовых МОП транзистороах, с полевым нанотранзистором с электрически изолированной областью, позволяет снизить габаритные размеры и увеличить размер памяти при реализации матрицы с элементами логической памяти на наноразмерном уровне.

Устройство флэш-памяти, показанное на рис. 4.7, содержит матрицу 1 с закреплёнными на ней элементами логической памяти 2 типа 0-1. Устройство так же содержит преобразователь магнитных сигналов 3, тактовый генератор 4 связанный с элементами логической памяти 2 и преобразователем электромагнитных сигналов 3. Элементы логической памяти 2 выполнены в виде одиночных многоходовых МОП транзисторов 5, с полевым нанотранзистором с электрически изолированной областью 6. Преобразователь магнитных сигналов 3 выполнен в виде приемника-передатчика 8, 7электронов – диэлектрики между плавающим затвором и подложкой (термический оксид кремния), и туннельными и управляющими диэлектриками между плавающим затвором и контрольным входом 9.

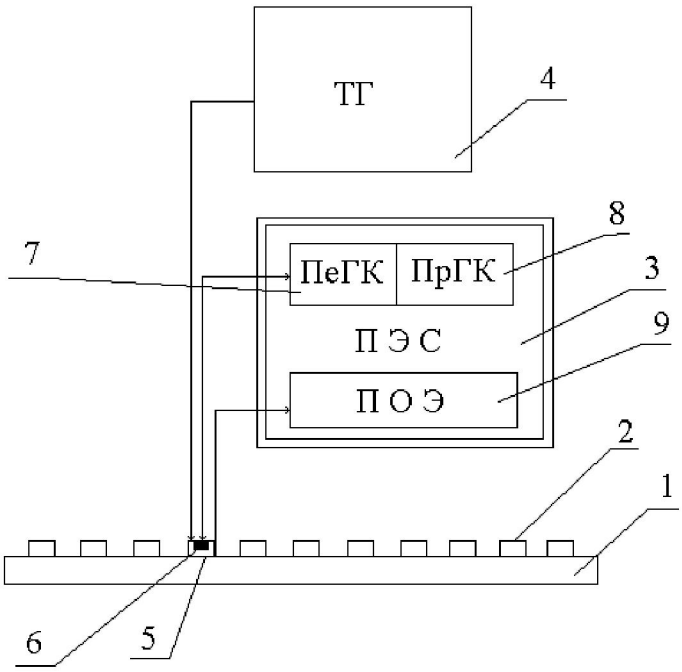


Рис. 4.7. Устройство флэш-памяти

Элементы логической памяти «1-0» 2, выполненные в виде квантовых точек 5, с помощью воздействия на них тактового генератора 4 вводятся в состояние насыщения которое соответствует 1 или 0, и фиксируются преобразователем электромагнитного сигнала 3. За счёт туннелирования электронов, пересылаемым на матрицу 1 диэлектриком выполняющим роль истока 7, полевые нанотранзисторы переходят в режим насыщения элетронами, в котором могут находится неограниченный период времени. Далее наличие или отсутсвие заряда на плавающем затворе 8 кодирует один бит информации.

При записи заряд помещается на плавающий затвор 5 одним из двух методов (зависит от типа ячейки): методом инжекции электронов или методом туннелирования электронов. Стирание содержимого ячейки (снятие заряда с плавающего затвора) производится методом туннелирования. Наличие заряда на транзисторе понимается как логический «ноль», а его отсутствие – как логическая «единица». Изменение порогового напряжения

ΔF_{TH}, вызванное хранением заряда Q_{FG} определяется

$$\Delta F_{TH} = -\frac{Q_{FG}}{C_{CG}} \ ,$$

где C_{CG} – ёмкость между контрольным и изолированным входом и задается формулой

$$C_{CG} = \frac{\varepsilon \cdot A}{t} \ ,$$

где A – площадь конденсатора, а ε и t – диэлектрическая константа и толщина управляющего диэлектрика соответственно [93, 108].

ГЛАВА 5. МЕТОДИКА ВЫБОРА ОПТИМАЛЬНОГО ВАРИАНТА ТЕХНОЛОГИЧЕСКОГО РЕШЕНИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ ТВЁРДОТЕЛЬНЫХ ОБЪЕКТОВ

5.1. Критерии вариантов технологического применения нейронных сетей

Модель формального нейрона не является биоподобной и скорее похожа на математическую абстракцию, чем на живой нейрон. Тем удивительнее оказывается многообразие задач, решаемых с помощью таких нейронов и универсальность получаемых алгоритмов. Поэтому при рассмотрении вариантов НС, основной упор делается на ограничения и условия связанные с моделью технического нейрона, а именно:

1. Вычисления выхода нейрона предполагаются мгновенными, не вносящими задержки. Непосредственно моделировать динамические системы, имеющие "внутреннее состояние", с помощью таких нейронов нельзя.
2. В модели отсутствуют нервные импульсы. Нет модуляции уровня сигнала плотностью импульсов, как в нервной системе. Не появляются эффекты синхронизации, когда скопления нейронов обрабатывают информацию синхронно, под управлением периодических волн возбуждения-торможения.
3. Нет чётких алгоритмов для выбора функции активации.
4. Нет механизмов, регулирующих работу сети в целом (например, гормональная регуляция активности в биологических нервных сетях).
5. Чрезмерная формализация понятий: "порог", "весовые коэффициен-

ты". В реальных нейронах нет числового порога, он динамически меняется в зависимости от активности нейрона и общего состояния сети. Весовые коэффициенты синапсов тоже не постоянны. "Живые" синапсы обладают пластичностью и стабильностью: весовые коэффициенты настраиваются в зависимости от сигналов, проходящих через синапс.

6. Существует большое разнообразие биологических синапсов. Они встречаются в различных частях клетки и выполняют различные функции. Тормозные и возбуждающие синапсы реализуются в данной модели в виде весовых коэффициентов противоположного знака, но разнообразие синапсов этим не ограничивается. Дендро-дендритные, аксо-аксональные синапсы не реализуются в модели ФН.
7. В модели не прослеживается различие между градуальными потенциалами и нервными импульсами. Любой сигнал представляется в виде одного числа [45, 95÷97].

5.1.1. Виды функций активации

Рассмотрим основные виды функций активации, получившие распространение в искусственных НС.

5.1.1.1. Жёсткая ступенька

$$\mathrm{OUT} = \begin{cases} 0. & \mathrm{NET} < \theta \\ 1. & \mathrm{NET} \geq \theta \end{cases} \tag{5.1}$$

Используется в классическом формальном нейроне. Развита полная теория, позволяющая синтезировать произвольные логические схемы на основе ФН с такой нелинейностью. Функция (рис. 5.1) вычисляется двумятремя машинными Инструкциями, поэтому нейроны с такой нелинейностью требуют малых вычислительных затрат. Эта функция чрезмерно уп-

рощена и не позволяет моделировать схемы с непрерывными сигналами. Отсутствие первой производной затрудняет применение градиентных методов для обучения таких нейронов. Сети на классических ФН чаще всего формируются, синтезируются, то есть их параметры рассчитываются по формулам, в противоположность обучению, когда параметры подстраиваются итеративно [84÷90].

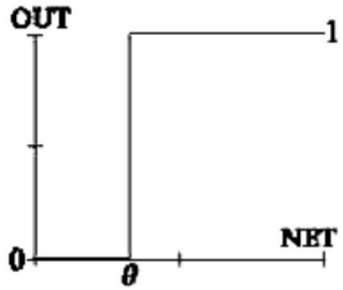


Рис. 5.1. Жёсткая ступенька

5.1.1.2. Логистическая функция, сигмоида, функция Ферми

$$\mathrm{OUT} = \mathrm{th}\left(\mathrm{NET}\right) = \frac{e^{\mathrm{NET}} - e^{\text{-NET}}}{e^{\mathrm{NET}} + e^{\text{-NET}}} \ . \tag{5.2}$$

Логистическая функция (рис. 5.2) тоже применяется часто для сетей с непрерывными сигналами. Функция симметрична относительно точки (0,0) Это преимущество по сравнению с сигмоидой. Производная также непрерывна и выражается через саму функцию.

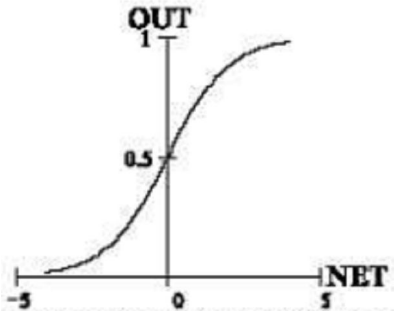


Рис. 5.2. Логистическая функция

5.1.1.3. Пологая ступенька

$$\mathrm{OUT} = \begin{cases} 0, & \mathrm{NET} \le \theta; \\ \dfrac{\mathrm{NET} - \theta}{\Delta}, & \theta \le \mathrm{NET} < \theta + \Delta; \\ 1, & \mathrm{NET} \ge \theta + \Delta. \end{cases} \tag{5.3}$$

Пологая ступенька (рис. 5.3) рассчитывается легко, но имеет разрывную первую производную в точках NET = θ, NET = θ + Δ. Это усложняет процесс обучения.

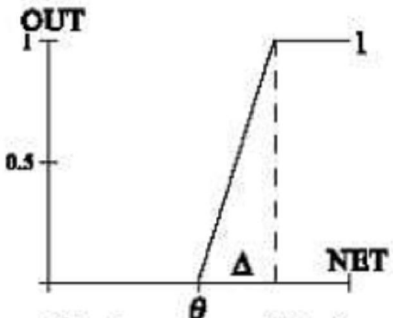


Рис. 5.3. Ступенька с пологой частью

5.1.1.4. Экспонента

$$\mathrm{OUT} = e^{-\mathrm{NET}} \ . \tag{5.4}$$

Применяется в специальных случаях.

5.1.1.5. SOFTMAX-функция

$$\mathrm{OUT} = \mathrm{th}\left(\mathrm{NET}\right) = \frac{e^{\mathrm{NET}}}{\sum_i e^{\mathrm{NET}_i}} \ . \tag{5.5}$$

Здесь суммирование производится по всем нейронам данного слоя сети. Такой выбор функции обеспечивает сумму выходов слоя, равную единице при любых значениях сигналов NET_i данного слоя. Это позволяет трактовать OUT_i как вероятности событий, совокупность которых (все выходы слоя) образует полную группу. Это полезное свойство позволяет применить SOFTMAX-функцию в задачах классификации, проверки гипотез, распознавания образов и во всех других, где требуются выходы-вероятности.

5.1.1.6. Участки синусоиды

$$\mathrm{OUT} = \sin\left(\mathrm{NET}\right) \quad \text{для } \mathrm{NET} = \left[-\frac{\pi}{2},\ \frac{\pi}{2}\right],\ \text{или } \mathrm{NET} = \left[-\pi,\ \pi\right]. \quad (5.6)$$

5.1.1.7. Гауссова кривая

$$\mathrm{OUT} = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(\mathrm{NET}-m)}{2\sigma^2}}. \quad (5.7)$$

Гауссова кривая (рис. 5.4) применяется в случаях, когда реакция нейрона должна быть максимальной для некоторого определенного значения NET.

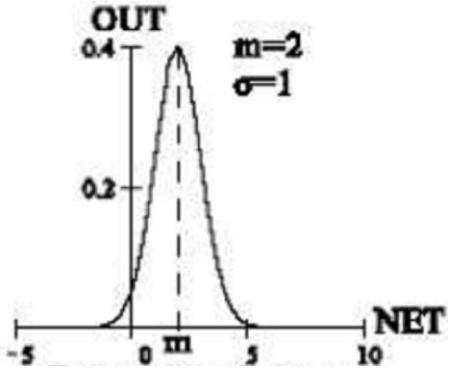


Рис. 5.4. Гауссова кривая

5.1.1.8. Линейная функция

$$\mathrm{OUT} = K \cdot \mathrm{NET}\ , \qquad K = \mathrm{const}. \tag{5.8}$$

Применяется для тех моделей сетей, где не требуется последовательное соединение слоев нейронов друг за другом.

5.1.1.9. Выбор функции активации

Выбор функции активации определяется:

1. Спецификой задачи.
2. Удобством реализации на ЭВМ, в виде электрической схемы или другим способом.
3. Алгоритмом обучения. Некоторые алгоритмы накладывают ограничения на вид функции активации и их нужно учитывать.

Чаще всего вид нелинейности не оказывает принципиального влияния на решение задачи. Однако удачный выбор может сократить время обучения в несколько раз [90÷95].

5.1.2. Варианты технологического применения нейросети

В качестве критериев вариантов технологического применения НС могут выступать структура нейросети и обучение нейронных сетей.

5.1.2.1. Структура нейросети

Искусственные нейронные сети различаются своей архитектурой: структурой связи между нейронами, числом слоёв, функцией активации нейронов, алгоритмом обучения. С этой точки зрения среди известных НС

можно выделить статические, динамические сети и fuzzy – структуры (последнее – термин теории нечётких множеств); однослойные и многослойные сети. Различия вычислительных процессов в сетях часто обусловлены способом взаимосвязи нейронов, поэтому выделяют следующие виды сетей:

1. Сети прямого распространения – сигнал проходит по сети от входа к выходу в одном направлении.
2. Сети с обратными связями.
3. Сети с боковыми обратными связями.
4. Гибридные сети.

В целом, по структуре связей НС могут быть сгруппированы в два класса: сети прямого распространения – без обратных связей в структуре и рекуррентные сети – с обратными связями. В первом классе наиболее известными и чаще используемыми являются многослойные нейронные сети, где искусственные нейроны расположены слоями. Связь между слоями однонаправленная и в общем случае выход каждого нейрона связан со всеми входами нейронов последующего слоя. Такие сети являются статическими, так как не имеют в своей структуре ни обратных связей, ни динамических элементов, а выход зависит от заданного множества на входе и не зависит от предыдущих состояний сети. Сети второго класса являются динамическими, так как из-за обратных связей состояние сети в каждый момент времени зависит от предшествующего состояния. (Ниже, на рис. 5.5, приведена классификация) [41÷45, 84÷90].

5.1.2.2. Обучение нейронных сетей

Среди всех интересных свойств искусственных нейронных сетей ни одно не захватывает так воображения, как их способность к обучению. Их обучение до такой степени напоминает процесс интеллектуального развития человеческой личности что может показаться, что достигнуто глубокое

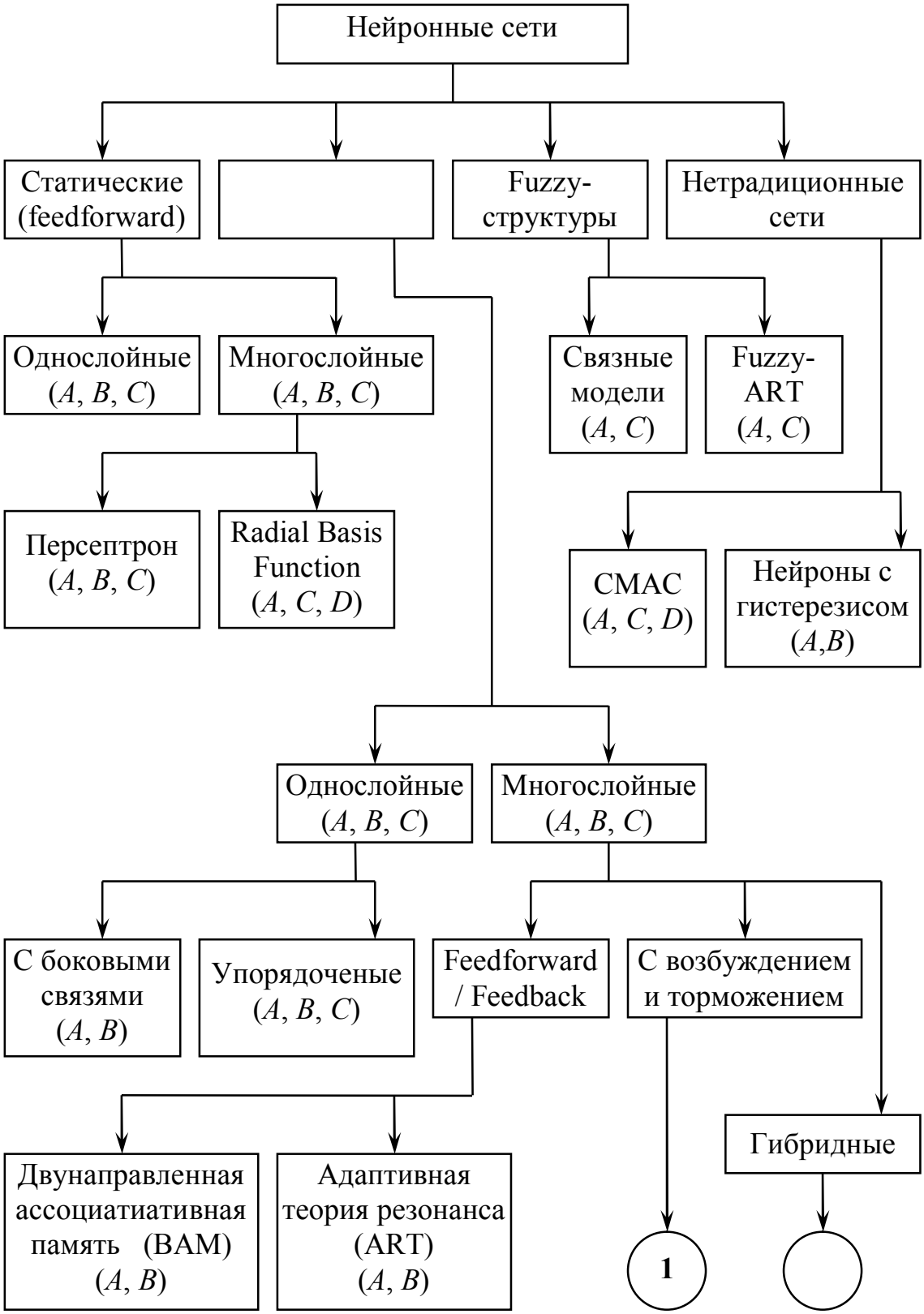


Рис. 5.5. Классификация нейронных сетей

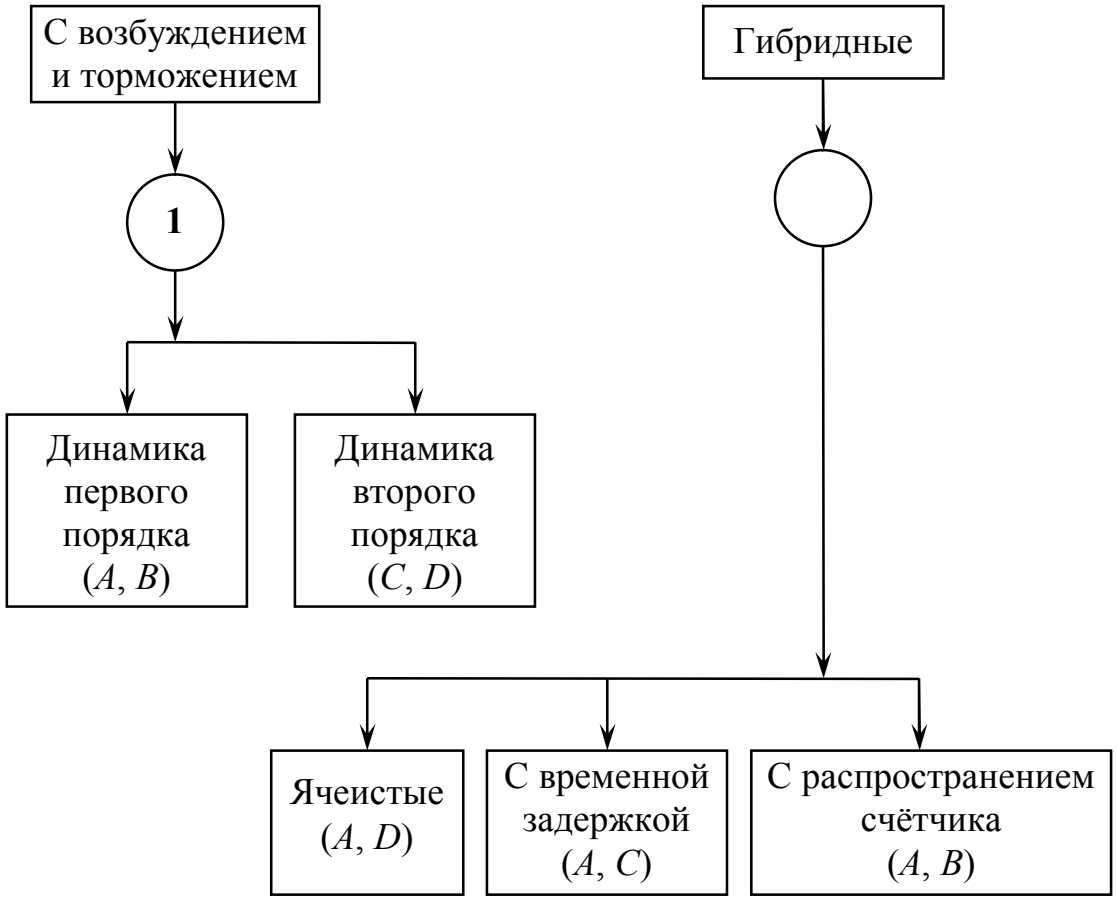


Рис. 5.5. Классификация нейронных сетей (продолжение)

понимание этого процесса. Но следует проявлять осторожность. Возможности обучения искусственных нейронных сетей ограниченны, и нужно решить много сложных задач, чтобы определить, в правильном ли направлении идут исследования. Тем не менее, уже получены убедительные достижения, такие как "говорящая сеть" Сейновского, и возникает много других практических применений [85, 87÷97].

Цель обучения

Сеть обучается, чтобы для некоторого множества входов давать желаемое (или, по крайней мере, сообразное с ним) множество выходов. Каждое такое входное (или выходное) множество рассматривается как вектор.

Обучение осуществляется путём последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определённой процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

Алгоритмы обучения

Существует великое множество различных алгоритмов обучения, которые, однако, делятся на два больших класса: детерминистские и стохастические. В первом из них, подстройка весов представляет собой жёсткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу.

Для конструирования процесса обучения, прежде всего, необходимо иметь модель внешней среды, в которой функционирует нейронная сеть – знать доступную для сети информацию. Эта модель определяет парадигму обучения. Во-вторых, необходимо понять, как модифицировать весовые параметры сети, какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существуют три парадигмы обучения: "с учителем", "без учителя" (самообучение) и смешанная.

Обучение с учителем предполагает (рис. 5.6), что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Разность (ошибка) с помощью обратной связи подаётся в сеть и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются

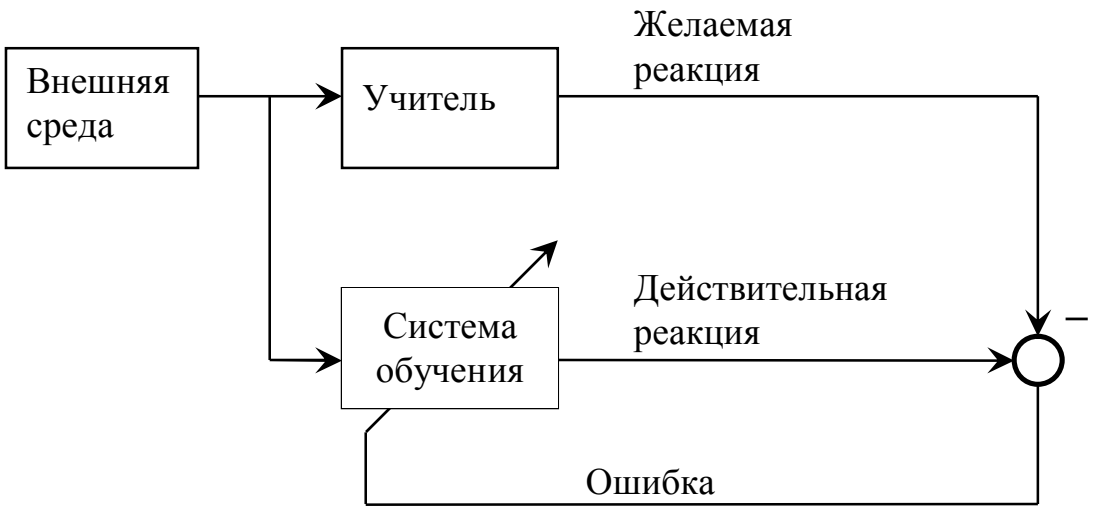


Рис. 5.6. Обучение с учителем

последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня [85].

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность.

Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Если допустить подобный механизм в мозге, то откуда тогда возникают желаемые выходы? Обучение без учителя (рис. 5.7) является намного более правдоподобной моделью обучения в биологической системе. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределёнными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, то есть чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения выделяет статистические

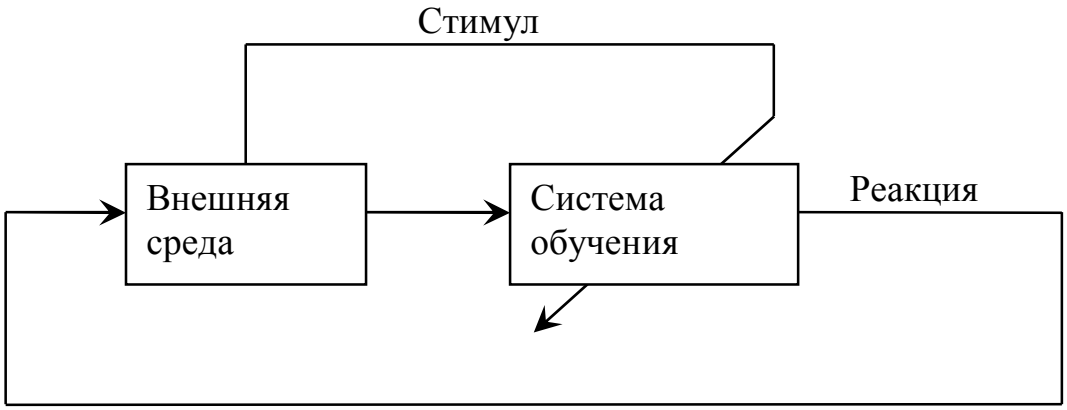


Рис. 5.7. Обучение без учителя

свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определённый выходной вектор. 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 [85, 86].

Большинство современных алгоритмов обучения выросло из концепций Хэбба. Им предложена модель обучения без учителя, в которой синаптическая сила (вес) возрастает, если активированы оба нейрона, источник и приёмник. Таким образом, часто используемые пути в сети усиливаются и феномен привычки и обучения через повторение получает объяснение.

В искусственной нейронной сети, использующей обучение по Хэббу, наращивание весов определяется произведением уровней возбуждения передающего и принимающего нейронов. Это можно записать как

$$w_{ij}(n+1) = w(n) + \mathrm{OUT}_i \, \mathrm{OUT}_j \, ,$$

где $w_{ij}(n)$ – значение веса от нейрона i к нейрону j до подстройки;

$w_{ij}(n+1)$ –значение веса от нейрона i к нейрону j после подстройки;

n – коэффициент скорости обучения;

OUT_i – выход нейрона i и вход нейрона j;

OUT_j – выход нейрона j.

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: ёмкость, сложность образцов и вычислительная сложность. Под ёмкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать "переобученность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо – на тестовых примерах, подчинённых тому же статистическому распределению. Известны четыре основных типа правил обучения: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования [88÷95].

5.2. Выбор оптимального варианта технологического решения с учётом себестоимости научно-технической продукции

Требование к стоимости научно-технической продукции устанавливают предельное значение себестоимости разработки образца, превышение которого приводит к выводу о нецелесообразности выполнения его разработки. Себестоимость научно-технической продукции представляет собой стоимостную оценку используемых в процессе научно-исследовательских и опытно-конструкторских работ (НИОКР), природных ресурсов, сырья, материалов, топлива, энергии, основных фондов, трудовых ресурсов, а также других затрат на выполнение соответствующих работ [100, 103].

Стоимость научно-технической продукции определяется:

$$C_{окр} = C_{мат} + C_{зп} + C_{доп} + C_{ка} ,$$

где $C_{мат}$ – материальные затраты на выполнение работ по выполнению образца;

$C_{зп}$ – затраты на оплату труда;

$C_{доп}$ – дополнительные затраты;

$C_{ка}$ – стоимость работ и услуг производственного характера, выполняемых сторонними предприятиями или производствами.

Требования к стоимости научно технической продукции, разрабатываемой в интересах создания образцов (предельной стоимости научно-исследовательских работ и опытно-конструкторских работ) могут определятся следующими методами: альтернативным, предельно результативным, интегральным, аналоговым и агрегатным.

При расчёте требований к стоимости научно-технической продукции, альтернативным методом в качестве критерия используется соответствие предстоящих полных затрат на создание образца дополнительным затратам по обеспечению требуемого уровня эффективности действующим оборудованием.

Обязательным условием применения альтернативного метода является совпадением множества задач образца, для которого производятся расчёты, и образца, на замену которому он предназначен. В противном случае должен быть произведён анализ возможности выполнения задач, для которых предназначен новый образец.

При расчёте требований к стоимости научно технической продукции предельно-результативным методом используется статистическая связь затрат на разработку и изготовления образцов.

При расчёте требований к стоимости научно-технической продукции интегральным методом в качестве критерия используется условие обеспе-

чения ассигнований, выделяемых на разработку, в таком объёме, чтобы с учётом сходимости закупки установленного количества образцов, не превысить суммарные лимиты ассигнования [101, 102].

При расчёте требований к стоимости научно-технической продукции, аналоговым методом в качестве критерия используется условие обеспечения полной стоимости разработки образца на уровне затрат на разработку его аналога с учётом отличия применяемой элементной базы и условий выполнения работы.

При расчёте требований к стоимости научно-технической продукции агрегатным методом в качестве критерия используется условие обеспечения стоимости разработки образца, не превышающей суммарную предельную стоимость проведения эскизного и технического проектирования, разработки конструкторской документации, а также изготовления и отладки опытного образца.

Определение затрат на математическое обеспечение производится по формуле

$$C_{МО} = C_K N_{ориг} + C_K \left(\frac{100 - \sum_{i=1}^{n} X_i}{100} \right) N_{заим} , \qquad (5.9)$$

где $C_{МО}$ – предельная стоимость разработки математического обеспечения;

C_K – стоимость разработки одной команды, рассчитываемая исходя из норм трудозатрат и стоимости нормо-часа, в соответствии с установленным порядком основания трудовых затрат на предприятии-разработчике;

$C_{ориг}$ – количество разрабатываемых (единичных) оригинальных команд;

X_i – коэффициент, зависящий от этапа разработки программ, на котором производится заимствование программного обеспечения. Принимаются значения коэффициента в зависимости от этапа раз-

работки программного обеспечения в соответствии с табл. 5.1;

$N_{заим}$ – количество заимствованных команд;

n – количество этапов разработки математического обеспечения, предшествующих тому этапу, на котором производится заимствование программного обеспечения [91, 92, 104÷110].

Таблица 5.1

Коэффициенты, применяемые при расчёте стоимости разработки программного обеспечения

Этап разработки программы	**Коэффициент X**
Техническое задание	5,0
Блок схемы	10,0
Программирование	15,0
Транслятор	3,0
Автономная отладка	24,0
Комплексная отладка	25,0
Стыковка программ	2,0
Документирование	6,0
Опытная эксплуатация	8,0
Корректировка	2,0

5.3. Алгоритм формирования нейронных сетей на основе твёрдотельных объектов

На основании анализа требований к техническим характеристикам, условий изготовления и эксплуатации НС, выявление тенденций развития

рассматриваемого класса составляет и корректируется с учётом нормативных документов исходное техническое задание на проектирование. На начальных стадиях проектирования требования технического задания конкретизируются в виде системы ограничений, которым должны удовлетворять характеристики НС, обеспечивающие успешное решение проектной задачи.

Комплекс требований к ПЭП можно представить в виде критериального множества

$$K = \{k_i,\ i = 1, 2, ..., N\},$$

где N – число требований [85].

По заданному вектору требований производится формирование и сравнение альтернативных вариантов проектных решений.

Каждый вариант представляет множество характеристик:

$$X = \{x_j,\ j = 1,2,...,M\}.$$

Отдельные элементы этого множества могут совпадать с соответствующими элементами множества требований, другие могут быть связаны косвенно. В общем виде $M \neq N$.

Формально процесс автоматизированного проектирования НС можно представить как последовательное преобразование некоторого первоначального информационного представления объекта посредством управляющих воздействий проектировщика в конечное состояние, однозначно отображаемое на следующем этапе проектирования в $\{x_j^*\}$, удовлетворяющее $\{k_i\}$.

Реализация технологии автоматизированного проектирования искусственных нейронных сетей предъявляет к разрабатываемой системе комплекс следующих требований:

• Возможность формулировать решаемые проектные задачи из пред-

метной области на различных языках, понятных проектировщику.

- Наличие средств для эффективной корректировки задания на проектирование с использованием простых форм входного языка (таблиц, бланков и т.п.).
- Отсутствие жёстких ограничений на структуру и объём входных данных и формы носителей информации, на которых они хранятся.
- Возможность оперативного подключения к программному обеспечению системы новых модулей и исключение устаревших.
- Представление проектировщику возможностей на основе промежуточных результатов принимать решение о выборе методов для продолжения проектной задачи, а так же изменения значений отдельных параметров в используемом методе решения.
- Возможность в ходе выполнения проектных операций прослеживать значения показателей процесса, свидетельствующих о его эффективности, и в зависимости от их значений корректировать вычислительный процесс.
- Допустимость включения обучающих программ для повышения квалификации проектировщика.

Для решения комплекса поставленных задач построена модель сложных процессов в системе с учётом взаимосвязи всех параметров при детерминированных и стохастических воздействиях [86].

На основе модели сформулирован обобщенный критерий оценки качества НС (рис. 5.8).

Обобщенный критерий K_N включает в себя функциональные, экологические и экономические локальные критерии.

Каждый из перечисленных локальных критериев определяется следующими параметрами:

функциональный – объем и производительность ассоциативной памяти;

экологический – уровень величины излучения изомерных квантовых то-

чек, элементов НС;

экономический – стоимость, окупаемость.

Представим процесс потери качества производительности искусственных нейронных сетей как некоторую абстрагированную математическую модель. Пусть X_1, X_2,..., X_k параметры НС, определяющие состояние, которые являются функциями времени t. Принадлежность состояния X

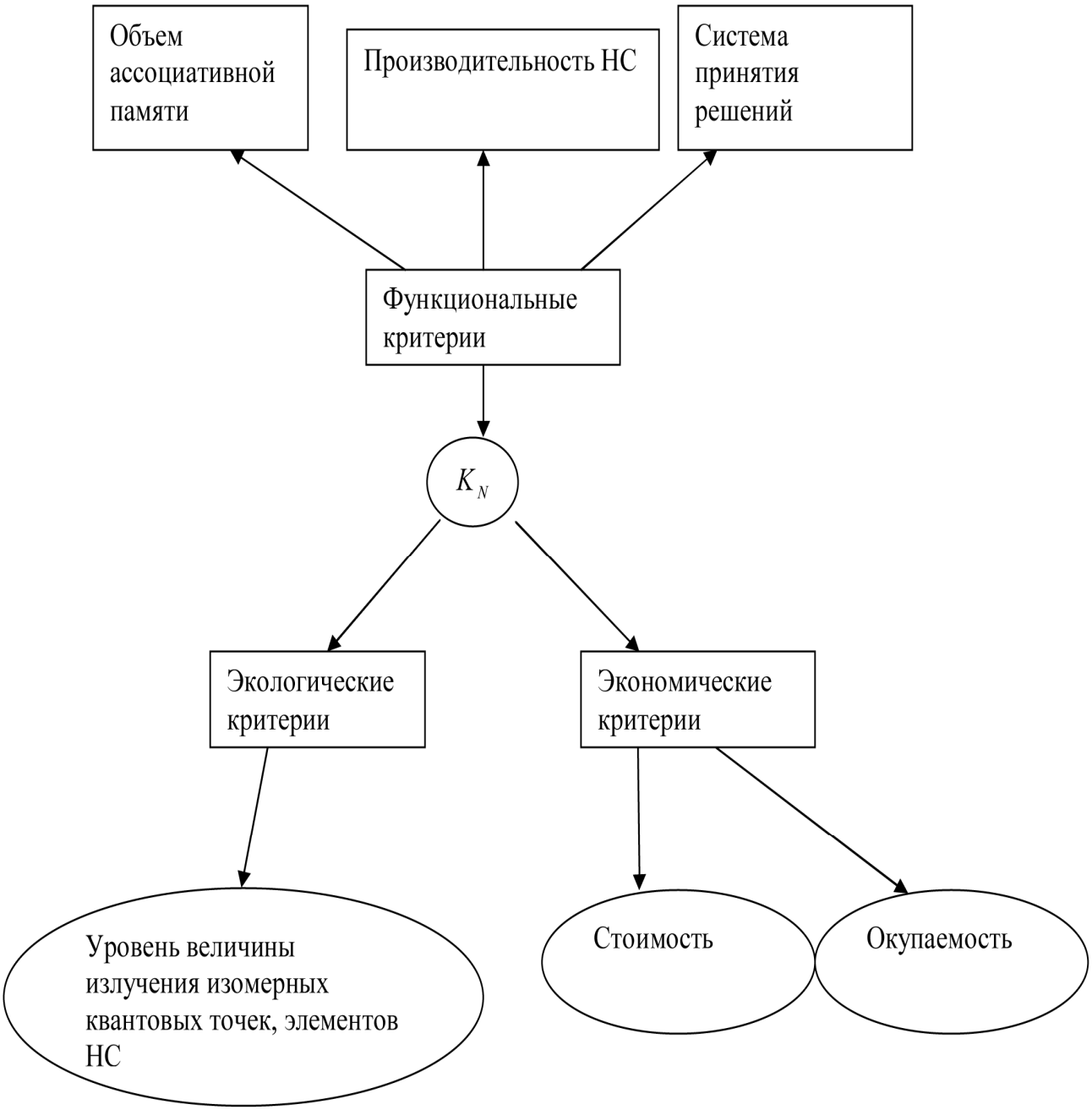


Рис. 5.8. Обобщённый критерий оценки качества нейронных сетей

множеству G_X свидетельствует о том, что НС отвечает критериям качества. Если значения параметров $X_1, X_2,..., X_k$ больше допустимых $X_{1p}, X_{2p},\ldots, X_{kp}$ то есть $X_1 > X_{1p}$, $X_2 > X_{2p}$, $X_k > X_{kp}$, то НС являются не удовлетворяющим параметрам качества. Если некоторые из значений параметров X будут больше допустимых, а другие меньше допустимых, то НС является частично удовлетворяющим параметрам качества. Для условия полного удовлетворения параметрам качества НС – $X_k < X_{kp}$. Это соответствует тому, что множество $G_x \subseteq G_{xp}$.

Алгоритм обучения архитектуры НС представлен на рис. 5.9.

На первом этапе осуществляется получения параметров обучения и протоколирования. При этом нейронная сеть обрабатывает практически любые входные данные. Формируются учебные и тестовые образы для расчёта параметров выходных данных [87].

На второй стадии осуществляем выбор оптимальных архитектур НС, принадлежащих множеству Парето (рис. 5.10).

Особенность состоит в том, что для расчёта параметров выходных данных, используется имитационная модель. Для неё набор данных, информация о внутренней структуре и содержании отсутствует полностью, но известны спецификации входных и выходных данных. Для получения такой информации проведены теоретические и аналитические исследования, изложенные в предыдущих главах.

Рис. 5.9. Алгоритм обучения архитектуры нейронных сетей

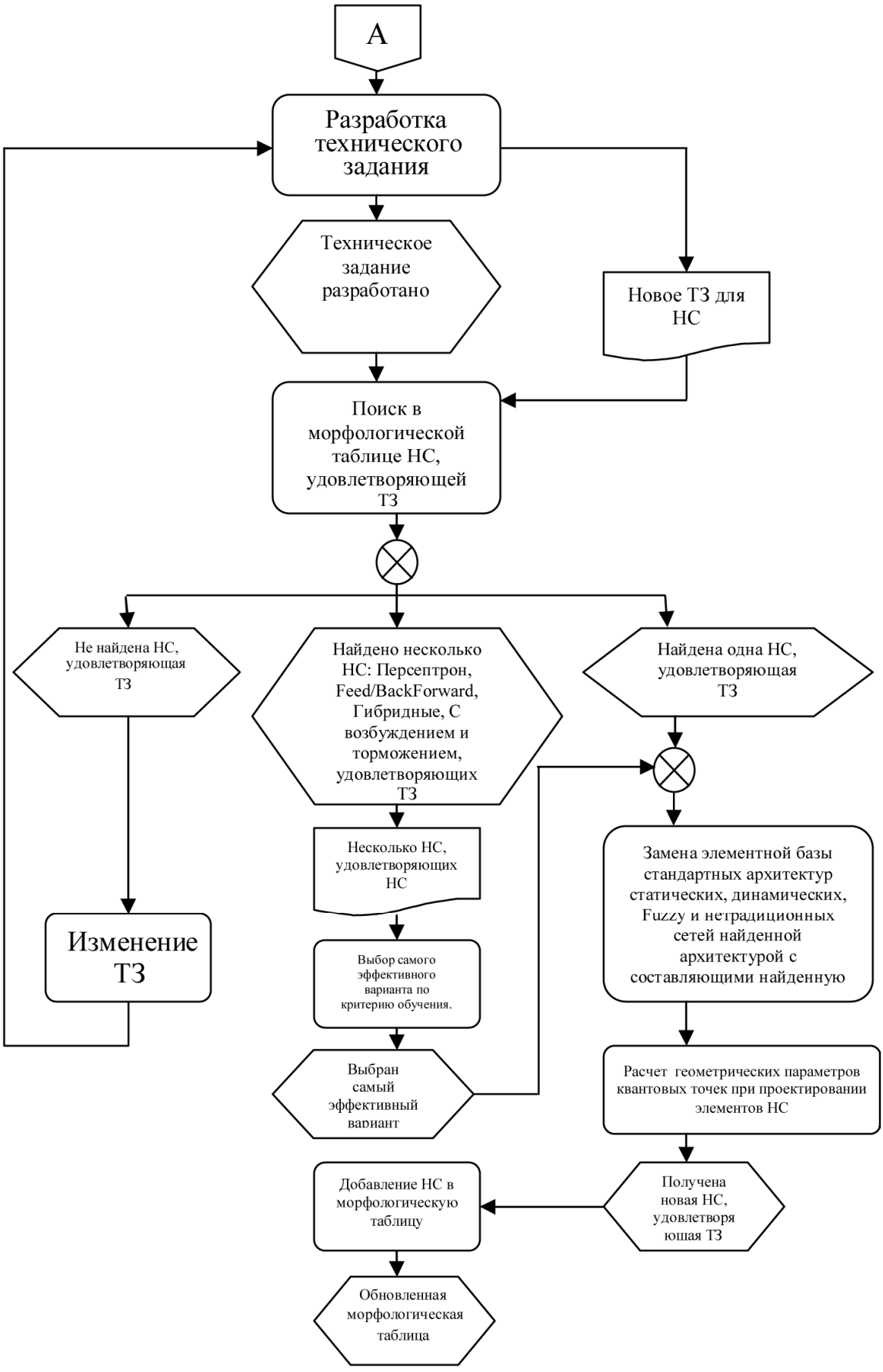


Рис. 5.10. Алгоритм обучения оптимальной архитектуры нейронной сети, удовлетворяющей техническому заданию

ГЛАВА 6. УПРАВЛЕНИЕ КАЧЕСТВОМ РАСПОЗНАВАНИЯ ОБРАЗОВ В КЛАСТЕРНЫХ СИСТЕМАХ ОБРАБОТКИ ИНФОРМАЦИИ

В главе рассматривается математическая модель функционирования кластерных систем обработки информации, алгоритмы и методы моделирования состояния подобных систем во времени, а также метод определения порога функционирования системы с использованием эволюционной стратегии (генетического алгоритма).

Кластер функционирует как единая система, то есть для пользователя или прикладной задачи вся совокупность вычислительной техники выглядит как один компьютер. Именно это и является самым важным при построении кластерной системы (КС). В настоящее время кластерные системы обработки информации получают все большее распространение в связи с удешевлением их компонентов и, как следствие, остро встает вопрос об управлении качеством функционирования подобных систем.

К общим требованиям, предъявляемым к кластерным системам, относятся:

1. Высокая готовность.
2. Высокое быстродействие.
3. Масштабируемость.

6.1. Теоретический анализ кластерных систем

Кластерная система обработки информации описывается:

- множеством состояний $\Theta[a_1,..., a_n]$, где $a_i \in A_i$ – состояние отдельного элемента системы, A – дискретное множество состояний,

которые может принимать *i*-й элемент системы;

- целевой функцией $F\left(\Theta_j\right) = \left[F_1, \ldots, F_m\right]$, где $\Theta_j \in \Theta$ – некоторое состояние системы; $\left[F_1, \ldots, F_m\right]$ – вектор целевых показателей, характеризующий систему в целом;
- архитектурой кластерной системы.

В общем случае кластерную систему можно представить в виде графа, узлы которого представляют собой устройства сбора и обработки информации, а ветви – каналы передачи данных. Наиболее часто такой граф имеет древовидную структуру, представленную на рис. 6.1, где $Д_1, \ldots, Д_n$ – датчики (устройства ввода информации), *РД* – резервные датчики, $ВУ_1, \ldots, ВУ_n$ – вычислительные узлы (устройства обработки информации), *РУ* – резервные узлы, *ЦУ* – центральный узел [117].

Архитектура системы находит отражение в целевой функции, так как целевые показатели вышестоящих элементов напрямую зависят от показателей нижестоящих элементов. Исходя из этого, состояние и целевые показатели *i*-го элемента будут являться функциями от состояний и показателей нижестоящих элементов:

$$a_i = \Theta\left[b_1, \ldots, b_n\right],$$

где a_i – состояние *i*-го элемента;

$\Theta\left[b_1, \ldots, b_n\right]$ – функция перехода состояний, учитывающая весовой коэффициент *j*-го элемента, показывающий его важность для функционирования элемента верхнего уровня и системы в целом:

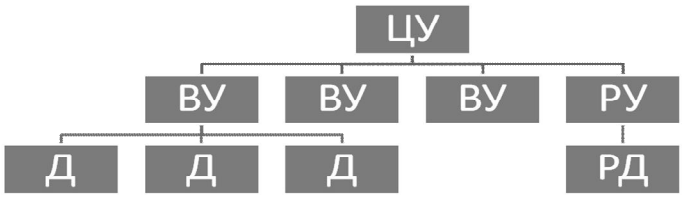


Рис. 6.1. Древовидная структура кластерной системы

$$\Theta\left[b_1,..., b_n\right] = \frac{\sum_{j=1}^{n} k_j b_j}{n},$$

где $b_j \in B$ – состояние j-го элемента нижнего уровня;

k_j – весовой коэффициент элемента.

Целевая функция состояния элемента a_i имеет вид

$$f\left(a_i\right) = \sum_{j=1}^{n} k_j f\left(b_j\right),$$

где $f(b_i)$ – целевая функция состояния для элементов нижнего уровня:

$$f\left(b_j\right) = \left[F_1,..., F_n\right].$$

Рассмотрим целевые показатели обработки информации для кластерных информационных систем.

Кластерную систему можно охарактеризовать следующими показателями:

- Вероятность ложного срабатывания W – это результирующая вероятность программной ошибки в каждом физическом элементе системы.
- Коэффициент доступности (или работоспособности) системы P – обусловлен вероятностью полной недоступности системы в связи с аппаратными или программными неполадками. Этот коэффициент выражает количественную меру работоспособности системы.
- Производительность системы – обусловлена временем, которое затрачивается системой на решение эталонного задания.
- Время отклика – это время, необходимое системе на обработку команды оператора или восприятие новой задачи.

Рассмотрим подробнее каждый целевой показатель.

Чтобы оценить эффективность вероятностных систем обработки ин-

формации на основе математического моделирования, можно использовать метод статистических испытаний. Для проведения таких испытаний может служить математическая модель функционирования системы, принципиальная схема которой представлена на рис. 6.2. Здесь БФРО – блок формирования распознаваемых объектов, БООП – блок ошибок определения признаков, БОАОК – блок ошибок априорного описания классов, БООАИ – блок ограничения объёма апостериорной информации, БР – блок распознавания, БОПЭ – блок оценки показателя эффективности, ДСЧ – датчик случайных чисел.

Принцип действия модели следующий. Для проведения каждого испытания с помощью ДСЧ формируется модель объекта, принадлежность которого к определённому классу заранее известна. Формирование модели объекта производится заданием совокупности числовых значений признаков $x_1,\ldots, x_N$, которые для объектов из класса Ω_i генерируются как реали-

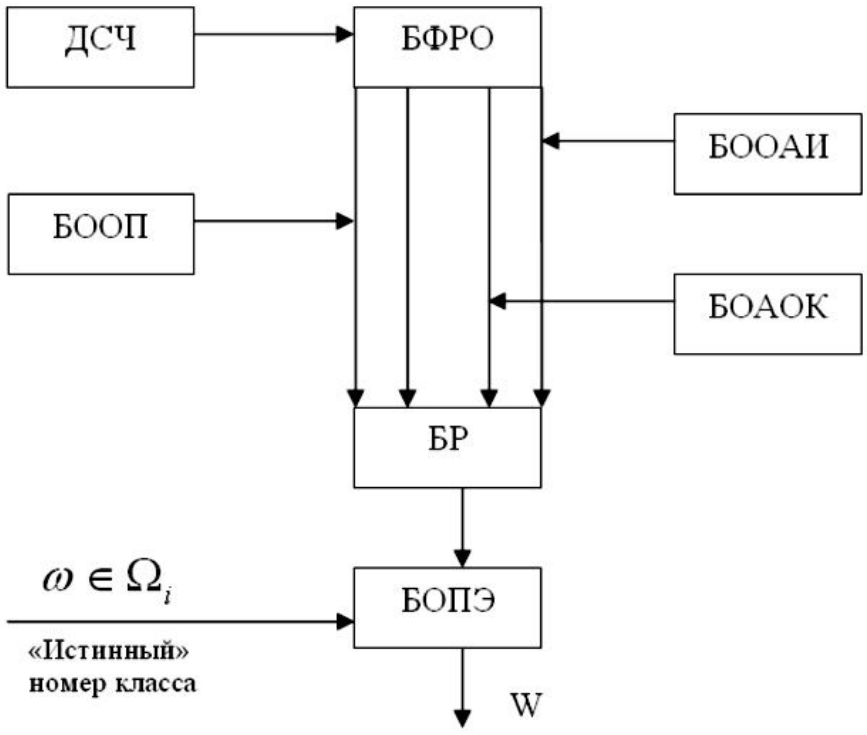


Рис. 6.2. Принципиальная схема системы обработки информации

зации многомерной случайной величины с заданным законом распределения $f_i(x_1,\ldots, x_N)$ по одному из известных алгоритмов.

Числовые значения параметров $x_1,\ldots, x_N$, представляющие собой обрабатываемый объект, подвергаются случайному искажению, что имитирует результат воздействия различных помех в процессе определения признаков $x_1,\ldots, x_N$ при использовании соответствующих технических средств с определёнными точностными характеристиками. Искажённые значения параметров $x'_1,\ldots,x'_N$, представляющие наблюдаемый объект в том виде, в каком его воспринимает система, поступают на вход БР, в котором определяется принадлежность объекта одному из классов $\Omega_1,\ldots, \Omega_N$. Блок БОПЭ сопоставляет номер класса, к которому отнесён объект блоком распознавания, с «истинным» номером, то есть с тем, который задавался на первом этапе формирования объекта, определяет правильность обработки информации и систематизирует соответствующую информацию для подсчёта оценок вероятностей верных и ошибочных решений. При обработке объектов из класса Ω_i оценкой p_i вероятности получения правильного решения служит отношение количества правильных ответов N^i_{np} к общему числу испытаний N^i над объектами класса Ω_i, т.е. $p_i \approx \frac{N^i_{np}}{N^i}$. Число испытаний N^i определяется доверительной вероятностью, задаваемой при формулировке задачи исследования.

В зависимости от задачи исследования искажению могут подвергаться также априорные данные о классах объектов, то есть функции распределения $f_i(x_i,\ldots, x_N)$ и $P(\Omega_i)$, информация о признаках $x_1,\ldots, x_N$ может урезаться, что соответствует отсутствию некоторых средств определения признаков и т.п.

Если априорные вероятности $P(\Omega_i)$ появления объектов из разных классов известны, то безусловная вероятность правильного решения задачи обработки информации данной системой может быть выбрана в качест-

ве критерия эффективности системы обработки информации:

$$W = \sum_{i=1}^{n} p_i P(\Omega_i).$$

Рассмотренная статистическая модель позволяет найти зависимость W от вида и количества привлекаемых для обработки признаков и точности $\sigma_1,\ldots, \sigma_s$ технических средств, которыми оснащается система обработки информации, т.е.

$$W = W(x_1,\ldots, x_N; \sigma_1,\ldots, \sigma_s).$$

Сведения, содержащиеся в этом равенстве – исходные для задач об определении состава технических средств наблюдений системы обработки информации, необходимых точностей их работы, об оптимальном, с точки зрения экономических соображений, распределении точностей по средствам и т.д.

Перейдём к рассмотрению следующего целевого показателя. Коэффициент доступности кластерной системы можно определить с помощью метода соотношений. Суть метода сводится к определению вероятности безотказного функционирования сложной многоуровневой кластерной системы.

Процесс функционирования кластерной системы организован таким образом, что система успешно решает свои задачи при условии, если в исправном состоянии находится хотя бы одно устройство ввода информации, все устройства обработки информации и центральный узел. Данное условие выполнения целевой функции системы можно наглядно представить в форме логической функции:

$$F(\text{КС}) = \left[F_1(Д)\right] \wedge \left[F_2(\text{ВУ})\right] \wedge F_3(\text{ЦУ});$$

$$F_1(Д) = F_1(Д_1 \vee \ldots \vee Д_n); \qquad (6.1)$$

$$F_2(\text{ВУ}) = F_2(\text{ВУ}_1 \wedge \ldots \wedge ВУ_n) \ .$$

Эти выражения означают, что устройство, указанное в скобках, работает исправно.

Представляет интерес также логическая зависимость, описывающая условия невыполнения системой своих целевых функций:

$$\begin{aligned} &\neg F(\text{КС}) = \neg\left[F_1(Д)\right] \vee \neg\left[F_2(\text{ВУ})\right] \vee \neg F_3(\text{ЦУ}) \ ; \\ &F_1(Д) = F_1(Д_1 \wedge \ldots \wedge Д_n) \ ; \\ &F_2(\text{ВУ}) = F_2(ВУ_1 \wedge \ldots \wedge ВУ_n) \ . \end{aligned} \qquad (6.2)$$

Последнее выражение может оказаться более удобным для решения поставленной задачи определения коэффициента доступности с учётом того, что

$$P\left[F(*)\right] = 1 - P\left[\neg F(*)\right] \ ,$$

где $P\left[F(*)\right]$ – вероятность истинности условия $F(*)$;

$P\left[\neg F(*)\right]$ – вероятность истинности отрицания истинности данного условия.

Перечисленные элементы КС имеют различное функциональное назначение и соединены так, что надёжность каждого из них оказывает непосредственное влияние на работоспособность всей системы в целом. Поэтому в качестве факторов для оценки надёжности функционирования КС следует взять вероятности P_i безотказного функционирования устройств в процессе решения системой поставленных задач. В общем случае вероятности P_i могут иметь различные значения. Вероятность $P\left[F(\text{КС})\right]$ безотказного функционирования КС в целом есть функция от вероятностей безотказного функционирования всех её элементов, вытекающая из рассмотренных выше логических условий.

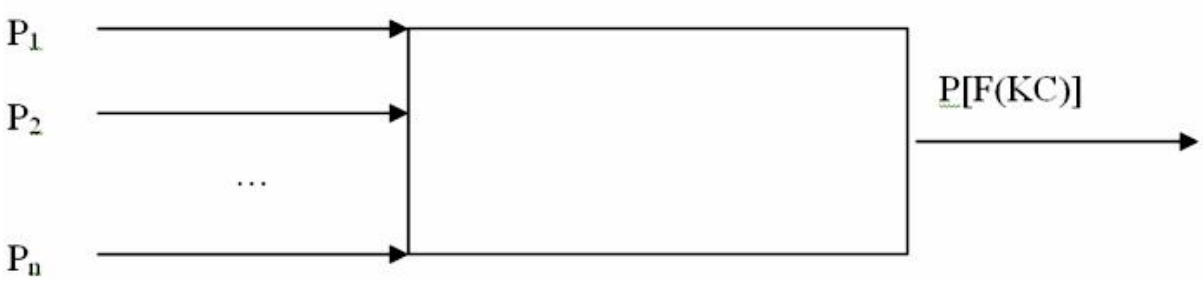


Рис. 6.3. Обобщённая схема математической модели, характеризующей безотказность функционирования КС

Обобщённая схема математической модели, характеризующей безотказность функционирования КС, имеет вид, показанный на рис. 6.3.

Проблема состоит в том, как из логических условий получить соответствующее выражение для количественного значения вероятности $P\left[F\left(\text{КС}\right)\right]$. Дело в том, что вероятность $P\left[F\left(\text{КС}\right)\right]$ определяется на множестве состояний $\Theta(t)$. Число состояний в данном множестве равно $2^n = N$, где n – число структурных элементов КС. Условие функционирования (6.1) определяет подмножество состояний системы, обеспечивающих выполнение системой заданных целевых функций, а условие (6.2) определяет подмножество состояний, в которых система оказывается неработоспособной.

Очевидно, для решения данной задачи таким способом придётся осуществить полный перебор всех N состояний системы или же придумать более эффективный способ определения работоспособных состояний, особенно если учесть, что число состояний системы находится в степенной зависимости от числа её элементов. Наиболее перспективными методами решения этой проблемы представляются метод имитационного моделирования и формализованный переход от логических функций к соответствующим формулам вероятностей сложных событий.

Рассмотрим производительность систем обработки информации. Общая производительность кластерной системы обработки информации

обусловлена производительностью каждого вычислительного элемента системы и определяется экспериментальным путём. Для этого каждому из элементов вычислительной системы дается эталонное задание и определяется время, затраченное на его решение. Исходя из затраченного на решение задачи времени, узлам назначаются весовые коэффициенты, характеризующие производительность вычислительного узла и системы в целом.

В реальных условиях производительность кластерной системы обработки данных зависит не только от производительности вычислительных узлов, но и от надёжности и пропускной способности каналов передачи данных. Таким образом, ко времени, затрачиваемому на решение эталонного задания всей системой, добавляется время, необходимое системе на подтверждение принятия задания, и время, затрачиваемое системой на передачу данных между вычислительными узлами к центру.

Производительность систем в реальных условиях можно вычислить следующим образом:

$$П = \sum_{i=1}^{M} w_i + 2L \ ,$$

где $П$ – производительность системы;

w_i – весовой коэффициент производительности вычислительного узла;

N – общее количество элементов системы;

M – количество вычислительных узлов;

L – время прохождения сигнала по каналам связи, определяемое по формуле

$$L = \min\left\{ \sum_{j=1}^{N-M} w_j \right\} ,$$

в которой w_j – весовой коэффициент пропускной способности канала связи.

Одним из важных параметров, описывающих кластерную систему обработки информации, является порог функционирования, то есть такое

значение целевой функции, при переходе через которое система перестает функционировать. Для относительно простых систем обработки информации это значение может быть получено экспериментальным или эмпирическим путём. Однако для систем с большим числом разнородных элементов это представляется затруднительным. Выходом в подобной ситуации может быть моделирование системы с использованием эволюционной стратегии, где критерием отбора будет являться наиболее функциональное состояние системы при максимальном количестве неисправностей. Эволюционные алгоритмы базируются на коллективном обучающем процессе внутри популяции индивидуумов, каждый из которых представляет собой поисковую точку в пространстве допустимых решений данной задачи [118]. Наиболее известными из класса эволюционных алгоритмов являются генетические алгоритмы. Генетический алгоритм (рис. 6.4) может быть легко применён для безусловной оптимизации функций, то есть для задачи отыскания значений параметров, которые минимизируют или максимизируют заданную целевую функцию и для безусловной комбинаторной оптимизации, то есть для задачи отыскания наилучшей комбинации вариантов, которая оптимизирует заданную целевую функцию. Их основные адаптивные процессы концентрируются на идее системы, получающей сенсорную информацию от окружающей среды через бинарные детекторы. В генетических алгоритмах существует строгое различие между фенотипом (решением) и генотипом (представлением решения). Генетический алгоритм работает только с генотипом, поэтому требуется процесс декодирования генотипа в фенотип и обратно («обобщенный» рост). Вещественные параметры могут быть представлены числами с фиксированной точкой или целыми числами путём масштабирования и дискретизации. Для вещественных параметров имеет место конфликт между желанием иметь как можно более короткий ген для обеспечения хорошей сходимости и необходимостью получить результат с определённой точностью.

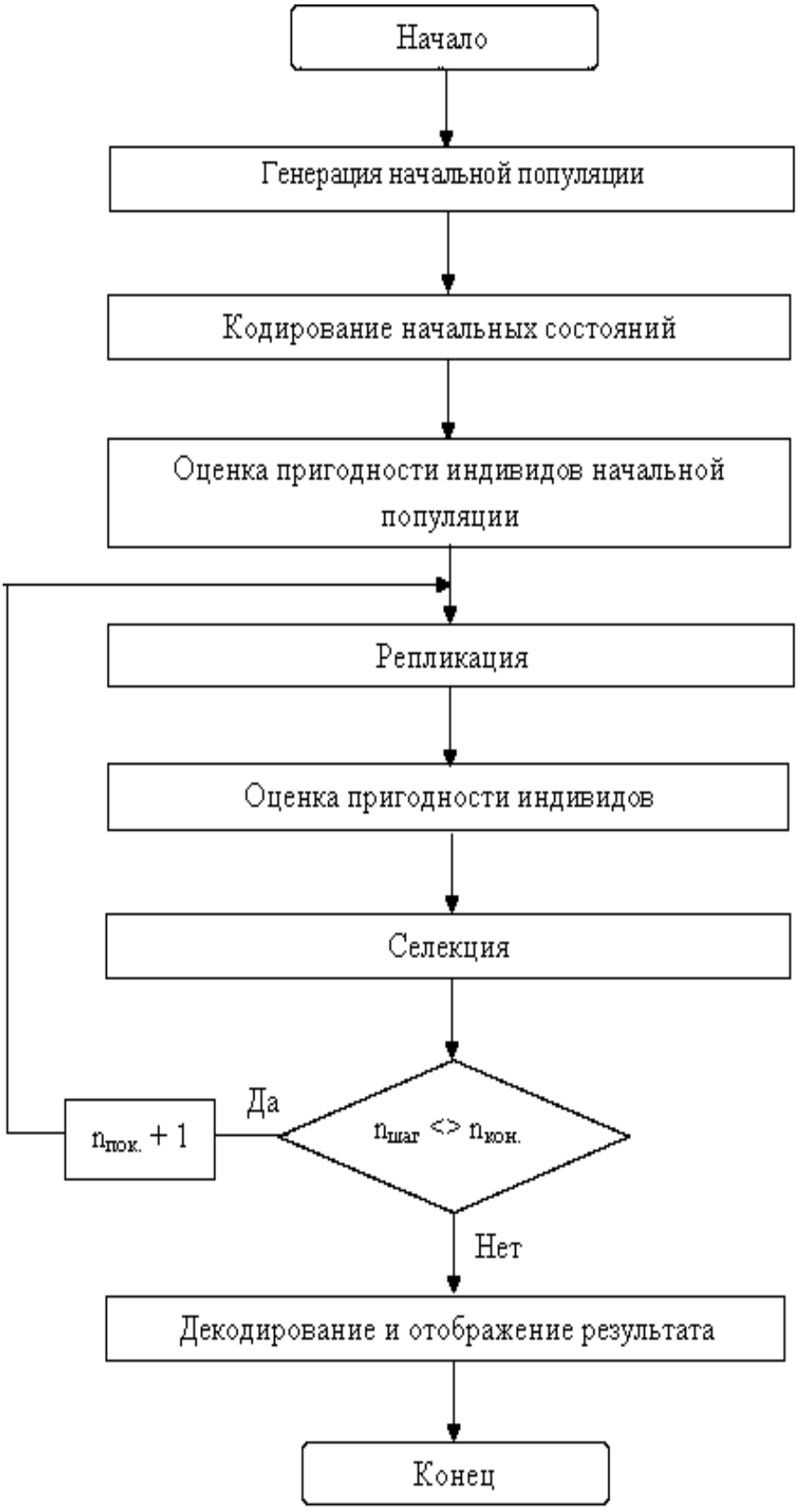


Рис. 6.4. Генетический алгоритм

6.2. Комментарии к генетическому алгоритму

Шаг **1**. Генерация начальной популяции. Случайным образом генерируется n уникальных состояний системы (индивидов), для каждого состояния вычисляется значение целевой функции и показатель работоспособности системы.

Шаг **2**. Кодирование состояний системы в бинарный код (составление хромосом).

Шаг **3**. Оценка пригодности каждого состояния. Для этого состояния ранжируются по значениям показателя работоспособности.

Шаг **4**. Репликация состояний, то есть генерация новой популяции: из $m < n$ состояний попарно генерируются потомки. В нашем случае потомком будет являться результирующее состояние, являющееся следствием событий состояний-родителей.

Шаг **5**. Оценка пригодности всех состояний, включая потомков.

Шаг **6**. Селекция. Для имитации естественной селекции состояния с более высокой пригодностью должны выбираться с большей вероятностью, поэтому из получившихся состояний выбирается n самых пригодных.

Шаг **7**. Проверка конечного условия: если номер поколения $n_{пок.}$ не равен заложенному на этапе инициализации конечному числу поколений $n_{кон.}$, то увеличение $n_{пок.}$ на единицу и переход на Шаг **4**.

Если $n_{пок.} = n_{кон.}$, то переход на Шаг **8**.

Шаг **8**. Декодирование и отображение полученного результата.

6.3. Кластеры повышенной производительности

Кластеры повышенной производительности обозначаются англ. аббревиатурой HPC (*High performance cluster*). Позволяют увеличить ско-

рость расчётов, разбивая задание на параллельно выполняющиеся потоки. Используются в научных исследованиях. Одна из типичных конфигураций – набор серверов с установленной на них операционной системой Linux. Такую схему принято называть кластером Beowulf. Для HPC создаётся специальное ПО, способное эффективно распараллеливать задачу.

Эффективные связи между серверами в кластере позволяют им поддерживать связь и оперативно обмениваться данными, поэтому такие кластеры хорошо приспособлены для выполнения процессов, использующих общие данные.

Физическая схема

- План стойки для каждого их типа (например, управляющие и вычислительные стойки).
- Поэтажный план расположения стоек во время процесса установки системы и при рабочем использовании, если они отличаются.
- Схемы внутренних соединений стоек для сети, цепей питания, пульта оператора и т.д.
- Схема внешних соединений для серверов системы хранения, терминальных серверов и т.д.

Логическая схема

- Схема сети, включая диапазоны IP-адресов, конфигурацию подсетей, соглашения по наименованию компьютеров и т.д.
- CSM-конфигурация по расположению пользовательских сценариев, аппаратные настройки и требования по мониторингу.
- Требования к операционным системам, список специализированных

пакетов и параметры конфигурации системы.

- Схема системы хранения данных, включая схему файловой системы, разбиение дисков, параметры репликации и т.д.

Кластер (рис. 6.5) состоит из компьютеров, работающих на процессорах Intel или AMD, с подключёнными подсистемами TotalStorage.

Для простоты соединения в кластере выполнены медным кабелем стандарта гигабитный Ethernet. Этот кабель обеспечивает хорошую скорость в большинстве случаев.

Сетевая топология имеет форму звезды – все стойки подключены к основному коммутатору управляющей стойки. В примере используется

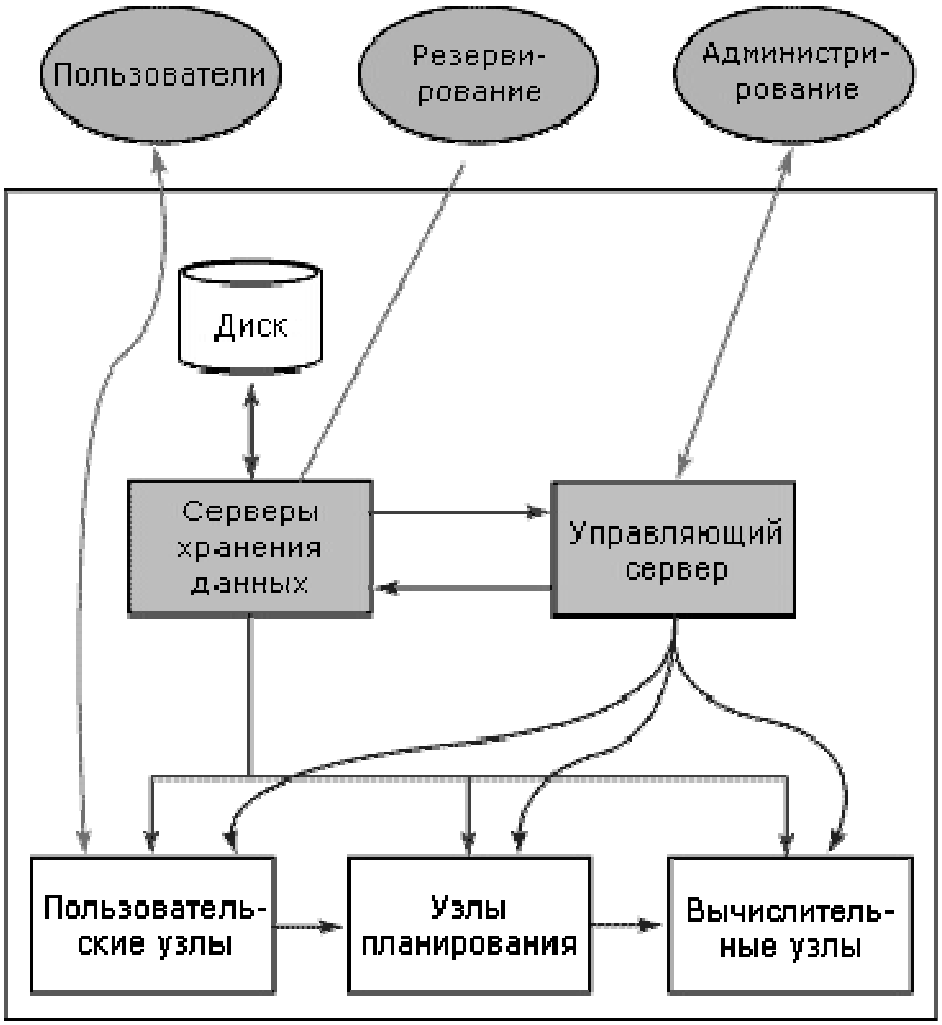


Рис. 6.5. Пример кластера

три сети: одна для управления/данных (вычислительная сеть), одна для кластерной файловой системы (сеть хранения данных) и одна для администрирования устройств. Первые две сети – это обычные IP-сети. Для большинства задач, включая межпроцессные взаимодействия (например, MPI) и управление кластером, используется вычислительная сеть. Сеть хранения данных используется исключительно для доступа и взаимодействия с кластерной файловой системой.

Управляющий сервер

Функция управляющего сервера может выполняться одним сервером или несколькими. В среде с одним сервером управляющий сервер функционирует в автономном режиме. Можно настроить также управляющие серверы с высокой готовностью. Для этого можно использовать программное обеспечение CSM для поддержки высокой готовности (high-availability -HA), которое будет выдавать тактовые импульсы ("heartbeat") между двумя серверами и поддерживать динамическое восстановление после сбоев при возникновении аварийных ситуаций. Другим возможным методом организации нескольких управляющих серверов является использование репликации, если поддержка HA не важна для вашей среды. В этом случае вы можете резервировать данные управляющего сервера на другую рабочую систему, которую можете, при необходимости, перевести в оперативный режим вручную. Управляющий сервер – это CSM-сервер, использующийся исключительно для внутреннего управления кластером при помощи CSM-функций: управление установкой системы, мониторинг, облуживание и другие задачи. В данном кластере присутствует только один управляющий сервер.

Серверы хранения данных и дисковые накопители

Можно подключить несколько серверов хранения данных к организованному на дисковых накопителях хранилищу данных при помощи различных механизмов. Подключить систему хранения данных к серверу можно напрямую: либо через SAN-коммутатор (storage area network – сеть хранения данных) по оптическому волокну или медному кабелю, либо используя оба типа соединений (см. рис. 6.5). Эти серверы предоставляют совместный доступ к системе хранения данных другим серверам кластера. Если необходимо резервирование базы данных, подключите резервное устройство к серверу хранения данных, используя дополнительное медное или оптическое соединение. В примере кластера хранилище представляет собой единую сущность, обеспечивающую доступ к общей файловой системе в пределах кластера.

Пользовательские узлы

В идеальном случае вычислительные узлы кластера не должны принимать внешние подключения. Они должны быть доступны только для системных администраторов через управляющий сервер. Пользователи системы могут зарегистрироваться на вычислительных узлах (или узлах регистрации) для выполнения своей работы в кластере. Каждый пользовательский узел состоит из образа с возможностями любого редактирования, необходимых библиотек разработчика, компиляторов и всего, что необходимо для создания кластерного приложения и получения реальных результатов.

Узлы планирования

Для запуска рабочей нагрузки на кластере пользователи должны передать свою работу узлу планирования. Фоновый процесс-планировщик (scheduler daemon), работающий на одном или нескольких узлах планирования, применяет предопределённую политику для запуска рабочих нагрузок в кластере. Аналогично вычислительным узлам, узлы планирования не должны принимать внешних подключений от пользователей. Системные администраторы должны управлять ими при помощи управляющего сервера.

Вычислительные узлы

Эти узлы выполняют рабочую нагрузку кластера, принимая задания от планировщика. Вычислительные узлы – это самые свободные части кластера. Системный администратор может легко переустанавливать или перенастраивать их при помощи управляющего сервера.

Ethernet-коммутаторы

Имеется две физических сети: одна для вычислений, а вторая для хранения данных. Стандартная ёмкость стойки в 32 узла требует применения двух 48-портовых коммутаторов на каждую стойку, по одному на каждую сеть. В маленьких кластерах в управляющей стойке также необходимо использовать два одинаковых коммутатора. Для больших кластеров 48-портов может оказаться недостаточно, и может потребоваться более мощный центральный коммутатор.

Терминальные серверы играют важную роль в больших кластерных

системах, использующих версии CSM, ниже чем CSM 1.4. Кластеры, использующие старые версии, нуждаются в терминальных серверах для сбора MAC-адресов при установке. При совместимости CSM и системных UUID, терминальные серверы становятся не так важны для установки более современного кластера. Однако если в большом кластере у вас имеется немного устаревшее оборудование или программное обеспечение, терминальные сервера все ещё остаются жизненно важными во время установки системы. Обеспечение корректной настройки терминального сервера само по себе может сэкономить значительное количество времени в дальнейшем в процессе установки системы. Кроме сбора MAC-адресов терминальные серверы могут также использоваться для просмотра терминалов из одной точки во время процедуры начального самотестирования (POST) и запуска операционной системы.

Когда компьютер получает DHCP-адрес во время PXE, конфигурационные файлы в **/tftpboot/pxelinux.cfg** ищутся в определённом порядке, и первый найденный файл используется в качестве загрузочной конфигурации для запрашивающего компьютера. Порядок поиска определяется путём преобразования запрашиваемого DHCP-адреса в шестнадцатиричные цифры и поиска первого подходящего имени файла в конфигурационном каталоге методом расширения подсетей – удаления цифр справа налево на каждом цикле поиска [119].

6.4. Коммуникационные библиотеки

Наиболее распространенным интерфейсом параллельного программирования в модели передачи сообщений является MPI. Рекомендуемая бесплатная реализация MPI – пакет MPICH, разработанный в Аргоннской Национальной Лаборатории. Для кластеров на базе коммутатора Myrinet разработана система HPVM, куда также входит реализация Message

Passing Interface (MPI, интерфейс передачи данных).

MPI – программный интерфейс (API) для передачи информации, который позволяет обмениваться сообщениями между компьютерами, выполняющими одну задачу.

MPI является наиболее распространённым стандартом интерфейса обмена данными в параллельном программировании, существуют его реализации для большого числа компьютерных платформ. Основным средством коммуникации между процессами в MPI является передача сообщений друг другу. Стандартизацией MPI занимается MPI Forum. В стандарте MPI описан интерфейс передачи сообщений, который должен поддерживаться как на платформе, так и в приложениях пользователя. В настоящее время существует большое количество бесплатных и коммерческих реализаций MPI. Существуют реализации для языков Фортран 77/90, Си и Си++.

6.5. Стандарты MPI

Большинство современных реализаций MPI поддерживают версию 1.1. Стандарт MPI версии 2.0 поддерживается большинством современных реализаций, однако некоторые функции могут быть реализованы не до конца.

В MPI 1.1 поддерживаются следующие функции:

- передача и получение сообщений между отдельными процессами;
- коллективные взаимодействия процессов;
- взаимодействия в группах процессов;
- реализация топологий процессов;

В MPI 2.0 дополнительно поддерживаются следующие функции:

- динамическое порождение процессов и управление процессами;
- односторонние коммуникации;
- параллельный ввод и вывод.

Для эффективной организации параллелизма внутри одной SMP-системы возможны два варианта.

1. Для каждого процессора в SMP-машине порождается отдельный MPI-процесс. MPI-процессы внутри этой системы обмениваются сообщениями через разделяемую память (необходимо настроить MPICH соответствующим образом).
2. На каждой машине запускается только один MPI-процесс. Внутри каждого MPI-процесса производится распараллеливание в модели "общей памяти", например, с помощью директив OpenMP.

После установки реализации MPI имеет смысл протестировать реальную производительность сетевых пересылок.

Кроме MPI, есть и другие библиотеки и системы параллельного программирования, которые могут быть использованы на кластерах.

Следует понимать, что использование для программирования OpenMosix – расширение (патч) ядра Linux, позволяющее создать единый кластер. Превращает сеть обычных персональных компьютеров в суперкомпьютер для Linux-приложений. Представляет собой полнофункциональную кластерную среду с единой операционной системой (SSI), автоматически распараллеливающую задачи между однородными узлами. Это позволяет миграцию процессов между машинами – узлами сети.

Кластер ведёт себя подобно SMP-машине (за исключением любых видов разделяемой памяти). При этом возможно наращивание до тысяч узлов, которые тоже могут быть SMP-машинами. Добавление новых узлов возможно параллельно работе кластера, добавленные ресурсы будут задействованы автоматически. OpenMosix также предоставляет оптимизированную файловую систему (OMFS) для HPC-приложений, которая, в отличие от NFS, поддерживает кэширование, отметки о времени и ссылки.

OpenMosix – это проект, являющийся продолжением проекта MOSIX, но под свободной лицензией GPL. Последние релизы MOSIX ста-

ли проприетарными (закрытыми) в конце 2001 года, а проект openMosix стартовал 10 февраля 2002. Инициатор проекта – Moshe Bar.

OpenMosix поставляется с набором утилит для администрирования кластера. Для этого имеется также удобное GUI-приложение openMosixview.

При обработке результатов сканирующей зондовой микроскопии часто встает вопрос о принадлежности объекта исследования тому или иному классу объектов. Подобные задачи решаются применением систем распознавания.

Распознавание представляет собой задачу преобразования входной информации, в качестве которой уместно рассматривать некоторые параметры, признаки распознаваемых объектов, в выходную, представляющую собой заключение о том, к какому классу объектов принадлежит распознаваемый объект.

Важнейшей характеристикой принимаемых в процессе распознавания решений и основным показателем качества распознавания является достоверность [124, 125].

В многоуровневых системах распознавания апостериорная информация о признаках определяется на основе косвенных измерений. Для таких измерений используются специализированные локальные системы распознавания. По данным технических средств $T1,..., Tp,..., Tn$ определяются признаки $x11,..., xk1;\ xp1,..., xlp;\ x1n,..., xmn$ (первичные признаки), которые используются локальными системами распознавания для определения признаков более высокого ровня, которые, в свою очередь, используются в процессе распознавания неизвестных объектов (рис. 6.6) [120, 121].

Результаты полученные при исследовании объектов, подвергаются статистическому анализу и фильтрации. Исключаются аномальные значения. Далее модель, полученная после фильтрации и статистического анализа в модуле распознавания нижнего уровня (Gauge system, рис. 6.7) [122],

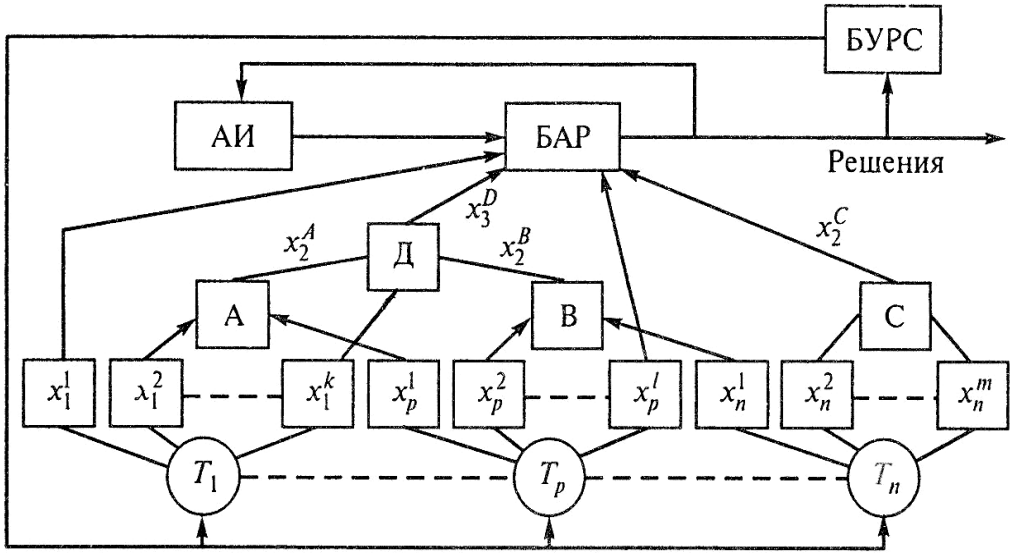


Рис. 6.6. Сложная многоуровневая система распознавания

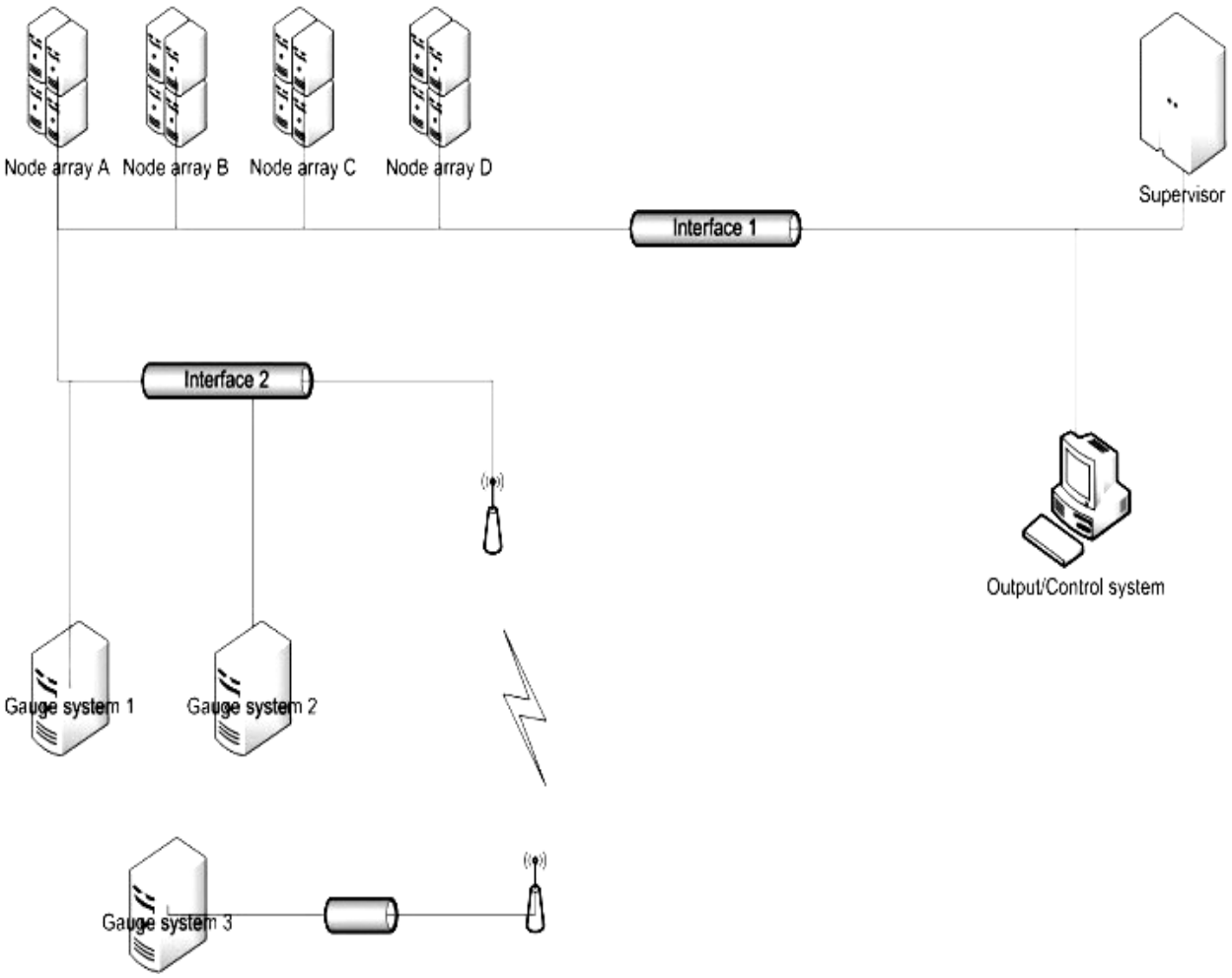


Рис. 6.7. Пример архитектурного решения кластера

представляется в виде карты высот. Карта высот подвергается нормализации и направляется в модуль распознавания верхнего уровня (Node array, см. рис. 6.7), где на основе данных от различных систем нижнего уровня происходит распознавание и моделирование свойств объекта. Результаты исследования, а также диагностическая информация представляются пользователю посредством системы контроля и управления [123] (Output / Control system, см. рис. 6.7).

6.6. Оценка достоверности

Оценка достоверности результатов моделирования может быть произведена различными методами, в частности, непараметрическим.

Непараметрический метод выявления отклонений заключается в ранжировании анализируемых значений $X1 <= X2 <= ... <= Xn$ и вычислениистатистики r для крайних значений [126]. Формулы расчёта зависят от числа анализируемых значений n и от того, значение с какого края проверяется на аномальность (самое большее или меньшее) (табл. 6.1).

Таблица 6.1

Формула для работы отклонений

Число значений	**Проверяется максимальное значение *Xn***	**Проверяется минимальное значение *X*1**
3<=<*n*<=7	$r10=(Xn-X(n-1))/(Xn-X1)$	$(X2-X1)/(Xn-X1)$
8<=*n*<=10	$r11=(Xn-X(n-1))/(Xn-X2)$	$(X2-X1)/(X(n-1)-X1)$
11<=*n*<=13	$r20=(Xn-X(n-2))/(Xn-X2)$	$(X3-X1)/(X(n-1)-X1)$
14<=*n*<=25	$r21=(Xn-X(n-2))/(Xn-X3)$	$(X3-X1)/(X(n-2)-X1)$

Полученное значение r сравнивается с критическим r и считается аномальным, если $r > ar$.

После этого процедура проверки повторяется для оставшихся n –1 значений.

Данный критерий оптимален для малых серий наблюдений (данных) и не зависит от числа имеющихся аномальных значений. В то время, как приведённый ранее параметрический критерий (особенно для небольших серий) оптимален, когда имеется лишь одно аномальное значение.

Рассматриваемая математическая модель в совокупности с эволюционной стратегией позволяет оценивать критические ситуации для кластерных систем обработки информации и выявлять их последствия, а также моделировать и оптимизировать адаптивными методами показатели качества подобных систем [128].

ГЛАВА 7. КВАНТОВЫЕ НАНОРАЗМЕРНЫЕ СТРУКТУРЫ ДЛЯ СИСТЕМ КОДИРОВАНИЯ И КРИПТОГРАФИИ

Проблема безопасности информационных технологий возникла на пересечении двух активно развивающихся и, одних из самых передовых, в плане использования технических достижений, направлений – безопасности технологий и информатизации. Сама проблема безопасности не является новой, так как обеспечение собственной безопасности – задача первостепенной важности для любой системы независимо от её сложности и назначения, будь то социальное образование, биологический организм или система обработки информации.

В жизни современного общества востребовано множество информационных технологий. Компьютеры обслуживают банковские системы, контролируют работу атомных реакторов, распределяют энергию, следят за расписанием поездов, управляют самолетами, космическими кораблями. Компьютерные сети и телекоммуникации предопределяют надёжность и мощность систем обороны и безопасности страны. Компьютеры обеспечивают хранение информации, её обработку и предоставление потребителям, реализуя информационные технологии.

Именно высокая степень автоматизации порождает риск снижения безопасности (личной, информационной, государственной и т.п.). Доступность и широкое распространение информационных технологий, ЭВМ делает их чрезвычайно уязвимыми по отношению к деструктивным воздействиям, в том числе и информационным. Тому есть множество примеров. Информационная безопасность, а главное надёжность и достоверность информации становится сегодня всё более и более определяющими параметрами при взаимодействии технологических систем друг с другом.

Следовательно, чтобы быть защищенной, система должна успешно противостоять многочисленным и разнообразным угрозам безопасности, действующим в пространстве современных информационных технологий, и главным образом тем, которые носят целенаправленный характер.

На каждом этапе развития существуют инструментальные и экономические ограничения, выражающиеся в уровне и качестве продукции на данном этапе развития цивилизации. Прогресс в познании строения вещества неукоснительно приводит к созданию новых научных открытий и порождаемых ими новых научных технологий.

С появлением нанотехнологий появилась техническая возможность сдвинуть ограничения на пространственное разрешение измерительных и исполнительных инструментов в нанометровую и атомарную область размеров. Это создало предпосылки развитии в направлениях нанотехнологии, молекулярной нанотехнологии, наноэлектроники, базирующихся на возможности оперировать с веществом на уровне молекул, молекулярных кластеров и отдельных атомов.

Прогресс в познании строения вещества неразрывно связан с возможностью визуализации описывающих его параметров с максимально осуществимым пространственным и временным разрешением. Следующим шагом познания является попытка использования полученных знаний для построения новых функциональных структур, максимально возможной информационной мощности, улучшения качества производимых продуктов, создания новых технологий.

Одним из наиболее мощных средств для исследования и проектирования технических систем является моделирование. Использование моделирования, начиная с ранних стадий, и постепенное накопление информации за счёт уточнения и детализации модели позволяет говорить о расширяемой адаптивной модели всего цикла проектирования. Соответственно, при анализе различных свойств объекта проектирования (ОП) модельное

представление должно формироваться наиболее подходящим для этой цели образом, независимо от конкретного процесса или этапа проектирования, и сохранять все требуемые свойства проектируемого объекта.

Для развития субмикронной и нанотехнологии, в отличие от традиционной технологии, характерен "индивидуальный" подход, при котором внешнее "управление" достигает отдельных атомов и молекул, что позволяет создавать из них как "бездефектные" материалы с принципиально новыми физико-химическими свойствами, так и новые классы устройств с характерными нанометровыми размерами – наноразмерные структуры.

Одним из направлений решения этой проблемы является создание и развитие автоматизированных систем проектирования различных нанотехнологических процессов, в том числе формирование элементов наноразмерных структур на основе квантовых точек для систем кодирования и криптографии.

Инструментальный базис нанотехнологий, позволяющий учёным и исследователям не только визуализировать атомные структуры, но и манипулировать отдельными атомами и строить новые молекулы, основан на использовании так называемого эффекта туннелирования электронов. Его применение на вершинах зондов специальных конструкций позволяет достигать высокой пространственной разрешающей способности управления атомно-молекулярными реакциями в отличие от известных групповых технологий осаждения материалов, методов оптической литографии, эпитаксии, а также электронной литографии, где высокая энергия фокусируемых электронов приводит к значительному разрушению используемых материалов.

Поэтому разработка элементов автоматизированной системы проектирования процесса формирования наноразмерных структур для систем кодирования и криптографии в туннельно-зондовой нанотехнологии является задачей актуальной и своевременной.

7.1. Основные концепции построения «защищённых» систем

7.1.1. Понятие «защищённая система». Определение и свойства

Многие полагают, что защищённая технологическая система – это система обработки информации, в состав которой включён тот или иной набор средств защиты. Очевидно, что это неправильный подход, так как наличие средств защиты является лишь необходимым условием и не может рассматриваться в качестве критерия защищённости системы от реальных угроз, поскольку безопасность не является абсолютной характеристикой и может рассматриваться только относительно некоторой среды, в которой действуют определённые угрозы. Поэтому считается, что: *защищённая система обработки информации для определенных условий эксплуатации обеспечивает безопасность (конфиденциальность и целостность) обрабатываемой информации и поддерживает свою работоспособность в условиях воздействия на неё заданного множества угроз* [129].

Взяв за основу предложенное определение, рассмотрим свойства, которыми должна обладать защищённая система.

Как и все автоматизированные (компьютерные) системы обработки информации, защищённые системы решают задачу автоматизации некоторого процесса обработки информации. Под процессом обработки информации понимаются действия, связанные с её хранением, преобразованием и передачей. В дополнение к этому, кроме традиционных свойств, которыми обладают автоматизированные системы – надёжности, эффективности, удобства использования и так далее, защищённая система обработки информации должна обладать ещё одним – свойством безопасности, кото-

рое является для неё самым главным. Наконец, поскольку проблема безопасности компьютерных систем изучается и прорабатывается уже достаточно давно, защищённая система должна соответствовать сложившимся требованиям и представлениям. Кроме того, необходимо обеспечить возможность сопоставления параметров и характеристик защищённых систем для того, чтобы их можно было сравнивать между собой.

Таким образом, под защищённой системой обработки информации предлагается понимать систему, которая обладает следующими тремя свойствами:

- осуществляет автоматизацию некоторого процесса обработки конфиденциальной информации, включая все аспекты этого процесса, связанные с обеспечением безопасности обрабатываемой информации;
- успешно противостоит угрозам безопасности, действующим в определённой среде;
- соответствует требованиям и критериям стандартов информационной безопасности.

Предложенный подход к определению понятия "защищённая система" отличается от существующих в первую очередь тем, что рассматривает проблему обеспечения безопасности систем как лежащую на стыке двух направлений: автоматизации обработки информации и общей безопасности. Это даёт возможность объединить задачи автоматизации обработки конфиденциальной информации и разработки средств защиты в одну проблему создания защищённых информационных систем и процессе её решения, применять методы и технологии, разработанные как в той, так и в другой области.

7.1.2. Стандарты безопасности «защищённых систем»

Безопасность является качественной характеристикой системы. Её нельзя измерить в каких-либо единицах, более того, нельзя даже с однозначным результатом сравнивать безопасность двух систем – одна будет обладать лучшей защитой в одном случае, другая – в другом. Кроме того, у каждой группы специалистов, занимающихся проблемами безопасности информационных технологий, имеется свой взгляд на безопасность и средства её достижения, а следовательно, и своё представление о том, что должна представлять собой защищённая система. Хотя любая точка зрения имеет право на существование и развитие, но для того, чтобы объединить усилия всех специалистов в направлении конструктивной работы над созданием защищённых систем всё-таки необходимо определить, что является целью исследований, что мы хотим получить в результате и чего в состоянии достичь?

Для ответа на эти вопросы и согласования всех точек зрения на проблему создания защищённых систем разработаны и продолжают разрабатываться стандарты информационной безопасности. Это документы, регламентируют основные понятия и концепции информационной безопасности на государственном или межгосударственном уровне. Достигнут существенный прогресс, закреплённый в новом поколении документов.

Наиболее значимыми стандартами информационной безопасности являются (в хронологическом порядке):

- "Критерии безопасности компьютерных систем Министерства обороны США" [130];
- Руководящие документы Гостехкомиссии России [131÷135] (только

для нашей страны);

- "Европейские критерии безопасности информационных технологий" [136];
- "Федеральные критерии безопасности информационных технологий США" [137];
- "Канадские критерии безопасности компьютерных систем" [138];
- "Единые критерии безопасности информационных технологий" [139].

7.1.3. Анализ существующих стандартов информационной безопасности

Главная задача стандартов информационной безопасности – согласовать позиции и цели производителей, потребителей и аналитиков – квалификаторов в процессе создания и эксплуатации продуктов информационных технологий. Каждая из перечисленных категорий специалистов оценивает стандарты и содержащиеся в них требования и критерии по своим собственным параметрам.

Проведём сравнительный анализ существующих стандартов безопасности. Несмотря на то, что практически каждый из стандартов представляет оригинальный подход к определению понятия безопасной системы обработки информации, существует ряд понятий и концепций, используемых всеми стандартами.

В качестве обобщённых показателей, характеризующих стандарты информационной безопасности и имеющих значение для всех трёх сторон, предлагается использовать универсальность, гибкость, гарантированность, реализуемость и актуальность.

Классификация рассмотренных стандартов информационной безопасности по предложенным показателям приведена в табл. 7.1.

Таблица 7.1

Формула для работы отклонений

Стандарты безопасности	**Показатели сопоставления стандартов информационной безопасности**				
	Универсальность	Гибкость	Гарантированность	Реализуемость	Актуальность
Оранжевая книга (1983 г.)	ограниченная	ограниченная	ограниченная	высокая (за исключением класса А)	умеренная
Европейские критерии (1986 г.)	умеренная	умеренная	умеренная	высокая	умеренная
Документы ГТК (1992 г.)	ограниченная	ограниченная	отсутствует	высокая	ограниченная
Федеральные критерии (1992 г.)	высокая	отличная	достаточная	высокая	высокая
Канадские критерии (1993 г.)	умеренная	достаточная	достаточная	достаточная	достаточная
Единые критерии (1999 г.)	превосходная	превосходная	превосходная	превосходная	превосходная

Степень соответствия стандартов предложенным показателям определяется по следующей качественной шкале:

- ограниченное соответствие – недостаточное соответствие, при применении стандарта возникают существенные трудности;
- умеренное соответствие – минимальное соответствие, при примене-

нии стандарта в большинстве случаев существенных трудностей не возникает;

- достаточное соответствие – удовлетворительное соответствие, при применении стандарта в большинстве случаев не возникает никаких трудностей, однако эффективность предлагаемых решений не гарантируется;
- высокое соответствие – стандарт предлагает специальные механизмы и процедуры, направленные на улучшение данного показателя, применение которых позволяет получать достаточно эффективные решения;
- превосходное соответствие – улучшение данного показателя рассматривалось авторами стандарта в качестве одной из основных целей его разработки, что обеспечивает эффективность применения предложенных решений.

Представленный анализ стандартов информационной безопасности и основных тенденций их развития позволяет сделать следующие выводы.

1) Развитие стандартов привело к отказу от единой шкалы ранжирования требований и критериев, замене их множеством независимых частных показателей и введению частично упорядоченных шкал.

2) Неуклонное возрастание роли требований адекватности реализации защиты и политики безопасности свидетельствует о тенденции преобладания "качества" обеспечения защиты над её "количеством".

3) Определение ролей производителей, потребителей и экспертов по квалификации ИТ-продуктов и разделение их функций в процессе создания защищённых систем обработки информации свидетельствует о полноправной интеграции стандартов обеспечения безопасности в сферу информационных технологий.

4) Сложившееся на основе современных стандартов, разделение ролей участников процесса создания и эксплуатации защищённых систем,

применение соответствующих механизмов и технологий привело к сбалансированному распределению ответственности между всеми участниками процесса.

5) Современные тенденции интеграции информационных технологий и стремление к созданию безопасного всемирного информационного пространства привели к необходимости интернационализации стандартов информационной безопасности.

7.2. Моделирование «защищённых» систем

7.2.1. Формальные модели безопасности

Под *политикой безопасности* понимается совокупность норм и правил, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от определённого множества угроз и составляет необходимое (а иногда и достаточное) условие безопасности системы. Формальное выражение политики безопасности называют *формальной моделью* политики безопасности. Формальные модели необходимы и используются достаточно широко, потому что только с их помощью можно доказать безопасность системы, опираясь при этом на объективные и неопровержимые постулаты математической теории, а также позволяют обосновать жизнеспособность системы и определяют базовые принципы её архитектуры и используемые при её построении технологические решения.

Основная цель создания политики безопасности информационной системы и описания её в виде формальной модели – это определение условий, которым должно подчиняться поведение системы, выработка критерия безопасности и проведение формального доказательства соответствия системы этому критерию при соблюдении установленных правил и ограничений.

Кроме того, формальные модели безопасности позволяют решить ещё целый ряд задач, возникающих в ходе проектирования, разработки и сертификации защищённых систем, поэтому их используют не только теоретики информационной безопасности, но и другие категории специалистов, участвующих в процессе создания и эксплуатации защищенных информационных систем (производители, потребители, эксперты-квалификаторы).

Среди моделей различных политик безопасности можно выделить два основных класса: дискреционные (произвольные) и мандатные (нормативные). Рассмотрим наиболее распространённые политики произвольного управления доступом, в основе которых лежат модель Харрисона-Руззо-Ульмана, модель типизованной матрицы доступа, фундаментальная нормативная модель безопасности Белла-ЛаПадулы.

7.2.2. Дискреционная модель Харрисона-Руззо-Ульмана

Модель безопасности Харрисона-Руззо-Ульмана [140], являющаяся классической дискреционной моделью, реализует произвольное управление доступом субъектов к объектам и контроль за распространением прав доступа.

В рамках этой модели система обработки информации представляется в виде совокупности активных сущностей – субъектов (множество S), которые осуществляют доступ к информации, пассивных сущностей – объектов (множество O), содержащих защищаемую информацию, и конечного множества прав доступа $R = \{r_1, ..., r_n\}$, означающих полномочия на выполнение соответствующих действий (например, *чтение*, *запись*, *выполнение*).

Причём для включения в область действия модели и отношения между субъектами, принято считать, что все субъекты одновременно являются и объектами – $S \subset O$. Поведение системы моделируется с помощью по-

нятия состояния. Пространство состояний системы образуется декартовым произведением множеств составляющих её объектов, субъектов и прав – $O \times S \times R$. Текущее состояние системы Q в этом пространстве определяется тройкой, состоящей из множества субъектов, множества объектов и матрицы прав доступа M, описывающей текущие права доступа субъектов к объектам, – $Q = (S, O, M)$. Строки матрицы соответствуют субъектам, а столбцы – объектам, поскольку множество объектов включает в себя множество субъектов, матрица имеет вид прямоугольника. Любая ячейка матрицы $M[s, o]$ содержит набор прав субъекта S к объекту o, принадлежащих множеству прав доступа R. Поведение системы во времени моделируется переходами между различными состояниями. Переход осуществляется путём внесения изменений в матрицу ***M*** с помощью команд следующего вида:

command $\alpha\ (x_1,.., x_k)$

if r_1 **in** $M[x_{s1}, x_{o1}]$ **and** (условия выполнения команды)

r_2 **in** $M\,[x_{s2}, x_{o2}]$ **and**

.

.

r_m **in** $M\,[x_{sm}, x_{om}]$ **and**

then

op_1, op_2… op_n (операции, составляющие команду).

Здесь α – имя команды;

x_i – параметры команды, являющиеся идентификаторами субъектов и объектов,

S_i, O_i – индексы субъектов и объектов в диапазоне от 1 до k;

op_i – элементарные операции.

Элементарные операции, составляющие команду, выполняются только в том случае, если все условия, означающие присутствие указанных прав доступа в ячейках матрицы M, являются истинными. В классической модели допустимы только следующие элементарные операции:

enter r **into** $M[s, o]$ (добавление субъекта s права r для объекта o)

delete r **from** $M[s, o]$ (удаление субъекта s права r для объекта o)

create subject s (создание нового субъекта s)

create object o (создание нового объекта o)

destroy subject s (удаление существующего субъекта s)

destroy object o (удаление существующего объекта o).

Применение любой элементарной операции op в системе, находящейся в состоянии $Q = (S, O, M)$, влёчет за собой переход в другое состояние $Q' = (S', O', M')$, которое отличается от предыдущего состояния Q по крайней мере одним компонентом. Выполнение базовых операций приводит к следующим изменениям в состоянии системы:

enter r **into** $M[s, o]$ (где $s \in S, o \in O$)

$O` = O$

$S` = S$

$M`[x_s, x_o] = M[x_s, x_o]$, если $(x_s, x_o) \neq (s, o)$

$M`[s, o] = M[s, o] \cup \{r\}$

delete r **from** $M[s, o]$ (где $s \in S, o \in O$)

$O` = O$

$S` = S$

$M`[x_s, x_o] = M[x_s, x_o]$, если $(x_s, x_o) \neq (s, o)$

$M`[s, o] = M[s, o] \setminus \{r\}$

create subject s (где $s \notin S$)

$O` = O \cup \{s\}$

$S` = S \cup \{s\}$

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S \times O$

$M`[s, x_o] = \theta$ для всех $x_o \in O`$

$M`[s, x_s] = \theta$ для всех $x_s \in S`$

destroy subject s (где $s \in S$)

$O` = O \setminus \{s\}$

$S` = S \setminus \{s\}$

$M[x_s, x_o]` = M[x_s, x_o]$ для всех $(x_s, x_o) \in S` \times O`$

create object o (где $o \in O$)

$O` = O \cup \{o\}$

$S` = S$

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S \times O$

$M`[x_s, o] = \theta$ для всех $x_s \in S`$

destroy object o (где $o \in O \setminus S$)

$O` = O \setminus \{o\}$

$S` = S$

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S` \times O`$.

Операция **enter** вводит право r в существующую ячейку матрицы доступа. Содержимое каждой ячейки рассматривается как множество, то есть если это право уже имеется, то ячейка не изменяется. Операция **enter** называется *монотонной*, поскольку она только добавляет права в матрицу доступа и ничего не удаляет. Действие операции **delete** противоположно действию операции **enter**. Она удаляет право из ячейки матрицы доступа, если оно там присутствует. Поскольку содержимое каждой ячейки рассматривается как множество, **delete** не делает ничего, если удаляемое право отсутствует в указанной ячейке. Поскольку **delete** удаляет информацию из матрицы доступа, она называется *немонотонной* операцией. Операции **create subject** и **destroy subject** представляют собой аналогичную пару монотонной и немонотонной операции.

Заметим, что для каждой операции существует ещё и предусловие её выполнения: для того чтобы изменить ячейку матрицы доступа с помощью операций **enter** или **delete** необходимо, чтобы эта ячейка существовала, то есть чтобы существовали соответствующие субъект и объект. Предусло-

виями операций создания **create subject / object** является отсутствие создаваемого субъекта/объекта, операций удаления **destroy subject / object** – наличие субъекта/объекта. Если предусловие любой операции не выполнено, то её выполнение безрезультатно.

Формальное описание системы Z,(G, R, C) состоит из следующих элементов:

1. конечный набор прав доступа $R = \{r_1, ..., r_n\}$;
2. конечные наборы исходных субъектов $S_o = \{s_1, ..., s\}$ и объектов $O_0 = \{o_1, ..., o_m\}$, где $S_0 \subseteq O_0$;
3. исходная матрица доступа, содержащая права доступа субъектов к объектам – M_0;
4. конечный набор команд $C = \{\alpha_i (x_1,..., x_k)\}$, каждая из которых состоит из условий выполнения и интерпретации в терминах перечисленных элементарных операций.

Поведение системы во времени моделируется с помощью последовательности состояний $\{Q_i\}$, в которой каждое последующее состояние является результатом применения некоторой команды из множества C к предыдущему $Q_{n+1} = C_n (Q_n)$. Таким образом, для заданного начального состояния только от условий команд из C и составляющих их операций зависит, сможет ли система попасть в то или иное состояние, или нет. Каждое состояние определяет отношения доступа, которые существуют между сущностями системы в виде множества субъектов, объектов и матрицы прав. Поскольку для обеспечения безопасности необходимо наложить запрет на некоторые отношения доступа, для заданного начального состояния системы должна существовать возможность определить множество состояний, в которые она сможет из него попасть. Это позволит задавать такие начальные условия (интерпретацию команд C, множества объектов O_0, субъектов S_0 и матрицу доступа M_0), при которых система никогда не сможет попасть в состояния, нежелательные с точки зрения безопасности.

Следовательно, для построения системы с предсказуемым поведением необходимо для заданных начальных условий получить ответ на вопрос: сможет ли некоторый субъект S когда-либо приобрести право доступа r для некоторого объекта O?

Поэтому критерий безопасности модели Харрисона-Руззо-Ульмана формулируется следующим образом.

Для заданной системы начальное состояние $Q_0 = (S_0, O_0, M_0)$ *является безопасным относительно права* **r**, *если не существует применимой к* Q_0 *последовательности команд, в результате которой право* **r** *будет занесено в ячейку матрицы* ***M***, *в которой оно отсутствовало в состоянии* Q_0 .

Смысл данного критерия состоит в том, что для безопасной конфигурации системы субъект никогда не получит право r доступа к объекту, если он не имел его изначально. На первый взгляд такая формулировка кажется довольно странной, поскольку невозможность получения права r вроде бы влечёт за собой отказ от использования команд, в которых присутствует операция **enter into** $M[s, o]$, однако это не так. Дело в том, что удаление субъекта или объекта приводит к уничтожению всех прав в соответствующей строке или в столбце матрицы, но не влёчет за собой уничтожение самого столбца или строки и сокращение размеров матрицы. Следовательно, если в какой-то ячейке в начальном состоянии существовало право r, и после удаления субъекта или объекта, к которым относилось это право, ячейка будет очищена, но впоследствии в результате создания субъекта или объекта появится вновь, и в эту ячейку с помощью соответствующей команды **enter** снова будет занесено право r, то это не будет означать нарушения безопасности.

Из критерия безопасности следует, что для данной модели ключевую роль играет выбор значений прав доступа и их использование в условиях команд. Хотя модель не налагает никаких ограничений на смысл прав и считает их равнозначными, те из них, которые участвуют в условиях вы-

полнения команд фактически представляют собой не права доступа к объектам (как, например, чтение и запись), а права управления доступом, или права на осуществление модификации ячеек матрицы доступа. Таким образом, по сути дела данная модель описывает не только доступ субъектов к объектам, а распространение прав доступа от субъекта к субъекту, поскольку именно изменение содержания ячеек матрицы доступа определяет возможность выполнения команд, в том числе команд, модифицирующих саму матрицу доступа, которые потенциально могут привести к нарушению критерия безопасности.

Необходимо отметить, что с точки зрения практики построения защищенных систем модель Харрисона-Руззо-Ульмана является наиболее простой в реализации и эффективной в управлении, поскольку не требует никаких сложных алгоритмов и позволяет управлять полномочиями пользователей с точностью до операции над объектом, чем и объясняется её распространенность среди современных систем. Кроме того, предложенный в данной модели критерий безопасности является весьма сильным в практическом плане, поскольку позволяет гарантированность недоступности определённого объекта для субъекта, которому изначально не выданы соответствующие полномочия.

Однако Харрисон, Руззо и Ульман доказали, что в общем случае не существует алгоритма, который может для произвольной системы, её начального состояния $Q_0 = (S_0, O_0, M_0)$ и общего права r решить, является ли данная конфигурация безопасной. Доказательство опирается на свойства машины Тьюринга, с помощью которой моделируется последовательность переходов системы из состояния в состояние.

Для того чтобы можно было доказать указанный критерий, модель должна быть дополнена рядом ограничений [142]. Не останавливаясь на математическимх выкладках, следует отметить, что указанная задача является разрешимой в любом из следующих случаев:

- команды α_i (x_1,. x_k) являются монооперационными, то есть состоят не более чем из одной элементарной операции;
- команды α_i (x_1,., x_k) являются одноусловными и монотонными, то есть содержат не более одного условия и не содержат операций **destroy** и **delete**;
- команды α_i {x_1,..., x_k) не содержат операций **create**.

Эти условия существенно ограничивают сферу применения модели, поскольку трудно представить себе реальную систему, в которой не будет происходить создания или удаления сущностей.

Таким образом, дискреционная модель Харрисона-Руззо-Ульмана в своей общей постановке не дает гарантий безопасности системы, однако именно она послужила основой для целого класса моделей политик безопасности, которые используются для управления доступом и контроля за распространением прав во всех современных системах.

7.2.3. Типизованная матрица доступа

Другая дискреционная модель, получившая название "Типизованная матрица доступа" (Type Access Matrix – далее TAM) [141], представляет собой развитие модели Харрисона-Руззо-Ульмана, дополненной концепцией типов, что позволяет несколько смягчить те условия, для которых возможно доказательство безопасности системы.

Формальное описание модели TAM включает следующие элементы:

1. конечный набор прав доступа $R = \{r_1, ..., r_i\}$;
2. конечный набор типов $T = \{ t_1, ..., r_g\}$;
3. конечные наборы исходных субъектов $S_0 = \{s_1, ..., s\}$ и объектов $O_0 = \{o_1, ..., o_m\}$, где $S_0 \subseteq O_0$;
4. матрица M, содержащая права доступа субъектов к объектам, и её на-

чальное состояние M_0;

5. конечный набор команд $C = \{ \alpha_i (x_1,. x_k)\}$, включающий условия выполнения команд и их интерпретацию в терминах элементарных операций.

Тогда состояние системы описывается четвёркой

$$Q = (S, O, t, M) ,$$

где S, O, и M обозначают соответственно множество субъектов, объектов и матрицу доступа, а $t: O \rightarrow T$ – функция, ставящая в соответствие каждому объекту некоторый тип.

Состояние системы изменяется с помощью команд из множества C. Команды TAM имеют тот же формат, что и в модели Харрисона-Руззо-Ульмана, но всем параметрам приписывается определенный тип:

command $\alpha (x_1: t_1,.., x_k: t_k)$

if r_1 **in** $M[x_{s1}, x_{o1}]$ **and** (условия выполнения команды)

r_2 **in** $M [x_{s2}, x_{o2}]$ **and**

.

.

r_m **in** $M [x_{sm}, x_{om}]$ **and**

then

op_1, $op_2 \ldots$ op_n (операции, составляющие команду).

Перед выполнением команды происходит проверка типов фактических параметров, и если они не совпадают с указанными в определении, команда не выполняется. Фактически, введение контроля типов для параметров команд приводит к неявному введению дополнительных условий, так как команды могут быть выполнены только при совпадении типов параметров. В модели используются следующие шесть элементарных операций, отличающихся от аналогичных операций модели Харрисона-Руззо-Ульмана только использованием типизованных параметров.

Монотонные операции	Немонотонные операции
enter r **into** $M[s, o]$	**delete** r **from** $M[s, o]$
create subject s **of type** t	**destroy subject** s
create object o **of type** t	**destroy object** o

Смысл элементарных операций совпадает со смыслом аналогичных операций из классической модели Харрисона-Руззо-Ульмана с точностью до использования типов:

enter r **into** $M[s, o]$ (где $s \in S,\ o \in O$)

$O` = O$

$S` = S$

$t`(o) = t(o)$ для всех $o \in O$

$M`[x_s, x_o] = M[x_s, x_o]$, если $(x_s, x_o) \neq (s, o)$

$M`[s, o] = M[s, o] \cup \{r\}$

delete r **from** $M[s, o]$ (где $s \in S, o \in O$)

$O` = O$

$S` = S$

$t`(o) = t(o)$ для всех $o \in O$

$M`[x_s, x_o] = M[x_s, x_o]$, если $(x_s, x_o) \neq (s, o)$

$M`[s, o] = M[s, o] \setminus \{r\}$

create subject s **of type** t_s (где $s \in S$)

$O` = O \cup \{s\}$

$S` = S \cup \{s\}$

$t`(o) = t(o)$ для всех $o \in O$

$t`(s) = t_s$

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S \times O$

$M`[s, x_o] = \theta$ для всех $x_o \in O`$

$M`[s, x_s] = \theta$ для всех $x_s \in S`$

destroy subject s (где $s \in S$)

$O` = O \backslash \{s\}$

$S` = S \backslash \{s\}$

$t`(o) = t(o)$ для всех $O \in O`$

$t`(s)$ = не определено

$M[x_s, x_o]` = M[x_s, x_o]$ для всех $(x_s, x_o) \in S` \times O`$

create object o **of type** t_o (где $o \in O$)

$O` = O \cup \{o\}$

$S` = S$

$t`(O) = t(O)$ для всех $O \in O$

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S \times O$

$M`[x_s, o] = \theta$ для всех $x_s \in S`$

destroy object o (где $o \in O \backslash S$)

$O` = O \backslash \{o\}$

$S` = S$

$t`(x_o) = t(x_o)$ для всех $x_o \in O`$

$t`(o)$ = не определено

$M`[x_s, x_o] = M[x_s, x_o]$ для всех $(x_s, x_o) \in S` \times O`$.

Таким образом, ТАМ является обобщением модели Харрисона-Руззо-Ульмана,. которую можно рассматривать как частный случай ТАМ с одним-единственным типом, к которому относятся все объекты и субъекты. Появление в каждой команде дополнительных неявных условий, ограничивающих область применения команды только сущностями соответствующих типов, позволяет несколько смягчить жесткие условия классической модели, при которых критерий безопасности является разрешимым.

В работе [142] Харрисон, Руззо и Ульман показали, что критерий безопасности дискреционной модели может быть доказан для систем, в которых все команды $\alpha_i(x_1,..., x_k)$ являются одноусловными и монотонными. Строгий контроль соответствия типов позволяет смягчить требование одноусловности, заменив его ограничением на типы параметров команд, при выполнении которых происходит создание новых сущностей.

Для того чтобы сформулировать это ограничение, определим отношения между типами. Пусть $\alpha(x_1{:}t_1, x_2{:}c, \ldots, x_k{:}t_k)$ – некоторая команда ТАМ. Будем говорить, что t_i является *дочерним типом* в α, если в теле α имеет место одна из следующих элементарных операций: **create subject** x_i **of type** t_i или **create object** x_i **of type** t_i. В противном случае будем говорить, что t_i является *родительским типом* в α.

Отметим, что в одной команде тип может быть одновременно и родительским, и дочерним, например:

command **foo** ($s_1{:}u$, $s_2{:}u$, $s_3{:}w$, $o{:}b$)
create subject s_2 **of type** u;
create subject s_3 **of type** v;
end.

Здесь u является родительским типом относительно s_1 и дочерним типом относительно s_2. Кроме того, w и b являются родительскими типами, а v – дочерним типом.

Тогда можно описать взаимосвязи между различными типами с помощью графа, определяющего отношение "наследственности" между типами, устанавливаемые через операции порождения сущностей (объектов и субъектов). Такой граф называется *графом создания* и представляет собой направленный граф с множеством вершин T, в котором ребро от u к v существует тогда и только тогда, когда в системе имеется создающая команда, в которой u является родительским типом, а v – дочерним типом.

Этот граф для каждого типа позволяет определить:

1. сущности каких типов должны существовать в системе, чтобы в ней мог появиться объект или субъект заданного типа;
2. сущности каких типов могут быть порождены при участии сущностей заданного типа.

Модель монотонной типизированной матрицы доступа (MTAM) идентична TAM за исключением того, что в ней отсутствуют немонотонные элементарные операции **delete**, **destroy subject** и **destroy object**.

Реализация MTAM, состоящая из множеств объектов, субъектов, типов, матрицы прав доступа и множества команд, называется *ациклической* тогда и только тогда, когда её граф создания не содержит циклов, в противном случае говорят, что реализация *циклическая*. Например, граф создания для приведённой выше команды **foo** содержит следующие ребра: $\{(u, u), (u, v), (w, u), (w, v), (b, u), (b, v)\}$. Реализация MTAM, содержащая эту команду, будет циклической, поскольку тип u является для неё одновременно и родительским и дочерним, что приводит к появлению на графе цикла (u, u).

Доказано, что критерий безопасности, предложенный Харрисоном, Руззо и Ульманом, разрешим для ациклических реализаций MTAM, и что требование одноусловности команд можно заменить требованием ациклличности графа создания [141]. Смысл этой замены состоит в том, что последовательность состояний системы должна следовать некоторому маршруту на графе создания, поскольку невозможно появление сущностей дочерних типов, если в системе отсутствуют сущности родительских типов, которые должны участвовать в их создании. Отсутствие циклов на графе создания позволяет избежать зацикливания при доказательстве критерия безопасности, так как количество путей на графе без циклов является ограниченным.

Это означает, что поведение системы становится предсказуемым, поскольку в любом состоянии можно определить сущности каких типов

могут появиться в системе, а каких – нет. Но, к сожалению, доказано, что в общем случае сложность проверки критерия безопасности для МТАМ является NP-трудной задачей, то есть с ростом размерности задачи (количества объектов и субъектов) время на её решение растёт в степенной зависимости от её размерности. Этот недостаток может быть преодолен с помощью *тернарной* ТАМ, в которой команды могут иметь не более трёх параметров. Тернарная МТАМ является монотонной версией тернарной ТАМ. Для тернарной МТАМ доказательство безопасности радикально упрощается, поскольку при проверке условной части команды всегда используется только небольшой фрагмент матрицы доступа. Тернарная МТАМ по своим выразительным способностям эквивалентна МТАМ, несмотря на это, доказано, что безопасность её ациклической реализации разрешима за время, зависящее от размера начальной матрицы.

Следовательно, введение строгого контроля типов в дискреционную модель Харрисона-Руззо-Ульмана позволило доказать критерий безопасности систем для более приемлемых ограничений, что существенно расширило область её применения.

7.2.4. Мандатная модель Белла-ЛаПадулы

Система в модели безопасности Белла-ЛаПадулы [143], как и в модели Харрисона-Руззо-Ульмана, представляется в виде множеств субъектов S, объектов O (множество объектов включает множество субъектов, $S \subset O$) и прав доступа **read** (чтение) и **write** (запись). В мандатной модели рассматриваются только эти два вида доступа, и хотя она может быть расширена введением дополнительных прав (например, правом на добавление информации, выполнение программ и так далее), все они будут отображаться в базовые (чтение и запись). Использование столь жёсткого подхода, не позволяющего осуществлять гибкое управление доступом, объясня-

ется тем, что в мандатной модели контролируются не операции, осуществляемые субъектом над объектом, а потоки информации, которые могут быть только двух видов: либо от субъекта к объекту (запись), либо от объекта к субъекту (чтение).

Уровни безопасности субъектов и объектов задаются с помощью функции уровня безопасности $F{:}S \cup O \rightarrow L$, которая ставит в соответствие каждому объекту и субъекту уровень безопасности, принадлежащий множеству уровней безопасности L, на котором определена рёшетка Λ [144].

7.2.4.1. Решётка уровней безопасности

Решётка уровней безопасности Λ – это формальная алгебра $(L, \leq, \bullet, \otimes)$, где L – базовое множество уровней безопасности, а оператор «≤» определяет частичное нестрогое отношение порядка для элементов этого множества, то есть оператор «≤» – антисимметричен, транзитивен и рефлексивен.

Отношение «≤» на L:

1) рефлексивно, если

   $\forall\, a \in L{:}\, a \leq a$;

2) антисимметрично, если

   $\forall\, a_1, a_2, \in L{:}\, (a_1 \leq a_2 \wedge a_2 \leq a_1) \Rightarrow a_1 = a_2$;

3) транзитивно, если

   $\forall\, a_1, a_2, a_3 \in L{:}\, (a_1 \leq a_2 \wedge a_2 \leq a_3\ ) \Rightarrow a_1 = a_3$.

Другое свойство решётки состоит в том, что для каждой пары a_1 и a_2 элементов множества L можно указать единственный элемент *наименьшей верхней границы* и единственный элемент *наибольшей нижней границы*. Эти элементы также принадлежат L и обозначаются с помощью операторов $\bullet$ и $\otimes$ соответственно:

$$a_1 \bullet a_2 = a \Leftrightarrow a_1, a_2 \le a \wedge \forall\, a` \in L: (a` \le a) \Rightarrow (a` \le a_1 \vee a` \le a_2);$$

$$a_1 \otimes a_2 = a \Leftrightarrow a \le a_1, a_2 \wedge \forall\, a` \in L: (a` \le a_1 \wedge a` \le a_2) \Rightarrow (a` \le a).$$

Смысл этих определений заключается в том, что для каждой пары элементов (или множества элементов, поскольку операторы ● и ⊗ транзитивны) всегда можно указать единственный элемент, ограничивающий её сверху или снизу таким образом, что между ними и этим элементом не будет других элементов.

Функция уровня безопасности F назначает каждому субъекту и объекту некоторый уровень безопасности из L, разбивая множество сущностей системы на классы, в пределах которых их свойства с точки зрения модели безопасности являются эквивалентными. Тогда оператор «<» определяет направление потоков информации, то есть, если $F(A) \le F(B)$, то информация может передаваться от элементов класса A элементам класса B.

Покажем, почему в модели Белла-ЛаПадулы для описания отношения доминирования на множестве уровней безопасности используется решётка.

Если информация может передаваться от сущностей класса A к сущностям класса B, а также от сущностей класса B к сущностям класса A, то классы A и B содержат одноуровневую информацию и с точки зрения безопасности эквивалентны одному классу (AB). Поэтому для удаления избыточных классов необходимо, чтобы отношение «≤» было антисимметричным.

Если информация может передаваться от сущностей класса A сущностям класса B, а также от сущностей класса B к сущностям класса C, то очевидно, что она будет также передаваться от сущностей класса A к сущностям класса C. Таким образом, отношение «≤» должно быть транзитивным.

Так как класс сущности определяет уровень безопасности содержащейся в ней информации, то все сущности одного и того же класса содержат с точки зрения безопасности одинаковую информацию. Следователь-

но, нет смысла запрещать потоки информации между сущностями одного и того же класса. Более того, из чисто практических соображений нужно предусмотреть возможность для сущности передавать информацию самой себе. Следовательно, отношение «≤» должно быть рефлексивным.

Покажем, что для любого множества сущностей должны существовать единственная наименьшая верхняя и наибольшая нижняя границы множества соответствующих им уровней безопасности. Для пары сущностей x и y, обладающих уровнями безопасности a и b соответственно, обозначим наибольший уровень безопасности их комбинации как $(a \cdot b)$, при этом $a \leq (a \cdot b)$ и $b \leq (a \cdot b)$. Тогда, если существует некоторый уровень с такой, что $a \leq c$ и $b \leq c$, то должно иметь место отношение $(a \cdot b) \leq c$, поскольку $(a \cdot b)$ – это минимальный уровень субъекта, для которого доступна информация как из x, так и из y. Следовательно, $(a \cdot b)$ должен быть наименьшей верхней границей a и b. Аналогично обозначим наименьший уровень безопасности комбинации сущностей x и y как $(a \otimes b)$, при этом $(a \otimes b) \leq$ а и $(a \otimes b) \leq b$. Тогда, если существует некоторый уровень c такой, что $c \leq a$ и $c \leq b$, то должно иметь место отношение $cv(a \otimes b)$, поскольку $(a \otimes b)$ – это максимальный уровень субъекта, для которого разрешена передача информации как в x, так и в y. Следовательно, $(a \otimes b)$ должен быть наибольшей нижней границей a и b.

Использование решётки для описания отношений между уровнями безопасности позволяет использовать в качестве атрибутов безопасности (элементов множества L) не только целые числа, для которых определено отношение "меньше или равно", но и более сложные составные элементы. Например, в государственных организациях достаточно часто в качестве атрибутов безопасности используется комбинация, состоящая из уровня безопасности, представляющего собой целое число, и набора категорий из некоторого множества. Такие атрибуты невозможно сравнивать с помо-

щью арифметических операций, поэтому отношение доминирования «≤» определяется как композиция отношения "меньше или равно" для уровней безопасности и отношения включения множеств $\subseteq$ для наборов категорий. Причём, это никак не сказывается на свойствах модели, поскольку отношения "меньше или равно" и "включение множеств" обладают свойствами антисимметричности, транзитивности и рефлексивности, и, следовательно, их композиция также будет обладать этими свойствами, образуя над множеством атрибутов безопасности решётку. Точно так же можно использовать любые виды атрибутов и любое отношение частичного порядка, лишь бы их совокупность представляла собой рёшетку.

7.2.4.2. Основная теорема безопасности Белла-ЛаПадулы

Система $\Sigma(v0, R, T)$ безопасна тогда и только тогда, когда:

Начальное состояние $v0$ безопасно и для любого состояния v, достижимого из $v0$ путём применения конечной последовательности запросов из R таких, что $T\{v, r) = v^*$, $v = (F, M)$ и $v^* = (F^*, M^*)$ для каждого $s \in S$ и $o \in O$ выполняются следующие условия:

если **read** $\in M^*[s, o]$ и **read** $\notin M[s, o]$, то $F^*(s) \geq F^*(o)$;

если **read** $\in M[s, o]$ и $F^*(s) < F^*(o)$, то **read** $\notin M^*[s, o]$;

если **write** $\in M^*[s, o]$ и **write** $\notin M[s, o]$, то $F^*(o) \geq F^*(s)$;

если **write** $\in M[s, o]$ и $F^*(o) < F^*(s)$, то **write** $\notin M^*[s, o]$.

Доказательство:

Необходимость. Если система безопасна, то состояние $v0$ безопасно по определению. Допустим, существует некоторое состояние v^*, достижимое из $v0$ путём применения конечного числа запросов из R и полученное путём перехода из безопасного состояния V: $T(v, r) = v^*$. Тогда, если при

этом переходе нарушено хотя бы одно из первых двух ограничений, накладываемых теоремой на функцию T, то состояние v^* не будет безопасным по чтению, а если функция T нарушает хотя бы одно из последних двух условий теоремы, то состояние v^* не будет безопасным по записи. В любом случае при нарушении условий теоремы система небезопасна.

Достаточность. Проведём доказательство от противного. Предположим, что система небезопасна. В этом случае либо $v0$ небезопасно, что явно противоречит условиям теоремы, либо должно существовать небезопасное состояние v^*, достижимое из безопасного $v0$ путём применения конечного числа запросов из R. В этом случае обязательно будет иметь место переход $T(v, r) = v^*$, при котором состояние v – безопасно, а состояние v^* – нет, однако четыре условия теоремы делают такой переход невозможным.

Таким образом, теорема утверждает, что система с безопасным начальным состоянием является безопасной тогда и только тогда, когда при любом переходе системы из одного состояния в другое не возникает никаких новых и не сохраняется никаких старых отношений доступа, которые будут небезопасны по отношению к функции уровня безопасности нового состояния. Формально эта теорема определяет все необходимые и достаточные условия, которые должны быть выполнены для того, чтобы система, начав свою работу в безопасном состоянии, никогда не достигла небезопасного состояния.

7.2.4.3. Безопасная функция перехода

Недостаток основной теоремы безопасности Белла-ЛаПадулы состоит в том, что ограничения, накладываемые теоремой на функцию перехода, совпадают с критериями безопасности состояния, поэтому данная теорема является избыточной по отношению к определению безопасного состояния. Кроме того, из теоремы следует только то, что все состояния,

достижимые из безопасного состояния при определённых ограничениях, будут в некотором смысле безопасны, но при этом не гарантируется, что они будут достигнуты без потери свойства безопасности в процессе осуществления перехода. Поскольку не имеется никаких определённых ограничений на вид функции перехода, кроме указанных в условиях теоремы, и допускается, что уровни безопасности субъектов и объектов могут изменяться, то можно представить такую гипотетическую систему (она получила название Z-системы [145]), в которой при попытке низкоуровневого субъекта прочитать информацию из высокоуровневого объекта будет происходить понижение уровня объекта до уровня субъекта и осуществляться чтение.

Функция перехода Z-системы удовлетворяет ограничениям основной теоремы безопасности, и все состояния такой системы также являются безопасными в смысле критерия Белла-ЛаПадулы, но вместе с тем в этой системе любой пользователь сможет прочитать любой файл, что, очевидно, несовместимо с безопасностью в обычном понимании.

Следовательно, необходимо сформулировать теорему, которая бы не только констатировала безопасность всех достижимых состояний для системы, соответствующей определённым условиям, но и гарантировала бы безопасность в процессе осуществления переходов между состояниями. Для этого необходимо регламентировать изменения уровней безопасности при переходе от состояния к состоянию с помощью дополнительных правил.

Такую интерпретацию мандатной модели осуществил Мак-Лин [146], предложивший свою формулировку основной теоремы безопасности, основанную не на понятии безопасного состояния, а на понятии безопасного перехода. При таком подходе функция уровня безопасности представляется с помощью двух функций, определённых на множестве субъектов и объектов: F_s: $S \to L$ и F_o: $O \to L$.

Функция перехода T считается безопасной по чтению, если для любого перехода $T(r, v) = v^*$ выполняются следующие три условия:

если **read** $\in M^*[s, o]$ и

read $\notin M[s, o]$ то:

$F_s^*(s) \geq F_o(o)$ и $F = F^*$;

если $F_s \neq F_s^*$ то:

$M = M^*$,

$F_o = F_o^*$,

для $\forall s$ и o, для которых $F_s^*(s) < F_o^*(o)$, **read** $\notin M[s, o]$;

если $F_o \neq F_o^*$ то:

$M = M^*$,

$F_s = F_s^*$,

для $\forall s$ и o, для которых $F_s^*(s) < F_o^*(o)$, **read** $\notin M[s, o]$.

Функция перехода T считается безопасной по записи, если для любого перехода $T(r, v) = v^*$ выполняются следующие три условия:

если **write** $\in M^*[s, o]$ и

write $\notin M[s, o]$ то:

$F_o^*(o) \geq F_s(s)$ и $F = F^*$;

если $F_s \neq F_s^*$ то:

$M = M^*$,

$F_o = F_o^*$,

для $\forall s$ и o, для которых $F_s^*(s) > F_o^*(o)$, **write** $\notin M[s, o]$;

если $F_o \neq F_o^*$ то:

$M = M^*$,

$F_s = F_s^*$,

для $\forall s$ и o, для которых $F_s^*(s) > F_o^*(o)$, **write** $\notin M[s, o]$.

Функция перехода является безопасной тогда и только тогда, когда она одновременно безопасна и по чтению и по записи.

Смысл введения перечисленных ограничений и их принципиальное отличие от условий теоремы Белла-ЛаПадулы состоит в следующем: нельзя изменять одновременно более одного компонента состояния системы – в процессе перехода либо возникает новое отношение доступа, либо изменяется уровень объекта, либо изменяется уровень субъекта.

Следовательно, *функция перехода является безопасной тогда и только тогда, когда она изменяет только один из компонентов состояния и изменения не приводят к нарушению безопасности системы*.

Поскольку безопасный переход из состояния v в состояние v^* позволяет изменяться только одному элементу из v и так как этот элемент может быть изменён только способами, сохраняющими безопасность состояния, была доказана следующая теорема о свойствах безопасной системы [145].

Теорема безопасности Мак-Лина. Система безопасна в любом состоянии и в процессе переходов между ними, если её начальное состояние является безопасным, а её функция перехода удовлетворяет критерию Мак-Лина.

Обратное утверждение неверно. Система может быть безопасной по определению Белла-ЛаПадулы, но не иметь безопасной функции перехода, о чём свидетельствует рассмотренный пример Z-системы.

Такая формулировка основной теоремы безопасности предоставляет в распоряжение разработчиков защищённых систем базовый принцип их построения, в соответствии с которым для того, чтобы обеспечить безопасность системы как в любом состоянии, так и в процессе перехода между ними, необходимо реализовать для неё такую функцию перехода, которая соответствует указанным условиям.

7.2.5. Моделирование квантовых наноразмерных структур для систем кодирования и криптографии

В качестве элемента системы кодирования и криптографии выступает изомерная квантовая точка. При описании модели физического явления удобно использовать оболочечную модель ядра. В оболочечной модели ядра принимается, что энергетическая структура (уровни энергии нуклонов) ядра подобна энергетической структуре электронной оболочки атома. Сильное взаимодействие нуклонов в ядре и малый радиус этого взаимодействия позволяет рассматривать нуклоны движущимися независимо друг от друга в поле, обладающем сферически симметричным потенциалом. При этом нуклоны могут находиться в различных энергетических состояниях. Основному состоянию ядра должно соответствовать заполнение всех нижних уровней. Потеря нуклоном энергии при межнуклонных столкновениях не может перевести его в более низкое состояние, ибо все они заняты в соответствии с принципом Паули. Это приводит к тому, что длина свободного пробега нуклона в невозбуждённом ядре становится больше радиуса ядра. Это означает возможность рассматривать нуклоны в рамках данной модели невзаимодействующими и несталкивающимися. Движение невзаимодействующих нуклонов в поле сферического потенциала, где орбитальный момент импульса является интегралом движения, характеризуется тем, что всем $2l+1$ возможным ориентациям вектора l соответствует одинаковый энергетический уровень. На этом уровне размещаются $2(2l+1)$ нуклонов данного типа. Таким образом, в оболочечной модели нуклоны располагаются в определенном количестве на энергетических *нуклонных оболочках.* Каждый нуклон характеризуется индивидуальной волновой функцией и индивидуальными квантовыми числами n и l.

Существуют две системы нуклонных состояний – одна для протонов, другая для нейтронов; обе системы уровней заполняются нуклонами независимо друг от друга. Ядра, имеющие только заполненные *нуклонные оболочки,* должны обладать повышенной устойчивостью (проявляющейся, например, в их большей распространённости в природе), а также должны иметь сферически симметричное распределение заряда.

Порядок заполнения нуклонных оболочек с ростом A сходен с порядком заполнения электронных оболочек с ростом Z. Ввиду сильной спин-орбитальной связи все уровни с $l \neq 0$ расщепляются на два подуровня с $j = l \pm 1/2$, заполняющихся независимо.

Предсказания оболочечной модели, в общем, соответствуют действительности. Наиболее устойчивым по сравнению с соседними ядрами являются ядра со значениями N или Z, равными **2, 8, 20, 28, 50, 82, 126** и **152**. Эти числа называются *магическими.* Распространённость в природе таких ядер наиболее велика, а квадрупольные моменты их близки к нулю. Ядра, у которых магическими числами являются и N и Z, называются *дважды магическими.* Эти ядра ($_2He^4$, $_8O^{16}$, $_{20}Ca^{40}$, $_{32}Pb^{208}$) отличаются особой устойчивостью, проявляющейся, в частности, в том, что они являются наиболее распространёнными в природе изотопами этих элементов.

Гамма-излучением называется жёсткое электромагнитное излучение, энергия которого высвобождается при переходах ядер из возбуждённого в основное или в менее возбуждённое состояние, а также при ядерных реакциях.

В первом случае энергия γ-квантов равна разности энергий конечного и начального уровней ядра. В каждом акте перехода ядро излучает один γ-квант. В связи с дискретностью энергетических уровней ядра гамма-излучение имеет линейчатый спектр. Частоты γ-квантов свя-

заны с разностью энергий условием частот Бора.

При испускании ядром γ-кванта само ядро, вследствие закона сохранения импульса, приобретает противоположно направленный импульс *(отдача)*. Если ядра, испускающие γ-кванты, находятся в твёрдом теле, то спектр гамма-лучей состоит из двух компонент:

а) компоненты с естественной шириной гамма-линии Γ, определяемой временем жизни ядер в данном возбужденном состоянии, с энергией E;

б) компоненты с шириной линии $\Gamma_R \sim E\frac{\bar{u}}{c} >> \Gamma$, где $\bar{u}$ – средняя квадратичная скорость теплового движения гамма-радиоактивных ядер в твёрдом теле; эта компонента имеет энергию, смещённую относительно значения E на величину энергии отдачи $R = \frac{E^2}{2M_0c^2}$, где M_0 – масса излучающего ядра (если считать его свободным и движущимся со скоростью $\overline{u << c}$).

В результате линии гамма-излучения и поглощения (той же линии) сильно размываются и, кроме того, сдвинуты по энергии друг относительно друга на величину ~ $2R$. Ввиду того, что для гамма-излучения R в общем не мало по сравнению с E, *явление резонансного поглощения гамма-лучей* ($E_{изл} = E_{погл}$ или $\nu_{изл} = \nu_{погл}$) обычно практически не наблюдается.

При определённых условиях удается добиться того, что излучаемый гамма-квант передает импульс не одному излучающему ядру, а всему кристаллу в целом. В результате излучаемой линии соответствует энергия отдачи $R \approx 0$ (M – велико) и $\Gamma_R \approx \Gamma$, то есть ширина линий приближается к естественной, а сдвиг по энергии практически исчезает. Одним из условий четкого проявления эффекта Мессбауэра является условие $R \leq 2k\Theta_D$, где Θ_D – дебаевская температура кристалла, k – постоянная Больцмана. При

$R << 2k\Theta_D$ гамма-переходы «без отдачи» можно наблюдать уже при комнатной температуре.

7.3. Варианты технических устройств для получения элементов систем кодирования и криптографии

7.3.1. Устройство формирования изомерных квантовых точек

В основу разработки положена задача создания наногетероструктур, способных сохранять возбуждённое состояние.

Устройство формирования изомерных квантовых точек 1 для систем (рис. 7.1) долговременной памяти на подложке 2, содержит зонд 3, закреплённый на пьезоприводе 4. Подложка 2 электрически связана (5) с зондом 3 и установлена в ванной 6 с соляным раствором вещества 7, ядра которого обладают ядерной изомерией.

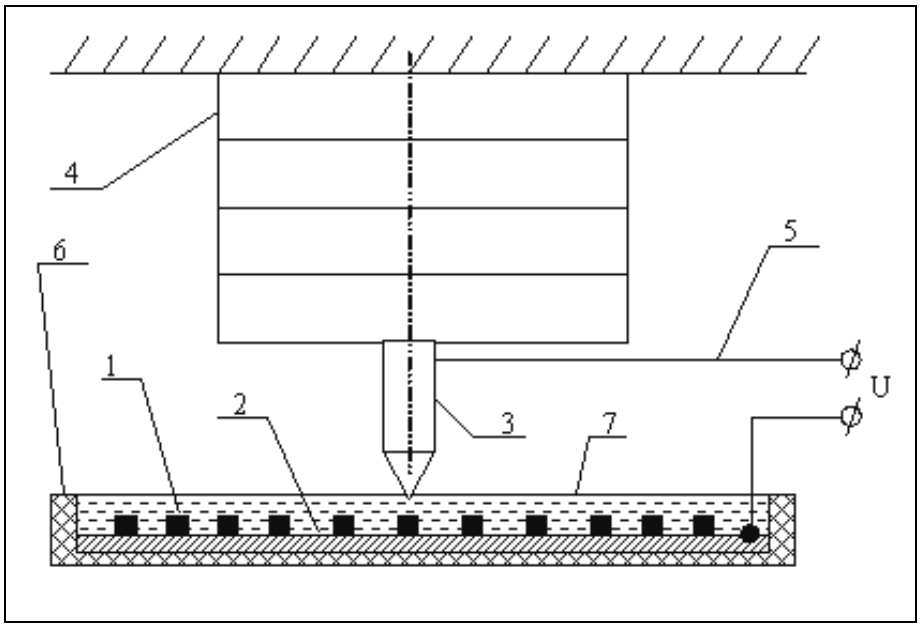


Рис. 7.1. Устройство формирования изомерных квантовых точек

При подаче рабочего напряжения в соляном растворе начинает протекать процесс электролиза, обеспечивающего формирование квантовых точек на подложке.

Применение устройства позволяет создать наногетероструктуры, способные сохранять возбуждённые состояния в течение нескольких лет.

7.3.2. Устройство долговременной памяти

В основу разработки положена задача повышения плотности записи информации на записывающей матрице.

Устройство долговременной памяти (рис. 7.2) содержит записывающую матрицу 1, излучатель электромагнитных волн 2, вещество-приёмник 3, расположенное на матрице 1. Вещество-приёмник 3 выполнено из радиоактивного материала с изомерными ядрами. Излучатель электромагнитных волн 2 выполнен в виде источника 4 гамма-лучей – фотонов большой энергии. В качестве излучателя гамма-лучей – фотонов большой энергии — использован лазерный генератор 5, с возможностью перевода изомерных ядер вещества – приемника 3 в возбуждённое состояние на период до нескольких лет. Записывающая матрица 1 связана с трёхкоординатным приводом 6, установленным на неподвижном основании 7.

Лазерный генератор 5 также закреплён на втором неподвижном основании 8, причём первое и второе основания 7,8 связаны между собой жёсткой планкой 9.

Источник 4 гамма-лучей (фотонов большой энергии) – лазерный генератор 5 воздействует на изомерные ядра вещества-приёмника 3, переводя их в возбуждённое состояние, в котором они могут находиться несколько лет. Одно изомерное возбуждённое ядро соответствует одному биту информации. Таким образом, плотность записи информации резко повышается.

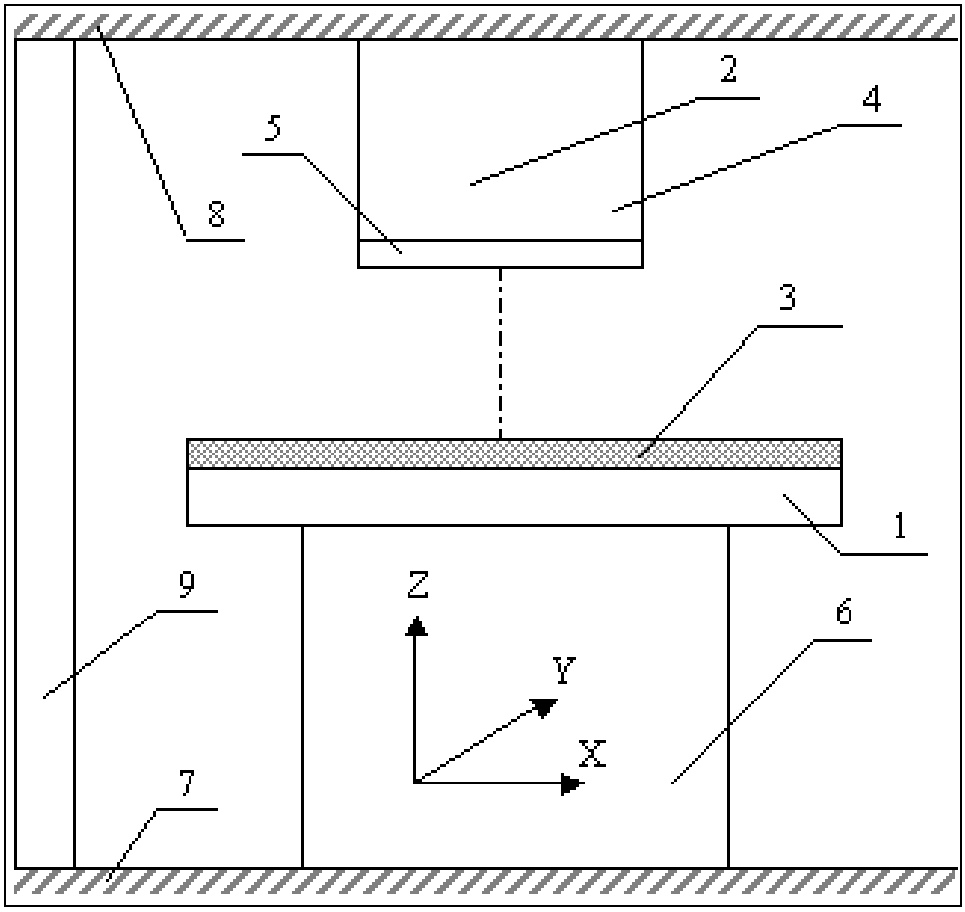


Рис. 7.2. Устройство долговременной памяти

Применение устройства долговременной памяти позволяет повысить плотность записи информации на три порядка, при котором размер одного бита информации составляет не более 1нм.

ЗАКЛЮЧЕНИЕ

В заключение необходимо отметить следующее.

Проведённый теоретический анализ и разработанная архитектура плоской двухслойной нейронной сети, выполненной на базе твёрдотельных объектов, является частью системы автоматизированного проектирования процесса производства элементов нейронных сетей. Показана экономическая и технологическая целесообразность разработки такой автоматизированной системы.

Разработанная модель обучения и оптимизации архитектур нейронных сетей является составной частью информационной системы поддержки принятия решений при проектировании процесса производства элементов нейронных сетей и направлена на решение задачи повышения производительности электронной схемы, а так же достижения высоких результатов обработки данных.

Разработанная структура позволяет проектировщикам, на основе морфологического анализа-синтеза, создавать технические решения устройств формирования элементов нейронных сетей, соответствующие критериям патентоспособной новизны, изобретательского уровня, промышленной применимости и обеспечивающие снижение затрат на производство элементов нейронных сетей – твёрдотельных объектов.

Разработанный комплекс моделей, алгоритмов функционирования и технических решений позволяет перейти к реализации гигабайтных твёрдотельных нейронных сетей на основе существующих и перспективных технологий микро- и наноэлектроники.

Предложенная схема комплексного моделирования позволяет принимать научно обоснованные, технически целесообразные, экономически и технологически выгодные решения при проектировании элементов нейронных сетей в электронном машино- и приборостроении.

Рассматриваемая математическая модель, в совокупности с эволюционной стратегией, позволяет оценивать критические ситуации для кластерных систем обработки информации и выявлять их последствия, а также моделировать и оптимизировать адаптивными методами показатели качества подобных систем.

Развитие нанотехнологий привело к возможности создания защищённых информационных технологических систем, элементами которых выступают наноструктуры. Монопольное обладание определённой информацией оказывается зачастую решающим преимуществом в конкурентной борьбе и предопределяет тем самым высокую ценность "информационного фактора".

ПРИЛОЖЕНИЕ. ЗНАЧЕНИЕ ОБЩЕНАУЧНЫХ МЕТОДОВ ПОЗНАНИЯ В НАНОТЕХНОЛОГИИ

Научное познание есть исторически меняющаяся деятельность, которая детерминирована с одной стороны характером исследуемых объектов и с другой стороны социальными условиями, свойственными каждому исторически определённому этапу развития цивилизации. Это касается и нанотехнологии.

Существует два типа познания – эмпирический и теоретический. Основными методами эмпирического исследования являются наблюдения, реальный эксперимент, описание, обработка статистических данных и другое. В теоретическом исследовании применяются методы идеализации – математические модели, метод восхождения от абстрактного к конкретному.

Стиль мышления – представление о нормах, объяснения, описания, доказательность, описание знания. Так, идеалы и описания средневековья отличны от тех, которые характеризуют науку нового времени. В средневековой науке опыт не есть критерий истины. Научная картина мира складывается в результате синтеза знаний, полученных в различных науках, и содержит общее представление о мире [149].

Философские идеи и принципы обосновывают как идеалы и нормы науки, так и содержательные представления научной картины мира. Философские обоснования науки не следует отождествлять с общим массивом философского знания. Философия базируется на всём культурном материале человека. Наука – лишь отдельная область этой культуры. И из большого поля философских проблем (что первично, что вторично), возникающих в культуре каждой исторической эпохи, наука использует в ка-

честве обосновывающих структур лишь некоторые её принципы.

В соответствии с двумя уровнями научного познания различают эмпирические и теоретические методы. К первым относят наблюдение, сравнение, измерение и эксперимент, ко вторым – идеализацию, формализацию, восхождение абстрактного к конкретному и др.

Рассмотрим методы эмпирического и теоретического познания в нанотехнологии.

Эмпирические методы

Наблюдение – это целенаправленное систематическое восприятие объекта, доставляющее первичный материал для научного исследования. Как метод научного познания наблюдение дает исходную информацию об объекте, необходимую для его дальнейшего исследования.

Сравнение и измерение. Сравнение представляет собой метод сопоставления объектов с целью выявления сходства или различия между ними. Если объекты сравниваются с объектом, выступающим в качестве эталона, то такое сравнение называется измерением. С помощью измерения устанавливаются численные характеристики объектов, а это имеет большое значение для многих областей научного познания, где необходимы точные количественные характеристики изучаемых объектов, прежде всего в естественных и технических науках. Что касается сравнения, то на этом методе основаны такие науки, как сравнительная анатомия, сравнительная эмбриология, сравнительное историческое языкознание и некоторые другие [149, 150].

Эксперимент (лат. – опыт, проба) — метод исследования объекта, при котором исследователь активно воздействует на объект, создает искусственные условия, необходимые для выявления определённых его свойств. Различают натуральный и модельный эксперимент. Если первый

ставится непосредственно с объектом, то второй — с его заместителем — моделью. Под моделью понимается мысленно представляемая или материально реализованная система, отображающая или воспроизводящая объект исследования, способная дать его свойства и давать или получать информацию о самом объекте. Моделью может быть как материальный предмет (например, модель самолета, испытываемая в аэродинамической трубе), так и мысленная копия объекта. Моделирование находит широкое применение в науке и технике — в физике, математике, кибернетике, в аэродинамике, кораблестроении, гидростроительстве. Применяется моделирование и в обществоведении, политике, теоретической и практической работе юриста.

Модельный эксперимент находит применение при решении многих криминалистических задач, в том числе при разработке планов по оперативному задержанию преступника, при составлении рекомендаций по криминалистической технике и т.п.

Теоретические методы

Идеализация. Этот метод основан на универсальном мыслительном приёме, применяемом в любом познавательном процессе — абстрагировании (лат. – отвлечение), которое представляет собой мысленное отвлечение от одних свойств предмета и выделение других его свойств. Результатом абстрагирования являются абстракции — понятия, категории, содержанием которых являются существенные свойства и связи явлений. В процессе последовательного абстрагирования образуются абстракции всё более высокой степени общности (планета Земля — планета Солнечной системы — планета — небесное тело — тело). Такое абстрагирование называется многоступенчатым.

Метод идеализации находит широкое применение в научном позна-

нии. Он позволяет переходить от эмпирических законов к теоретическим, формулировать их на языке науки. В современной науке всё более широкое применение находит формализация – метод изучения некоторых областей знания в формализованных системах с помощью искусственных языков. Таковы, например, формализованные языки химии, математики, логики.

Одним из наиболее влиятельных направлений философского мышления является позитивизм. Как самостоятельное течение позитивизм оформился в 30-е годы XIX века. В центре внимания позитивистов неизменно находился вопрос о взаимоотношении философии и науки. Главный тезис позитивизма состоит в том, что всё подлинное, положительное («позитивное») знание в действительности может быть получено лишь в виде результатов отдельных специальных наук или их «синтетического» объединения, и что философия как особая наука, претендующая на содержательное исследование особой сферы реальности, не имеет права на существование.

Позитивизм объявил единственным источником истинного знания конкретные частные науки и выступил против философии как метафизики, но за философию как особую науку. Под метафизикой он понимает умозрительную философию бытия (онтологию, гносеологию).

Позитивизм – философия позитивного знания, отвергающая теоретические спекуляции и умозрения как средства получения знания. Он утверждает, что только совокупность наук даёт право говорить о мире в целом. То есть, если философия научна, то она должна распрощаться с попыткой судить о мире в целом [149].

Общенаучные методы познания нашли применение во всех или почти во всех науках. Их своеобразие, и отличие от всеобщих (философских) методов в том, что они находят применение не на всех, а лишь на определённых этапах процесса познания. Например, индукция играет ведущую

роль на эмпирическом, а дедукция – на теоретическом уровне познания. Анализ преобладает на начальной стадии исследования, а синтез – на заключительной и т.д.

Характеристика некоторых общенаучных методов исследования

На эмпирическом уровне находят применение следующие методы научного познания: наблюдение и эксперимент. Наблюдение – это преднамеренное и целенаправленное восприятие явлений и процессов без прямого вмешательства в их течение, подчинённое задачам научного исследования. Основные требования к научному наблюдению следующие: однозначность цели, замысла; системность в методах наблюдения; объективность; возможность контроля либо путём повторного наблюдения, либо с помощью эксперимента.

Наблюдение используется, как правило, там, где вмешательство в исследуемый процесс нежелательно либо невозможно. Эксперимент в отличие от наблюдения – это метод познания, при котором явления изучаются в контролируемых и управляемых условиях. Эксперимент, как правило, осуществляется на основе теории или гипотезы, определяющих постановку задачи и интерпретацию результатов.

Методами обработки и систематизации знаний эмпирического уровня являются анализ и синтез. Анализ – процесс мысленного, а нередко и реального расчленения предмета, явления на части (признаки, свойства, отношения). Процедурой, обратной анализу, является синтез. Синтез – это соединение выделенных в ходе анализа сторон предмета в единое целое.

Значительная роль в обобщении результатов наблюдения и экспериментов принадлежит индукции, особому виду обобщения данных опыта. При индукции мысль исследователя движется от частного (частных факто-

ров) к общему. Противоположностью индукции является дедукция, движение мысли от общего к частному. В отличие от. индукции, с которой дедукция тесно связана, она в основном используется на теоретическом уровне познания.

В процессе познания используется и такой приём, как аналогия – умозаключение о сходстве объектов в определённом отношении на основе их сходства в ряде иных отношений. С этим приёмом связан метод моделирования, получивший особое распространение в современных условиях. Моделирование используется в тех случаях, когда сам объект либо труднодоступен, либо его прямое изучение экономически невыгодно и т.д. [149].

Существенное место в современной науке занимает системный метод исследования или (как часто говорят) системный подход, Системный подход – это способ теоретического представления и воспроизведения объектов как систем.

Неопозитивизм – очередной этап в развитии позитивизма. Неопозитивизм начинается с 20-х годов XX века и продолжается до настоящего времени. Неопозитивизм часто называется на Западе аналитической философией. Неопозитивизм, уходя от решения коренных философских проблем, сосредотачивается на частных логико-методологических исследованиях, на анализе языка науки.

Предметом философии не может быть теория познания, так как её решения заставляют выходить на мировоззренческую проблематику.

Неопозитивизм истолковывал истину как совпадение высказываний с непосредственным опытом человека.

Философия вообще не имеет предмета исследования, так как не является содержательной наукой о какой-то реальности, а представляет собой род деятельности, особый способ теоретизирования.

Основную задачу неопозитивизм видит в анализе логической структуры языка, терминов и предложений, которые употребляются в научном

языке.

Одной из важных задач является отделение предложений, которые имеют смысл, от тех которые лишены его с научной точки зрения, и таким образом очищение науки от бессмысленных предложений. Неопозитивисты различают три типа осмысления предложений:

1. Высказывания об эмпирических фактах (если говорят о фактах и ни о чём более);
2. Предложение, содержащие логические следствия этих высказываний и построенные в соответствии с логическими правилами (могут быть сведены к высказываниям об эмпирических фактах);
3. Предложения логики и математики (не содержат высказывание о фактах, не дают нового знания о мире, необходимые для формального преобразования уже имеющегося знания).

Задача философии – логический анализ научных высказываний.

Чтобы выяснить имеет ли предложение смысл, необходим специальный метод — верификация (от лат. verus – истинный и facio – делаю).

Принцип верификации – любое высказывание в науке, практике, философии подлежит опытной проверке на истинность.

Истина – совпадение высказывания с непосредственным опытом человека. Вопрос, ответ на который не может быть проконтролирован, верифицирован в опыте, называется "псевдовопросом".

Целесообразно рассматривать законы развития электроники, которая составляет основополагающую часть всей нанотехнологии [151, 152].

1. Закон прогрессивной эволюции электроники. В вакуумных устройствах с одинаковой функцией переход от поколения к поколению происходит при наличии необходимого научно-технического уровня и социально-экономической целесообразности. Прогрессивная эволюция продолжается до максимального значения показателя эффективности Q, например количества элементов в единице объёма кристалла или изделия наноэлектроники.

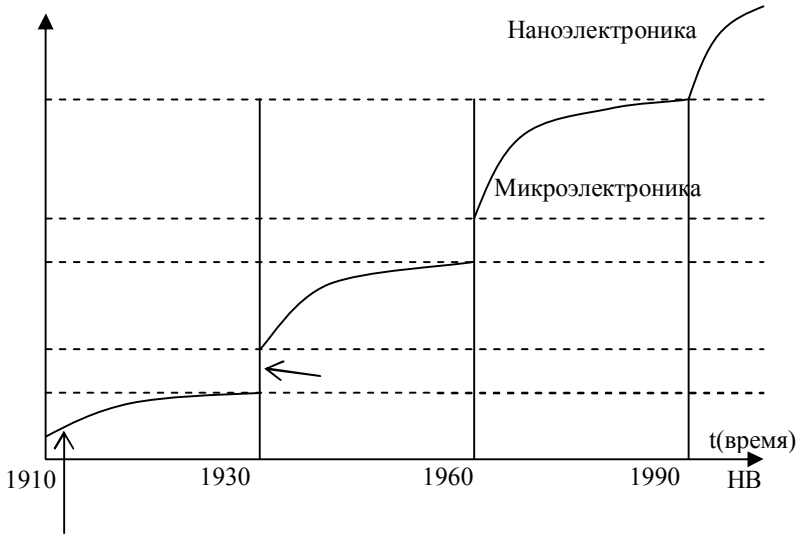


Рис. П.1. Схематическая иллюстрация законов прогрессивной эволюции и скачкообразного развития

2. Закон скачкообразного развития наноэлектроники. Этот закон отражает революционные изменения в процессе развития (рис. П.1). Переход к каждой очередной стадии происходит при исчерпывании природных возможностей человека в улучшении показателей эффективности выполнения функций данным устройствам.

3. Закон соответствия между функцией и структурой. Главная суть закона заключается в том, что в правильно спроектированном электронном устройстве каждый элемент имеет вполне определённую функцию по обеспечению его работоспособности. Исключение элемента приводит ухудшению какого-либо показателя эффективности.

4. Первый и второй законы развития наноэлектроники тесно связаны с диалектическим законом перехода количественных изменений в качественные. Главная проблема в нанотехнологии – проблема верификации, заключающаяся в невозможности в настоящее время проверить некоторые теории опытным путём. Это связано с тем, что почти во всех экспериментах используется метод косвенных измерений. Соотношение принципов

верификации и фальсификации, то есть не подтверждение на истинность, а опровержение неистинности, это также проблема нанотехнологии. Наряду с вышеуказанными проблемами в нанотехнологии, как нигде более очевидными становятся размытые грани между различными категориями философии, такими как “причина – повод – следствие”, “количество – условие – качество”, “единичное – особенное – всеобщее”, “сущность – наблюдение – явление”, “необходимость – действие – случайность”, ”возможность – предрасположенность – действительность” и др.

Анализ задач математического, физического и технического моделирования в нанотехнологии позволяет констатировать отсутствие в природе *физического нуля*, то есть абсолютного ничего, пустоты. Так, например, предельное остаточное давление в вакуумной камере – величина бесконечно малая, но не равная нулю, тоже самое можно сказать относительно силы тяжести или абсолютно отрицательной температуры (абсолютного нуля).

Таким образом, в нанотехнологии мы имеем дело с бесконечно малыми (не равными нулю) величинами. В общем случае, бесконечно малые величины – антиподы бесконечно больших. Например, при увеличении радиуса до бесконечно большой величины, окружность превращается в прямую, а при уменьшении до бесконечно малой – в точку. Таким образом, прямая – синоним бесконечно большой величины, точка – бесконечно малой.

Техноэволюция нанотехники осуществляется под действием закона информационного отбора Б.И. Кудрина [151, 152]. Действие этого закона проявляется в наследственных изменениях вида нанотехники точнее – популяций изделий, занимающих определенную экологическую нишу. В отличие от закона естественного отбора Ч. Дарвина в вакуумной технике имеет место более разумная целенаправленная изменчивость: появляются, как правило, только такие новые варианты конструкторских решений, которые по основным показателям (критериям эффективности) обеспечивают повышение конкурентоспособности (см. рис. П.1), а подавляющее

большинство изменений связано с улучшением наиболее актуальных показателей, которые в данный момент требуется улучшить, например, скорость откачки вакуумного насоса и величину предельного вакуума.

Техноценоз – сообщество всех изделий и оборудования конкретного участка, цеха или предприятия для определенного момента или отрезка времени. Существующие НИИ или КБ нанотехники в основном сосредоточены на изучении и проектировании отдельных изделий, а изучением, проектированием техноценозов никто серьезно не занимается.

В каждом конкретном случае существует свой оптимальный состав оборудования в техноценозе, который находится между двумя крайними предельными случаями, когда все изделия в техноценозе различны и все изделия одинаковы.

Задача выбора оптимального состава оборудования техноценоза – очень сложная задача нелинейного программирования.

Попробуем разобраться, какую помощь могут оказать людям нанороботы и какую угрозу для человечества они представляют.

Перспективы просто фантастические, иначе не скажешь. Например, за счёт внедрения в организм молекулярных роботов, предотвращающих старение клеток, а также перестраивающих и "облагораживающих" ткани организма можно будет достигнуть бессмертия человека, не говоря об оживлении и излечении безнадежно больных и людей, которые были заморожены методами крионики.

Наноробот, введённый в организм человека, сможет самостоятельно передвигаться по кровеносной системе и очищать его от микробов или зарождающихся раковых клеток, а саму кровеносную систему – от отложений холестерина. Он сможет изучить, а затем и исправить характеристики тканей и клеток.

В промышленности произойдёт замена традиционных методов производства сборкой молекулярными роботами предметов потребления не-

посредственно из атомов и молекул, вплоть до персональных синтезаторов и копирующих устройств, позволяющих изготовить любой предмет [153].

Замена произойдёт и в сельском хозяйстве: комплексы из молекулярных роботов придут на смену "естественным машинам" для производства пищи растениям и животным. Их искусственные аналоги будут воспроизводить те же химические процессы, что происходят в живом организме, однако более коротким и эффективным путём.

Биологи смогут "внедряться" в живой организм на уровне атомов и станут возможными и "восстановление" вымерших видов, и создание новых типов живых существ, в том числе биороботов.

В кибернетике произойдёт переход к объёмным микросхемам, а размеры активных элементов уменьшаться до размеров молекул. Рабочие частоты компьютеров достигнут терагерцовых величин. Получат распространение схемные решения на нейроноподобных элементах. Появится долговременная быстродействующая память на белковых молекулах, ёмкость которой будет измеряться терабайтами. Станет возможным "переселение" человеческого интеллекта в компьютер.

За счёт внедрения логических наноэлементов во все атрибуты окружающей среды она станет "разумной" и исключительно комфортной для человека. На всё это, по разным оценкам, понадобится около 100 лет.

Однако новые открытия могут иметь и негативные последствия. Представим себе, что в устройстве, предназначенном для разборки промышленных отходов до атомов, произойдет сбой, и оно начнёт уничтожать полезные вещества биосферы, обеспечивающие жизнь людей. При этом самым неприятным может оказаться то, что это будут нанороботы, способные к самовоспроизводству (саморепликации, размножению).

Можно представить себе и нанороботов, запрограммированных на изготовление уже существующего оружия. Овладев секретом создания подобного робота или каким-то образом достав его, воспроизвести универ-

сального "малыша" в большом количестве сможет небольшая группа людей или даже террорист-одиночка.

Отметим также принципиальную возможность создания разрушительных устройств, например, воздействующих на определённые этнические группы или заданные географические районы. Как видно, нанороботы, вышедшие из-под контроля, могут стать оружием массового поражения.

Так или иначе, но главный шаг на пути создания нанороботов группа нью-йоркских учёных, по собственному признанию, уже сделала. Судя по тому, что на создание первой ДНК-машины ушло около 10 лет, первый наноробот появится максимум лет через 5-7.

Устройства микроэлектромеханических систем (MEMS) действуют как и устройства макроразмеров и даже выглядят также – с моторами, передачами и рычагами, изготовленными из стекла, керамики или металла. Но наноразмерные структуры, в частности NEMS, будут строиться, и действовать совершенно по-другому: они формируются и функционируют на основе других физических законов. На молекулярном уровне перестают действовать законы механики, используемые для расчётов узлов обычных машин. Законы сопротивления материалов и гидравлики уже не применимы – вместо этого вступают в действие законы квантовой механики, которые приводят к совершенно неожиданным, с точки зрения классической механики, последствиям.

Сегодня практическая нанотехнология ориентирована на решение следующих задач:

— создание твёрдых тел и поверхностей с требуемой молекулярной структурой;
— создание новых химических веществ посредством конструирования молекул (с участием и без участия химических реакций);
— разработка устройств различного функционального назначения (компоненты наноэлектроники, нанооптики, наноэнергетики, нанороботы и

нанокомпьютеры, нанолекарства, наноинструменты и т.д.);

— создание наноразмерных самоорганизующихся и самореплицирующихся структур.

Инструментальный базис нанотехнологий, позволяющий учёным и исследователям не только визуализировать атомные структуры, но и манипулировать отдельными атомами и строить новые молекулы, основан на использовании так называемого эффекта туннелирования электронов. Его применение на вершинах зондов специальных конструкций позволило достичь высокой пространственной разрешающей способности управления атомно-молекулярными реакциями в отличие от известных групповых технологий осаждения материалов, методов оптической литографии, эпитаксии, а также электронной литографии, где высокая энергия фокусируемых электронов приводит к значительному разрушению используемых материалов.

За 20 с небольшим лет с момента появления техники сканирующей зондовой микроскопии и изобретения сканирующего туннельного, а затем и атомно-силового микроскопов, в разных странах были получены впечатляющие результаты по наблюдению наноразмерных частиц и структур на их основе и поставлена задача создания технологических машин, позволяющих осуществить атомно-молекулярную сборку вещества и конструирование отдельных узлов и устройств различного функционального назначения.

Внедрение наносхемотехники и нанороботов позволит создать микроскопические компьютеры небывалой производительности [151]. Более того, они станут саморемонтирующимися и самовоспроизводящимися. Это означает, что в зависимости от потребности вычислительной системы она будет увеличиваться и уменьшаться сама. Применение десятиатомных транзисторов позволит подойти вплотную к имитации мыслительных процессов человека и уже к середине XXI столетия создать настоящий искусственный интеллект – саморазвивающуюся мыслительную среду. Станет

возможным также и внедрение человеческого сознания в компьютерные программы.

Впервые идея о новом направлении была высказана лауреатом Нобелевской премии Р.Фейнманом в 1959 г. Позже, в 1980-х годах, появились приборы, способные оперировать с отдельным атомом, например, взять его и переставить на другое место. Созданы отдельные элементы нанороботов: опытный механизм шарнирного типа на основе нескольких цепочек ДНК, способный сгибаться и разгибаться по химическому сигналу, первые образцы нанотранзисторов или электронных переключателей, состоящие из небольшого числа атомов. В нанотехнологию ежегодно инвестируются сотни миллионов долларов, разработками заняты многие десятки фирм.

Нанороботы – гипотетические механизмы размером десятки и сотни нанометров (миллионные доли миллиметра), разработка которых начата не так давно. Как и роботы обычных размеров, нанороботы будут иметь самые различные конструкции и назначения: смогут двигаться, производить механические и другие операции, управляться извне или встроенными компьютерами. Они смогут собирать механизмы, создавать новые вещества; для таких устройств используют названия "ассемблер" (сборщик) или "репликатор". Возможна настройка их на переработку или уничтожение каких-либо веществ. Венцом этого направления могут стать нанороботы, самостоятельно собирающие свои копии, то есть практически способные к размножению.

Нанороботов условно разделяют на два вида: способных конструировать что-либо, например, самовоспроизводиться (ассемблеры), или деконструировать, разбирать (дизассемблеры). Молекулярные ассемблеры – основной инструмент человека для манипуляций в наномире. Любой вирус в определённом смысле также является ассемблером. Нанороботов нередко так и называют – "искусственные вирусы".

Микроскопические роботы, способные манипулировать объектами

размером в несколько нанометров (10^{-9} м), могли бы оказаться весьма полезны во многих отраслях народного хозяйства.

В то же время в настоящий момент у роботов отсутствуют навыки обращениями с "предметами" меньше чем несколько микрон (10^{-6} метра). Правда, не очень понятно, уместно ли для таких микроскопических тварей название "робот".

Конечно, за кипучей деятельностью этих механизмов невозможно наблюдать невооружённым глазом, требуется сканирующий электронный микроскоп. Идея изобретения состоит в том, чтобы использовать микроскоп не только для наблюдения, но и для обратной связи – отдачи роботу производственных указаний.

Для реализации такого взаимодействия использованы свойства сплавов с эффектом памяти формы (Shape Memory Alloys – SMA), пластически деформированные изделия из которых способны при нагревании восстанавливать свои первоначальные очертания. Собственно, SMA-сплавам на титано-никелевой основе уже давно прочат переворот в нанотехнологиях, однако идея использования луча микроскопа для нагрева манипулятора запатентована только сейчас.

Как показывает практика, манипуляция объектами размером меньше микрона требует создания манипуляторов микронного размера, причём сила воздействия такого привода должна быть неуловимо мала. Существующие типы приводов (электромагнитный, пьезоэлектрический) не удовлетворяют этим параметрам.

SMA-устройства раньше не делались меньше, чем в несколько сот микрон. Следовательно, было необходимо ответить на два вопроса. Во-первых, каковы минимальные размеры, при которых сплавы сохраняют свои свойства? И, во-вторых, насколько малый объект можно выборочно нагреть, чтобы привести устройство в действие?

Предыдущие исследования показали, что плёнка из SMA на титано-

никелевой основе с добавлением кремния и оксида кремния толщиной в 100 нанометров (всего около 200 атомных слоев) всё еще способна предсказуемо менять форму при нагревании.

Что же касается электронного сканирующего микроскопа, то его лучом можно производить нагрев области микронного диаметра. Для нагревания до необходимой температуры образца размером, например, 4×10×100 мкм необходимо выделение энергии $1{,}3 \cdot 10^{-5}$ Дж, то есть возможной мощности луча $2 \cdot 10^{-3}$ Вт достаточно, чтобы выделить тепло за 6 мс.

Путём деформации достаточно толстой перфорированной плёнки из SMA и последующего нагрева лучом микроскопа удалось продемонстрировать прототип манипулятора с диаметром рабочего элемента 2 мкм и длиной в 20 мкм.

Проект манипулятора уже достаточно подробно описан в литературе. У позиционирующего устройства "руки" может быть шесть степеней свободы. Каждая будет управляться своим "храповиком", приводимым в действие давлением инертного газа, цилиндрами будут служить углеродные нанотрубки. На первый взгляд все выглядит достаточно просто, однако такая "рука" ещё не создана.

Взгляд с общенаучных методов познания позволил авторам предъявить новые подходы и концепции в создании оборудования для нанотехнологий наиболее ярко отражённых в работах [154-159].

Нанотехнолгическая революция – процесс длительный и займёт десятилетия. В соответствии с Федеральной целевой программой, на развитие нанотехнологий выделяется в год порядка 100 млрд. рублей. По оценкам специалистов, рынок нанотехнологий России в 2013÷2015 гг. будет измеряться триллионом рублей, а мировой рынок превысит триллион долларов.

ЛИТЕРАТУРА

1. Анил К. Джейн, Жанчанг Мао, Моиуддин К.М. Введение в искусственные нейронные сети // Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, 1996. – P. 31-44.
2. McCulloch W.S. and Pitts W. A logical Calculus of Ideas Immanent in Nervous Activity // Bull. Mathematical Biophysics, Vol. 5, 1943. – P. 115-133.
3. Hertz J., Krogh A. and Palmer R.G. Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, Mass., 1991.
4. Haykin S, Neural Networks: A Comprehensive Foundation, MacMillan College Publishing Co., New York, 1994.
5. Rosenblatt R. Principles of Neurodynamics, Spartan Books, New York, 1962.
6. Mimnsky M. and Papert S. Perceptrons: An Introduction to Computational Geometry, MIT Press, Cambridge, Mass., 1969.
7. Hopfield J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities // In Proc. National academy of Sciences, USA 79, 1982. – P. 2554-2558.
8. Werbos P. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences // PhD Thesis, Dept. of Applied Mathematic, 1974.
9. Дорогов А.Ю., Алексеев А.А. Категории ядерных нейронных сетей / Труды Всерос. науч.-техн. конф.“Нейроинформатика-99” // Сб. науч. тр. Часть 1. – М., 1999. – С. 55-64.
10. Дорогов А.Ю., Алексеев А.А. Быстрые нейронные сети // Труды международной научно-технической конференции “Пятьдесят лет развития кибернетики”. – Санкт-Петербург, 1999. – С. 120-121.
11. Пак М.М Моделирование нейронных сетей на основе твёрдотельных объектов. Автореферат дисс. на соиск. уч. ст. к.т.н. – М., 2007. – 23 с.
12. Zurada J. M. Introduction to artificial neural systems // PWS Publishing

Company, 1992. – 785 p.

13. Haykin S. Neural networks. A comprehensive foundations. McMillan College Publ. Co. N.Y., 1994. – 696 p.
14. Ezhov A.A., Ventura D. Quantum neural networks. In: Future directions for Intelligent Information Systems and Information Sciences. Kasabov N, Physica-Verlag, Heidelberg, 2000. – P. 213-235 (Studies in Fuzziness and Soft Computing, vol.45).
15. Deutsch D. The fabric of reality. Alen Lane: The Penguin Press, 1997.
16. Narayanan A. Quantum algorithms // Technical Report 374; Department of Computer Science, University of Exeter, 1998.
17. Perus M. Common mathematical foundations of neural and quantum informatics // Zeitschrift fur Angewandte Mathematik und Mechanik, 1998; 78: – P. 23-26.
18. Behrman E.C., Niemel J., Steck J.E. et al. A quantum dot neural network. In: Proceedings of the 4th Workshop on Physics of Computation, Boston, 1996. – P. 22-24.
19. Menneer T., Narayanan A. Quantum inspired neural networks // Department of Computer Sciences, University of Exeter, UK, 1995. http:/www.dcs.ex.ac.uk/reports/reports.html
20. Menneer T. Quantum artificial neural networks // PhD Thesis, Faculty of Science, University of Exeter, UK, 1998.
21. Menneer T., Narayanan A. Quantum artificial neural networks vs. Classical artificial neural networks: Experiments in Simulation // Proceedings of the Fifth Joint Conference on Information Sciences, Atlantic City, 2000; 1. – P. 757-759.
22. Ventura D. and Martinez T. Quantum associative memory with exponential capacity // Proceedings of the International Joint Conference on Neural Networks, 1998. – P. 509-513.
23. Chrisley R.L. Learning in Non-superpositional Quantum Neurocomputers,

In Pylkkanen, P. and Pylkko, P. (Eds.) Brain, Mind and Physics. IOS Press, Amsterdam, 1997. – P. 126-139.

24. Крисли Р.Л. Бомовские квантовые нейронные сети (в сборнике [23]).
25. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. – Новосибирск: Наука (Сиб. отделение), 1996. – 276 с.
26. Le Cun Y., Denker J.S., Solla S.A. Optimal Brain Damage // Advances in Neural Information Processing Systems II (Denver 1989). – San Mateo, Morgan Kaufman, 1990. –P. 598-605.
27. Горбань А.Н. Обучение нейронных сетей. – М.: Изд-во СССР- США СП "ParaGraph", 1990. – 160 с (English Translation: Traning Neural Networks // AMSETransaction, Scientific Siberian, A, 1993, Vol. 6. Neurocomputing. – P. 1-134).
28. Prechelt L. Comparing Adaptive and Non-Adaptive Connection Pruning With PureEarly Stopping // Progress in Neural Information Processing, Springer, 1996, Vol. 1. – P. 46-52.
29. Cybenko G. Approximation by superposition of a sigmoidal function // Mathematics of Control, Signals, and Systems, 1989, Vol. 2. – P. 303 - 314.
30. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators // Neural Networks, 1989, Vol. 2. – P. 359 - 366.
31. Kochenov D.A., Rossiev D.A. Approximations of functions of C[A,B] class by neural-net predictors (architectures and results) // AMSE Transaction, Scientific Siberian, A. 1993,Vol. 6. Neurocomputing. – P. 189-203. Tassin, France.
32. Gilev S.E., Gorban A.N. On completness of the class of functions computable byneural networks // Proc. of the World Congress on Neural Networks (WCNN '96), San Diego, CA, Lawrens Erlbaum Accociates, 1996. – P. 984-991.
33. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа

переменных // Докл. АН СССР, 1956. Т. 108, №. 2. – С. 179-182.

34. Арнольд В.И. О функциях трёх переменных // Докл. АН СССР, 1957. Т. 114, №4. – С. 679-681.
35. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного // Докл.АН СССР, 1957. Т. 114, № 5. – С. 953-956.
36. Витушкин А.Г. О многомерных вариациях. – М.: Физматгиз, 1955.
37. Stone M.N. The generalized Weierstrass approximation theorem // Math. Mag., 1948.V.21. – PP. 167-183, 237-254.
38. Шефер Х. Топологические векторные пространства. – М.: Мир, 1971. – 360 с.
39. Шик А.Я. Квантовые нити // Соросовский Образовательный Журнал, 1997. № 5. – С. 87-92.
40. Демиховский В.Я. Квантовые ямы, нити, точки: Что это такое // Соросовский Образовательный Журнал, 1997. № 5. – С. 81-86.
41. Молекулярно-лучевая эпитаксия и гетероструктуры / Под ред. Л. Ченга, Л. Плога. – М.: Мир, 1989. – 582 с.
42. Беляховский В.И. Физические основы полупроводниковой технологии // Соросовский Образовательный Журнал. 1998. № 10. – С. 92-98.
43. Крисилов В.А., Олешко Д.Н., Трутнев А.В. Применение нейронных сетей в задачах интеллектуального анализа информации // Труды Одесского политехнического университета, 1999. Вып.2 (8). – С. 134.
44. Авеньян Э.Д. Алгоритмы настройки многослойных нейронных сетей // Автоматика и телемеханика, 1995. № 5. – С.106-118.
45. Минский М., Пейперт С. Персептроны. – М.: Мир, 1971. – С. 261.
46. Tiwary S., Rana F., Chan K., Hanafi H., Chan Wei, Buchanan D. Volatile and non-volatile memories in silicon with nano-crystalstorage // IEEE International Electron Device Meeting. – Washington, DC, USA, 1995. – P. 521-524.

47. Tiwary S., Rana F., Hanafi H., Harstein A., Crabbe E.F. Chan K. A silicon nanocrystals based memory // Appl. Lett., 1996. 68. – P. 1377-1379.
48. Bonafos C., Carrada M., Cherkashin N., Coffin H., Chassaing D., Assayag G.B. Claverie A., Muller T., Heinig K.H., Perego M., Fanciulli M., Dimitrakis P., Normand P. Manipulation of two-dimensional arrays of Si nanocrystals embedded in thin SiO2 layers by low energy ion implantation // J. appl. Phys., 2004. 48. – P. 1175-1179.
49. Чаплыгин Ю.А. Нанотехнология в электронике. – М.: МИЭТ, 2005. – С. 153- 170.
50. Patrick P. Minimisation methods for training feedforward Neural Networks // Neural Networks, 1994. Volume 7, Number 1. – P. 1-11.
51. Джеффри Е. Хинтон. Как обучаются нейронные сети // В мире науки, 1992. №11-12. – С. 103-107.
52. Rumelhart B.E., Minton G.E., Williams R.J. Learning representations by back propagating error // Wature, 1986. V. 323. – P. 1016-1028.
53. Bohm D. and Hiley B. The Undivided Universe: An Ontological Interpretation of Quantum Mechanics. Routledge, London, 1993.
54. Chrisley R. Learning in non-superpositional quantum neurocomputers. In P.Pylkkanen and P.Pylkko, editors, Brain, Mind and Physics, IOS Press, Amsterdam, 1997. – P. 126-139.
55. L. de Broglie. Non-Linear Wave Mechanics: A Causal Interpretation. Elsevier, New York, 1960.
56. Deutsch D. Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. Royal Society of London, A400, 1985. – P. 97-117.
57. Everett H. «relative state» formulation of quantum mechanics // Review of modern physics, 1957. Vol.29. – P. 454-462.
58. Gould L. Reflections on the relevance of nonlocality to cognitive science and the philosophy of mind. In P.Pylkkanen and P.Pylkko, editors, New directions in cognitive science: Proceedings of the international symposium,

Saarisetka, , Lapland, Finland, 1995. – P. 104-114. Finnish Association of Artificial Intelligence, Helsinki.

59. Hiley B. and Pylkkanen P. Representation and interpretation in quantum physics. In D. Peterson, editor, Forms of Representation. – Intellect Books, Exeter, 1996.
60. Menneer T. Quantum Artificial Neural Networks. University of Exeter, Exeter, D.Phil. thesis, 1999.
61. Menneer T. and Narayanan A. Quantum-inspired neural networks. Technical Report 329, Department of Computer Science, University of Exeter, 1995.
62. Moore M. and Narayanan A. Quantum-inspired computing. Technical Report 341, Department of Computer Science, University of Exeter, 1996.
63. Perus. Analogies between quantum and neural processing – consequences for cognitive science. In P.Pylkkanen and P.Pylkko, editors, New directions in cognitive science: Proceedings of the international symposium, Saarisetka, Lapland, Finland, Helsinki, 1995. – P. 115-123. Finnish Association of Artificial Intelligence.
64. Shor P.W. Algorithms for quantum computation: Discrete log and factoring. In S. Goldwasser, editor, Proceedings of the 35th Annual Symposium on the Foundations of Computer Science, 1994. – P.124-134. IEEE Computer Society Press.
65. Ventura D. Implementing competitive learning in a quantum system. In Proceedings of the 1999 International Joint Conference on Neural Networks, 1999.
66. Shor P.W. Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer // SIAM Journal on Computing, 1997. 26. – P. 1484-1509.
67. Grover L.K. A fast quantum mechanical algorithm for database search // Proc. of the 28th Annual ACM Symposium on the Theory of Computation, 1996. – P. 212-219.

68. Simon D. SIAM Journal on Computing, 1997. 26. – P. 1474.
69. Deutsch D. and Jozsa R. Rapid solutions of problems by quantum computations //Proc. Roy.Soc.London, 1992. Ser. A. – PP. 439, 553.
70. Hogg T. Journal of Atrificial Intelligence Research, 1996. №4. – P. 91.
71. Jackson J. Journal of Computer and System Sciences, 1997. №55. – P. 414.
72. Bshouty N.H. and Jackson J. Proceedings of the 8th Annual Conference on Computational Learning Theory, Santa Cruz, 1995. W. Maass. ed. (ACM Press).
73. Fredkin E. and Toffoli T. Conservative Logic, Int. J. Theor. Physics, 1982. №21. – P. 219.
74. Bernstein E. and Vazirani U. SIAM J. Comput, 1997. №26. – P. 1411.
75. Biron D. et al. Proceedings of the 1st NASA Int. Conf. Quantum Computing and Quantum Communications, Palm Springs, C.Bennet (ed), 1998.
76. Steane A. M. Quantum computing // Rep.Progr.Phys, 1998. V. 61. – P. 117-173.
77. Wilson H. and Cowan J. Amathematical theory of the functional dynamics of cortical and thalamic nervous tissue // Cybernetic, 1973. V.13. N 1. – P. 55- 80.
78. Ораевский А. Н. О квантовых компьютерах // Квантовая электроника, 2000. Т.30. № 5. – С. 457 – 458.
79. Feynman R. Lectures of physics. Addison-Wisley, 1964. V.2.
80. Ландау Л. Д., Лифшиц Е. М. Квантовая механика. – М.: Наука, 1964. – С. 173-174.
81. Barenco, Bennet C. H., Cleve R., DiVincenzo D.P., Margolus N., Shor P., Steator T., Smolin I. A. and Weinfurter H. Elementary gates for quantum computation // Phys. Rev. A, 1995. V.52. – P. 3457-3467.
82. Gershenfeld N. A. and Chuang I. L. .Bulk spin–resonanse quantum computation // Science, 1997. V.275. – P. 350-356.
83. Ventura D. and Martinez T. Quantum associative memory with exponential

capacity // Proc. Int. Joint Conf. on Neural Networks. Anchorage, 1998. – P. 509-513.

84. Кессель А. Р., Ермаков В. Л. Виртуальные кубиты – многоуровневость вместо многочастичности // ЖЭТФ, 2000. Т.117. № 3. – С. 517-525.
85. Ивашов Е.Н., Реутова М.В. Технологические устройства для получения наноструктур с использованием углеродных нанотрубок // Сборник докладов научно-технической конференции «Вакуумная наука и техника». – М.:МГИЭМ, 2003.
86. Ивашов Е.Н., Павлов А.Ю., Пискарев Д.А., Реутова М.В., Степанов М.В. Колебательный контур для наноэлектроники. Патент РФ на ПМ №40539. Опубл. 16.03.2004, Бюл. №25.
87. Петренко А.И., Семенков О. И. Основы построения систем автоматизированного проектирования. – 2-е изд., стер. – Киев: Вища шк. Головное издательство, 1985. – 294 с.
88. Семенкин Е.Н., Семенкина О.Э., Терсков В.А. Методы оптимизации в управлении сложными системами: Учебное пособие. – Красноярск: сибирский юридический институт МВД России, 2000. – 254 с.
89. Ивашов Е.Н., Реутова М.В. Применение метода Саати при структурировании множества альтернатив получения углеродных нанотрубок, фуллеренов и кластеров. Деп. Рук. ВИНИТИ № 2325. В 2003; 31.12.2003. – 12 с.
90. Смирнов С.А. Оценка интеллектуальной собственности. – М.: Финансы и статистика, 2002. – 352 с.
91. Ивашов Е.Н., Степанчиков С.В., Реутова М.В., Дульцев А.А. Расчёт магнитных систем вакуумного технологического оборудования. Свидетельство РФ об официальной регистрации программы для ЭВМ №2003611934. Зарегистр. в Реестре 22.08.2003.
92. Ивашов Е.Н., Степанчиков С.В., Реутова М.В., Самухов И.В. Расчёт электромагнитных систем вакуумного технологического оборудова-

ния TSE RIS. Свидетельство об официальной регистрации программа для ЭВМ №2003611935. Зарегистр. в Реестре 22.08.2003.

93. Ивашов Е.Н. Пак М.М. и др. Системы формирования и сканирования нанообъектов // НТК «Датчики и преобразователи информации систем измерения, контроля и управления». – М.: МГИЭМ, 2004, материалы конференции.
94. Ивашов Е.Н., Пак М.М. и др. Построение аналитических устройств наноэлетроники на основе квантомеханического подхода Деп. Рук. ВИНИТИ № 1204. В 2004; 13.07.2004.
95. Ивашов Е.Н., Пак М.М. и др. Механическое и полевое тестирование модифицированных наноструктур. Деп. Рук. ВИНИТИ № 1202. В 2004; 13.07.2004.
96. Ивашов Е.Н., Пак М.М. и др. Туннельный метод измерения нанорельефа поверхности. Деп. Рук. ВИНИТИ № 1201. В 2004; 13.07.2004.
97. Ивашов Е.Н., Пак М.М. и др. Оптоволоконная нанотехнология в электронике и методы её реализации. Деп. Рук. ВИНИТИ № 1203. В 2004; 13.07.2004.
98. Ивашов Е.Н., Пак М.М. и др. Устройство перемещения для нанотехнологии. Патент РФ на ПМ № 37580. Опубл. 27.04.2004, Бюл. № 12.
99. Ивашов Е.Н., Пак М.М. и др. Устройство для регистрации химического состава. Патент РФ на ПМ № 43104. Опубл. 27.12.2004, Бюл.№ 36.
100. Ивашов Е.Н., Пак М.М. и др. Устройство для получения нанодорожек. Патент РФ на ПМ № 42696. Опубл. 27.12.2004, Бюл. № 34.
101. Ивашов Е.Н., Пак М.М. и др. Измерительное устройство для нанотехнологии. Патент РФ на ПМ № 42697. Опубл. 27.12.2004, Бюл. № 34.
102. Ивашов Е.Н., Пак М.М. Искусственная нейронная сеть. Патент РФ на ПМ № 62609. Опубл. 27.04.2007, Бюл. № 9.
103. Ивашов Е.Н., Пак М.М. Устройство флэш-памяти. Патент РФ на ПМ № 66863. Опубл. 27.09.2007, Бюл. № 27.

104. Пак М.М. и др. Методы измерения наноструктур материалов на основе устройств получения нанодорожек, определения химического состава и нанорельефа // НТК «Вакуумная наука и техника». – М.: МГИЭМ, 2003, материалы конференции.
105. Пак М.М. и др. Определение наноструктуры поверхности подложки // Ежегодная научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ. – М.: МИЭМ, 2004, материалы конференции.
106. Пак М.М. и др. Устройство сканирования нанодорожек на подложке // Ежегодная научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ.– М.: МИЭМ, 2004, материалы конференции.
107. Ивашов Е.Н., Пак М.М. и др. Аспекты наноробототехники // IV Российский философский конгресс «Философия и будущее цивилизации». – М., 2005, материалы конференции.
108. Ивашов Е.Н., Пак М.М. и др. Методологические аспекты нанотехнологии // Всероссийская междисциплинарная конференция «Философия искусственного интеллекта». – М., 2005, материалы конференции.
109. Ивашов Е.Н., Пак М.М. и др. Моделирование наноструктур ассемблерами на основе искусственных нейронных сетей // НТК «Нанотехнологии 2005». – Владимир, 2005, материалы конференции.
110. Пак М.М. и др. Применение туннельного метода в исследовании наношероховатости поверхности подложки // НТК «Прогрессивные машиностроительные технологии: Образование через науку». – М.: МИЭМ, 2005, материалы конференции.
111. Пак М.М. и др. Формирование наноструктур на атомарном уровне // НТК «Прогрессивные машиностроительные технологии: Образование через науку». – М.: МИЭМ, 2005, материалы конференции.
112. Пак М.М. и др. Моделирование наноструктур ассемблерами на основе

искусственных нейроных сетей // Ежегодная научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ. – М.: МИЭМ, 2005, материалы конференции.

113. Ивашов Е.Н., Пак М.М. и др. Расчёт геометрических параметров квантовых точек при их автоматизированном проектировании Свидетельство РФ об официальной регистрации программы для ЭВМ №2005613155.
114. Пак М.М. и др. Построение системы квантовой нейронной обработки в автоматизированном проектировании искусственных нейросетей // Ежегодная научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ. – М.: МИЭМ, 2006, материалы конференции.
115. Пак М.М. и др. Квантовая нейронная обработка в автоматизированном проектировании искусственных нейросетей // XIV Международная студенческая школа-семинар «Новые информационнын технологии». – М.: МИЭМ, 2006, материалы конференции.
116. Парфенов И.И. Гущин Ю.Г. Тхам Ф.З. Пак М.М. Функциональная компьютерная систематика для моделирования трансформации информации в эргатических структурах управления // Системы управления и информационные технологии. – 2006, №1.2(23) (спецвыпуск рубрики «Перспективные исследования»).
117. Кравчук И.С., Тихоглаз Ю.С., Занг Н.Ч. Математическая модель и алгоритм управления качеством в кластерных системах сбора и обработки информации // Системы управления и информационные технологии, 2008, 1.2(31). – С. 299-303.
118. Кравчук И.С., Тихоглаз Ю.С., Ву Тхи Тует Ланг. Эволюционная стратегия управления в задачах распознавания образов // Системы управления и информационные технологии, 2008, 2.3(32). – С. 358-360.
119. Слободин М.Ю., Царев Р.Ю. Компьютерная поддержка многоатрибу-

тивных методов выбора и принятия решения при проектировании корпоративных информационно-управляющих систем. – СПб.: Инфода, 2004. – 223 с.

120. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978.
121. Резников Б.А. Методы и алгоритмы оптимизации на дискретных моделях сложных систем. – Л.: ВИКИ им. А.Ф. Можайского, 1983. – 215 с.
122. Schwefel H.-P. Evolution and Optimum Seeking. – N.Y.: Whiley Publ., 1995. – 612 p.
123. Goldberg D.A. Genetic algorithm in search, optimization and machine learning. Addison-Wesley, Reading MA, 1989.
124. Вальков В.М., Вершин В.Е. Автоматизированные системы управления технологическими процессами. – М.: Машиностроение, 1977.
125. Кравченко В.А., Цидилин С.М., Федосеева Т.Л. Алгоритмы решения задач многокритериальной оптимизации: Учебное пособие. – М.: МИЭМ, 1988. – 74 с.
126. Рапопорт Э.Я. Анализ и синтез систем автоматического управления с распределёнными параметрами: Учебное пособие.– М.: Высш. шк., 2005. – 292 с.
127. Оптимальное управление. Сборник. – М.: Знание, 1978. – 144 с. (Нар. Ун-т. Естественнонаучный фак.).
128. Производство тонкоплёночных структур в электронном машиностроении: Учебник для вузов в 2-х томах. Т.2 / А.Т.Александрова, Е.Н. Ивашов, С.В.Степанчиков и др. – М.: Машиностроение, 2006. – 427с.
129. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: «Горячая линия – Телеком», 2000.
130. Trusted Computer System Evaluation Criteria. Us Department of Defense 5200.28-STD, 1993.
131. Гостехкомиссия России. Руководящий документ. Концепция защиты средств вычислительной техники от несанкционированного доступа к

информации. – М., 1992.

132. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищённости от несанкционированного доступа к информации. – М., 1992.
133. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. – М., 1992.
134. Гостехкомиссия России. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники. – М., 1992.
135. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. – М., 1992.
136. Information Technology Security Evaluation Criteria. Harmonized Criteria of France-Germany-Netherlands-United Kingdom. – Department of Trade and Industry, London, 1991.
137. Federal Criteria for Information Technology Security // National institute of Standards and Technology & National Security Agency. Version 1.0, December 1992.
138. Canadian Trusted Computer Product Evaluation Criteria // Canadian System Security Centre Communication Security Establishment, Government of Canada. Version 3.0e. January 1993.
139. Common Criteria for Information Technology Security Evaluation // National Institute of Standards and Technology & National Security Agency (USA), Communication Security Establishment (Canada), Communication-Electronics Security Group (United Kingdom), Bundesamt fur Sichereit in

der Informationstechnik (Germany), Service Central de la Securite des Systemes d`Information (France), National Communications Security Agency (Netherlands). Version 2.1, August 1999.

140. Harrison M., Ruzzo W., Uhlman J. Protection operating systems // Communications of the ACM, 1976.
141. Ravi S. Sandhu The Typed Acess Matrix Model // Proceedings of IEEE Symposium on Security and Privacy. – Oakland, California, May 4-6, 1992. – P. 122-136.
142. Harrison M., Ruzzo W. Monotonic protection systems // Foundation of secure computation, 1978.
143. Leonard J. LaPadula and D. Elliot Bell. Secure Computer Systems: A Mathematical Model // MITRE Corporation Technical Report, 2547, V. II, 31 May 1973.
144. Ciaran Bryce Lattice-Based Enforcement of Access Control Policies // Arbeitspapiere der GMD (Research Report), N. 1020, August 1996.
145. John McLean The Specification and Modeling of Computer Security // Computer, 23(1):9-16, January 1990.
146. John McLean Security Models // Encyclopedia of software engineering, 1994.
147. Ивашов Е.Н., Панов А.В. и др. Устройство для получения нанодорожек. Патент РФ на ПМ № 42696.- Опубл. 10.12.2004, Бюл. №34.
148. Ивашов Е.Н., Львов Б.Г., Степанчиков С.В. Способ получения нанотрубок. Патент РФ на изобретение №2225655. Опубл. 10.03.2004., Бюл. №7.
149. Степин В.С., Горохов В.Т., Розоб М.А. Философия науки и техники. – М.: Контакт Альфа, 1995. – 384 с.
150. Митчем К. Что такое философия техники / Перевод с английского под ред. В. Г. Горохова – М.: Аспект Пресс, 1995 – 149 с.
151. Кудрин Б.И. Научно-технический прогресс и формирование техноценозов // ЭКО, 1980, №8. – С. 15-28.
152. Кудрин Б.И. Отбор: энергетический, естественный, информационный,

документальный. Общность и специфика // Электрификация металлургических предприятий Сибири. Вып. 5. – Томск: Изд. ТГУ, 1981. – С. 111-187.

153. Нанотехнологии в электронике / под ред. Ю.А. Чаплыгина / – М.: Техносфера, 2005. – 447 с.
154. Васин В.А., Ивашов Е.Н., Степанчиков С.В. Нанотехнологические процессы и оборудование электронной техники. – М.: МИЭМ, 2009 – 264 с.
155. Васин В.А., Ивашов Е.Н., Степанчиков С.В. Идеология проектирования автоматизированного оборудования современных вакуумных технологий // Автоматизация и современные технологии, 2008, № 8. – С. 3-8.
156. Васин В.А., Ивашов Е.Н., Степанчиков С.В. Тенденции проектирования внутрикамерных вакуумных систем в электронном машиностроении // Вестник машиностроения, 2008, №9. – С. 60-62.
157. Ивашов Е.Н., Васин В.А., Степанчиков С.В. // Механические системы электронного машиностроения на основе многокоординатных исполнительных устройств // Вестник машиностроения, 2009, №7. – С. 3-10.
158. Васин В.А. Насосы и термосорбционные компрессоры на основе сплавов накопителей водорода для вакуумного технологического оборудования // Конструкции из композиционных материалов, 2008," №3. – С. 52-57.
159. Александрова А.Т., Васин В.А., Горюнов А.А. Новые принципы прецизионного дозирования вакуумных потоков / Научно технический семинар «Контроль герметичности-98», Санкт-Петербург // Тезисы докладов. – СПб.: ОАО «Завод Измеритель», 1998. – С. 7.

ОГЛАВЛЕНИЕ