

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. БАУМАНА
КАЛУЖСКИЙ ФИЛИАЛ**

На правах рукописи

Лавренков Юрий Николаевич

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА КОМБИНИРОВАННЫХ
НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ДЛЯ ПОВЫШЕНИЯ
ЭФФЕКТИВНОСТИ БЕЗОПАСНОЙ МАРШРУТИЗАЦИИ
ИНФОРМАЦИИ В СЕТЯХ СВЯЗИ**

Специальность 05.13.17 – Теоретические основы информатики

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель: д.т.н., профессор
Комарцова Л.Г.

Калуга - 2014

Оглавление

Введение	3
1 Анализ методов и алгоритмов интеллектуального управления потоками информации	9
1.1 Современные технологии для обеспечения безопасной маршрутизации информации	9
1.2 Интеллектуальные информационные технологии для решения трудно формализуемых задач	17
1.2.1 Искусственные нейронные сети.....	17
1.2.2 Инициализация начальных состояний нейросетевых экспертов.....	21
1.2.3 Алгоритмы обучения нейронных сетей	24
1.2.4 Ансамбли нейронных сетей.....	29
1.3 Выводы по главе 1	41
2 Разработка алгоритмов настройки и обучения комитета нейросетевых экспертов.....	42
2.1 Структурная схема разрабатываемой системы.....	42
2.2 Проектирование алгоритмов обучения экспертов	44
2.3 Определение начальных состояний экспертов	53
2.4 Проектирование нечёткой системы оценки качества обучения нейросетевых экспертов.....	68
2.5 Разработка алгоритма предобработки сигналов от нейросетевых экспертов.....	88
2.6 Разработка алгоритма обеспечения безопасности системы	102
2.7 Проектирование алгоритма решения задачи безопасной маршрутизации	117
2.8 Выводы по главе 2	130
3 Практическая реализация разработанных моделей и алгоритмов.....	131
3.1 Требования к программной эмуляции нейроимитаторов	131
3.2 Основные компоненты архитектуры программной системы.....	132
3.3 Особенности реализации инструментальной среды	138
3.4 Перспективы развития программной системы	151
3.5 Выводы по главе 3	152
4 Исследование разработанных алгоритмов и оценка их эффективности при решении практических задач.....	154
4.1 Анализ алгоритмов обучения экспертов	154
4.2 Анализ алгоритмов предобработки сигналов	161
4.3 Результат работы алгоритмов генерации случайных чисел.....	173
4.4 Анализ результатов выполнения маршрутизации на основе оценки качества канала связи	181
4.5 Выводы по главе 4	191
Заключение.....	192
Список литературы.....	194
Приложение 1	206
Приложение 2	207

Введение

Актуальность темы исследования. Развитие сложных вычислительных систем и комплексов основано на транспортировке информации от отправителя к получателю. Проблему выбора оптимального пути, при котором информация проходит по маршруту, соответствующему определённым критериям, решают алгоритмы маршрутизации. Маршрутизация – процесс передвижения информации от источника к пункту назначения через объединенную сеть. При этом, как правило, на пути встречается, по крайней мере, один узел. Маршрутизация включает в себя два основных компонента: определение оптимальных путей маршрутизации и транспортировка информационных сообщений. Определение маршрута представляет собой сложный процесс и базируется на различных показателях или комбинациях показателей. Если процесс маршрутизации происходит в динамическом режиме, т.е. путь, по которому передаётся информация, рассчитывается не на начальном этапе передачи сообщения, а по мере продвижения его по сети, то сложность расчёта маршрута возрастает. В практических задачах возникают ситуации, когда необходимо выполнить передачу информации в открытом виде, т.е. без использования средств шифрования. Такой способ передачи требует построения маршрутов продвижения информации по каналам связи, обладающих определённой степенью надёжности и защищённости от вмешательства злоумышленников, в частности от прямого физического подключения к среде передачи. При маршрутизации на основе заданных условий необходимо выполнять оценку не только характеристик, обеспечивающих быструю доставку информации получателю, но в процессе поиска оптимального маршрута учитывать параметры безопасности среды передачи.

Существующие алгоритмы маршрутизации требуют наличия информации о полной структуре сети, в которой будет организовываться передача данных. Если происходит частое изменение топологии сети, появление и удаление новых соединений, изменения в среде передачи, то маршрутизирующие алгоритмы теряют способность поддерживать оперативный информационный обмен в сети.

Для поддержания способности телекоммуникационной сети выполнять доставку информационных сообщений необходимо применять современные методы, способные решать задачи при неполных или противоречивых входных данных. К таким методам относятся вычислительные методы на базе нейросетевых, эволюционных и нечётких алгоритмов. Для успешного решения задачи безопасной маршрутизации требуется развитие и комбинирование вычислительных структур на основе приведённых интеллектуальных подходов.

Применение интеллектуальных технологий позволит выполнять передачу информации в распределённых сетях даже в случаях их частичной деградации или нарушения целостности из-за действия третьих лиц. Таким образом, задача исследования и проектирования алгоритмов безопасной нейросетевой маршрутизации является актуальной и практически значимой.

Целью работы является повышение эффективности функционирования комбинированных нейросетевых методов для решения задачи маршрутизации информации в сетях связи. Для достижения указанной цели необходимо решить следующие **задачи**:

1. Провести анализ алгоритмов построения оптимальных маршрутов в компьютерных сетях. Провести классификацию алгоритмов, а также проанализировать способы получения данных, необходимых для работы маршрутизирующей системы.
2. Выполнить построение модели нейросетевого маршрутизатора, представленного комитетом, состоящим из трёх нейросетевых экспертов: сети прямого распространения, рекуррентной нейронной сети, радиально-базисной нейронной сети.
3. Разработать способ объединения решений, полученных от различных нейросетевых структур (НС). Проанализировать существующие методы объединения результатов работы различных алгоритмов. Выполнить разработку алгоритма оценки компетентности каждого эксперта.

4. Спроектировать методы обучения каждого типа нейросетевого эксперта, базирующиеся на стандартных градиентных алгоритмах оптимизации, но с применением комбинированных эвристических процедур.

5. Проанализировать алгоритмы инициализации начального состояния нейронных сетей перед выполнением процедуры обучения. Разработать алгоритм настройки параметров НС с учётом того, что все сети функционируют в составе комитета, представляющего собой единую вычислительную структуру – ассоциативную машину.

6. Выполнить разработку алгоритма, обеспечивающего безопасное функционирование нейросетевого комитета при выполнении функции маршрутизации информации.

7. Спроектировать набор программно-аппаратных средств для получения оперативной информации о состоянии физических линий передачи информации в телекоммуникационной сети. На основе полученных данных выполнить построение обучающей и тестовой выборки для ассоциативной машины.

Объект и предмет исследования. Объектом исследования является процесс объединения мнений отдельных нейросетевых экспертов.

Предметом исследования является применение и разработка комплекса нейросетевых алгоритмов для получения комбинированных решений от различных вычислительных структур.

Методы исследования основаны на теории принятия решений, нейроинформатике, теории оптимизации, генетических и эволюционных алгоритмах, методах математической статистики.

Достоверность результатов подтверждена данными экспериментов и компьютерного моделирования, сравнением полученных результатов с данными, полученными другими авторами.

Основные результаты, выносимые на защиту:

1. Общие принципы построения систем маршрутизации информации.
2. Методы обучения комитета нейросетевых экспертов на основе обратного распространения ошибки, модифицированные алгоритмом поиска с переменным

шагом, градиентный метод настройки параметров радиально-базисной нейронной сети, модифицированный эвристической процедурой упреждающего поиска, метод наискорейшего спуска с эвристикой на основе алгоритма комплексов.

3. Процедура определения начального состояния ассоциативной машины, выраженного предварительной установкой параметров весовых коэффициентов, основанная на применении кооперативного иммунного алгоритма с эволюционным алгоритмом генерации искусственных антител.

4. Модель объединения мнений нейросетевых экспертов на основе нечёткой системы оценки компетентности каждого эксперта и системы из трёх модифицированных нейронов, которая выполняет предобработку сигналов перед подачей на финальную сигма-пи нейронную сеть.

5. Модель нейросетевого генератора случайных чисел, базирующаяся на множестве взаимодействующих сетей Хопфилда, для обеспечения безопасного функционирования нейросетевого маршрутизатора.

6. Алгоритм оценки состояния канала связи, и принципы формирования обучающей выборки для разработанной нейросетевой структуры, основанные на оценке изменения параметров ёмкости и сопротивления канала передачи информации, а также методах спектрального анализа отражённого из канала связи зондирующего сигнала.

Научная новизна работы заключается в следующем:

1. Разработана архитектура ассоциативной машины на основе трёх нейронных сетей с различной архитектурой, которая способна объединять выходные сигналы нейронных сетей для решения трудно формализуемых задач. Спроектированная система включает в себя комплекс нейронных элементов для предобработки сигналов от нейросетевых экспертов на основе информации, полученной от нечёткой системы, применяемой для оценки возможностей каждой нейронной сети получить решение конкретной вычислительной задачи.

2. Модифицированы алгоритмы обучения нейронных сетей, входящих в ассоциативную машину, комбинированными эвристическими процедурами и

стратегией инициализации начального состояния комитета экспертов с помощью кооперативного иммунного алгоритма оптимизации.

3. Спроектирован механизм получения финального решения на основе нейронной сети типа сигма-пи, обучение которой выполнялось на основе разработанного комплекса методов случайного поиска, применяемого для оптимизации переменных параметров разработанной нейросетевой архитектуры. Для управления процедурой случайного поиска разработан нейросетевой генератор случайных чисел с источником энтропии, который позволяет выполнять оптимизацию параметров нейросетевых экспертов из случайного начального состояния.

4. Спроектирована методика оценки безопасности канала передачи информации на основе анализа изменения параметров среды передачи и спектра отражённого зондирующего сигнала.

Теоретическая значимость работы заключается в развитии технологий для объединения результатов работы различных интеллектуальных вычислительных алгоритмов, применяемых для повышения эффективности функционирования алгоритмов безопасной маршрутизации в условиях неполной или противоречивой информации.

Практическая ценность работы заключается в том, что разработанные методы и модели нейросетевой безопасной маршрутизации позволяют:

1. Повысить эффективность безопасной передачи информации в телекоммуникационных сетях, сократить издержки на применение алгоритмов шифрования данных.

2. Повысить вероятность доставки информационных сообщений в условиях деградации сети связи, за счёт применения аппроксимирующих способностей комплекса нейронных сетей.

3. Выполнить построение оптимального маршрута передачи данных в условиях неполной информации о состоянии каналов связи.

Апробация результатов работы. Основные положения работы были доложены на семинарах кафедры ЭИУ2-КФ КФ МГТУ им. Н.Э. Баумана в 2008-

2014 гг.; на «Всероссийской научно-технической конференции студентов, аспирантов и молодых учёных – Наукоёмкие технологии в приборо - и машиностроении и развитии инновационной деятельности в ВУЗЕ», г. Калуга, 2008, 2009, 2010, 2011, 2012, 2013 гг.; «Региональной научно-технической конференции студентов, аспирантов и молодых учёных – Наукоёмкие технологии в приборо - и машиностроении и развитии инновационной деятельности в ВУЗЕ», г. Калуга, 2008, 2009, 2010, 2011, 2012, 2013, 2014 гг.; «Восьмой межрегиональной научно – технической конференции студентов и аспирантов - Применение кибернетических методов в решении проблем общества 21 века», г. Обнинск, 2010 г.; «Международном симпозиуме «Интеллектуальные системы», г. Москва, 2010, 2012 гг.; на «Всероссийской научно-технической конференции «Нейроинформатика», г. Москва, 2010, 2011, 2012, 2013, 2014 гг.; «Всероссийской научной конференции «Нейрокомпьютеры и их применение», г. Москва, 2012 г.; «Международной научно-технической конференции OSTIS», г. Минск, 2012, 2014 гг.; «Международной научно-технической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте», г. Коломна, 2013 г.; «1-м международном симпозиуме под ред. проф. А.В. Колесникова», г. Калининград, 2012 г.; «Международной конференции «Интеллектуальный анализ информации», г. Киев, 2012 г.; «Всероссийской научно-практической конференции с международным участием «Информационные технологии в профессиональной деятельности и научной работе», г. Йошкар-Ола, 2012 г.

Публикации по теме работы. Результаты, полученные при выполнении диссертационной работы, опубликованы в 46 печатных работах, из них 6 – в периодических изданиях, рекомендованных ВАК, получено свидетельство о государственной регистрации программы для ЭВМ.

Структура и объём работы. Диссертация состоит из введения, четырёх глав, заключения, списка литературы (136 наименований) и 2 приложений. Объём работы составляет 208 страниц, содержит 111 рисунков и 12 таблиц.

1 Анализ методов и алгоритмов интеллектуального управления потоками информации

1.1 Современные технологии для обеспечения безопасной маршрутизации информации

Основная задача различного рода телекоммуникационных сетей – транспортировка информации от отправителя к получателю [78]. В большинстве случаев приходится совершать несколько пересылок между узлами для достижения конечного пункта в передаче данных. Как правило, путей передачи информации в сети может быть несколько (рисунок 1.1) [80]. На рисунке видно, что от узла *A* к узлу *B* через сеть может проходить множество различных маршрутов.

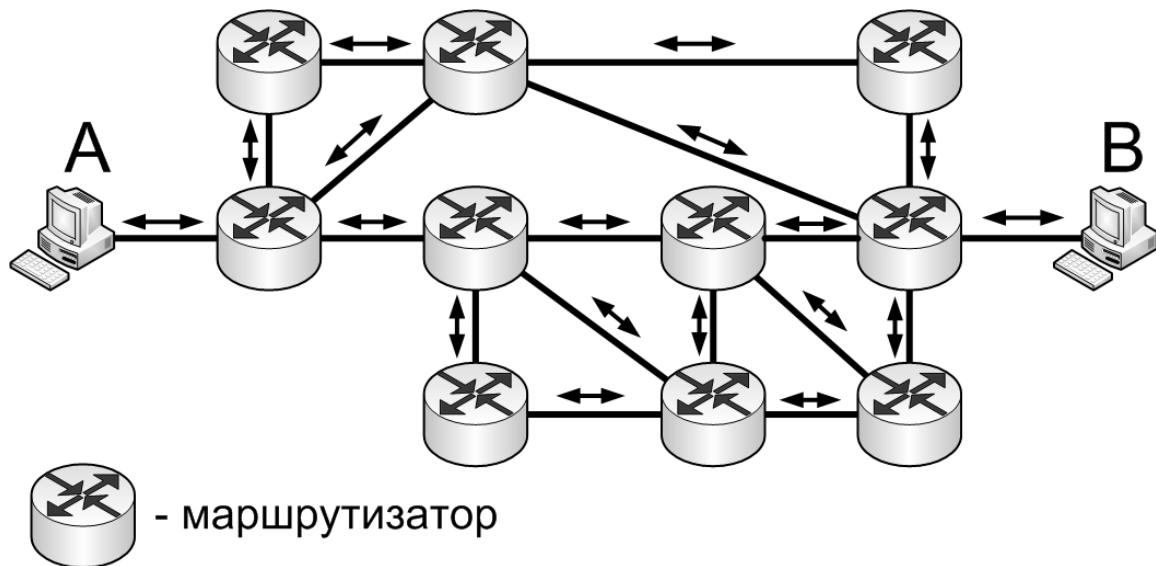


Рисунок 1.1 – Проблема выбора оптимального маршрута

Для выбора оптимального пути, соответствующего заданным критериям, применяют алгоритмы маршрутизации. Маршрутизация включает в себя два параллельных процесса [103]:

- Создание таблицы маршрутизации – информационной структуры, в которой содержится информация о следующем пункте передачи информации для оптимальной доставки данных узлу назначения.
- Управление потоками передачи информации с помощью полученной таблицы.

Программные или аппаратные средства, выполняющие построение таблицы маршрутизации, получили название маршрутизаторов. Маршрутизаторы осуществляют выбор оптимального пути на основе различных критериев – метрик. Метрики характеризуют предпочтительность выбора маршрута [104]. Выбор применяемых показателей качества зависит от специфики работы сети. Для быстрой доставки информации необходимо применять критерии, характеризующие расстояние, преодолеваемое посылкой данных при движении к узлу назначения. Алгоритмы маршрутизации учитывают пропускную способность каналов связи, задержку передачи, стоимость передачи информации и т.д. В данной работе особое внимание уделяется безопасности передачи информации в процессе выполнения её доставки к заданному узлу, т.е. доставка информации в открытом виде по тем каналам передачи, которые гарантируют невозможность её перехвата или искажения третьими лицами. На первое место выходят критерии, характеризующие физическую безопасность канала связи. Автором предлагаются способы анализа целостности каналов передачи информации для формирования критериев, применяемых при прокладке маршрутов. Чем больше метрик учитываются маршрутизатором, тем эффективнее производится генерация маршрутов, отвечающих заданным требованиям.

В работе рассматривается разработка нейросетевого алгоритма распределённой адаптивной маршрутизации. Под адаптивной или динамической маршрутизацией подразумевается система, которая все изменения конфигурации сети автоматически отражает в таблицах маршрутизации [111]. Необходимость в создании алгоритмов такого типа возникает из-за наличия уязвимостей в существующих алгоритмах динамической маршрутизации. В настоящее время в большинстве телекоммуникационных сетей применяют адаптивные распределенные алгоритмы маршрутизации: дистанционно-векторные алгоритмы и алгоритмы состояния связей. При распределённом подходе все маршрутизаторы в сети находятся в одинаковых условиях, они обнаруживают маршруты, строят таблицы маршрутизации, взаимодействуют друг с другом [106]. Такой подход обладает большим преимуществом перед централизованным подходом, когда в

сети присутствует только один маршрутизатор, который собирает информацию о сети от других маршрутизаторов и конфигурирует маршруты продвижения данных. В случае отказа центрального маршрутизатора вся сеть выйдет из строя, поэтому такой подход не получил широкого распространения при проектировании сетей.

При использовании маршрутизаторами дистанционно-векторных алгоритмов (ДВА) производится регулярный обмен копиями таблиц маршрутизации [109]. При совершении регулярных обновлений маршрутизаторы сообщают друг другу об изменении топологии сети. Таким образом, каждый маршрутизатор получает через соседние маршрутизаторы информацию обо всех имеющихся узлах в сети. Поиск оптимального маршрута базируется на использовании вектора расстояния, который показывает необходимое количество переходов для достижения заданного пункта. В каждой из позиции таблицы маршрутизации есть суммарный вектор, который показывает, на каком расстоянии находится соответствующая сеть или узел (рисунок 1.2).

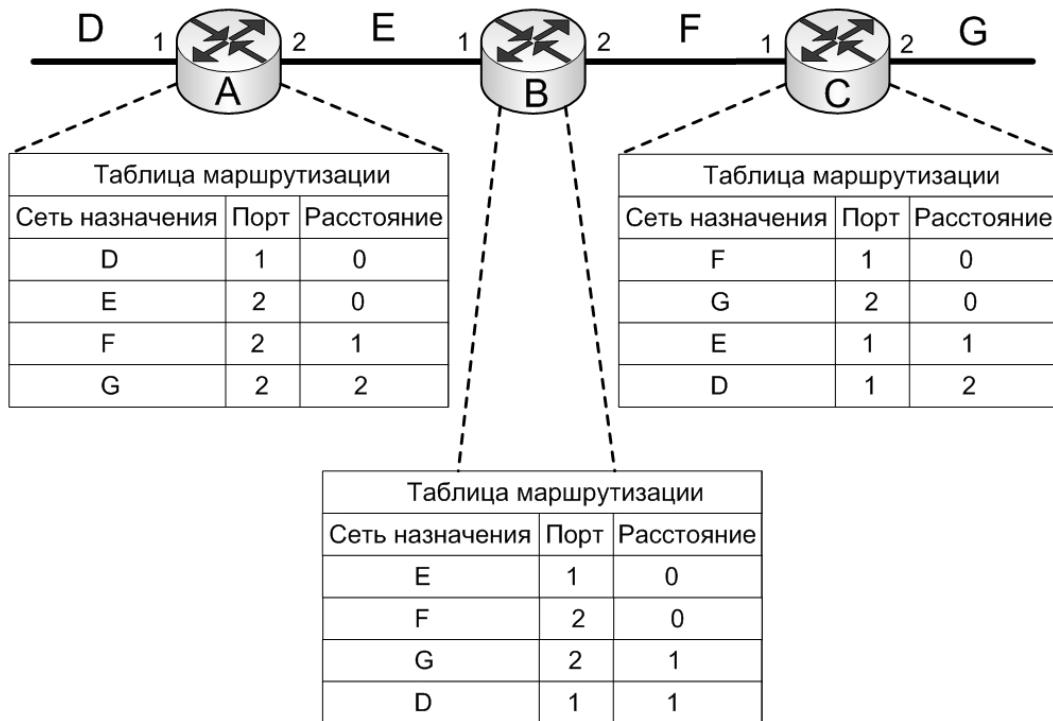


Рисунок 1.2 – Формирование таблиц маршрутизации ДВА

На основе ДВА базируется дистанционно-векторный протокол маршрутизации группового вещания DVMRP (Distance Vector Multicast Routing

Protocol) [78]. Данный протокол был одним из первых протоколов для определения пути продвижения группового трафика, то есть доставки данных нескольким получателям из одного источника. Данный тип передачи информации играет важную роль для рассылки команд управления или данных большому количеству узлов. При выборе маршрутов рассылки также актуальна проблема выбора безопасного маршрута с надёжной доставкой информации. Развитие протоколов широковещательной рассылки началось с алгоритма пересылки информации маршрутизатором на все интерфейсы, кроме входного. Такая стратегия приводит к генерации большого количества бесполезного трафика в сети [82]. Модификации алгоритма позволяют выполнить распространение трафика от источника к получателю так, чтобы пакеты продвигались только по тем путям, которые оптимальным образом соединяли источник с каждым получателем. На рисунке 1.3 исключены маршруты группового трафика от источника к тем получателям, для которых он не предназначен.

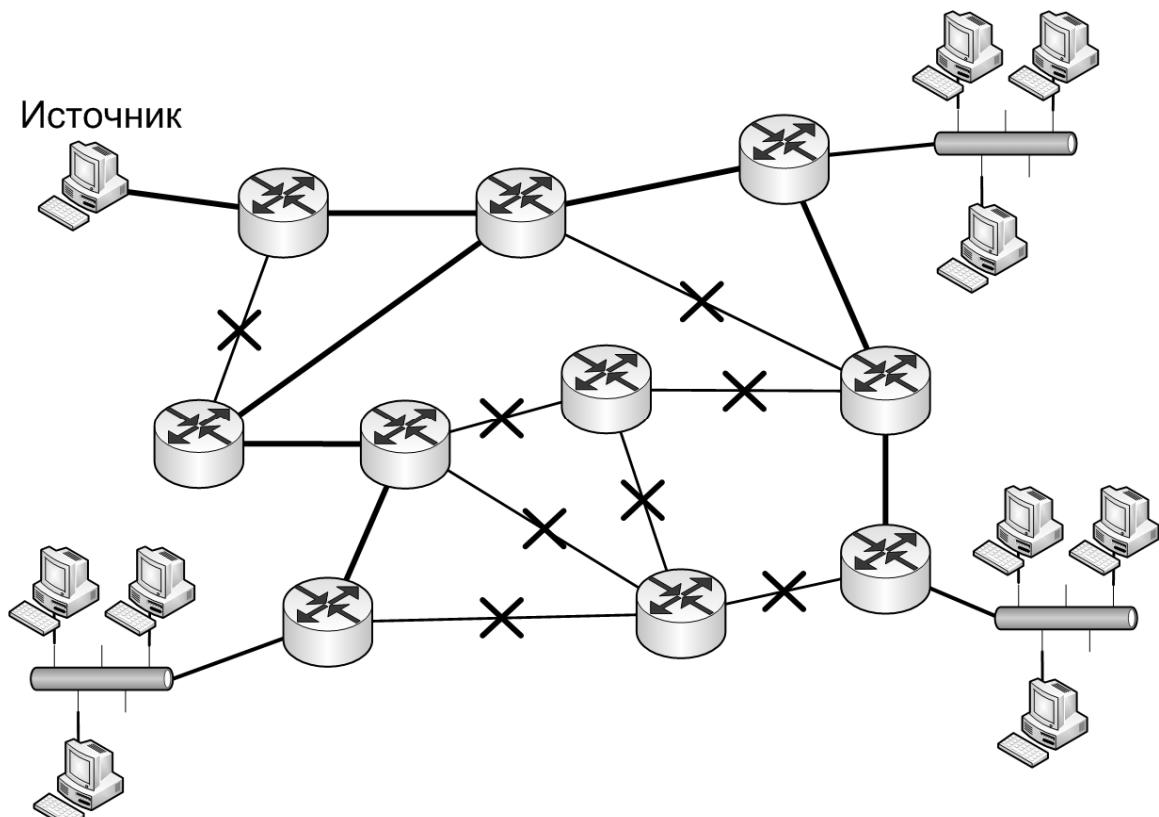


Рисунок 1.3 – Управление групповой передачей информации

В результате необходимо выполнить построение дерева с вершиной в источнике передаваемой информации. Структура дерева соединяет все

маршрутизаторы, к которым непосредственно подключены локальные сети, содержащие получателей данной группы, наилучшими путями [7]. Применение нейросетевых технологий при модификации данного типа алгоритмов позволит выполнить построение маршрута продвижения групповой информации по наиболее безопасному пути с учётом изменений в структуре сети.

К недостаткам приведённого алгоритма маршрутизации можно отнести следующие:

1. Стабильная работа алгоритма обеспечивается в небольших сетях. В больших сетях производится интенсивный обмен таблицами маршрутизации, что приводит к дополнительной нагрузке линий связи.
2. Изменение конфигурации сети не всегда может быть корректно обработано данными типами алгоритмов маршрутизации, т.к. маршрутизаторы не обладают сведениями о точной топологии сети.
3. Протоколы, использующие в своей работе ДВА маршрутизации (например, RIP – Routing Information Protocol), трудно адаптируются к потере маршрута, т.к. они передают информацию, необходимую для пополнения таблиц маршрутизации.
4. Возможно зацикливание информации – постоянное передвижение информационных групп от одного маршрутизатора к другому.

Существует вероятность появления ложных маршрутов, которые возникают при использовании информации о несуществующих маршрутах [19].

Алгоритмы состояния связей (ACC) обеспечивают каждый маршрутизатор информацией, которой достаточно для построения точного графа сети [77]. Маршрутизаторы, работающие на основе ACC, поддерживают сложную базу, содержащую информацию о топологии соединений в сети. Функционирование всех маршрутизаторов основано на одном графике сети, что делает процесс маршрутизации устойчивым к изменениям конфигурации. ДВА маршрутизации не содержат информации об удалённых сетях и маршрутизаторах.

Процедура построения таблицы маршрутизатора производится в два этапа [103]:

1. Построение базы данных о состоянии связей в сети. Топология сети представляется в виде графа, в котором вершинами графа являются маршрутизаторы, а ребрами – связи между ними. Маршрутизаторы обмениваются с соседними устройствами информацией о графе сети, которой каждый владеет на текущий момент. Маршрутизаторы не производят модификацию информации при передаче. В результате все маршрутизаторы сети получают идентичные сведения о структуре сети. Вся информация упорядочивается в логическую топологию, представленную деревом связей. Корень дерева – текущий маршрутизатор, а ветви – возможные маршруты ко всем подсетям. Если состояние связей изменилось, то процесс построения графа повторяется.

2. Нахождение оптимальных маршрутов и генерация таблицы маршрутизации. После построения дерева связей, необходимо решить трудоёмкую задачу нахождения оптимального пути на графике. Протоколы маршрутизации, основанные на АСС (например, OSPF – Open Shortest Path First [76, 106]), для поиска оптимального маршрута используют итеративный алгоритм Дейкстры. Каждый маршрутизатор выполняет поиск оптимальных путей от своих интерфейсов до всех известных подсетей. В каждом найденном пути запоминается только первый шаг. Именно он заносится в таблицу маршрутизации.

На рисунке 1.4 приведён пример сети, состоящей из шести маршрутизаторов (R_1-R_6) и подсетей (N_1-N_6). На основе оценки состояния каналов, каждому приписывается мера стоимости. На рисунке 1.5 показано дерево кратчайших путей для маршрутизатора R_3 . Таким образом, каждый маршрутизатор может отслеживать альтернативные пути и производить выбор лучшего для каждой конечной точки.

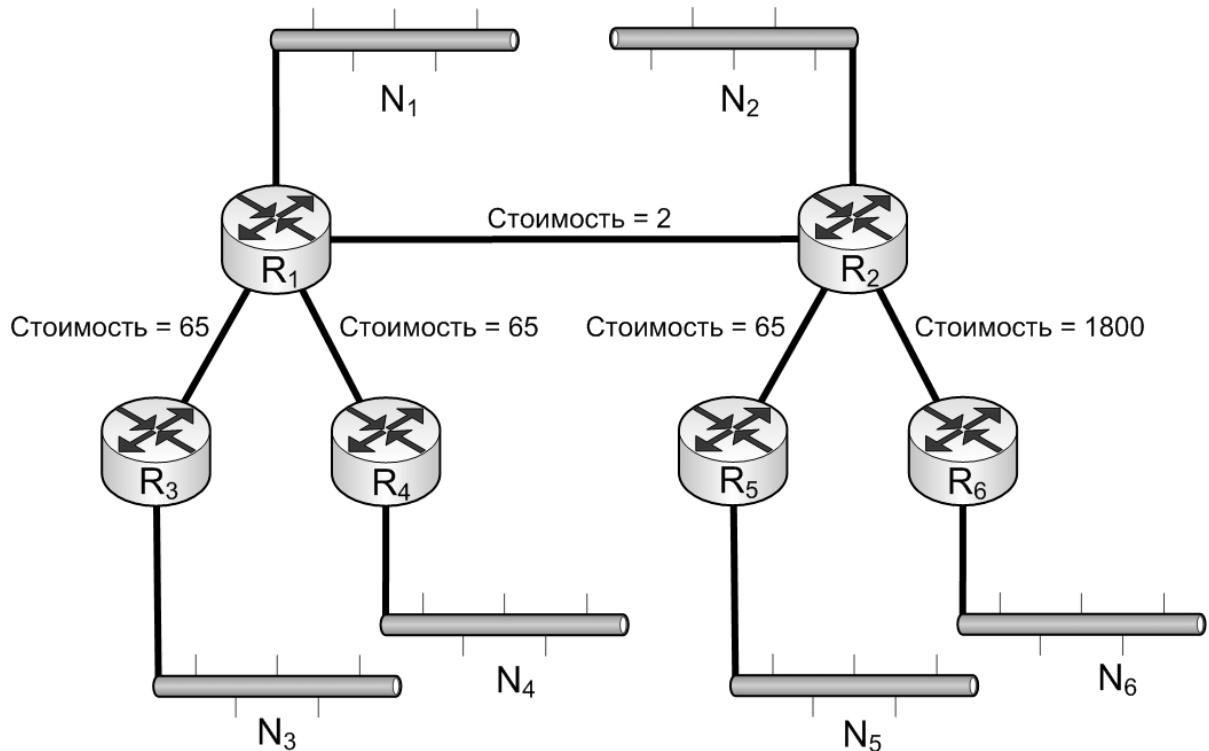


Рисунок 1.4 – Пример организации сети и оценка каналов

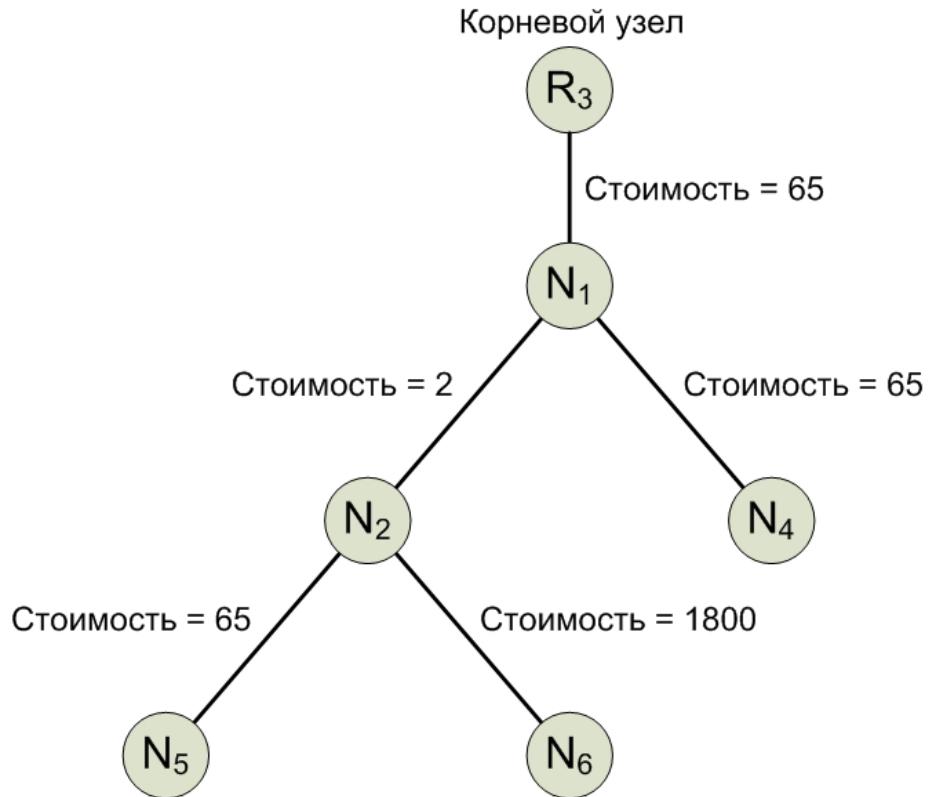


Рисунок 1.5 – Дерево кратчайших путей для маршрутизатора R_3

На основе ACC функционирует и протокол MOSF (Multicast extensions to OSPF), предназначенный для группового вещания [89]. Маршрутизаторы добавляют к информации о состоянии каналов передачи информацию о

членстве в группах различных узлов сети. В результате маршрутизаторы формируют не только общий граф сети, но получают информацию о составе групп для рассылки сообщений. На основании этой информации маршрутизирующие алгоритмы находят дерево кратчайших путей для каждой группы. Это позволяет доставлять информационные сообщения по кратчайшим путям от источника до подсетей, в которых находятся члены группы, которым предназначена информация.

При использовании алгоритма маршрутизации, организованного на основе оценки состояний каналов передачи, возникают следующие проблемы:

1. Алгоритм требует большого количества памяти и вычислительных ресурсов для обработки большого потока информации, поддержки логического дерева и таблицы маршрутизации.
2. Динамическое определение структуры сети, приводит к генерации большого объёма трафика, необходимого для обмена информацией между маршрутизаторами.
3. Если сетевая структура обладает сложной топологией с множеством связей, то частое изменение характеристик каналов связи, появление новых соединений, отключение некоторых каналов связи приводит к полному отказу алгоритма маршрутизации.

Одной из основных проблем, обнаруженной при анализе недостатков рассмотренных алгоритмов, является выполнения процесса маршрутизации при неполной информации [82]. Информация от отдельных элементов телекоммуникационной сети сначала пересыпается маршрутизаторам, но в таблицах маршрутизации может содержаться неполная маршрутная информация. Нарушение функций маршрутизирующей системы может произойти в результате отказа отдельных каналов связи или самих маршрутизаторов. В случае если маршрутизаторы не обладают информацией, необходимой для выбора оптимального пути к узлу назначения, они пересыпают данные по стандартным, заранее определённым маршрутам [111]. Маршрутизирующая система может отправить данные за пределы локальной автономной системы [103], что может

привести к возврату этих данных, и произойдет «зацикливание» трафика. Отказ каналов связи, обеспечивающих функционирование статических маршрутов резервной передачи информации, приведёт к полной остановке работы сети. Для решения выявленных проблем в работе предлагается использовать комитет нейросетевых экспертов – ассоциативную машину [110]. Аппроксимирующие способности нейронных сетей могут быть использованы для поддержания работы сети в моменты нестабильной работы стандартных протоколов маршрутизации. Основное внимание будет уделено метрикам канала связи, характеризующим его безопасность, с целью поддержания безопасной передачи потоков цифровой информации.

1.2 Интеллектуальные информационные технологии для решения трудно формализуемых задач

1.2.1 Искусственные нейронные сети

В качестве основного механизма принятия решений в трудно формализуемых задачах (в контексте данной работы – задача безопасной маршрутизации цифровой информации в сетях связи) в работе предлагается использовать комитет нейронных сетей (ассоциативную машину). Основу данного комитета будут составлять нейросетевые эксперты трёх различных архитектур: многослойный персепtron, радиально-базисная нейронная сеть, рекуррентная сеть Эльмана. Применение нейронных сетей обусловлено следующими причинами [9, 15, 110]:

1. Для каждой из рассматриваемых НС существуют алгоритмы обучения и настройки их внутренней структуры, которые способны эффективно производить оптимизацию компонентов сети для решения прикладных задач.
2. Системы, основу управляющих модулей которых составляют нейросетевые технологии, являются адаптивными структурами, для которых нет необходимости производить точный расчёт параметров перед началом работы.
3. Этап обучения НС позволяет выполнить адаптацию параметров для настройки на конкретные условия сигнала и шума.

4. Структура нейросетевых экспертов может быть адаптирована к конкретной вычислительной задаче.

Рассмотрим особенности архитектуры компонентов ассоциативной машины – экспертов, представленных различными типами нейронных сетей (нейросетевых экспертов).

Первый компонент – многослойный персептрон, в котором каждый элемент вычисляет взвешенную сумму своих входов с поправкой в виде порога активации нейрона и затем преобразует эту величину с помощью передаточной функции, формируя выходное значение этого элемента [18].

Элементы организованы в послойную топологию с прямой передачей сигнала (рисунок 1.6). На рисунке введены следующие обозначения: $w_{ij}^{[a]}$ – значение весового коэффициента, соединяющего i -й и j -й нейроны, a – номер слоя, u – взвешенная сумма входных сигналов, $\psi(\cdot)$ – функция активации нейрона, o – значение выходного сигнала. Такую сеть можно интерпретировать как модель вход-выход, в которой веса и пороговые значения являются настраиваемыми параметрами модели [36]. Сеть может моделировать функцию практически любой степени сложности [54, 73], причем число слоев и число элементов в каждом слое определяют сложность функции.

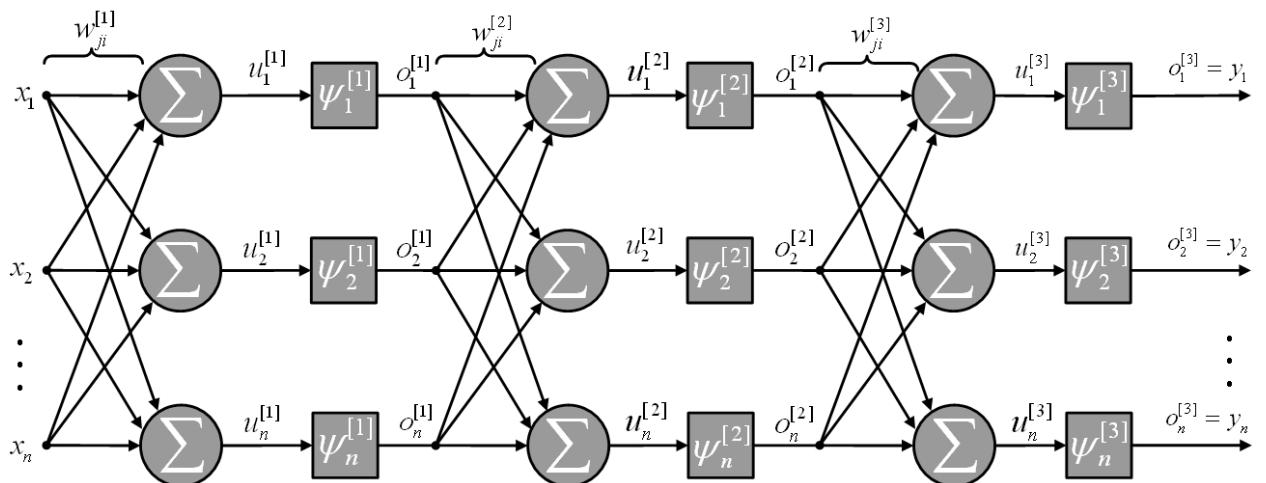


Рисунок 1.6 – Трёхслойный персептрон

В качестве второго нейросетевого эксперта применяется радиально-базисная нейронная сеть. Функционирование такой сети базируется на ином принципе формирования выходного сигнала. В нервных системах организмов

присутствуют нейроны, выходной сигнал которых «настроен» на ограниченную область входного пространства [28]. Сеть, построенная на искусственных нейронах, обладающих выраженным локальными характеристиками (нейроны реализуют функции, радиально изменяющиеся вокруг выбранного центра и принимающие ненулевые значения только в окрестности этого центра [9]), является альтернативой многослойным персепtronам и получила название радиально-базисной нейронной сети (РБФ) [24]. Данный тип нейронной сети, также как и многослойный персептрон, относится к классу универсальных аппроксиматоров [110].

На рисунке 1.7 показана схема радиально-базисной сети с n - входами и m - выходами, выполняющая нелинейное преобразование вида:

$$y_j = w_{j0} + \sum_{i=1}^h w_{ji} \cdot \varphi_i(x), \quad (1.1)$$

где $\varphi_i(x)$ – радиально-базисные функции, определяющие процесс преобразования из n - мерного пространства входов в m - мерное пространство выходов, h – количество нейронов в скрытом слое, w_{ij} – весовой коэффициент, характеризующий силу синаптической связи между нейронами i и j .

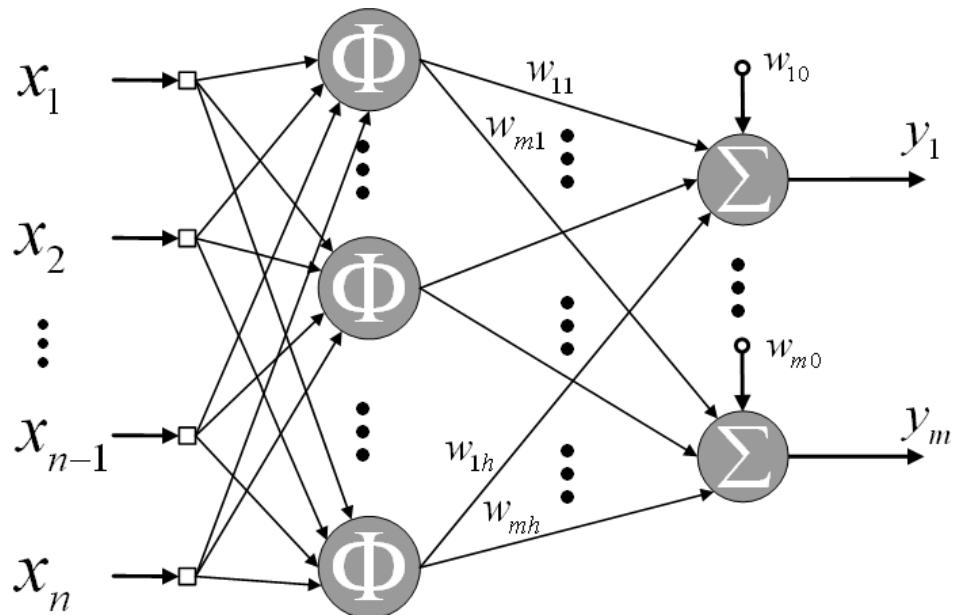


Рисунок 1.7 – Радиально-базисная нейронная сеть

Свойства такой сети полностью определяются радиально-базисными функциями [53], которые являются основными компонентами в нейронах скрытого слоя и формирующими базис для входных векторов x . Радиально-базисная функция:

$$\phi(x) = \Phi(\|x - c\|, \sigma) = \Phi(r, \sigma) \quad (1.2)$$

- это многомерная функция, зависящая от расстояния $r = \|x - c\|$ между входным вектором x и собственным центром c и параметра ширины σ . Наибольшее распространение получили гауссовские функции, принимающие максимальное значение в центре c и монотонно убывающие по мере удаления от центра.

Рассмотренные сети с прямой передачей информации характеризуются распространением и обработкой сигналов в одном направлении - от входа к выходу ИНС. Другую группу нейронных сетей представляют рекуррентные нейронные сети, имеющие замкнутые петли обратной связи между слоями [81]. Входные сигналы в ИНС такого типа должны быть заданы в форме временной последовательности.

Рассмотрим архитектуру многослойной рекуррентной нейронной сети Эльмана [81], которая нашла применение для выявления границ между отдельными словами в потоке фонем. Структурный график сети приведён на рисунке 1.8. Кроме скрытого и выходного слоя в сети присутствует слой обратной связи, называемый контекстным или слоем состояний. Выходные сигналы нейронов данного слоя поступают на входной слой через элементы задержки. Обрабатываемая информация сохраняется в течение одного такта работы. «Строительными блоками» сети Эльмана являются стандартные нейроны с активационными функциями типа гиперболического тангенса, аддитивные линейные ассоциаторы и элементы задержки.

На основе приведённых нейросетевых моделей выполняется построение комитета нейронных сетей. Входящие в состав ассоциативной машины нейронные сети обладают тремя различными архитектурами, что приводит к

различным механизмам поиска решений – это позволяет повысить вероятность получения оптимального решения.

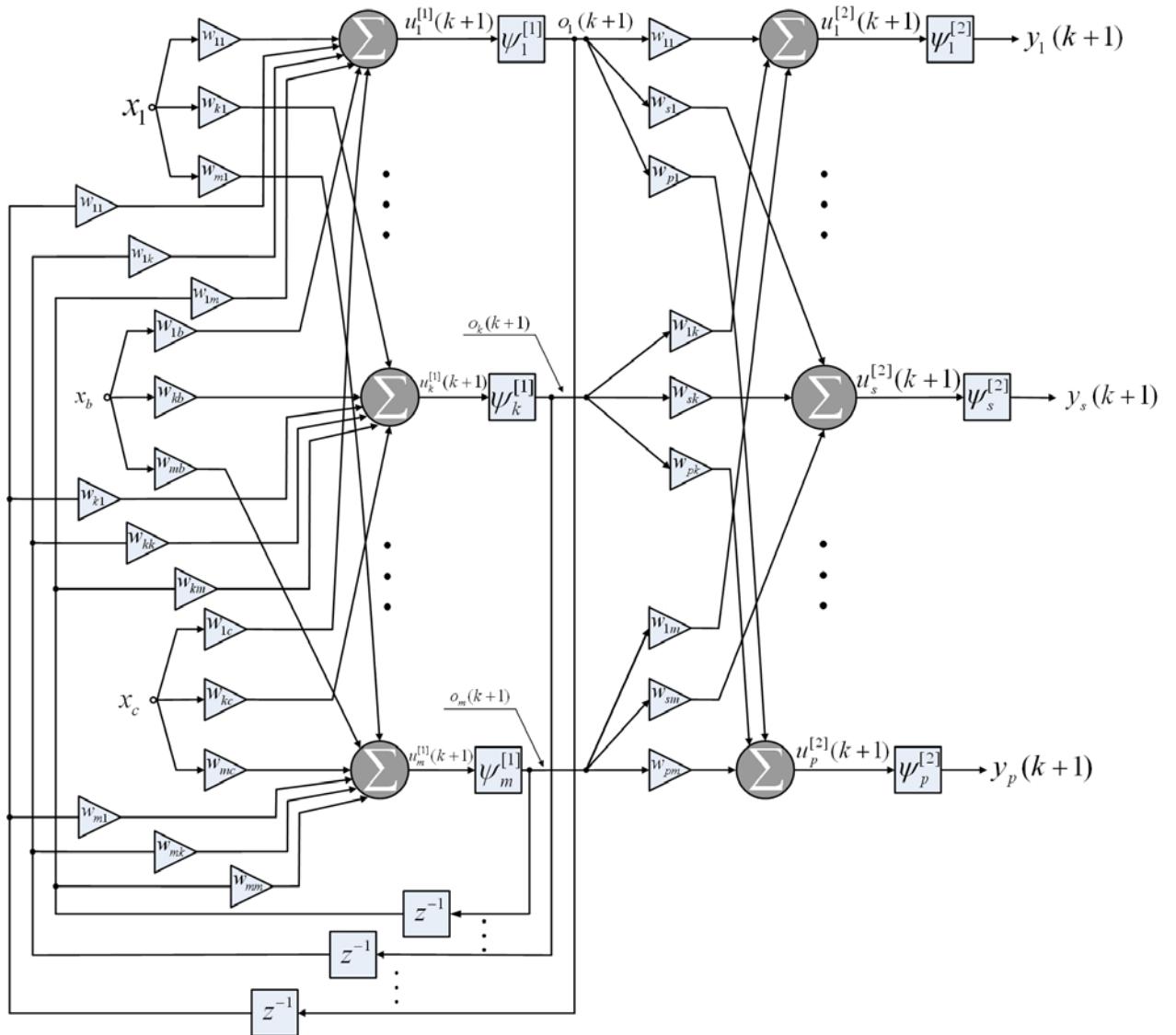


Рисунок 1.8 – Нейронная сеть Эльмана

Для успешного применения комитетов в задачах распознавания образов, аппроксимации функций, решения задач оптимизации необходимо выполнить настройку архитектуры нейронной сети, определить параметры функций активации нейронов, начальные состояния, способ объединения решений от различных НС [71]. Рассмотрим возможные способы решения данных проблем.

1.2.2 Инициализация начальных состояний нейросетевых экспертов

Начальные состояния нейросетевых экспертов – это состояние весовых коэффициентов синаптических связей и параметров функций активации до начала процедуры обучения НС [12]. Данные параметры оказывают влияние на весь

процесс обучения. В качестве оптимальных начальных значений весовых коэффициентов считаются параметры, установленные таким образом, что сеть выдаёт ожидаемую реакцию на элементы обучающей выборки. В результате происходит ускорение процесса обучения и устранение задержек в локальных минимумах [20]. В практических задачах такое состояние определить практически невозможно, и для инициализации параметров НС перед процедурой обучения применяют различные методы [29].

Процедура первоначальной инициализации параметров весовых коэффициентов играет важную роль для дальнейшего обучения отдельных нейросетевых экспертов. При неправильной установке начальных значений становиться возможным введение некоторых сигмоидальных функций активации нейронов в режим насыщения, что приводит к исключению данных нейронов из процесса функционирования всей нейронной сети. В результате происходит ослабление аппроксимирующих возможностей отдельных компонентов ассоциативной машины, приводящее к её деградации и неспособности решать поставленные проблемы.

Одним из основных способов инициализации весов считается метод случайной установки параметров весовых коэффициентов. В результате такой установки стараются получить начальное состояние каждого нейрона, которое лежит на значительном расстоянии от точки его насыщения. На основании многочисленных экспериментов, различными исследователями предлагаются разнообразные интервалы, в которых могут находиться значения данных весовых коэффициентов [43], в большинстве случаев допустимым является интервал $(0, \dots, 1)$. Нулевое значение не разрешается присваивать первоначальным значениям весовых коэффициентов, т.к. не удастся произвести оптимизацию градиентными методами из-за равенства нулю производных. В этом случае обучение необходимо начинать не с градиентного подхода – использовать методы оптимизации нулевого порядка [6]: метод случайного поиска, метод покоординатного спуска, метод многогранника и т.д.

У метода случайной настройки параметров НС перед процессом обучения существует множество модификаций [9], в которых начальные значения весовых коэффициентов могут быть сгенерированы на основе различного рода распределений [22]. У всех методов данного класса выделяется существенный недостаток: не производится анализ особенностей обучающей выборки. Процесс обучения может развиваться неограниченно долгий промежуток времени.

Альтернативой методу случайной настройки параметров НС перед обучением является метод главных компонент. Суть процесса заключается в последовательном выделении главных компонент с помощью одного из методов [41]: алгоритм Ойя, фильтр Хебба, алгоритм АРЕХ и т.д. и пропускании их через слои нейронов. Выходной слой инициализируется на основе линейной регрессии выходов выборки по выходам последнего слоя НС. Положительной стороной этого алгоритма является возможность параллельной организации вычислений начальных значений. Недостатком алгоритма являются ограничения на применение алгоритмов вычисления главных компонент, что накладывает ограничения на структуру нейронной сети. Для устранения этого недостатка требуется применение процедур изменения структуры сети [38].

В данной работе предлагается использовать метод инициализации начальных значений многослойной нейронной сети с учётом специфики применения каждой НС. Все нейронные сети функционируют в составе структуры, которая принимает финальное решение на основе результатов функционирования каждого нейросетевого эксперта. Т.к. ассоциативная машина обладает сложной внутренней топологией, то необходимо применять метод, комбинирующий в себе настройку начальных параметров сетей как на основе предварительных знаний об элементах обучающей выборки, так и с учётом возможностей способа случайной инициализации весов. Для решения данной проблемы предлагается метод инициализации, основанный на применении кооперативного иммунного подхода. Данный способ учитывает структуру всей решающей системы и позволяет сформировать алгоритм настройки параметров

НС перед обучением таким образом, чтобы при дальнейшем обучении и функционировании вся совокупность НС действовала максимально эффективно.

1.2.3 Алгоритмы обучения нейронных сетей

Определение числа промежуточных слоев и числа элементов в них является важным вопросом при конструировании нейронных сетей [53]. Количество входных и выходных элементов определяется условиями задачи. После того, как определено число слоев и число элементов (в зависимости от предметной области) в каждом из них, нужно найти значения для весов и порогов сети, которые бы минимизировали ошибку прогноза, выдаваемого сетью. Для этого служат алгоритмы обучения. С применением собранных статистических данных веса и пороговые значения автоматически корректируются с целью минимизировать эту ошибку. Этот процесс представляет собой адаптацию модели, которая реализуется сетью, к имеющимся обучающим данным [15, 17]. Ошибка для конкретной конфигурации сети определяется путем пропускания через сеть всех имеющихся наблюдений и сравнения реально выдаваемых выходных значений с желаемыми (целевыми) значениями. Для применения комплекса нейронных сетей в системах безопасной маршрутизации информации необходим алгоритм оптимизации параметров нейронных сетей, который быстро сходится за приемлемое время и формирует нейросетевого эксперта таким образом, чтобы минимизировать вероятность ошибочных решений. Для этой цели может быть использовано множество различных алгоритмов обучения, базирующихся на принципе минимизации суммарной ошибки распознавания [121].

Наиболее известным и чаще всего применяемым для многослойных нейронных сетей алгоритмом обучения является метод обратного распространения ошибки [55]. В данной работе для обучения эксперта на основе многослойного персептрона используется именно этот метод.

При обучении многослойного персептрона методом обратного распространения ошибки необходимо задать множество векторов входных и желаемых выходных сигналов [58]: $\{X_1, X_2, \dots, X_p\}$ и $\{D_1, D_2, \dots, D_p\}$

соответственно, где p - количество элементов в обучающем множестве. При функционировании сети вектор X_q поступает на входные рецепторы, на выходе получаем вектор Y_q , где $q = 1, 2, \dots, p$ – номер предъявляемого образа. Выходной вектор сети Y_q будет отличаться от соответствующего вектора D_q . Ошибкой сети можно считать отклонение полученного выходного вектора от желаемых значений: $E = D_q - Y_q$ для каждой пары значений вход-выход. Выполняя настройку весовых коэффициентов синаптических связей, производится минимизация ошибки, с которой сеть воспроизводит исходную зависимость из обучающей выборки. Данный метод реализует итеративную процедуру обучения с двумя проходами по сети: прямым и обратным, при которых соответственно вычисляются функциональные преобразования вектора входного сигнала и корректируются параметры весовых коэффициентов.

РБФ сети подобно многослойным ИНС являются универсальными аппроксиматорами, но, так как в них присутствует только один нелинейный скрытый слой, а настраиваются параметры линейного выходного слоя, для их обучения могут быть использованы стандартные процедуры обучения [83].

Процесс обучения сети RBF с учетом выбранного типа радиальной базисной функции сводится [2, 15]:

- к подбору центров и параметров формы базисных функций;
- к подбору весов нейронов выходного слоя.

Радиально-базисные нейронные сети обучаются с использованием алгоритма, минимизирующего функционал ошибки:

$$E = \sum_{i=1}^p \left[\sum_{k=1}^m w_k \cdot f(\|x - c_k\|) - d_i \right]^2, \quad (1.3)$$

где p – число обучающих выборок, $d = (d_1, d_2, \dots, d_p)$ – вектор ожидаемых значений.

Оптимальная настройка центров радиально-базисных функций может быть проведена следующими способами [20]:

1. Выбор фиксированных значений центров, путём отбора определённых экземпляров входных данных или на основе некоторого вероятностного закона.

2. Заданное множество точек данных разбивается на кластеры с помощью алгоритма кластеризации [36]. Центры кластеров принимаются за центры радиально-базисных функций.

3. Центры кластеров выступают в качестве параметров оптимизации наравне с другими параметрами радиально-базисных функций.

Анализ литературы [43, 68, 73] показал, что одним из часто применяемых методов, используемых для настройки центров радиально-базисных функций, является применение алгоритма кластеризации k – средних. Согласно данному алгоритму центры нейронов скрытого слоя РБФ сети располагаются в областях пространства с максимальной концентрацией информативных данных.

Для выбора параметров ширины радиально-базисных функций применяют следующие варианты [92]:

1. Параметр фиксирован и выбирается заранее.
2. Ширина радиально-базисной функции определяется на основе информации о полученных кластерах.
3. Параметр ширины рассматривается как настраиваемый параметр и подвергается оптимизации.

После начальной настройки параметров радиально-базисных функций производится их подстройка, и настройка весовых коэффициентов с помощью градиентного алгоритма обучения, который применяется для настройки весов, центров и ширин радиально-базисных функций нейронов [97]:

$$w_k^t = w_k^{t-1} - \eta \cdot \frac{\partial E}{\partial w_k^{t-1}}, c_k^t = c_k^{t-1} - \beta \cdot \frac{\partial E}{\partial c_k^{t-1}}, \sigma_k^t = \sigma_k^{t-1} - \alpha \cdot \frac{\partial E}{\partial \sigma_k^{t-1}}, \quad (1.4)$$

где η, α, β - коэффициенты обучения, w, c, σ - настраиваемые веса, центры и параметры ширины радиально-базисных функций.

Основная проблема, возникающая при синтезе рекуррентных сетей, связана с обеспечением их устойчивости, от которой зависит возможность решения поставленной задачи [121].

Кроме скрытого и выходного слоя в сети присутствует слой обратной связи, называемый контекстным или слоем состояний. Выходные сигналы нейронов данного слоя поступают на входной слой через элементы задержки. Обрабатываемая информация сохраняется в течение одного такта работы. В качестве универсального алгоритма для настройки нейронной сети Эльмана на решение большого класса задач наиболее часто применяется метод наискорейшего спуска, основанный на работах Вильямса и Зипсера [81]. Для возможности применять данный метод необходимо рассчитывать градиент целевой функции в текущий момент. Целевая функция – сумма квадратов разностей между значениями полученных выходных сигналов и ожидаемыми значениями:

$$E = \frac{1}{2} \sum_{i=1}^p [y_i - d_i]^2 = \frac{1}{2} \sum_{i=1}^p e_i^2, \quad (1.5)$$

где y_i – выходной сигнал i -го нейрона, d_i – желаемый выходной сигнал i -го нейрона.

Рассмотренные архитектуры нейронных сетей чаще всего обучаются с применением градиентных методов. У методов данного типа есть определённые преимущества [90]:

1. Для реализации алгоритма обучения используется информация об ограниченном количестве связей текущего нейрона с остальными элементами сети. Данная особенность позволяет успешно распараллелить процесс обучения.
2. Градиентные алгоритмы характеризуются высокой степенью схожести между собой. Метод обучения легко воспроизвести для произвольного числа слоёв, размерности входов и выходов, функций активации. Кроме того, градиентные алгоритмы (например, алгоритм обратного распространения ошибки) можно применять совместно с различными градиентными методами оптимизации: методом наискорейшего спуска, сопряженных направлений, методом Ньютона и др.

3. Алгоритмы обратного распространения ошибки обладают эффективным механизмом вычисления градиентов, который требует меньшего числа операций по сравнению с прямым вычислением градиента.

Кроме приведённых достоинств, градиентные алгоритмы обладают рядом недостатков [73, 81, 110] (таблица 1.1):

1. Возможность блокировки сети. Алгоритм не эффективен в случае, когда производные по различным весам сильно отличаются.
2. Возможна медленная сходимость процесса обучения.
3. Переобучение. Высокая точность, которая достигается на обучающей выборке, может привести к неприемлемым результатам на тестовой выборке.
4. «Попадание» в локальный минимум.

В случае, когда множество нейронных сетей применяются в составе ассоциативной машины для коллективного принятия решений, необходимо провести процедуру их обучения так, чтобы по возможности устранить или свести к минимуму приведённые недостатки. Для достижения поставленной задачи применяются различного рода комбинированные эвристические процедуры, с помощью которых можно осуществить небольшие случайные изменения весовых коэффициентов нейронной сети [98]. В данной работе осуществляется разработка трёх эвристических алгоритмов на основе комбинированных методов оптимизации и реализуется способ объединения двух методов настройки НС. Комбинация градиентных алгоритмов и эвристических процедур позволяет воспользоваться преимуществами градиентных методов оптимизации, что гарантирует получение решения, а применение эвристик повышает вероятность нахождения решения, близкого к глобальному оптимуму.

Таблица 1.1 Анализ алгоритмов обучения НС различных типов

Вид нейросетевого эксперта	Базовый алгоритм обучения	Задачи, которые необходимо решить эвристическим алгоритмом
Многослойный персепtron	Алгоритм обратного распространения ошибки	1. Выход из локальных минимумов. 2. Расширение пространства поиска оптимальной конфигурации сети. 3. Возможность блокировки отдельных синаптических связей или всей сети.
Радиально-базисная нейронная сеть	Алгоритм кластеризации для настройки параметров радиально-базисных нейронов + градиентный алгоритм оптимизации.	1. Уточнение параметров радиально-базисных элементов в сети. 2. Уточнение параметров весовых коэффициентов скрытого слоя. 3. Настройка полей влияния РБФ функций.
Рекуррентная нейронная сеть Эльмана	Метод наискорейшего спуска.	1. Увеличение параметров весовых коэффициентов скрытого слоя и слоя состояний. 2. Определение пороговых значений нейросетевых элементов сети. 3. Повышение устойчивости сети.

1.2.4 Ансамбли нейронных сетей

При решении сложных задач может возникнуть ситуация, когда попытки получить приемлемое решение или необходимое качество аппроксимирующей зависимости, даже при использовании различных алгоритмов, параллельно обрабатывающих и решающих одну и ту же задачу, не дают результатов [100]. В этом случае объединение нескольких алгоритмов в композицию позволяет решить поставленную задачу. При решении задач с помощью нейросетевых методов, построенных на применении нескольких нейронных сетей – ансамблей, входные данные обрабатываются с помощью нескольких нейросетевых методов. При решении сложных задач классификации, регрессии, прогнозирования часто оказывается, что ни один из алгоритмов не обеспечивает желаемого качества восстановления зависимости. В таких случаях необходимо строить композиции алгоритмов, в которых ошибки отдельных алгоритмов взаимно компенсируются [90].

Вычислительные задачи, содержащие сложные внутренние зависимости, могут быть решены при помощи их разбиения на множество небольших и

простых задач с последующим объединением полученных решений. При обучении с учителем вычислительная простота достигается за счет распределения задачи обучения среди множества экспертов, которые, в свою очередь, разбивают входное пространство на множество подпространств. Комбинацию таких экспертов, в случае, когда в качестве экспертов применяются искусственные нейронные сети, называют комитетом нейронных сетей (ассоциативной машиной) [43]. Данный механизм производит интеграцию знаний, накопленных экспертами, в общее решение, которое имеет приоритет над каждым решением отдельного эксперта. Считается, что решения экспертов, полученные на основе обработки данных, связанных с решаемой задачей, позволяют получить решение необходимого качества.

Если решение получается от множества экспертов, то в процессе его формирования необходимо ответить на ряд вопросов [53, 59, 67]:

1. Выводы экспертов могут противоречить друг другу. Как скомбинировать несколько экспертных оценок отдельных экспертов для получения верного решения?

2. Как оценить величину доверия нейросетевому эксперту? Нейронные сети могут быть обучены с помощью различных алгоритмов: базирующихся на принципах эволюции – иммунный и генетический алгоритмы, алгоритм обратного распространения ошибки, стохастические алгоритмы оптимизации [105, 115]. Кроме процедуры обучения необходимо сформировать оптимальную структуру сети: определить количество нейронов и топологию связей между ними, выбрать функцию активации, определиться с необходимостью обратных связей [121]. Если нейросетевые эксперты будут настроены неверно (например, наличие избыточных весов приводит к снижению качества обобщения, т.к. в процессе функционирования нейронной сети они могут принимать произвольные значения, а также приводят к необходимости большего числа итераций алгоритмов обучения), то будет получено решение, базирующееся на мнении некомпетентного эксперта.

Комитеты экспертов можно разбить на две категории [110]:

1. Статические структуры. В этом классе ассоциативных машин результаты функционирования отдельных экспертов объединяются, но при этом не учитывается входной сигнал. В данной категории можно выделить следующие методы получения решения:

- **Усреднение по ансамблю.** Выходной сигнал вычисляется как линейная комбинация выходов отдельных алгоритмов, принимающих решения.
- **Усиление.** При котором слабый алгоритм обучения превращается в алгоритм, достигающий произвольной заданной точности.
- **Стэкинг.** Способ синтеза составных моделей экспертов с использованием множества уровней нейросетевых экспертов.
- **Бэггинг.** Формирование множества экспертов на основе совокупности подмножеств обучающей выборки и последующего объединения мнений экспертов.

2. Динамические структуры. Если ассоциативная машина строится по данному принципу, то при получении входного сигнала учитываются значения входного сигнала в механизме объединения выходных сигналов экспертов. Различают два подхода при реализации динамических структур [62, 73].

- **Смешение мнений экспертов** – результаты функционирования отдельных экспертов нелинейно объединяются в единую шлюзовую сеть.
- **Иерархическое объединение мнений экспертов** – решения, полученные от отдельных экспертов, нелинейно объединяются с помощью нескольких шлюзовых сетей, организованных в иерархическую структуру.

Рассмотрим особенности реализации приведённых способов получения решений от множества экспертов.

Метод усреднения по ансамблю [110] предполагает применение некоторого множества отдельно обучаемых нейронных сетей. Выходные сигналы экспертов комбинируются определённым образом для получения решения. Применение алгоритма обусловлено следующими причинами [12]:

1. Каждая сеть в объединяемом множестве сетей обладает более простой архитектурой, чем нейронная сеть способная заменить ансамбль.

2. У сетей со сложной внутренней архитектурой возрастает риск переобучения, т.к. размер обучающей выборки может быть не достаточным для корректного завершения алгоритма обучения.

Объединение выходного сигнала осуществляется с помощью линейной комбинации выходных сигналов отдельных нейросетевых экспертов. Простейшим примером объединения мнений различных экспертов является простое голосование [66]:

$$y(x) = F(y_1(x), y_2(x), \dots, y_T(x)) = \frac{1}{T} \sum_{g=1}^T y_g(x), \quad (1.6)$$

где $x = (x_1, x_2, \dots, x_k)$ – входной вектор, k – число рецепторов нейронной сети, $y(x) = (y_1, y_2, \dots, y_n)$ – вектор значений выходного сигнала от комитета нейронных сетей, полученных как отклик на воздействие входного вектора, n – количество выходных сигналов эксперта, F – корректирующая операция, вычисляющая результирующее решение, T – количество нейросетевых экспертов в ансамбле. Если в качестве примера применения подобного типа голосования рассмотреть задачу классификации, то классифицируемый объект будет отнесён к тому классу, к которому его относит большинство нейросетевых экспертов.

Рассмотрим достоинства метода усреднения по ансамблю:

1. Простота реализации.
2. Вычислительная сложность метода значительно ниже алгоритмов обучения отдельных нейросетевых экспертов.

К недостаткам метода можно отнести следующие:

1. Зависимость конечного результата комитета экспертов от правильного определения коэффициентов компетентности отдельных экспертов. В случае если коэффициенты распределены не верно, будет получен некорректный результат.
2. В практических задачах обучающая выборка содержит шумовые выбросы, которые приводят к увеличению вероятности возникновения ошибок в процессе функционирования отдельных экспертов, т.к. попытка обучающего

алгоритма настроится на «шум» приводит к уменьшению аппроксимирующих способностей сети. Поэтому для устранения этого недостатка необходимо использовать дополнительные алгоритмы анализа данных для снижения уровня помех: случайный и стратификационный сэмплинг, обработка дубликатов и противоречий, обнаружение аномальных значений, корректировка пропусков данных. Данные модификации значительно увеличивают вычислительную сложность алгоритма [92].

Метод усиления является вторым способом реализации класса "статических" ассоциативных машин [120]. В комитетах нейронных сетей, основанных на усреднении по ансамблю, все эксперты обучаются на одном и том же множестве данных. Сети отличаются друг от друга только выбором исходного состояния: начальным значением весовых коэффициентов, порогов, коэффициентов скорости обучения и т.д. В противоположность этому, сети-эксперты, работающие на основе метода усиления, обучаются на примерах, принадлежащих совершенно различным распределениям. Данный метод построения ассоциативных машин может быть применён для увеличения производительности любого алгоритма [58].

По сравнению с предыдущим алгоритмом метод усиления представляет собой более сложную процедуру. В методе усреднения по ансамблю модели могут быть построены параллельно и независимо друг от друга. В методе усиления каждая новая модель эксперта синтезируется на основе результатов ранее построенных, в результате модели создаются последовательно.

Метод усиления может быть реализован различными способами [67, 83]:

1. Усиление за счет фильтрации.
2. Усиление путём перевзвешивания.

При использовании усиления за счет фильтрации ассоциативная машина состоит из K экспертов. Алгоритм, используемый для такого обучения, называют алгоритмом усиления. При этом эксперты произвольно маркируются как G_1, G_2, \dots, G_K . Они обучаются по отдельности по следующей стратегии [3]:

1. Первый эксперт обучается на множестве, состоящем из N_1 примеров.

2. Обученный первый эксперт используется для фильтрации второго множества примеров.

3. После обучения второго эксперта множество примеров обучения для следующего эксперта формируется по нижеприведённому алгоритму:

- Новый пример подаётся на входы первого и второго экспертов. Если эксперты выдают одинаковый выходной сигнал, то пример отклоняется. Если выводы нейросетевых экспертов различны, то данный образец выборки используется для обучения следующего эксперта.
- Алгоритм продолжает работу, пока не будет отфильтровано все множество из N_1 примеров. Полученное множество примеров затем используется для обучения третьего эксперта.

После окончания обучения последнего нейросетевого эксперта на отфильтрованном множестве примеров процесс обучения всего комитета считается завершенным.

Второй способ организации метода усиления базируется на использовании подвыборок [116]. Вместо извлечения выборок из исходного множества данных применяется взвешивание примеров. Весовой коэффициент каждого примера устанавливается в соответствии с его влиянием на процесс обучения эксперта. На каждой итерации работы алгоритма вес изменяется таким образом, чтобы характеризовать эффективность данного нейросетевого эксперта. Общая схема формирования множества экспертов показана на рисунке 1.9.

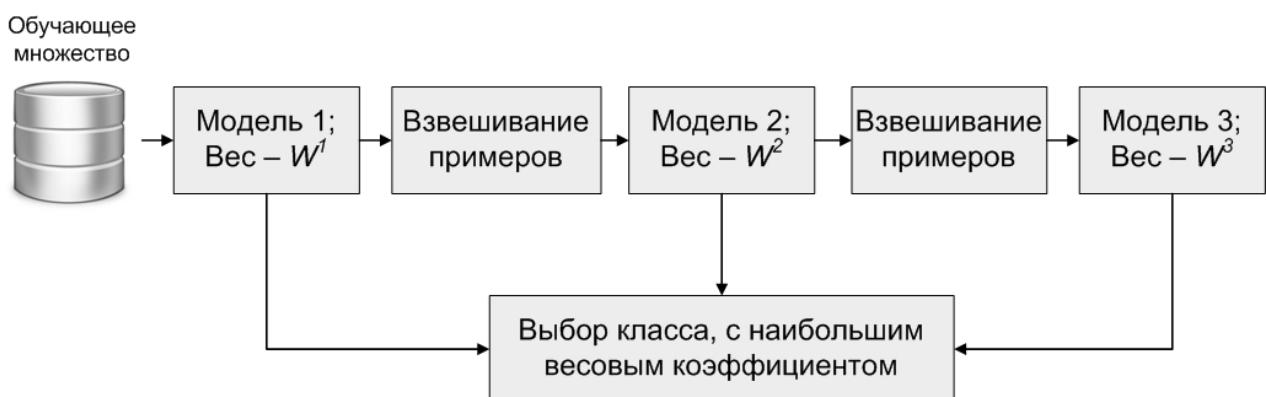


Рисунок 1.9 – Схема формирования множества экспертов

Результаты экспериментов [83] показали, что если ошибка отдельного эксперта ε^t всегда меньше 0,5, то значение ошибки итогового эксперта стремится к 0 по экспоненте при увеличении числа итераций. Таким образом, данный алгоритм усиления позволяет усилить обобщающую способность финального эксперта G^f , который получается точнее, чем отдельные эксперты и меньше подвержен влиянию шумовых составляющих данных.

К основным недостаткам метода усиления можно отнести следующие:

1. Для успешной реализации алгоритма требуется большое количество примеров обучающей выборки.
2. Построение системы экспертов сводится к жадному добавлению новых базовых нейронных сетей, что может приводить к построению неэффективных нейросетевых структур на поддержание и обучение которых потребуются значительные вычислительные ресурсы.
3. В процессе функционирования алгоритма усиления происходит исключение различных частей первоначальной выборки из процесса обучения, что приводит к вырождению обучающей выборки, на которой строятся последние модели.

Стэкинг обычно не используется в тех случаях, когда базовые алгоритмы принятия решений базируются на одном принципе [83], поэтому в качестве нейросетевых экспертов необходимо применять различные нейронные сети: многослойный персептрон, сеть Хопфилда, сигма-пи сеть, спайковые нейронные сети и т.д. Когда имеется несколько алгоритмов, принимающих решения, обычная процедура заключается в определении компетентности каждого алгоритма и выборе лучшего из них, который и будет использоваться для принятия решений. В отличие от данного подхода в стэкинге используется концепция метаобучения, которая является альтернативой процедуре голосования [90]. Структурная схема алгоритма стэкинга показана на рисунке 1.10.

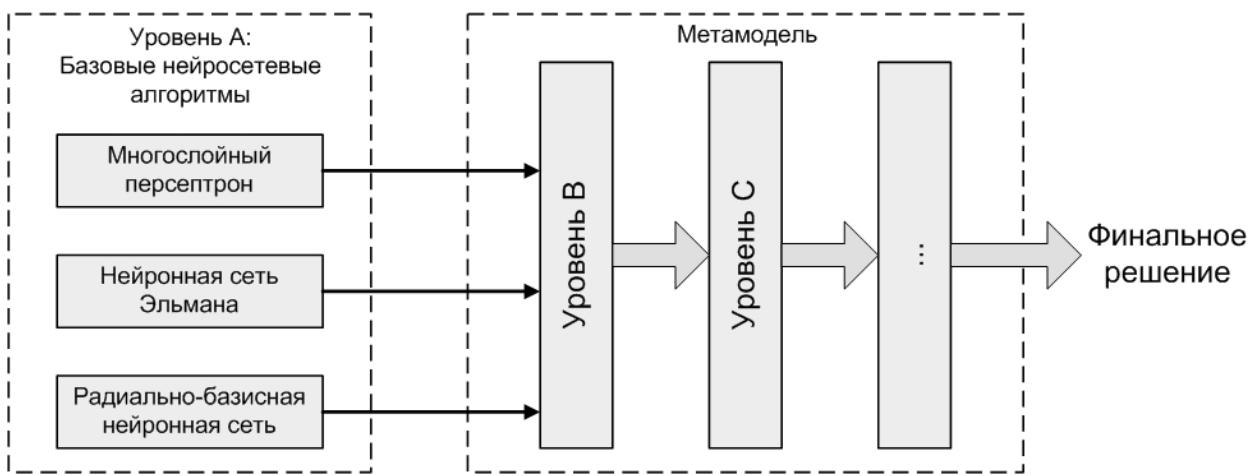


Рисунок 1.10 – Структурная схема стэкинга

Все базовые эксперты (в данном контексте работы представлены нейронными сетями различной конфигурации и архитектуры) образуют уровень A. На вход метамодели (уровень B) подаются результаты с выходов базовых моделей. Процесс продолжается до тех пор, пока не будет выполнено условие прекращения работы алгоритма: достижение заданной точности, использование всех возможных вычислительных ресурсов и т.д.

К основным недостаткам модели стэкинга можно отнести следующие:

1. Сложность теоретического анализа из-за множества последовательно формируемых моделей.
2. Возможно разрастание уровней метамодели, что может привести к быстрому сокращению вычислительных ресурсов.

Следующий способ объединения мнений экспертов относится к стохастическим методам [100]. Базовые алгоритмы настраиваются независимо друг от друга на случайно выбранных подмножествах обучающей выборки. Формирование подвыборок производится случайно, поэтому некоторые примеры могут присутствовать по нескольку раз, а другие ни разу. Затем на основе каждого подмножества обучающей выборки производится построение нейросетевого эксперта. Результаты функционирования полученных алгоритмов объединяются в композицию с помощью простого или взвешенного голосования. В основе работы бэггинга – технология «возмущение и комбинирование», основанная на том, что даже небольшие изменения в обучающем множестве способны сформировать несколько абсолютно разных нейросетевых моделей. В

данном случае неустойчивость моделей применяется для создания ансамблей. Под «возмущением» понимается процесс внесения случайных изменений в компоненты обучающей выборки с целью дальнейшего синтеза нескольких альтернативных моделей, основанных на изменённых данных.

Схема бэггинга представлена на рисунке 1.11.

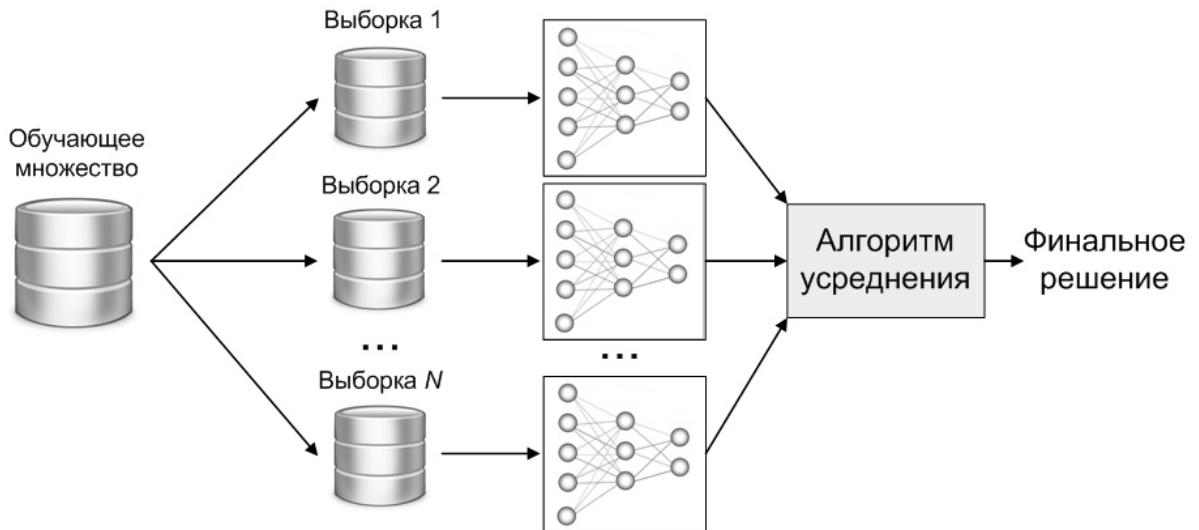


Рисунок 1.11 – Структурная схема процедуры бэггинга

К недостаткам метода можно отнести следующие:

1. Дополнительные вычислительные затраты, связанные с необходимостью формировать большое количество подмножеств обучающей выборки.
2. Подмножества примеров отличаются друг от друга, но не являются независимыми, поскольку все они основаны на одном и том же множестве.
3. Для работы алгоритма необходимо большое множество данных для настройки и обучения.

Рассмотрим следующий механизм объединения сигналов с множества нейронных сетей – смешение мнений экспертов [110]. Рассмотрим конфигурацию сети, показанную на рисунке 1.12. Такая сеть носит название смешения мнений экспертов и состоит из Q модулей, обучаемых с учителем и называемых экспертами. Эксперты строятся на основе предположения, что каждый эксперт настроен на решение задачи в определённой области пространства. Элемент, интегрирующий решения, носит название сеть шлюза. Сеть шлюза определяет

эксперта, который сможет оптимально произвести обработку заданного входного воздействия.

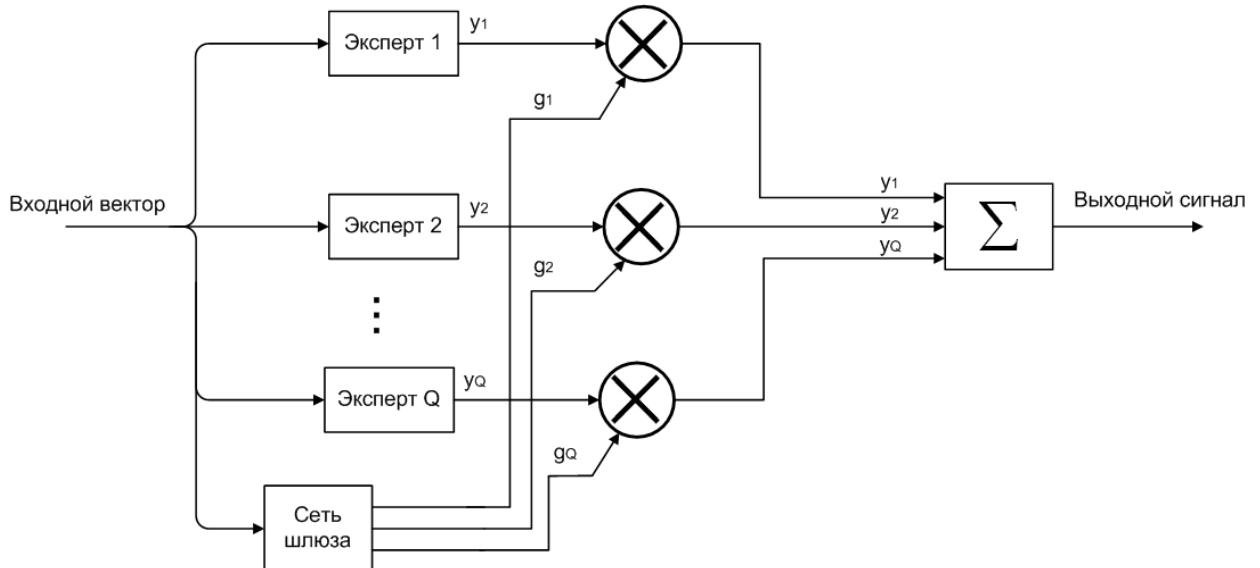


Рисунок 1.12 – Блочная диаграмма модели смешения мнений экспертов

Нейроны сети шлюза являются нелинейными, и их функция активации описывается следующим образом:

$$g_k = \frac{\exp(u_k)}{\sum_{j=1}^Q \exp(u_j)}, k = 1, \dots, Q. \quad (1.7)$$

где u_k - результат скалярного произведения входного вектора x и вектора синаптических весов нейрона сети шлюза a_{ki} ($i = 1, \dots, m$, m – размерность вектора входного сигнала) [120]. В результате данное экспоненциальное преобразование – обобщение логистической функции для нескольких входов. Сеть шлюза выдаёт значения вероятности, которые показывают какому эксперту на обработку необходимо отправить данное входное воздействие для получения желаемого результата.

Ассоциативная машина может быть спроектирована и для двух уровней иерархии. Модель, представленная на рисунке 1.13 [110], имеет два уровня иерархии. Информационная структура производит разбиение входного пространства на множество вложенных подпространств, и информация распределяется между экспертами под управлением нескольких сетей шлюзов, представляющих собой сложную иерархическую структуру.

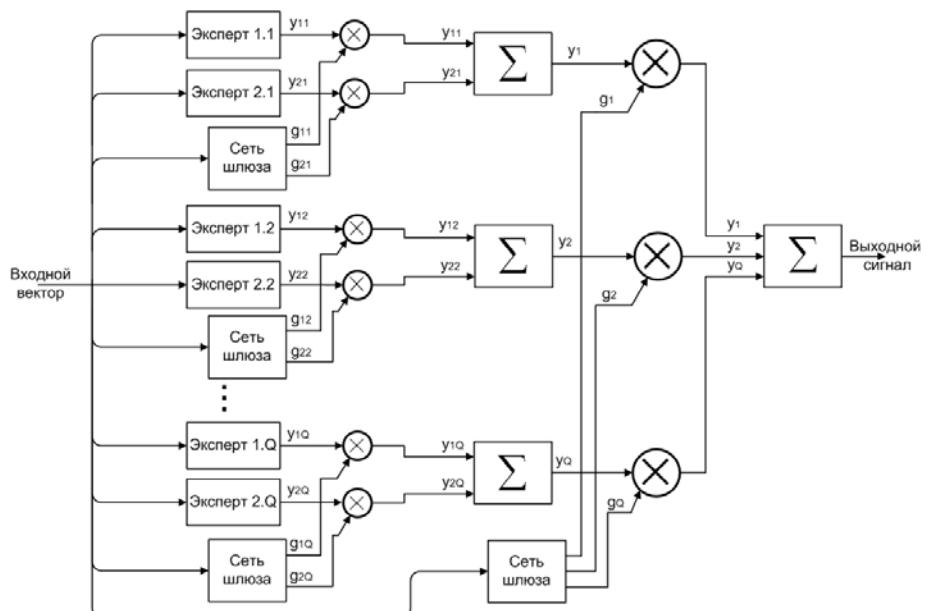


Рисунок 1.13 – Иерархическое смещение мнений экспертов

К недостаткам работы алгоритма можно отнести следующие:

1. Алгоритм требователен к вычислительным ресурсам при разбиении исходного пространства на регионы. Становится возможным создание большого количества областей, что приведёт к излишней кластеризации пространства и создаст большую группу базовых экспертов.
2. Сложность процедуры обучения и настройки иерархической модели. Чаще всего применяют процесс обучения на основе стохастического градиента. Данный метод основан на настройке весовых коэффициентов эксперта, сети шлюза первого и второго уровня, что приводит к сложному алгоритму комплексной оптимизации всей нейросетевой машины.

Основные алгоритмы объединения мнений экспертов и их недостатки приведены в таблице 1.2. Для устранения выявленных уязвимостей в данной работе предлагается применять комбинированную систему оценки компетентности экспертов на основе аппарата нечёткой логики в совокупности с системой предобработки сигналов от нейросетевых экспертов, основанной на комплексе из модифицированных нейронов Фукушимы, N – адалины и квадратичного нейрона. В качестве финальной структуры, принимающей решение на основе сигналов от экспертов, предлагается использовать сигма-пи нейронную сеть.

Таблица 1.2 Характеристики алгоритмов объединения мнений экспертов

Технология	Методология получения результата	Недостатки
Статические структуры		
Усреднение по ансамблю	Линейная комбинация выходных сигналов нейросетевых экспертов	1. Зависимость результата от правильного определения компетентности экспертов. 2. Увеличение сложности алгоритма за счёт применения алгоритмов коррекции «шумовых выбросов».
Усиление	Каждый новый эксперт базируется на результатах ранее построенных.	1. Наличие большего числа примеров обучающей выборки. 2. Вырождение комитета экспертов в сложную неэффективную нейросетевую структуру, требующую большого количества вычислительных ресурсов. 3. Последние модели экспертов обучаются на самых «сложных» примерах.
Стэкинг	Применение концепции метаобучения	1. Сложность теоретического анализа из-за множества последовательно формируемых моделей. 2. Возможно разрастание уровней метамодели, что может привести к быстрому истощению вычислительных ресурсов.
Бэггинг	Формирование множества экспертов на основе совокупности подмножеств обучающей выборки и последующего объединения мнений экспертов.	1. Дополнительные вычислительные затраты, связанные с необходимостью формировать большое количество подмножеств обучающей выборки. 2. Подмножества примеров отличаются друг от друга, но не являются независимыми, поскольку все они основаны на одном и том же множестве. 3. Для работы алгоритма необходимо большое множество данных для настройки и обучения.
Динамические структуры		
Смешение мнений экспертов	Объединение знаний экспертов за счёт применения сети шлюза	1. Алгоритм требует на высоком уровне вычислительных ресурсов при разбиении исходного пространства. Становится возможным создание большого количества областей, что приведёт к излишней кластеризации пространства и создаст большую группу базовых экспертов со сложным механизмом взаимодействия посредством сетей шлюза.
Иерархическое объединение	Объединение знаний экспертов за счёт применения иерархической сети шлюза	2. Обучение и настройка иерархической модели представляет собой сложный вычислительный процесс. Процесс обучения на основе стохастического градиента основан на настройке весовых коэффициентов эксперта, сети шлюза первого и второго уровня, что приводит к сложному алгоритму комплексной оптимизации всей нейросетевой машины.

1.3 Выводы по главе 1

1. Выполнен анализ алгоритмов, которые применяются для построения оптимальных маршрутов продвижения информации в телекоммуникационных сетях. Основной недостаток современных алгоритмов маршрутизации – потеря их работоспособности в случае полного или частичного отсутствия информации о состоянии отдельных каналов передачи.

2. Проанализирован процесс обучения нейронных сетей. Градиентные алгоритмы, применяемые как универсальные методы настройки НС на решение прикладных задач, требуют введения эвристических методов для адаптации к конкретным решаемым задачам.

3. Выявлены особенности процесса инициализации начального состояния нейронных сетей. Оптимальная настройка начальных параметров нейронной сети перед процедурой обучения позволяет значительно сократить время обучения и препятствовать попаданию сети в локальный минимум.

4. Проведён анализ алгоритмов для объединения решений, получаемых от различных интеллектуальных структур. Особенности получения подобного рода решений требуют предварительной оценки компетентности нейросетевых экспертов с последующим получением финального решения на основе произведённых оценок способности сети выдавать адекватную реакцию на входные воздействия.

2 Разработка алгоритмов настройки и обучения комитета нейросетевых экспертов

2.1 Структурная схема разрабатываемой системы

На рисунке 2.1 представлена общая структурная схема разрабатываемой системы. Основной подход для решения трудно формализуемых задач (задача безопасной маршрутизации информации) базируется на применении множества нейросетевых экспертов – совокупности нейронных сетей различной архитектуры с механизмом объединения решений (ассоциативная машина) [15].

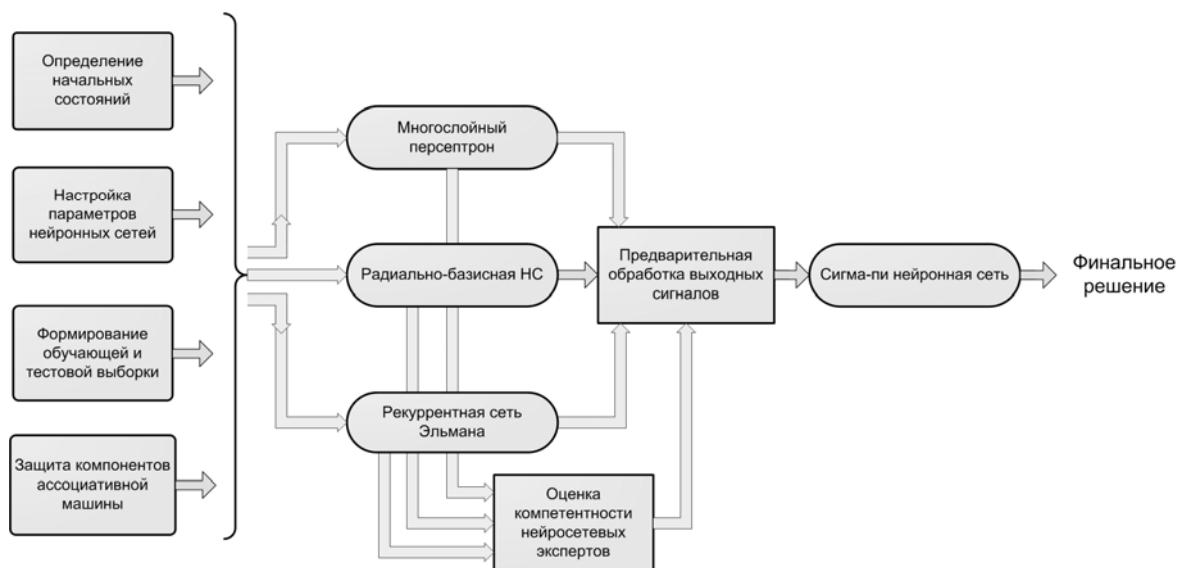


Рисунок 2.1 – Структурная схема комитета экспертов

Для возможности функционирования системы необходимо подготовить и систематизировать данные, на основе которых производится обучение отдельных нейросетевых компонентов системы. Блок *формирования обучающей и тестовой выборки* позволяет получить структурированные данные, отражающие отдельные факты предметной области. Информация для обучения НС должна быть определённым образом упорядочена и организована с целью обеспечения возможности её дальнейшей обработки с помощью нейросетевых технологий. Этот этап работы алгоритма является наиболее важным, т.к. он необходим для приобретения совокупности нейронных сетей способности к обобщению. В работе разработан алгоритм анализа каналов передачи информации и способ получения данных для формирования обучающего и тестового множества ассоциативной машины.

Блок *определения начальных состояний* формирует значения параметров всей системы перед началом её обучения. Если настраиваемые переменные обучаемой системы инициализировать таким образом, чтобы они были приближены к оптимальным значениям, то процедура обучения будет сведена к «подстройке» модели. Синтез оптимального алгоритма инициализации значительно сократит время обучения нейросетевых экспертов. В качестве алгоритма начальной установки параметров был разработан кооперативный иммунный алгоритм с генерацией решений (искусственных антител) на основе процедуры генетического поиска.

Блок *настройки параметров нейронных сетей* выполняет адаптацию компонентов нейронной сети для решения поставленной задачи. В работе процедура обучения осуществлялась для всех нейронных сетей по алгоритмам, адаптированным к их архитектурам. Для настройки нейронных сетей, являющихся экспертами, применялись градиентные алгоритмы, но модифицированные разработанными эвристиками, которые позволяют повысить эффективность процесса обучения. Для обучения финального эксперта (сигма - пи нейронная сеть) был разработан комплекс алгоритмов случайного поиска.

Блок *защиты компонентов ассоциативной машины* представляет собой нейросетевой генератор случайных чисел, построенный на основе совокупности нейронных сетей Хопфилда. Совместно с разработанным источником энтропии становится возможным получение последовательности чисел, характеристики которой приближаются к случайным. Для обеспечения нормального функционирования системы применяются алгоритмы, для работы которых необходимо множество случайных чисел. Применение данного блока позволяет сделать процесс настройки компонентов системы непредсказуемым, развивающимся по случайному направлению. В результате, состояние системы трудно предсказать в определённый момент времени, что позволяет защитить систему от нежелательного воздействия.

Блок *оценки компетентности нейросетевых экспертов* выполняет оценку нейросетевого эксперта решать заданную задачу. На основе анализа графика

среднеквадратической ошибки и параметров организации топологии нейронной сети нечёткая подсистема выдаёт коэффициент, показывающий степень компетентности нейронной сети для решения задачи.

Блок *предварительной обработки выходных сигналов* на основе полученного коэффициента компетентности от предыдущего блока выполняет подавление выходного сигнала нейронной сети пропорционально степени её компетентности. Механизм подавления сигнала основан на использовании трёх модифицированных нейронов: Фукушимы, квадратичного нейрона, N – адалины. На основе анализа выходных сигналов нейронов финальная сигма-пи сеть выдаёт решение всего ансамбля нейронных сетей.

2.2 Проектирование алгоритмов обучения экспертов

При решении сложных задач может возникнуть ситуация, когда попытки получить приемлемое решение даже при использовании различных алгоритмов, параллельно обрабатывающих одну задачу, не дают результатов. В этом случае объединение нескольких алгоритмов в композицию позволяет найти решение поставленной задачи [14]. При решении задач с помощью нейросетевых методов, построенных на применении множества нейронных сетей – ансамблей, входные данные обрабатываются с помощью нескольких НС [10]. В данной работе многослойный персепtron объединялся в ансамбль нейронных сетей совместно с радиально-базисной нейронной сетью и рекуррентной сетью Эльмана. Обучение каждой нейронной сети проводилось с применением градиентных методов с применением комбинированных эвристик.

Ансамбль для решения задачи распознавания образов состоял из трёх нейронных сетей. Многослойный персепtron обучался с использованием алгоритма обратного распространения ошибки с модификацией на основе алгоритма случайного поиска с переменным шагом. При обучении радиально – базисной нейронной сети применялся градиентный алгоритм обучения, но модифицированный эвристикой, в основе которой лежит алгоритм на основе метода упреждающего поиска. Обучение третьей НС проводилось с

использованием градиентного метода наискорейшего спуска с эвристикой на основе метода комплексов.

Перед выполнением процедуры обучения всех нейросетевых экспертов необходимо спроектировать их архитектуру, т.е. осуществить правильный выбор количества слоёв и элементов в каждом слое. Количество нейронов во входном и выходном слоях всех нейросетевых экспертов определяется условиями решаемой задачи [43]. Число скрытых слоёв необходимо выбрать в зависимости от того, насколько сложную зависимость сеть должна воспроизвести [8]. Для эксперта, представленного многослойным персепtronом, основываясь на сложности решаемой задачи, было выбрано три скрытых слоя. В радиально-базисной нейронной сети, как и в стандартной модели, был применён только один скрытый слой, состоящий из радиально-базисных нейронов. В рекуррентной сети Эльмана был использован только один слой скрытых нейронов.

После определения количества слоёв необходимо правильно подобрать количество нейронов в скрытых слоях, число которых напрямую не определяется исходными данными решаемой задачи. Метод упрощения структуры сети [121] в данной работе применять нецелесообразно, т.к. сети функционируют в составе комитета. Инициализация вычислительных нейросетевых структур, обладающих ресурсами, значительно превышающими потребности вычислительной задачи, приводит к трудоёмким экспериментам. Для решения данной проблемы, каждый нейросетевой эксперт на начальном этапе работы всей ассоциативной машины обладает минимальным количеством нейронов в скрытых слоях (для многослойного персептрана первоначальное число нейронов в скрытых слоях – 5, для РБФ сети определялось процедурой кластеризации методом k -средних, для нейронной сети Эльмана – 4 нейрона). После предварительного определения архитектуры всех сетей, производилось постепенное добавление нейронов в скрытые слои сети до достижения необходимой минимально среднеквадратической ошибки обучения.

Определив топологию нейронных сетей, применяют методы обучения. Алгоритмы обучения сетей, основанные на градиентном подходе, применяются

до тех пор, пока разница ошибок обучения для двух соседних итераций была больше некоторого порога, если величина изменения ошибки опускалась ниже заданного порога, то применялась эвристика. Вероятность применения эвристики уменьшалась по мере развития процесса обучения сети, чтобы на начальном этапе обучения препятствовать попаданию в локальный минимум и дать возможность более детальной настройки параметров сети на завершающих этапах [20].

Рассмотрим алгоритм обучения многослойного персептрона, модифицированный методом случайного поиска (рисунок 2.2):

1. Выбрать нейронный элемент из скрытого или выходного слоя нейронной сети. Веса данного нейрона рассматриваем как многомерную начальную точку x_0 с числом компонент $i = 0, 1, 2, \dots, N$, равным количеству весов, входящих в выбранный нейрон. Полагаем $F_{\min} = f(x_0)$, где F_{\min} – ошибка обучения НС, $f(\cdot)$ – ошибка обучения НС после подстановки нового значения.

2. Получить случайный вектор q единичной длины и вычислить $x_1 = x_0 + a \cdot q$, где a – величина шага.

3. Если при подстановке x_1 ошибка обучения НС изменилась не более чем на 15% и $f(x_1) < f(x_0)$, то вычислить $y = x_0 + a_s(x_1 - x_0)$, где a_s – параметр увеличения длины шага, и перейти к шагу 4. В противном случае принять $m = m + 1$ и перейти к шагу 5 (m – количество итераций, которые не дали уменьшения функции ошибки НС).

4. Если y – допустимая точка (условие допустимости из пункта 3) и $f(y) < f(x_0)$, то $a = a_s a$, $x_1 = y$ и перейти к шагу 6. В противном случае сразу перейти к шагу 2.

5. Если m превысило максимально допустимое число итераций, то выполнить изменение шага $a = a_f a$ (a_f – коэффициент уменьшения длины шага), принять $m = 0$ и перейти к шагу 6. В противном случае сразу выполнить переход к шагу 6.

6. Если достигнута требуемая ошибка обучения многослойного персептрона, то остановка алгоритма, в противном случае перейти к шагу 1.

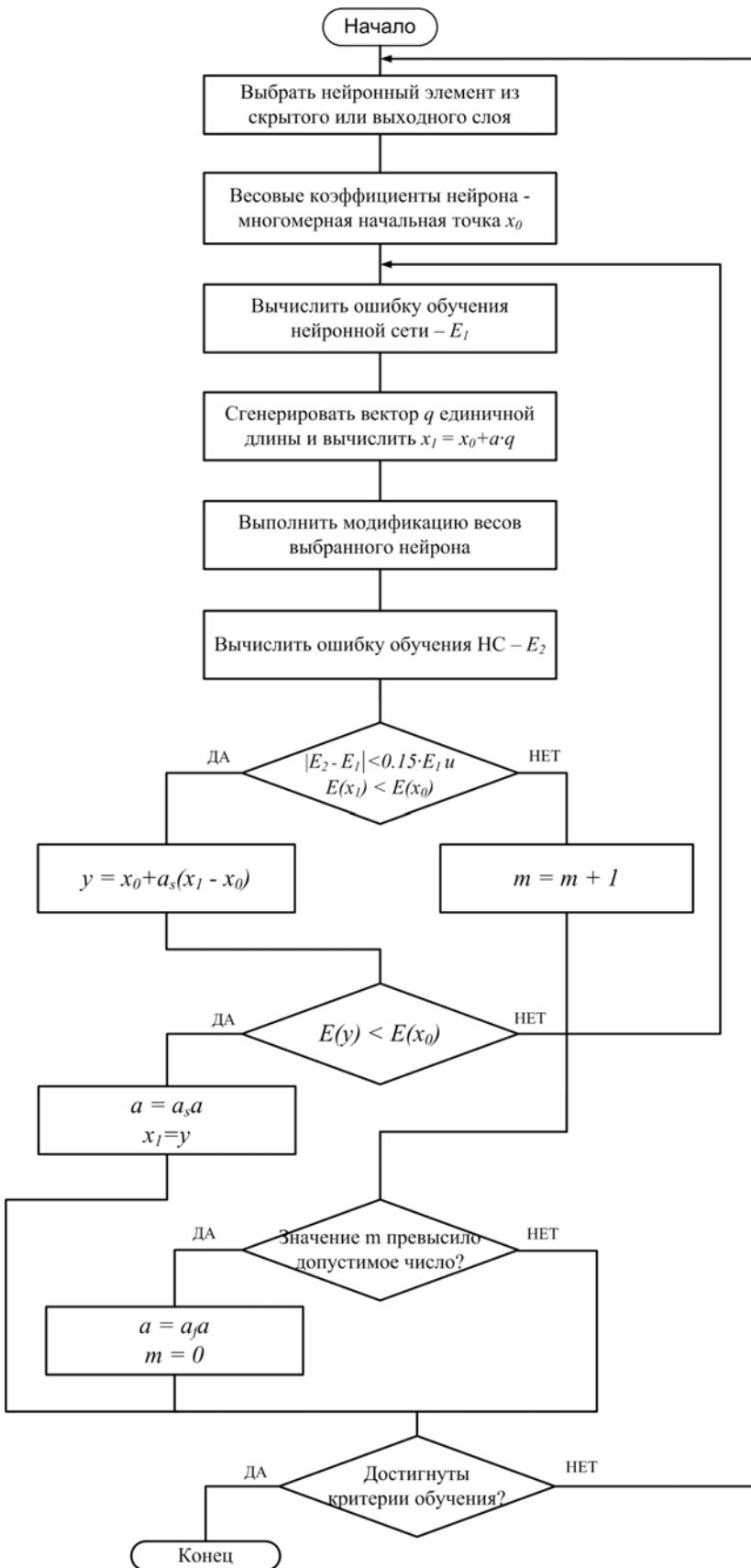


Рисунок 2.2 – Алгоритм обучения с применением эвристической процедуры для многослойного персептрона

Рассмотрим разработанный комбинированный эвристический алгоритм упреждающего поиска более подробно (рисунок 2.3):

1. Выбрать нейронный элемент из скрытого или выходного слоя нейронной сети. Веса данного нейрона рассматриваем как многомерную начальную точку x_0 с числом компонент $i = 0, 1, 2, \dots, N$, равным количеству весов, входящих в выбранный нейрон. Полагаем $F_{\min} = f(x_0)$, где F_{\min} – ошибка обучения НС.

2. По каждой i -й компоненте, входящей в выбранную точку, провести оптимизацию, зафиксировав остальные:

a. Выбрать случайным образом возможные значения незадействованной i -й компоненты для образования r допустимых комбинаций весовых коэффициентов, так чтобы у выбранных комбинаций ошибка обучения НС была меньше, чем у x_0 . Если этого достичь не удалось, повторить шаг 2 для следующей $i+1$ компоненты.

b. Определить наилучшее из r найденных комбинаций весовых коэффициентов и положить значение минимальной ошибки обучения равное F_r .

c. Произвести упреждающий поиск:

1. Для каждой из допустимой комбинации весовых коэффициентов, найденных на шаге 2(a), провести случайный выбор одного из r возможных значений следующей $i+1$ компоненты, если предположить, что она не зафиксирована.

2. Выбрать наилучшую из найденных комбинаций и зафиксировать значение компоненты i как оптимальное.

d. Если $i = N$, перейти к шагу 3. В противном случае выполнить шаг 2 для компоненты $(i+1)$.

3. Провести случайный поиск для определения наилучшего значения переменной N при фиксированных значениях других переменных. Найденную точку принять за новую базовую точку x_0 , а ошибку обучения при подстановке комбинации найденных весов из базовой точки за F_{\min} .

4. Перейти к шагу 2 с $i = 1$, если не выполнены условия окончания вычислений.

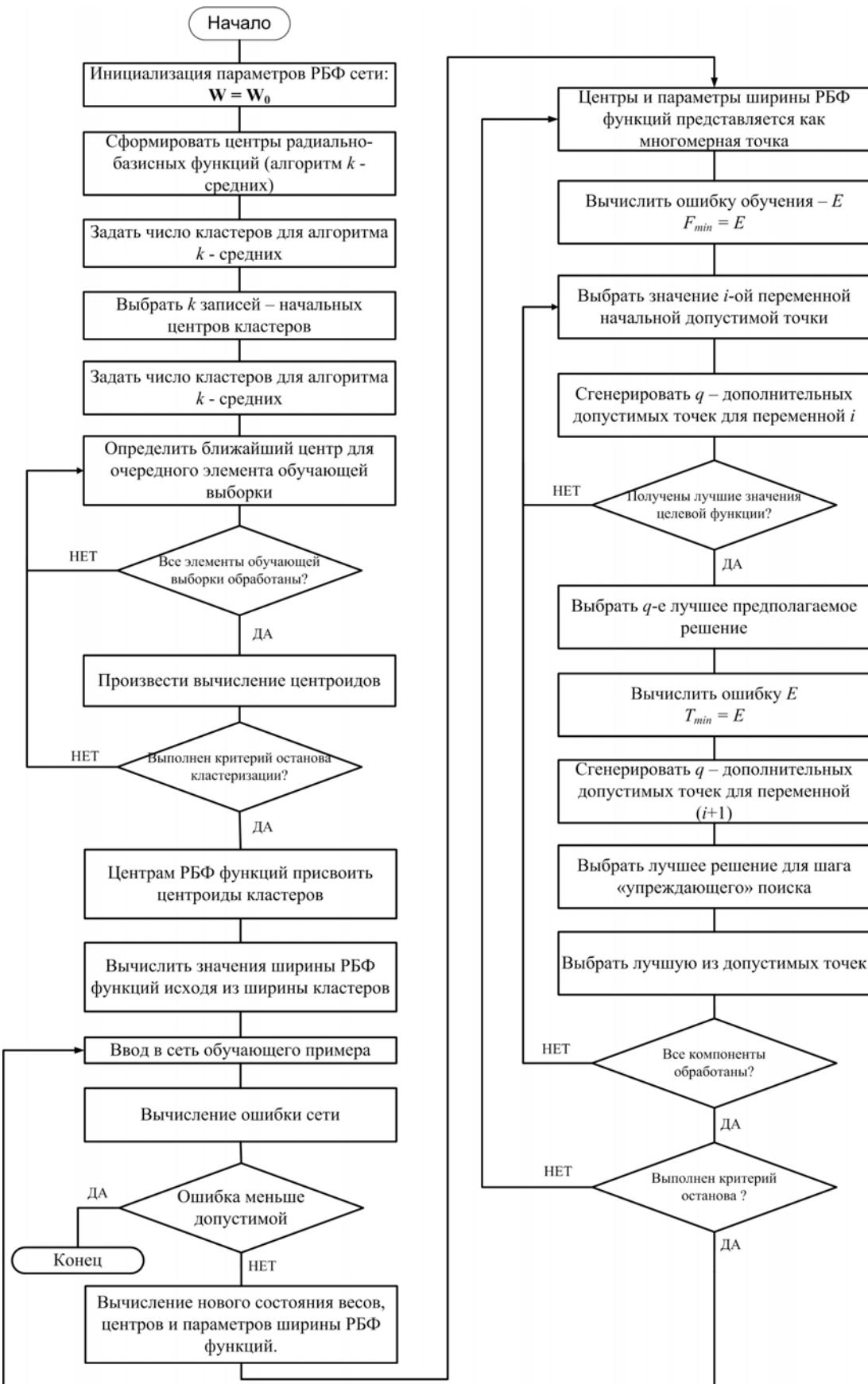


Рисунок 2.3 – Алгоритм обучения с применением эвристической процедуры для РБФ

Выполним разработку комбинированного эвристического алгоритма на основе метода комплексов [52], модифицированного для возможности его применения при обучении нейронной сети (рисунок 2.4):

1. Выбрать нейронный элемент из скрытого или выходного слоя нейронной сети. На основе весовых коэффициентов строим комплекс, состоящий из P допустимых предполагаемых значений весовых коэффициентов. Количество предполагаемых значений в комплексе зависит от числа весовых коэффициентов k в выбранном нейроне и определяется по формуле:

$$P = \text{ceiling}((\log(0,75k))^{-3} + e^{\log(0,2k)}), \quad (2.1)$$

где $\text{ceiling}(v)$ – функция, возвращающая наименьшее целое число, которое не меньше чем v , представленное в виде значения с плавающей точкой. Из анализа формулы (2.1) видно, что чем больше количество весовых коэффициентов у выбранного нейронного элемента, тем меньше предполагаемых допустимых значений весовых коэффициентов будет сгенерировано. Данная процедура необходима для предотвращения возрастания вычислительной сложности алгоритма при увеличении числа нейронов в топологии нейронной сети. В работе предполагаемое значение весового коэффициента считается допустимым, если при его подстановке в нейронную сеть ошибка обучения изменится не более чем на 15%. Для каждой точки $p = 0, 1, 2, \dots, P$ необходимо выполнить следующие действия:

- a. Случайным образом определяем допустимые предполагаемые значения весовых коэффициентов x^r .
- b. Если получены недопустимые значения, найти центр тяжести \bar{x} уже найденных значений весовых коэффициентов и выполнить преобразование для каждой компоненты предполагаемых значений весовых коэффициентов:

$$x^r = x^r + 0,5 \cdot (\bar{x} - x^r) \quad (2.2)$$

Повторять процедуру до тех пор, пока точка не станет допустимой.

- c. Повторить для всех остальных предполагаемых весовых коэффициентов.
2. Выполнить отражение комплекса:

а. Выбрать предполагаемые допустимые значения весовых коэффициентов для данного нейронного элемента, для которых при подстановке в нейронную сеть ошибка обучения будет максимальной:

$$f(x^R) = \max(f(x^r)) = F_{\max} \quad (2.3)$$

б. Найти центр тяжести \bar{x} и новые значения:

$$x^m = \bar{x} + \beta \cdot (\bar{x} - x^r), \quad (2.4)$$

где β - параметр, который задаёт расстояние отражения.

с. Если на предыдущем этапе получена допустимая комбинация и $f(x^m)$ больше F_{\max} , то необходимо уменьшить в два раза расстояние между текущей точкой и центром тяжести и продолжить поиск.

д. Если полученная комбинация допустима и ошибка обучения меньше F_{\max} , то перейти к шагу 3.

е. Если на предыдущем этапе получена недопустимая комбинация, то необходимо уменьшить в два раза расстояние до центра тяжести и продолжать до тех пор, пока комбинация не станет допустимой.

3. Если достигнуты критерии качества обучения, то прекратить поиск и продолжить обучение с помощью метода обратного распространения ошибки.

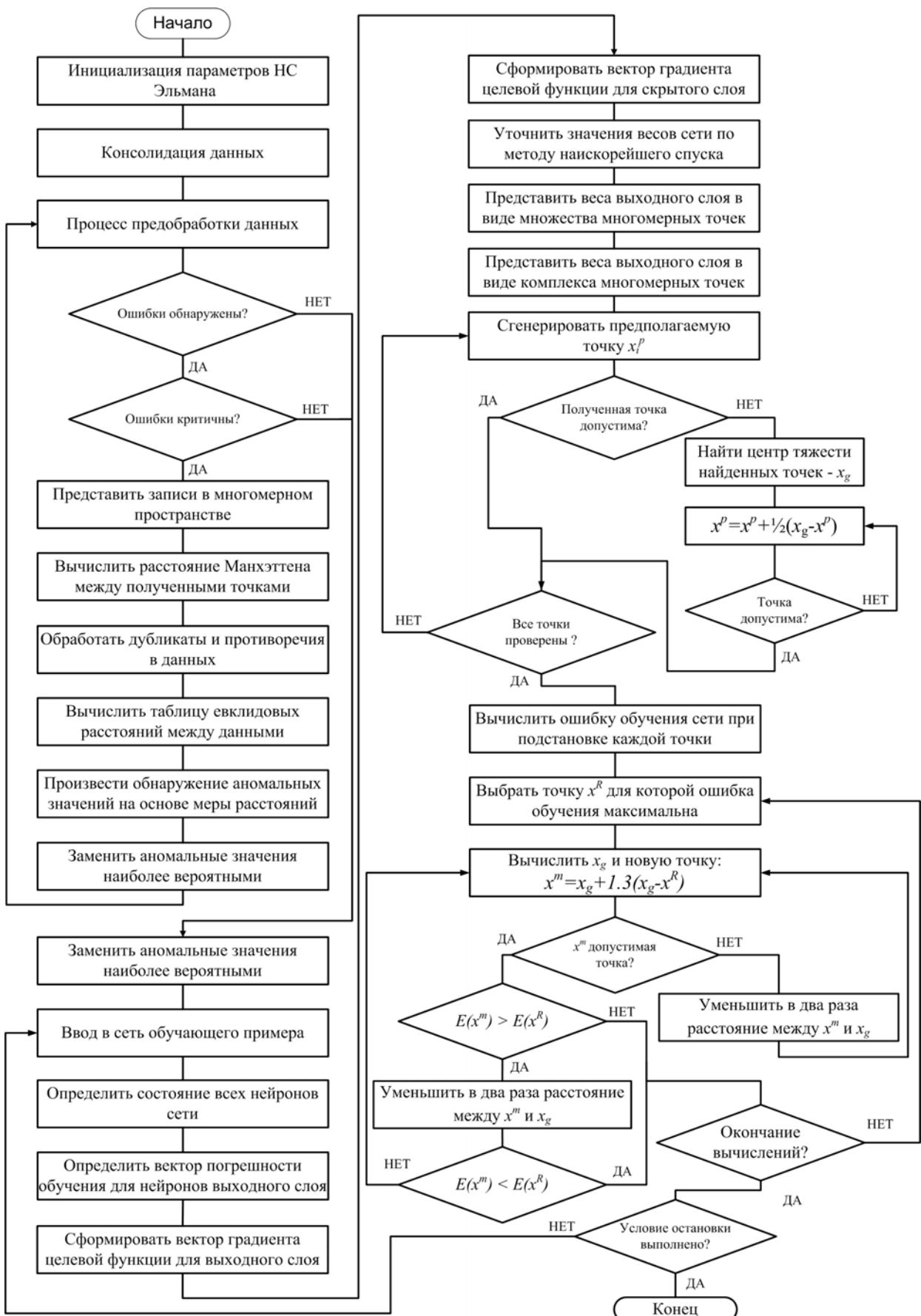


Рисунок 2.4 – Алгоритм обучения с применением эвристической процедуры для нейронной сети Эльмана

2.3 Определение начальных состояний экспертов

Для оптимального функционирования экспертов необходимо сформировать их начальные состояния, выраженные предварительной установкой весовых коэффициентов нейронных сетей. При этом следует учесть, что в данной задаче сети функционируют не каждая по отдельности, а в ансамбле. В связи с этим необходимо использовать принципы оптимизации на основе кооперативной коэволюции с несколькими популяциями [14], учитывающими совместное функционирование экспертов. В основу видоизменения популяций для получения решений в данной работе был положен иммунный алгоритм оптимизации, построенный на основе принципов иммунитета живых организмов. Предполагаемые веса нейронных сетей кодируются в антителах, образующих популяцию. В качестве антигена рассматривается задача инициализации начального состояния экспертов.

Предположим, что необходимо найти такое сочетание начальных состояний всех трёх нейросетевых экспертов, при котором процесс обучения будет происходить за наименьшее время и обеспечивать эффективность их совместного функционирования. Для каждой нейронной сети процесс оптимизации идёт независимо от остальных. Он заключается в создании новых антител с помощью операторов мутации, рекомбинации и различного рода операторов инверсии [40]. Когда возникает необходимость оценки эксперта из какой-либо популяции, то эксперты, начальные состояния которых обусловлены функционированием трёх различных популяций антител, объединяются в один ансамбль, и осуществляется оценка эффективности обучения и функционирования НС экспертов из этих начальных состояний. Таким образом, каждая популяция кооперативной коэволюции пытается найти оптимальную часть общего решения. Такой подход позволяет разбить задачу на подзадачи, что снижает сложность решения [44].

Искусственный иммунный алгоритм оптимизации функционирует на основе принципов иммунитета живых организмов – соматической теории и сетевой гипотезы [40]. Соматическая теория утверждает, что генерация новых антител

происходит за счёт мутации и рекомбинации отдельных генов хромосом. Сетевая гипотеза основана на возможности контроля развития популяции за счёт взаимного распознавания антител и контроля численности. В результате, в естественной иммунной системе осуществляется поддержание разнообразия антител и реализуется контроль над их численностью. Построенная на основе данных принципов искусственная иммунная система обладает гибкими способностями для обработки информации.

По аналогии с естественной иммунной системой молекулы в искусственной иммунной системе могут быть представлены в виде совокупности параметров объектов, выраженных в векторной форме. Математически обобщенная форма любой молекулы в пространстве может быть представлена как строка свойств длины L [49]. Эта строка может состоять из атрибутов любого типа: вещественные числа, целые числа, биты или символы. Тип атрибутов обычно зависит от предметной области задачи, которую решает искусственная иммунная система (ИИС).

Рассмотрим молекулу антитела, представленную набором параметров персептрона, и антигена. С точки зрения задачи распознавания иммунной системой антитела и антигена, связь между антителом и антигеном вычисляется как мера расстояния или мера аффинности между соответствующими строками свойств. Аффинность антитела – это скалярная величина оценки, показывающая близость результата к оптимальному значению. В качестве аффинности в данной задаче используется ошибка обучения комитета нейронных сетей: чем она меньше, тем лучше антитело. Таким образом, антиген можно рассматривать как задачу, которую необходимо решить, а антитело — метод решения задачи.

В качестве популяции антигенов выступает область всех возможных значений векторов весов и порогов нейронов. Каждое антитело кодирует векторы весовых коэффициентов и пороги нейронов. Пример возможного кодирования показан на рисунке 2.5.

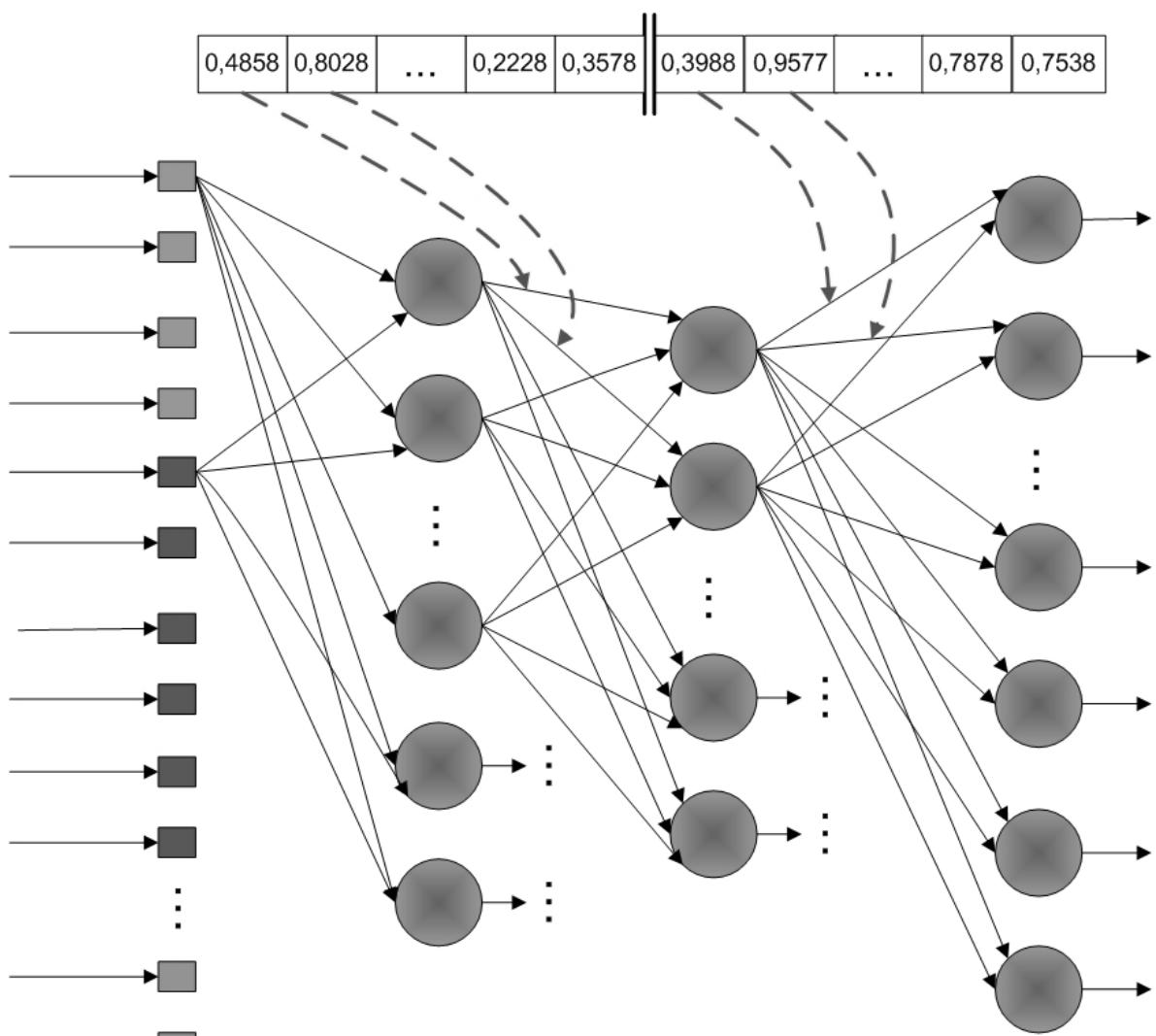


Рисунок 2.5 – Структура антитела при настройке нейроконтроллера

Под кодирование каждого параметра весового коэффициента отводится 20 бит информации. Антитело имеет разрядность кратную 20 битам и в нём закодированы все весовые коэффициенты нейросетевого эксперта. В нейронную сеть последовательно подставляются параметры, закодированные в каждом из антител популяции. Вычисляется ошибка обучения для каждого антитела. Чем меньше ошибка обучения, т.е. чем меньше аффинность (степень соответствия антитела антигену), тем лучше сформирован эксперт.

Основу видоизменения решений в иммунном алгоритме представляет собой использование операторов мутации для антител [96]. Данные типы перестроек играют важную роль в естественном иммунитете, т.к. они позволяют получать новые антитела. Использование классических операторов мутаций является не всегда эффективным, т.к. низкий процент изменения антитела может не повлиять

на решение [97]. Поэтому в данной задаче предлагается использовать два этапа мутации антител. Первый этап предполагает радикальное изменение структуры антитела. Все антитела, присутствующие в системе, упаковываются в трёхмерный параллелепипед, как показано на рисунке 2.6. Так как разрядность антитела кратна 20 битам, то длина одной стороны фигуры будет равна количеству битов q , закодированных в антителе, и далее из антител формируется горизонтальный срез. Количество антител в срезе r зависит от размера популяции. Таких срезов будет g . Получили структуру, которая в упакованном виде содержит $r \cdot g$ антител. Для возможности выполнения операторов мутации данный параллелепипед разбивается на кубики. Каждый маленький куб имеет размерность 8x8x8. Таким образом, получаем $(r \cdot g \cdot q)/512$ кубических структур, в которых содержится информация из всей популяции антител.

Операция мутации заключается в выборе в горизонтальном или вертикальном срезе одного кубика и его перемещение на место другого куба из того же среза, что приведёт к изменению антитела. Данный тип мутации является наиболее радикальным и может изменить до 50% структуры антитела, поэтому необходимо ввести вероятности выбора кубиков. Так как антитела упакованы по горизонтали, то есть вероятность перестановки кубов внутри одного антитела, что приведёт к значительному изменению всей структуры антитела. Вероятность этой перестановки должна быть достаточно мала и определяться по закону распределения Пуассона (закон редких событий) [87]. Перестановки в вертикальном срезе подчиняются экспоненциальному закону.

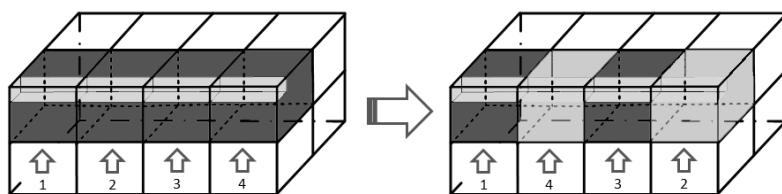


Рисунок 2.6 – Упаковка антител в операторе мутации и обмен кубиками 2 и 4

Второй этап мутации антител предназначен для изменения информации непосредственно в кубической структуре, для этого используются точечные иммунные операторы [31]. Выбирается один из кубиков и к битам, которые он содержит, применяют одно- и двухточечную мутацию, одноточечную инверсию.

Для перехода в другие области поискового пространства используют cataclysmic mutation, при применении которого изменяется до 40% битов кубика.

Иммунным алгоритмам свойственен внутренний параллелизм и возможность получения нескольких альтернативных решений [32]. Поэтому одним из способов распараллеливания иммунного алгоритма является одновременное развитие нескольких популяций, взаимодействие между которыми осуществляется с помощью механизма миграций. Событие, по которому популяции начинают взаимодействовать между собой и обмениваться решениями, наступает, когда сумма отклонений функции аффинности Aff_{max} в текущем поколении антител и Aff_{max} за последние k поколений не превосходит некоторого числа h , т.е. развитие популяции не приводит к появлению лучших решений и наступает период взаимодействия. Во время обменами антителами происходит ранжирование всех антител по функции аффинности (по возрастанию). Из каждой популяции удаляется процент худших антител, и на их место включаются антитела из другой популяции. Порядок взаимодействия популяций определяет структура, изображённая на рисунке 2.7. В работе осуществляется развитие 20 независимых популяций (P_1-P_{20}), взаимодействие между ними осуществляется в соответствии с архитектурой 2D тора.

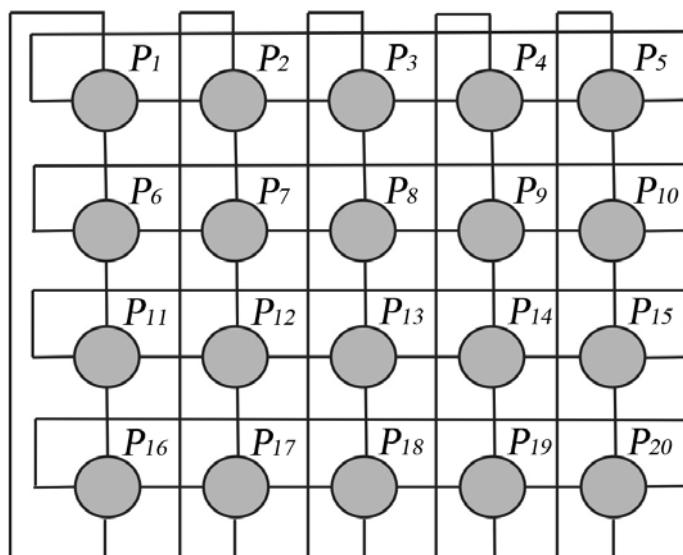


Рисунок 2.7 – Способ взаимодействия популяций – 2D тор

В данном алгоритме особое место занимает клонирование антител [40]. Клонирование – метод получения нескольких идентичных организмов путем

бесполого размножения. Однако термин «клонирование» в работе используется в более узком смысле и означает копирование антител. Полученные в результате бесполого размножения экземпляры по определению генетически одинаковы, однако и у них можно наблюдать наследственную изменчивость, обусловленную случайными мутациями или создаваемую искусственно в процессе получения оптимального решения. В работе был рассмотрен способ клонирования пропорционально аффинности. После того, как пользователь введёт процент клеток, которые необходимо воспроизвести, определённое количество клеток подвергнется копированию, а остальные будут уничтожены, так как они не прошли проверку по значению аффинности. Основной вопрос заключается в том, как выбрать эти клетки. Оптимальный выбор будет приводить к лучшему решению за меньшее число итераций работы алгоритма.

Рассмотрим алгоритм отбора антител пропорционально аффинности [96]:

1. Вычисление аффинности всех антител:

- a. Все параметры НС, закодированные в антителе, подставляются в соответствующего нейросетевого эксперта.
- b. Вычисляется ошибка обучения.

2. Найти сумму всех значений аффинности.

3. На полученную сумму делятся значения аффинности всех антител и находятся вероятности отбора антитела в новую популяцию.

$$P = \frac{Aff(P_k)}{\sum_{k=1}^n Aff(P_k)}, \quad (2.5)$$

где $Aff(P_k)$ – аффинность антитела с номером k . Чем выше уровень приспособленности для решения поставленной задачи, тем больше будет вероятность подвергнуться клонированию и получить больше соответствующих клонов.

Алгоритм обучения представляется следующим образом:

1. Задать обучающую и тестовую выборку. Обучающая выборка может быть получена на основе экспертных оценок.

2. Сформировать топологии нейроконтроллеров: задать количество и тип нейронов в каждом слое.

3. Задать функцию аффинности (функцию приспособленности) в виде:

$$E = \frac{1}{P} \sum_{i=1}^P (d_i - y_i)^2, \quad (2.6)$$

где P – число примеров в обучающей выборке, d_i - желаемая величина выхода, y_i - реальное значение выхода.

4. Используя метод вещественного кодирования сгенерировать антитела в соответствии с топологией нейроконтроллера, введя ограничения на возможные значения весовых коэффициентов с учетом вида функции активации (для сигмоидальной функции активации диапазон изменения значений весовых коэффициентов (-1, +1)).

5. Выполнение цикла для каждого поколения антител:

5.1. Реализовать цикл для каждого антитела:

5.1.1. Вычислить аффинность (с соответствующим антигеном) всех антител.

Для этого все параметры персептрона, закодированные в антителе, подставить в функционирующий нейроконтроллер и получить ошибку обучения. Эта ошибка и есть аффинность антитела.

5.1.2. Выбрать n (или $n\%$) лучших (с минимальной аффинностью) антител.

Для выполнения этого пункта необходимо отсортировать все антитела по значению аффинности и выбрать необходимое количество антител. Сколько антител необходимо выбрать, определяется на основе исследования предметной области, к которой относится решаемая с помощью данной технологии задача.

5.1.3. Клонировать выбранные клетки. Количество клонов каждого антитела обратно пропорционально его аффинности: ниже аффинность – больше клонов. Клонирование заключается в непосредственном копировании антител и добавлении их в текущую популяцию.

5.1.4. Применить операторы мутации ко всем полученным клонам.

Операторы мутации следует применять в зависимости от аффинности антител, чем ниже аффинность, тем меньше операторы мутации должны изменять

антитело (например, применить оператор одноточечной мутации). В случае если закодированное в антителе решение не соответствует требуемым критериям, то следует применить операторы, которые способны изменить структуру антитела и перейти в новые области поискового пространства.

5.1.5. Вычислить аффинность всех клонов.

5.1.6. Повторно выбрать % лучших клеток (с минимальной аффинностью к соответствующему антигену) из множества измененных клонов.

5.1.7. Удалить из популяции те клетки, аффинность которых выше порогового уровня, что приведет к уменьшению размера популяции. Пороговый уровень необходимо выбирать в зависимости от этапа решения задачи. На начальном этапе решения задачи целесообразно использовать пороговый уровень, при котором из системы удаляются не менее 30% антител (% антител определяется от характера решаемой задачи на основе экспертных оценок). Это необходимо, чтобы удалённые антитела были заменены новыми, а это увеличит пространство поиска, что позволит искать решение в разных областях допустимых комбинаций параметров в антителе.

5.1.8. Вычислить аффинности полученной популяции.

5.1.9. Удалить из популяции те клетки, аффинность которых выше порога сжатия популяции антител, что приведет к повторному уменьшению размера популяции. Порог сжатия способствует удалению из системы антител и их модифицированных клонов, которые обладают неудовлетворительными характеристиками для данной задачи. Порог сжатия следует варьировать при решении задачи с помощью данной технологии, т.к. удаление слишком большого количества антител на начальных этапах может привести к преждевременной стагнации системы и получению локального минимума решаемой задачи, а не глобального.

5.2. Вычислить аффинности всех вновь полученных антител и удалить те клетки, аффинность которых выше порога сжатия сети.

5.3. Заменить % худших антител новыми случайно сгенерированными антителами.

5.4. Вычислить функцию аффинности, соответствующую ошибке обучения E для каждого антитела по формуле в пункте 3.

5.5. Вычислить $E_{общ.среднее}$ – среднюю ошибку обучения персептрона на всей тестовой выборке.

5.6. Проверить условие останова ($E_{общ.} < \xi$, где ξ - допустимая погрешность, задаётся пользованием на начальном этапе функционирования системы).

5.7. Если условие пункта 5.6 выполнено, то вернуться к шагу 5.1., в противном случае перейти к шагу 6.

6. Конец.

При обмене антителами из популяций удаляется часть антител, также это происходит и после применения оператора мутации, т.к. для получения правильного куба из антител необходимо выполнять их клонирование для получения нужного количества. Простое удаление худших антител может привести к удалению определённой части информации, что приводит к неэффективному функционированию алгоритма обучения и увеличению времени на поиск оптимального решения. Для того чтобы сохранить информацию, накопленную в антителях, был использован адаптивный метод кластеризации по структуре основанный на применении нейронной сети АРТ – 1 [57]. Сеть обучается без учителя и реализует простой алгоритм кластеризации. В соответствии с этим алгоритмом первое антитело считается образцом первого кластера. Следующее антитело сравнивается с образцом первого кластера. Антитело принадлежит первому кластеру, если расстояние до образца первого кластера меньше порога. В противном случае, второе антитело — образец второго кластера. Этот процесс повторяется для всех следующих антител. После того, как вся популяция антител будет разбита на кластеры, вычисляется средняя аффинность каждого кластера. Антитела сначала удаляются из худшего кластера и далее из всех кластеров по порядку в порядке аффинности. Это позволяет сохранить разнообразную структуру антител.

Алгоритм замены антител с помощью алгоритма АРТ – 1 можно представить следующим образом (пример с многослойным персептроном):

1. После инициализации массива сети значениями, закодированными в антителах, они переводятся в бинарный код, т.к. сеть АРТ – 1 позволяет работать только с бинарными сигналами.
2. Запуск алгоритма АРТ – 1, разбиение антител на кластеры.
3. Вычисление аффинности всех кластеров:
 - a. Все параметры персептрона, закодированные в соответствующих антителях каждого кластера, подставляются в нейроконтроллер.
 - b. Вычисляется ошибка обучения персептрона для каждого антитела, входящего в кластер. Таким образом определяется аффинность каждого антитела Aff_i , где i – номер антитела.
 - c. Вычисляется аффинность кластера, путём сложения всех значений аффинности антител, входящих в кластер. И полученное значение суммы аффинности разделить на количество антител в кластере:

$$Aff = \frac{1}{m} \sum_{i=1}^m Aff_i, \quad (2.7)$$

где m – количество антител в кластере.

4. Полученные кластеры сортируются по значению аффинности.
5. Начиная с кластера, обладающего наибольшим значением аффинности, проводят анализ антител внутри кластера. Находят антитело с худшими свойствами (с максимальным значением ошибки обучения нейроконтроллера), и производится замена удалённого антитела на новое (антитело конструируется из генетического материала алгоритма *CNC*). Далее продолжают анализ кластеров до тех пор, пока не будет заменено заданное количество антител в популяции.

Необходимым условием для оптимальной работы данного алгоритма является метод генерации начальной популяции антител и способ поддержания разнообразия в популяции. Так как в предложенном алгоритме предлагается использование популяции антител небольшого размера, то от качественных характеристик конкретного антитела зависит результат функционирования всей системы.

Для генерации антител в данной работе был использован параллельный генетический алгоритм *CНС* (*Cross-population selection, Heterogeneous recombination and Cataclysmic mutation*) [84] данный алгоритм довольно быстро сходится из-за того, что в нем нет мутаций, следующих за оператором кроссинговера, используются популяции небольшого размера, и отбор особей в следующее поколение ведется и между родительскими особями, и между их потомками, что позволяет увеличить генетическое разнообразие популяции. В данном методе для кроссинговера выбирается случайная пара, но не допускается, чтобы между родителями было маленькое хеммингово расстояние.

Для скрещивания используется разновидность однородного кроссинговера, при котором потомку переходит ровно половина битов каждого родителя [56].

В модели *CHC* размер популяции относительно мал: около 65 особей (данное количество оптимально для поставленной задачи и было найдено путём проведения ряда экспериментов). Это оправдывает использование однородного кроссинговера и позволяет алгоритму генерировать антитела, необходимые для работы системы. Для изменения решения посредством применения операторов мутации в *CHC* используется *cataclysmic mutation*. Все хромосомы, кроме самой приспособленной, подвергаются мутации, при которой изменяется до тридцати процентов структуры хромосомы. Далее алгоритм перезапускается и продолжает работу, применяя только кроссинговер.

В разработанном алгоритме особенное внимание уделено процессу генерации антител из имеющегося хромосомного набора. Существенным недостатком алгоритмов, в которых вся структура антитела полностью кодируется в одной хромосоме (в контексте данной задачи все весовые коэффициенты в нейросетевом эксперте), являются значительные временные затраты на преобразование популяции хромосом и необходимость поддерживать популяцию хромосом большого размера. Это требует использования мощных вычислительных систем с высокой степенью распараллеливания алгоритмов поиска оптимального решения, что практически не позволяет использовать эти методы для работы в реальном режиме времени с быстро развивающимися и

динамически меняющимися внешними условиями. В связи с этим возникает необходимость в разработке всевозможных модификаций алгоритма, которые, при сохранении основной концепции, обеспечивали бы необходимую скорость обучения и, следовательно, возможность работы алгоритма в реальном масштабе времени.

Основная идея предложенного подхода заключается в том, что каждое антитело кодируется определёнными генами разных хромосом, как показано на рисунке 2.8. В пределах популяции гены хромосом соединяются друг с другом случайным образом, и получается структура, которая несёт в себе решение задачи. Такой метод позволяет уменьшить затраты времени на модификацию хромосомного набора. Т.к. решение получается путем случайной перекомбинации отдельных генов, то нет необходимости поддерживать большой размер популяции исходного набора хромосом, что приводит к тому, что искусственная иммунная система становится более гибкой и динамичной, чем та же система, основанная на принципе «в одной хромосоме – одно антитело».

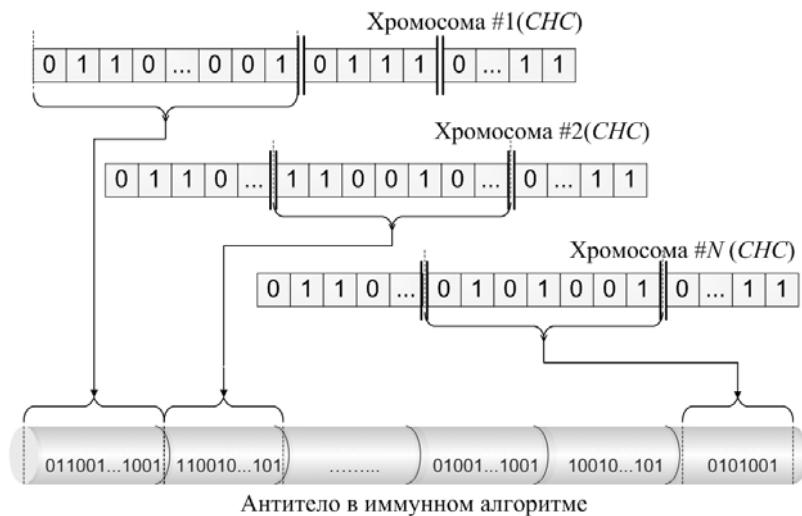


Рисунок 2.8 – Способ формирования антитела

Алгоритм генерации антител, а также генерация антител для последующей замены в уже существующей популяции можно представить следующим образом:

1. Сконструировать начальную популяцию хромосом, которая будет использована для создания антител. Ввести точку отсчёта поколений $t = 0$.

2. Выполнить начальную генерацию антител в системе по сформированной начальной популяции $P_0(t)$ путем случайного объединения в одном антителе генов различных хромосом.

3. Запустить иммунный алгоритм обучения персептрона.

4. Вычислить процент антител с минимальной аффинностью.

5. Определить, из каких генов были сформированы лучшие антитела.

6. Хромосомы, которым принадлежат данные гены, скопировать в следующую популяцию.

7. Вычислить целевую функцию для каждой хромосомы, которая вычисляется как суммарная аффинность всех антител, которые были сформированы из генетического материала данной хромосомы:

$$Fit = \sum_{i=1}^m Aff_i, \quad (2.8)$$

где Fit - функция приспособленности хромосомы для генерации антител, m – количество антител сгенерированных с использованием данного генетического материала.

8. Произвести отбор хромосом, в зависимости от рассчитанного значения целевой функции: чем меньше Fit , тем больше вероятность отбора.

9. Установить $t = t + 1$. Произвести выбор двух родительских хромосом для применения оператора однородного кроссинговера.

10. Сформировать генотип потомков. Для этого с заданной вероятностью произвести оператор однородного кроссинговера над генотипами выбранных хромосом. Далее с вероятностью 0,5 выбрать один из потомков $P_i(t)$ и сохранить его как члена новой популяции.

11. Применить оператор мутации *cataclysmic mutation*.

12. Исключить из популяции хромосомы, которые непригодны для генерации антител (из их генетического материала было сформировано мало антител с достаточной аффинностью), чтобы её размер оставался постоянным.

13. Сформировать новую популяцию хромосомных наборов для генерации антител.

14. Произвести определение функции приспособленности всей полученной популяции.

15. Проверить: если $t = t_{заданному}$ ($t_{заданное}$ – максимальное количество поколений, определяется пользователем), то переход к пункту 16, иначе к пункту 9.

16. Конец работы алгоритма.

Перед началом каждой итерации применяется метод оптимизации на основе имитации отжига для того, чтобы определить, как изменились хромосомы и антитела в популяции и привели ли эти изменения к улучшению решения. Для этого пользователь перед стартом алгоритма задаёт начальную «температуру», начиная с которой работает метод. Схема алгоритма функционирования системы представлена на рисунке 2.9.

Таким образом, обмен между разными популяциями в генетическом алгоритме, обмен популяциями антител в иммунном алгоритме и их совместный обмен решениями, благодаря свободной миграции хромосом и антител, контролируется методом имитации отжига (МИО). Контроль со стороны МИО позволяет не допустить получения нежизнеспособных популяций хромосом и антител, что, в свою очередь, позволяет использовать меньше итераций для получения оптимального решения.

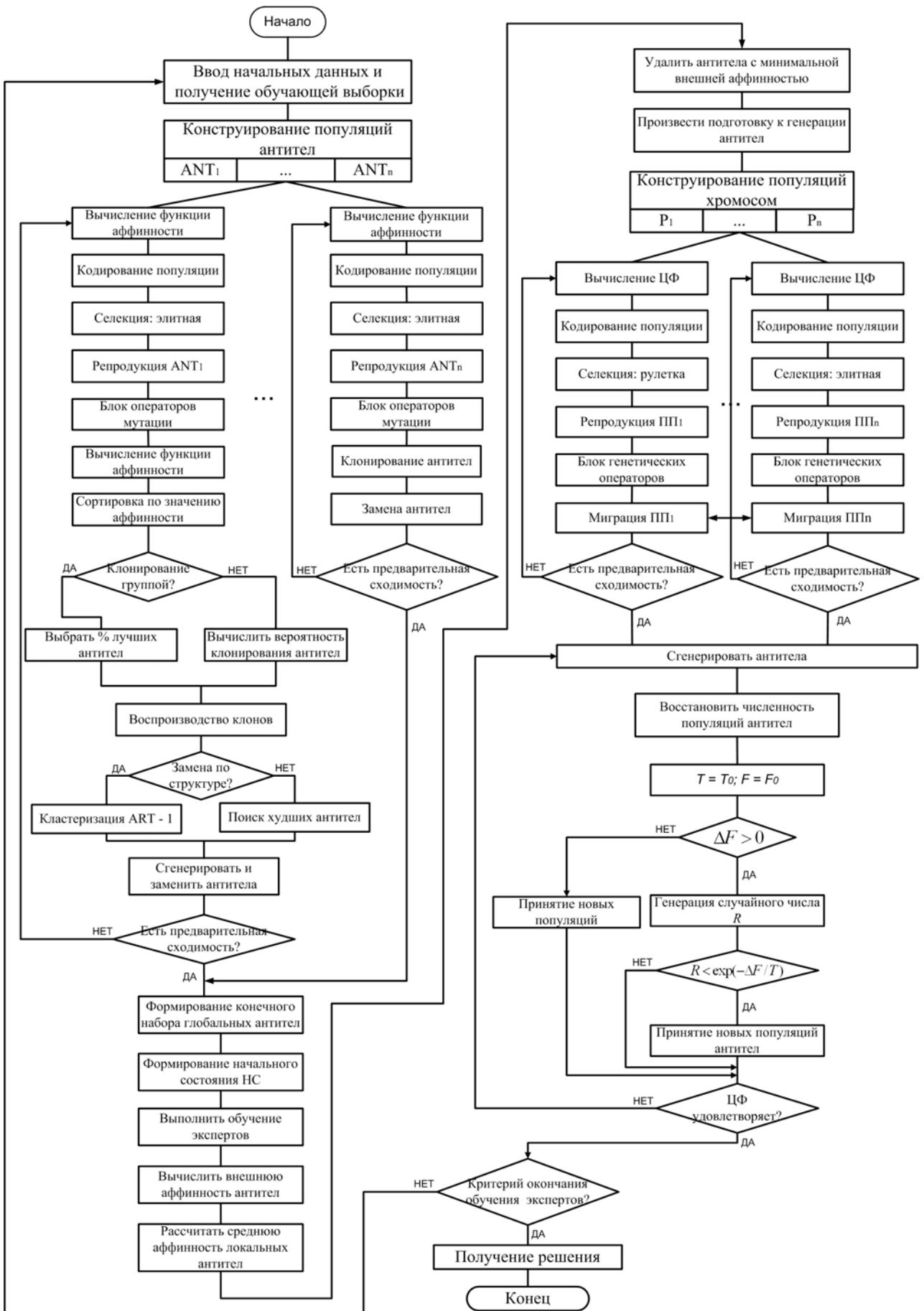


Рисунок 2.9 - Алгоритм настройки начального состояния нейронных сетей

В результате, для каждого нейросетевого эксперта создаётся отдельный комплекс популяций антител, внутри каждой популяции производится развитие антител, мутация и удаление. После изменения решений иммунными операторами производится запуск механизма миграций. При проверке эксперта производится удаление тех антител из популяции, которые не удовлетворяют критериям функционирования нейросетевой ассоциативной машины. Даже если в антителе закодировано лучшее решение для конкретного эксперта, а на уровне ассоциативной машины оно показало неудовлетворительный результат, то оно будет удалено.

2.4 Проектирование нечёткой системы оценки качества обучения нейросетевых экспертов

Нейронные сети, входящие в ансамбль и образующие ассоциативную машину, отличаются между собой архитектурой, количеством слоёв, количеством нейронов в каждом слое, начальными состояниями весовых коэффициентов. Нейронные сети должны быть достаточно различимы, чтобы погрешности, которые возникают при поиске решения одной нейронной сетью, были скомпенсированы другими членами ансамбля. После принятия решения каждой нейронной сетью все полученные выходные сигналы комбинируются, в соответствии с заданным алгоритмом.

Полученные базовые нейросетевые эксперты, представленные многослойным персепtronом, радиально-базисной нейронной сетью и сетью Эльмана, объединяются в композицию при помощи обработки их выходных сигналов комплексом нейросетевых элементов (раздел 2.5). Для обработки сигналов необходимо знать уровень компетентности каждого эксперта, в случае неправильной оценки возможностей решения поставленной задачи комитетом нейронных сетей весь ансамбль может выдать неприемлемое решение, и преимущество использования ассоциативной машины из нескольких нейронных сетей будет потеряно. Для устранения этого недостатка в данной работе применяется система предобработки сигналов от экспертов, подавляющая выходной сигнал некомпетентного эксперта (раздел 2.5). Но для нормальной

работы этой системы необходимо сформировать сигнал, характеризующий компетентность каждого нейросетевого эксперта при решении данной задачи. Для оценки компетентности применяются две нечёткие системы, генерирующие сигнал компетентности для системы предобработки сигналов.

Первая нечёткая система на этапе обучения нейронных сетей, из которых состоит ансамбль, производила оценку динамики процесса обучения. Чем лучше сформирована архитектура отдельных членов ансамбля, тем лучше они способны воспринимать примеры обучающей выборки. Для оценки качества обучения по среднему значению среднеквадратической ошибки на всём множестве примеров из обучающей выборки необходим метод, способный дать представление о качестве обучаемой нейронной сети. Одним из самых информативных способов проверки качества обучения нейронной сети является построение графика изменения среднеквадратической ошибки обучения нейронной сети.

Для получения характеристики качества каждой НС при решении данной задачи применялся метод наименьших квадратов [100], с помощью которого проводилась линейная аппроксимация графика ошибки обучения отдельных нейронных сетей из ансамбля. С помощью метода наименьших квадратов находят коэффициенты линейной зависимости, при которых функция двух переменных

$$H(a,b) = \sum_{i=1}^n (y_i - (a \cdot x_i + b))^2, \quad (2.9)$$

принимает наименьшее значение (x_i – номер итерации алгоритма обучения и y_i – среднеквадратическая ошибка обучения нейронной сети на данной итерации). При оптимальном подборе коэффициентов a и b , сумма квадратов отклонений экспериментальных данных от найденной прямой будет наименьшей. После предъявления всех экземпляров из обучающей выборки и коррекции весовых коэффициентов нейронной сети вычисляется ошибка обучения на данной итерации. В результате получаем зависимость, отражающую весь процесс обучения. Далее график ошибки преобразуется с помощью метода наименьших квадратов, и он представляется в виде прямой линии. Производится анализ угла наклона аппроксимирующей прямой относительно оси абсцисс, и вычисляется

коэффициент, показывающий нестабильность процесса обучения, который вычисляется по формуле:

$$F_{inst} = \sum_{i=2}^N (E(i) - E(i-1)) \cdot E(i-1), \quad (2.10)$$

где N – количество отсчётов среднеквадратической ошибки обучения, $E(i)$ – i -ый отсчёт значения ошибки. Например, исходя из рисунка 2.11 видно, что нейронная сеть данной архитектуры (в данном случае многослойный персепtron) лучше адаптирована для решения задачи, чем сеть, график ошибок которой изображён на рисунке 2.10.

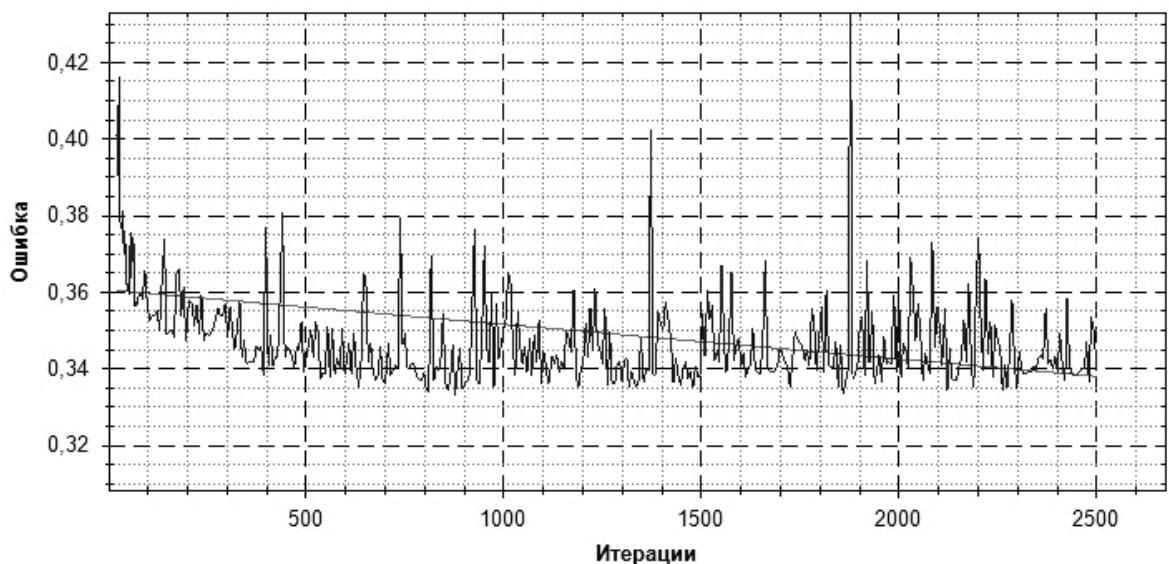


Рисунок 2.10 – График ошибки и его аппроксимация для первой НС

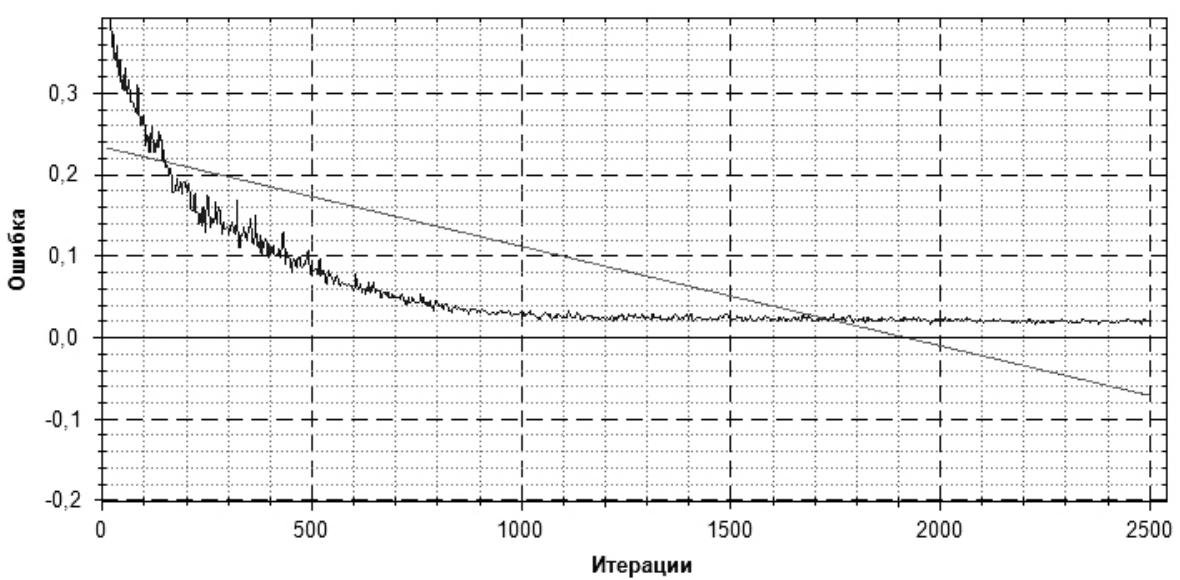


Рисунок 2.11 – График ошибки и его аппроксимация для второй НС

Вторая нечёткая подсистема проводит анализ качества сформированной структуры нейросетевого эксперта. Производится оценка сети принимать верные решения в случае определённого воздействия на элементы её структуры. Суть алгоритма заключается в целенаправленном упрощении сети. Нейронная сеть, структура и настраиваемые компоненты которой оптимизированы для решения определённой задачи, подвергается процессу упрощения её структуры за счёт избирательного отключения синаптических связей определённого нейрона, в результате чего, определённый нейронный элемент не участвует в вычислительном процессе НС для принятия решения.

Второй способ воздействия на сеть заключается в изменении порогов отдельных нейронных элементов на величину равную половине линейной комбинации входных воздействий. После осуществления этих двух операций производится оценка способности сети принимать приемлемое решение. Но на вход сети поступают данные, содержащие шумовые составляющие. Рассмотрим подробную организацию нечётких подсистем.

Процедура построения нечёткой модели

Разбиение пространства входных и выходных переменных. Для решения задачи определения компетентности каждого эксперта необходимо создать две подсистемы с базами нечётких правил следующего типа [18]:

$$\begin{aligned} \Pi_i: \text{ЕСЛИ } x_1 \text{ есть } A_{i1} \text{ И } \dots \text{ И } x_j \text{ есть } A_{ij} \dots \text{ И } x_m \text{ есть } A_{im}, \\ \text{ТО } y \text{ есть } B_i, i = 1, \dots, n, \end{aligned}$$

где Π_i - i - е правило, $i = 1, \dots, n$, n – количество правил, x_j ($j = 1, \dots, m$) – множество входных переменных, $x_j \in X_j$, X_j – область определения предпосылки, y – нечёткие выходные переменные, A, B – лингвистические термы, представленные нечёткими множествами с функциями принадлежности $\mu_A(x) \in [0, 1]$ и $\mu_B(x) \in [0, 1]$.

Для реализации двух нечётких подсистем необходимо создать две базы нечётких правил с MISO – структурой (Multi Inputs – Single Output): с двумя входными (x_1, x_2) и одной выходной переменной y . Для этого необходимо создать обучающую выборку, состоящую из множества примеров:

$$(x(k)_1, x(k)_2, d(k)), k = 1, \dots, K,$$

где $x(k)_1, x(k)_2$ – значение входных переменных x_1, x_2 в k – м примере, d – желаемое значение выходной переменной в k – м примере, K – общее количество примеров в обучающей выборке.

Рассмотрим универсальный метод построения базы нечетких правил на основе численных данных. Для настройки системы характеризовать качество обучения нейросетевого эксперта построим базу из правил для нечеткой системы. Задача заключается в формировании таких нечетких правил, чтобы сконструированный на их основе модуль управления при получении входных сигналов генерировал корректные выходные сигналы компетентности нейросетевых экспертов.

Допустим, что нам известно минимальное и максимальное значения каждого сигнала. По ним можно определить интервалы, в которых находятся допустимые значения. Например, для входного сигнала x_1 такой интервал обозначим $[x_1^{min}, x_1^{max}]$. Если значения x_1^{min}, x_1^{max} неизвестны, то можно воспользоваться обучающими данными и выбрать из них соответственно минимальное и максимальное значения. Аналогично для сигнала x_2 определим интервал $[x_2^{min}, x_2^{max}]$, а для выходного сигнала y — интервал $[y^{min}, y^{max}]$.

Для каждой из нечетких подсистем разобьем области определения переменных на отрезки, пример разбиения для первой подсистемы показан на рисунке 2.12. В данной задаче число этих отрезков фиксировано, а их длина может быть оптимизирована с помощью алгоритма настройки (рассматривается ниже). На каждом из отрезков задана одна функция принадлежности. Для первой нечеткой подсистемы сигнал x_{11} – это угол наклона прямой, аппроксимирующей график ошибки обучения нейросетевого эксперта. Параметры *min* и *max* задаются в зависимости от решаемой задачи (проведение эксперимента), т.к. значение сигнала сильно зависит от типа нейросетевого эксперта и его архитектуры. Для x_{11} определены нечеткие множества с лингвистическими значениями: NB – отрицательное большое, NM – отрицательный средний, NS – отрицательное малое, ZE – нуль, PS – положительное малое, PM - положительный средний, PB – положительное большое. Функция принадлежности для первого правила

формируется из семи элементарных функций, т.к. угол наклона аппроксимирующей прямой надо отслеживать с большой детализацией. Для создания функции применяются элементарные функции в виде трапеций и треугольников. Производя настройку параметров элементарных функций и координат вершин, получаем функцию принадлежности необходимую для правильной оценки нейросетевого эксперта.

Второй входной сигнал первой подсистемы x_{21} – это параметр, характеризующий величину колебаний графика ошибки обучения НС во времени (значение коэффициента F_{inst} , вычисляемого по формуле 2.10). Для того чтобы правильно определить области в функциях принадлежности, необходимо найти максимальное и минимальное значения возможных параметров. После этого полученный интервал разделим на 5 областей для второго правила (NB – отрицательное большое, NS – отрицательное малое, ZE – нуль, PS – положительное малое, PB – положительное большое).

Выходной сигнал первой подсистемы y_1 – сигнал компетентности эксперта. Для y_1 определены нечёткие множества с лингвистическими значениями: NB – отрицательное большое, NS – отрицательное малое, ZE – нуль, PS – положительное малое, PB – положительное большое (рисунок 2.12).

Вторая нечёткая подсистема предназначена для оценки возможностей распознавания нейросетевыми экспертами примеров обучающей выборки. Перед оценкой эксперта создаётся дубликат обучающей выборки, но в каждом примере производится изменение 20% внутренней структуры. Данная выборка подаётся на нейронную сеть, в которой производится удаление 15% нейронных элементов и пороговые значения нейронов которой подвергаются модификации в соответствии со следующей зависимостью:

$$\Theta_{ij}^1 = \Theta_{ij}^0 + \text{Sign}(a - 0,5) \cdot \left(\frac{b \cdot \Theta_{ij}^0}{100} \right), \quad (2.11)$$

Θ_{ij}^0 – первоначальное значение порогового значения j -го нейрона в i -ом слое, Θ_{ij}^1 – значение порогового значения j -го нейрона в i -ом слое после

модификации, $Sign()$ – функция, возвращающая знак аргумента, a – случайное число от 0 до 1, b – процент изменения порогового значения (в данной работе – 20%).

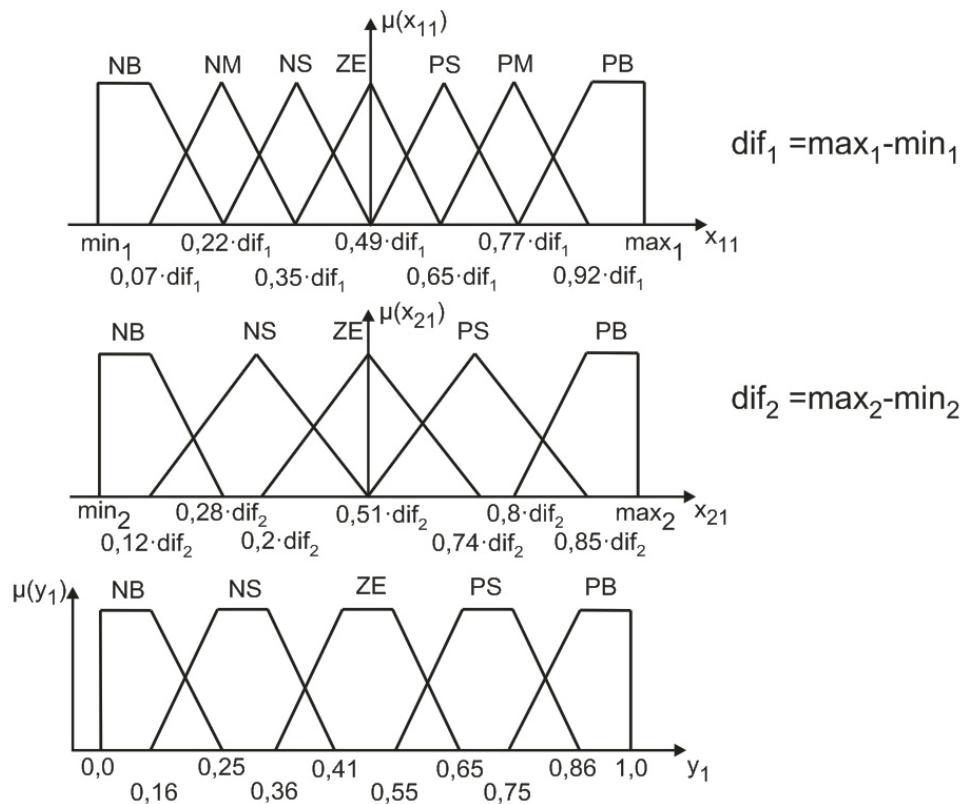


Рисунок 2.12 – Разделение пространства входных переменных x_{11}, x_{21} и входной переменной y_1

Для второй нечёткой подсистемы сигнал x_{12} – процент правильно классифицируемых примеров из модифицированной обучающей выборки, при удалении 15% нейронных элементов в нейросетевом эксперте. Для x_{12} определены нечёткие множества с лингвистическими значениями: IM – неприемлемо, NSF – не удовлетворительно, SF – удовлетворительно, GD – хорошо, VGD – очень хорошо (рисунок 2.13). Функция принадлежности для первого правила формируется из пяти элементарных функций. Для создания функции применяются элементарные функции сигмоидного типа и гауссовые функции (начальные значения переменных указаны до начала процесса оптимизации нечёткой модели):

$$\begin{aligned}\mu_{IM}(x_{12}) &= \frac{1}{1+e^{q(x_{12}-w)}}, q = 0,5, w = 15,5; \\ \mu_{NSF}(x_{12}) &= e^{-0,5 \cdot \frac{(x_{12}-a)^2}{b}}, a = 24, b = 6; \\ \mu_{SF}(x_{12}) &= e^{-0,5 \cdot \frac{(x_{12}-t)^2}{s}}, t = 43,5, s = 8,5; \\ \mu_{GD}(x_{12}) &= e^{-0,5 \cdot \frac{(x_{12}-h)^2}{k}}, h = 65,5, k = 7,5; \\ \mu_{VGD}(x_{12}) &= \frac{1}{1+e^{-v(x_{12}-m)}}, v = 0,5, m = 74,5.\end{aligned}$$

Второй входной сигнал второй подсистемы x_{22} – это параметр, характеризующий процент правильно распознанных примеров обучающей выборки при изменении порогов в 20% нейронных элементах эксперта. Для того чтобы правильно определить области в функциях принадлежности, необходимо найти максимальное и минимальное значение возможных параметров. После этого полученный интервал разделим на шесть областей для второго правила: IM – неприемлемо, NSF – не удовлетворительно, SF – удовлетворительно, GD – хорошо, VGD – очень хорошо, EXC – превосходно (рисунок 2.13):

$$\begin{aligned}\mu_{IM}(x_{22}) &= \frac{1}{1+e^{q_1(x_{22}-w_1)}}, q_1 = 0,5, w_1 = 10,5; \\ \mu_{NSF}(x_{22}) &= e^{-0,5 \cdot \frac{(x_{22}-a_1)^2}{b_1}}, a_1 = 18, b_1 = 6; \\ \mu_{SF}(x_{22}) &= e^{-0,5 \cdot \frac{(x_{22}-t_1)^2}{s_1}}, t_1 = 33,5, s_1 = 8,5; \\ \mu_{GD}(x_{22}) &= e^{-0,5 \cdot \frac{(x_{22}-h_1)^2}{k_1}}, h_1 = 47,5, k_1 = 7,5; \\ \mu_{VGD}(x_{22}) &= e^{-0,5 \cdot \frac{(x_{22}-v_1)^2}{m_1}}, v_1 = 65,5, m_1 = 7,5. \\ \mu_{EXC}(x_{22}) &= \frac{1}{1+e^{-z_1(x_{22}-p_1)}}, z_1 = 0,5, p_1 = 74,5.\end{aligned}$$

Выходной сигнал второй подсистемы y_2 – сигнал компетентности эксперта. Для y_2 определены нечёткие множества с лингвистическими значениями: EXC –

превосходно, GD – хорошо, SF – удовлетворительно, IM – неприемлемо (рисунок 2.13):

$$\mu_{EXC}(y_2) = \frac{1}{1 + e^{z_2(y_2 - p_2)}}, z_2 = 0,5, p_2 = 10,5;$$

$$\mu_{GD}(y_2) = e^{-0,5 \cdot \frac{(y_2 - h_2)^2}{k_2}}, h_2 = 28, k_2 = 10;$$

$$\mu_{SF}(y_2) = e^{-0,5 \cdot \frac{(y_2 - t_2)^2}{s_2}}, t_2 = 55,5, s_2 = 10,5;$$

$$\mu_{IM}(y_2) = \frac{1}{1 + e^{-q_2(y_2 - w_2)}}, q_2 = 0,5, w_2 = 74,5.$$

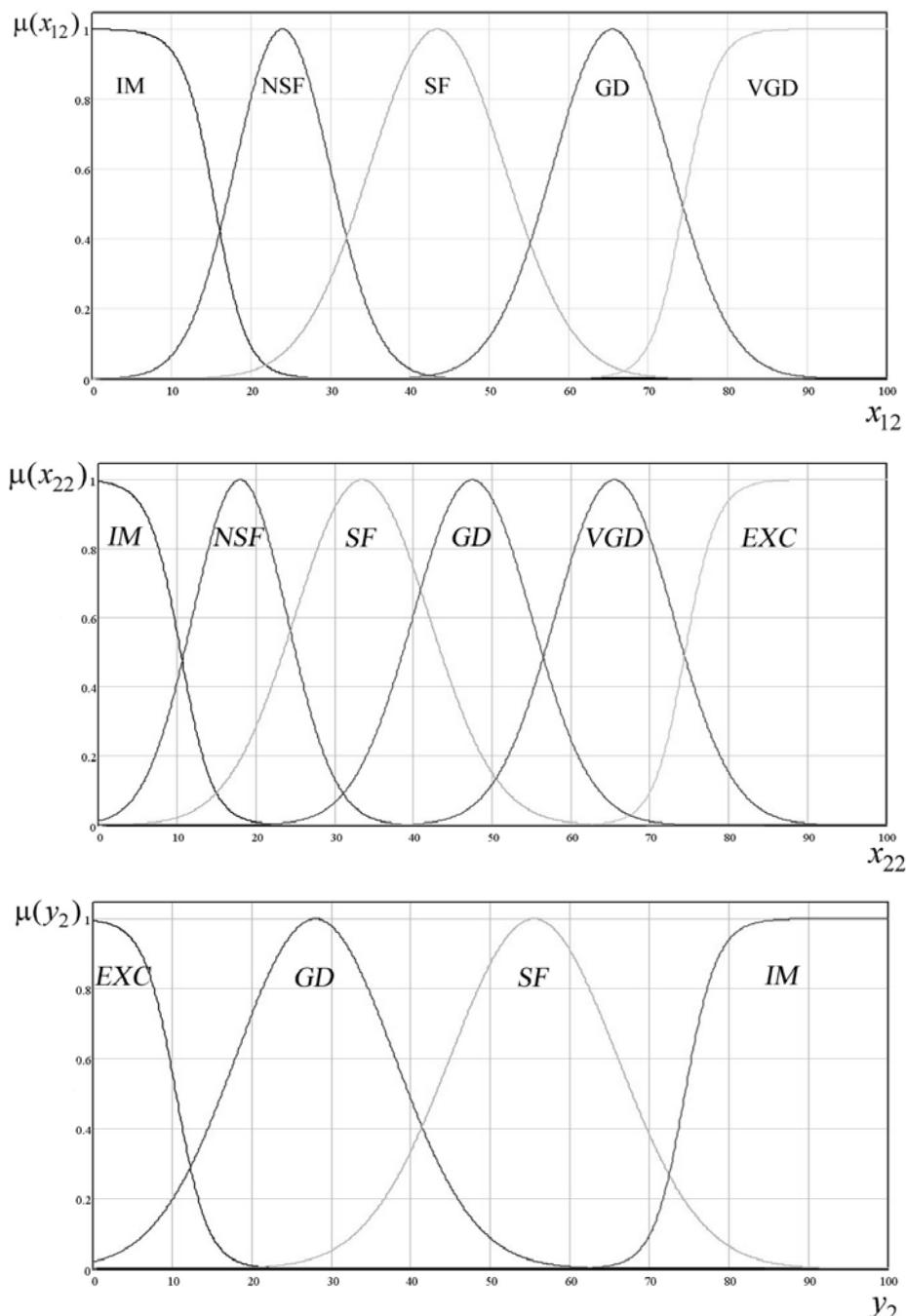


Рисунок 2.13 – Разделение пространства входных переменных x_{12} , x_{22} и входной переменной y_2

В результате настройки системы на решение прикладной задачи оценки компетентности экспертов вид функций принадлежности и способ разбиения пространств входных и выходных переменных могут измениться.

Формирование базы правил. База правил содержит логические правила, характеризующие причинно-следственные отношения между нечёткими значениями входных и выходных сигналов. Генерация множества правил для нечёткой системы осуществляется на основе анализа возможных сочетаний нечётких высказываний в предпосылках и заключениях правил. Максимальное число правил в базе для каждой из двух нечётких подсистем определяется следующим соотношением:

$$K = k_1 \cdot k_2 \cdot k_y, \quad (2.12)$$

где k_1, k_2 – число функций принадлежности для входных переменных x_{1p}, x_{2p} (p – номер нечёткой подсистемы), k_y – число функций принадлежности для выходных переменных.

В результате, сформированная для первой нечёткой подсистемы база правил будет содержать 175 правил, а вторая – 120. При таком подходе базы правил являются избыточными и содержат некоторые правила, которые никогда не будут задействованы. На основе информации о работе системы базы можно представить в следующем виде:

Таблица 2.1 Результирующая форма базы правил первой нечёткой подсистемы

x_{11}	x_{21}				
	NB	NS	ZE	PS	PB
NB	NB	NS	ZE	PS	PB
NM	NB	NS	ZE	PS	PB
NS	NB	NS	ZE	PS	PB
ZE	NB	NS	ZE	PS	PB
PS	NB	NS	ZE	PS	PB
PM	NB	NS	ZE	PS	PB
PB	NB	NS	ZE	PS	PB

Таблица 2.2 Результирующая форма базы правил второй нечёткой подсистемы

x_{12}	x_{22}					
	IM	NSF	SF	GD	VGD	EXC
IM	IM	IM	IM	SF	SF	SF
NSF	IM	IM	SF	SF	SF	SF
SF	IM	SF	SF	GD	GD	GD
GD	SF	GD	GD	GD	EXC	EXC
VGD	SF	SF	SF	GD	EXC	EXC

Введение нечёткости. Введение нечёткости в модель – процедура получения значений функций принадлежности нечётких множеств по значениям входных переменных [65]. В данной задаче входные переменные для двух подсистем имеют четкие значения, таким образом, степень истинности нечёткого выражения определяется значением функции принадлежности. Данное значение является результатом выполнения нечёткого высказывания в предпосылке правила.

Агрегирование степени истинности предпосылок правил. В двух подсистемах правила строятся по следующему принципу [91]:

$$\Pi_i: \text{ЕСЛИ } x_1 \text{ есть } A_1 \text{ И } x_2 \text{ есть } A_2 \text{ ТО } y \text{ есть } B.$$

В правиле присутствуют две предпосылки, по каждой из этих предпосылок на предыдущем этапе определяются численные значения функции принадлежности. Так как в правилах в качестве нечётких логических связок применяются связки «И», то может быть применен один из следующих операторов t -нормы (определяется алгоритмом оптимизации нечёткой системы, который будет описан ниже) [121]:

- \min – конъюнкция.
- Алгебраическое произведение степеней истинности.
- Произведение Гамахера.
- Произведение Эйнштейна.
- Ограниченнная разность.

Активизация заключений правил. На данном этапе определяются модифицированные функции принадлежности заключений отдельных правил, применяются только те правила, у которых выполнены условия.

Аккумулирование активизированных заключений правил. Вычисляется результирующая функция принадлежности с помощью процесса аккумуляции модифицированных функций принадлежности. Здесь применяется один из операторов s – нормы (определяется алгоритмом оптимизации нечёткой системы, который будет описан ниже) [121]:

- Максимум;
- Алгебраическая сумма;
- Сумма Гамахера;
- Сумма Эйнштейна;
- Ограниченнная сумма.

Приведение к чёткости (Дефазификация). Для приведения к чёткости необходимо выполнить преобразование нечётких значений найденных выходных переменных в чёткие. В данной работе применялся метод дефазификации – метод центра сумм [65]. По сравнению с другими методами (центр тяжести, центр площади, первый максимум, средний максимум и др. [86]) данный способ приведения к чёткости имеет следующие преимущества:

- Обладает невысокой стоимостью вычислений.
- В дефазификации участвуют все активизированные функции принадлежности, что обеспечивает более высокую чувствительность нечёткой модели к изменению входных сигналов.

После выполнения операции дефазификации получаем два коэффициента c_1 и c_2 (от каждой из нечётких подсистем), которые характеризуют компетентность нейросетевого эксперта в составе ассоциативной машины. Производить проектирование системы объединения мнений экспертов на основе двух параметров компетентности может привести к сложной конфигурации модулей, выдающих финальное решение. Для этого применяется механизм

позволяющий объединить два сигнала в один. Для этой цели был использован нечёткий нейрон «И», структурная схема которого изображена на рисунке 2.14.

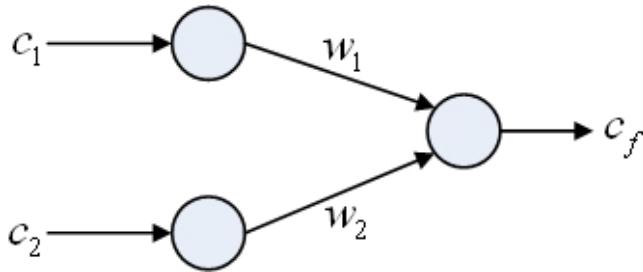


Рисунок 2.14 – Структурная схема нечёткого нейрона «И»

Итоговое значение компетентности c_f нейронной сети вычисляется по формуле:

$$c_f = T(S(w_1, c_1), S(w_2, c_2)), \quad (2.13)$$

где w_1, w_2 – настраиваемые весовые коэффициенты нейрона, сигналы c_i и w_i комбинируются с помощью операции S нормы:

$$S(w, c) = 1 - \frac{1}{\left(\frac{1}{(1-w)^\gamma} + \frac{1}{(1-c)^\gamma} - 1\right)^{\frac{1}{\gamma}}}, \quad (2.14)$$

где γ – настраиваемый параметр.

Выходное значение агрегируется с помощью операции T – нормы:

$$T(S(w_1, c_1), S(w_2, c_2)) = \frac{1}{\left(\frac{1}{S(w_1, c_1)^\gamma} + \frac{1}{S(w_2, c_2)^\gamma}\right)^{\frac{1}{\gamma}} - 1} \quad (2.15)$$

После того, как сформированы все компоненты системы, оценивающие вычислительные способности каждой нейронной сети для решения поставленной задачи, необходимо выполнить подстройку всех переменных и весовых коэффициентов, входящих в состав системы. Для этого необходимо выполнить настройку параметров функций принадлежности и весовых коэффициентов нечёткого нейрона «И», генерирующего выходной сигнал. Общий алгоритм настройки системы можно представить следующим образом:

1. Задать обучающую выборку следующего типа:

$$(x_1^{(g)}, x_2^{(g)}, \dots, x_h^{(g)}, d^{(g)}), \quad (2.16)$$

где $x_1^{(g)}, x_2^{(g)}, \dots, x_h^{(g)}$ – значения входных переменных, $d^{(g)}$ – желаемое значение выходной переменной в g -ом примере, $g = 1, \dots, S$, S – общее количество примеров в выборке.

2. Значения входных переменных каждого примера обучающей выборки подставляются в нечёткую модель, и вычисляется отклик модели – y .

3. Вычисляется функция ошибки для всех примеров:

$$E^{(g)} = \frac{1}{2}(y^{(g)} - d^{(g)})^2 \quad (2.17)$$

4. Производится корректировка параметров функции принадлежности нечётких правил с помощью оптимизационного алгоритма: градиентного, эволюционного и др. [86].

5. Проверка выполнения условия остановки:

$$E = \frac{1}{S} \sum_{g=1}^S (y^{(g)} - d^{(g)})^2 < \varepsilon, \quad (2.18)$$

где ε – допустимая погрешность.

Перед применением оптимизационного алгоритма необходимо разработать способ представления функции принадлежности в виде кодовой структуры. Способ кодирования различных функций принадлежности показаны в таблице 2.3.

На основании таблицы 2.3 параметры функций принадлежности становятся возможным представить в виде кодовой последовательности следующего типа (рисунок 2.13, верхняя функция принадлежности):

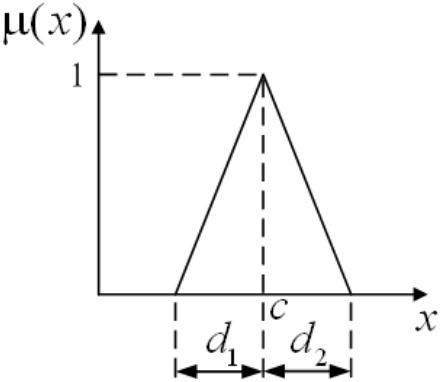
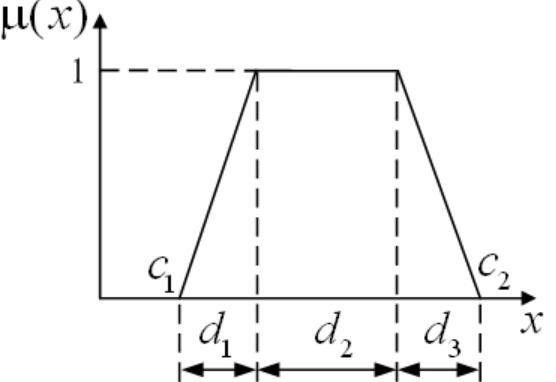
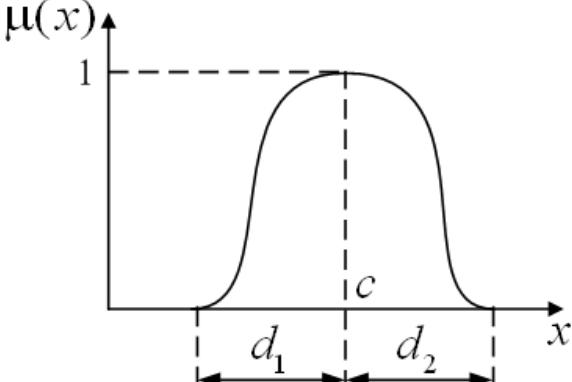
c^{IM}	d_1^{IM}	d_2^{IM}	c^{NSF}	d_1^{NSF}	d_2^{NSF}	c^{SF}	d_1^{SF}	d_2^{SF}	c^{GD}	d_1^{GD}	d_2^{GD}	c^{VGD}	d_1^{VGD}	d_2^{VGD}
----------	------------	------------	-----------	-------------	-------------	----------	------------	------------	----------	------------	------------	-----------	-------------	-------------

Для возможности осуществления процедуры оптимизации вводится несколько условий:

1. Минимальная площадь перекрытия двух функций принадлежности должна составлять не менее 10%.

2. Максимальная площадь перекрытия двух смежных функций принадлежности должна быть не более 55%.

Таблица 2.3 Оптимизируемые параметры функций принадлежности

Тип функции принадлежности	Способ представления в кодовой форме
	c – координаты вершины, d_1 и d_2 – параметры ширины функции принадлежности.
	c_1 и c_2 – координаты размещения функции принадлежности, d_1, d_2, d_3 – параметры функции принадлежности.
	c – координаты центра функции принадлежности, d_1, d_2 – параметры ширины функции принадлежности.

Основу алгоритма оптимизации составляет механизм перемещения в пространстве допустимых значений переменных на основе жестко фиксированного «скелета» - совокупности базовых узлов с фиксированными связями-переходами, включающими в себя определённый набор точек пространства, находящихся вблизи предварительно выделенных центров

кластеров поискового пространства (поисковый граф). Работу алгоритма можно представить следующим образом:

1. Процесс построения графа начинается с предварительного задания количества базовых узлов. Каждый узел может иметь определённое количество связей (в данной работе число связей и их топология генерируется случайно (нейросетевой ГСЧ, раздел 2.6)) с другими компонентами «скелета». Основным правилом при построении переходов между узлами является наличие только одной связи между смежными узлами – два узла не могут иметь два перехода между собой. Пример построения базового «скелета» приведён на рисунке 2.15. На рисунке видно, что узлы 3, 5, 6 не имеют свободных связей, для остальных вершин необходимо дополнить структуру данного графа дополнительными элементами.

2. После формирования «скелета» из минимального количества узлов алгоритм продолжает работу путем добавления дополнительных узлов, пока у всех узлов не исчерпается запас «пустых» связей (рисунок 2.15).

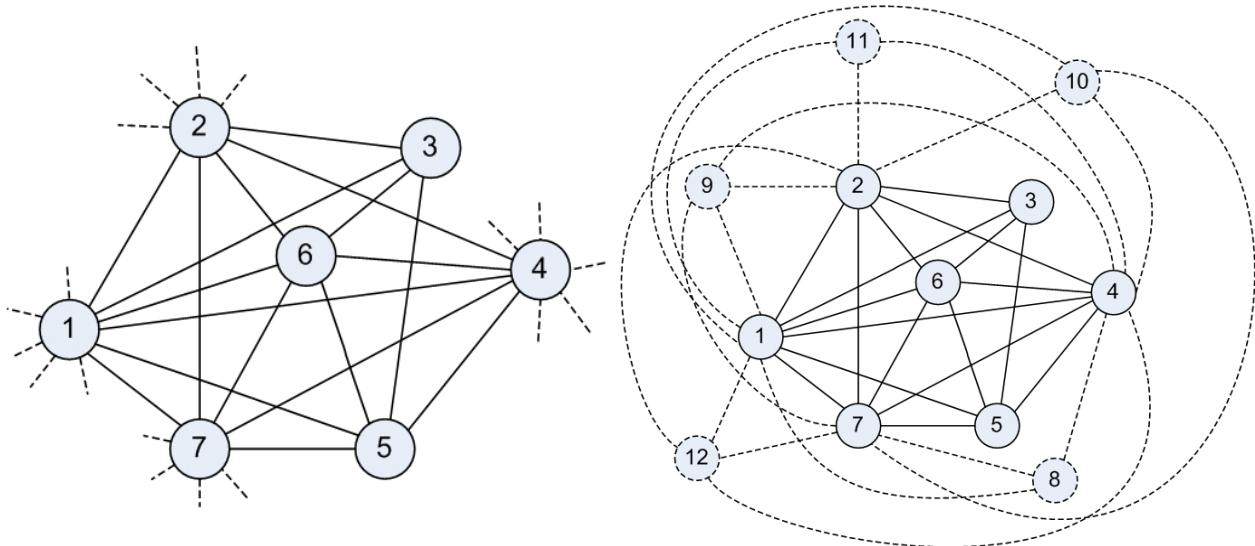


Рисунок 2.15 – Выполнение шагов №1 и №2 алгоритма оптимизации

3. Выполняется раскраска графа с помощью алгоритма последовательного раскрашивания [76]. После выполнения данной процедуры каждая вершина графа будет иметь определённый цвет, такой что две смежные вершины не получают одинаковых цветов. Один из возможных результатов работы алгоритма приведён на рисунке 2.16.

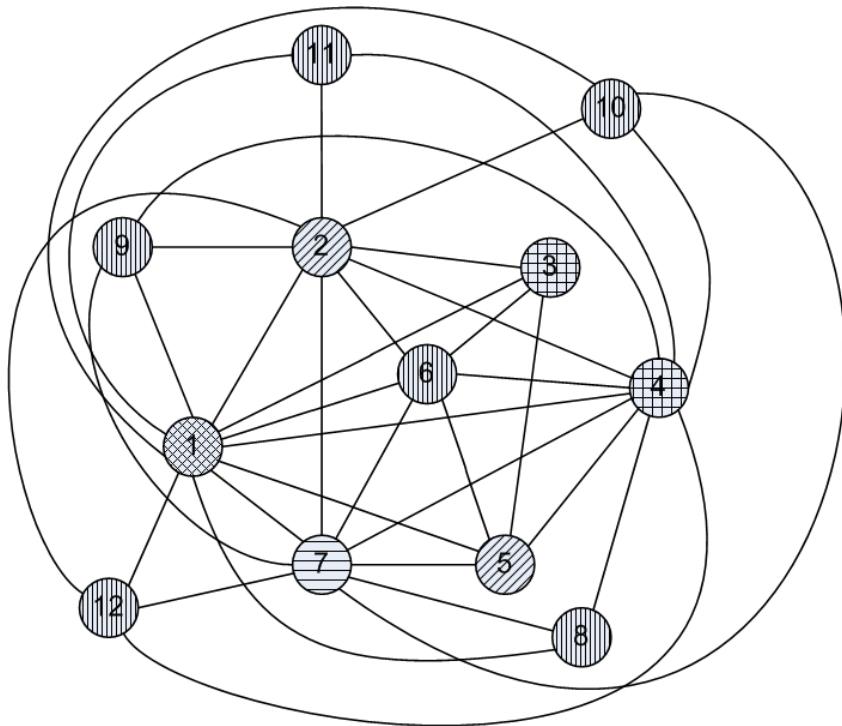


Рисунок 2.16 – Граф после раскраски

4. В каждом узле графа необходимо разместить некоторое множество допустимых предполагаемых решений, структура которых соответствует возможному варианту конфигурации нечёткой системы. Узлы с разными цветами должны содержать решения максимально отличающиеся по внутренней структуре. В качестве такой меры близости решений в данной работе использовалось евклидово расстояние:

$$\rho(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (2.19)$$

где x, y – векторы данных, n – размерность вектора. Для наполнения вершин графа данными в соответствии с ограничениями, накладываемыми на интервалы значений входных переменных, производится генерация некоторого множества предполагаемых решений (МПР) равномерно распределённых в поисковом пространстве. Далее множество сгенерированных решений необходимо разбить на группы – кластеры. Для решения данной задачи был применён алгоритм «Пульсар» [100], модифицированный для эффективного функционирования в данной вычислительной системе:

а. Перед началом работы алгоритма необходимо определить ряд констант, определяющих развитие процесса кластеризации. Для начала необходимо задать r_{min} и r_{max} – максимальный и минимальный радиусы кластеров:

$$r_{min} = 0,25 \cdot R_{max}, \quad r_{max} = 0,95 \cdot R_{max}, \quad (2.20)$$

где R_{max} – расстояние до элемента МПР максимально отдалённого от центра C , отдельные компоненты которого вычисляются по формуле (среднее всех элементов МПР):

$$C_i = \frac{1}{N} \sum_{j=1}^N g_{ji}, \quad (2.21)$$

где N – общее число предполагаемых решений в МПР, $i = 1\dots k$, k – размерность вектора предполагаемого решения, g_{ji} – i -ая компонента j -го предполагаемого решения.

Далее необходимо определить n_{min} и n_{max} – минимальное и максимальное количество элементов в кластере. В каждом узле предполагается размещать по 100 возможных решений. На основании этого n_{min} и n_{max} вычисляем по формулам:

$$n_{min} = clr_i \cdot 100, \quad n_{max} = 0,35 \cdot clr_i \cdot 100, \quad (2.22)$$

где clr_i – количество узлов, раскрашенных в i – й цвет, т.к. количество кластеров равно количеству цветов, присутствующих в раскрашенном графе.

В процессе работы алгоритма радиус кластера будет изменяться, причём изменение величины может быть как в большую, так и в меньшую сторону. Если знак приращения радиуса изменился, то считается, что произошло колебание радиуса. Величина μ_{max} – задаёт допустимое число колебаний радиуса (в данной работе $\mu_{max} = 10$). Процесс изменения адаптации радиуса кластера зависит от величины Θ – порога, регулирующего данное изменение (в данной работе $\Theta = 0,05 \cdot R_{max}$).

Следующим подготовительным этапом является определение центров предполагаемых кластеров. Для равномерного формирования кластеров по

всему объёму поискового пространства сформируем допустимую область, в которой разрешается размещать их центры:

$$0,4 \cdot R_{\max} < R_{perm} < 0,6 \cdot R_{\max}, \quad (2.23)$$

где R_{perm} – допустимый радиус нахождения центра кластера. Так на рисунке 2.17 дано пояснение для случая, когда вектора решений имеют размерность равную 3 (для процесса поиска решений с большей размерностью вся процедура проводится аналогично). Внешняя сфера имеет радиус R_{\max} . Две другие сферы проведены с радиусами $0,6 \cdot R_{\max}$ и $0,4 \cdot R_{\max}$. Допустимое пространство для размещения центров кластеров расположено между двумя этими сферами.

После случайного выбора координат центра первого кластера, определяется точка пространства максимально удалённая от центра уже сформированного кластера, но лежащая в пределах допустимой области – это и будет центр второго кластера. Аналогичным способом определяются координаты центров всех предполагаемых кластеров.

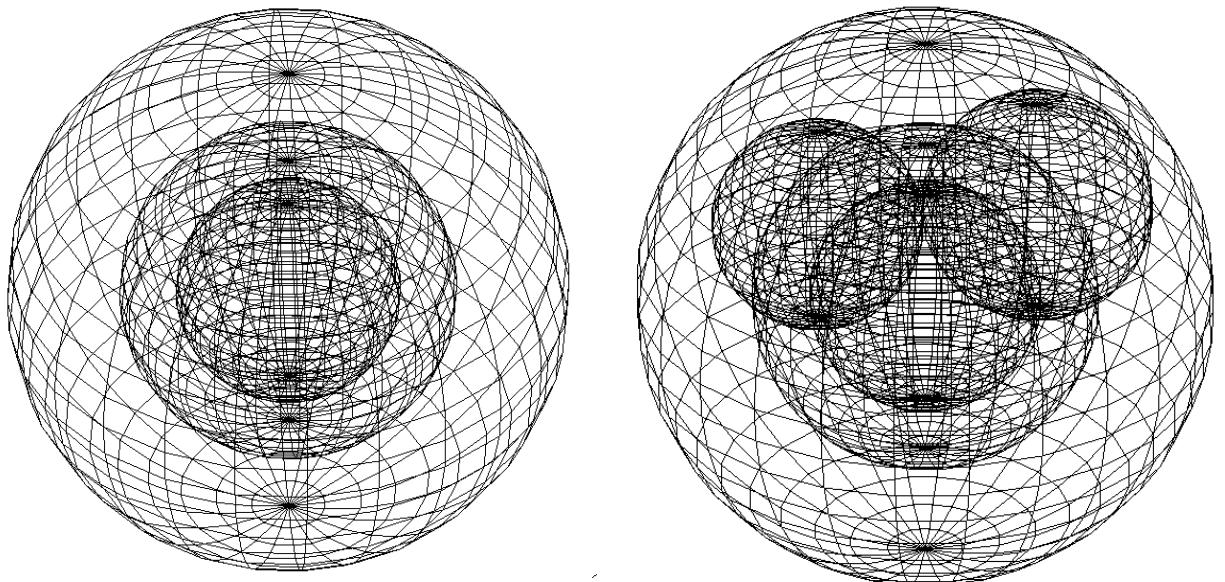


Рисунок 2.17 – Выбор центров кластеров

b. Начинается процесс кластеризации. Выбирается центр первого кластера, r_h и μ (h – номер шага):

$$r_h = \frac{r_{\min} + r_{\max}}{2}, \quad \mu_0 = \mu_1 = 1. \quad (2.24)$$

- c. Выбирается множество элементов, удалённых от центра кластера не более чем на r_h .
- d. Производится обновление координат центра. Центр вычисляется как среднее всех компонентов элементов, определённых на текущем шаге.
- e. Производится корректировка радиуса кластера. Если число выделенных элементов не превосходит n_{min} , то вычисляем радиус по следующей формуле:

$$r_{h+1} = \min\left(r_h + \frac{\Theta}{\mu_h}, r_{max}\right). \quad (2.25)$$

Если число элементов больше n_{max} :

$$r_{h+1} = \max\left(r_h - \frac{\Theta}{\mu_h}, r_{min}\right) \quad (2.26)$$

- f. Если произошло колебание радиуса, то увеличиваем μ_h на 1.
- g. Если центр и радиус перестали изменяться или $\mu_h > \mu_{max}$, переходим к формированию нового кластера, если произошли изменения, то к шагу c.

После выполнения процедуры кластеризации, каждому из цветов вершины графа сопоставляется кластер. Примеры, попавшие в кластер, используются для размещения их данных в одноцветных узлах графа. В результате, в каждом узле графа, имеющем определённый цвет, концентрируются данные из разных областей поискового пространства.

5. Процедура оптимизации нечёткой системы начинается со случайного выбора начального узла. Далее выбирается один из переходов, связывающий текущий узел с остальными узлами данного графа, и осуществляется переход к следующему узлу. Для процесса оптимизации необходимо знать полную историю перехода – узел, из которого был совершен переход (обозначим как вершина A), а также узел, в который перешли (вершина B). В узле A выбираем одно решение из имеющихся вариантов предполагаемых комбинаций, аналогичный выбор осуществляем и для узла B. Вычисляем евклидово расстояние между выбранными решениями – $dist$. На основе этой информации производим некоторое множество вариантов организации нечёткой системы, путём выбора точек поискового

пространства относительно центра A , но находящихся в пределах радиуса $dist$. Если полученные решения не удовлетворяют ограничениям, накладываемым на значения входных переменных, то данный вариант системы не принимается. В результате таких переходов между узлами производится генерация предполагаемых решений и настройка нечёткой системы.

2.5 Разработка алгоритма предобработки сигналов от нейросетевых экспертов

Применение ансамблей нейронных сетей для решения сложных задач идентификации и распознавания набора данных включает в себя алгоритмы для объединения мнений экспертов [69]. При использовании ассоциативных машин в качестве механизма для объединения решений множества решающих элементов необходимо применять структуры, модифицированные под решение конкретной задачи, на основе искусственных нейронов. Для каждого из типа экспертов применяется определённый тип искусственного нейрона. Система, построенная для смешения мнений экспертов, производит предварительную обработку сигнала для его последующей выдачи финальной нейронной сети, которая и примет итоговое решение по итогам функционирования ассоциативной машины. На основе информации из нечёткой подсистемы оценки эффективности функционирования нейросетевых экспертов производится подавление или усиление сигнала от соответствующей нейронной сети. Этот механизм позволяет акцентировать внимание финальной НС на тех решениях, которые имеют наибольшую достоверность.

В качестве первой структуры для предобработки сигналов от первого эксперта, представленного многослойным персепtronом, рассмотрим нейрон Фукусимы. В структуру данного искусственного нейрона были внесены изменения для успешного решения поставленной задачи коррекции сигнала от нейросетевого эксперта. Нейрон Фукусимы отличается от нейрона Маккалоха – Питтса тем, что у последнего весовые коэффициенты могут быть отрицательными, нулевыми, положительными [53]. В модели Фукусимы все веса полагаются неотрицательными, а входы представлены двумя группами

рецепторов – возбуждающими $x_1 \dots x_n$ и тормозящими $v_1 \dots v_p$, как показано на рисунке 2.18.

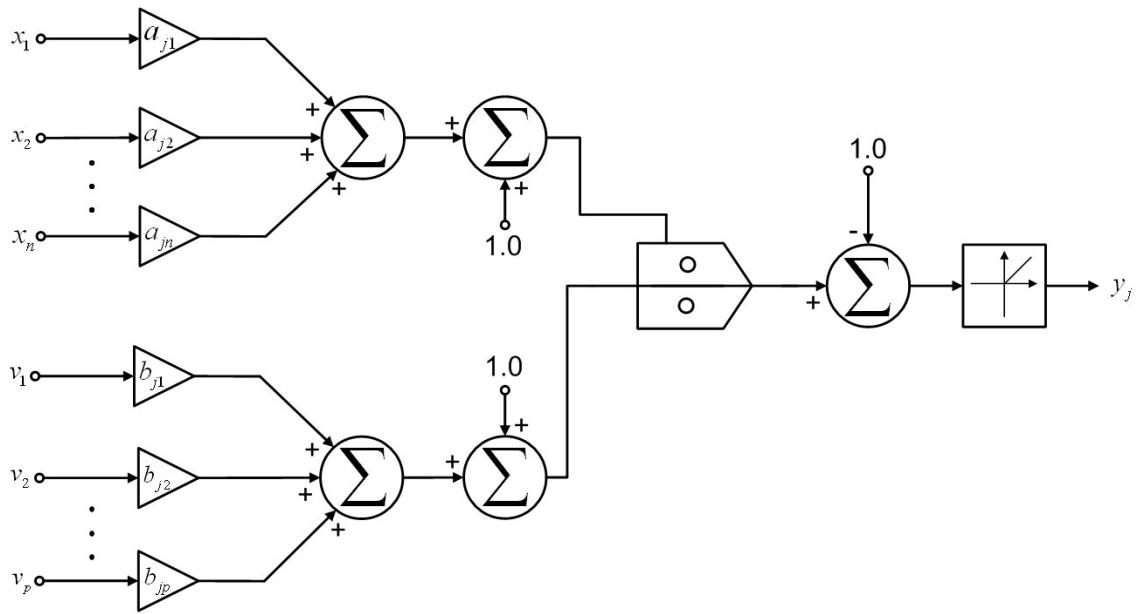


Рисунок 2.18 – Нейрон Фукушимы

Нечёткая система оценивает состояние эксперта и качество его обучения, и эти данные подаются на тормозящие входы нейрона Фукушимы. В результате преобразования сигнал с нейросетевого эксперта будет уменьшаться по амплитуде в соответствии с алгоритмом, выполняемым нейроном. В соответствии с рисунком 2.19 в структуру нейрона Фукушимы были добавлены элементы задержки, в результате чего при обработке сигнала учитывается «история» функционирования соответствующей нейронной сети. В дополнение к этому в структуру нейрона были добавлены логические связи, подавляющие сигналы с выходов нейронного элемента, если они не соответствуют соответствующим сигналам с входных рецепторов (элементы AND – логическое «И», функция *sign* – определение знака входного аргумента.).

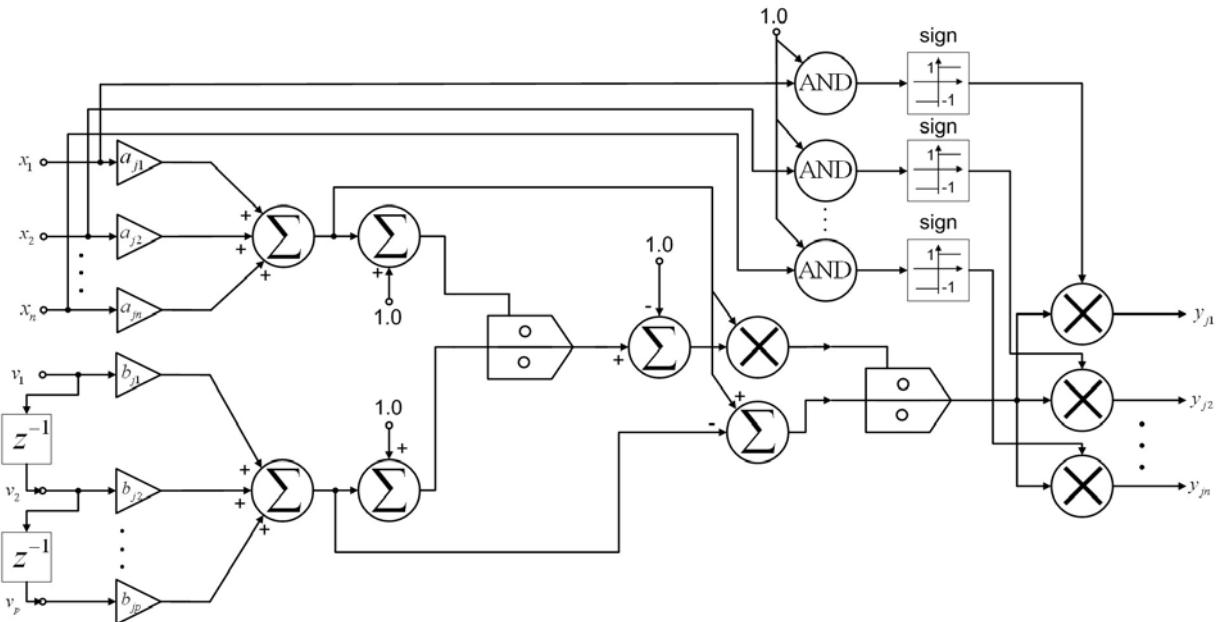


Рисунок 2.19 – Модифицированная структура нейрона Фукушимы

Вторая структура для предобработки сигнала была построена на основе квадратичного нейрона (рисунок 2.20). Квадратичный нейрон вычисляет функцию [71]:

$$y_j = \theta_j + \sum_{i=1}^n w_{ji} x_i + \sum_{p=1}^n \sum_{l=1}^n w_{jpl} x_p x_l, \quad (2.27)$$

где θ_j - пороговое значение данного нейрона, w - значения весовых коэффициентов, x - компоненты входного сигнала.

Удобство применения данной вычислительной структуры обусловлено простотой реализации. На основе стандартных арифметических блоков - умножителей и сумматоров - производится построение данного нейрона. Схема блока предобработки сигналов для второго эксперта, представленного радиально-базисной нейронной сетью, приведена на рисунке 2.21.

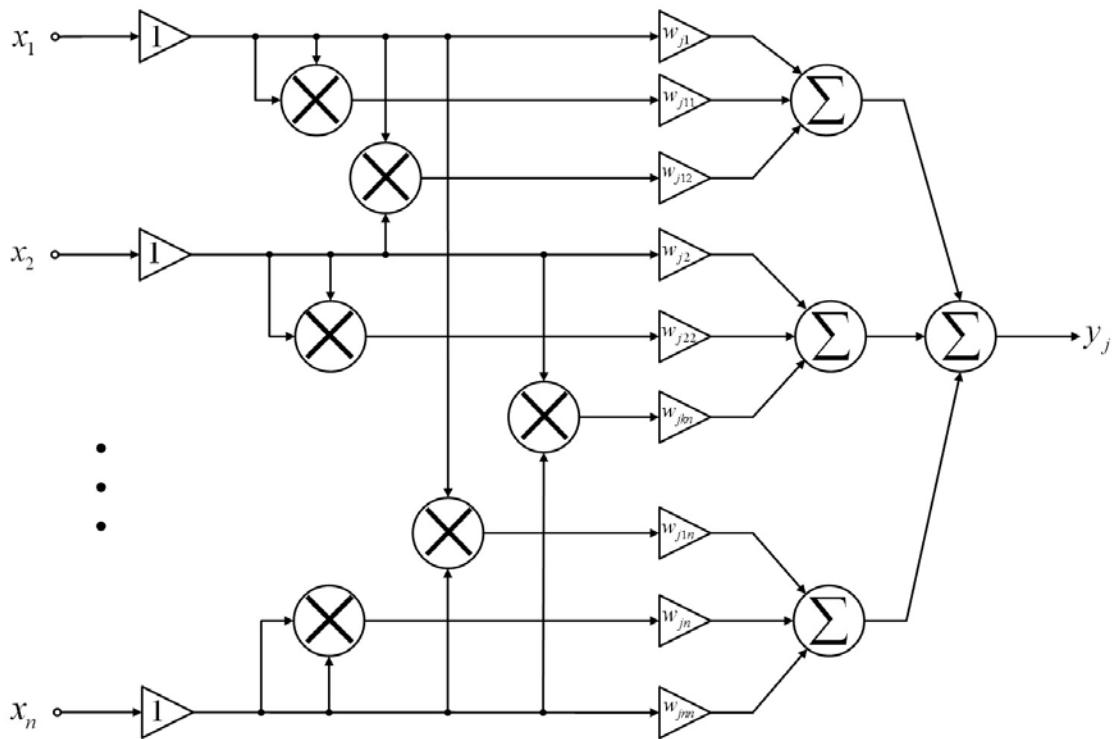


Рисунок 2.20 – Структура квадратичного нейрона

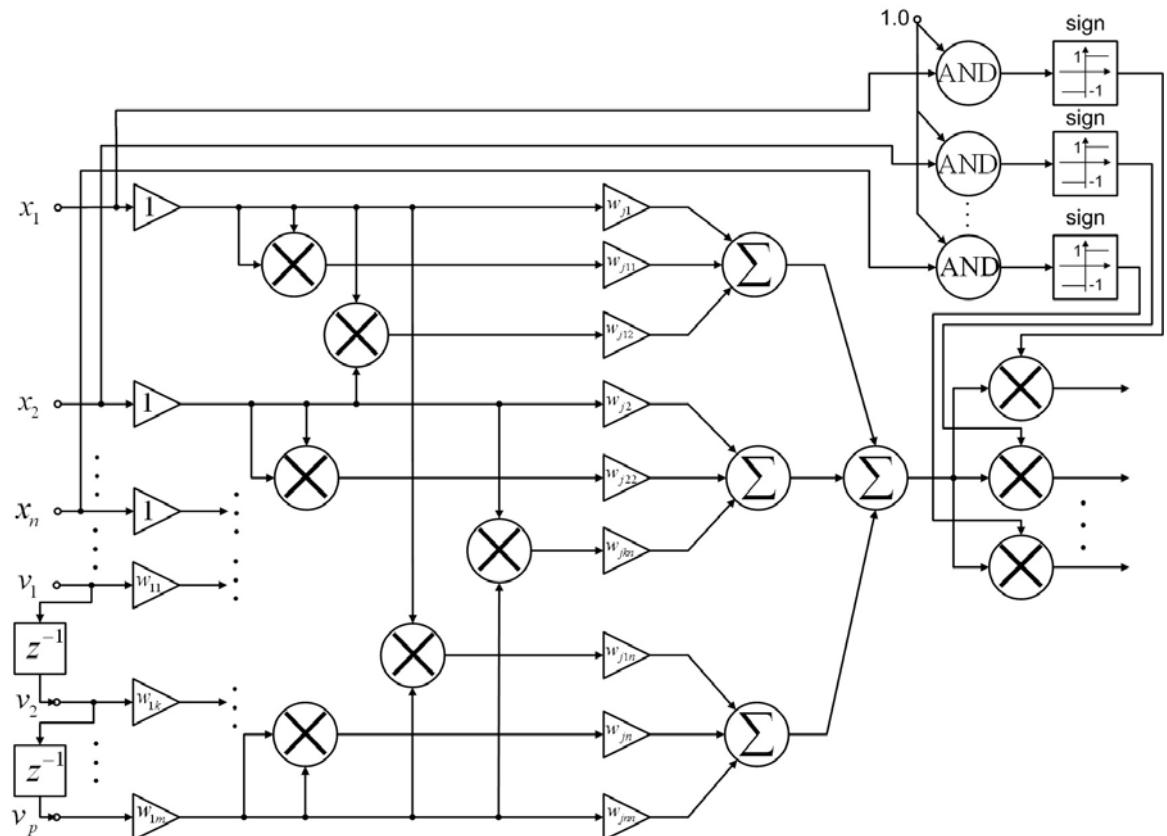


Рисунок 2.21 – Модифицированная структура квадратичного нейрона

Структура на основе квадратичного нейрона работает по тому же принципу, что и модифицированный нейрон Фукушимы. В данном случае амплитуда выходного сигнала может варьироваться в интервале от -1 до 1. Использование

различных структур для коррекции сигнала и его последующего объединения финальным экспертом с различными возможными значениями максимальной и минимальной амплитуды позволяют корректировать выходной сигнал с нейросетевых экспертов различными способами, что не позволяет финальному эксперту отдавать предпочтение тому или иному решению группы сетей только на основе алгоритма предобработки сигналов. Использование квадратичных зависимостей в структурах по предобработке сигналов от НС позволяет избежать ситуации, когда функционирование финального эксперта будет сводиться к простому линейному преобразованию входных сигналов, т.е. к алгоритмической композиции, в результате чего финальный эксперт может утратить способность к обобщению.

Структура для предобработки сигнала от третьего эксперта строится на основе N -адалины и представлена на рисунке 2.22. Она представляет собой адаптивный линейный ассоциатор с двумя входами и нелинейным препроцессором, который состоит из трёх стандартных блоков умножения [81]. В результате данная структура вычисляет квадратичную комбинацию:

$$y = w_0 + w_1 x_1 + w_2 x_i^2 + w_3 x_i x_p + w_4 x_p^2 + w_5 x_p \quad (2.28)$$

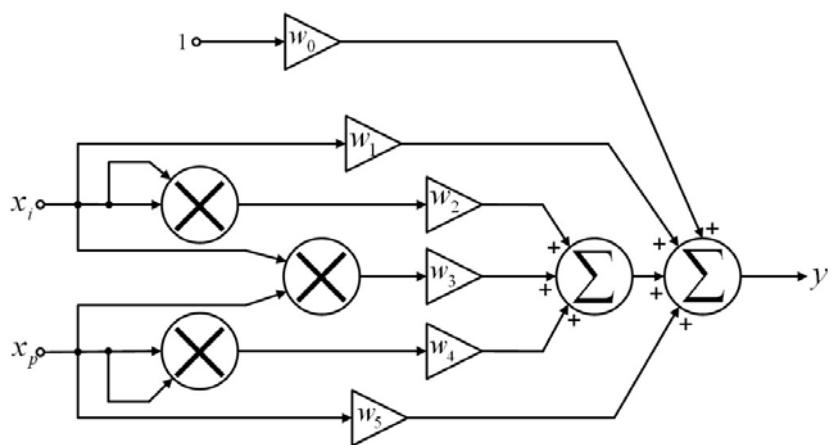


Рисунок 2.22 – N - адалина

Данный нейрон был модифицирован элементами задержки во входном слое и элементами, обеспечивающими корректный вывод выходного сигнала (блоки AND и sign), а также выходной структурой, реализующей функцию

гиперболического тангенса, для приведения выходных значений нейрона в интервал от -1 до 1. Структурная схема нейрона показана на рисунке 2.23.

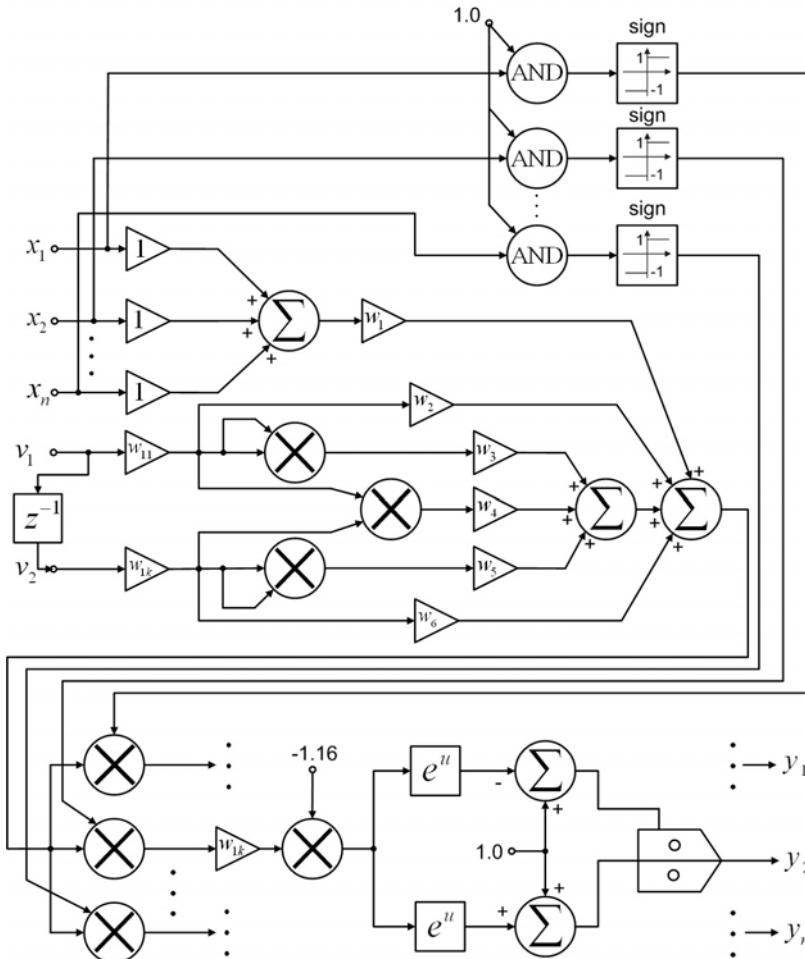


Рисунок 2.23 – Модифицированная структура N - адалины

Наиболее важным узлом для систем, основанных на применении нейросетевых экспертов, является механизм объединения мнений экспертов. Как было описано выше, в данном случае для этой цели используются модифицированные нейроны. Для повышения эффективности функционирования системы объединения мнений экспертов, сигналы от модифицированных нейронов Хопфилда, N – адалины и квадратичного нейрона подавались на систему латеральных связей, обеспечивающую конкуренцию между ними. Этот механизм позволяет акцентировать внимание финального эксперта на приоритетных решениях. Структура такой системы приведена на рисунке 2.24.

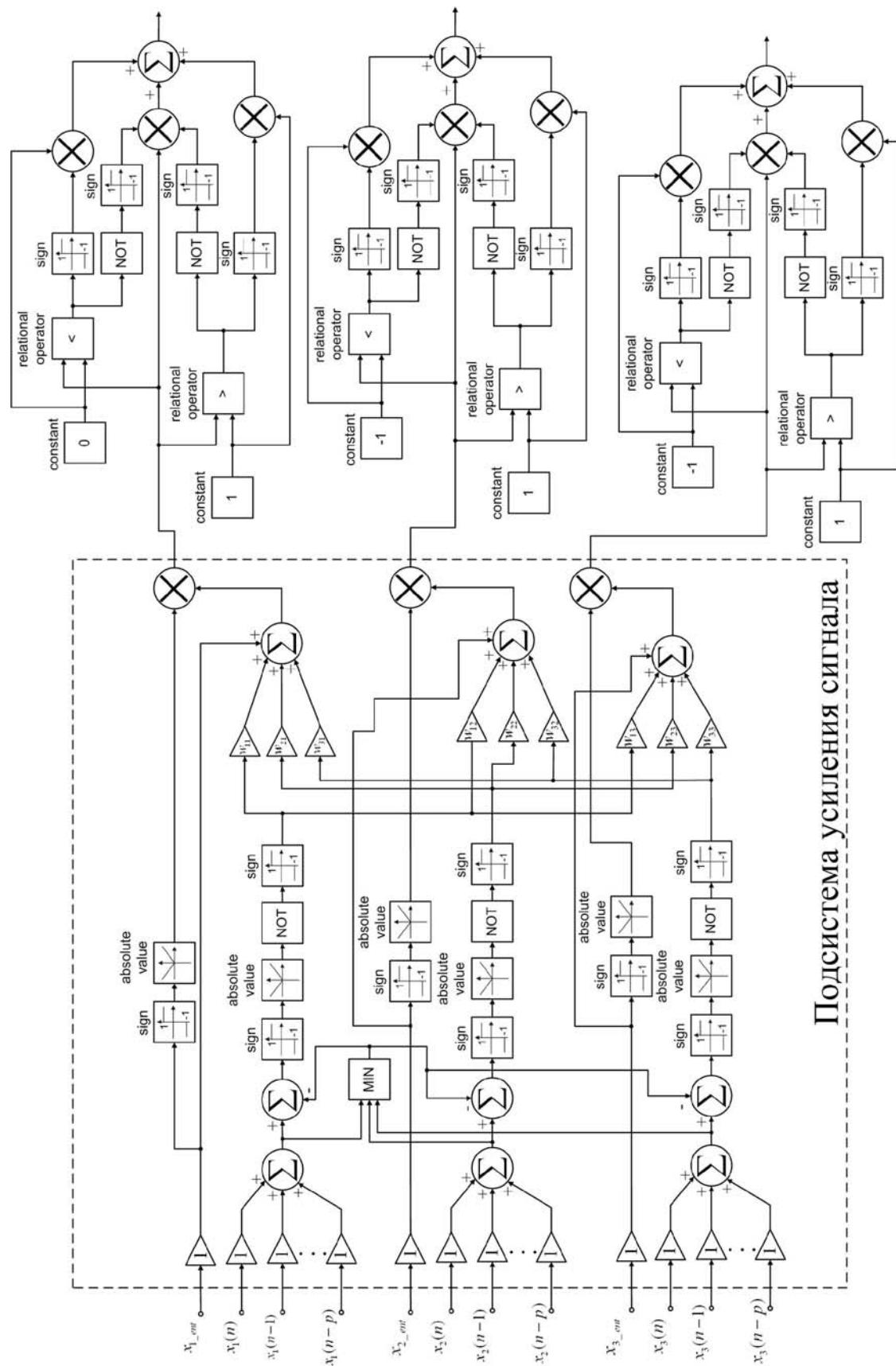


Рисунок 2.24 – Схема постнейронных латеральных связей

Данная система состоит из двух блоков. В первом блоке производится поиск минимальной суммы коэффициентов, формируемых нечёткой системой

оценки качества обучения экспертов. Чем меньше эта сумма, тем лучше сформирована нейронная сеть для решения поставленной задачи. К выходному сигналу лучшего эксперта прибавляется величина, равная половине амплитуды максимально возможного значения сигнала (0,5), значения остальных сигналов уменьшаются на эту же величину. В результате данного преобразования могут получаться амплитуды сигналов, превышающие по модулю единицу. Чтобы этого избежать, после системы усиления сигнала располагается блок ограничения амплитуды сигнала до заданного диапазона.

После предобработки сигналов с помощью структур, приведённых выше, финальный эксперт получает не единичные значения «выигравших нейронов» (все эксперты строятся по принципу «победитель забирает всё»), а изменённые значения амплитуд выходных сигналов экспертов в соответствии с качеством процесса обучения. По значению амплитуды финальный эксперт может определить, какой из них лучше всего приспособлен для решения поставленной задачи. В результате предпочтение будет дано эксперту, который лучше всего обучен для решения данной задачи.

В структуру каждого модифицированного нейрона для предобработки информации включены настраиваемые весовые коэффициенты. В результате применения этих параметров совокупность из этих трёх нейронов образует обучаемую структуру, которую можно настроить на определённый алгоритм обработки сигнала, а не на простое линейное преобразование. Реализация градиентного алгоритма обучения для такой структуры требует больших вычислительных затрат, т.к. нейроны, входящие в неё, обладают различными характеристиками, поэтому здесь целесообразно использовать алгоритмы случайного поиска, базирующиеся на принципах эволюции. Рассмотрим алгоритм настройки системы предобработки сигналов [31]:

1. Сгенерировать допустимые значения весовых коэффициентов полученной системы, рассматривать эти значения, как многомерную точку U_1, \dots, U_N , где N – количество оптимизируемых параметров.

2. Вычислить степень приспособленности каждой точки, путём вычисления пригодности системы предобработки сигналов с заданными предполагаемыми значениями весовых коэффициентов.

3. Каждая сгенерированная точка даёт «потомство», внося случайные изменения в компоненты весовых коэффициентов, по следующему алгоритму:

$$U_{ji} = U_{ji} + \xi_{ji} \quad (i = 1, \dots, k_j), \quad (2.29)$$

где k_j – число потомков j -й особи, ξ_{ji} – случайное отклонение, которое соответствует изменению i -й компоненты в j -м потомке.

4. Среди всего множества полученных решений производится отбор, в котором с большей вероятностью удаляются решения с неудовлетворительной степенью приспособленности. Вероятность удаления решения пропорциональна значению минимизируемой функции приспособленности Q :

$$p_{ij} = \frac{Q(U_{ij})}{\sum_{i,j} Q(U_{ij})} \quad (2.30)$$

Процесс уничтожения решений продолжается до тех пор, пока количество решений не достигнет первоначального.

5. Если достигнуты требуемые критерии обучения, то переход к шагу 6, в противном случае перейти к шагу 2.

После настройки компонентов предобработки сигналов от экспертов производится генерация финального решения с помощью сигма-пи нейронной сети. Выбор данной нейронной сети в качестве финального эксперта базируется на сочетании в сигма-пи сети положительных качеств многослойный нейронных сетей (персепtronов) и радиально-базисных нейронных сетей. На рисунке 2.25 приведена схема сигма-пи сети: $\psi(\bullet)$ – сигмоидальные функции активации; $\varphi(\bullet)$ – радиально-базисные функции активации; a , b , w – настраиваемые весовые коэффициенты.

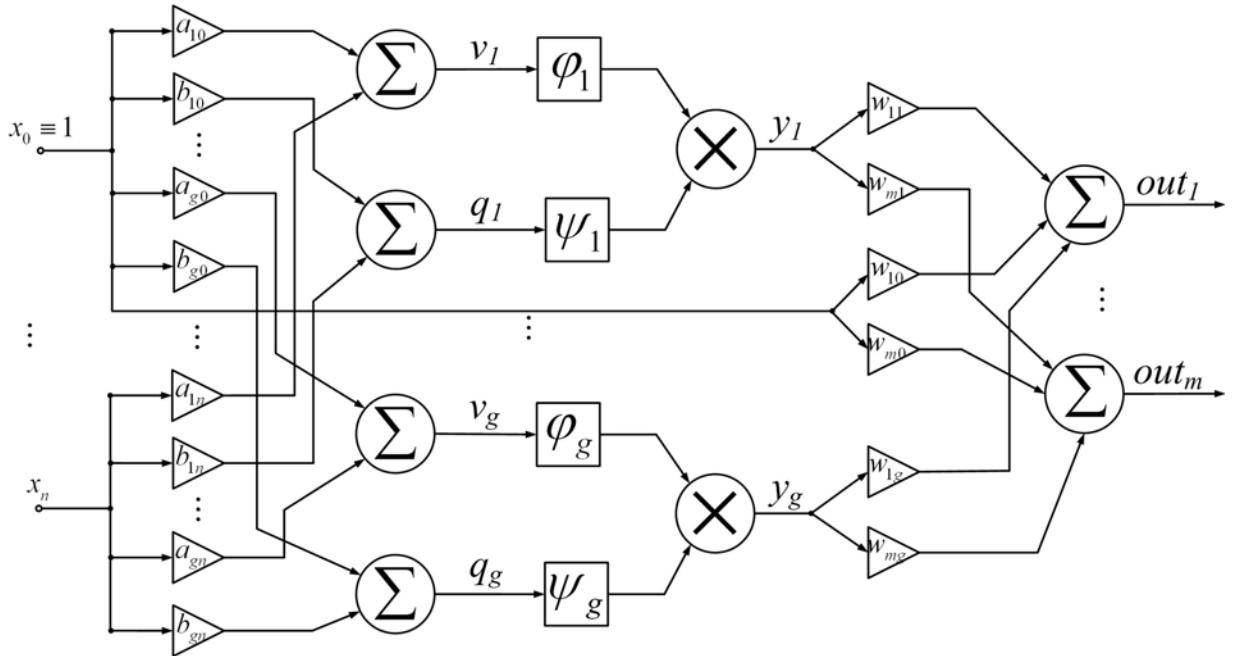


Рисунок 2.25 – Структурный график сигма-пи сети

Исходя из того, что все нейросетевые эксперты обучаются по принципу «победитель забирает всё», количество решений, которое может принимать нейронная сеть, соответствует числу нейронов в выходном слое. В результате получаем, что количество входов сигма-пи сети рано суммарному количеству выходов всех нейросетевых экспертов. Число выходов сигма-пи сети определяется количеством возможных принимаемых решений.

Как видно из структурного графа сигма-пи сети, для настройки данной системы требуется оптимизация большого количества параметров: групп весов для сигмоидальных и радиально-базисных частей нейронов, параметров функций активации, весовых коэффициентов выходного слоя. Обучение рассматриваемой сети можно осуществить путём задания целевой функции в виде:

$$E(t) = \frac{1}{2} e^2(t) = \frac{1}{2} (d(t) - y(t))^2, \quad (2.31)$$

где $d(t)$ – вектор ожидаемых значений, $y(t)$ – вектор полученных значений, t – отсчёт времени работы системы.

Далее получаем систему дифференциальных уравнений, описывающих процедуру настройки весов. Сигма-пи сеть в данной работе представляет собой структуру, ответственную за принятие финального решения ассоциативной машины, и её структура может значительно варьироваться в зависимости от

решаемой задачи. При большой размерности входного пространства вычисление коэффициентов коррекции настраиваемых параметров на основе градиентных методов может занимать длительное время. В данной работе предлагается выполнить обучение сигма-пи сети на основе комбинации алгоритмов случайного поиска, применяемых в определённом порядке и для настройки конкретных параметров сети. Алгоритм обучения сигма-пи сети представим следующим образом:

1. Первый этап обучения начинается с инициализации настраиваемых параметров случайными значениями из интервала $(0, 1)$.
2. Сигма-пи сеть обладает сложной внутренней структурой, поэтому алгоритм обучения разобьем на следующие этапы: настройка весовых коэффициентов a и b , затем адаптация весов выходного слоя w , после настройки переменных параметров весов, выполняется оптимизация параметров функций активации.
3. Первый этап оптимизации начинается с фиксирования всех параметров кроме весовых коэффициентов a и b . Из всех значений весовых коэффициентов a и b составляется два вектора, которые подвергаются оптимизации с целью минимизации критерия ошибки E (2.31). Для адаптации данных параметров применялся метод оптимизации с парными пробами [94]. Данный алгоритм в процессе своей работы генерирует два вектора $a(k) \pm \mu r(k)$ (аналогично и для вектора b : $b(k) \pm \mu r(k)$), где $r(k)$ – случайный вектор, определяющий направление поиска, k – номер итерации, μ - коэффициент, определяющий скорость развития алгоритма. После вычисления проб рабочий шаг алгоритма делается в направлении наименьшего значения полученной ошибки:

$$\begin{aligned} a(k+1) &= a(k) - \eta r(k) \operatorname{sign}(E(a(k) + \mu r(k)) - E(a(k) - \mu r(k))), \\ b(k+1) &= b(k) - \eta r(k) \operatorname{sign}(E(b(k) + \mu r(k)) - E(b(k) - \mu r(k))), \end{aligned} \quad (2.32)$$

где η - коэффициент скорости обучения, $\operatorname{sign}()$ – функция, возвращающая знак своего аргумента. Характерной особенностью данного алгоритма является тенденция к поиску нового решения, даже в том случае, когда приемлемое

решение найдено. Это свойство может быть полезно при обучении нейронной сети сложной архитектуры, т.к. алгоритм не остановит свою работу в случае попадания в локальный минимум.

4. На втором этапе производится настройка весовых коэффициентов выходного слоя с помощью алгоритма пересчёта параметров при неудачном шаге [100]. Из всех значений весовых коэффициентов w формируется вектор, который подвергается оптимизации. Алгоритм начинает работу с продвижения в случайному направлению. Шаг возврата при неудачной пробе не производится, вместо этого алгоритм делает новую попытку найти приемлемое направление поиска, но при этом компенсирует неудачную предыдущую попытку:

$$w(k+1) = w(k) + \begin{cases} \eta r(k), & E(k+1) < E(k) \\ -\eta r(k-1) + \eta r(k), & E(k+1) \geq E(k) \end{cases}. \quad (2.33)$$

Основное преимущество алгоритма – высокое быстродействие.

5. В данной работе в качестве функций активации сигмоидальных нейронов и радиально базисных были выбраны следующие:

$$\psi(x) = \frac{1}{1 + e^{-\gamma x}}; \varphi(x) = (\|x - c\|^2 + \sigma^2)^{\frac{1}{2}}, \quad (2.34)$$

где c - центр радиально базисной функции, σ - параметр ширины. В результате необходимо оптимизировать три параметра каждого сигма-пи нейрона: γ, c, σ . Из параметров одного типа формируется вектор для оптимизации и выполняется настройка НС с помощью алгоритма наилучшей пробы [94]. Этот алгоритм базируется на многократной случайной выборке. Из исходного вектора параметров генерируется h случайных векторов для каждого оптимизируемого параметра:

$$\begin{aligned} &c + \eta r^1(k), c + \eta r^2(k), \dots, c + \eta r^h(k); \\ &\sigma + \eta r^1(k), \sigma + \eta r^2(k), \dots, \sigma + \eta r^h(k); \\ &\gamma + \eta r^1(k), \gamma + \eta r^2(k), \dots, \gamma + \eta r^h(k). \end{aligned} \quad (2.35)$$

После выполнения расчёта ошибки обучения принимается то решение, которое показало наименьшую ошибку.

6. В результате выполнения данной последовательности этапов проводим обучение сети до заданной минимальной ошибки обучения.

Алгоритм обучения рассмотренной НС базируется на вычислении оптимального соотношения внутренних параметров нейронной сети – весовых коэффициентов и параметров функций активации – путём подачи на вход сети некоторого входного вектора и фиксации полученного значения. На основе ошибки, которая возникает при несоответствии ожидаемых выходных сигналов и фактически получаемых, строится алгоритм обучения. Рассмотренный алгоритм обучения имеет определённую стратегию формирования нового решения, в результате чего он может сосредоточиться на определённой части поискового пространства и не обнаружить комбинацию параметров, которая в данной задаче является наилучшей. Для перемещения алгоритма в другую точку пространства спроектируем модель, описывающую систему взаимодействия некоторого множества геометрических примитивов, для перемещения алгоритма поиска в разные части поискового пространства.

Алгоритм для перевода поиска в другую точку пространства базируется на процедуре изменения состояния следующей геометрической модели:

1. В трёхмерном пространстве выполняется построение группы сфер (в данной задаче 24 сферы), центры которых расположены в случайном порядке. При выборе центра сферы значения координат по оси абсцисс и ординат могут быть выбраны произвольно из заданного интервала (в данной работе ограничительный интервал для координаты x – от -17 до 19, для y – от -18 до 18), а значение координаты по оси аппликат равняется нулю. Сфера в пространстве расположены так, что их оси вращения параллельны оси аппликат.

2. Выполняется построение графиков функций двух переменных:

$$\begin{aligned} F_1(x, y) &= (\cos(\alpha \cdot x + (1 - \alpha) \cdot y))^2; \\ F_2(x, y) &= \cos(2 + (1 - \beta) \cdot x - \beta \cdot y). \end{aligned} \quad (2.36)$$

Построение графиков необходимо выполнить таким образом, чтобы все построенные сферы оказались между двумя поверхностями этих двух функций:

$$\begin{aligned} F_1(x, y) &= (\cos(\alpha \cdot x + (1 - \alpha) \cdot y))^2 - 10; \\ F_2(x, y) &= 10 + \cos(2 + (1 - \beta) \cdot x - \beta \cdot y). \end{aligned} \quad (2.37)$$

После преобразования уравнений (2.36) в (2.37) поверхность функции F_1 будет находиться на уровне $z = -10$, а поверхность функции F_2 на уровне $z = 10$. Параметры α и β определяют поведение функции, и от их значений зависит процесс оптимизации. Значения данных величин инициализируются нейросетевым ГСЧ (раздел 2.6). В процессе работы алгоритма перевода поиска в другую точку пространства производится изменение этих параметров, что приводит к управляемому процессу поиска предполагаемых решений.

3. В каждой сфере на уровне большого круга, проведённого перпендикулярно оси аппликат, выделяют 9 точек: (x_c, y_c) , $(x_c + r, y_c)$, $(x_c - r, y_c)$, $(x_c, y_c + r)$, $(x_c, y_c - r)$, $(x_c + r/2, y_c)$, $(x_c - r/2, y_c)$, $(x_c, y_c + r/2)$, $(x_c, y_c - r/2)$, x_c – значение координаты центра по оси абсцисс, y_c – значение координаты центра по оси ординат, r – радиус сферы.

4. На верхнюю полусферу каждой сферы «оказывает влияние» функция F_2 , а на нижнюю F_1 . Степень данного влияния вычисляется следующим образом для каждой сферы с координатами центра в точке (x_c, y_c) :

$$Infl_F_2 = F_2(x_c, y_c) + F_2(x_c + r, y_c) + F_2(x_c - r, y_c) + F_2(x_c, y_c + r) + F_2(x_c, y_c - r) + F_2(x_c + r/2, y_c) + F_2(x_c - r/2, y_c) + F_2(x_c, y_c + r/2) + F_2(x_c, y_c - r/2).$$

$$Infl_F_1 = F_1(x_c, y_c) + F_1(x_c + r, y_c) + F_1(x_c - r, y_c) + F_1(x_c, y_c + r) + F_1(x_c, y_c - r) + F_1(x_c + r/2, y_c) + F_1(x_c - r/2, y_c) + F_1(x_c, y_c + r/2) + F_1(x_c, y_c - r/2).$$

Величина $Infl_F_2$ характеризует степень влияние функции F_2 на верхнюю полусферу каждой сферы, $Infl_F_1$ – степень влияния F_1 на нижнюю полусферу каждой сферы.

5. В результате взаимодействия сферы и двух поверхностей функций, осуществляется поворот каждой сферы на определенный угол относительно оси абсцисс. Угол поворота вычисляется следующим образом:

$$\Theta = \frac{\pi}{2} \cdot \frac{abs(Infl_F_1 + Infl_F_2)}{mx}, \quad (2.38)$$

где mx – максимальное значение функции $abs(Inf_F_1 + Inf_F_2)$ при установленных параметрах α и β . Для вычисления этого значения после определения параметров функций Inf_F_2 и Inf_F_1 необходимо провести исследование данных функций на заданных интервалах по оси абсцисс и ординат с целью выявления максимального значения модуля их суммы.

6. Поворот каждой сферы возможен на угол $\pi/2$ радиан. Для возможности перехода к новым точкам в поисковом пространстве необходимо сопоставить каждый угол возможного поворота сферы от 0 до $\pi/2$ радиан числу от 0 до 255.

Значение угла порота сферы делится на константу равную $0,006135923 \left(\frac{\pi}{2} \cdot \frac{1}{256}\right)$,

дробная часть результата отбрасывается.

7. В результате работы алгоритма получаем 24 байта, характеризующие текущее состояние построенной геометрической модели. После проведения процесса нормализации все 24 полученных числа будут лежать в интервале от 0 до 1. Далее случайно выбираем 24 параметра сигма-пи сети и производим замену параметров на сгенерированные. Если требуется модификация большого числа параметров, то необходимо увеличить количество сфер в модели.

Разработанный алгоритм обучения позволяет выполнить настройку финального нейросетевого эксперта и на основе результатов, полученных от нейронных сетей, синтезировать ответ ассоциативной машины.

2.6 Разработка алгоритма обеспечения безопасности системы

В работе применяются алгоритмы, для нормального функционирования которых необходимо постоянно формировать последовательности псевдослучайных чисел: в алгоритмах оптимизации, базирующихся на применении методов эволюционного моделирования, таких, как генетический и иммунный алгоритм оптимизации, необходимо выбирать точки кроссинговера и мутации для получения нового решения, в градиентных алгоритмах, применяемых для настройки нейронных сетей, требуется выбирать начальные состояния весовых коэффициентов, порогов, параметры функций активации,

псевдослучайные числа применяются для генерации начальных состояний систем, для определения случайного направления поиска и т.д. Ансамбль нейронных сетей, используемый в данной работе, содержит в себе множество параметров, состояния которых перед началом работы системы желательно определять случайным образом. Это необходимо для обеспечения безопасности самой системы, т.к. необходимо, чтобы в произвольные моменты времени комитет из множества экспертов, представленный нейронными сетями, находился в случайном состоянии, которое трудно спрогнозировать. Такие методы генерации псевдослучайных чисел, как линейные конгруэнтные генераторы, сдвиговые регистры с линейной обратной связью, аддитивный генератор чисел имеют линейную алгебраическую структуру, что позволяет раскрыть механизм генерации чисел. Существует возможность вычисления порождающего полинома и его начальных состояний, который генерирует числа данной последовательности, на основе информации, получаемой при наблюдении генерируемых чисел. Это может быть осуществлено путём решения системы линейных уравнений, которые полностью определены после наблюдения $(2n-1)$ битов последовательности, где n – число неизвестных. Для исключения проблемы линейности необходимо использовать нелинейные методы генерации случайных чисел с надёжным источником энтропии. Рассмотрим структурную схему разработанного в данной работе генератора случайных чисел.

Информационная энтропия — мера хаотичности информации [114]. Источники энтропии используются для накопления энтропии и получения из неё начального значения, которое будет использовано генератором случайных чисел для формирования чисел. Создание надёжного источника энтропии позволит генератору производить последовательность близкую к случайной, т.к. устраняется недостаток генератора псевдослучайных чисел, использующего один раз для процесса инициализации случайную величину. Если использовать для генератора качественные случайные величины, то выдаваемые им последовательности будут приближаться к случайным. Источники энтропии зависят от реализации [50]. При накоплении необходимого количества энтропии,

его можно использовать для создания начального значения. В основе физического источника энтропии могут лежать различные процессы: тепловой шум (или шум Джонсона), источник радиоактивного распада, генератор свободных колебаний (рисунок 2.26).

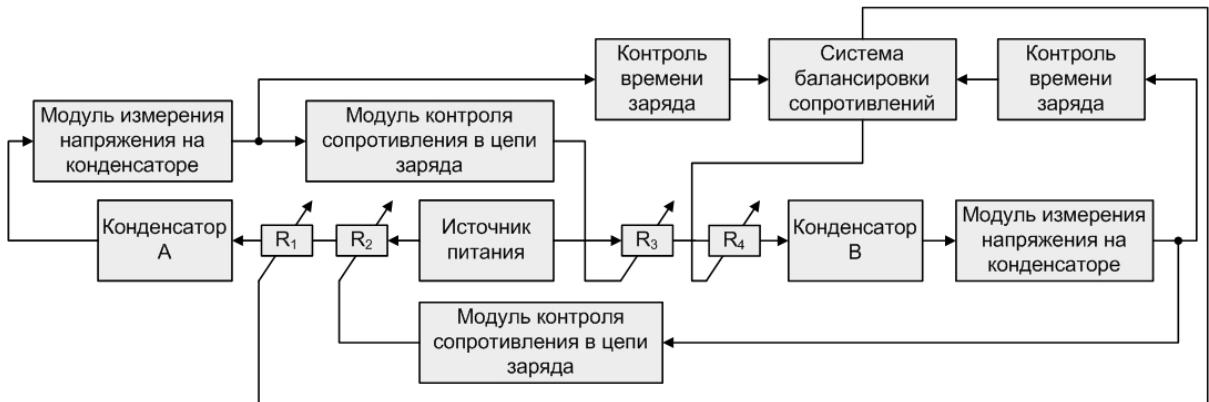


Рисунок 2.26 – Структурная схема источника энтропии

В данной работе была разработана электрическая схема, выступающая в качестве источника энтропии, которая была основана на создании гонок в процессе перезаряда двух ёмкостей. Основу данного источника составили две системы: первая система (2 модуля контроля сопротивления в цепи заряда) уменьшала сопротивление в цепи заряда первой ёмкости при увеличении заряда второй ёмкости, что приводило к увеличению скорости заряда первой ёмкости. И наоборот: производилось уменьшение сопротивления в цепи заряда второй ёмкости при увеличении заряда первой ёмкости, что приводило к увеличению скорости заряда второй ёмкости. Таким образом, возникает «соревнование» между ёмкостями в процессе заряда. Вторая система (система балансировки сопротивлений) выполняла функцию компенсации сопротивления в цепи заряда каждой ёмкости, с целью свести разницу во времени заряда двух ёмкостей до максимального напряжения к минимуму. В результате этих попыток двух систем создать баланс во всей системе перезаряда ёмкостей возникает стохастический процесс, который может быть положен в основу работы источника энтропии.

Рассмотрим назначение отдельных блоков:

1. Модуль контроля сопротивления в цепи заряда. Данный модуль производит изменение сопротивления в цепи заряда в зависимости от напряжения на противоположном конденсаторе. В качестве основного элемента здесь может

быть использован полевой транзистор с изолированным затвором и N – каналом ($2N6660$), вывод затвора которого подключен к положительному выводу электролитического конденсатора большой ёмкости (2200 мкФ). Сток транзистора подключен к катоду светодиода оптопары ($4N35$). В результате заряда конденсатора напряжение между истоком транзистора и затвором будет расти, что приводит к росту напряжения на фотодиоде и постепенному открыванию фототранзистора оптопары. Фототранзистор включен параллельно с базовым сопротивлением (50 кОм) в цепь заряда противоположного конденсатора. При повышении напряжении на конденсаторе A происходит уменьшение сопротивления в цепи заряда конденсатора B и наоборот. В результате функционирования системы создаются «гонки» между ёмкостями.

2. Система балансировки сопротивлений. Если назначение предыдущего блока создать неопределенность в процессе заряда конденсаторов, то данный блок предназначен для выравнивания сопротивления в цепях заряда конденсаторов с целью минимизировать значение разницы времени заряда двух ёмкостей. Основу данного блока составляет схема подсчёта интервала времени необходимого для заряда конденсатора до определённого уровня напряжения. Данный уровень определяется уровнем напряжения, при котором происходит переключение триггера Шмита ($SN74HC14$) [112]. Система фиксирует моменты времени переключения и получает время заряда каждого конденсатора. Для отсчёта моментов времени необходимо использовать осциллятор со стабильной частотой колебаний. В данной работе был применён кварцевый генератор ($ECS-300CX-184$) колебания которого отсчитывались двумя 14 разрядными счётчиками. При частоте колебаний 1,152 МГц. Счётчики могут производить отсчёт до полного обнуления в течение 233 секунд. После вычисления разности интервалов времени производится увеличение сопротивления в цепи опережающего конденсатора пропорционально разности интервалов заряда с помощью потенциометра ($AD5160BRJZ100-R2$).

Приведённая схема позволяет сформировать вектора, необходимые для настройки генератора случайных чисел (ГСЧ). В данной работе в качестве такого

генератора была использована структура, состоящая из 24 нейронных сетей Хопфилда, функционирующих в нестандартном режиме (обоснование именно такого количества сетей будет дано ниже). В качестве основного элемента ГСЧ применяется нейронная сеть Хопфилда, содержащая 9 нейронных элементов. Схема организации нейросетевого ГСЧ приведена на рисунке 2.27.

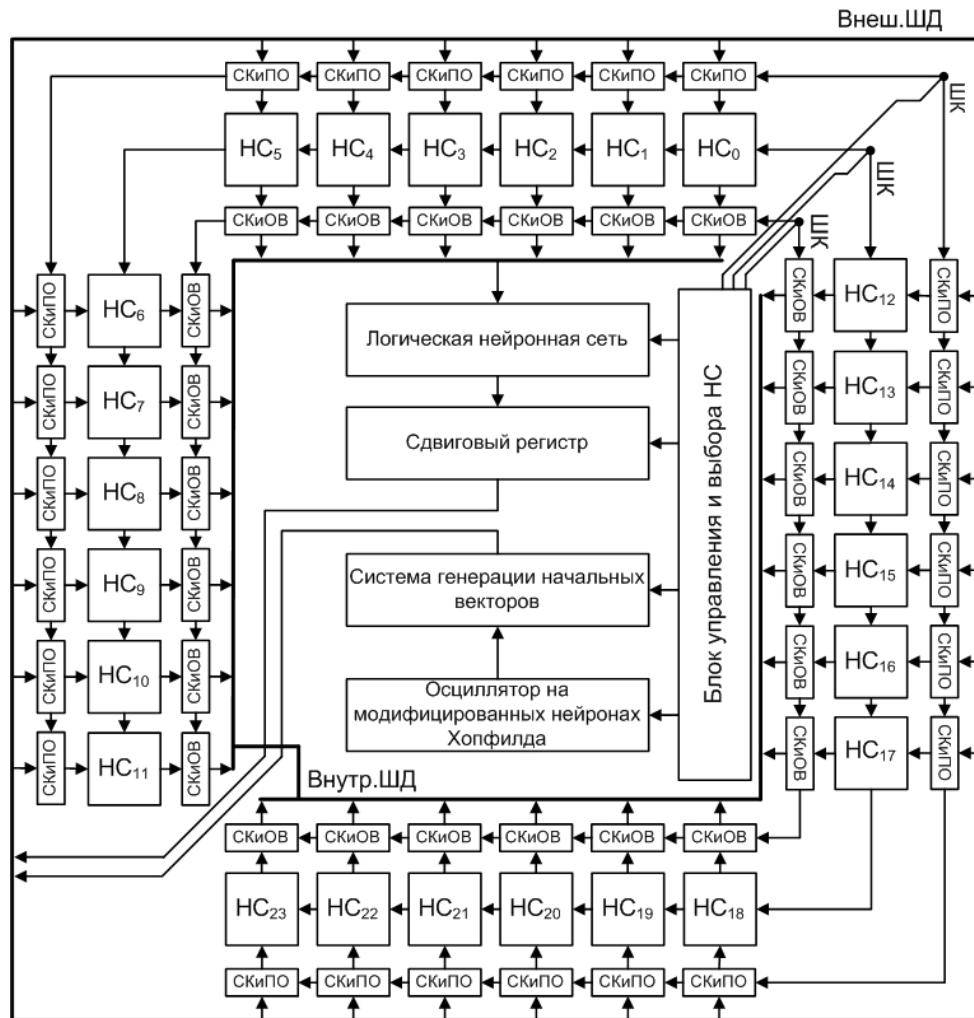


Рисунок 2.27 – Нейросетевой генератор случайных чисел

Данный нейросетевой модуль состоит из следующих подсистем:

1. Блок управления и выбора НС, осуществляет управление всеми структурами, путём передачи команд по шине команд (ШК). Одна из основных функций - это определение очерёдности активации нейронных сетей и нейронных элементов в каждой сети при асинхронном извлечении информации.

2. Осциллятор на модифицированных нейронах Хопфилда.

Колебательный процесс в данной структуре является источником случайных битовых последовательностей, которые применяются в **блоке генерации**

начальных векторов для определения начального состояния функционирования всего нейросетевого ГСЧ. На основе получаемой последовательности определяется очередность активации сетей Хопфилда и нейронов в их внутренней структуре.

3. Логическая нейронная сеть. Выполняет обработку данных, полученных от сетей Хопфилда. Осуществляя логическое преобразования над входными битовыми последовательностями, уничтожает зависимости в случайных числах, получаемых от нейросетевого ГСЧ, что не позволит злоумышленнику вскрыть алгоритм генерации чисел.

4. Блок сдвигового регистра. Формирует выходное значение нейросетевого ГСЧ.

5. Блоки СКиПО (Система Коммутации и Предобработки Данных). Выполняет подключение нейронной сети, определённой блоком управления и выбора НС, к внешней шине данных (Внеш. ШД), для передачи её начального входного вектора. Алгоритм предобработки заключается в выполнении побитовой логической операции «исключающее ИЛИ» над битами входного вектора и вектором, полученным от осциллятора.

6. Блок СКиOB (Система Коммутации и Организации Вывода). Выполняет подключение нейронной сети, определённой блоком управления и выбора НС, к внутренней шине данных (Внутр. ШД).

Алгоритм работы нейросетевого ГСЧ можно представить следующим образом:

- На основе схемы генерации начальных векторов производится по 25 векторов для каждой нейронной сети. На основе этих 9 битовых векторов производится обучение нейронных сетей.

- Блок управления и выбора НС (БУиВНС) даёт команду осциллятору на генерацию последовательности из 5 бит. Если после перевода в десятичную систему получено значение большее 24, то попытка повторяется до получения числа от 1 до 24. Это и будет номер первой активизированной сети Хопфилда.

3. При первом запуске ГСЧ необходимо сгенерировать вектор начального состояния, подаваемый на нейронную сеть. Так как НС Хопфилда содержат по 9 нейронных элементов, БУиВНС отдаёт команду осциллятору сгенерировать 9 бит информации.

4. Производится подключение выбранной НС к внутренней и внешнейшине данных с помощью блоков СКиПО и СКиВО.

5. Перед подачей вектора начального состояния на вход сети, выполняется побитовая логическая операция «исключающего ИЛИ» над значением вектора начального состояния и 9 мерным вектором, полученным от осциллятора.

6. Выполняется подача вектора начального состояния на вход выбранной НС Хопфилда. Выполняется процесс активации нейронных элементов в порядке, определяемом осциллятором на основе нейронов Хопфилда.

7. Полученный вектор от НС подаётся на логическую нейронную сеть и формируется один бит выходной информации.

8. Подача входного вектора осуществляется девять раз, для получения выходного значения (9 бит). В качестве вектора начального состояния используется первоначальный вектор, полученный от системы генерации начальных векторов, но в логической операции «исключающее ИЛИ» производится генерация нового значения второго аргумента, получаемого от осциллятора.

9. В результате в сдвиговом регистре за 9 тактов работы формируется одно из значений случайной последовательности.

10. Получение следующего числа случайной последовательности, производится аналогично, начиная со второго шага. Но вместо начального вектора, полученного от системы генерации начальных векторов, используется предыдущее значение уже полученной случайной последовательности.

11. Алгоритм продолжает работу до получения последовательности из 3000 чисел, после чего переход к шагу 1.

Далее рассмотрим подробное описание процесса функционирования для каждого блока.

Чтобы сеть Хопфилда могла функционировать как ГСЧ, был изменён алгоритм её обучения, причём структура сети остаётся стандартной. Алгоритм функционирования такой сети состоит из двух фаз: фазы сохранения и фазы извлечения. Рассмотрим модификацию каждой фазы.

Фаза сохранения. Если в памяти сети Хопфилда необходимо сохранить определённый набор векторов (ячеек фундаментальной памяти), то для настройки весовых коэффициентов необходимо применить правило Хебба:

$$w_{ij} = \sum_{k=1}^K x_i^k x_j^k, \quad (2.39)$$

где w_{ij} – синаптический вес, направленный от нейрона i к нейрону j , K – количество запоминаемых образов, x_i^k – i -й элемент фундаментальной памяти x^k . Для нормальной работы сети Хопфилда поддерживается условие $w_{ii} = 0$ для всех значений i . При таком процессе обучения нейронная сеть может запомнить $0,14 \cdot N$ различных образов [15], где N – размер сети. Если архитектура НС Хопфилда содержит 9 нейронов, то количество максимально возможных образов, которые может запомнить сеть будет равно одному. В данной работе НС Хопфилда предлагалось запомнить $25 N$ – мерных векторов, полученных от источника энтропии. Недостаток сети Хопфилда, заключающийся в небольшом объёме памяти и появлении ложных состояний (устойчивых состояний, отличных от ячеек фундаментальной памяти), можно использовать для генерации случайных чисел. Но для этого необходимо «переобучить» сеть на множество примеров, которая она не в состоянии сохранить.

Фаза извлечения. Данный этап направлен непосредственно на формирование случайного бинарного вектора. Во время данной фазы N – мерный вектор Q_{prob} подаётся на сеть Хопфилда в качестве начального состояния. Процесс формирования этого вектора зависит от этапа функционирования всей системы и определяется следующей зависимостью:

$$Q_{\text{prob}} = \begin{cases} Q_{\text{энтроп}}, & t = 0 \\ Q_d^{t-1}, & t > 0 \end{cases}, \quad (2.40)$$

где $Q_{\text{энтроп}} -$ вектор, полученный на первом такте работы системы, генерируется источником энтропии, $t -$ торт функционирования системы, $Q_d^{t-1} -$ вектор значений, полученных на предыдущем этапе работы системы от сети Хопфилда с номером $d = 1..H$, где $H -$ максимальное количество задействованных сетей в ГСЧ. После этого для извлечения вектора, сгенерированного сетью, задействуется динамическое правило, в котором все нейроны сети активируются в определенном порядке, определяемом колебаниями осциллятора, построенном на основе модифицированного нейрона Хопфилда (алгоритм функционирования будет рассмотрен ниже). Такая асинхронная процедура коррекции, для экономии времени, может продолжаться до тех пор, пока не будет произведено обновление состояния каждого нейрона. Значения выходов нейронов, зависящие от матрицы весов и состояний других нейронов, вычисляются по формуле:

$$Q_i(t+1) = f\left(\sum_{j=0}^{N-1} T_{ij} Q_j(t)\right), \quad (2.41)$$

где $T -$ матрица весовых коэффициентов, $Q_i(t) -$ состояние $i -$ го нейрона в момент времени t ,

$$f(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (2.42)$$

Кроме того, можно производить асинхронное извлечение информации до тех пор, пока сеть не сойдётся к аттрактору, и изменения не перестанут происходить.

После выполнения асинхронного извлечения информации будет получен N -мерный вектор, произведенный сетью. Для формирования случайного числа данная процедура выполняется N раз (размерность числа составляет N бит), за каждый этап происходит формирование одного бита информации из N мерного вектора, произведённого сетью. Данная операция выполняется с помощью логической нейронной сети, реализующей некоторое множество функций алгебры логики и выполняющей соответствующее логическое преобразование над входными данными. Цель данного преобразования состоит в уничтожении

зависимости, с помощью которой получаются случайные числа. Кроме того, логическая сеть способствует повышению линейной сложности для выходной последовательности чисел. Рассмотрим внутреннюю архитектуру логической нейронной сети (ЛНС), показанную на рисунке 2.28.

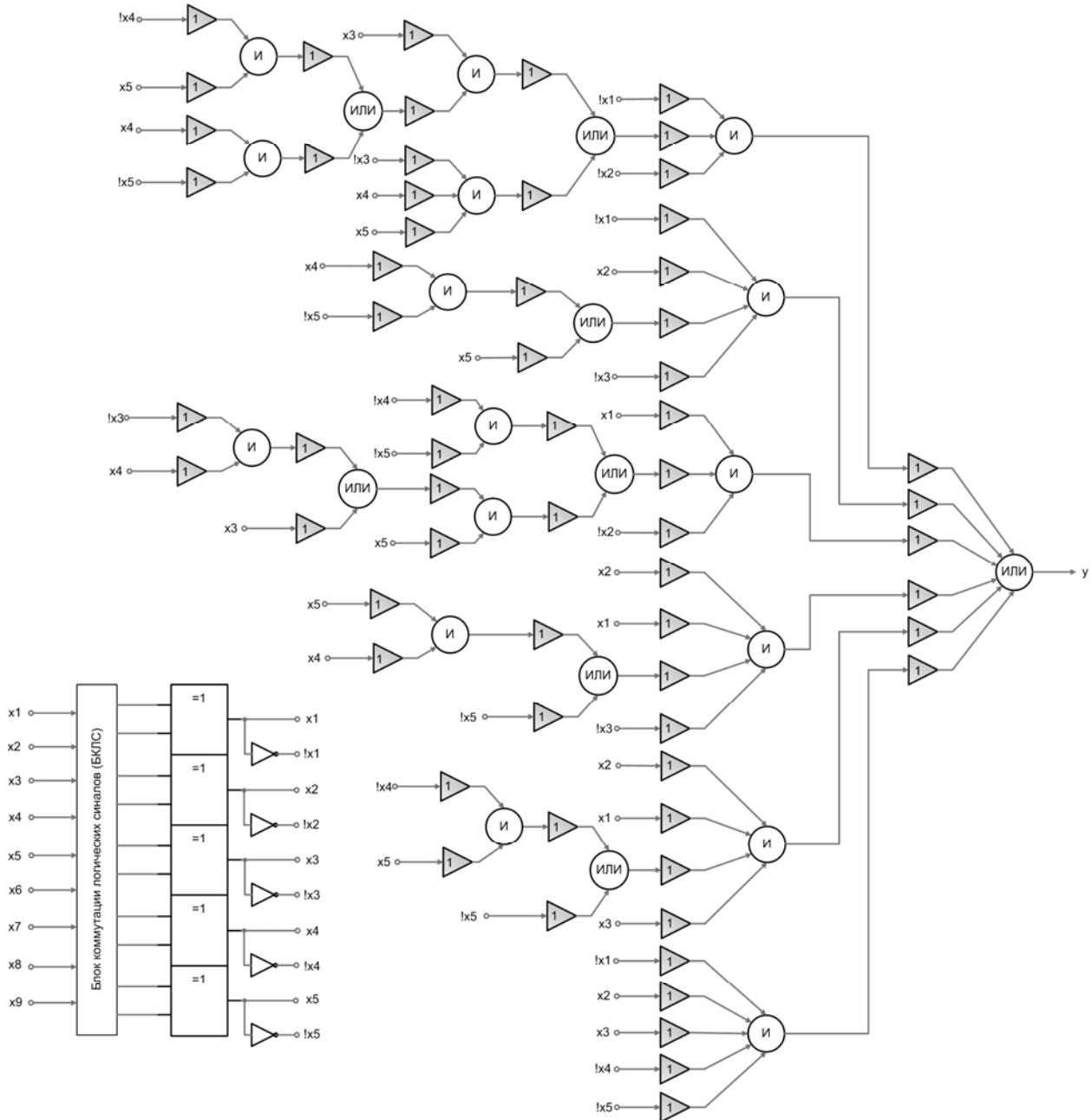


Рисунок 2.28 – Структурный график логической нейронной сети

ЛНС состоит из двух блоков: блок коммутации логических сигналов (БКЛС) выполняет поразрядную логическую операцию исключающего ИЛИ (XOR) над элементами входного вектора. В состав БКЛС входят и инверторы, с помощью которых получаем инвертированные значения логических сигналов (на

рисунке обозначены со знаком «!»), необходимых для функционирования ЛНС. Выполнение операции XOR производится по следующим правилам:

1. Каждый элемент XOR может выполнять преобразование не более чем над двумя компонентами входного вектора.
2. В качестве параметра пятого элемента XOR (10-й параметр для которого нет соответствующего значения входного вектора) допускается повторное использование значения одного из разрядов входного вектора.

В результате данного преобразования размерность вектора уменьшается с 9 разрядов до 5. Из полученных пяти разрядов, первые три разряда являются управляющими, они определяют, какую логическую функцию над оставшимися двумя битами выполнит логическая нейронная сеть. Схема преобразования выходного N -мерного вектора нейронной сети Хопфилда показана на рисунке 2.29.

Вектор, полученный от НС Хопфилда

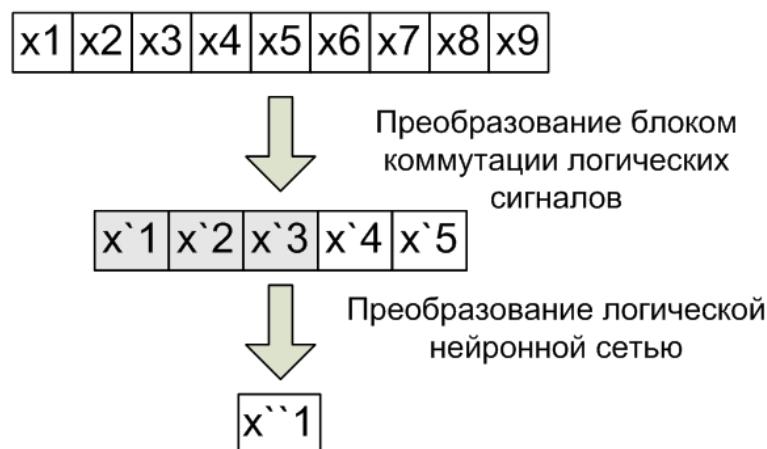


Рисунок 2.29 – Преобразование входной битовой последовательности

Логическая нейронная сеть строится из нейронов, приведённых в таблице 2.4. В качестве функции активации в логических нейронах используется функция активации единичного скачка $thr()$.

$$thr(v) = \begin{cases} 1, v > 0; \\ 0, v \leq 0, \end{cases} \quad (2.43)$$

где v – это индуцированное локальное поле нейрона [110].

Таблица 2.4 Компоненты логической нейронной сети

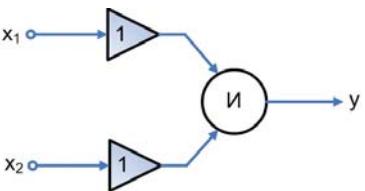
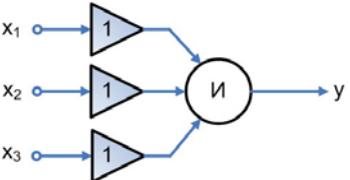
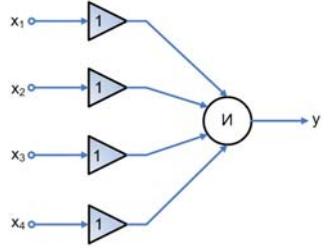
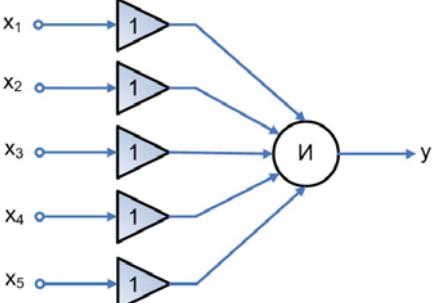
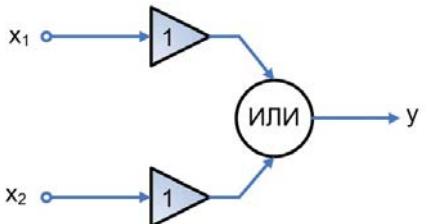
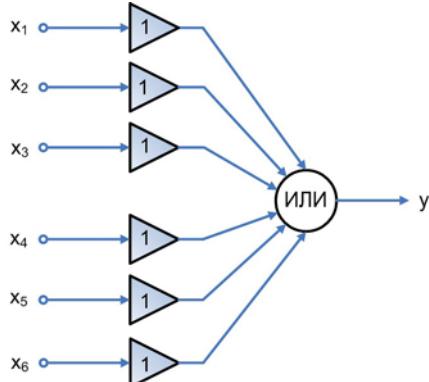
	<p>Нейрон 2-И.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1 - 1).$
	<p>Нейрон 3-И.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 1 - 2).$
	<p>Нейрон 4-И.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 1 + x_4 \cdot 1 - 3).$
	<p>Нейрон 5-И.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 1 + x_4 \cdot 1 + x_5 \cdot 1 - 4).$
	<p>Нейрон 2-ИЛИ.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1).$
	<p>Нейрон 6-ИЛИ.</p> $y = \text{thr}(x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 1 + x_4 \cdot 1 + x_5 \cdot 1 + x_6 \cdot 1).$

Таблица истинности для представленной логической нейронной сети представлена в таблице 2.5.

Таблица 2.5 Таблица истинности для ЛНС

x_1	x_2	x_3	x_4	x_5	Результат преобразования	Выполняемая логическая функция
0	0	0	0	0	0	Конъюнкция
0	0	0	0	1	0	
0	0	0	1	0	0	
0	0	0	1	1	1	
0	0	1	0	0	0	Исключающее «ИЛИ»
0	0	1	0	1	1	
0	0	1	1	0	1	
0	0	1	1	1	0	
0	1	0	0	0	0	Дизъюнкция
0	1	0	0	1	1	
0	1	0	1	0	1	
0	1	0	1	1	0	
0	1	1	0	0	1	Функция Пирса
0	1	1	0	1	0	
0	1	1	1	0	0	
0	1	1	1	1	0	
1	0	0	0	0	1	Логическая равнозначность
1	0	0	0	1	0	
1	0	0	1	0	0	
1	0	0	1	1	1	
1	0	1	0	0	1	Импликация от x_5 к x_4
1	0	1	0	1	1	
1	0	1	1	0	0	
1	0	1	1	1	1	
1	1	0	0	0	1	Импликация от x_4 к x_5
1	1	0	0	1	0	
1	1	0	1	0	1	
1	1	0	1	1	1	
1	1	1	0	0	1	Функция Шеффера
1	1	1	0	1	1	
1	1	1	1	0	1	
1	1	1	1	1	0	

Для формирования последовательности, в которой будет производиться активация сетей Хопфилда, управление процессом асинхронного извлечения информации из этой сети и генерация битовых последовательностей для

выполнения предобработки информации необходима структура способная генерировать «информационный шум». За основу такой системы был взят нейрон Хопфилда, показанный на рисунке 2.30 [55]. Данный тип нейрона относится к динамическим нейронам, поведение которых определяется предысторией их функционирования. Нейрон Хопфилда описывается рекуррентным уравнением:

$$x(k+1) = \psi\left(\sum_{i=1}^n w_i x_i(k) + \theta\right), \quad (2.44)$$

где k – отчёты моментов времени, x – входные сигналы, w – весовые коэффициенты, ψ – функция активации нейрона, θ – порог нейронного элемента. Нейрон Хопфилда в цепи обратной связи имеет элемент задержки, реализующий операцию сдвига назад во времени ($z^{-1}x(k+1) = x(k)$).

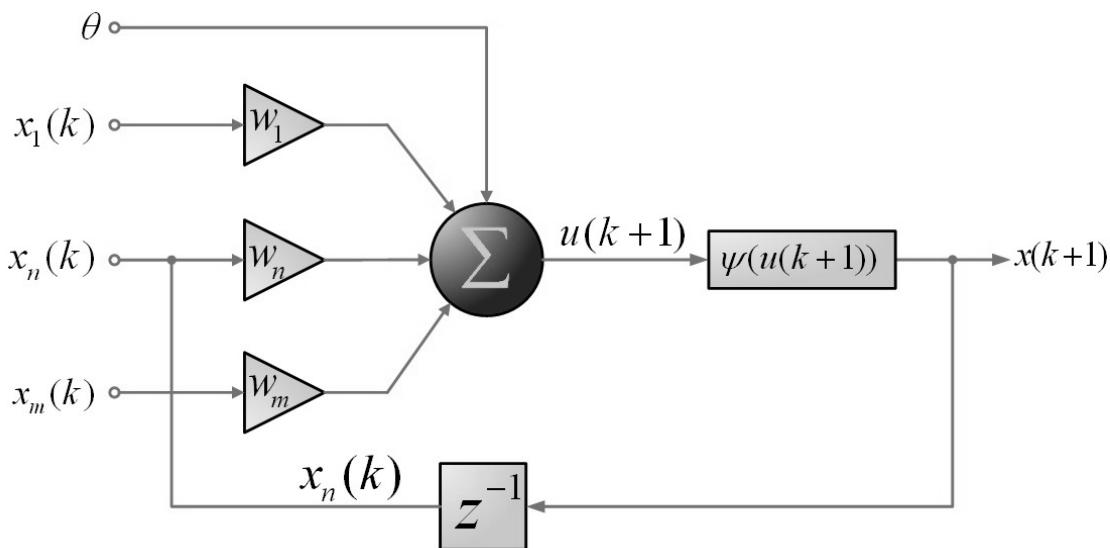


Рисунок 2.30 – Нейрон Хопфилда

На основе рекуррентного нейрона производится построение нейронной сети – осциллятора. Данная структура состоит из 4-х модифицированных нейронных элементов, в которых особое внимание уделяется колебательному режиму работы. Структурная схема сконструированной нейронной сети – осциллятора показана на рисунке 2.31.

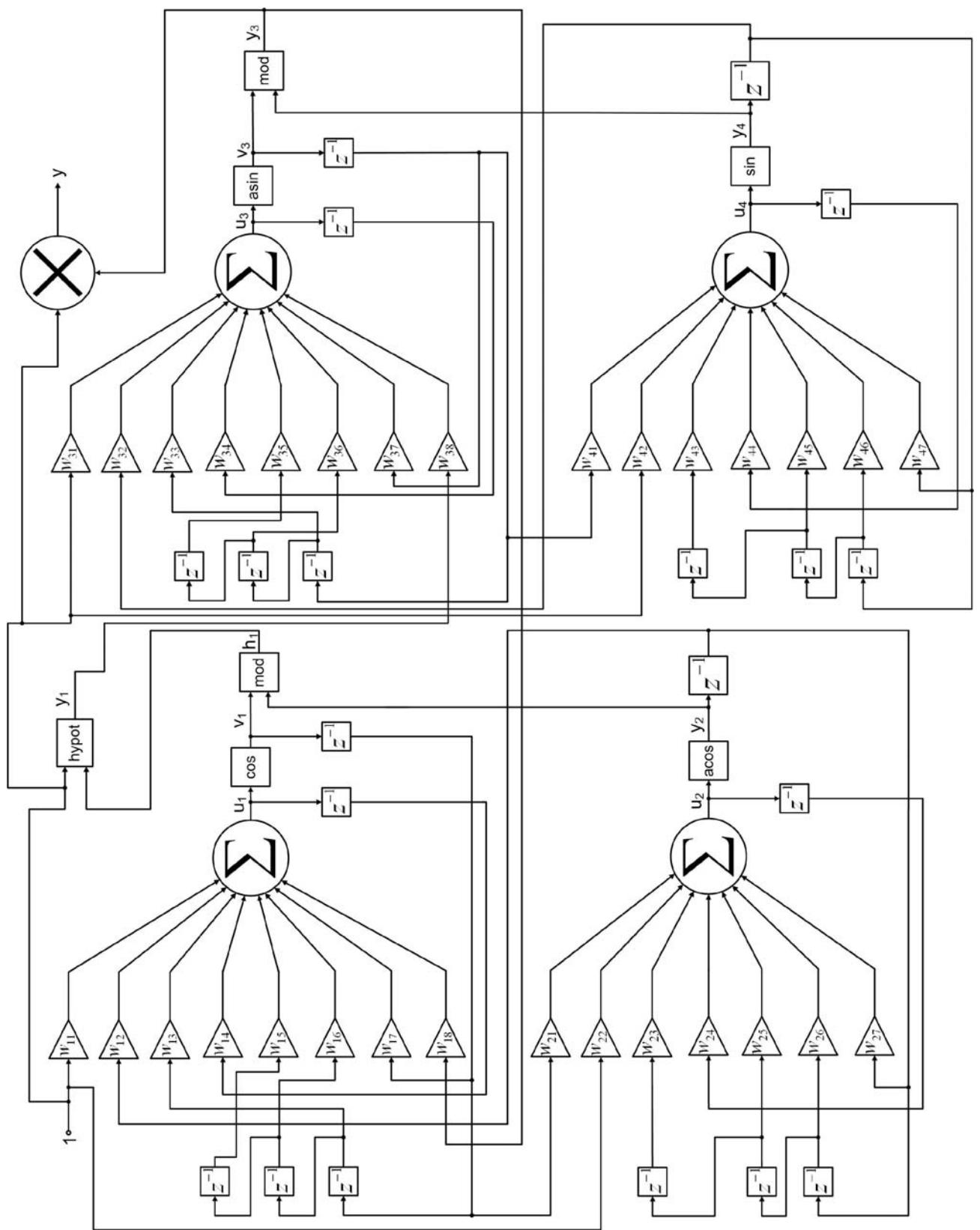


Рисунок 2.31 – Нейронная сеть - осциллятор

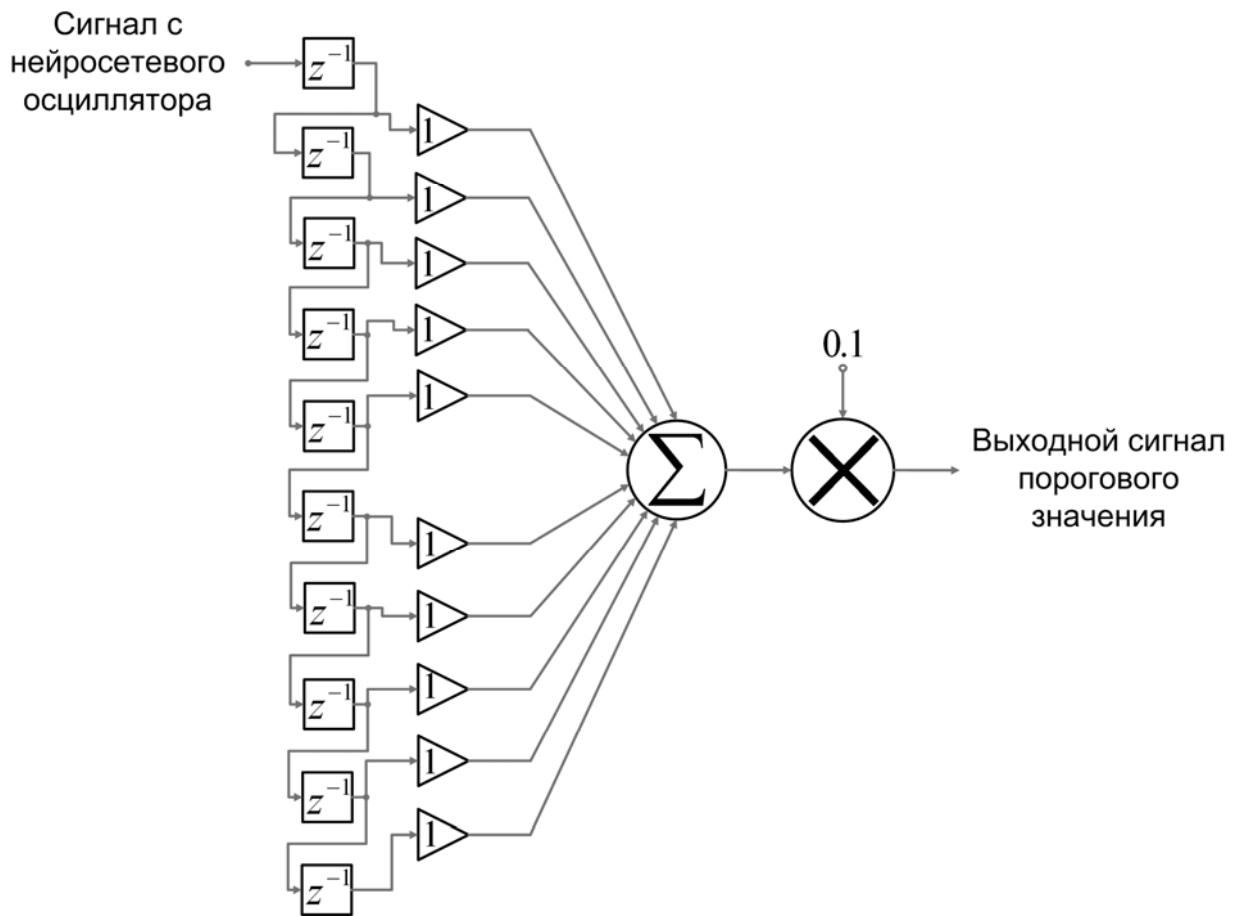


Рисунок 2.32 – Структурная схема формирования сигнала порогового значения

Выходной сигнал данной системы имеет сложный характер, поэтому при его преобразовании в бинарную последовательность задаётся определённое значение порога. Если значение сигнала на текущем такте работы системы выше порогового уровня – это «логическая единица», в противном случае – «логический ноль». Для формирования порога используется среднее значение предыдущих 10 отсчётов. Структурная схема системы, формирующей пороговое значение, представлена на рисунке 2.32.

2.7 Проектирование алгоритма решения задачи безопасной маршрутизации

На основе обученного комитета нейросетевых экспертов строится адаптивный маршрутизатор. В предлагаемом способе маршрутизации предполагается использовать ассоциативную машину в каждом узле компьютерной сети. Количество выходов финального эксперта – сигма-пи сети, а также количество выходов каждого нейросетевого эксперта соответствует количеству исходящих соединений. Параметры нейронных сетей определяются с

помощью разработанных алгоритмов обучения и настройки начальных состояний. Когда необходимо выбрать путь в передаче данных, информация о пункте назначения и о состоянии компьютерной сети подаётся на рецепторы, а в выходном слое, построенном по принципу «победитель забирает всё», активируется номер нейрона, который показывает следующий путь в передаче данных. Для того чтобы маршрутизатор мог функционировать, необходимо сформировать информацию о состоянии соединений, исходящих из данного узла. Каждому соединению присваивается определённый вектор параметров, компоненты которого характеризуют определённую составляющую физического соединения. Одними из важнейших параметров, которые необходимо учитывать при выборе дальнейшего пути маршрутизации информации, является тип канала связи, его пропускная способность и надёжность. Для некоторых каналов связи характеристики, используемые при обучении и функционировании комитета сетей, приведены в таблице 2.6 [13]. Параметры пропускной способности и надёжности представляются значениями в интервале от 0 до 1, которые характеризуют тип канала и кабеля связи по сравнению с параметрами, выбранными в качестве эталонных и имеющими максимальные значения пропускной способности и надёжности.

Нейросетевым экспертам необходимо определить, в каком направлении необходимо осуществлять передачу. Для этого один из компьютеров сети берётся в качестве начала координат, а координаты остальных рассчитываются относительно данного компьютера. В качестве примера на рисунке 2.33 показан пример сети связи, которая состоит из 17 узлов. Узел №8 принят за начало координат.

Рассмотренные параметры оценки каналов передачи информации относятся к классу статических, которые сравнительно долгое время остаются стабильными во времени. Для оперативной реакции маршрутизатора на возможные попытки подключения к каналу связи автором был разработан метод анализа, позволяющий получить динамические параметры характеристик канала связи.

Таблица 2.6 Характеристики качества канала связи

Анализируемая характеристика				
Пропускная способность			Надёжность	
Тип канала	Скорость передачи	Параметр пропускной способности	Тип кабеля	Параметр надёжности
Ethernet	10 Мбит/с	0,8	Коаксиальный кабель	
Fast Ethernet	100 Мбит/с	0,9	«Толстый» коак. кабель	0,31
Gigabit Ethernet	1000 Мбит/с	0,95	«Тонкий» коак. кабель	0,22
Канал Т-1	1,544 Мбит/с	0,45	Телевизионный кабель	0,15
Канал Т -2	6,312 Мбит/с	0,61	Витая пара	
Канал Т-3	44,736 Мбит/с	0,85	Экранированная	0,6
Канал Т-4	274 Мбит/с	0,93	Неэкранированная	0,5
Канал 56 Кбит/сек.	56 Кбит/с	0,33	Волоконно - оптический	
Канал Е-2	8,488 Мбит/с	0,65	Многомодовый	0,8
Канал Е-1	2,048 Мбит/с	0,55	Одномодовый	1,0

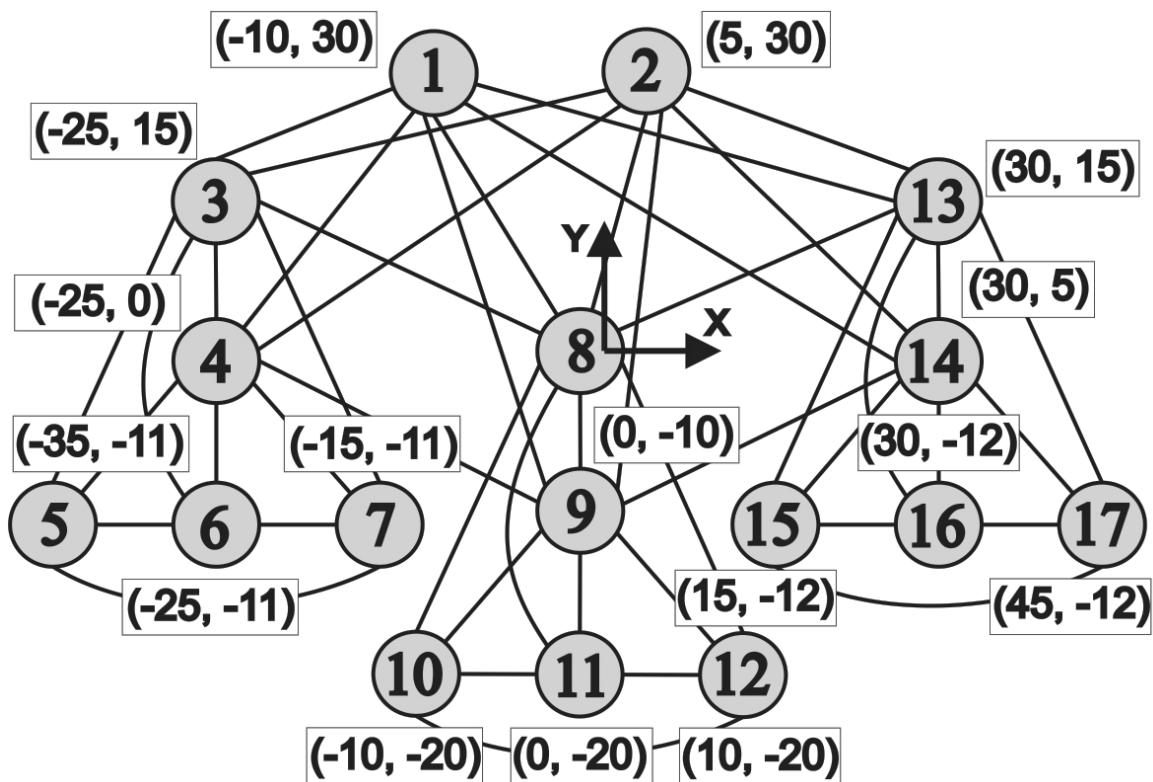


Рисунок 2.33 – Формирование данных о координатах узлов

Формирование обучающей выборки для системы экспертов производится путём анализа линии связи, которая используется в рассматриваемой локальной сети. Вторжение может быть осуществлено путём прямого подсоединения к каналу связи и считывании информации с помощью технических средств. Вследствие этого должна быть возможность определения попыток подключения к каналу.

Электрические свойства линий связи и качество передачи по ним полностью характеризуются первичными параметрами передачи [112]: активным сопротивлением R , индуктивностью L , емкостью C , проводимостью изоляции G . Эти параметры не зависят от напряжения и передаваемого тока, а определяются лишь конструкцией линии, используемыми материалами и частотой тока.

На рисунке 2.34 представлена эквивалентная схема участка цепи связи. Здесь включенные последовательно R и L образуют суммарное сопротивление, а G и C – суммарную проводимость.

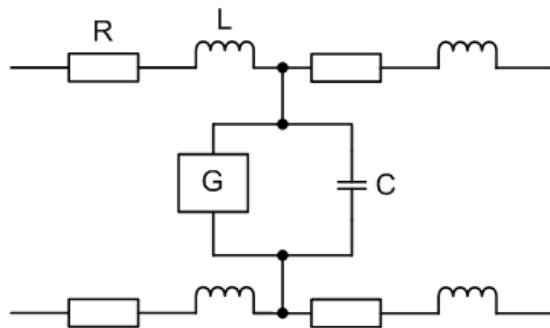


Рисунок 2.34 – Эквивалентная схема линии связи

Выявление вторжения в канал связи основано на детектировании изменений параметров линии связи. Для этой цели был использован генератор прямоугольных импульсов, построенный на микросхеме NE555 [108]. В качестве канала передачи данных в данной работе была рассмотрена „витая пара“. Отдельные жилы кабеля подключались к RC – цепочке генератора и в случае несанкционированного подключения происходило изменение частоты колебаний. Мониторинг канала связи проводился при различных частотах генератора. Частота генератора регулировалась путём изменения сопротивления резисторов в RC цепочке, что делало невозможным злоумышленнику имитировать частоту и скрыть своё подключение. Изменение сопротивления осуществлялось

применением в качестве сопротивления в RC цепочках полевых транзисторов, изменяя напряжение между затвором и истоком (например, с помощью цифроаналогового преобразователя *AD7524*) можно настроить колебания генератора на определенной частоте. Подключение к каналу связи влечёт за собой увеличение ёмкости конденсатора RC цепочки и уменьшение частоты колебаний. Обучающая выборка для нейросетевых экспертов формируется с учётом именно подключения к каналу, и, как следствие, происходит изменение ёмкости или сопротивления линии связи.

Необходимо выявлять изменения в характеристике канала, которые связаны с попаданием в канал влаги или воздействиями расположенных рядом источников помех.

Способы подключения отдельных жил кабеля к RC цепи генератора могут быть различными. Основную роль здесь играют параметры анализируемой линии, её сопротивление и ёмкость. Способ соединения и анализируемый параметр выбирается, исходя из максимальной чувствительности к тому или иному параметру. Рассмотрим варианты соединения в случае анализа изменения сопротивления в канале передачи информации. В качестве проводника используется экранированная витая пара.

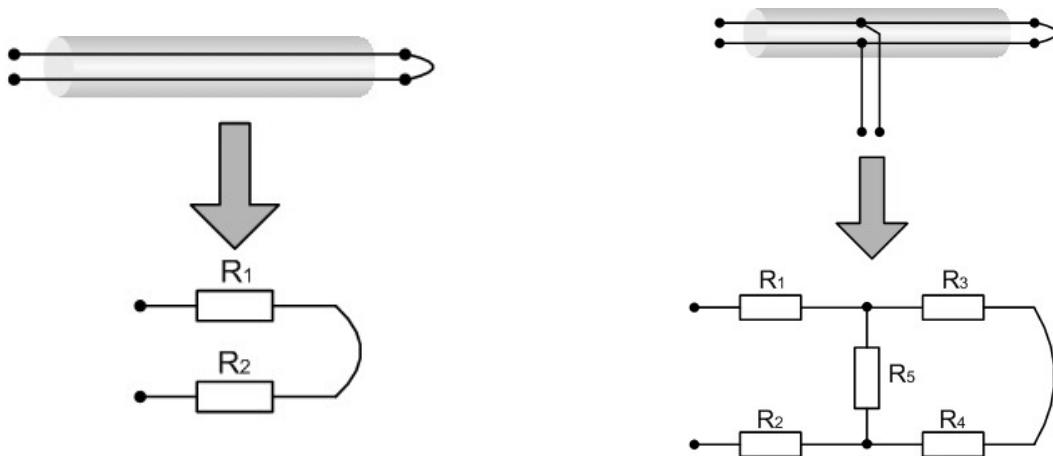


Рисунок 2.35 – Эквивалентная схема подключения

Два проводника при их замыкании представляют собой сопротивление, которое может быть использовано в RC цепочке генератора, как показано на рисунке 2.35. Изменение этого сопротивления будет влиять на частоту колебаний,

что и позволит детектировать стороннее подключение. Эквивалентная схема при наличии отводов показана на рисунке 2.35 справа.

Вторым параметром для анализа состояния кабельной линии является ёмкость. Способы подключения здесь также могут быть различными. На рисунке 2.36 представлен способ подключения, при котором определяется значение ёмкости между двумя жилами кабеля.

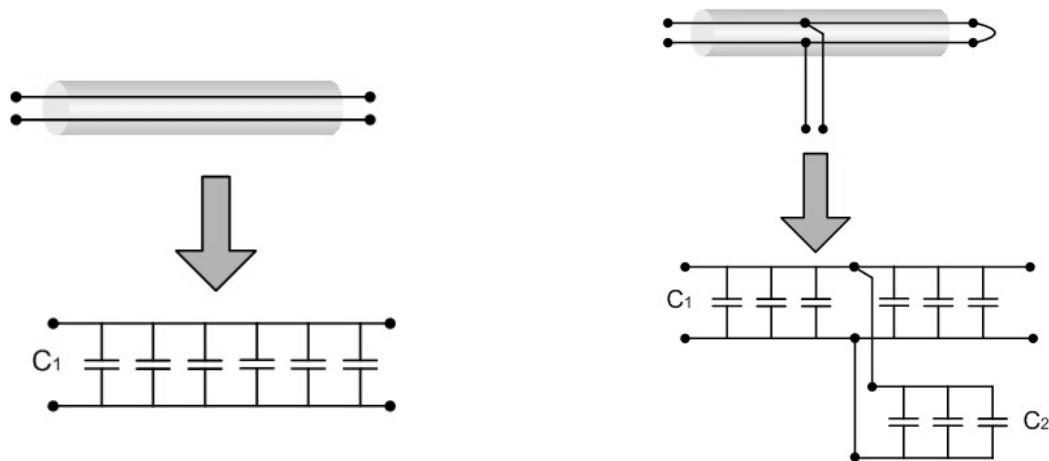


Рисунок 2.36 – Эквивалентная схема для определения ёмкости

При наличии отводов в кабеле ёмкость увеличится, что приведёт к изменению частоты генераторов. Это позволит детектировать подключения к каналу. Необходимо учитывать изменение ёмкости, связанное не только со сторонним подключением. Так, наличие влаги в кабеле способствует увеличению ёмкости. Необходимо обучить ассоциативную машину выявлять ситуации такого рода.

Второй способ получения динамических характеристик основан на использовании принципов рефлектометрии. Для анализа состояния кабельных линий связи чаще всего применяют рефлектометр – прибор, функционирующий по тем же принципам, что и радар [101]. Производится подключение к исследуемому кабелю, после чего рефлектометр посылает последовательность импульсов в кабель связи и фиксирует полученный отражённый сигнал. Энергия импульса будет отражаться от различных неоднородностей канала связи, скруток, несанкционированных подключений. По отражённому сигналу можно судить о

состоянии всей кабельной системы в целом. Существенным недостатком данного метода является высокая стоимость устройств такого типа. В данной работе в качестве основного механизма для анализа канала связи предлагается использовать рассмотренную выше нейросетевую структуру в совокупности с разработанной схемой анализа канала передачи (рисунок 2.37).

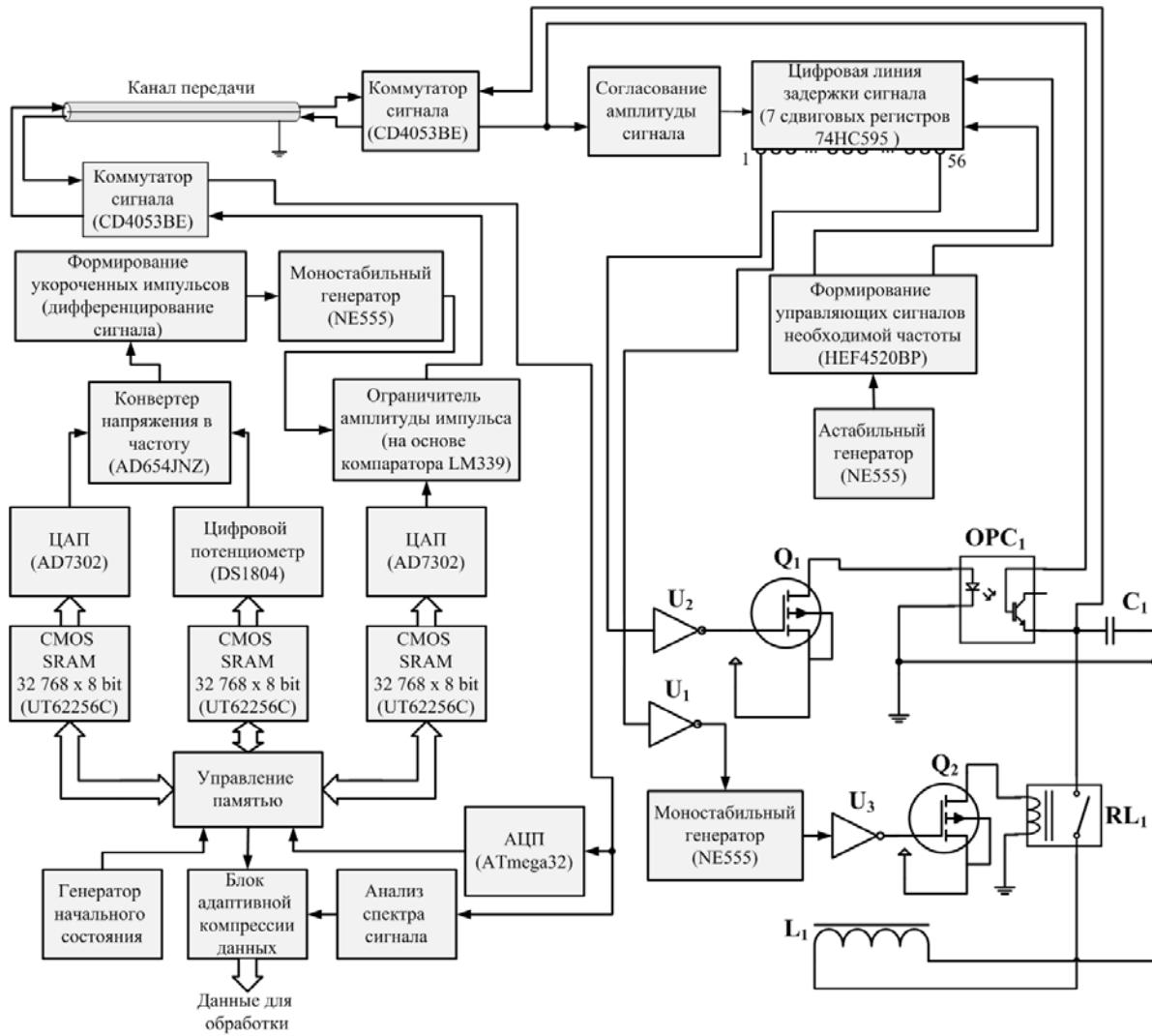


Рисунок 2.37 – Структурная схема анализа канала передачи информации

Основной проблемой является вопрос формирования обучающей выборки для комитета нейронных сетей, чтобы в дальнейшем он мог взять на себя функции по анализу состояния канала связи. Для формирования числовых последовательностей, характеризующих состояние канала связи, был разработан следующий алгоритм. Электрическая схема, реализующая данный метод, располагается на противоположной стороне канала передачи информации, в то

время как генератор зондирующих импульсов располагается перед входом в канал.

1. В качестве зондирующего импульса используется единичный импульс от моностабильного генератора, выполненного на микросхеме интегрального таймера *NE555*. Изменяя параметры внешних компонентов генератора – ёмкости и сопротивления – задаётся необходимый период сигнала. Применение моностабильного генератора позволит регулировать следование импульсов, как в ручном, так и в автоматическом режиме (например, с помощью микроконтроллера *ATmega8A*), что необходимо на начальных этапах настройки данной системы. Импульс в канал подаётся через блок ограничителя амплитуды (компаратор *LM339* в совокупности с микросхемой *CD4053BE* – мультиплексора сигналов) для обеспечения безопасности самой системы (будет рассмотрено ниже).

2. Сгенерированный прямоугольный импульс поступает в канал передачи через коммутатор сигналов (микросхема *CD4053BE*). После прохождения канала связи на другой стороне сигнал проходит через аналогичное коммутирующее устройство. На данном этапе сигнал, полученный из канала связи используется не только для получения его характеристик, но и для управления процедурой «отражения» обратно в канал. Для осуществления функции управления его амплитуда стабилизируется до напряжения 5 вольт с помощью блока согласования амплитуд (выполнен на компараторе *LM339*). После чего сигнал поступает на вход данных цифровой линии задержки (ЦЛЗ). ЦЛЗ выполнена на семи 8-ми разрядных сдвиговых регистрах *74HC595*. Данный тип регистров можно соединить последовательно для увеличения разрядности. В результате данный регистр применяется как элемент задержки зондирующего сигнала. Кроме входа данных регистр *74HC595* обладает входами для тактовых импульсов, осуществляющих сдвиг данных, и входом для «зашёлкивания» данных (отображения внутреннего состояния регистра на линии вывода). Путём настройки частоты следования тактовых импульсов (блок формирования управляющих сигналов, выполненный на счётчике *HEF4520BP*) для этих входов

можно управлять продвижением сигнала по линии задержки, выполненной на регистрах. Для этой цели применялся интегральный таймер *NE555*, работающий в режиме астабильного генератора. Изменение параметров таймера, приводит к изменению характеристик линии задержки, что позволяет настроить систему на заданную частоту анализа канала связи.

3. После генерации зондирующего импульса он поступает на линию задержки, представленную набором регистров. С первого информационного разряда первого регистра (ПИРПР) положительный импульс поступает на вход инвертора U_2 . Выход инвертора соединён с затвором полевого транзистора Q_1 (*IRF4905*) который управляет светодиодом оптопары $OPC_1(4N35)$. Поступление положительно импульса вызывает увеличение проводимости фототранзистора оптопары, эмиттер которого соединён с выходом коммутатора сигнала, а коллектор с анодом конденсатора ёмкостью 22 мкФ. В результате, пока действует положительное значение сигнала на ПИРПР, производится накопление энергии пришедшего импульса. При этом часть энергии будет потеряна в линии связи за счёт затухания сигнала и частичного его отражения от различных неоднородностей.

4. На следующем этапе работы устройства необходимо вернуть энергию пришедшего импульса обратно. Это необходимо для удобства анализа линии связи, т.к. будет отсутствовать необходимость генерировать зондирующие импульсы на одной стороне канала передачи, а производить измерения на другой. Для невозможности фальсификации возвращённого импульса (что может быть сделано при попытке злоумышленника скрыть несанкционированные манипуляции с каналом передачи информации) производиться не просто разряд конденсатора в линию связи и формирование прямоугольного импульса. Создаётся простейший колебательный контур из уже заряженного конденсатора и катушки индуктивности L_1 . После того как зондирующий сигнал пройдёт через линию задержки и на последнем информационном разряде последнего сдвигового регистра будет сформирован положительный импульс, который через инвертор U_1 (*HEF4049BP*) подаётся на вход второго моностабильного генератора,

формирующего импульс заданной длительности. Продолжительность импульса выбирается достаточной для завершения всех переходных процессов при замыкании заряженного конденсатора на катушку индуктивности L_1 с помощью электромагнитного реле RL_1 (*HK23F-DC5V-SHG*). Импульс на катушку реле подаётся не напрямую от генератора, а через инвертор (*74HC14*) сигнал поступает на затвор полевого транзистора Q_2 (*IRF4905*), исток которого соединён с положительным полюсом источника питания, а сток с выводом катушки реле.

5. После выполнения процедуры быстрого преобразования Фурье (БПФ) над отсчётами полученного «отражённого» сигнала, производиться формирование обучающей выборки для нейронной сети.

При проверке состояния канала передачи информации в автоматическом режиме необходимо обеспечить последовательное повторение пунктов 1-5 данного алгоритма. При этом необходимо заметить, что генератор зондирующих импульсов расположен на одной стороне анализируемого канала передачи, а устройство «отражения» сигнала на другом. Это делает возможным осуществить атаку на всю проверяющую систему типа «человек посередине». Данный тип вредоносного воздействия основан на имитации злоумышленником «отражённого» сигнала путём подключения непосредственно к линии связи, что позволяет скрыть следы манипуляций с каналом передачи. Для устранения этой проблемы необходимо изменять периодичность и амплитуду импульсов сигналов запуска поступающих на моностабильный генератор зондирующих импульсов. Для генерации управляющих импульсов применялся конвертер напряжения в частоту (*AD654JNZ*), частотой колебаний данного генератора управляют три внешних компонента: ёмкость, сопротивление и входное напряжение. В данной работе ёмкость была фиксирована – 6800 пФ. Сопротивление изменялось с помощью двух цифровых потенциометров *DS1804* (каждый на 10 кОм.). Управление входным напряжением осуществлялось с помощью цифро-аналогового преобразователя (ЦАП) *AD7302*. Таким образом, изменяя два параметра во времени, становится возможным регулировать частоту управляющих импульсов. Сигналы с конвертера поступают на

дифференцирующую цепочку (конденсатор и резистор) для укорачивания сигнала. Данная процедура необходима для устранения возможности генерации более одного колебания генератором зондирующих импульсов от одного управляющего импульса.

Амплитуда импульса зондирующего сигнала изменялась путём ограничения верхнего значения амплитуды с помощью компаратора *LM339*. Когда происходило превышение значения напряжения амплитуды сигнала выше заданного напряжения, то компаратор ограничивал сигнал на заданном максимальном значении напряжения. Формирование значения напряжения для ограничения верхней границы амплитуды импульса осуществлялось ЦАП *AD7302*. ЦАП и цифровой потенциометр (ЦП) имеют цифровой интерфейс: для управления значениями напряжения и сопротивления необходимо по 1 байту данных. Т.к. в данной работе было использовано два ЦАП и один ЦП, то необходимо три байта данных. Для получения необходимых управляющих байт было применено 3 банка памяти на основе микросхемы *CMOS SRAM UT62256C* ($32\ 768 \times 8\ bit$). Для того чтобы процесс анализа состояния канала связи был непредсказуем и при анализе развивался по случайному направлению, перед процедурой запуска зондирования необходимо инициализировать память. В основу данного процесса положен механизм самоанализа канала связи. Для этого память инициализируется фиксированными значениями, например от генератора случайных чисел. После этого начинается процесс предварительного зондирования канала. После возвращения сигнала производится не анализ спектра, а вычисляется среднее значение амплитуды вернувшегося сигнала (вычисления осуществляются с помощью АЦП микроконтроллера *ATmega32*). После чего выполняется операция исключающего «ИЛИ» между соответствующими управляющими байтами памяти и значением амплитуды вернувшегося сигнала. В данной работе применялось 50 таких генераций, в результате чего память инициализировалась уникальными для данного канала значениями. После данного процесса начинал работу алгоритм анализа канала связи.

В данной работе для анализа состояния канала передачи производится оценка 1400 значений спектра сигнала. Входной вектор такой размерности приведёт к значительному усложнению архитектуры нейросетевого эксперта, который будет проводить анализ вектора входного сигнала такой размерности. Перед подачей на вход сети вектор необходимо сжать. В данной работе был разработан алгоритм компрессии данных с переменными параметрами, определяемыми характеристикой анализируемого канала, для обеспечения безопасной работы всей системы. В качестве основного базового компонента алгоритма сжатия данных применялся модифицированный нейрон N – адалина (рисунок 2.38). В данном нейроне блок \sin выполняет преобразование: $\sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$, блок \cos – $\cos(\pi \cdot x) \cdot \cos(\pi \cdot y)$, блок \tanh – $\tanh(x) \cdot \tanh(y)$. Из представленного нейросетевого элемента строится алгоритм нейросетевой компрессии данных. Входной слой имеет размерность равную размерности сжимаемого вектора (1400). Каждый информационный вход имеет по три отвода для соединения с нейронами, т.к. базовая структура имеет два входа, то первый слой состоит из 2100 элементов. Последующие слои обладают только одним информационным выводом и если предыдущий слой содержит нечётное количество элементов, то его значение игнорируется. На основе данных принципов выполняется построение всей нейросетевой структуры для сжатия данных. Полученная сеть будет иметь 7 слоёв: 1400 – 2100 – 1050 – 524 – 262 – 130 – 64 – 32.

В результате функционирования данной сети производится сжатие вектора исходных данных до 32 значений. При компрессии такого высокого уровня необходимо учитывать, что основным элементом принятия решения в данной задаче является нейронная сеть. Алгоритм сжатия должен быть чувствителен к малейшим изменениям в первоначальной структуре данных, чтобы нейросетевой эксперт мог адекватно реагировать на изменения, происходящие в среде передачи. Для проверки способности данной нейросетевой структуры адекватно реагировать на изменения в исходном векторе производился анализ реакции

системы на изменение значения сотой доли в одной компоненте данных (всего 1400).

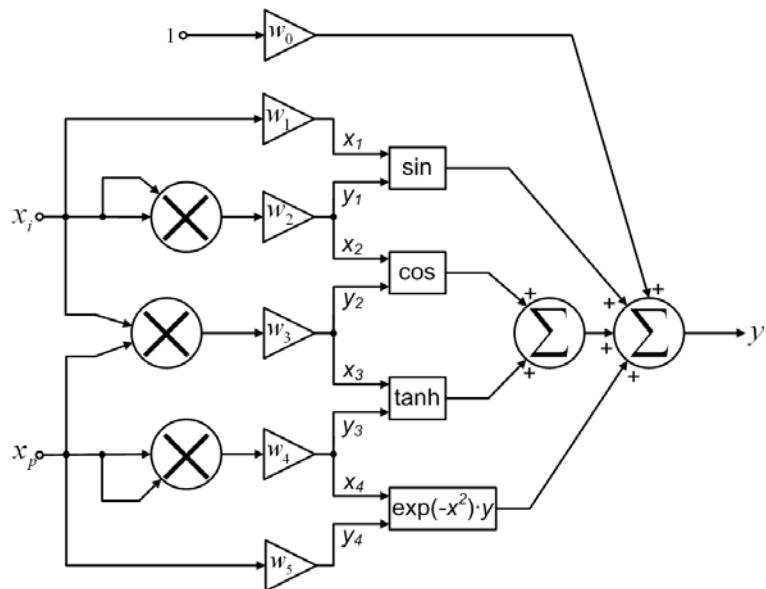


Рисунок 2.38 – Базовая нейросетевая структура алгоритма сжатия данных

Эксперименты показали, что при таком незначительном изменении входного вектора происходит изменение в среднем 3-х компонент из 32 выходного вектора. В результате нейросетевой эксперт сможет адекватно реагировать даже на малейшие изменения в векторах, характеризующих канал передачи информации.

Основным вопросом остаётся начальное формирование структуры сети, осуществляющую компрессию данных: весовых коэффициентов и топологии связей между слоями. Весовые коэффициенты выбиралась так, чтобы процесс преобразования был уникален для каждого канала связи: начальные значения выбирались из трёх модулей памяти *CMOS SRAM*. Всего в нейросетевой структуре для сжатия данных содержится 4162 нейронных элемента, каждый содержит по 6 весовых коэффициентов. В результате 24972 значения определяются путём комбинации 3 байт из каждого модуля памяти и последующего нормирования. Адрес каждого байта в модуле памяти определяется как порядковый номер весового коэффициента в нейросетевой структуре. Топология связей настраивается с помощью генетического алгоритма, где целевой функцией является увеличение чувствительности нейросетевого алгоритма сжатия к изменению входных данных.

2.8 Выводы по главе 2

1. Настройка и обучение комитета экспертов, состоящего из множества нейронных сетей, требует выполнения сложной процедуры оптимизации параметров нейронных сетей. Процесс обучения требует выполнения оценки компетентности нейросетевой структуры решать поставленную задачу после выполнения процесса обучения. Итоговое решение будет принято на основании результатов функционирования сетей, адаптированных под решение задачи.
2. Градиентные алгоритмы обучения нейронных сетей, модифицированные комбинированными эвристическими процедурами, позволяют перевести процесс поиска оптимальных параметров нейронных сетей в другую точку поискового пространства, что позволит избежать стагнации ассоциативной машины.
3. Разработанная нечёткая модель для анализа эффективности процедуры обучения НС позволит отдать приоритет в принятии решения только тем сетям, которые способны принять правильное решение в условиях неполной информации. Для обработки сигналов нейросетевых экспертов в работе разработан комплекс нейронных элементов в совокупности с нейронной сетью типа сигма-пи.
4. Спроектированный кооперативный иммунный алгоритм установки параметров НС перед процессом обучения позволит сократить время настройки сетей.
5. Разработанный способ генерации случайных чисел на основе множества взаимодействующих сетей Хопфилда выполняет начальную инициализацию всех параметров системы перед процедурой обучения. Запуск процедуры обучения из случайного состояния позволит обеспечить безопасность всей нейросетевой структуры, т.к. становится невозможным предсказать состояние системы в произвольные моменты времени.
6. Предложенный способ анализа каналов передачи информации позволит получить необходимые данные для начала процедуры обучения. Информация о состоянии каналов связи в дальнейшем используется комитетом сетей для определения безопасного маршрута передачи информации.

3 Практическая реализация разработанных моделей и алгоритмов

3.1 Требования к программной эмуляции нейроимитаторов

Программная реализация разработанного комитета нейросетевых экспертов должна выполнять следующие действия:

1. Реализовывать все разработанные алгоритмы обучения НС.
2. Осуществлять визуализацию полученных отсчётов среднеквадратической ошибки обучения.
3. Зафиксировать обученные нейронные сети.
4. Обеспечить формирование элементов обучающей выборки.
5. Выполнять конфигурацию и обеспечивать взаимодействие иммунных и эволюционных методов оптимизации для настройки начального состояния сетей перед выполнением процедуры обучения.
6. Осуществлять связь с элементами аппаратной части для генерации случайных чисел и оценки состояния каналов связи.
7. Выполнять компрессию данных, характеризующих состояние сетевых соединений, для последующего их применения при построении обучающей выборки.
8. Реализовывать информационный обмен между модулями, выполняемыми на разных узлах вычислительной сети, для обновления информации о состоянии сетевых соединений.
9. Выполнять построение модели вычислительной сети для отладки и настройки параметров алгоритмов.

Функционирование нейронной сети определяется множеством факторов: выбором модели нейронов, видом функции активации, количеством слоёв, числом нейронных элементов в каждом слое и т.д. Нейроимитатор должен правильно определить и выявить особенности решаемой задачи, оценить степень её сложности, выявить особенности входных данных. Программная реализация нейроимитатора должна обладать необходимыми функциональными возможностями для определения нейросетевых архитектур и конфигурации системы экспертов для решения поставленной задачи. Для повышения

эффективности обучения необходимо провести анализ исходных данных перед процедурой настройки ассоциативной машины с целью локализации и удаления аномальных значений, полученных на этапах генерации и сохранения данных.

3.2 Основные компоненты архитектуры программной системы

Автором был разработан программный продукт в среде Microsoft Visual Studio 2010 на языке программирования C#. Система предназначена для получения сведений о каналах передачи информации, обучении на основе полученных данных комитета нейронных сетей и выдачи финального результата, определяющего маршрут продвижения данных. Разработанное программное обеспечение может быть использовано для повышения эффективности современных средств маршрутизации информации в локальных сетях.

Программный продукт позволяет выполнить следующие действия:

1. Получение информации о состоянии имеющихся каналов передачи.
2. Построение модели анализируемой локальной сети с целью выявления маршрутов продвижения информации, обладающих наибольшей безопасностью.
3. Формирование обучающих векторов для нейросетевых экспертов на основе информации полученной в пункте 2. В качестве базового алгоритма для получения маршрутов, используемых в процессе обучения НС, применялся алгоритм Флойда.
4. Формирование архитектуры нейронных сетей.
5. Обучение нейросетевых экспертов с помощью градиентных алгоритмов с применением эвристик.
6. Оценку компетентности НС в данной предметной области, выполненной на основе нечёткой модели.
7. Предварительную обработку сигналов от НС модифицированными нейронами, и анализ полученных решений финальным экспертом, представленным сигма-пи нейронной сетью.
8. Генерацию случайных чисел на основе множества сетей Хопфилда.

9. Поддерживать связь с аппаратным источником энтропии, необходимым для нормального функционирования нейросетевого ГСЧ.

Программный продукт позволяет автоматически оценить сложность полученной модели локальной вычислительной сети и на основе этой оценки выполнить конфигурацию системы. Автоматическая настройка системы позволяет пользователям, не владеющим знаниями по теории применения гибридных нейросетевых и эволюционных алгоритмов, решать задачу безопасной маршрутизации информации.

Рассмотрим состав внутренней архитектуры программного обеспечения, приведённый на рисунке 3.1.

Модули источников данных представлены системами динамической и статической оценки параметров канала связи на основе мониторинга измерения ёмкости и сопротивления канала связи, а также оценки изменения спектра зондирующего сигнала. Учётная система поддерживает связь с аппаратурой модуля динамического анализа канала передачи информации. Модуль оценки параметров канала связи на основе предварительно введённых данных о типе канала связи, основываясь на данных из внешних источников (данные в локальных файлах, электронные архивы, базы данных) выдаёт качественную оценку физических маршрутов передачи информации. Модуль оценки качества данных выявляет проблемы с представлением отдельной информации. В процессе профайлинга [83] определяется информация об определённом атрибуте входного обучающего вектора и производится проверка условий, накладываемых на данный параметр. Если анализируемое значение удовлетворяет ограничениям, то считается, что векторы для обучения нейронных сетей соответствуют требуемому уровню качества. На основе проведённого исследования входных векторов формируется стратегия очистки и предобработки данных перед их использованием в процедуре обучения НС (*модули предварительной оценки данных*).

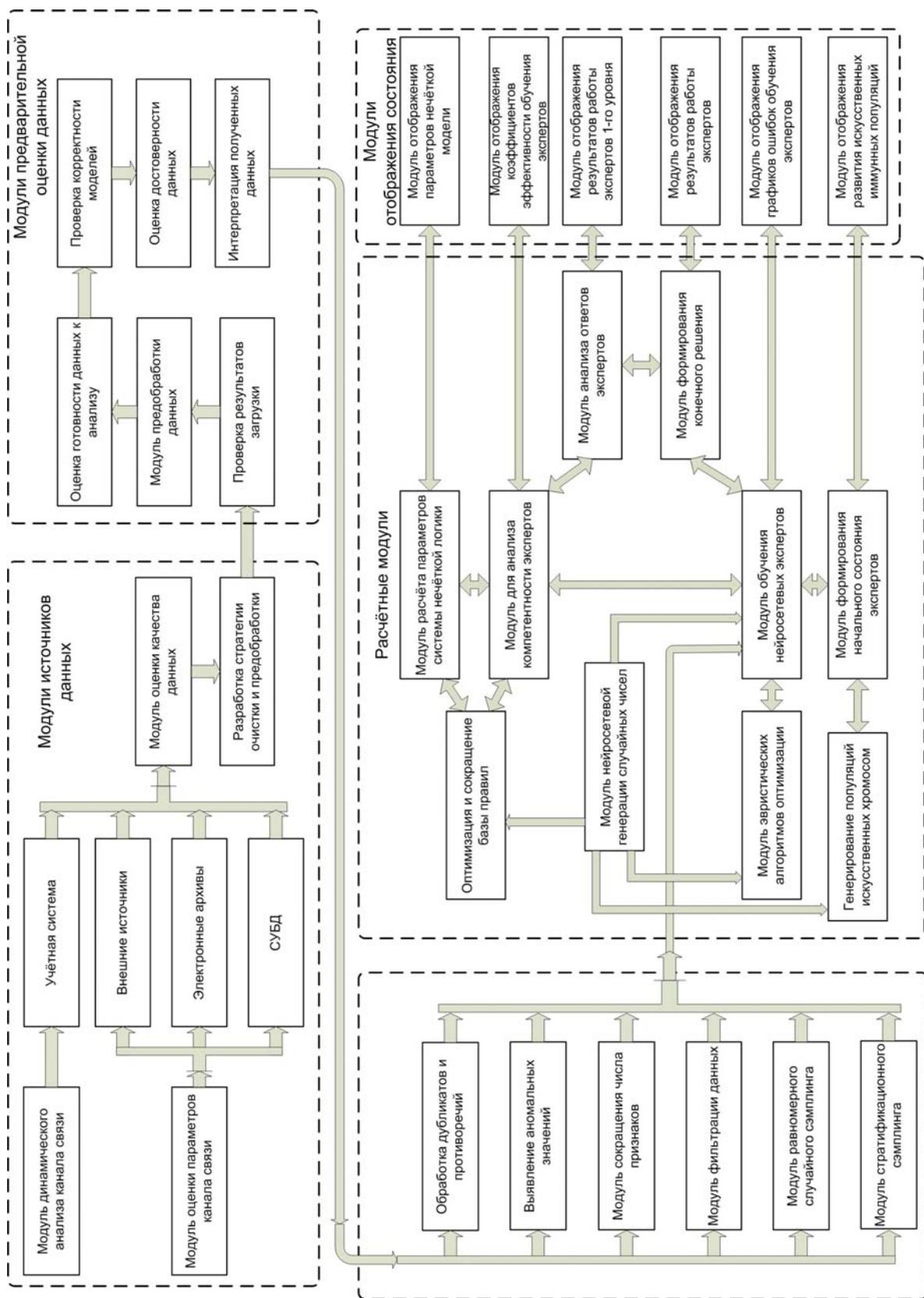


Рисунок 3.1 – Структурные модули разработанной системы

Разработанная система имеет сложную внутреннюю архитектуру с развитым механизмом взаимодействия нейросетевых элементов между собой. Возникает необходимость всестороннего анализа информации, на основании

которой будет произведён процесс обучения нейросетевых экспертов. Предобработка данных может быть рассмотрена как комбинация методов очистки и процедур оптимизации для решения конкретной задачи. В процессе очистки данных программные модули выполняют удаление векторов с нарушением структуры данных, отслеживают наличие пропусков, дубликатов, противоречий. Проблемы, возникающие при обработке некорректных данных, необходимо устранить на начальном этапе функционирования системы. В противном случае придётся выполнять сложный вычислительный процесс переобучения всех нейросетевых экспертов. Модули для предобработки данных и подготовки их к дальнейшему применению выполняют следующий набор операций:

- **Обработка дубликатов и противоречий.** Данные о состоянии вычислительной сети поступают от различных источников. Дубликаты представляют собой два идентичных вектора, характеризующих одно состояние системы. Такие вектора должны быть удалены. Векторы данных содержат противоречия, если они нарушают логику причинно-следственных связей: при одинаковых входных воздействиях система должна отреагировать различными значениями выходных сигналов. Наличие противоречий в системе значительно ухудшает процесс обучения НС [30]. Устранение противоречий требует проверки соответствия анализируемых данных предметной области.
- **Выявление аномальных значений.** Чаще всего аномалии в данных появляются из-за ошибок ввода. Если аномалии возникают в процессе получения исходных данных, то они также оказывают негативное влияние на систему, т.к. создают трудности в обнаружении закономерностей в анализируемом процессе. В процессе обучения НС производилось удаление векторов с аномальными значениями отдельных компонентов. Векторы, вызвавшие реакцию модуля, сохраняются для последующего выявления причин возникновения нестандартного поведения системы.
- **Сокращение числа признаков.** Для сокращения времени обучения НС из входных векторов необходимо выделять компоненты, исключение которых не повлияет на результаты работы системы. В качестве алгоритмов сжатия

применяется разработанный нейросетевой алгоритм (раздел 2.7) и карты Кохонена.

- **Модуль фильтрации данных.** Не все получаемые сведения необходимы для решения поставленной проблемы. Между входными и выходными значениями должна быть высокая степень взаимосвязи. Путём выявления векторов, которые оказывают слабое влияние на процесс обучения, выделяют данные, которые могут быть удалены без существенного уменьшения качества принимаемых решений.

- **Модули равномерного случайного и стратификационного сэмплинга.** Данные модули предназначены для конечного формирования обучающей и тестовой выборки. Когда исходные данные характеризуют определённую часть предметной области и достаточно однородны, система применяет равномерный случайный сэмплинг [41]. Все сведения разделяются на группы и из каждой группы выбирают записи, для размещения их в результирующих обучающих выборках.

Если исходные данные неоднородны, производится разбиение на относительно однородные подгруппы. Затем запускается процедура случайного сэмплинга [100].

На текущем этапе работы система получает сформированную обучающую выборку. На основании полученной выборки становится возможным произвести конфигурацию всех нейросетевых элементов системы. В работу включаются *расчётные модули*. *Модуль обучения нейросетевых экспертов* включает в себя программную реализацию градиентных алгоритмов обучения: для персептрона – алгоритм обратного распространения ошибки, для радиально-базисной нейронной сети – градиентный алгоритм, для сети Эльмана – градиентный метод наискорейшего спуска. *Модуль эвристических алгоритмов оптимизации* реализует эвристические алгоритмы, которыми были модифицированы градиентные алгоритмы обучения нейросетевых экспертов: алгоритм случайного поиска с переменным шагом, комбинаторный эвристический алгоритм упреждающего поиска, эвристический алгоритм на основе метода комплексов.

Модуль формирования начального состояния экспертов содержит программную реализацию кооперативного иммунного алгоритма, который применяется для настройки начального состояния всех нейросетевых структур перед процедурой обучения. Поддержание разнообразия решений и переход в новые точки поискового пространства осуществляются с помощью модуля *генерирования популяций искусственных хромосом*. Процесс получения финального решения в виде оптимального направления передачи информации основан на использовании разработанной нечёткой системы оценки компетентности экспертов, расположенной в *модуле для анализа компетентности экспертов*. Два модуля: модуль расчёта параметров системы нечёткой логики и модуль оптимизации и сокращения базы правил содержат программную реализацию алгоритмов настройки и оптимизации параметров нечёткой системы.

После формирования решения всеми нейронными сетями производится обработка их выходных сигналов с помощью трёх модифицированных нейронов: Фукушимы, N – адалины и квадратичного нейрона. Программная реализация нейронов находится в *модуле анализа ответов экспертов*. *Модуль формирования конечного решения* представлен программной реализацией модели сигма-пи сети с разработанным алгоритмом обучения на основе комбинации алгоритмов случайного поиска. Результатом функционирования модуля является финальное решение о следующем пути в маршруте продвижения информации.

Разработанный программный продукт содержит набор *модулей отображения состояния*, которые предназначены для графического изображения результатов работы алгоритмов. Модуль способен строить графики среднеквадратических ошибок обучения нейронных сетей, отображать аппроксимацию графика с помощью метода наименьших квадратов. После получения информации о сети становится возможным отображение её в виде графа: узлы соответствуют компьютерам в сети, а дуги характеризуют наличие дуплексной связи между ними. Существует возможность визуализации сигналов,

полученных от экспертов. Они изображаются в виде графика, на котором показаны состояния выходных нейронов в заданном временном интервале.

Программная среда обеспечивает связь между модулями и их успешное совместное функционирование, благодаря которому система способна решать поставленную задачу.

3.3 Особенности реализации инструментальной среды

Рассмотрим основные возможности системы, начав с реализации алгоритма инициализации начального состояния (рисунок 3.2). В программе существует возможность выбора операторов иммунного алгоритма: операторы одноточечной и двухточечной мутации, операторы инверсии, Cataclysmic mutation, транслокации. После выбора комбинационных процедур для получения нового решения производится конфигурирование параметров иммунной популяции: количество антител, процент решений для клонирования и замены, количество итераций работы алгоритма. Клонирование антител в процессе работы алгоритма предполагает применение двух стратегий:

1. Клонирование группой – осуществляется отбор антител в интервале от $\min_{-}Aff$ до $0,2 \cdot \min_{-}Aff$ ($\min_{-}Aff$ – антитело с минимальной аффинностью). Отобранные антитела подвергаются копированию до достижения популяцией заданной численности, с удалением заменяемых антител, не вошедших в группу.

2. Клонирование пропорционально аффинности – производится сортировка антител по возрастанию функции аффинности. Каждому антителу сопоставляется значение вероятности копирования в новую популяцию обратно пропорционально значению аффинности. На основе полученных вероятностей происходит формирование новой совокупности решений.

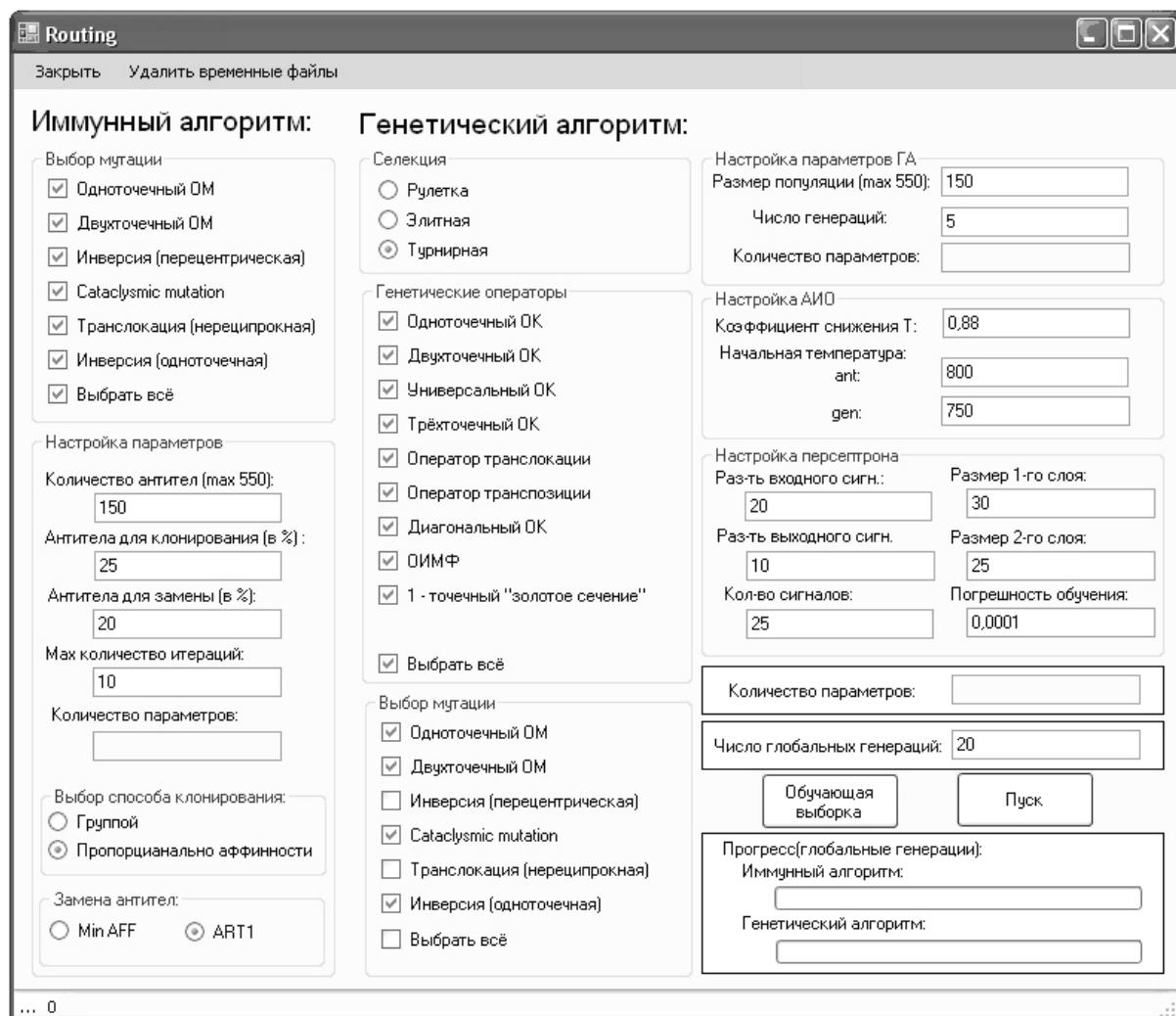


Рисунок 3.2 – Окно для настройки кооперативного иммунного алгоритма

Для настройки генетического алгоритма, с помощью которого производится генерация антител, пользователь определяет тип селекции хромосом: рулетка, элитная, турнирная. Способ модернизации и получения новых решений основан на применении генетических операторов. В программе предусмотрено использование следующих операторов: одноточечный, двухточечный, универсальный, трёхточечный, диагональный операторы кроссинговера, операторы транслокации, транспозиции и т.д. Изменение решений осуществляется с помощью набора операторов мутации: одноточечный, двухточечный, операторы инверсии, транслокации и т.д. После определения количества хромосом и числа генераций необходимо выбрать параметры метода имитации отжига для контроля развития эволюционных алгоритмов популяции. Основным параметром является начальная температура отжига для генетического

и иммунного алгоритма. Затем определяется коэффициент уменьшения температуры, определяющий развитие алгоритмов.

Перед запуском алгоритма настройки первоначального состояния необходимо определить конфигурацию нейронных сетей и количество оптимизируемых параметров. Размерность вектора входного и выходного сигнала является фиксированной величиной, определяемой условиями решаемой задачи. Параметры конфигурации многослойного персептрона задаются пользователем. Если достичь заданного минимума ошибки не удалось, то программа увеличит количество нейронов. Число нейронов в скрытом слое нейронной сети РБФ определяется алгоритмом кластеризации k – средних. Полностью сформированная архитектура нейросетевых экспертов позволяет выполнить подсчёт переменных параметров в ассоциативной машине и начать процедуру оптимизации.

Перед стартом алгоритма необходимо задать число глобальных генераций. Каждая глобальная генерация включает в себя заданное количество итераций для иммунного и генетического алгоритма.

В разделе 2.7 выполнена разработка алгоритма анализа характеристик канала передачи данных, а также приведён метод компрессии данных с помощью карты Кохонена. Разработанная программная система может выполнять построение карт с максимальным размером 40x40 нейронов. Пользователь может уменьшить размерность карты с учётом требуемой предобработки входных данных для комитета экспертов. При построении карт нельзя допустить объединения или дробления классов, которым соответствуют плотно распределённые входные образы. Для решения этой проблемы, программа содержит реализацию метода выпуклой комбинации [71]. Для возможности управления режимом построения карты реализовано два способа активации нейронов в сети Кохонена: режим аккредитации и режим интерполяции [61]. При построении карты в первом режиме производится активация только одного выходного нейрона, сигналы других подавляются. Режим интерполяции допускает наличие нескольких «победителей». При наличии большого количества исходных данных необходимо обеспечить равномерную активацию нейронов

карты Кохонена. Данная процедура необходима для того, чтобы избежать появления «мёртвых» нейронов, которые никогда не будут активироваться. Введение процедуры «штрафования» часто активизируемых нейронов, будет приводить к применению всех доступных нейронов сети Кохонена для анализа данных. Пример возможной карты для максимальной конфигурации сети показан на рисунке 3.3. Для удобства анализа карты, производится раскрашивание карты. В нижней части формы расположена цветовая полоса. Крайнему левому значению цвета соответствует минимальное значение выходы нейрона, а крайнему правому – максимальное значения. Нейронам присваиваются цвета на основе значений их выходных сигналов.

Формирование топологии нейронных сетей осуществлялось в автоматическом режиме путём анализа изменения среднеквадратической ошибки. Функционирование началось на основе базовых моделей, приведённых в разделе 2.2. После предъявления примеров обучающей выборки производилось усиление топологии за счёт добавления по одному нейронному элементу в каждый слой. Если происходило изменение среднеквадратической ошибки в сторону увеличения, то изменения отменялись постепенно: производилось удаление по одному нейрону последовательно из каждого слоя НС, до достижения приемлемых критериев качества обучения.

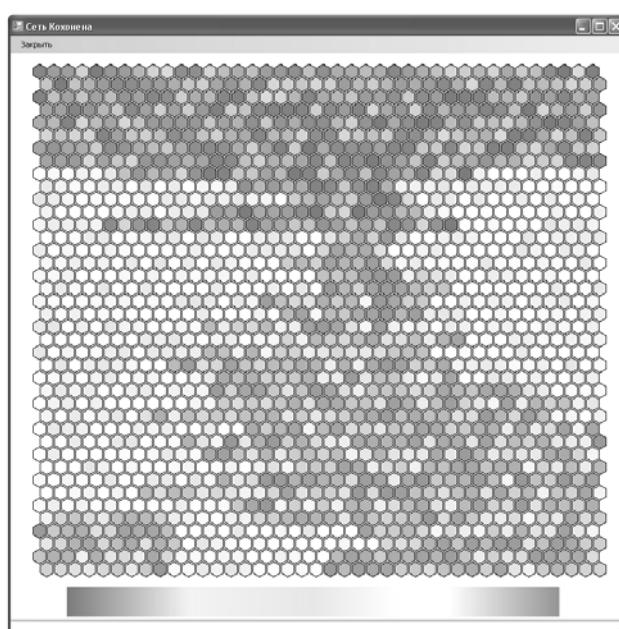


Рисунок 3.3 – Пример функционирования сети Кохонена

Программная система может выполнять построение графиков среднеквадратичной ошибки обучения экспертов, а также выполнять аппроксимацию полученной зависимости с помощью метода наименьших квадратов (рисунок 3.4).

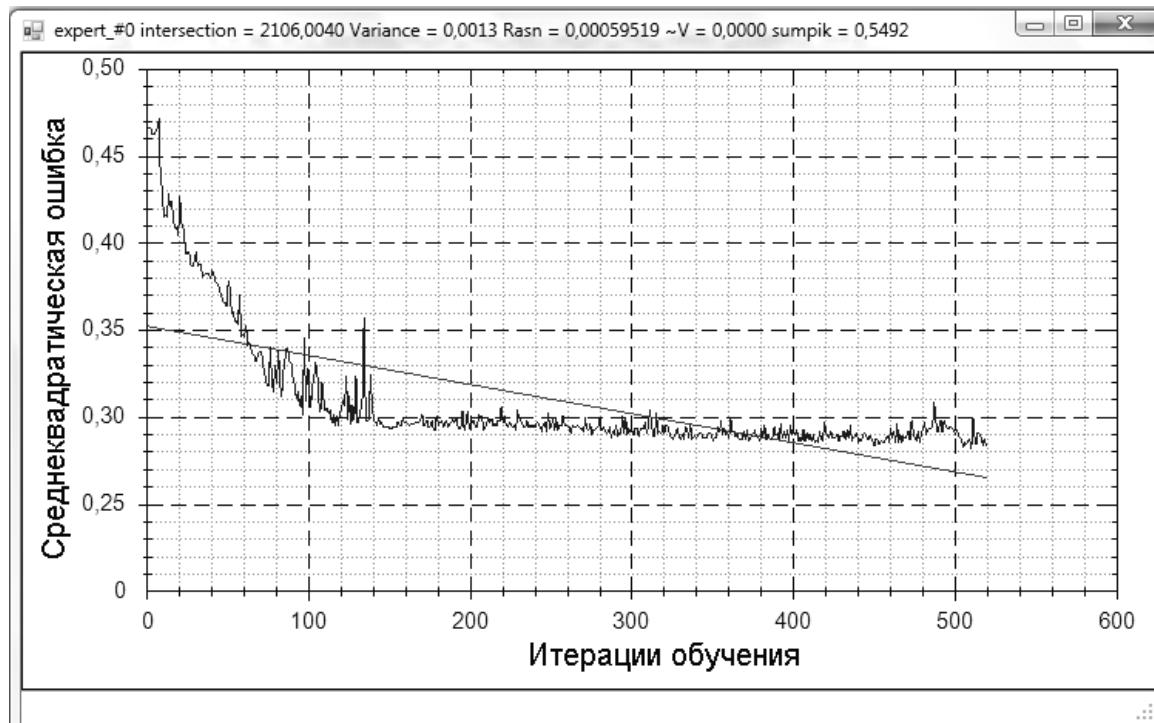


Рисунок 3.4 – Пример вывода результатов обучения НС

Программа предназначена для определения оптимальных маршрутов и их коррекции в таблицах маршрутизации отдельных узлов. Для определения оптимальных маршрутов система должна знать общее состояние сети. Алгоритм анализа канала связи, разработанный в разделе 2.7 позволяет всем узлам сформировать векторы, характеризующие исходящие соединения. Полученные векторы должны пересыпаться по сети и добавляться к локальным векторам узлов. После получения всех векторов соединений и обработки их с помощью нейронной сети Кохонена, в каждом узле будет сформирована информация о состоянии компьютерной сети, необходимая для прокладки безопасного маршрута. Для обмена данными между узлами была применена клиент-серверная технология (рисунок 3.5). В каждом узле серверная часть приложения ожидает подключения клиентских модулей других узлов для передачи информации о состоянии своих сетевых соединений.

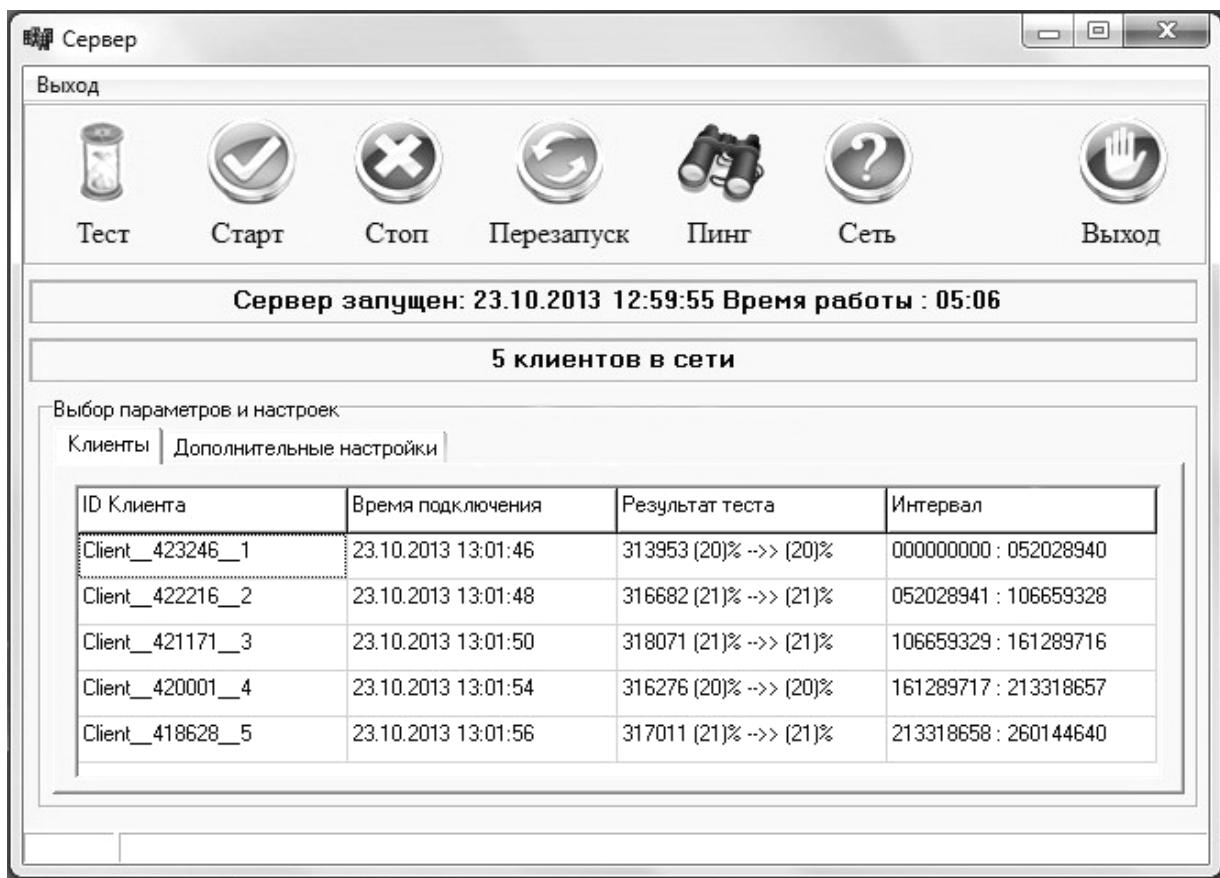


Рисунок 3.5 – Окно серверной части приложения

В окне серверной части отображаются идентификаторы (*ID*) подключенных клиентских частей других узлов, время подключения, результат теста, интервал. Интервал присваивается индивидуально каждому подключенному клиенту для идентификации поступающих сообщений. Для оптимизации маршрутов в сети следует учитывать не только возможности каналов связи, но и произвести оценку вычислительной мощности узла, который будет обеспечивать функционирование нейросетевого алгоритма безопасной маршрутизации. При синтезе оптимальных маршрутов необходимо оптимально распределить имеющиеся вычислительные возможности узлов. Для оптимального включения узлов в маршруты ассоциативной машине необходимо иметь представление об арифметических способностях клиентов. После подключения клиентских машин, сервер отдаёт команду на выполнение тестирования. Тестовый модуль представляет собой алгоритм, который при запуске организует пять потоков вычислений значений функций двух переменных. В каждом потоке организуется примерно около

1 000 000 вычислений значений функций при случайной инициализации её аргументов. В качестве функций для арифметического теста выбраны следующие:

1. Мультипликативная потенциальная функция.

$$I(x_1, x_2) = -z(x_1)z(x_2), z(x) = -\frac{1}{(x-1)^2 + 0,2} - \frac{1}{2(x-2)^2 + 0,15} - \frac{1}{3(x-3)^2 + 0,3} \quad (3.1)$$

2. Функция Розенброка.

$$I(x, y) = 100 \cdot (y - x^2)^2 + (1 - x)^2 \quad (3.2)$$

3. Функция Растрогина овражная с поворотом осей.

$$I(x, y) = (0,1 \cdot K_x \cdot A)^2 + (0,1 \cdot K_y \cdot B)^2 - 4 \cdot \cos(0,8 \cdot K_x \cdot A) - 4 \cdot \cos(0,8 \cdot K_y \cdot B) + 8 \quad (3.3)$$

где $A = x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$, $B = x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$, K_x, K_y - растяжение/сжатие по x, y , α - угол поворота, $\alpha = \pi/2$, $kx = 1,5, ky = 0,8$;

4. Функция Griewank.

$$I(x, y) = \frac{-10}{0,005 \cdot (x^2 + y^2) - \cos(x) \cdot \cos(\frac{y}{\sqrt{2}}) + 2} + 10 \quad (3.4)$$

5. Функция «Сомбреро».

$$F(x, y) = \frac{1 - \sin^2(\sqrt{x^2 + y^2})}{1 + 0,001 \cdot (x^2 + y^2)}. \quad (3.5)$$

После выполнения всех вычислений, клиенты передают серверу данные о времени выполнения расчетов в миллисекундах. Сервер собирает все данные и, после получения полной информации о подключенных к нему клиентских системах, производит расчёт коэффициента арифметической мощности. Алгоритм расчёта представим в следующем виде:

1. Получить общее время выполнения арифметических тестов на ПК.

$$SUM = \sum_{i=1}^N \sum_{j=1}^M t_{ij}, \quad (3.6)$$

где SUM – общее время тестов, N – количество клиентов, подключенных к серверу, M – количество потоков тестирования на конкретном клиенте, t_{ij} - время выполнения теста на i клиенте на j потоке.

2. Получить процентное соотношение времени выполнения на каждом компьютере:

$$P_i = \sum_{j=1}^M t_{ij} / SUM \cdot 100\% , \quad (3.7)$$

где P – процент времени теста компьютера i клиента от общего теста.

3. Полученные во 2 пункте проценты показывают, сколько времени от общего количества потратил клиентский компьютер на тестирование. Чем меньше процент, тем лучше арифметическая база клиентского компьютера. Данный коэффициент используется как один из параметров, определяющих построение безопасного маршрута.

Кнопка «Пинг» позволяет протестировать соединение с удалённым компьютером (рисунок 3.6). Для этого пользователь вводит адресата, время ожидания, время жизни пакета. После этого система проверяет достижимость указанного адресата с компьютера сервера.

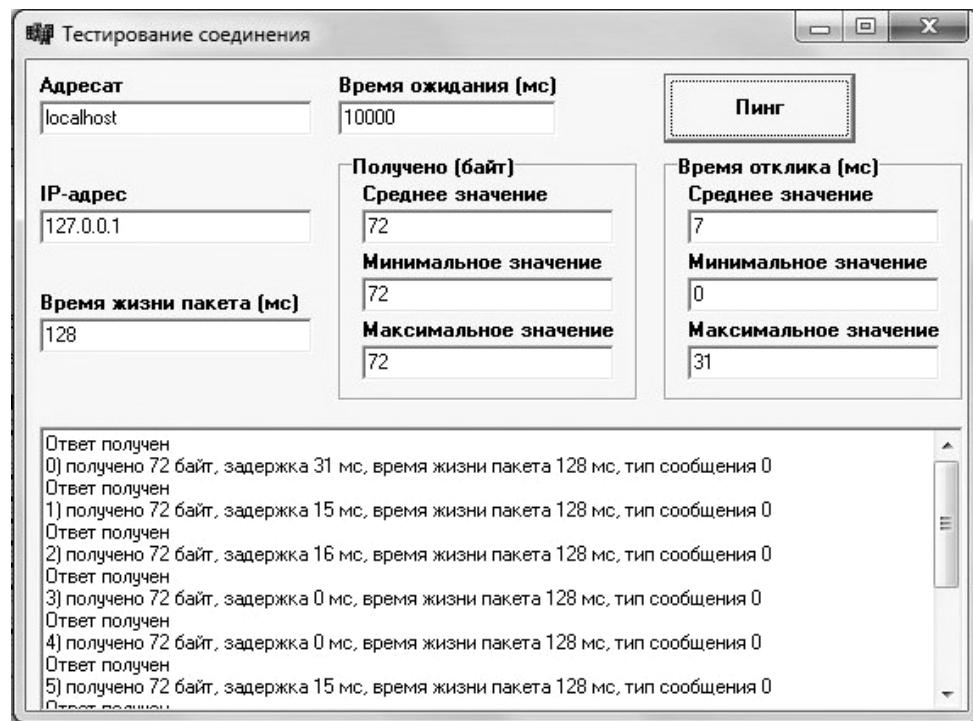


Рисунок 3.6 – Окно инструмента «Пинг»

Кнопка «Сеть» позволяет оценить качество соединений между клиентами и сервером, для оценки времени прохождения сообщений. Пользователь указывает адресата и байтовую строку, которую необходимо послать для проверки. После

выполнения запроса, программа отображает строку, которая вернулась, с целью контроля её целостности и время ответа на запрос (рисунок 3.7).

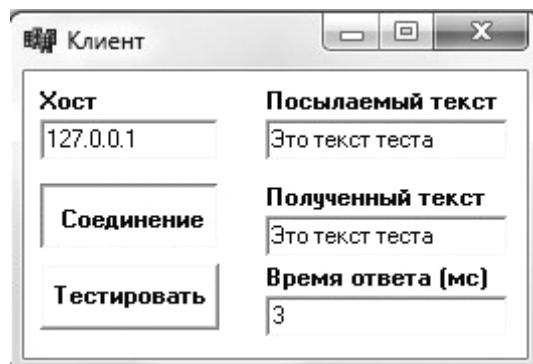


Рисунок 3.7 – Результат выполнения теста

Серверная часть разработанной программной системы может выводить полученные данные о состоянии соединений на экран для их визуального или статистического анализа. На рисунке 3.8 показан процесс вывода информации: сначала выводится идентификатор посылаемого сообщения, а затем полученное 16 байтное сообщение в шестнадцатеричной системе счисления.

Рисунок 3.8 – Полученные данные о состоянии каналов передачи

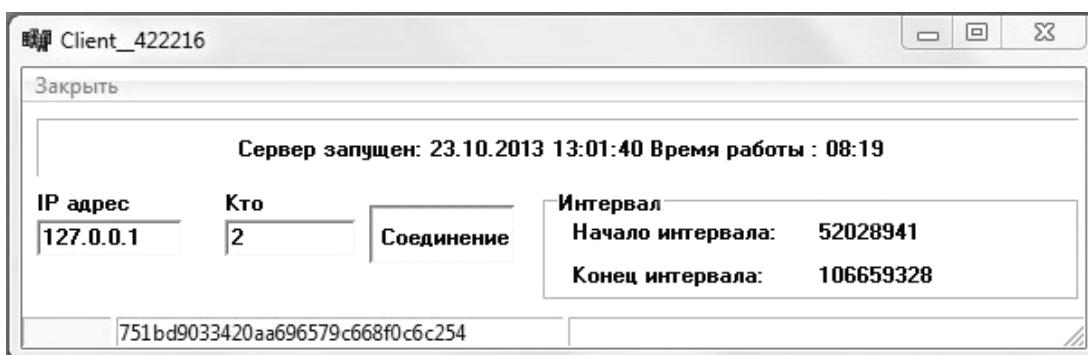


Рисунок 3.9 – Окно клиентской части приложения

Клиентская часть приложения отвечает за передачу данных и проведения тестирования вычислительных возможностей узла (рисунок 3.9).

В разделе 2.7 был приведён разработанный алгоритм анализа состояния каналов передачи информации на основе детектирования изменения параметров ёмкости, сопротивления, а также оценки спектра сигнала, прошедшего через канал связи. Для возможности обработки информации с помощью нейронных сетей разработанный программный продукт включает в себя программные модули для получения информации от аппаратной части. В качестве основного устройства сопряжения был использован мост USB-UART (UART – универсальный синхронно/асинхронный приёмо-передатчик микроконтроллера *ATmega32A*, который выполнял основные управляющие функции) на основе микросхемы *CP2110*. На рисунке 3.11 показано окно приложения, в котором выводится частота одного из генераторов *NE555*, применяемого для определения изменения параметров канала связи. В первом столбце отображается номер отсчёта полученного пакета (10 байт). Во втором – время получения. Затем программа показывает содержимое пакета и частоту соответствующего генератора. Полученная информация используется при построении элементов обучающей выборки. Аналогично передаётся информация от разработанного в разделе 2.6 источника энтропии для нейросетевого генератора случайных чисел. Реализация источника включает в себя дисплей (2 строки на 16 символов), который производит отображение сгенерированных последовательностей (рисунок 3.10).

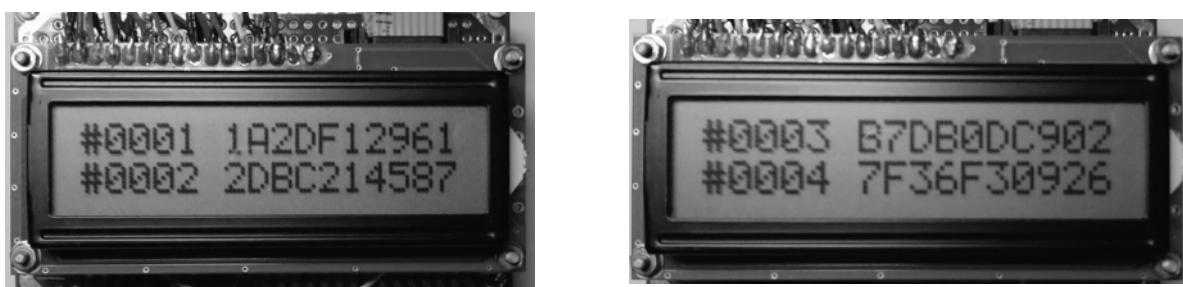


Рисунок 3.10 – Функционирование источника энтропии

#	Время	Принято	ASCII
000001	11:56:50.562	30 30 30 30 32 35 34 31 37 39	0000254179
000002	11:56:51.937	30 30 30 30 32 34 38 34 34 32	0000248442
000003	11:56:53.312	30 30 30 30 32 35 31 37 30 38	0000251708
000004	11:56:54.671	30 30 30 30 32 35 35 34 39 36	0000255496
000005	11:56:56.046	30 30 30 30 32 34 35 35 34 36	0000245546
000006	11:56:57.421	30 30 30 30 32 35 35 30 39 34	0000255094
000007	11:56:58.781	30 30 30 30 32 34 39 34 37 33	0000249473
000008	11:57:00.156	30 30 30 30 32 34 39 37 35 31	0000249751
000009	11:57:01.515	30 30 30 30 32 35 31 33 33 33	0000251333
000010	11:57:02.890	30 30 30 30 32 35 31 35 37 36	0000251576
000011	11:57:04.265	30 30 30 30 32 35 35 32 34 39	0000255249
000012	11:57:05.625	30 30 30 30 32 35 30 36 33 32	0000250632
000013	11:57:07.000	30 30 30 30 32 35 34 35 38 34	0000254584
000014	11:57:08.359	30 30 30 30 32 34 38 34 37 39	0000248479
000015	11:57:09.734	30 30 30 30 32 35 31 36 36 36	0000251666
000016	11:57:11.109	30 30 30 30 32 35 32 35 31 30	0000252510
000017	11:57:12.468	30 30 30 30 32 34 34 34 30 36	0000244406
000018	11:57:13.843	30 30 30 30 32 35 33 37 39 31	0000253791
000019	11:57:15.218	30 30 30 30 32 35 35 30 35 35	0000255055
000020	11:57:16.578	30 30 30 30 32 34 38 39 34 33	0000248943

Рисунок 3.11 – Информация от аппаратной части

Для проверки работоспособности алгоритма программная система содержит комплекс методов для моделирования работы локальной компьютерной сети. Подсистема необходима для отладки работы алгоритмов обучения нейронных сетей, и оценки их способностей синтезировать оптимальные маршруты. Модель компьютерной сети может быть задана на основе анализа реальной сети, в которой предполагается функционирование разработанного нейросетевого алгоритма безопасной маршрутизации. Другой способ синтеза топологии модели вычислительной сети основан на случайном формировании связей между узлами на основе информации, заданной пользователем (рисунок 3.12).

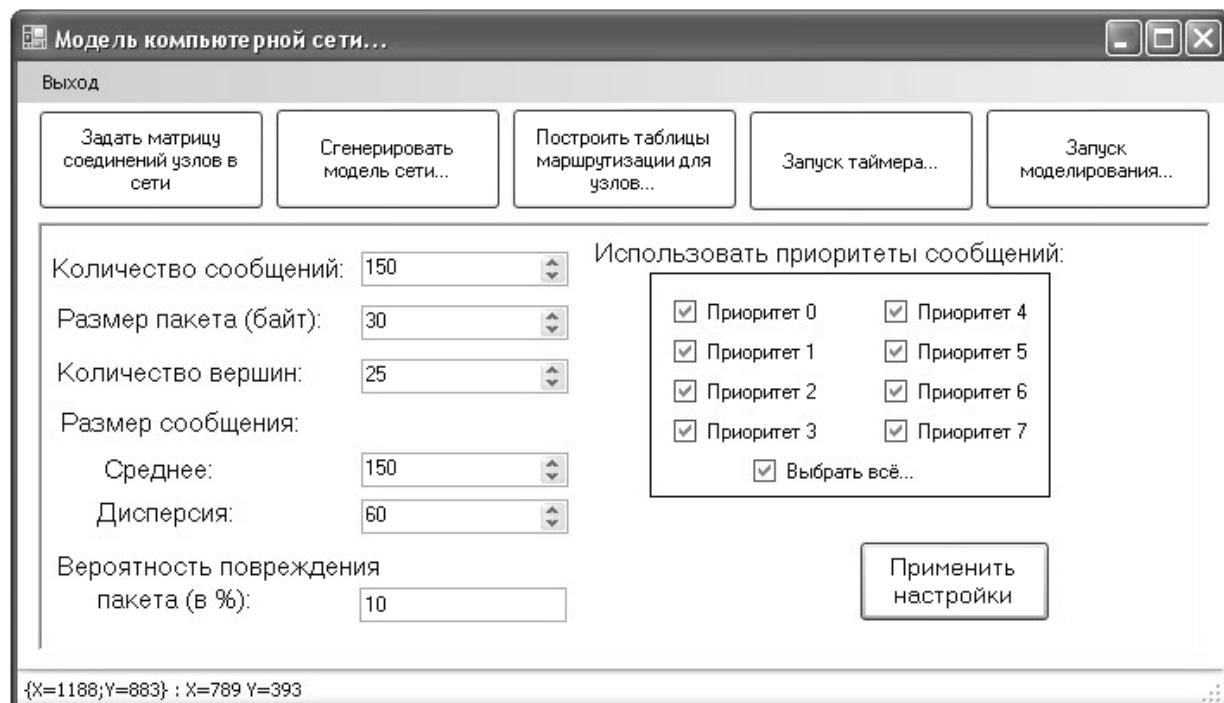


Рисунок 3.12 – Форма настройки модели компьютерной сети

Пользователь определяет количество узлов в сети, число сообщений, пересылаемых по сети во время процесса моделирования, размер сообщения в байтах. Каждое сообщение при пересылке разбивается на пакеты, размер которых пользователь может установить. В программе возможна установка приоритетов сообщений: данные с наибольшим приоритетом обрабатываются в первую очередь. После задания основных параметров модели необходимо определить топологию сети, определив соединения между узлами (рисунок 3.13). Выполнив все процедуры настройки, модель строит компьютерную сеть заданной топологии (рисунок 3.14). Сформировав сеть, программная среда запускает нейросетевой алгоритм построения оптимальных маршрутов. В результате в каждом узле формируется таблица маршрутизации (рисунок 3.15), которая имеет следующий формат:

Узел назначения	Следующий узел для передачи информации
-----------------	--

В системе существует возможность альтернативного построения маршрута с помощью алгоритма Флойда, для оценки эффективности разработанного алгоритма маршрутизации. В процессе моделирования производится отображение информации об отправленных и принятых пакетах, временные интервалы

создания и удаления пакета, информация, необходимая для правильной сборки пакета в узле назначения и контроля его целостности (рисунок 3.16).

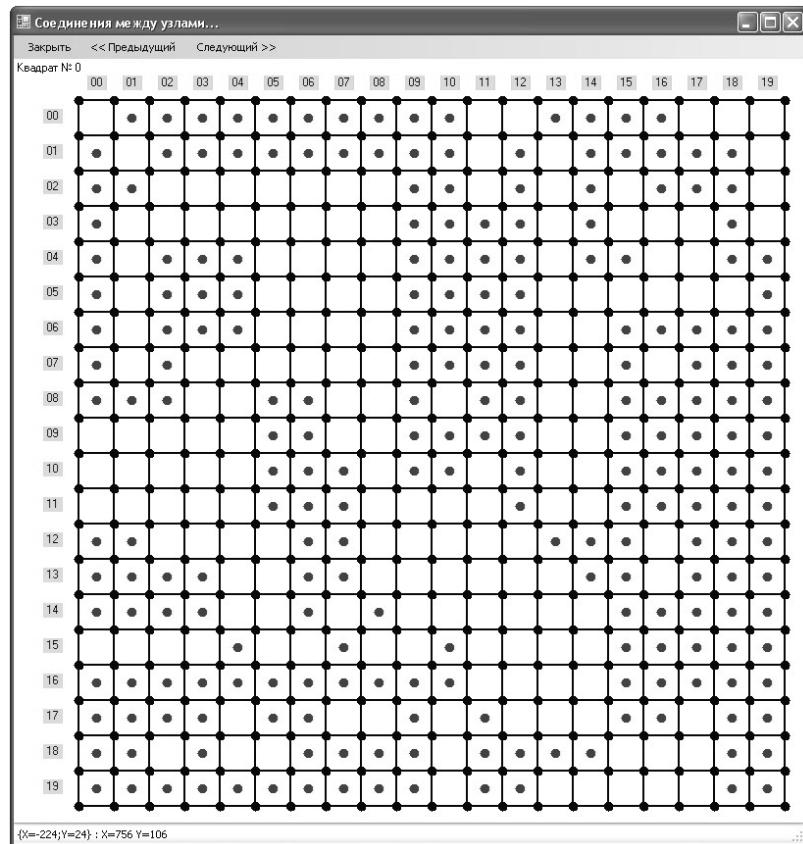


Рисунок 3.13 – Окно для формирования связей между узлами

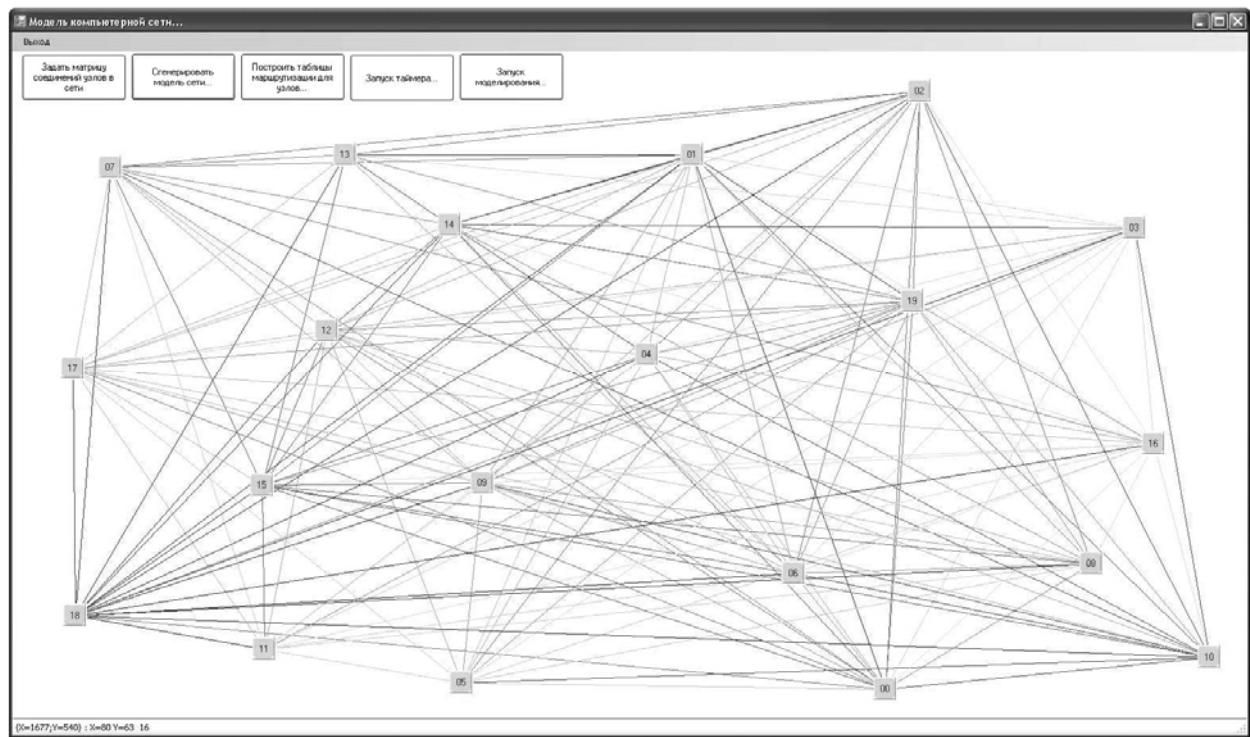


Рисунок 3.14 – Сгенерированная модель компьютерной сети

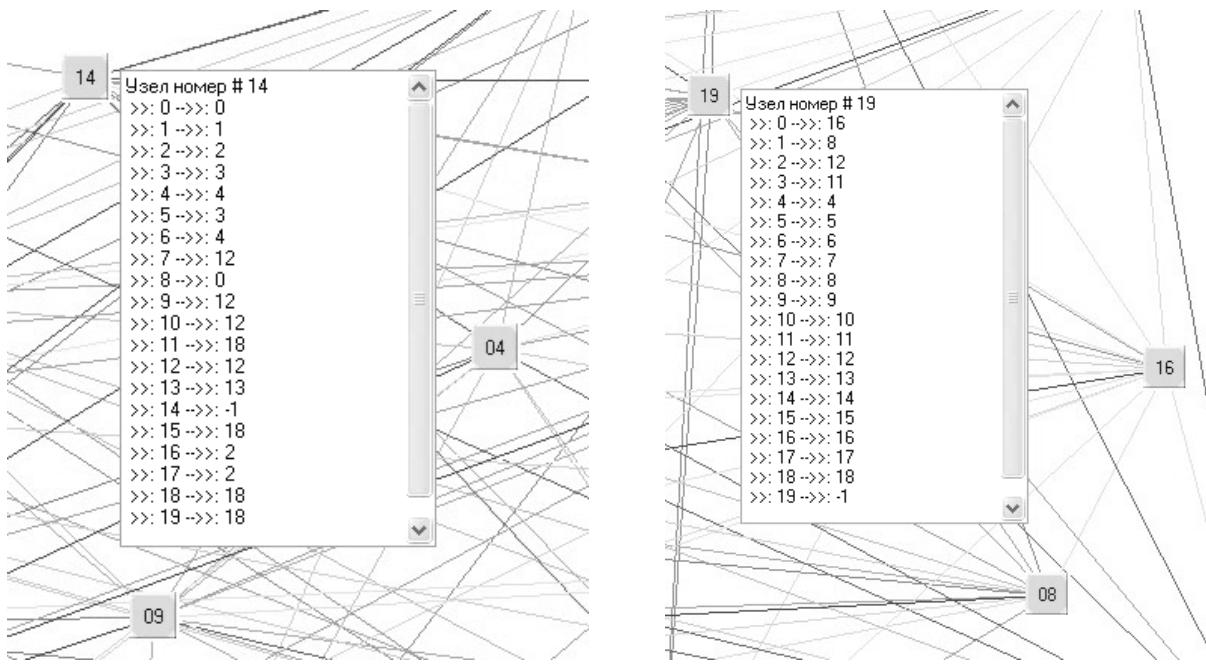


Рисунок 3.15 – Пример формирования таблицы маршрутизации

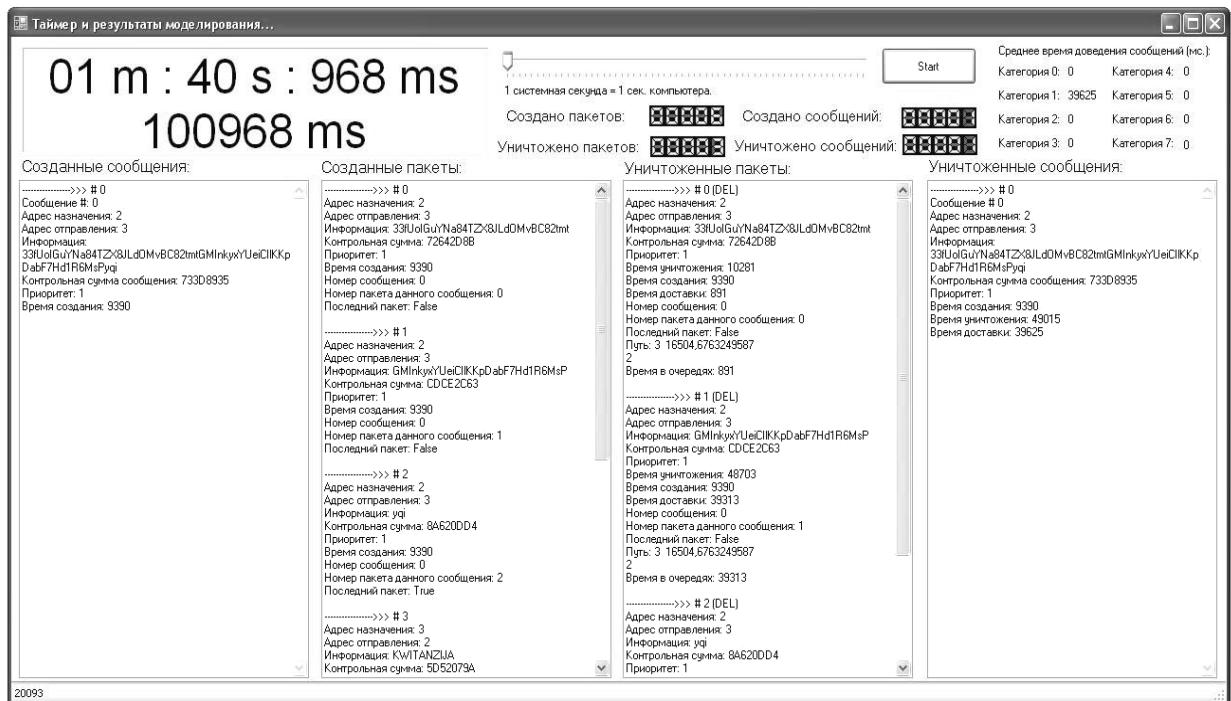


Рисунок 3.16 – Результат работы системы моделирования

3.4 Перспективы развития программной системы

- Обеспечение возможности подключения дополнительных нейросетевых модулей для расширения интеллектуальных способностей системы принимать решения о безопасных маршрутах продвижения информации.
- Увеличение количества применяемых эвристических методов для увеличения эффективности градиентных алгоритмов обучения НС.

3. Применение комплекса нейронных сетей для получения финального решения. Такой способ организации ассоциативной машины, включающей множество экспертов не только на этапе обработки данных множеством нейронных сетей, но и на стадии получения финального решения, позволит параллельно выполнять построение нескольких альтернативных маршрутов.

4. Замена нечёткой системы оценки параметров компетентности экспертов (НСОПКС), базирующейся на геометрическом представлении функций принадлежности нечётких множеств, нечёткой нейронной сетью. Несмотря на усложнение внутренней структуры НСОПКС, данная модификация позволит применять развитый аппарат алгоритмов обучения для нейронных сетей.

5. Введение в систему физической возможности поддержания методов анализа для оптических и беспроводных сред передачи информации. Данная модификация позволит проводить анализ сетей связи, обладающих разнообразными каналами передачи информации на различных этапах.

6. Добавление дополнительных источников энтропии в нейросетевой генератор случайных чисел, позволит сделать процесс развития и обучения системы более непредсказуемым, чем с применением одного источника. Предполагаемая модификация обеспечит более высокую стойкость системы к различного рода атакам со стороны третьих лиц.

3.5 Выводы по главе 3

1. Выполнена разработка структурной схемы программного обеспечения, приведено описание основных блоков и способ взаимодействия между ними.

2. Разработана программная инструментальная среда, реализующая спроектированные алгоритмы настройки параметров ассоциативной машины с помощью градиентных методов, модифицированных эвристическими алгоритмами.

3. Система, на основе проведённого анализа компьютерной сети, способна выполнить построение её модели для анализа эффективности функционирования обученного комитета нейросетевых экспертов.

4. Реализована возможность взаимодействия отдельных компонентов программной системы, запущенной на разных узлах сети, которая позволяет оперативно получать информацию об изменении состояния каналов передачи.

5. Организован способ связи программной системы с разработанным аппаратным обеспечением, выполняющим анализ состояния каналов передачи информации. Модуль нейросетевого ГСЧ также получает информацию от разработанного источника энтропии.

4 Исследование разработанных алгоритмов и оценка их эффективности при решении практических задач

4.1 Анализ алгоритмов обучения экспертов

В данном разделе рассмотрим изменение динамики процесса обучения нейросетевых экспертов при использовании эвристик, спроектированных в разделе 2.2. Для определения влияния различного рода комбинированных эвристических процедур на процесс обучения нейронных сетей с различной топологией рассмотрим трансформацию временных зависимостей, отражающих изменение среднеквадратических ошибок обучения НС во времени. Получение графиков ошибок осуществлялось на основе обучающих выборок, принцип построения которых рассмотрен в 2.7. В качестве эталонных значений среднеквадратических ошибок (СКО) рассмотрим графики, полученные с помощью стандартных алгоритмов обучения для различных нейронных сетей, рассмотренных в разделе 1.2.

Общий алгоритм обучения для нейросетевых экспертов представим в следующем виде:

1. Выполнить процедуру инициализации синаптических весов и параметров функций активации с помощью алгоритма, описанного в разделе 2.3.
2. Предъявить примеры обучающей выборки. В нейронные сети экспертов подаются образы из обучающего множества. Для каждого примера последовательно выполняются два прохода по сети: прямой и обратный [38].
3. При прямом проходе на рецепторы НС необходимо подать входной вектор. Произвести вычисление откликов всех нейронов и реакцию сети на входное воздействие.
4. Вычислить сигнал ошибки, представляющий собой разницу между желаемым значением выходного вектора сети и полученным вектором.
5. Выполнить обратный проход по нейронной сети. Произвести коррекцию весовых коэффициентов и параметров функции активации нейронов.
6. Последовательно продолжать выполнять прямой и обратный проход, предъявляя сети все примеры из обучающей выборки. Процесс обучения

необходимо продолжать до тех пор, пока среднеквадратическая ошибка сети не достигнет желаемого значения.

Для оценки эффективности применения различных эвристических процедур при обучении НС используются два графика СКО: один для процесса обучения с помощью стандартного алгоритма, а другой отражает изменение ошибки обучения для нейронной сети той же архитектуры и с таким же количеством примеров обучающей выборки, но при этом в алгоритм обучения вводится эвристика. Для всех НС процесс обучения длится по 500 итераций. В данной работе такого количества итераций достаточно для достижения нейронной сетью минимума на поверхности ошибок.

На рисунке 4.1 приведён график среднеквадратической ошибки обучения для нейронной сети типа многослойный персептрон, обучение которой проводилось с помощью алгоритма, основанного на методе обратного распространения ошибки.

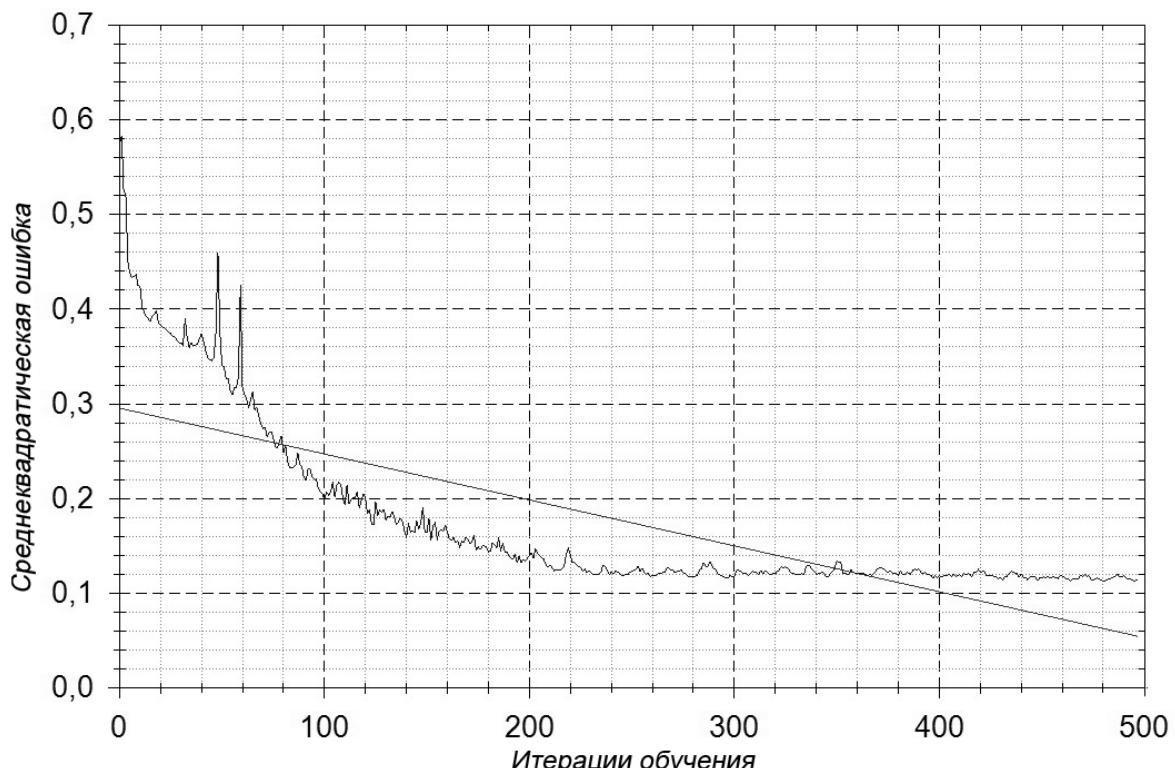


Рисунок 4.1 – Усреднённая кривая обучения для многослойного персептрана (алгоритм обратного распространения ошибки)

На рисунке 4.2 приведён график изменения среднеквадратической ошибки для многослойного персептрана, обучаемого с помощью алгоритма обратного

распространения ошибки, но с применением эвристики на основе алгоритма случайного поиска с переменным шагом.

Анализ полученных результатов показывает, что применение эвристики на основе алгоритма случайного поиска с переменным шагом позволяет снизить среднеквадратическую ошибку обучения многослойного персептрона и уменьшить величину колебаний ошибки на финальных стадиях обучения.

Процесс изменения ошибки обучения для нейронной сети РБФ, настраиваемой с помощью градиентного алгоритма, в которой начальная установка центров радиально-базисных нейронов производится алгоритмом k -средних, показан на рисунке 4.3.

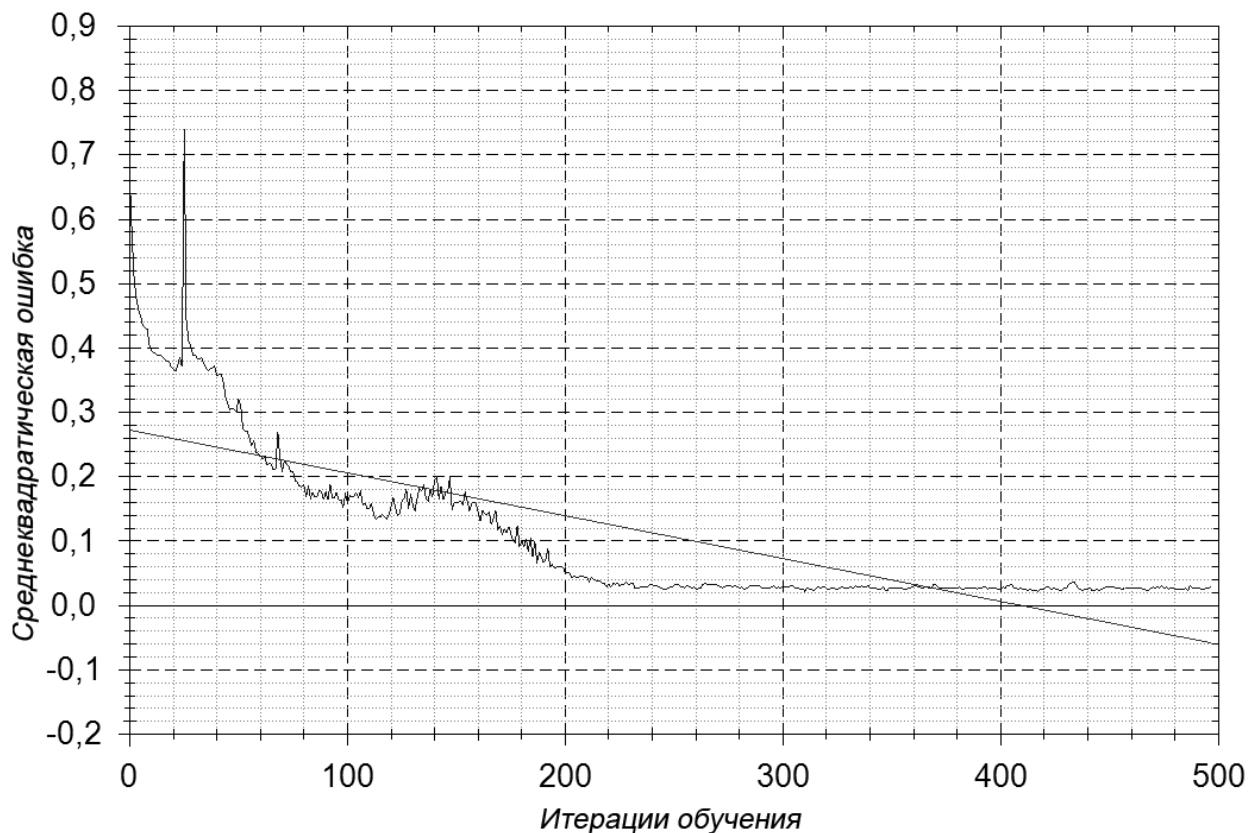
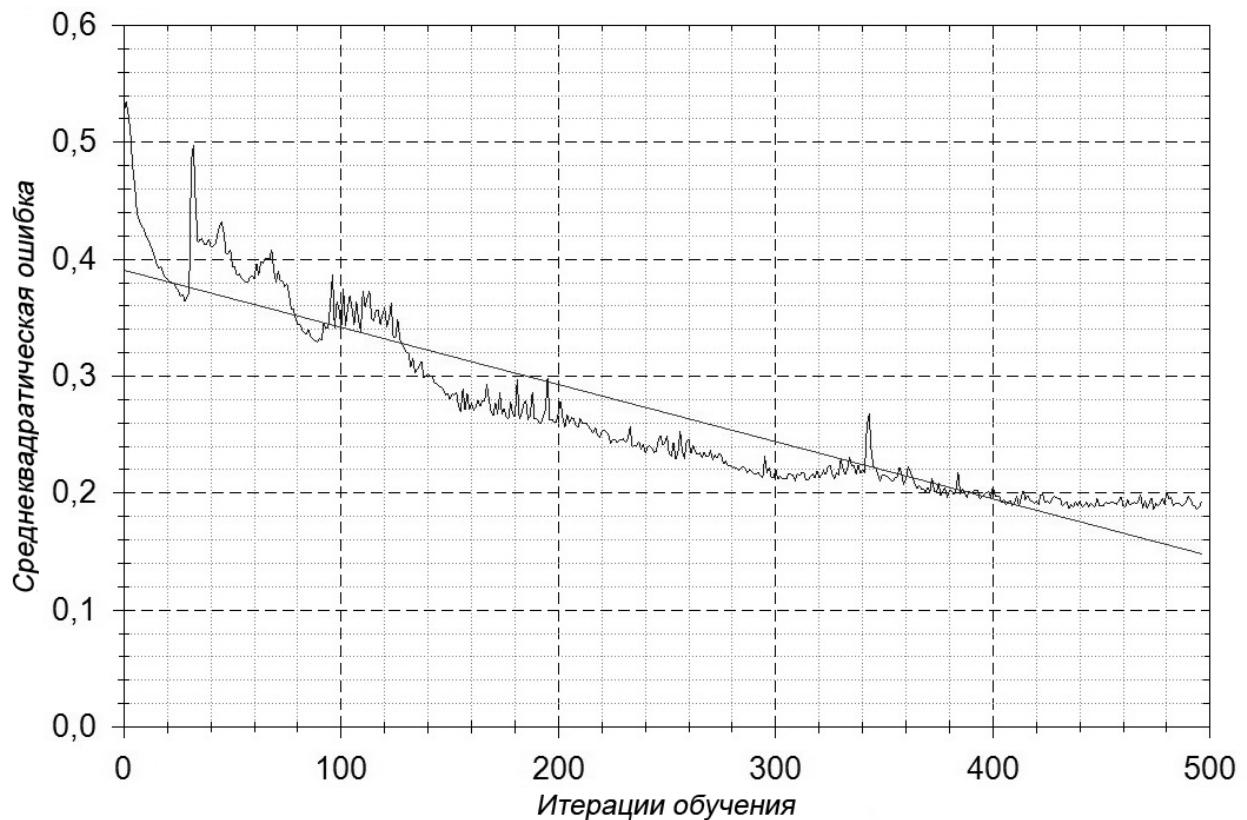
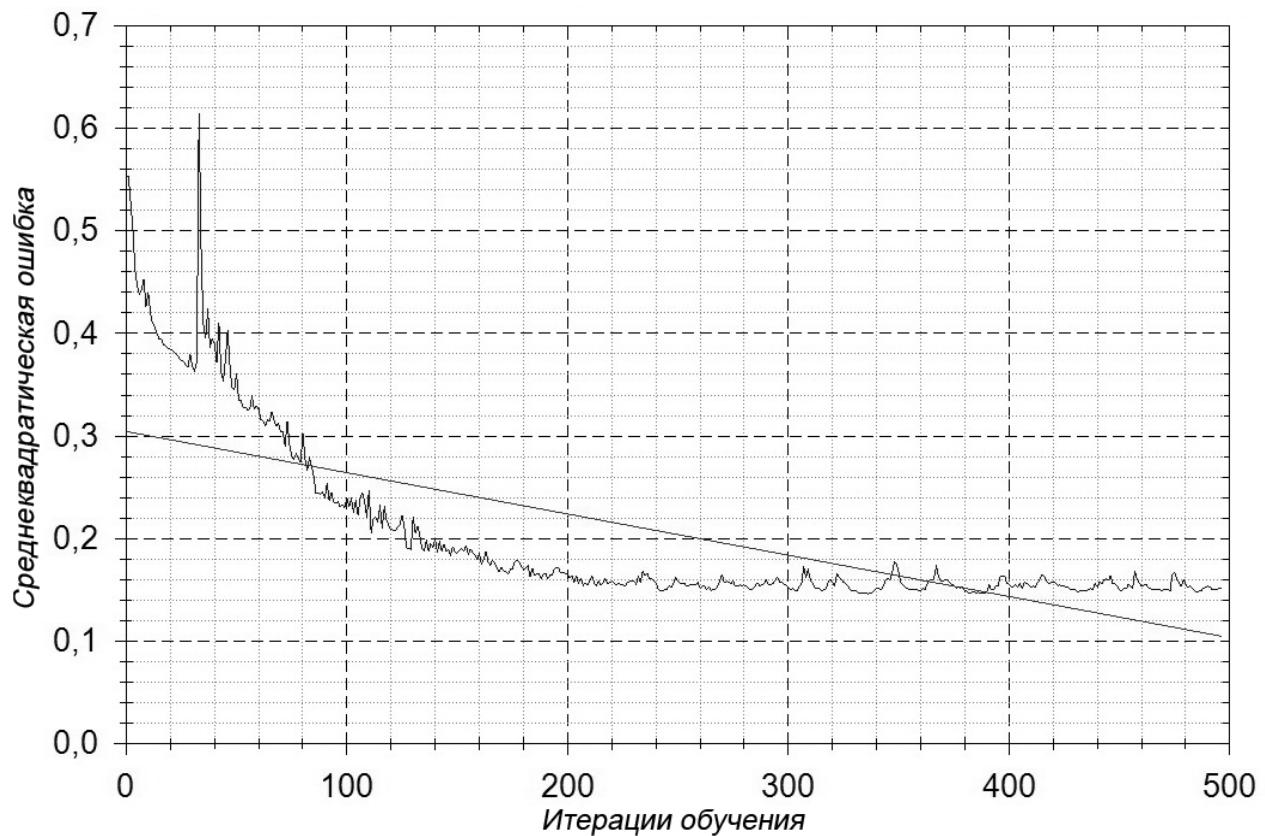


Рисунок 4.2 – Усреднённая кривая обучения для многослойного персептрона (алгоритм обратного распространения ошибки с эвристикой)

На рисунке 4.4 приведён график изменения среднеквадратической ошибки для радиально-базисной нейронной сети, обучаемой с помощью градиентного алгоритма, но с применением эвристики на основе метода упреждающего поиска.



**Рисунок 4.3 – Усреднённая кривая обучения для нейронной сети РБФ
(градиентный алгоритм обучения)**



**Рисунок 4.4 – Усреднённая кривая обучения для нейронной сети РБФ
(градиентный алгоритм обучения с применением эвристики)**

Применение при обучении нейронной сети РБФ эвристики на основе метода упреждающего поиска позволяет выполнить оценку возможных будущих состояний сети. В процессе обучения данная стратегия делает возможным исключить развитие алгоритма обучения в тех направлениях, в которых сеть не получает желаемых аппроксимирующих способностей. Анализ полученных результатов показывает улучшение динамики обучения радиально-базисной нейронной сети при использовании градиентного алгоритма обучения совместно с эвристической процедурой.

Рассмотрим изменение среднеквадратической ошибки обучения нейронной сети Эльмана, настройка которой проводилась с использованием градиентного алгоритма наискорейшего спуска (рисунок 4.5).

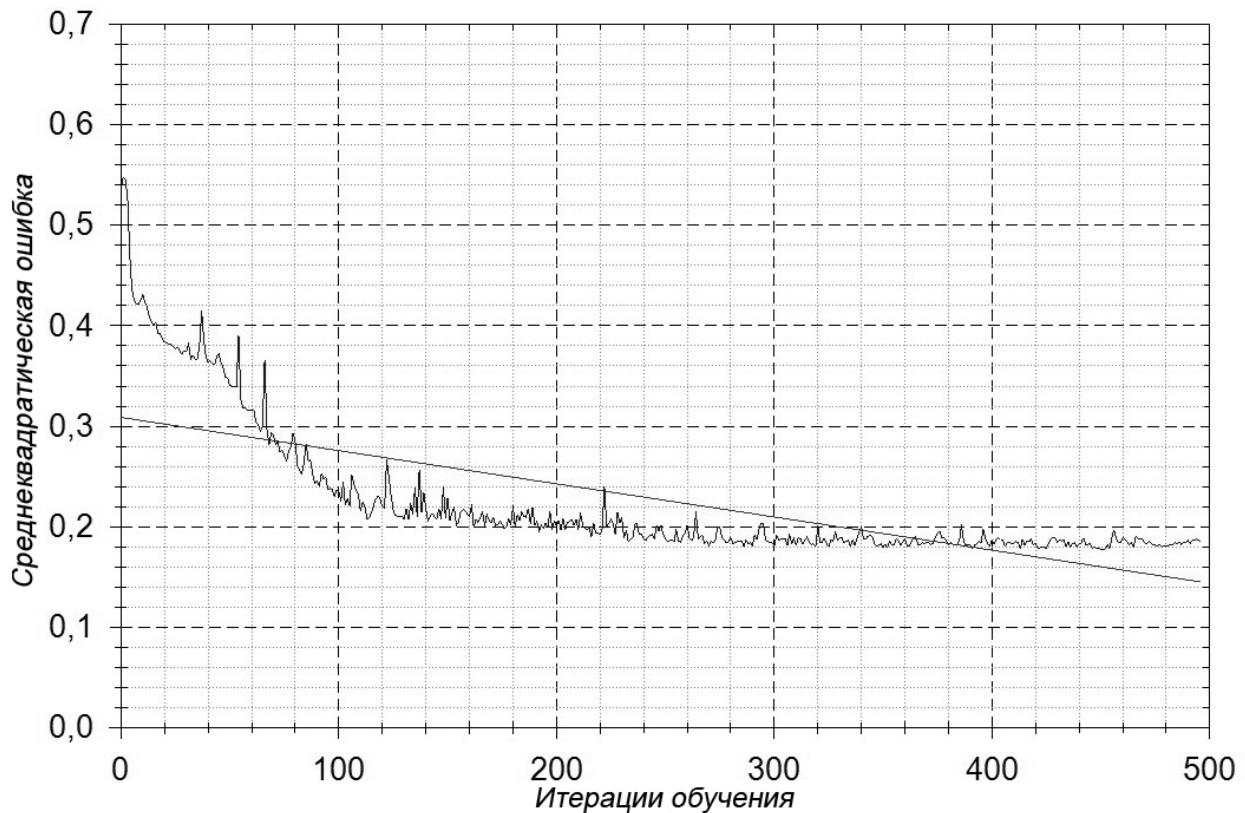


Рисунок 4.5 – Усреднённая кривая обучения для нейронной сети Эльмана (градиентный метод наискорейшего спуска)

На рисунке 4.6 приведён график изменения среднеквадратической ошибки для нейронной сети Эльмана, обучаемой с помощью градиентного алгоритма, но с применением эвристики на основе метода комплексов. Данная эвристика позволяет одновременно развивать поиск в нескольких направлениях, что

увеличивает вероятность нахождения оптимального сочетания значений весовых коэффициентов для настройки НС на решение прикладной задачи.

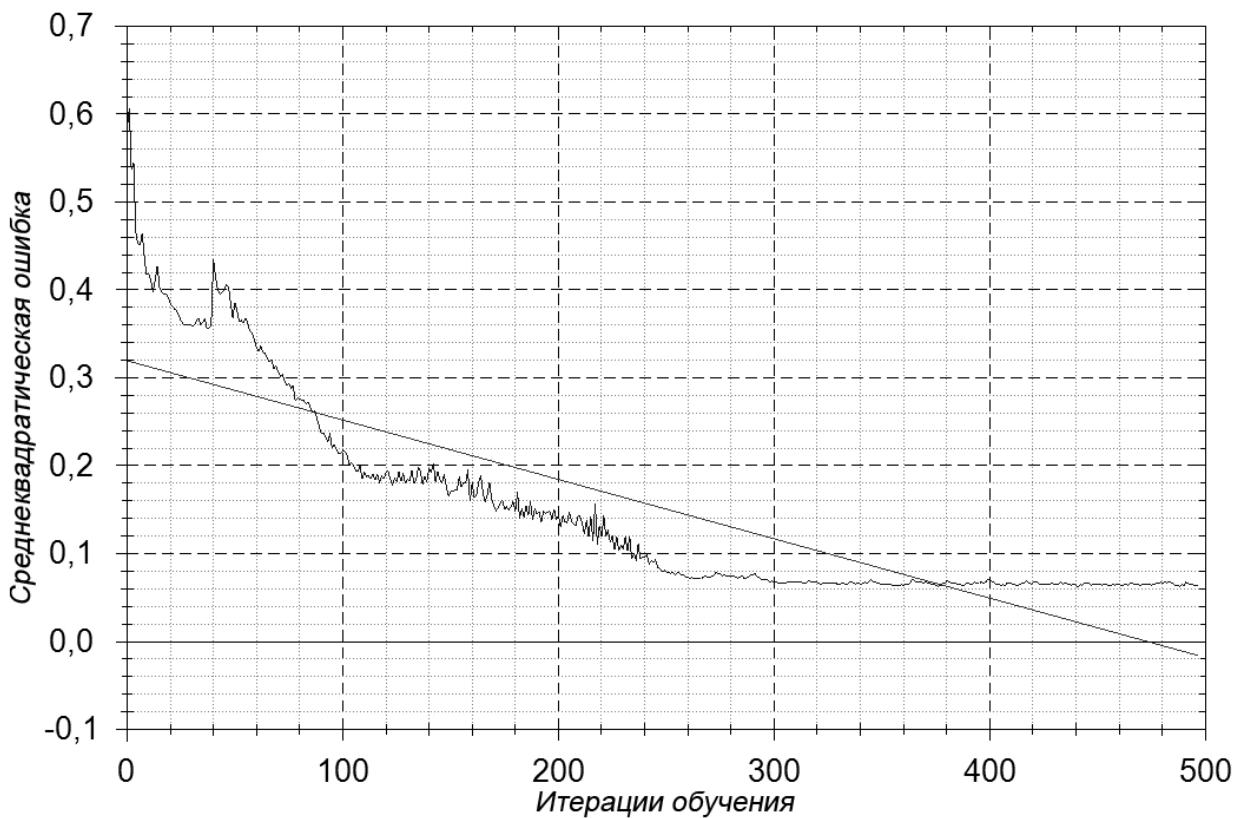


Рисунок 4.6 – Усреднённая кривая обучения для нейронной сети Эльмана (градиентный метод наискорейшего спуска с эвристикой на основе метода комплексов)

Для повышения эффективности обучения комплекса нейросетевых экспертов в работе был разработан способ инициализации начальных состояний нейронных сетей (ИНСНС), входящих в ассоциативную машину, на основе алгоритма кооперативной коэволюции. Разработанный алгоритм является альтернативой для случайной процедуры установки начальных параметров. Применяемый метод требует использования большего времени функционирования, чем процедура генерации множества случайных первоначальных параметров НС. Несмотря на длительную процедуру оптимизации, разработанный способ ИНСНС позволяет начать процесс обучения каждого компонента ассоциативной машины с состояния, которое позволяет выполнить оптимальную настройку всего ансамбля нейросетевых экспертов. Оценка эффективности разработанного алгоритма проводилась на основе проверки качества обучения всего комитета нейронных сетей.

Качественная процедура инициализации начального состояния наиболее важна в задачах с большой размерностью входных данных. В ассоциативной машине количество входов каждого эксперта соответствует количеству переменных в обучающей выборке. Для тестирования разработанного алгоритма были рассмотрены нейросетевые эксперты с фиксированным количеством входных рецепторов $n = 15$. В ходе проведения экспериментов оценивалась среднеквадратическая ошибка обучения каждого нейросетевого эксперта на момент начала процедуры обучения, начальное состояние которого определялось алгоритмом на основе случайной инициализации параметров и разработанным кооперативным иммунным алгоритмом.

На рисунке 4.7 показан график изменения среднеквадратической ошибки на момент начала обучения каждого из нейросетевых экспертов при инициализации начальных параметров с помощью разработанного алгоритма, а на рисунке 4.8 – график изменения ошибки при инициализации весовых коэффициентов случайными значениями. Количество компонент входного сигнала для каждого эксперта равно 15, по оси абсцисс ведётся отсчёт количества примеров для обучения, а по оси ординат – значение среднеквадратической ошибки перед началом процедуры обучения нейросетевых экспертов.

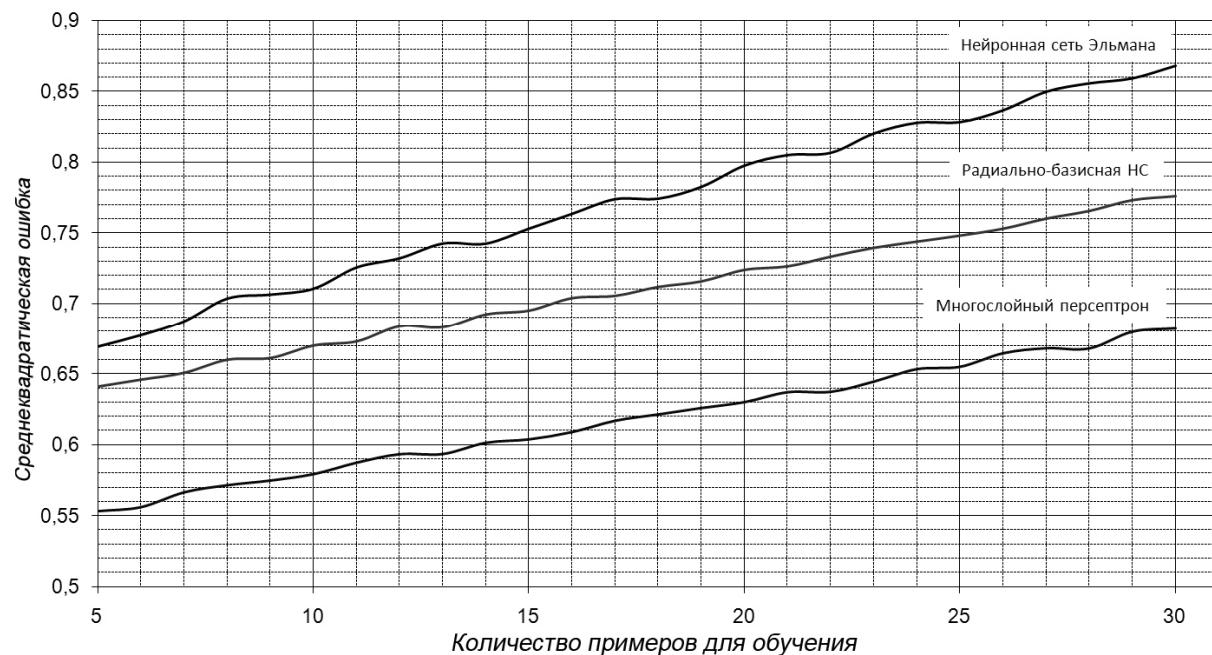
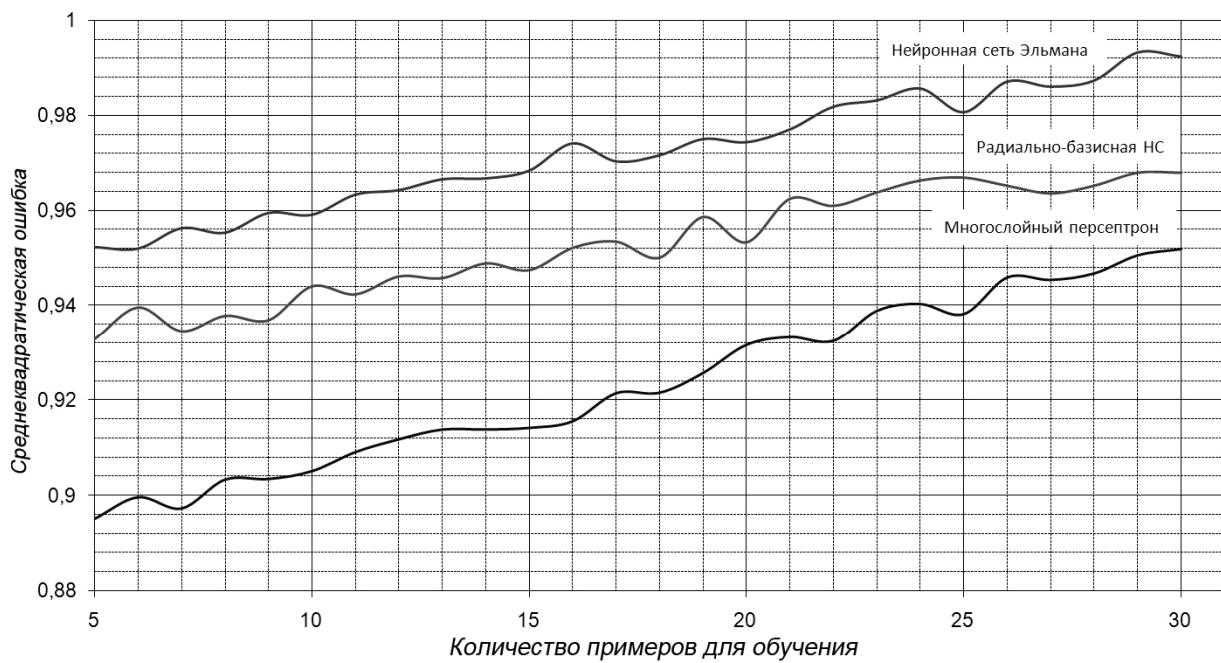


Рисунок 4.7 – Начальное значение ошибки обучения (инициализация параметров на основе коэволюционного иммунного алгоритма оптимизации)



**Рисунок 4.8 – Начальное значение ошибки обучения
(инициализация параметров случайными числами)**

Результаты экспериментов показывают, что применение методов первоначальной настройки нейронной сети на основе иммунного подхода с применением принципов коэволюции позволяет начать процедуру обучения НС из состояния, близкого к оптимальной конфигурации весовых коэффициентов. В результате применения разработанного алгоритма становится возможным уменьшить время обучения нейронных сетей, что в результате приводит к эффективному функционированию всей ассоциативной машины.

4.2 Анализ алгоритмов предобработки сигналов

Для упрощения методов исследования системы объединения мнений экспертов рассмотрим работу системы на примере трёх выходных нейронов. По оси абсцисс показаны отсчёты времени, по оси ординат – амплитуда сигнала.

Для начала рассмотрим процесс получения сигнала от нечёткой подсистемы показанного на рисунке 4.11. В разделе 2.4 приводится механизм получения коэффициента компетентности нейросетевого эксперта на основе результатов работы двух нечётких подсистем с помощью нечёткого нейрона «И». Сигнал от первой нечёткой подсистемы на интервале времени равном 30 тактам работы системы показан на рисунке 4.9.

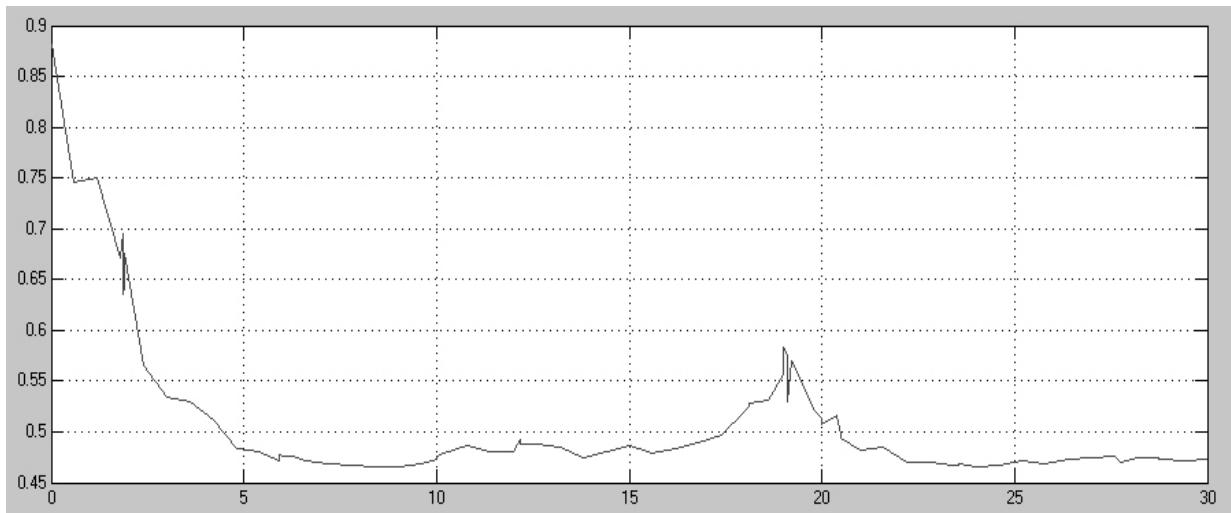


Рисунок 4.9 – Сигнал с первой нечёткой подсистемы

Вторая нечёткая подсистема анализирует оптимальность сформированной архитектуры нейросетевых экспертов, её выходной сигнал показан на рисунке 4.10.

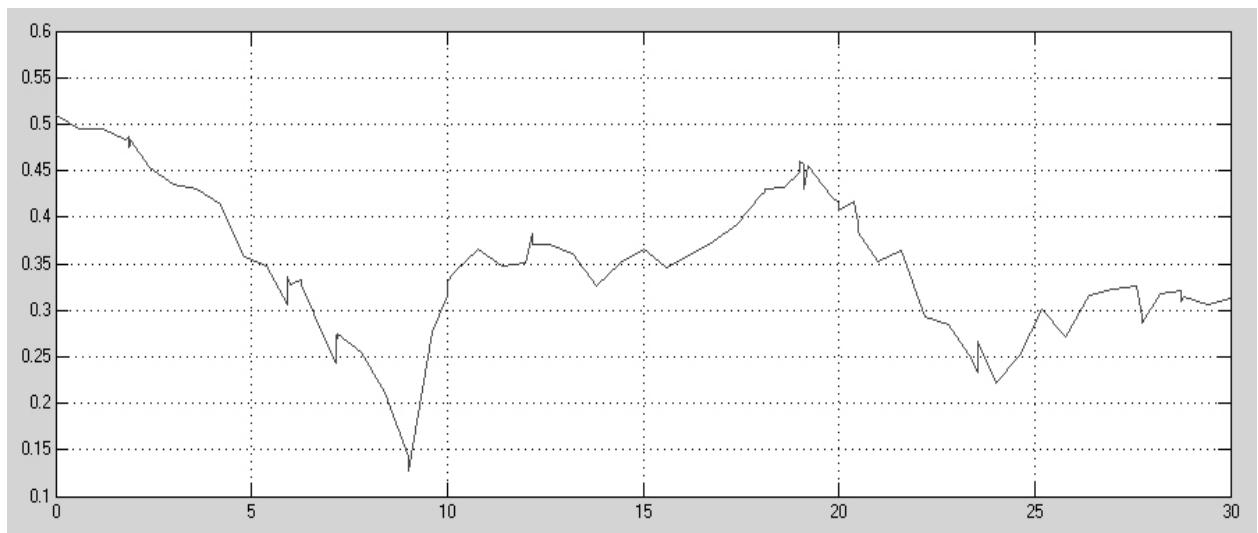


Рисунок 4.10 – Сигнал со второй нечёткой подсистемы

Для функционирования системы предобработки сигналов от экспертов сигналы объединяются с помощью нечёткого нейрона «И», значения весов которого определяются с помощью алгоритма оптимизации нечёткой подсистемы. На рисунке 4.11 показан результирующий сигнал компетентности нейронной сети для значений весов $w_1 = -0,2785$ и $w_2 = -0,1785$.

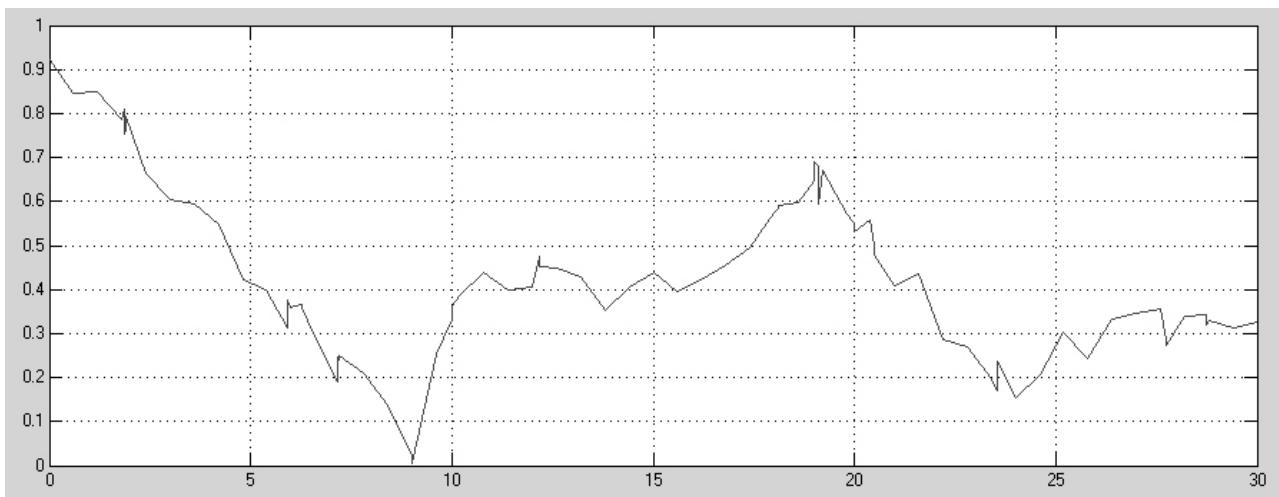


Рисунок 4.11 – Результирующий сигнал компетентности нейросетевого эксперта

Процесс функционирования нечёткой подсистемы определяет те нейросетевые структуры ассоциативной машины, способные выдать адекватную реакцию на входные воздействия. Для настройки нечётких моделей в разделе 2.4 был разработан алгоритм на основе случайных переходов в графовой модели (СПГМ). Оценка эффективности разработанного алгоритма оптимизации проводилась на основе сравнения результатов работы в одинаковых экспериментальных условиях с генетическим алгоритмом оптимизации, структура хромосомы которого была аналогична структуре закодированного решения, приведённого на рисунке 4.12. Такой способ представления хромосомы позволяет использовать стандартные генетические операторы и операторы мутации для поиска нужного решения.

Рассмотрим альтернативный алгоритм настройки нечёткой модели на основе генетического алгоритма.

1. Задать обучающую и тестовую выборку. Обучающая выборка может быть получена на основе проведения измерений моделируемой системы.
2. Определить количество функций принадлежности, их форму и расположение.
3. Задать функцию фитнесса (функцию приспособленности) в виде:

$$Fit = \frac{1}{2} \sum_{i=1}^P (d_i - y_i)^2, \quad (4.1)$$

где P – число примеров в обучающей выборке, d_i – желаемая величина выхода, y_i – реальное значение выхода.

4. Используя метод бинарного кодирования сгенерировать начальную популяцию хромосом в соответствии с архитектурой нечёткой модели (НМ), введя ограничения на возможные значения входных переменных.

5. Случайно выбрать пары хромосом для последовательного применения к ним генетических операторов.

6. Сформировать генотип потомков. Для этого с заданной вероятностью произвести применение генетических операторов к популяции.

7. Вычисление функции фитнесса всех хромосом:

a. Все параметры функций принадлежности, закодированные в соответствующих генах хромосом, подставляются в НМ.

b. Вычисляется ошибка обучения для каждой хромосомы.

8. Применить к полученной популяции оператор селекции.

9. Применить операторы мутации.

10. Определить количество хромосом, необходимых для формирования новой популяции.

11. Проверить выполнено ли нужное число генераций, если да, то перейти к пункту 12, в противном случае к п.5.

12. Конец.

Сравнение разработанного алгоритма оптимизации на основе модели переходов в пространстве состояний, базирующейся на графовой модели, и генетического алгоритма проводилось на основе возможностей этих алгоритмов провести улучшение решения на величину равную 50%. Тестирование алгоритмов проводилось на одной и той же нечёткой модели, но при этом производилось увеличение количества входных переменных, путем задания дополнительных компонентов для нечётких моделей. На рисунке 4.12 приводится зависимость, которая показывает, как изменяются относительные затраты времени, необходимые для улучшения первоначального решения на 50%, т.е. уменьшения ошибки настраиваемой модели в два раза. По оси абсцисс ведётся отсчёт

количества переменных, участвующих в оптимизации, а по оси ординат – время в относительных единицах. Относительный отсчёт времени вводится для устранения зависимости алгоритма от конкретной вычислительной платформы. В качестве значения времени, принимаемого за единицу, выбирается максимальное время работы одного из оптимизационных алгоритмов с нечёткой моделью, содержащей 50 переменных. При увеличении числа переменных увеличивается длина хромосом ГА, что приводит к усложнению операторов мутации, скрещивания и селекции. В то время как СПГМ для получения новых решений необходимо выбрать точку поискового пространства в окрестности выбранной точки одной из вершины графа. В результате решения данной задачи алгоритм СПГМ показал большую эффективность, чем ГА.

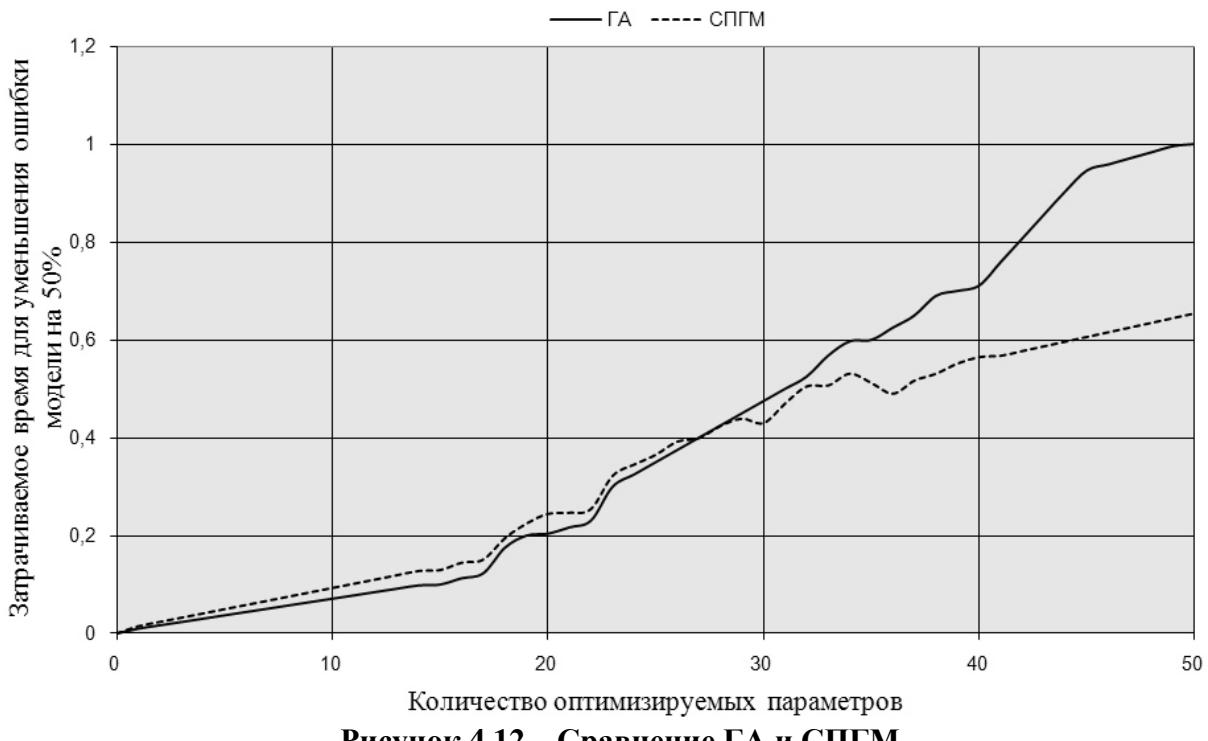


Рисунок 4.12 – Сравнение ГА и СПГМ

Рассмотрим совместную работу системы оценки качества экспертов и их влияние на обработку сигнала от нейронных сетей. На рисунке 4.13 в моменты времени от 0 до 10 наблюдается активность первого нейрона нейросетевого эксперта, в моменты времени от 10 до 20 – активность второго нейрона, от 20 до 30 – активность третьего нейрона. На графике показан сигнал от нечёткой системы, используя его значения амплитуды, система на основе модифицированного нейрона Фукусимы производит подавление входного

сигнала. В моменты времени от 0 до 10 наблюдается улучшение характеристик первого эксперта, это приводит к увеличению амплитуды сигнала, который подаётся на финального эксперта. Несмотря на то, что сигнал, характеризующий эксперта, в момент времени 9 уменьшился практически до нуля, выходной сигнал первого эксперта увеличился лишь до амплитуды 0,7, а не принял максимальное значение. Здесь можно наблюдать влияние элементов задержки, которые расположены в слое тормозящих рецепторов нейрона Фукусимы. Это не позволяет сигналу с эксперта скачкообразно возрастать, что может привести к нестабильной работе алгоритмов обучения финального эксперта. В момент времени 20 происходит возрастание амплитуды сигнала качества обучения эксперта, что приводит к снижению амплитуды сигнала первого эксперта. Выходной сигнал первого эксперта после предобработки представлен на рисунке 4.14.

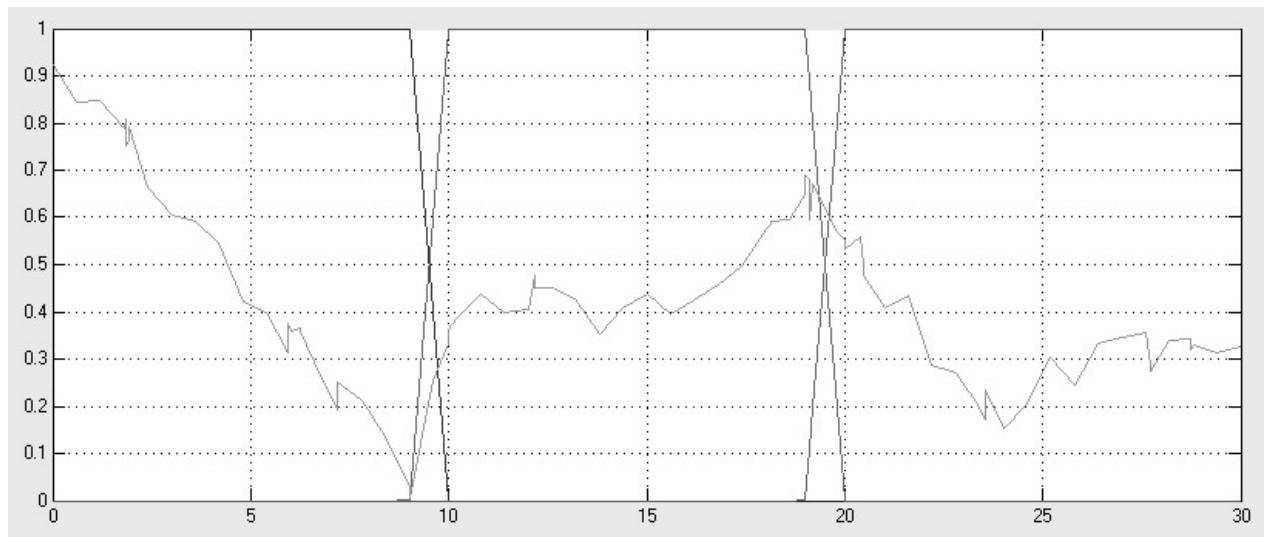
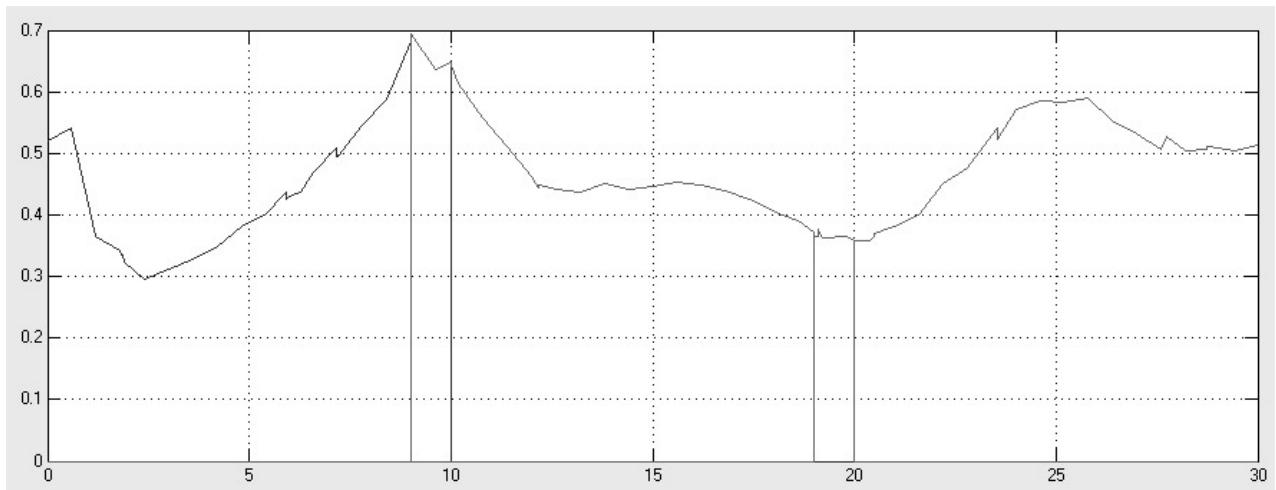


Рисунок 4.13 – График сигналов с первого эксперта



**Рисунок 4.14 – График сигналов после обработки структурой на основе нейрона
Фукусимы**

Аналогичным образом производится обработка сигналов от других двух экспертов. Сигналы для второго и третьего эксперта представлены на рисунках 4.15 – 4.16 и 4.17 – 4.18. Во время функционирования системы предобработки сигналов оценивается «история» качества обучения эксперта. Если динамика обучения показывает уменьшение ошибки, то сигнал соответствующего эксперта увеличивается пропорционально интервалам времени, в течение которого он показывает приемлемый результат.

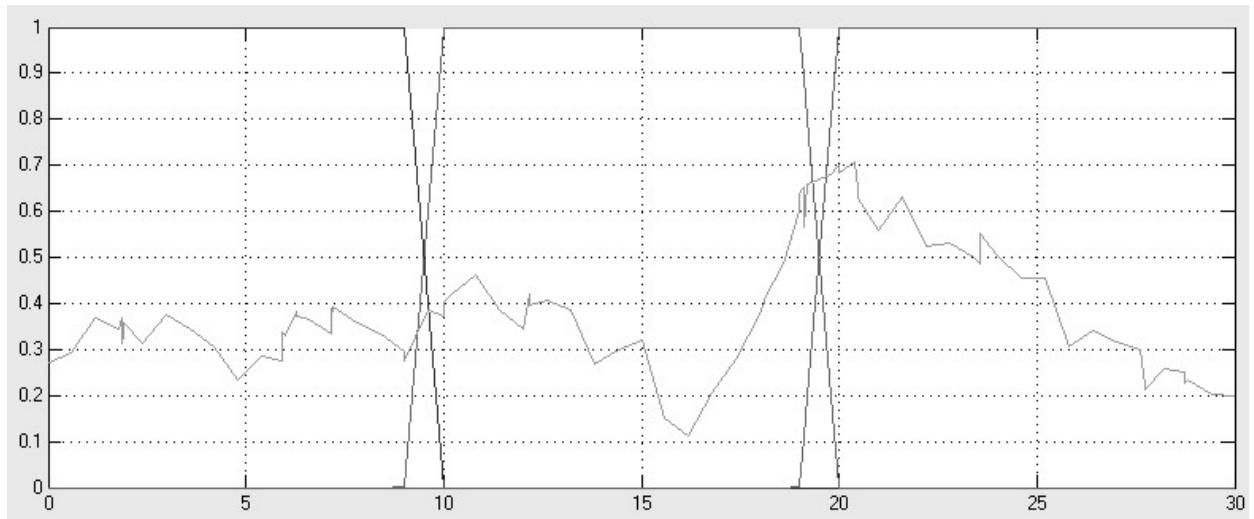


Рисунок 4.15 – Выходной сигнал со второго эксперта

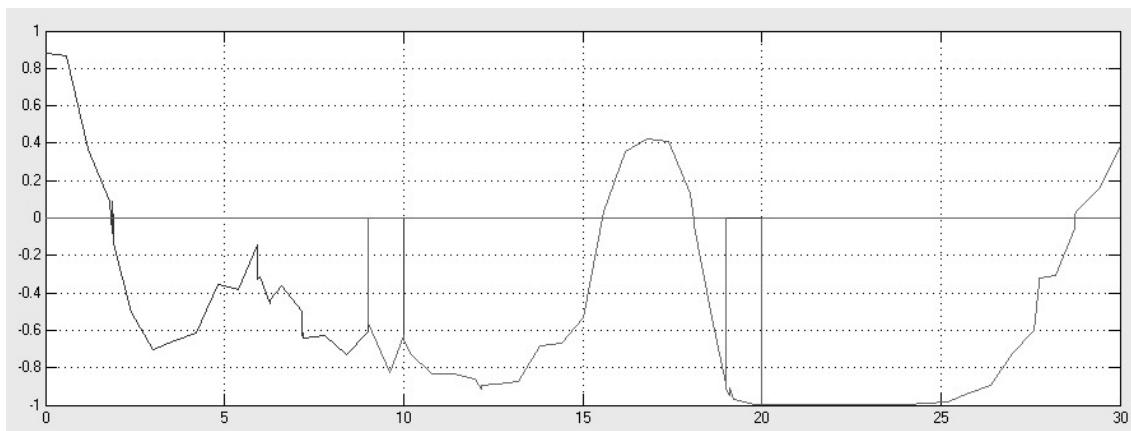


Рисунок 4.16 – График выходного сигнала после предобработки модификацией квадратичного нейрона

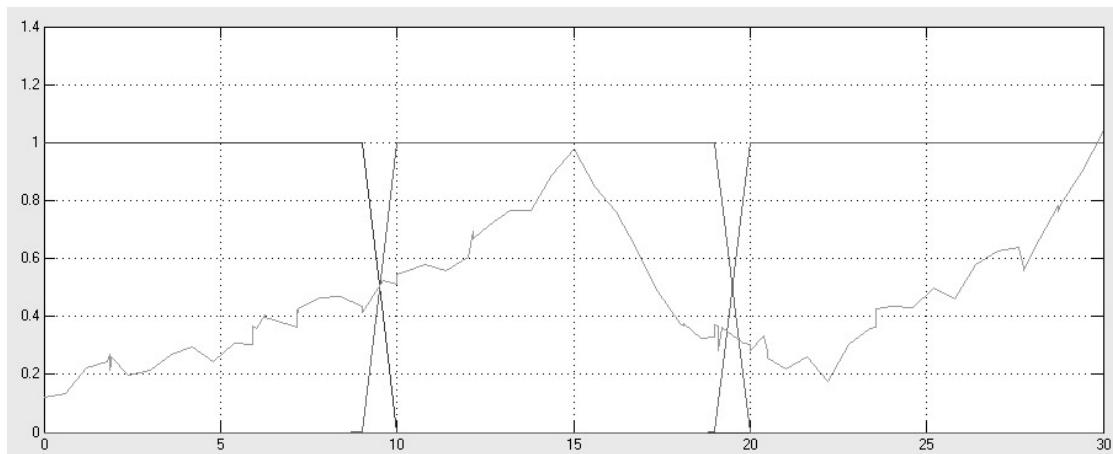


Рисунок 4.17 – Графики сигналов с третьего эксперта

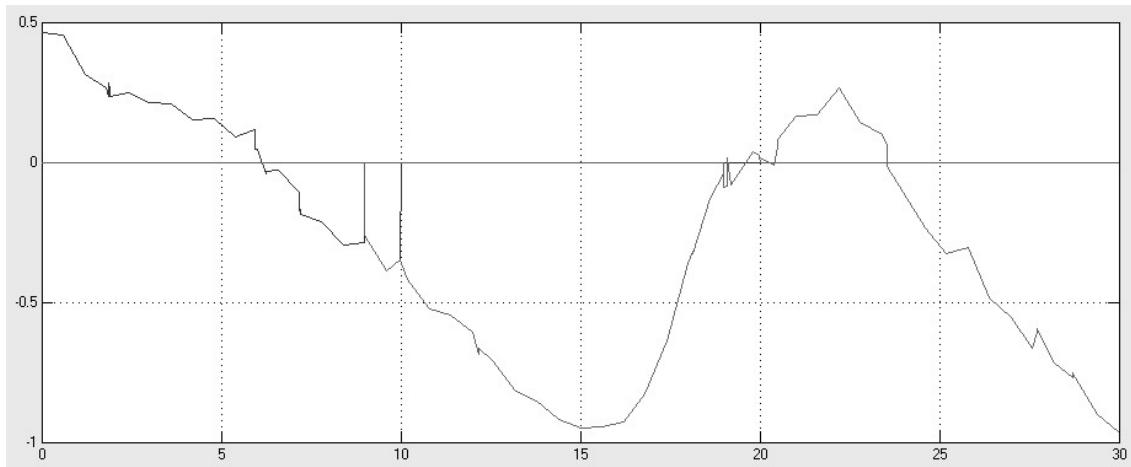


Рисунок 4.18 – График выходного сигнала после предобработки модификацией N -адалины

Сигналы от трёх экспертов после предобработки нейроном Фукушимы, квадратичным нейроном и N – адалиной подаются на систему латеральных связей, описанную в разделе 2.5. Сигнал от нейросетевого эксперта,

представленного многослойным персептроном показан на рисунке 4.19. Сигналы от радиально-базисной сети и нейронной сети Эльмана приведены на рисунках 4.20 и 4.21 соответственно.

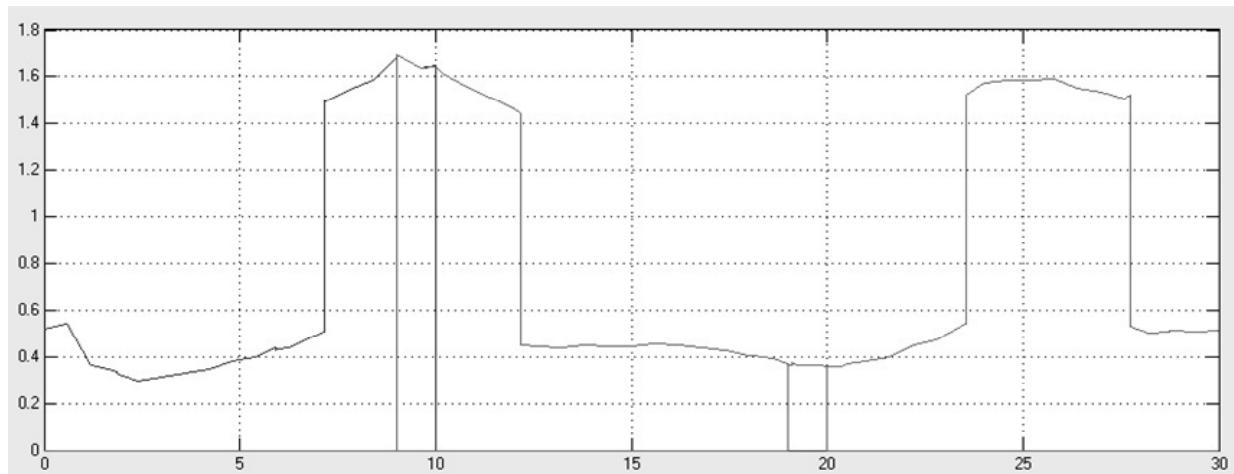


Рисунок 4.19 – Сигнал с многослойного персептрана после коррекции

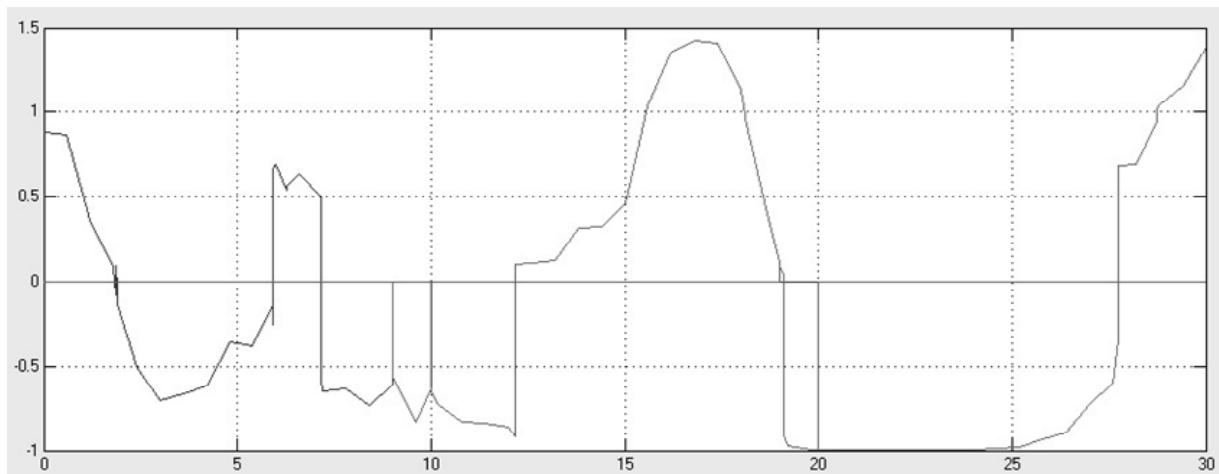


Рисунок 4.20 – Сигнал с радиально-базисной нейронной сети после коррекции

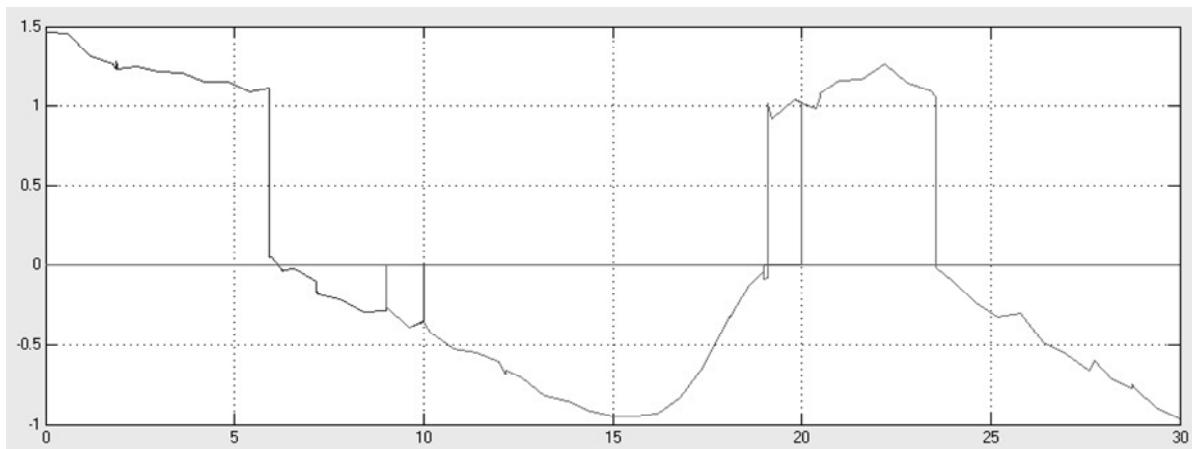


Рисунок 4.21 – Сигнал с рекуррентной сети Эльмана после коррекции

Как видно из графиков, на сигнал с нейросетевого эксперта влияет не только нейронечеткая нейронная сеть, но и история сигнала о качестве обучения экспертов. После обработки системой, содержащей латеральные связи, амплитуда сигнала может выйти за рамки допустимых значений (от 0 до 1, от -1 до 1). При этом необходимо выполнять ограничение амплитуды сигнала по модулю до заданных допустимых интервалов. Сигналы от трёх экспертов после ограничения по амплитуде показаны на рисунке 4.22 для многослойного персептрона, на рисунке 4.23 для RBF сети и на рисунке 4.24 для рекуррентной сети Эльмана.

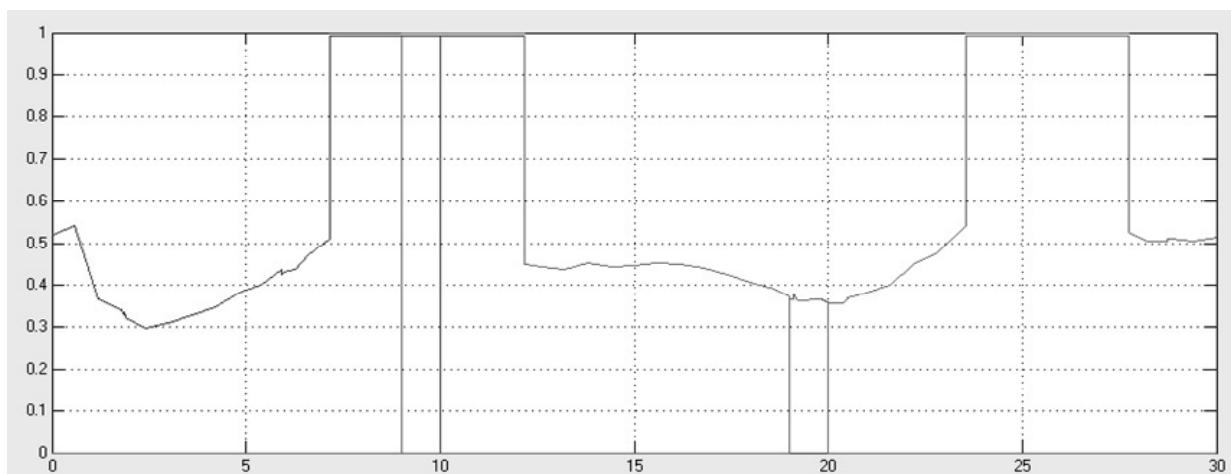


Рисунок 4.22 – Сигнал с многослойного персептрона после ограничения

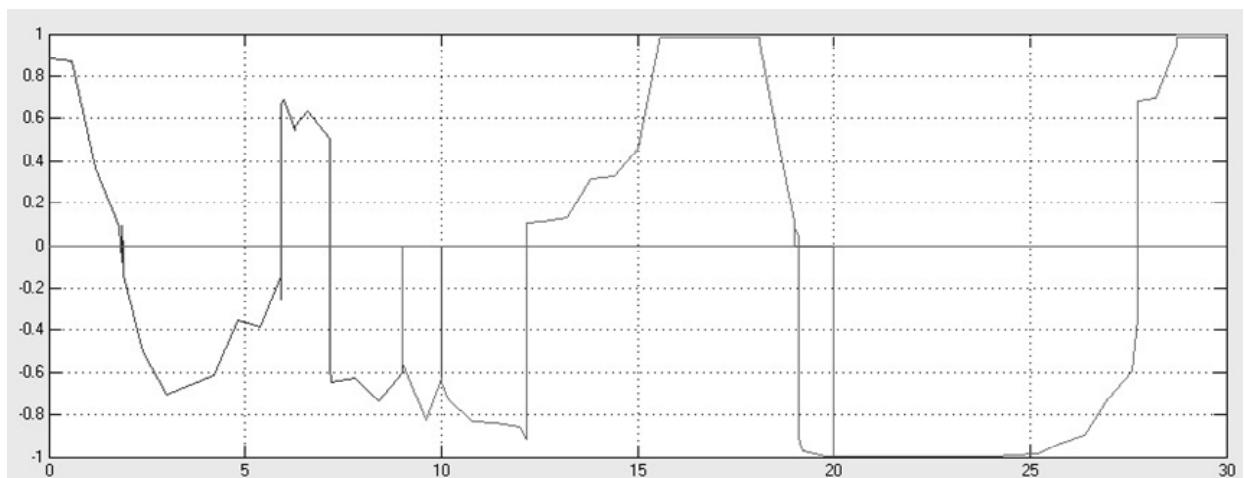


Рисунок 4.23 – Сигнал с радиально-базисной нейронной сети после ограничения

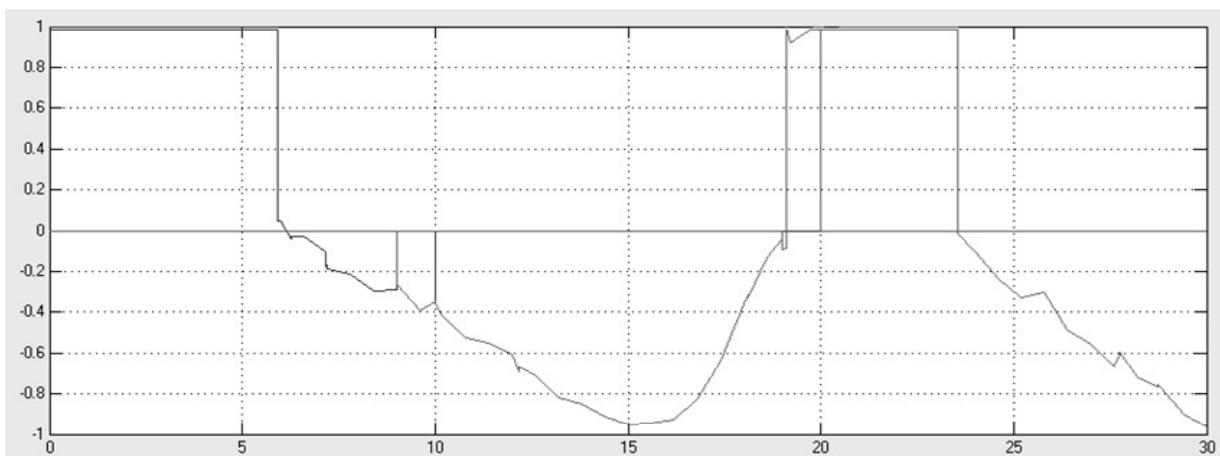
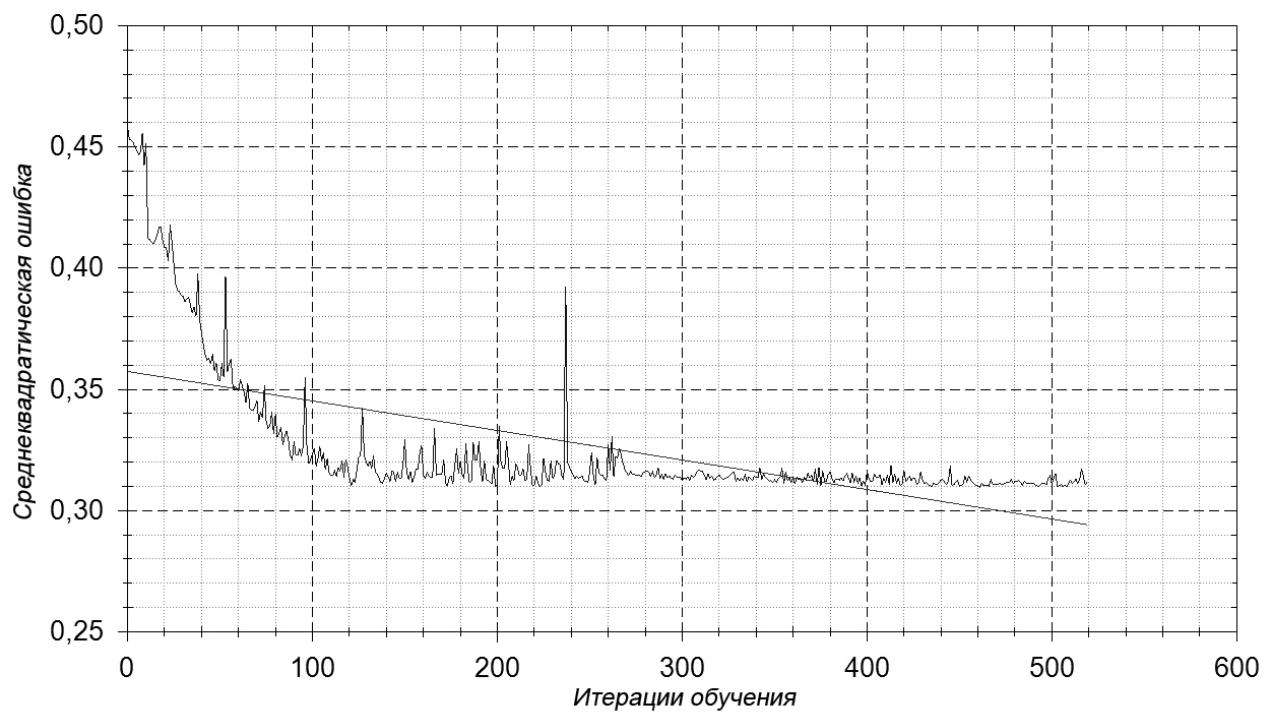


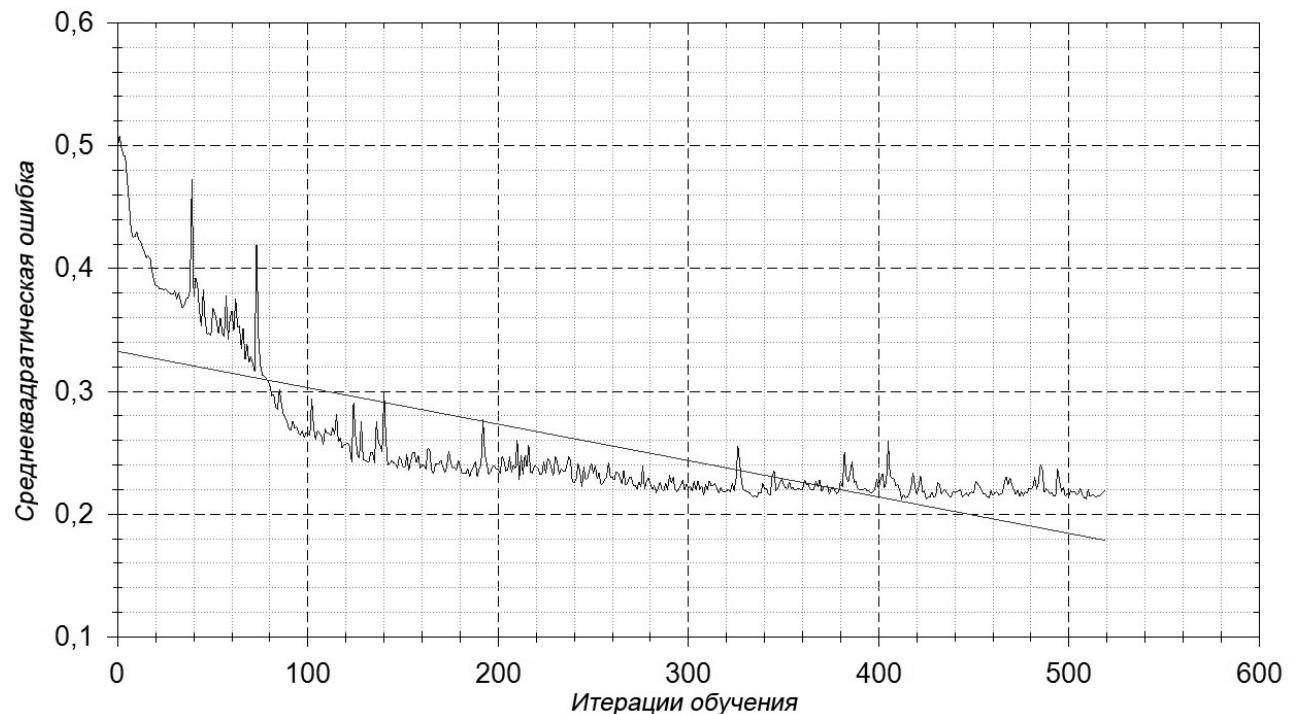
Рисунок 4.24 – Сигнал с рекуррентной сети Эльмана после ограничения

После выполненной предобработки сигналы от экспертов подаются на финальную сигма-пи нейронную сеть. Все сигналы с выходных нейронов всех экспертов подаются на входы сигма-пи сети. В результате функционирования финальной сети определяется «победивший» нейрон с максимальным значением выходного сигнала, характеризующий одно из решений, принятых сетью, а значение выходного сигнала данного нейрона показывает степень достоверности принятого решения.

В разделе 2.5 рассмотрен алгоритм настройки сети на основе комбинации алгоритмов случайного поиска с применением эвристики. Результаты экспериментов показывают, что алгоритм перехода в другую область поискового пространства, основанный на оценке изменения состояния геометрической модели, позволяет повысить эффективность обучения финальной НС для принятия окончательного решения.



**Рисунок 4.25 – Усреднённая кривая обучения для сигма-пи нейронной сети
(комбинация алгоритмов случайного поиска)**



**Рисунок 4.26 – Усреднённая кривая обучения для сигма-пи нейронной сети
(комбинация алгоритмов случайного поиска с эвристикой)**

На рисунке 4.25 показано изменение среднеквадратической ошибки для процесса обучения сигма-пи сети на основе комбинации алгоритмов случайного поиска. Применение эвристики на основе оценки изменения графовой модели позволяет осуществить переход в различные области возможного пространства

поиска, что повышает вероятность нахождения оптимальных параметров нейронной сети для аппроксимации желаемой зависимости. На основе приведённых результатов (рисунок 4.26) можно сделать вывод об эффективности применения подобных эвристических алгоритмов для оптимизации сложных систем.

4.3 Результат работы алгоритмов генерации случайных чисел

Основной элемент генератора – нейронная сеть Хопфилда, которая подвергается процессу переобучения. Сеть утрачивает способность к распознаванию образов, которые её предъявлялись во время обучения, но сохраняет возможность сходиться к аттракторам. В качестве примера рассмотрим организацию первой нейронной сети из множества НС. Для сети с девятью нейронами, построенной из синаптической матрицы (4.2) согласно формуле (4.3).

$$T = \begin{bmatrix} 0 & -1 & 5 & -9 & 1 & 7 & 9 & -1 & 1 \\ -1 & 0 & 3 & 1 & -5 & 1 & 3 & 1 & 3 \\ 5 & 3 & 0 & 7 & 5 & -1 & 1 & -5 & -11 \\ -9 & 1 & 7 & 0 & -1 & 5 & -5 & -3 & -5 \\ 1 & -5 & 5 & -1 & 0 & -1 & 5 & -1 & -3 \\ 7 & 1 & -1 & 5 & -1 & 0 & -1 & 5 & -1 \\ 9 & 3 & 1 & -5 & 5 & -1 & 0 & -1 & 1 \\ -1 & 1 & -5 & -3 & -1 & 5 & -1 & 0 & -5 \\ 1 & 3 & -11 & -5 & -3 & -1 & 1 & -5 & 0 \end{bmatrix} \quad (4.2)$$

$$Q_i(t+1) = f\left(\sum_{j=0}^{N-1} T_{ij} Q_j(t)\right), \quad f(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (4.3)$$

где T – матрица весовых коэффициентов, $Q_i(t)$ – состояние i -го нейрона в момент времени t .

Пусть h – набор аттракторов, а Q – множество сходящихся областей соответствующих аттракторов. В результате получаем:

$$h = \{h_i \mid i = 1, 2, \dots, p\},$$

где p – число аттракторов сети. Области сходимости представлены в таблице 4.1. Из таблицы видно, что данная сеть является устойчивой и все возможные входные состояния сходятся к определённому аттрактору сети.

Для остальных 23 сетей Хопфилда выполняются такие же правила, за исключением индивидуально настроенных матриц весовых коэффициентов. Различные матрицы формируют в процессе функционирования различные аттракторы и области сходимости. В результате образуется множество аттракторов и областей сходимости, что приводит к сложному процессу генерации последовательности чисел. Если учесть, что асинхронное извлечение информации из нейронной сети может происходить и не до достижения аттрактора, то количество выходных состояний сети может быть увеличено до всех возможных комбинаций, определяемых количеством нейронных элементов в сети Хопфилда. Несмотря на сложность организации данной системы, нейронная сеть Хопфилда имеет параллельную структуру и может быть распараллелена, что позволяет получить высокое быстродействие.

Для генерации векторов, применяемых для настройки нейронных сетей Хопфилда, применяется источник энтропии. Процесс перезаряда двух ёмкостей лежит в основе получения последовательности битов. Если первая ёмкость зарядилась до максимального напряжения быстрее второй, то система находится в первом состоянии – логический ноль, а если вторая зарядилась быстрее первой – то логическая единица. В результате получаем бинарную последовательность. Описанный процесс генерации представлен на рисунке 4.27. Сплошной линией обозначено изменение напряжения на первой ёмкости, пунктирной – изменение напряжения на второй. «Гонки» во время заряда и система, стремящаяся минимизировать разницу во времени заряда каждой ёмкости до максимального напряжения, приводят к процессу, показанному на рисунке. На интервале от 0 до 11 секунд быстрее зарядилась вторая ёмкость, считаем, что источник энтропии произвёл логическую единицу, на интервале от 12 до 23 снова быстрее зарядилась вторая ёмкость – логическая единица, от 24 до 35 первая ёмкость зарядилась быстрее второй – логический ноль и т.д.

Таблица 4.1 Область сходимости для сети Хопфилда

h_i	010000011	011100000	001110000
Q	000 001 002 003 272 273 274 275 280 281 282 283 284 285 286 287 328 329 330 331 334 335 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 406 407 408 409 410 411 412 413 414 415 456 457 458 459 460 461 462 463 472 473 474 475 478 479	004 005 006 007 012 013 014 015 044 045 046 047 316 317 318 319	008 009 010 011 016 017 018 019 022 023 024 025 026 027 028 029 030 031 040 041 042 043 056 057 058 059 060 061 062 063 072 073 074 075 076 077 078 079 088 089 090 091 094 095 344 345 346 347
h_i	101111100	111111110	101010100
Q	020 021 032 033 034 035 048 049 050 051 052 053 054 055 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 176 177 178 179 180 181 182 183	036 037 038 039 096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 164 165 166 167 224 225 226 227 228 229 230 231 232 233 236 237 238 239 252 253 254 255	064 065 066 067 080 081 082 083 084 085 086 087 092 093 376 377 378 379
h_i	111010101	011101010	010101011
Q	068 069 070 071 332 333 340 341 342 343 348 349 350 351 360 361 362 363 364 365 366 367 372 373 374 375 380 381 382 383 476 477 504 505 506 507 508 509 510 511	128 129 130 131 140 141 142 143 144 145 146 147 148 149 150 151 162 163 168 169 170 171 172 173 174 175 440 441 442 443	132 133 134 135 424 425 426 427 428 429 430 431 444 445 446 447
h_i	001111010	110001111	101011110
Q	136 137 138 139 152 153 154 155 156 157 158 159 184 185 186 187 188 189 190 191 200 201 202 203 204 205 206 207 216 217 218 219 220 221 222 223 234 235	160 161 260 261 288 289 290 291 292 293 294 295 404 405 416 417 418 419 420 421 422 423 432 433 434 435 436 437 438 439 448 449 450 451 452 453 454 455 468 469 470 471 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 500 501 502 503	192 193 194 195 196 197 198 199 208 209 210 211 212 213 214 215 240 241 242 243 244 245 246 247 248 249 250 251
h_i	010100001	110000101	100010101
Q	256 257 258 259 262 263 264 265 266 267 268 269 270 271 296 297 298 299 300 301 302 303 312 313 314 315	276 277 278 279 304 305 306 307 308 309 310 311 320 321 322 323 324 325 326 327 352 353 354 355 356 357 358 359	336 337 338 339 368 369 370 371
h_i	100011111		
Q	464 465 466 467 496 497 498 499		

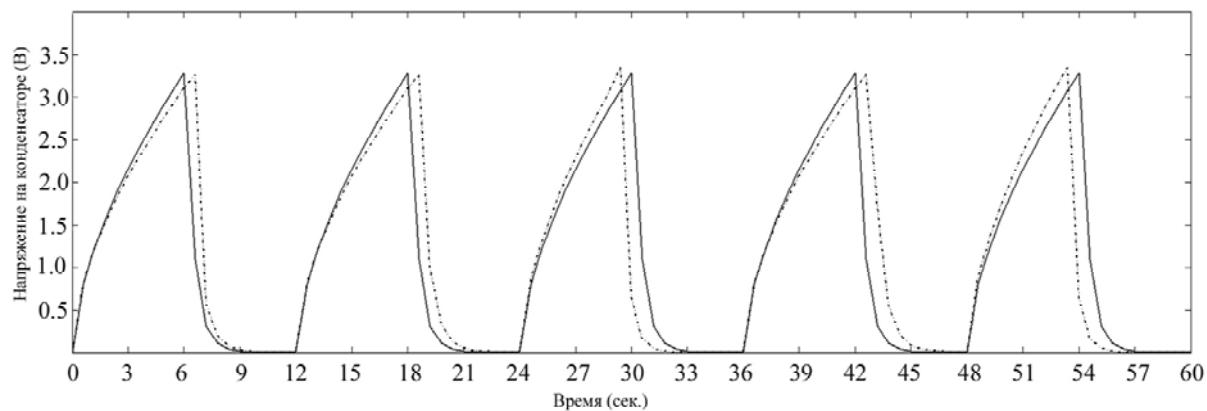


Рисунок 4.27 – Осциллограмма заряда конденсаторов

В результате функционирования источника энтропии в течение длительного времени становится возможным произвести накопление энтропии. Далее из полученной бинарной последовательности сформировать вектора необходимой размерности для использования их в алгоритмах настройки и управления.

Функционирование следующего компонента – осциллятора, построенного на основе модифицированных нейронов Хопфилда, определяется начальными состояниями весовых коэффициентов. Рассмотрим пример работы осциллятора с весовыми коэффициентами, приведёнными в таблице 4.2.

Таблица 4.2 Начальные значения весовых коэффициентов осциллятора

Порядковый номер нейрона	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
№1	0,0009	-0,14	0,78	-0,55	0,98	0,37	0,88	0,47
№2	0,26	0,0009	0,78	-0,55	-0,68	0,37	-0,88	-
№3	0,0009	-0,14	0,78	-0,55	0,98	0,37	0,88	0,47
№4	0,26	0,0009	0,78	-0,55	-0,68	0,37	-0,88	-

Выходной сигнал осциллятора показан на рисунке 4.28. Отсчёт времени проводится в тактах работы системы. Такт работы осциллятора – это время необходимое для генерации нового состояния. Выходной сигнал показывает, что детерминированная система (внутреннее состояние которой развивается по детерминированным правилам) ведёт себя сложным образом. Даже если состояния осциллятора определяются предыдущими выходными значениями, небольшие случайные изменения весов, которые могут быть произведены

сигналами от источника энтропии, делают невозможным предсказание состояний системы на длительный интервал времени.

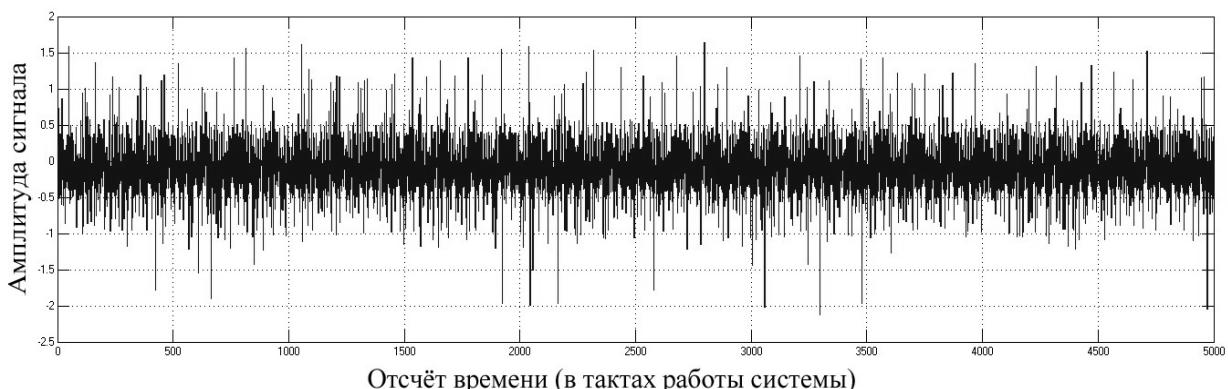


Рисунок 4.28 – Выходной сигнал осциллятора

Рассмотрим сигнал, приведённый на рисунке 4.29, на интервале времени от 2700 до 2800 тактов. Данную зависимость необходимо преобразовать в бинарную последовательность. Сложная форма сигнала не позволяет задать фиксированное значение порога и считать логической единицей случай, когда значение сигнала выше порога, а логическим нулем состояние системы, когда выходной сигнал ниже заданного порогового уровня. В данной системе значение порога формируется на основе среднего значения десяти предыдущих отсчетов сигнала. Значения сигнала порога системы показаны на рисунке 4.30 и 4.31.

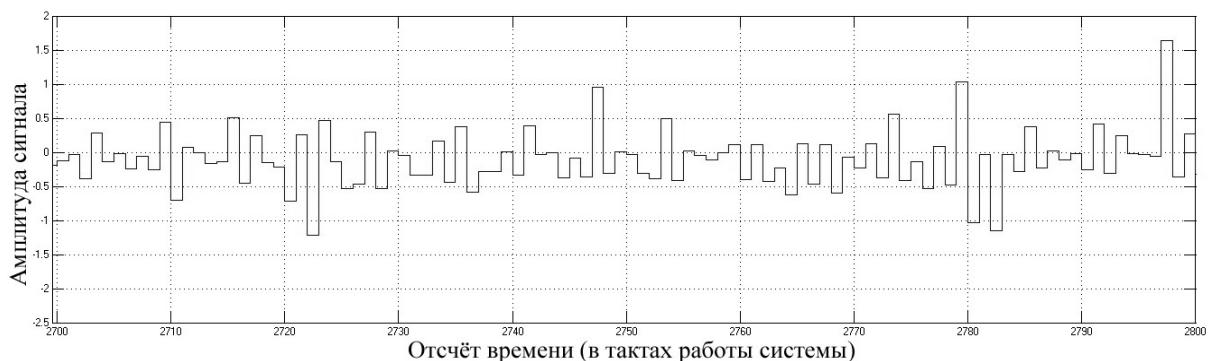


Рисунок 4.29 – Выходной сигнал осциллятора на ограниченном интервале

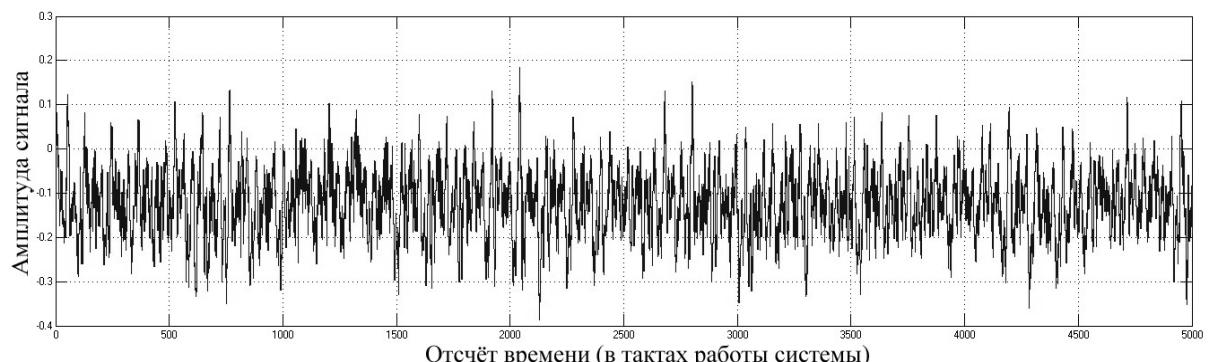


Рисунок 4.30 – Изменение порога системы во времени

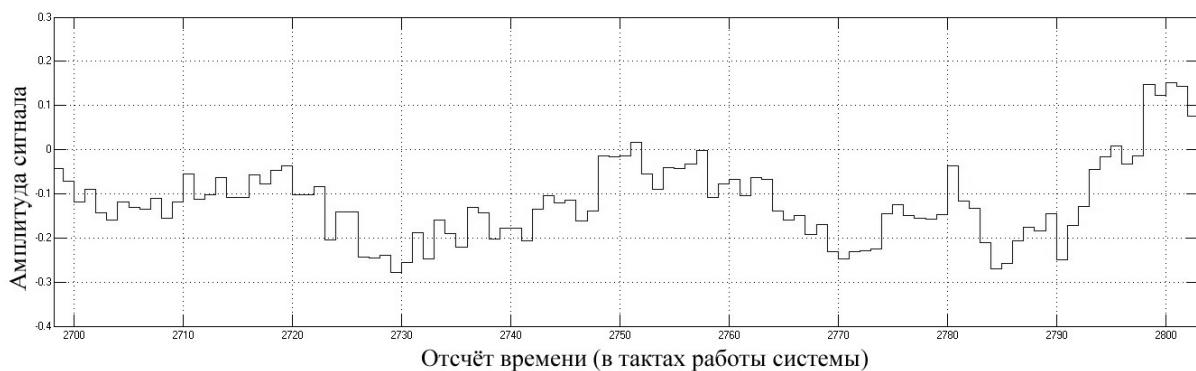


Рисунок 4.31 – Значение порога системы на ограниченном интервале времени

В результате преобразования получаем бинарную последовательность, из которой в дальнейшем формируются вектора для инициализации отдельных подсистем и генерации векторов управления (рисунок 4.32).

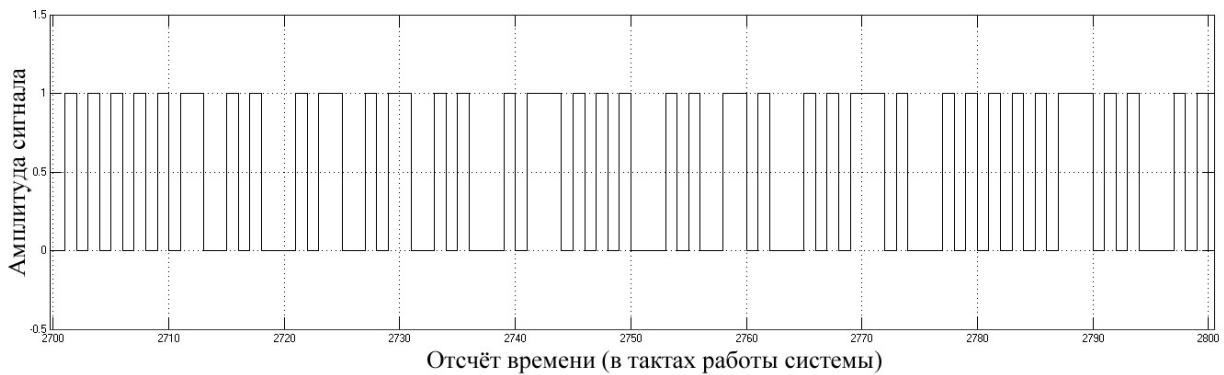


Рисунок 4.32 – Выходной бинарный сигнал осциллятора на ограниченном интервале

Функционирование логической нейронной сети показывают временные диаграммы всех возможных состояний. В зависимости от управляющих векторов выполняется та или иная логическая функция (рисунок 4.33).

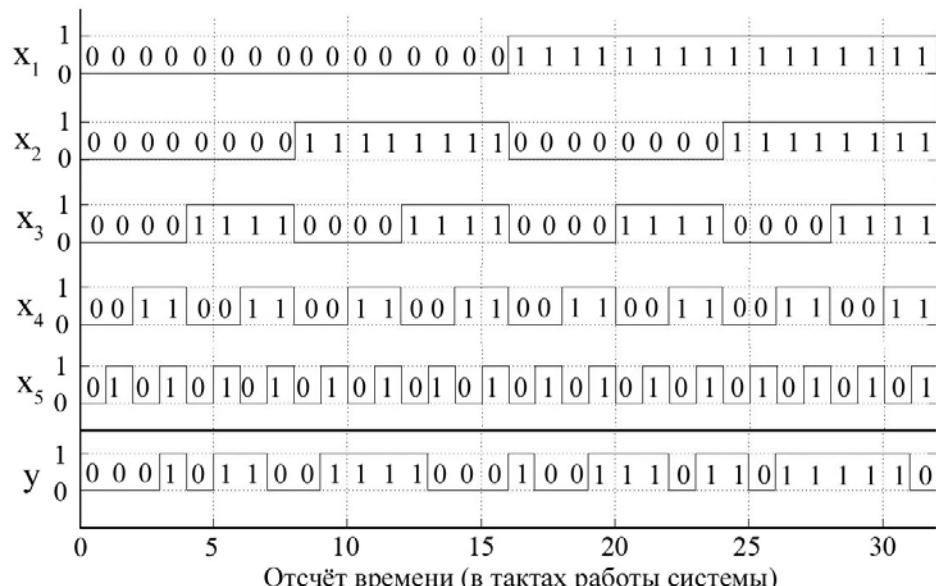


Рисунок 4.33 – Возможные состояния логической нейронной сети

В результате работы системы генерируется последовательность случайных чисел. Для применения такого ГСЧ в системах необходимо провести проверку корреляции чисел друг с другом. Получаемые числа должны быть как можно более независимы, чтобы на раскрытие механизма их генерации уходило большое количество времени и вычислительных ресурсов. Для этого необходимо провести несколько статистических тестов.

Тест для проверки зависимостей между элементами исследуемой последовательности случайных чисел – «тест распределения на плоскости» [50]. На поле размером $(2^R - 1) \times (2^R - 1)$ (R – разрядность чисел, $R = 9$) строится множество точек с координатами (x_i, x_{i+1}) , где x_i – элементы исследуемой последовательности, $i = 1, \dots, (N-1)$, N – длина последовательности ($N = 3000$). Пример полученной зависимости приведён на рисунке 4.34. Как видно из рисунка, точки расположены хаотично и между ними отсутствует зависимость, что может свидетельствовать об успешном прохождении теста.

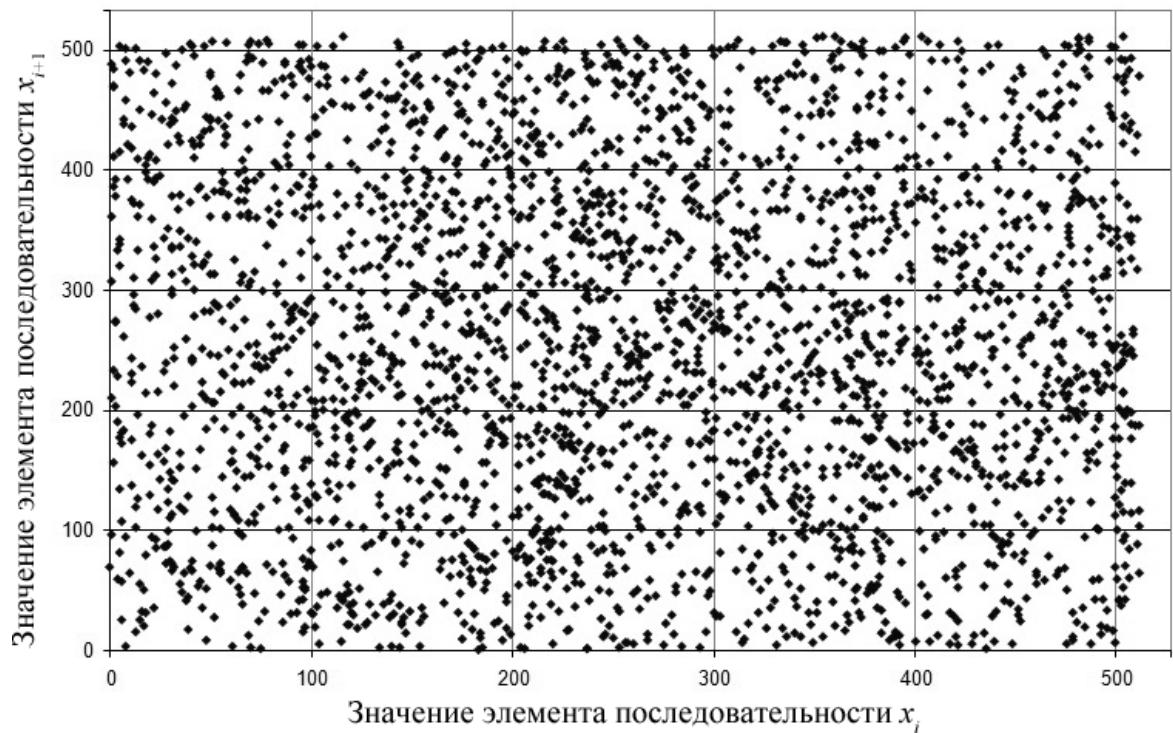


Рисунок 4.34 – Результаты теста на наличие зависимости в последовательности

Следующий тест – «гистограмма распределения элементов» [50]. Данный тест направлен на исследование равномерности распределения полученных чисел, путём определения частоты появления каждого числа. Гистограмма строится

следующим образом: в полученной последовательности подсчитывается количество элементов, имеющих одно значение. После этого строится график зависимости числа элементов от их численного представления. При этом в гистограмме должны присутствовать все возможные элементы рассматриваемой разрядности. Полученная гистограмма приведена на рисунке 4.35.

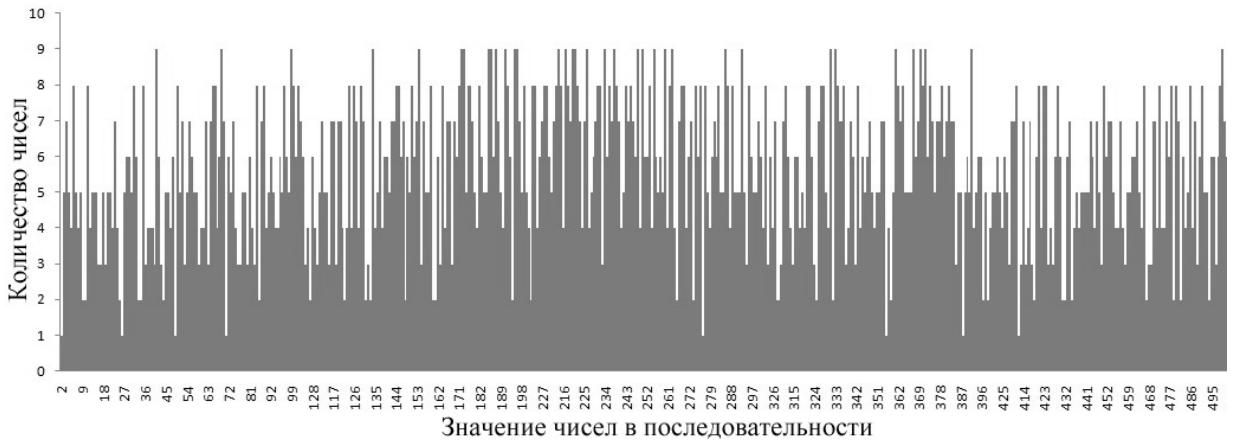


Рисунок 4.35 – Гистограмма частот появления чисел в последовательности

Результаты графических тестов интерпретирует пользователь, статистические тесты позволяют выявить величины, по которым можно однозначно судить пройден тест или нет. Статистические тесты применяются для исследования того, являются ли псевдослучайные последовательности чисел достаточно случайными. Одним из основных критериев является критерий χ^2 (хи-квадрат) [50]. Данный тест – это основная методика для оценки последовательности из набора множества методов. Для вычисления критерия χ^2 предположим, что каждое из генерируемых системой чисел может принадлежать к одной из k категорий (т.к. сеть Хопфилда содержит 9 нейронных элементов, то количество категорий равно $2^9 = 512$). Проведём n независимых наблюдений, что означает независимость исхода текущего наблюдения от остальных. Вычислим значение по формуле:

$$\chi^2 = \sum_{s=1}^k \frac{(Y_s - np_s)^2}{np_s}, \quad (4.4)$$

где p_s – вероятность того, что результат испытаний попадёт в категорию s , Y_s – число испытаний, реально попавших в категорию s . Для оценки полученного

результата используются таблицы распределения χ^2 . В строке таблицы расположены степени свободы $v = k - 1$, а в столбцах вероятности p . Если в таблице в строке v и колонке p будет находиться некоторое число, то это означает, что значение χ^2 будет больше заданного числа с вероятностью p .

Таблица 4.3 Распределение с числом степеней свободы равным 511

	$p = 1\%$	$p = 5\%$	$p = 25\%$	$p = 50\%$	$p = 75\%$	$p = 95\%$	$p = 99\%$
$v = 511$	439,583	459,577	489,09	510,334	532,184	564,696	588,298

После наблюдения последовательности из 3000 чисел вычисленный критерий составил $\chi^2 = 532,4587$. Таким образом, χ^2 будет больше 532,4587 в 25% случаях, что свидетельствует о приемлемом прохождении данного теста.

4.4 Анализ результатов выполнения маршрутизации на основе оценки качества канала связи

Алгоритм оценки качества канала связи был описан в разделе 2.7. На рисунке 4.36 показ график частот генераторов, сконфигурированных и настроенных на канал связи. На графике показаны частоты колебаний генераторов. В данном случае, частоты принимаются за эталонные, и любое отклонение от полученного графика может свидетельствовать о наличии несанкционированного подключения к каналу передачи информации. Изменение частоты колебаний может быть следствием не только подключения к каналу связи, но и физическим повреждением кабеля или экрана, наличием рядом с кабелем источников помех, повреждение изоляции жил кабеля и т.д. Комитет нейронных сетей обучается выявлять именно несанкционированное подключение по характеру изменения частоты генераторов и степени отклонения частот колебаний от эталонных значений.

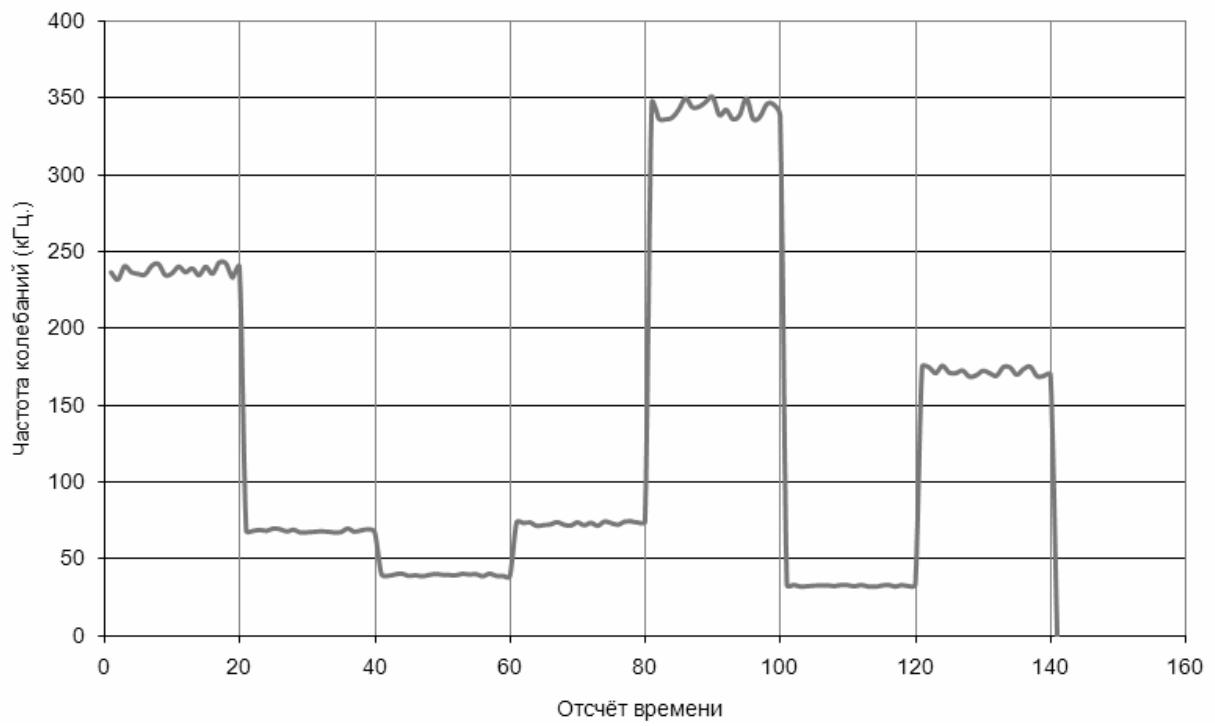


Рисунок 4.36 – Ожидаемые частоты генераторов при эталонном состоянии канала

В качестве тестового канала связи применялся кабель типа «витая пара» длиной 21 метр. После подключения на расстоянии 9 метров от генератора зондирующих импульсов параллельного отвода длиной 8 метров произошло изменение частоты генератора на интервале от 40 до 60 (рисунок 4.37). В данный промежуток времени частота генератора уменьшилась, что было спровоцировано подключением к каналу связи и увеличению его ёмкости. Увеличение ёмкости в RC цепочке генератора приведёт к уменьшению частоты колебаний.

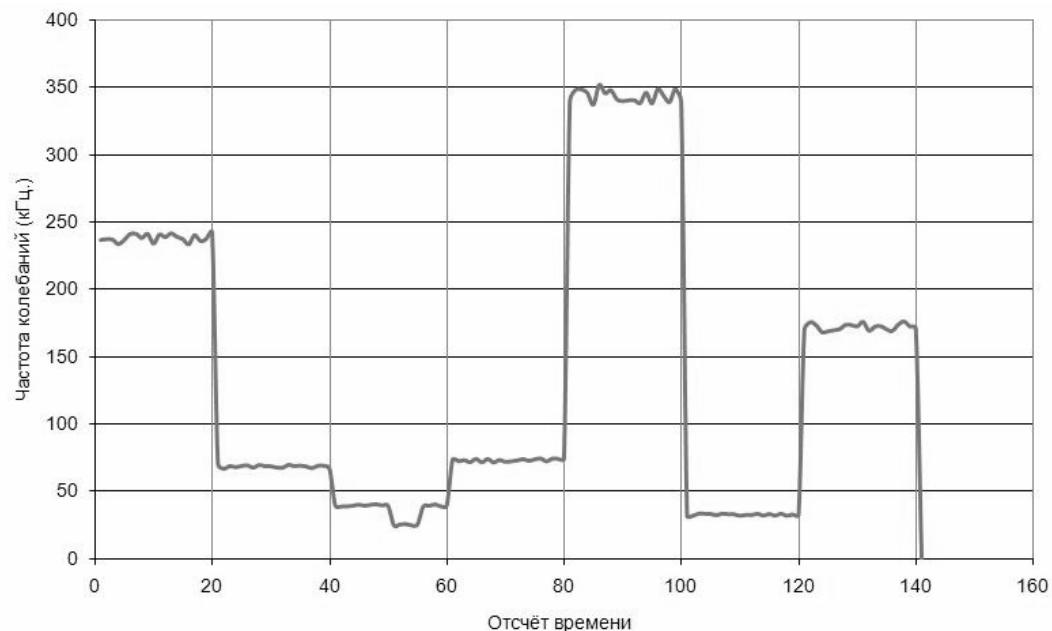


Рисунок 4.37 – Изменение частоты на интервале времени от 51 до 56

Резкое увеличение ёмкости витой пары по сравнению с её номинальным значением может свидетельствовать о наличии влаги в сердечнике кабеля. Необходимо выделять данные ситуации с целью уменьшения частоты ложных срабатываний.

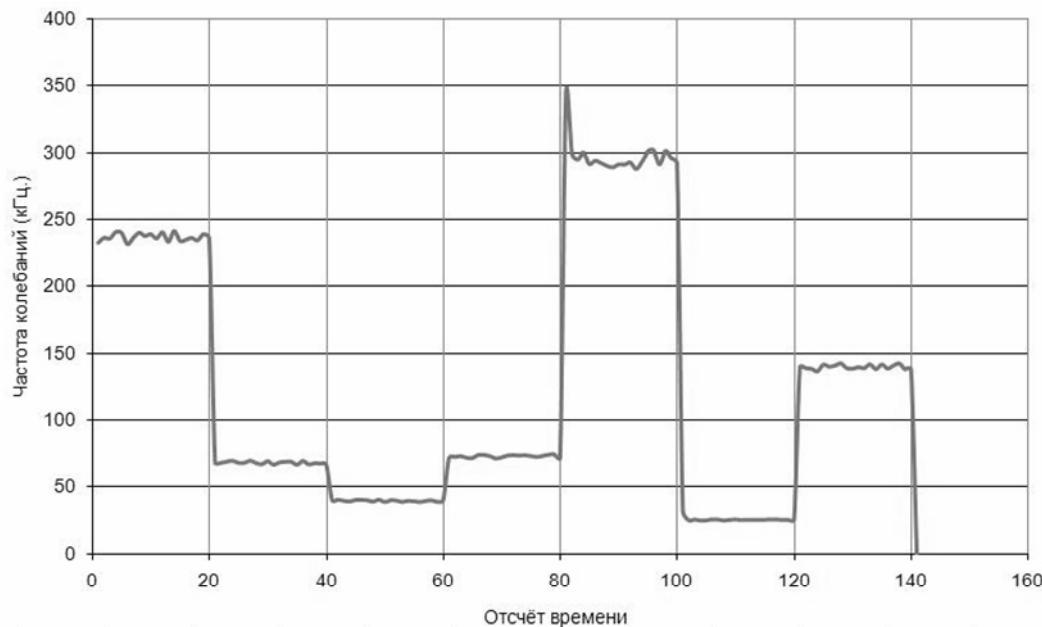


Рисунок 4.38 – Изменение частоты генераторов на интервале времени от 80 до 140

На рисунке 4.38 видно, что на интервале времени от 80 до 140 наблюдается резкое уменьшение частот генераторов, в результате увеличения ёмкости канала связи.

На входы ассоциативной машины подаётся информация о разнице эталонной частоты колебаний и фиксируемой в данный момент частоте, но только после выполнения процедуры масштабирования и приведения значений в отрезок от 0 до 1. В данной работе у каждой нейронной сети, входящей в ассоциативную машину, группа рецепторов состояла из трёх подгрупп, содержащих по 10 рецепторов в каждой. На каждую подгруппу подавалась информация об изменении соответствующей частоты за единицу времени наблюдения (в данной работе за две секунды) от соответствующего генератора. Выходной слой содержал 10 нейронов, каждый нейрон соответствовал определённой детектируемой угрозе. Прежде чем задействовать ассоциативную машину для обнаружения вторжения, состояние канала передачи информации фиксировали на интервале времени в 60 секунд с интервалом в 2 секунды, что соответствует

оценке параметров канала на трёх различных частотах (частоты генераторов изменялись каждые 20 секунд). Формирование значений элементов выборки для обучения НС осуществлялось по формуле:

$$n_i = \frac{|freq_{obs} - freq_{st}|}{freq_{st}}, \quad (4.5)$$

где $n_i = 1..30$ – значение i -ой компоненты обучающей выборки, $freq_{obs}$ – фиксируемая частота, $freq_{st}$ – ожидаемое значение частоты на данном временном интервале. Для примера, приведённого на рисунке 4.37, одна из сигнатур для обучающей выборки может быть сформирована так, как показано в таблице 4.4. Первый выходной нейрон соответствовал отсутствию угрозы вторжения, второй нейрон был сопоставлен началу вторжения в сеть (сеть предварительно исследуется перед началом вредоносной деятельности), третий нейрон соответствовал попаданию в канал жидкости и отсутствию вредоносного поведения и т.д. Выходные нейроны сопоставлялись угрозам, которые необходимо было детектировать в локальной сети связи. На основе информации, полученной путём мониторинга канала связи, строятся элементы обучающей выборки. При идентификации поведения на рецепторы экспертов подаётся информация о состоянии сетевых каналов связи. На основе этого эксперты принимают решение о наличии или отсутствии вредоносного поведения.

Необходимо также учесть и поведение сети во время обнаружения вторжения. В случае, если ансамбль НС обнаруживает только признаки физического подключения к каналу и отсутствие какой-либо активности атакующей стороны, то нет необходимости прекращать передачу данных, если используется стойкий криптографический алгоритм шифрования передаваемой информации.

Таблица 4.4 Пример образца обучающей выборки для ассоциативной машины

i	Отсчёт времени (сек.)	Значение ожидаемой частоты $freq_{st}$ (кГц.)	Значение фиксируемой частоты $freq_{obs}$ (кГц.)	n_i	Желаемое значение выходных нейронов
1	1	237	242,3682	0,0226	0100000000
2	3	237	237,4873	0,0020	
3	5	237	241,4417	0,0187	
4	7	237	239,964	0,0125	
5	9	237	245,9534	0,0377	
6	11	237	238,2143	0,0051	
7	13	237	240,1977	0,0134	
8	15	237	242,7183	0,0241	
9	17	237	238,472	0,0062	
10	19	237	246,0638	0,0382	
11	21	70	72,54652	0,0363	
12	23	70	70,69781	0,0099	
13	25	70	73,45286	0,0493	
14	27	70	70,0729	0,0010	
15	29	70	72,45329	0,035	
16	31	70	70,79125	0,0113	
17	33	70	71,28306	0,0183	
18	35	70	72,12428	0,0303	
19	37	70	73,08958	0,0441	
20	39	70	70,43456	0,0062	
21	41	37	38,06307	0,0287	
22	43	37	38,56843	0,0423	
23	45	37	37,21083	0,0056	
24	47	37	37,23643	0,0063	
25	49	37	10,28053	0,7221	
26	51	37	10,09115	0,7272	
27	53	37	10,29551	0,7217	
28	55	37	37,58171	0,0157	
29	57	37	38,42204	0,0384	
30	59	37	37,90976	0,0245	

Вторая группа элементов обучающей выборки строилась на основе анализа спектра «отражённого» сигнала. Возвращённый сигнал имеет вид, показанный на рисунке 4.39. Спектр возвращенного сигнала для эталонного канала связи показан на рисунке 4.40. После подключения параллельного отвода длиной 8 метров произошло изменение спектра сигнала (рисунок 4.41). По оси абсцисс ведётся отсчёт времени с точностью до тысячных долей секунды, а по оси ординат – отсчёты величины амплитуды сигнала в вольтах. В момент времени 5,158 сек. начинается процесс разряда конденсатора через катушку индуктивности. Переходный процесс завершится к моменту времени 5,164.

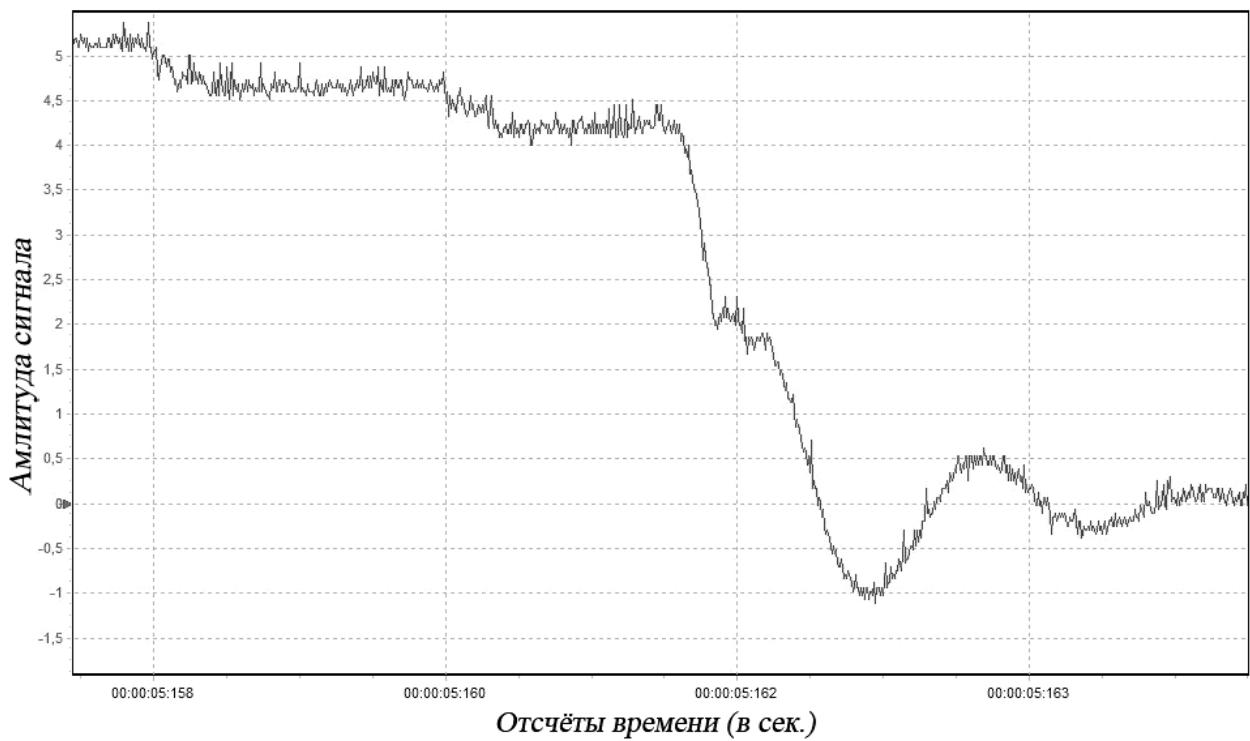


Рисунок 4.39 – Осциллограмма «отражённого» сигнала

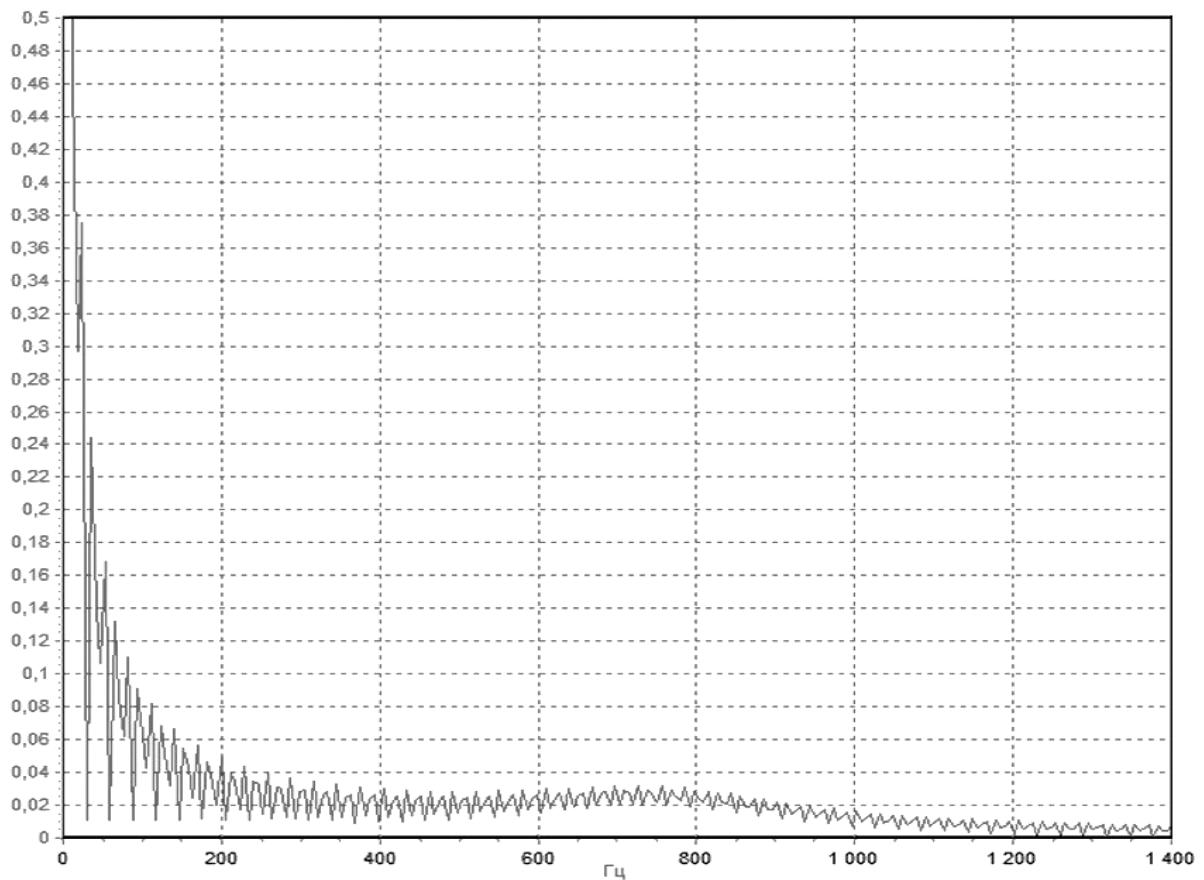


Рисунок 4.40 – Результат выполнения ДПФ для сигнала с эталонного канала связи

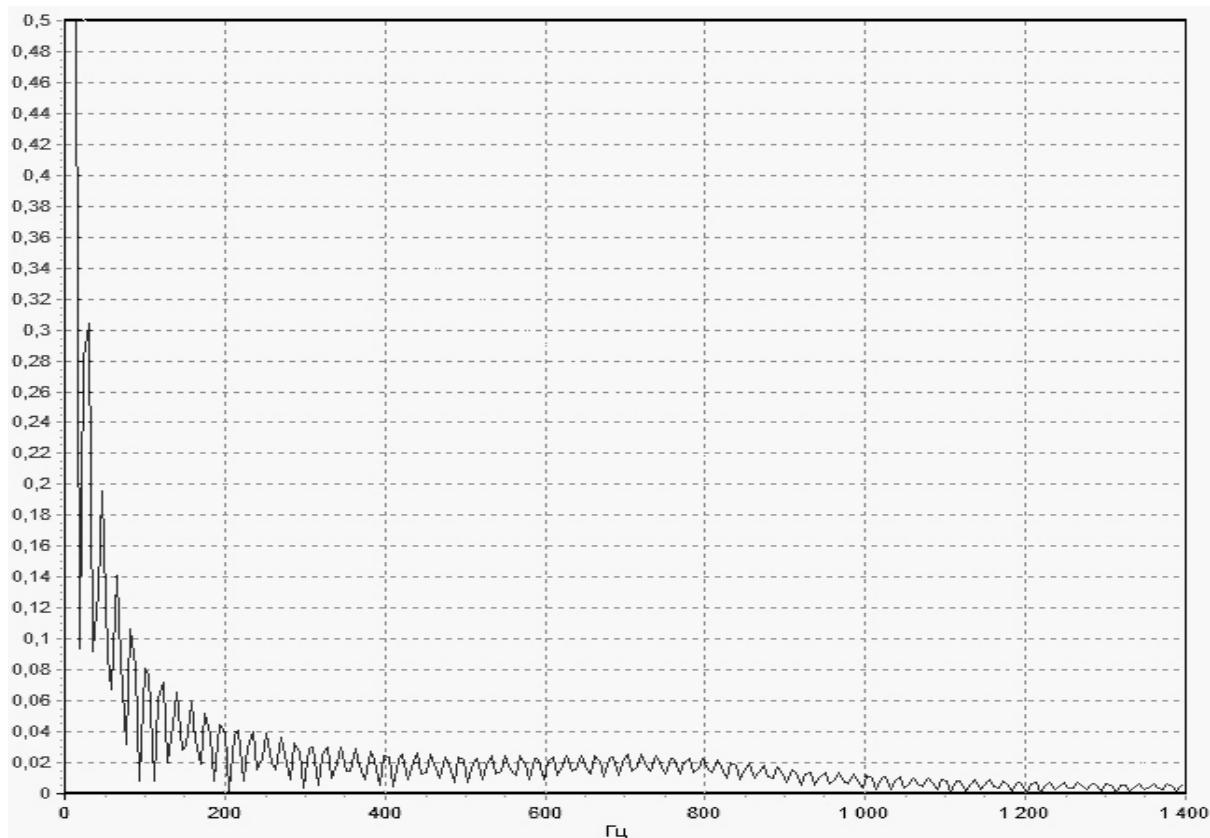


Рисунок 4.41 – Результат выполнения ДПФ для сигнала с тестового канала связи

На основе полученных данных становится возможным сформировать обучающую и тестовую выборку для нейронной сети.

Рассмотренные два способа формирования исходных данных для нейросетевого эксперта в совокупности со статистическими оценками канала используется для формирования обучающей и тестовой выборки нейросетевой ассоциативной машины. Вектор входных данных будет иметь следующую структуру:

Координаты узла назначения	Пропускная способность канала	Надёжность	Изменение ёмкости и сопротивления канала	Результат анализа спектра «отражённого сигнала»
----------------------------	-------------------------------	------------	--	---

Размерность вектора входных признаков, в результате анализа которого ассоциативная машина примет решение о дальнейшем маршруте продвижения информации, будет содержать размерность равную 66 вещественным переменным, что затрудняет их последующую обработку. Для сжатия данных применялась карта Кохонена.

Для получения состояния исходящих соединений, данные вектора, характеризующие все соединения, подаются на карту Кохонена [61] (рисунок 4.42). Для того чтобы не происходила изоляция узлов, все узлы сети должны иметь информацию об исходящих соединениях с узлами, с которыми они имеют связь. Для этого в компьютерной сети будут присутствовать моменты времени по обмену векторами состояния соединений и добавление их к локальным векторам исходящих соединений. Использование карты Кохонена позволяет провести сжатие данных, что уменьшает время функционирования нейроконтроллера. Таким образом, в каждом узле получаем карту о состоянии исходящих соединений. Для получения следующего пункта в передаче данных алгоритм подаёт векторы состояния сетевых соединений с карты Кохонена на входы ассоциативной машины и получает следующий пункт в построении маршрута данных.

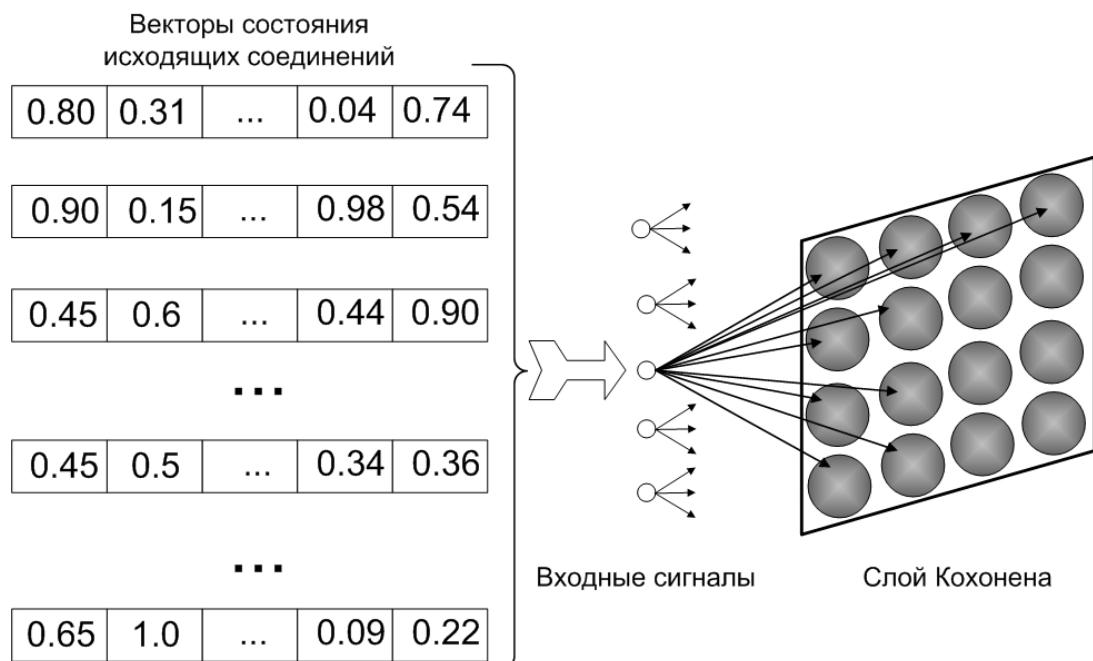


Рисунок 4.42 – Схема компрессии данных перед обработкой

Проверка эффективности обученного комитета НС была исследована на модели компьютерной сети (рисунок 3.14). В качестве параметров сетевых соединений рассматривались пропускная способность и качество каналов связи [13]. Как альтернатива использования нейроконтроллера был рассмотрен алгоритм маршрутизации OSPF, выполняющий поиск оптимального маршрута с помощью алгоритма Флойда [76] в совокупности с алгоритмом нахождения

максимального потока в сети, основанным на теореме Форда-Фалкерсона. Цель сравнения двух алгоритмов – получение зависимости задержки по времени прохождения пакетов между узлами сети от количества изменений в структуре сети связи. Результат сравнения алгоритмов представлен на рисунке 4.43. Результаты экспериментов показали, что комитет экспертов способен адаптивно реагировать на изменение топологии связей компьютерной сети.

Адекватная реакция экспертов позволит сохранить оптимальную скорость передачи данных даже в случае направленного вредоносного поведения атакующей стороны.

Среднее значений комитета дает лучшие предсказания, чем средний эксперт из этого же комитета, причем снижение ошибки может быть довольно заметным. Таким образом, метод комитетов может существенно повысить качество прогнозирования.

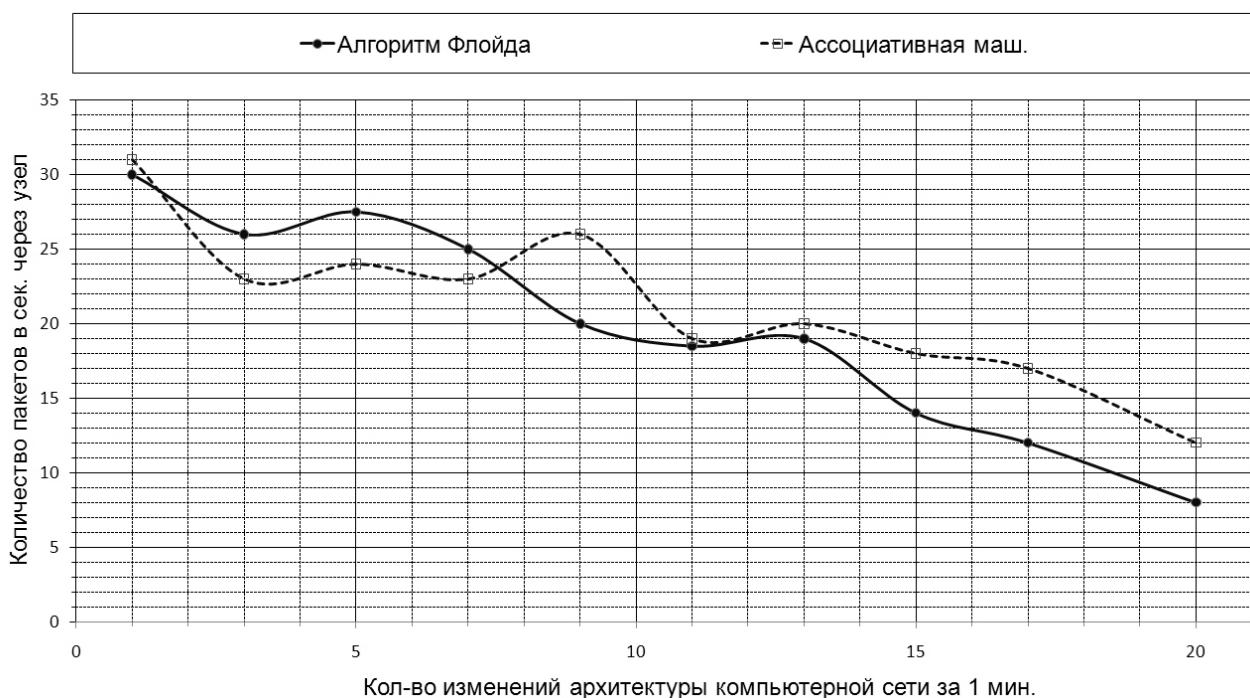


Рисунок 4.43 – Сравнение результатов работы алгоритмов

Как уже отмечалось ранее, использование нейросетевых экспертов при решении задачи безопасной маршрутизации в локальной сети базируется на оценке зависимости изменения параметров каналов передачи информации. Поскольку выявить все такие факторы (и степень их влияния) однозначно, как

правило, сложно, это является дополнительным аргументом в пользу использования не единственной нейронной сети, а комитета нейронных сетей.

Сети-эксперты, входящие в нейросетевой комитет, отличаются архитектурами, типами нейронов, начальными значениями весовых коэффициентов и т.д. Задача подготовительного этапа в этом случае заключается еще и в эффективной организации комитета, исключении из него заведомо «некомпетентных» (дающих неудовлетворительные прогнозы) сетей-экспертов, разработка механизма согласования прогнозов с целью получения единого решения.

В рассматриваемой ассоциативной машине все нейронные сети обучаются на одних и тех же данных, но в комитет изначально включаются сети, отличающиеся принципиально разной архитектурой (многослойный персепtron, рекуррентная сеть Эльмана, RBF – сеть). Каждая из этих сетей решает задачу нелинейной аппроксимации. Поэтому разница в прогнозах, выдаваемых экспертами, будет следствием не только случайности начальных значений весовых коэффициентов сетей, но и принципиального различия этих сетей.

После получения ответов, сигналы от нейросетевых экспертов проходят обработку с помощью модифицированных нейронов Фукушимы, квадратичного нейрона и N – адалины. Объединение экспертов в ансамбль, при решении задачи обнаружения вторжений, производится финальной нейронной сетью, представленной сигма-пи нейронной сетью. Предложенный подход является примером нелинейного объединения алгоритмов в комитет.

Эффективность применения нейросетевой ассоциативной машины оценивалась по качеству выполнения обобщения и корректности выдаваемых результатов, как показано на рисунке 4.44.

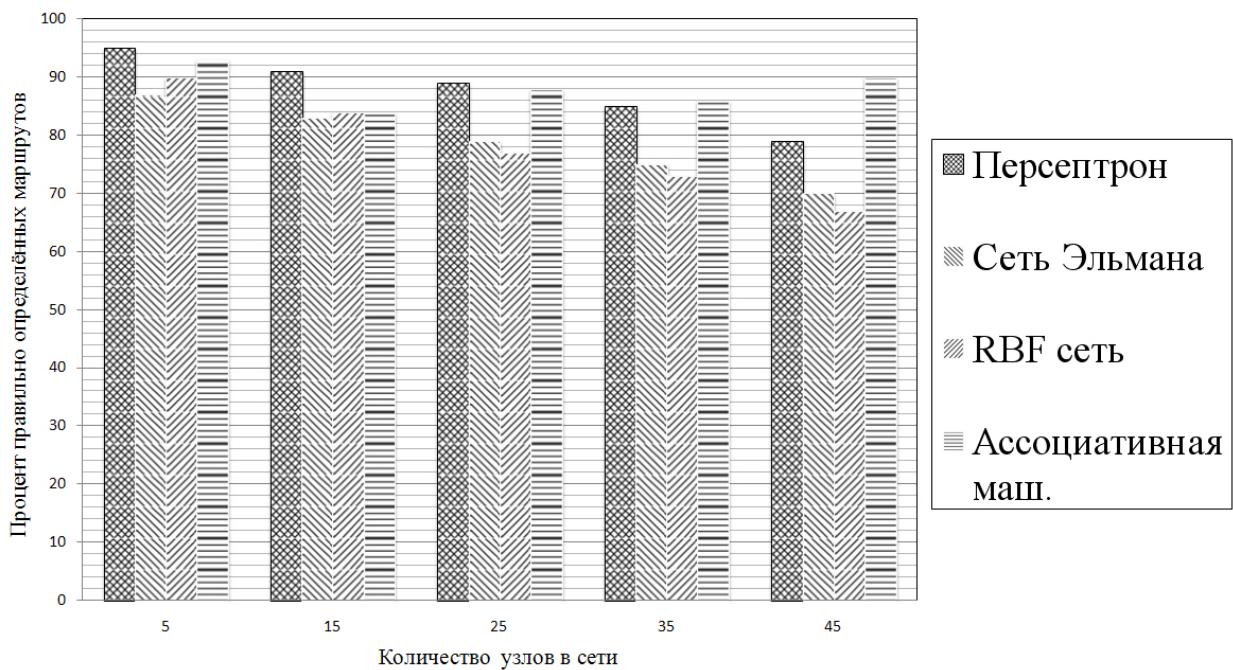


Рисунок 4.44 – Сравнение эффективности отдельных НС и ассоциативной машины

4.5 Выводы по главе 4

1. Спроектированные методы обучения компонентов комитета нейросетевых экспертов показали свою эффективность при выполнении процедуры обучения нейронных сетей. В одинаковых экспериментальных условиях динамика процесса обучения улучшилась по сравнению со стандартными градиентными алгоритмами обучения нейронных сетей.
2. Система по предобработке сигналов от нейросетевых экспертов показала свою эффективность при выполнении уменьшения амплитуды сигналов от экспертов, компетентность которых оценивалась с помощью спроектированной нечёткой модели.
3. Генератор случайных чисел на основе сетей Хопфилда с источником энтропии успешно прошёл тесты на случайность и предсказуемость генерируемых последовательностей, что доказывает его эффективность при обеспечении безопасного функционирования всего нейросетевого комитета.
4. Анализ каналов связи на основе детектирования изменения параметров кабельных линий и оценки спектра отражённого сигнала позволяет сформировать обучающую выборку, которая необходима для получения оптимальных маршрутов для безопасной передачи информации.

Заключение

1. Проведён анализ современных алгоритмов построения оптимальных маршрутов в телекоммуникационных сетях. Для устранения выявленных недостатков разработан нейросетевой алгоритм маршрутизации, основанный на механизме объединения решений трёх различных нейронных сетей.
2. Выполнена разработка гибридного алгоритма обучения нейронных сетей на основе стандартных градиентных алгоритмов, но модифицированных эвристиками. Разработанные модификации алгоритмов позволили повысить эффективность обучения НС по сравнению со стандартными градиентными методами настройки параметров нейронных сетей.
3. Спроектированный кооперативный иммунный алгоритм настройки параметров нейросетевых структур перед процессом обучения позволяет начать процедуру обучения с начального состояния, приближенного к оптимальной конфигурации нейросетевого эксперта. Данная модификация позволяет сократить время настройки всего комитета НС.
4. Разработанная система объединения мнений экспертов, синтезированная из модифицированных нейронных элементов Фукушимы, N -адалины и квадратичного нейрона, показала высокую эффективность обработки сигналов от нейросетевых экспертов на основе информации, получаемой от нечёткой подсистемы оценки качества обучения отдельных нейронных сетей.
5. Применение множества сетей Хопфилда в качестве генератора случайных чисел позволяет начать процедуру обучения нейронных сетей из случайного начального состояния. Применение источника энтропии, основанного на стохастическом процессе перезаряда двух ёмкостей, делает невозможным прогнозирование состояний всей системы в произвольные моменты времени. Тестирование системы показало высокий уровень защищённости нейросетевого маршрутизатора.
6. Разработанная система оценки параметров надёжности и безопасности канала передачи информации на основе оценки изменений параметров линии

передачи информации позволяет оперативно сформировать обучающую выборку для надёжного функционирования комитета НС.

7. Оценка возможностей обученного комитета сетей выполнять процедуру маршрутизации в условиях неполной информации об отдельных каналах передачи информации показала способность комитета принимать решение о продвижении данных в условиях деградации телекоммуникационной сети.

8. В результате выполнения исследования был разработан программный продукт, в котором реализованы все разработанные алгоритмы для обучения комплекса из трёх нейронных сетей решать задачу безопасной маршрутизации информации.

Список литературы

1. Аверкин, А.Н. Нечеткие множества в моделях управления и искусственного интеллекта / А.Н. Аверкин, И.З. Батыршин, А.Ф. Блишун и др.; под ред. Д.А.Поспелова. – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 312 с. – (Проблемы искусственного интеллекта).
2. Аверченков В.И. Эволюционное моделирование и его применение: монография / В.И. Аверченков, П.В. Казаков. – Брянск: БГТУ, 2009. – 199 с.
3. Акулов О.А. Информатика: базовый курс: учеб. для студентов вузов, бакалавров, магистров, обучающихся по направлению «Информатика и вычисл. техника» / О.А. Акулов, Н.В. Медведев. – 5-е изд., испр. и доп. – М.: Омега-Л, 2008. – 574 с.
4. Алексеев В.М., Тихомиров В.М., Фомин С.В. Оптимальное управление. – М.: Наука. Главная редакция физико-математической литературы, 1979. – 430 с.
5. Анфилатов В.С. Системный анализ в управлении: Учеб. пособие / В.С. Анфилатов, А.А. Емельянов, А.А. Кукушкин; Под ред. А.А. Емельянова. – М.: Финансы и статистика, 2002. – 368 с.: ил.
6. Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: Учебник для вузов / Под ред. Зарубина В.С., Крищенко А.П. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 440 с. (Серия Математика в техническом университете; Выпуск XIV).
7. Бакланов И.Г. NGN: принципы построения и организации / под ред. Ю.Н. Чернышова. – М.: Эко-Трендз, 2008. – 400 с.: ил.
8. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с.: ил.
9. Барский А.Б. Логические нейронные сети: учеб. пособие / А.Б. Барский. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2012. – 352с.

10. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176 с.: ил. – (Прикладные информационные технологии).
11. Батищев Д.И. Генетические алгоритмы решения экстремальных задач: Учебное пособие. Воронеж, 1995. – 69 с.
12. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304с.
13. Бигелоу С. Сети: поиск неисправностей, поддержка и восстановление: Пер. с англ. – СПб.: БХВ-Петербург, 2005. – 1200 с.: ил.
14. Бионические информационные системы и их практические применения/Под ред. Л.А. Зинченко, В.М. Курейчика, В.Г. Редько. – М.: ФИЗМАТЛИТ, 2011. – 288 с. – ISBN 978-5-9221-1302-1.
15. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения. – Харьков: ТЕЛЕТЕХ, 2004 – 369 с.: ил.
16. Бойт К. Цифровая электроника.–Москва: Техносфера, 2007. – 472 с.
17. Бокс, Дж. Анализ временных рядов, прогноз и управление. В 2-х т.: Пер. с английского / Г.М. Дженкинс, Дж. Бокс М.: Издательство «Мир», 1974. – 608 с.
18. Борисов В. В., Круглов В. В., Федулов А. С. Нечёткие модели и сети. – 2-е изд., стереотип. – М.: Горячая линия – Телеком, 2012. – 284 с.: ил.
19. Брокмайер Джо, Лебланк Ди-Энн, Маккарти Рональд. Маршрутизация в Linux.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 240 с.:ил. – Парал. тит. англ.
20. Бэстенс Д.-Э., ван ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки: принятие решений в торговых операциях. – Москва: ТВП, 1997. – 236 с.
21. Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления с использованием генетических алгоритмов: Учебное пособие/Уфимск. гос. авиац. техн. ун-т. – Уфа, 1999. – 105с.
22. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и её инженерные приложения. Учеб. пособие для втузов. – 2-е изд., стер. – М.:Высш.шк., 2000. – 480 с.: ил.

23. Волков И.К., Зуев С.М., Цветкова Г.М. Случайные процессы: Учебник для вузов / Под ред. Зарубина В.С., Крищенко А.П. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 448 с. (Серия Математика в техническом университете; Выпуск XVIII).
24. Вороновский Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Х.: ОСНОВА, 1997. – 112 с.
25. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 384 с.: ил.
26. Гадре Д. Занимательные проекты на базе микроконтроллеров tinyAVR / Дхананья Гадре, Нигул Мэлхотра: Пер. с англ. – СПб.: БХВ-Петербург, 2012. – 352 с.: ил.
27. Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 608 с : ил. – (Сер. Информатика в техническом университете.)
28. Галушкин А.И. Нейрокомпьютеры. Кн.3/Под общ ред. А. И. Галушкина. – М.: ИПРЖР, 2000.
29. Галушкин, А. И. Нейроматематика (проблемы развития). — М.: Радиотехника; 2003. — 40 с.
30. Галушкин А.И. Теория нейронных сетей. Кн. 1: Учеб. пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2000. – 416 с.: ил. (Нейрокомпьютеры и их применение).
31. Гладков Л.А., Зинченко Л.А., Курейчик В.В., Курейчик В.М., Лебедев Б.К., Лебедев В.Б., Лебедев О.Б., Нужнов Е.В., Сороколетов П.В. Оптимизация на основе методов гомеостатики, эволюционного развития и самоорганизации. Монография / Под ред. В.М. Курейчика. Таганрог: Изд-во ТРТУ, 2006. – 308 с.
32. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы: Учебное пособие. Под ред. В.М. Курейчика. – Ростов-на-Дону: ООО «Ростииздат», 2004. – 400 с.

33. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы:/ Под ред. В.М. Курейчика. – 2-е изд., испр. и доп. М:ФИЗМАТЛИТ, 2006. -320 с.
34. Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов В.М., Биоинспирированные методы в оптимизации. М.:ФИЗМАТЛИТ, 2009.–384 с.
35. Гмурман В.Е., Теория вероятностей и математическая статистика: Учеб. пособие для вузов/В.Е.Гмурман. – 9-е изд., стер. – М.: Высш.шк., 2003. – 479 с.: ил.
36. Головко В. А. Нейронные сети: обучение, организация и применение. Кн. 4. Учеб. пособие для вузов / Общая ред. А.И. Галушкина. — М.: ИПРЖР, 2001. – 256 с.
37. Горбань А. Н. Обучение нейронных сетей. — М.: СП ПараГраф,, 1990. — 160 с.
38. Горбань А. Н., Россиев Д. А. Нейронные сети на персональных компьютерах. — Новосибирск: Наука, 1996. – 276 с.
39. Гужва А.Г., Доленко С.А., Персианцев И.Г., Шугай Ю.С. Многоступенчатый алгоритм на основе комитета нейронных сетей для анализа многомерных временных рядов // «Нейрокомпьютеры: разработка, применение» : науч.-тех. журн. 2010. № 3
40. Дасгупта Д. Искусственные иммунные системы и их применение. М.: ФИЗМАТЛИТ, 2006. – 344с.
41. Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка, М.С. Олдендерфер, Р.К. Блэшфилд. Факторный, дискриминантный и кластерный анализ: Пер. с англ./ Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка и др.; Под ред. И.С. Енюкова. – М.: Финансы и статистика, 1989. – 215 с.: ил.
42. Джонс М.Т. Программирование искусственного интеллекта в приложениях. Пер. с англ. Осипов А.И. – М.: ДМК Пресс, 2006 – 312 с.
43. Ежов А.А. Шумский С.А. Нейрокомпьютинг и его применение в экономике и бизнесе.– М.: МИФИ, 1998. – 224с.
44. Емельянов, В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2003.

45. Жданов А. А. Метод автономного адаптивного управления // Новости искусственного интеллекта, 2003, №5.
46. Жульков Е.В., Томилин В.Н. Поиск уязвимостей сетевых систем обнаружения вторжения. // Проблемы информационной безопасности. Компьютерные системы. СПб.: 2003. – №2.
47. Жульков Е.В., Томилин В.Н. Поиск уязвимостей сетевых систем обнаружения вторжения. // Открытые системы. М.: 2004. – №7-8.
48. Заде Л. Понятие лингвистической переменной и его применение к принятию приближённых решений / под ред. Н.Н. Моисеева, С.А. Орловского. – М.: Мир, 1989.
49. Зарубин В.С. Математическое моделирование в технике: учеб. для вузов / В.С. Зарубин. – 3-е изд. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. – 495, [1] с. – (Серия «Математика в техническом университете». Вып. XXI, заключительный).
50. Иванов М.А., Чугунков И.В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. – М.: КУДИЦ-ОБРАЗ, 2003. – 240 с. – (СКБ – специалисту по компьютерной безопасности).
51. Игошин В.И. Математическая логика и теория алгоритмов: учеб. пособие для студ. высш. учеб. заведений / В.И. Игошин. – 2-е изд., стер. – М.: Издательский центр «Академия», 2008. – 448 с.
52. Казаков И.Е., Гладков Д.И. Методы оптимизация стохастических систем. – М.: Наука. Гл. ред. физ.-мат., 1987. – 304 с.
53. Калан, Роберт. Основные концепции нейронных сетей.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 287 с.: ил. – Парал. тит. англ.
54. Колмогоров, А. Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // ДАН СССР. 1956. – Т. 108. – №2. – С. 179 – 182.
55. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: Учебное пособие для вузов. – 2-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 400 с.: ил. – (Информатика в техническом университете.).

56. Комарцова Л.Г. Исследование алгоритмов обучения многослойного персептрона // Нейрокомпьютеры. Разработка и применение. М.: Радиотехника. 2002. № 12.
57. Комарцова, Л.Г. Исследование нейросетевых и гибридных методов и технологий в интеллектуальных системах поддержки принятия решений: дис. ... доктора техн. наук. – Калуга: КФ МГТУ им. Н.Э.Баумана, 2003.
58. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в системах управления и связи. – М.: Горячая линия – Телеком, 2003. – 94 с.
59. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ: пер. с англ. М.: МЦНМО, 2001. – 960с.
60. Кохонен Т. Ассоциативные запоминающие устройства: Пер. с англ. – М.: Мир, 1982. – 384 с., ил.
61. Кохонен Т. Самоорганизующие карты; пер. 3-го англ. изд. – М.: БИНОМ. Лаборатория знаний, 2010. – 655 с.
62. Курейчик В.В. Эволюционные, синергетические и гомеостатические методы принятия решений: монография / В.В. Курейчик. – Таганрог: Изд-во ТРТУ, 2001.
63. Лафоре Р. Структуры данных и алгоритмы Java. Классика Computers Science. 2-е изд. – СПб. Питер, 2011. – 704 с.: ил.
64. Лачин В.И. Электроника: учеб. пособие / В.И. Лачин, Н.С. Савёлов. – Изд. 7-е. – Ростов н/Д: Феникс, 2009. – 703 с.
65. Леоненков А. В. Нечёткое моделирование в среде MATLAB и Fuzzy Tech. -СПб.: БХВ-Петербург, 2003. 736 с.
66. Лихачева Г.Н., Гаспариан М.С. Информационные технологии: Учебно-практическое пособие. – М.: Изд. центр ЕАОИ. – 2007. – 189 с.
67. Люгер Дж. Ф. Искусственный интеллект. - Изд. 4-е. - М., Санкт-Петербург, Киев, 2003. – 485 с.
68. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 864 с.:ил. – Парал. тит. англ.

69. Мазуров В.Д. Метод комитетов в задачах оптимизации и классификации. – М.: Наука. Гл. ред. физ.-мат. Лит., 1990. – 248 с.
70. Мандель И.Д. Кластерный анализ. – М.: Финансы и статистика. 1988. – 176 с.: ил.
71. Матвеев М.Г. Модели и методы искусственного интеллекта. Применение в экономике: учеб. пособие. – М.: Финансы и статистика; ИНФРА-М, 2008. – 448 с.
72. Матвеев М.Г. Система поддержки принятия решений в условия нечёткой информации / М.Г. Матвеев, А.С. Свиридов, М.Е. Семенов // Вестник ВГТУ. Серия «Вычислительные и информационно-телекоммуникационные системы». – Вып. 8.1. – Воронеж: ВГТУ, 2001.
73. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем – СПб.: Наука и Техника, 2003. – 384 с.:ил.
74. Немнюгин С. Параллельное программирование для многопроцессорных вычислительных систем / С. Немнюгин, О. Стесик СПб.: «БХВ-Петербург», 2002. – 400 с.
75. Новак Д., Норткэт С., Маклахлен Д. Как обнаружить вторжение в сеть. Настольная книга специалиста по системному анализу. – Издательство «Лори», 2012. – 384 с.
76. Новиков Ф.А. Дискретная математика для программистов: Учебник для вузов. 3-е изд. – Спб.: Питер, 2009. – 384с.: ил.
77. Ногл М. TCP/IP. Иллюстрированный учебник – М.: ДМК Пресс, 2001. – 480 с.: ил.
78. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е издание. Спб.: Питер, 2006.
79. Олифер В., Олифер Н. Новые технологии и оборудование IP-сетей. - СПб: Изд-во "Питер", 2000. 512с.
80. Олифер В.Г., Олифер Н.А. Сетевые операционные системы, 2-е изд. Спб.: Питер, 2008.

81. Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.: ил.
82. Остерлох Хезер. TCP/IP. Семейство протоколов передачи данных в сетях компьютеров: Пер. с англ./Хезер Остерлох – СПб.: ООО «ДиаСофтЮП», 2002. – 576 с.
83. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям: Учебное пособие. 2-е изд., испр. – СПб.:Питер, 2013. – 704 с.: ил.
84. Панченко Т.В., Генетические алгоритмы: учебно-методическое пособие / под ред. Ю.Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.
85. Паркер Тим, Сиян Каранжит. TCP/IP. Для профессионалов. 3-е изд. / Т. Паркер, К. Сиян. – СПб.: Питер, 2004. – 859 с.: ил.
86. Пегат А. Нечёткое моделирование и управление / А. Пегат; пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2012. – 798 с.: ил. – (Адаптивные и интеллектуальные системы).
87. Печинкин А.В., Тескин О.И., Цветкова Г.М., Теория вероятностей: Учебник для вузов / Под ред. Зарубина В.С., Крищенко А.П. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 456 с. (Серия Математика в техническом университете; Выпуск XVI).
88. Под ред. Поспелова Д. А. Нечёткие множества в моделях управления и искусственного интеллекта – М.: Наука, 1986. – 312 с.
89. Поляк-Брагинский А.В. Локальная сеть. Самое необходимое. – 2-е изд., перераб. И доп. – СПб.: БХВ-Петербург, 2011. – 576 с.: ил.
90. Попов Э.В., Фридман Г.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. М.: Наука, 1976. – 455 с.
91. Прикладные нечеткие системы. Под ред. Т. Тэррано, К. Асай, М. Сугено. – М.: Мир, 1993. – 368 с., ил.
92. Растигин Л.А. Адаптация сложных систем. – Рига: Зинатне, 1981. – 375 с.

93. Ратнер А.Н. Генетика. Молекулярная кибернетика. – Новосибирск: Наука, 2002. – 272 с.
94. Реклейтис А., Рейвиндран А. и др. Оптимизация в технике. – В 2-х кн. – М.:Мир, 1988. – 670 с.
95. Розенблatt, Ф. Принципы нейродинамики: персептроны и теория механизмов мозга. М.: Мир, 1965. – 480 с.
96. Ройт А, Бростофф, Мейл Д. Иммунология: Пер. с англ. М., изд-во Мир,2000. – 592 с.
97. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечёткие системы: Пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2006. – 452 с.: ил.
98. Сергиенко, Р.Б. Генетический алгоритм решения сложных задач оптимизации / Р.Б. Сергиенко, Е.С. Семенкин // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решентнева. №2 (23). – 2009. – С. 17-21.
99. Синдеев Ю.Г. Электротехника с основами электроники: учебное пособие / Ю.Г. Синдеев. – Изд. 10-е – Ростов н/Д: Феникс, 2008. – 407, [1] с. – (НПО).
100. Скиена С. Алгоритмы. Руководство по разработке. – 2-е изд.: Пер с англ. – СПб.:БХВ-Петербург, 2011. – 720 с.
101. Склар Бернард. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с.: ил. – Парал. тит. англ.
102. Соболь Б.В. Практикум по вычислительно математике / Б.В. Соболь, Б.Ч Месхи, И.М. Пешхоеv. – Ростов н/Д: Феникс, 2008. – 342, [1] с. – (Высшее образование).
103. Столингс В. Современные компьютерные сети. СПб.: Питер. 2003. – 783 с.
104. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация. – СПб.: Питер, 2007. – 704 с.: ил.

105. Терехов В.А. Нейросетевые системы управления: Учеб. пособие для вузов / В.А. Терехов, Д.В. Ефимов, И.Ю. Тюкин. – М.: Высш. шк. 2002. – 183 с.: ил.
106. Томас Том М. Структура и реализация сетей на основе протокола OSPF, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 816 с.: ил. – Парал. тит. англ.
107. Трекин А.Г. Структурный синтез вычислительной системы с помощью генетических алгоритмов: дис. ... канд. физ.-мат. наук / А.Г. Трекин. – М., 2002.
108. Уэйкерли Дж.Ф. Проектирование цифровых устройств, том 1. Москва: Постмаркет, 2002. – 544 с.
109. Фейт С., TCP/IP Архитектура, протоколы, реализация. – Издательство «Лори», 2000. – 424 с.
110. Хайкин С. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 1104с.
111. Хелеби Сэм, Мак-Ферсон Денни. Принципы маршрутизации в Internet, 2-е издание.: Пер с англ. М.: Издательский дом «Вильямс», 2001. – 448 с.: ил. – Парал. тит. англ.
112. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ. – Изд. 7-е. – М.: Мир, БИНОМ, 2011. – 704 с., ил.
113. Хорошевский В.Г. Архитектура вычислительных систем: Учеб. пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 512 с.:ил. – (Информатика в техническом университете).
114. Червяков Н.И., Евдокимов А.А., Галушкин А.И., Лавриненко И.Н., Лавриненко А.В. Применение искусственный нейронных сетей и системы остаточных классов в криптографии. – М.:ФИЗМАТЛИТ, 2012. – 280 с. – ISBN 978-5-9221-1386-1.
115. Чуликов В.А., Астахова И.Ф., Потапов А.С. Системы искусственного интеллекта. Практический курс: учебное пособие – М.: БИНОМ. Лаборатория знаний, 2008. – 292 с.

116. Шампандер Алек Дж. Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2007. – 768 с.: ил. – Парал. Тит.англ.
117. Шапорев С.Д. Дискретная математика. Курс лекций и практических занятий. – Спб.: БХВ – Петербург, 2007. – 400 с.
118. Шевкопляс Б.В. Вероятностная синхронизация в телекоммуникационных системах : учебное пособие / Б.В. Шевкопляс. – М.:БИНОМ. Лаборатория знаний, 2012. – 168 с.: ил.
119. Шибзухов З.М. Конструктивные методы обучения сигма-пи нейронных сетей / З.М. Шибзухов; [отв. Ред. Ю.И. Журавлев]; НИИ приклад. Математики и автоматизации Кабардин.-Балкар. НЦ РАН. – М.:Наука, 2006. – 159 с.
120. Ясницкий Л. Н. Введение в искусственный интеллект. — 1-е. — М.: Издательский центр «Академия», 2005. — 176 с.
121. Яхъяева Г.Э. Нечеткие множества и нейронные сети: Учебное пособие / Г.Э. Яхъяева. – М.: Интернет – Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 316 с.: ил., табл. – (Серия «Основы информационных технологий»).
122. Baglietto M., Battistelli G., Vitali F., Zoppoli R. Shortest path problems on stochastic graphs: a neuro dynamic programming approach // 42nd IEEE International Conference on Decision and Control. 2003. V. 6. P. 6187-6193.
123. Choi J., Lee G. The Bifurcating Neuron Network 3 // Proceedings of International Joint Conference on Neural Networks. 2005. P. 2184-2189.
124. Dong J.-Y., Wang W.-J., Zhang J.-Y. Accumulative competition neural network for shortest path tree computation // Proceedings of the 2003 International Conference on Machine Learning and Cybernetics. 2003. V. 2. P. 1157-1161.
125. Herrera F., Lozano F. Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers // Genetic Algorithms and Soft Computing. – Berlin: Physica-Verlag, 1999. – P.95-125.

126. Herrera F., Lozano F. Fuzzy Genetic Algorithms: Issues and Models // Technical Report DECSAI-98116, Dept. of Computer Science and A.I., University of Granada, June 1998.
127. Holland J.H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence. – Ann Arbor: University of Michigan Press, 1975.
128. Kirkpatrick S., Gellat C.D. Vecchi M. Optimization by Simulated Annealing. – Science. – 1983. – Vol. 220.
129. Kohonen T. // Self-Organization and Associative Memory. — Springer, Berlin, 1988.
130. Koza, John R. Genetic programming tutorial. URL: <http://www.genetic-programming.com/gpanimatedtutorial.html>.
131. Koza, John R. The Genetic Programming Paradigm: Genetically Breeding Populations of Computer Programs to Solve Problems / John R. Koza. -Cambridge, MA: MIT Press, 1992.
132. Serpen G., Livingston D.L. A neural network for path search in directed graphs // IEEE SOUTHEASTCON'90: Technol. Today and Tomorrow. 1990. V.2. P.558-561.
133. Schmitt L.J. An evaluation of a genetic algorithmic approach to the vehicle routing problem // Working paper, Department of Information Technology Management. — Christian Brothers University, Memphis, 1995.
134. Shi H., Wang L. A Hybrid Neural Network for Optimal TDMA Transmission Scheduling in Packet Radio Networks // Proceedings of International Joint Conference on Neural Networks. 2005. P. 3210-3213.
135. Taillard E.D. Parallel iterative search methods for vehicle routing problems // Networks. 1993. - № 23. - P. 661-673.
136. Thompson P.M. Cyclic transfer algorithms for the multivehicle routing and scheduling problems / P.M. Thompson, H.N. Psaraftis // Operations Research. — 1993. — № 41. P. 935-946.

Приложение 1

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014613291

**Эволюционно-иммунный алгоритм обучения многослойного
персептрона для решения задачи маршрутизации
информации**

Правообладатель: **Лавренков Юрий Николаевич (RU)**

Автор: **Лавренков Юрий Николаевич (RU)**

Заявка № **2014610552**

Дата поступления **28 января 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **21 марта 2014 г.**

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов



Приложение 2

УТВЕРЖДАЮ
Заместитель директора по
научно-исследовательской работе
ФГБОУ ВПО КФ МГТУ им. Н.Э. Баумана
Столяров А.А.
«30» мая 2014 г.



АКТ

об использовании результатов диссертационной работы
Лавренкова Юрия Николаевича «Исследование и разработка
комбинированных нейросетевых технологий для повышения эффективности
безопасной маршрутизации информации в сетях связи», представленной на
соискание учёной степени кандидата технических наук по специальности:
05.13.17 – Теоретические основы информатики.

Мы, нижеподписавшиеся, и.о. заведующего кафедрой «Компьютерные
системы и сети» к.т.н., доцент Родионов А.В., к.т.н., доцент кафедры
«Компьютерные системы и сети» Онуфриева Т.А., к.т.н., доцент кафедры
«Компьютерные системы и сети» Дерюгина Е.О. составили настоящий акт о
том, что результаты диссертационной работы Лавренкова Ю.Н.
«Исследование и разработка комбинированных нейросетевых технологий для
повышения эффективности безопасной маршрутизации информации в сетях
связи» были применены для повышения эффективности передачи
информации между вычислительными кластерами информационной системы
кафедры ЭИУ2 – КФ «Компьютерные системы и сети» ФГБОУ ВПО
Калужский филиал «Московского государственного технического
университета имени Н.Э. Баумана».

С помощью разработанного в диссертации комплекса алгоритмов для
обучения и настройки компонентов комитета нейросетевых экспертов была
повышена эффективность работы кафедральной информационной системы
«Контроль остаточных знаний», предназначеннной для автоматизированного
составления тестов и контроля знаний по курсам «Системы искусственного
интеллекта» и «Нейрокомпьютеры». Для обеспечения безопасности передачи
данных тестовых модулей применялась система шифрования, базирующаяся

на криптографическом алгоритме с открытым ключом RSA, реализованного с применением программно-аппаратной архитектуры параллельных вычислений CUDA.

Применение разработанного нейросетевого алгоритма позволило организовать безопасную передачу данных между отдельными элементами вычислительных кластеров без использования алгоритмов шифрования, что повысило общее быстродействие системы в 1,4 раза.

Организация моделирования вычислительной сети кафедры ЭИУ2 – КФ с помощью алгоритмов построения модели компьютерной сети и анализа характеристик среды передачи информации, разработанных для формирования векторов обучающей выборки ассоциативной машины, позволила выявить слабые места в вычислительных кластерах кафедральной компьютерной сети. Замена каналов связи, характеристики которых показали неудовлетворительный результат при функционировании нейросетевого маршрутизатора, позволила увеличить пропускную способность локальной сети.

Повышение эффективности выполнения маршрутизации информации на основе методов, разработанных в диссертации, подтверждается экспертной оценкой специалистов. Эффект от внедрения разработанного программного обеспечения заключается в повышении производительности локальной сети кафедры ЭИУ2 – КФ. Внедрение разработанного программного обеспечения планируется в других информационных системах КФ МГТУ им. Н.Э. Баумана.

К.т.н., доцент, и.о. заведующего кафедрой
«Компьютерные системы и сети»


/Родионов А.В./

К.т.н., доцент кафедры
«Компьютерные системы и сети»


/Онуфриева Т.А./

К.т.н., доцент кафедры
«Компьютерные системы и сети»


/Дерюгина Е.О./