


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМ. Н.П. ОГАРЁВА»**

**Институт электроники и светотехники**

**Кафедра инфокоммуникационных технологий и систем связи**

УТВЕРЖДАЮ

Зав. кафедрой  
канд. техн. наук, доц.

  
В. В. Никулин  
(подпись)

« 14. » 06 20 18 г.

**БАКАЛАВРСКАЯ РАБОТА**

**РАЗРАБОТКА ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ  
РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ**

Автор бакалаврской работы

  
(подпись)

14.06.18  
(дата)

К. О. Потапкин

Обозначение бакалаврской работы БР-02069964-11.03.02-17-18

Направление 11.03.02 инфокоммуникационные технологии и системы связи

Руководитель работы

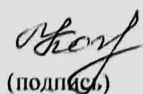
  
(подпись)

14.06.18  
(дата)

И. В. Маняев

ст. преподаватель

Нормоконтролер

  
(подпись)

14.06.18г.  
(дата)

Е. А. Кошечая

канд. культурологии, доц.

Саранск  
2018

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМ. Н.П. ОГАРЁВА»**

**Институт электроники и светотехники**

**Кафедра инфокоммуникационных технологий и систем связи**

УТВЕРЖДАЮ

Зав. кафедрой  
канд. техн. наук, доц.

  
В. В. Никулин  
(подпись)

« 28 » 11 20 17 г.

**ЗАДАНИЕ НА БАКАЛАВАРСКУЮ РАБОТУ**

Студент Потапкин Кирилл Олегович

1 Тема «Разработка искусственной нейронной сети для решения задачи маршрутизации»

Утверждена приказом № 9516-с от «24» ноября 2017 г.

2 Срок представления работы к защите 06.06.18

3 Исходные данные для научного проектирования: труды отечественных и зарубежных ученых и практиков, техническая литература по темам маршрутизация и нейронные сети

4 Задание на работу:

4.1 Анализ и описание способов организации потоков информации

4.2 Описание и применение искусственных нейронных сетей

4.3 Методы обучения нейронной сети

4.4 Разработка и анализ алгоритмов работы сети Хопфилда

4.5 Практическая реализация искусственной нейронной сети

5 Приложение А (обязательное) Функциональный код программы

Руководитель работы

28.11.17  
(дата)



(подпись)

И. В. Маняев

Задание принял к исполнению

28.11.17  
(дата)



(подпись)

К. О. Потапкин

## РЕФЕРАТ

Бакалаврская работа содержит 62 страницу, 17 рисунков, 20 формул, 1 листинг программы, 25 использованных источников, 1 приложение.

### ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ, СЕТЬ ХОПФИЛДА, ИСКУССТВЕННЫЕ ИНТЕЛЛЕКТ, МАРШРУТИЗАЦИЯ, ЗАДАЧА КОММИВАОЯЖОРА

Объектом исследования являются нейронные сети, моделирующие задачи оптимизации в частности нахождение кратчайшего пути.

Цель работы – проанализировать литературу по теме исследования, провести анализ нейронных сетей и методов их функционирования, реализовать сеть на основе рассмотренных алгоритмов.

В виду того, что с каждым годом количество устройств в информационных сетях растет, а также усложняется их структура, задача маршрутизации решается не в полной мере. В большинстве случаев это связано с маршрутизаторами, не справляющимися с поддержанием таблиц маршрутизации и выбором оптимальных маршрутов для передачи трафика. Поэтому возникает задача исследования существующих алгоритмов маршрутизации с целью улучшения их характеристик, или создания новых алгоритмов маршрутизации на основе таких технологий, как искусственные нейронные сети.

В процессе работы использовался опыт современных экспериментов в области искусственного интеллекта.

В результате исследования изучена методика решения задач, проанализирован метод работы нейронной сети Хопфилда, и реализована нейронная сеть на языке C++.

БР-02069964-11.03.02-17-18

Изм.	Лист	№ докум.	Подпись	Дата	Разработка искусственной нейронной сети для решения задачи маршрутизации	Лит.	Лист	Листов
Разраб.		К. О. Потапкин	<i>Потапкин</i>	14.06.18				4
Провер.		И. В. Маняев	<i>Маняев</i>	14.06.18				
Реценз								
Н. Контр.		Е. А. Кошева	<i>Кошева</i>	14.06.18				
Утверд.		В. В. Никулин	<i>Никулин</i>	14.06.18				

ИЭС, ИКТСС, 431

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
1 Описание способов организации потоков информации .....	8
1.1 Современные технологии для обеспечения маршрутизации .....	8
1.2 Дистанционно-векторные алгоритмы .....	10
1.3 Алгоритмы состояния каналов .....	13
2 Описание искусственных нейронных сетей.....	18
2.1 Преимущества нейронных сетей.....	18
2.2 Область применения нейронных сетей.....	23
2.3 Общие черты нейронных сетей .....	23
2.4 Типы функций активации.....	26
3 Методы обучения нейронной сети .....	29
3.1 Обучение с учителем .....	29
3.2 Обучение без учителя .....	31
4 Разработка и анализ алгоритмов работы сети Хопфилда .....	36
4.1 Нейронная сеть Хопфилда .....	36
4.2 Методы нахождения кратчайшего пути .....	38
5 Практическая реализация искусственной нейронной сети.....	44
5.1 Выбор среды проектирования.....	44
5.2 Разработка программного кода.....	46
5.3 Анализ полученных данных.....	48
ЗАКЛЮЧЕНИЕ .....	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	54
ПРИЛОЖЕНИЕ А (обязательное) Функциональный код программы .....	57

## ВВЕДЕНИЕ

Становление сложных электронно-вычислительных комплексов и систем базируется на передаче данных от отправителя к получателю. Задачу подбора оптимального пути, который обеспечивает прохождение информации по маршруту, отвечающему определённым критериям, решают методы маршрутизации.

Маршрутизация – это процесс определения пути следования информации в сетях связи. При этом, как правило, на пути встречается, по крайней мере, один узел. Маршрутизация включает в себя две главные составляющие: определение оптимальных путей маршрутизации и передача информационных сообщений. Определение маршрута характеризуется сложностью процесса и основывается на различных показателях или их комбинация. Когда информация передается по пути, заранее не заданном, и определяется на каждом узле по мере продвижения информации, т.е. процесс маршрутизации происходит в динамическом режиме, сложность расчёта маршрута возрастает.

В практических задачах появляются такие требования, что передачу информации необходимо осуществить открытым виде, т.е. без использования средств шифрования. В этом случае необходимо построение маршрутов транспортировки данных по каналам связи, обладающих требуемой степенью надёжности и защищённости от вмешательства злоумышленников, в том числе от непосредственного физического подключения к среде передачи. При маршрутизации на основе установленных требований нужно выполнять оценку не только характеристик, которые обеспечивают быструю передачу информации получателю, но и в процессе поиска оптимального пути учитывать критерии безопасности среды передачи.

Имеющиеся методы маршрутизации требуют наличия информации о полной структуре сети, в которой будет производиться передача данных. Если топология сети подвергается частым изменениям, появление и удаление новых

соединений и узлов, изменения в среде передачи, то маршрутизирующие алгоритмы утрачивают способность поддерживать эффективный обмен информации в сети.

Для того чтобы телекоммуникационная сеть поддерживала свои функции по обеспечению передачи информационных сообщений, требуется применение современных методов, которые способны решать задачи при неполных, неверных, или противоречивых входных данных. Такой способностью обладают вычислительные методы на основе нейросетевых алгоритмов.

Использование интеллектуальных разработок позволит выполнять передачу данных в распределённых сетях связи даже при условии частичного вывода из строя оборудования маршрутизации или линии связи, а также в случаях сбоя, вызванных действиями третьих лиц. Таким образом, задача разработки искусственной нейронной сети для решения задачи маршрутизации является актуальной и практически значимой.

# 1 Описание способов организации потоков информации

## 1.1 Современные технологии для обеспечения маршрутизации

Телекоммуникационные сети применяются для того, чтобы транспортировать информацию от отправителей к получателям. Обычно информация пересылается от одного узла к другому до момента, пока не достигнет конечного пункта, на который должны быть переданы данные. В большинстве случаев существует несколько путей передачи информации (рисунок 1.1). Рисунок демонстрирует возможность передачи информации от узла *A* к узлу *B* по нескольким маршрутам.

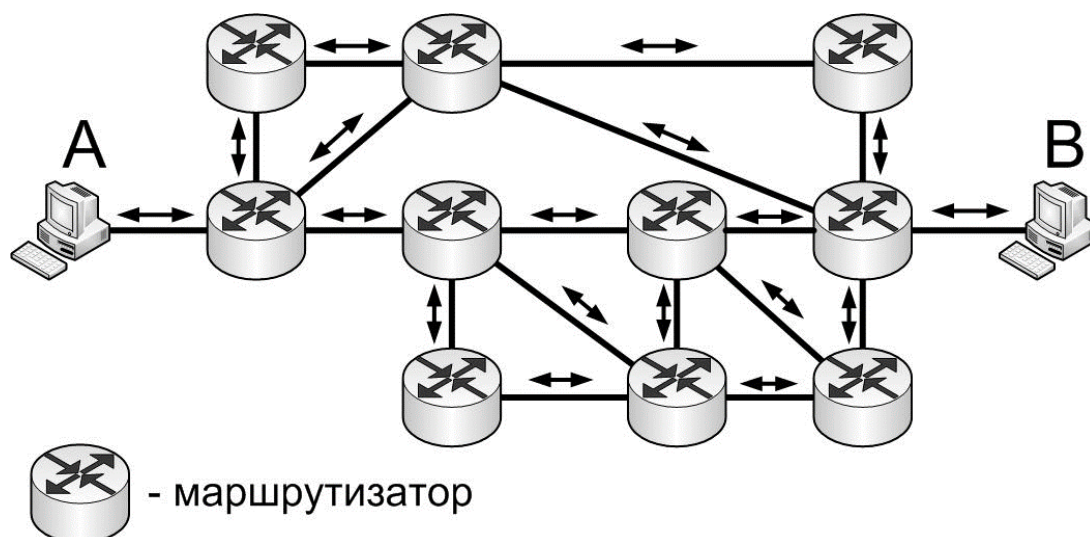


Рисунок 1.1 – Проблема выбора оптимального маршрута

Чтобы выбрать оптимальный путь, соответствующий определенным критериям, применяются алгоритмы маршрутизации.

В рамках маршрутизации осуществляются следующие параллельные процессы:

- формируется таблица маршрутизации – информационная структура, включающая данные о том, какой пункт информация должна пройти следующим, чтобы маршрут доставки ее на узел назначения был оптимальным;







рассылки актуальной является и вопрос о выборе маршрута с необходимой степенью безопасности, обеспечивающего надежную доставку информации.

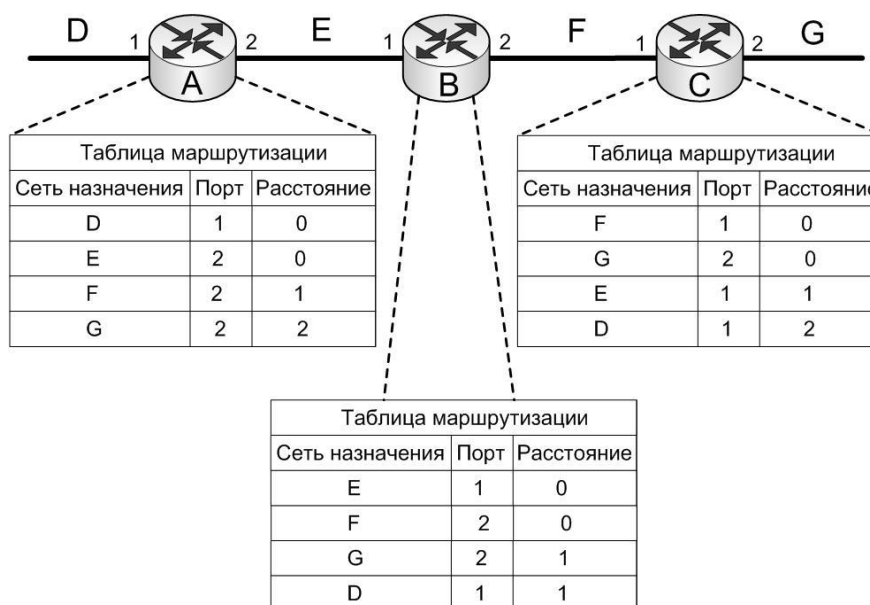


Рисунок 1.2 – Пример таблицы маршрутизации ДВА

Начальным этапом развития протоколов широковещательной рассылки стало создание алгоритма, в соответствии с которым маршрутизатор пересылает информацию на каждый из интерфейсов, за исключением входного. Подобная стратегия обеспечивает генерацию значительного объема трафика в сети, не несущего полезной нагрузки. Модификации алгоритма дают возможность распространять трафик от источника до получателя таким образом, чтобы продвижение пакетов происходило исключительно по путям, которые наиболее оптимально соединяют источник и каждого из получателей.

Рисунок 1.3 отображает ситуацию, когда маршруты группового трафика от источника до получателей, которым они не предназначаются, исключены. Таким образом, существует необходимость в формировании дерева, вершина которого будет находиться в источнике информации, запланированной к передаче. Дерево должно обеспечивать соединение всех маршрутизаторов с подключенными к ним локальными сетями, в которых содержатся получатели данной группы, по

наиболее оптимальным путем [3]. Применяя нейросетевые технологии для модификации алгоритмов данного типа, можно добиться построения маршрута движения групповой информации по пути, который будет наиболее безопасным, и при этом обеспечить учет изменений топологии сети.

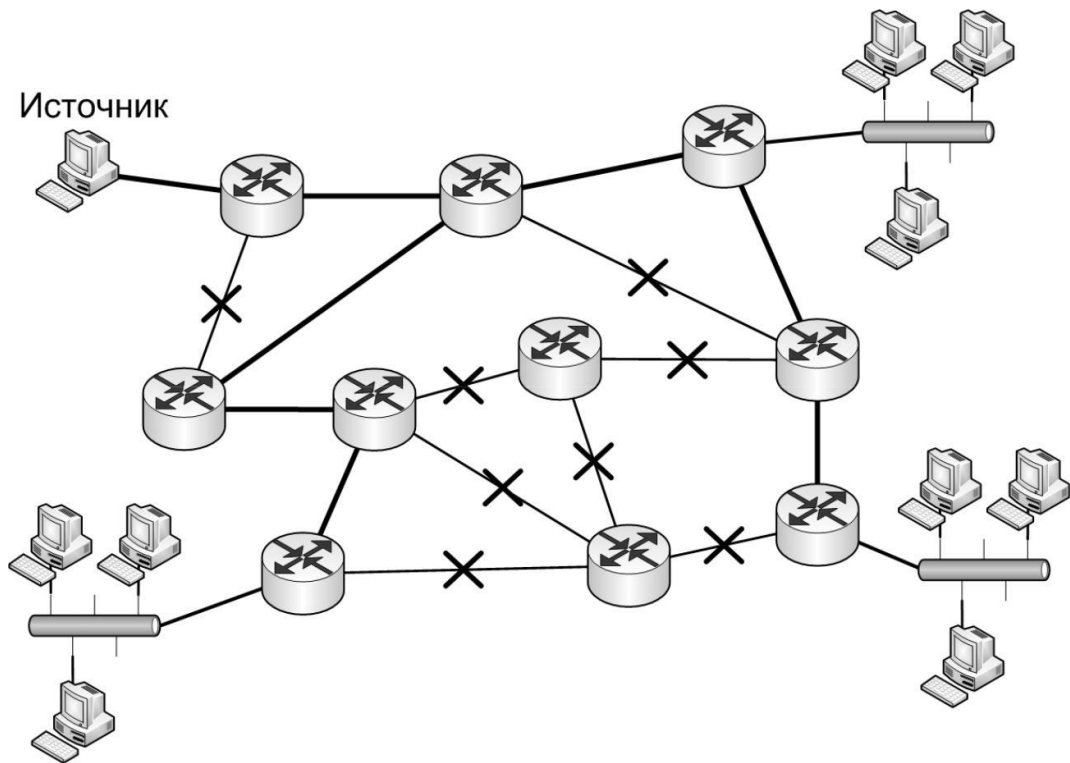


Рисунок 1.3 – Управление групповой передачей информации

Недостатки представленного алгоритма могут быть представлены следующим образом.

1) Алгоритм стабильно функционирует в случае, если масштабы сети являются сравнительно небольшими. Если сеть имеет значительный масштаб, процесс интенсивного обмена таблицами маршрутизации вызывает дополнительную нагрузку на линии связи.

2) Обработка изменений конфигурации сети подобными алгоритмами маршрутизации не всегда может быть корректной, в силу того, что у маршрутизаторов нет сведений о том, какова точная топология сети.

Лит	Изм.	№ докум.	Подп.	Дата

3) Адаптация протоколов, которые используют ДВА маршрутизации (к примеру, RIP – Routing Information Protocol) к утрате маршрута вызывает определенные затруднения, поскольку они передают информацию, которая необходима, чтобы пополнять таблицы маршрутизации.

4) Информация может заикливаться, т.е. информационные группы могут постоянно передвигаться между маршрутизаторами.

Возможна вероятность того, что появятся ложные маршруты, возникающие в случае, когда используется информация о маршрутах, которые не существуют.

### 1.3 Алгоритмы состояния каналов

За счет алгоритмов состояния каналов (АСК) все маршрутизаторы обеспечиваются информацией, которая является достаточной для того, чтобы построить точный граф сети. Если маршрутизаторы работают на базе АСК, они в состоянии поддерживать сложную базу с информацией о том, какова топология соединений в сети. Все маршрутизаторы функционируют на основе одного графа сети. В силу подобной особенности маршрутизация становится устойчивой к изменению конфигурации. В случае, когда осуществляется ДВА-маршрутизация информация об удаленных сетях и маршрутизаторах отсутствует.

Процесс формирования таблицы маршрутизатора является двухэтапным.

1) На первом этапе формируется база данных, включающая сведения о состоянии связей в сети. Топология имеет вид графа. В качестве вершин выступают маршрутизаторы, ребра графа представлены связями маршрутизаторов. Маршрутизаторы осуществляют обмен с соседними маршрутизаторами информацией о графе сети, которая имеется у каждого из маршрутизаторов на данный момент. Информация в процессе передачи маршрутизаторами не модифицируется. Каждый из маршрутизаторов в итоге имеет идентичную информацию о том, какова структура сети. Производится

упорядочение информации, и в итоге формируется логическая топология, которая представляет собой дерево связей. Корень дерева представлен в виде текущего маршрутизатора, ветви – в виде возможных маршрутов к каждой из подсетей. При изменении состояния происходит повторение процесса формирования графа.

2) В рамках второго этапа определяются оптимальные маршруты. Генерируются таблицы маршрутизации. После того, как будет построено дерево связей, начинается трудоемкий процесс поиска оптимального пути в рамках графа. В протоколах маршрутизации, в основе которых – АСК (к примеру, OSPF – Open Shortest Path First), при поиске указанного пути применяется итеративный алгоритм Дейкстры. Каждым маршрутизатором производится поиск оптимальных путей от собственного интерфейса до каждой из известных подсетей. По каждому из найденных путей осуществляется запоминание только первого шага, который и включается в таблицу маршрутизации.

Рисунок 1.4 отображает сеть, включающую шесть маршрутизаторов ( $R_1 - R_6$ ) и подсетей ( $N_1 - N_6$ ). Проводится оценка состояния каналов, с приписыванием каждому каналу меры стоимости. Рисунок 1.5 отражает дерево наиболее коротких путей применительно к маршрутизатору  $R_3$ . Соответственно, каждым из маршрутизаторов может осуществляться отслеживание альтернативных путей и осуществляться выбор пути до каждой из конечных точек, который будет оптимальным.

На основе АСК осуществляется функционирование и протокола MOSF (Multicast extensions to OSPF), который применяется при групповом вещании. Маршрутизаторами производится добавление к информации о состоянии каналов передачи данных о том, в каких группах состоят различные узлы сети. Это позволяет маршрутизаторам осуществлять формирование не только общего графа сети, но и получать информацию о том, каков состав групп для рассылки сообщений. На основе этих данных маршрутизирующими алгоритмами производится определение дерева наиболее коротких путей применительно к

каждой группе. Это дает возможность доставки информационных сообщений по наиболее коротким путям от источника до подсетей, в которых имеются члены группы – конечные получатели информации.

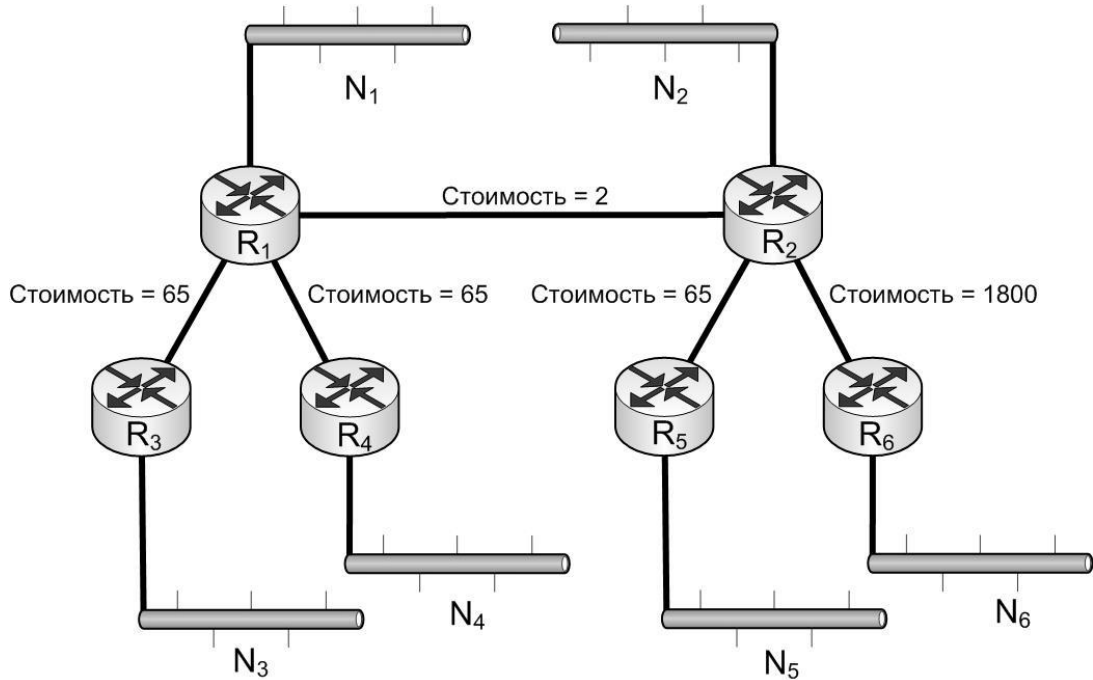


Рисунок 1.4 – Пример конфигурации сети и состояния каналов

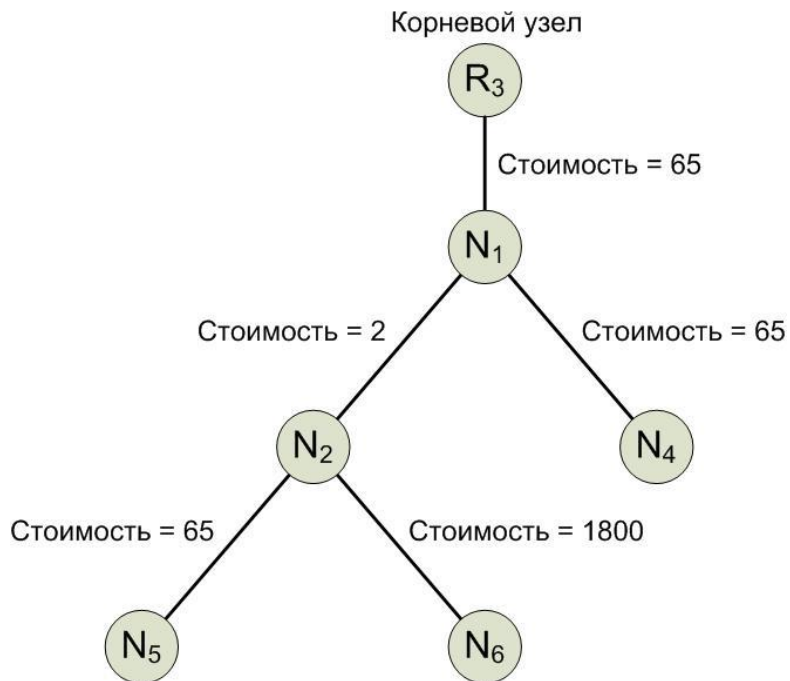


Рисунок 1.5 – Кратчайшие маршруты от узла R3

Лит	Изм.	№ докум.	Подп.	Дата

На основе АСК осуществляется функционирование и протокола MOSF (Multicast extensions to OSPF), который применяется при групповом вещании. Маршрутизаторами производится добавление к информации о состоянии каналов передачи данных о том, в каких группах состоят различные узлы сети. Это позволяет маршрутизаторам осуществлять формирование не только общего графа сети, но и получать информацию о том, каков состав групп для рассылки сообщений. На основе этих данных маршрутизирующими алгоритмами производится определение дерева наиболее коротких путей применительно к каждой группе. Это дает возможность доставки информационных сообщений по наиболее коротким путям от источника до подсетей, в которых имеются члены группы – конечные получатели информации.

Затруднения, возникающие в случае, когда используется алгоритм маршрутизации, основанный на оценке состояния каналов передачи, следующие:

1) для реализации алгоритма необходим значительный объем памяти. Обработка больших потоков информации, поддержка логического дерева и таблиц маршрутизации также вызывает необходимость в существенных вычислительных ресурсах;

2) следствием динамического определения структуры сети является генерация значительных объемов трафика, который требуется, чтобы обеспечивать обмен маршрутизаторов информацией друг с другом;

3) если в сетевой структуре имеется сложная топология, с множественными связями, то при частом изменении характеристик каналов связи, возникновении новых соединений, отключении части каналов связи возможен полный отказ алгоритма.

Одна из наиболее важных проблем, выявленная в процессе исследования недостатков представленных алгоритмов, состоит в выполнении маршрутизации, когда информация является недостаточно полной. Маршрутизаторы получают информацию от отдельных элементов сети, однако



таблицы маршрутизации могут содержать неполную маршрутную информацию. Если маршрутизирующая система начинает выполнять свои функции с нарушениями, отдельные каналы связи или сами маршрутизаторы могут отказать. Если у маршрутизаторов отсутствует информация, которая требуется, чтобы выбрать оптимальный путь до узла назначения, пересылка данных осуществляется в соответствии со стандартными, заранее определенными маршрутами.

Маршрутизирующей системой данные могут быть отправлены за пределы локальной автономной системы. Вследствие этого указанные данные могут вернуться обратно, и трафик может заиклиться. При отказе каналов связи, за счет которых обеспечиваются статические резервные маршруты передачи информации, происходит полная остановка функционирования сети. Использование аппроксимирующих способностей нейронных сетей обеспечивает поддержание функционирования сети в моменты, когда стандартные протоколы маршрутизации работают нестабильно.

## 2 Описание искусственных нейронных сетей

Нейронные сети представляют собой гигантские распределенные параллельные процессоры, состоящие из базовых устройств обработки информации для сбора экспериментальных знаний и обеспечения их дальнейшего использования. Нейронная сеть похожа на человеческий мозг по двум соображениям:

- знания в нейронную сеть принимаются из окружающей среды и продолжают использоваться в процессе обучения;
- связи между нейронами, называемые синаптическими весами, используются для накопления знаний.

### 2.1 Преимущества нейронных сетей

Совершенно очевидно, что нейронные сети наиболее перспективны и мощны, во-первых, из-за параллелизма в обработке информации, во-вторых, из-за способности к самообучению, т.е. создавать обобщения. Обобщение термин, означающий способность получать достоверные результаты на основе данных, которые были получены в учебном процессе. Благодаря этим свойствам нейронным сетям не составляет труда принимать решения по сложным (масштабным) задачам, считающимся трудноразрешимыми в настоящее время. Однако, как показывает практика, нейронные сети не способны автономно обеспечить нас готовыми решениями. Для этого необходима интеграция в сложные системные комплексы. В частности, сложные задачи можно поделить на некоторую последовательность сравнительно простых, решаемость которых обеспечивается нейронными сетями. Для создания компьютерной архитектуры, которая будет способна имитировать человеческий мозг (если такое окажется возможным вообще), придется пройти долгий и



акцентирования того, что изначально не существует никакой predetermined статистической модели входных данных [3]. Например, рассмотрим задачу определения изображений. В ней требуется сопоставить входной сигнал, указывающий на материальный объект, или событие, с какой-либо заданной категорией (классом). В таком подходе к этой проблеме необходимо оценить рамки принятия решений в множестве входного сигнала на основе серии выборок. При этом никакая модель распределения вероятностей не используется. Аналогичный подход применяется и в парадигме обучения с учителем. Это вновь подчеркивает параллельную взаимосвязь между отражением входных сигналов во выходные, которые осуществляются нейронной сетью, и непараметрическим статистическим обучением.

Способностью нейронные сети подстраивать свои синаптические веса к изменениям в окружающей среде считается адаптивностью. В частности, обученные нейронные сети, которые действуют в определенной среде, могут быть легко использованы после переобучения в работе, когда условия параметров среды находятся в незначительные колебания. Также, при работе в нестационарной среде создаются нейронные сети, которые изменяют синаптические веса в реальном времени. Архитектура нейронных сетей, естественная для определения изображений, обработки сигналов и задач управления, может быть скомбинирована с их способностью к адаптивному, что приводит к созданию моделей адаптивного определения изображений, адаптивной обработки сигналов и адаптивного управления. Установлено, что чем выше способность системы адаптироваться, тем более стабильной будет ее работа в нестационарной среде. В тоже время хотелось бы отметить, что адаптивность не всегда ведет к стабильности; иногда она приводит к результатам совершенно противоречивым. Например, адаптивная система, параметры которой очень быстро изменяются во времени, может также очень быстро

реагировать на незнакомые раздражители, что вызывает потерю производительности. Чтобы в полной мере воспользоваться всеми преимуществами адаптивности, основные параметры системы должны оставаться достаточно стабильными, для того чтобы внешние помехи не учитывались в системе, и достаточно гибкими, для того чтобы при значительных изменениях в окружающей среде можно было обеспечить ответ. Эта задача обычно называется дилеммой стабильности-пластичности [4].

В концепции задачи определения изображений можно развивать нейронную сеть, которая определяет информацию не только для установления конкретного типа, но и для повышения надежности принимаемого решения. В дальнейшем эта информация может быть использована для устранения подозрительных решений, что может повысить эффективность нейронной сети.

В соответствии со структурой нейронной сети, информация в ней представляется с помощью ее состояния активации. Каждый нейрон сети потенциально может быть затронут влиянию всех остальных ее нейронов. В результате, существование нейронных сетей напрямую связано с контекстной информацией.

Нейронные сети, модулируемые блоками электроники, потенциально являются отказоустойчивыми. Это означает, что при стечении неблагоприятных условий, их производительность падает незначительно. Например, если поврежден какой-то нейрон или его соединения, извлечение сохранённой информации затрудняется. Однако, учитывая то, что нейронная сеть имеет распределенный характер хранения информации, она может гарантировать, что

только серьезные повреждения структуры нейронной сети могут оказывать существенное влияние на ее работоспособность. Поэтому снижение качества работы нейронной сети происходит медленно. Небольшое повреждение структуры никогда не приведет к катастрофическим последствиям. Это очевидное преимущество нейронных вычислений, однако его часто не принимают в расчет. Чтобы алгоритмы обучения гарантировали отказоустойчивость работы нейронной сети, в них нужно закладывать соответствующие поправки.

Параллельная архитектура нейронных сетей значительно ускоряет решение некоторых задач и обеспечивает масштабируемость нейронных сетей в рамках технологии VLSI (very large scale integrated). Одним из преимуществ технологий VLSI является возможность представления достаточно сложного поведения с помощью иерархической структуры.

Нейронные сети представляют из себя универсальный способ обработки информации. Это означает, что одно и то же спроектированное решение нейронной сети способно работать и во многих других предметных областях. Это свойство проявляется несколькими способами.

– Нейроны являются основной составляющей частью в той или иной форме для любой нейронной сети.

– Благодаря этому сходству возможно применять одни и те же теории и алгоритмы обучения для построения различных приложений на основе нейронных сетей.

– Блочные сети могут быть спроектированы с использованием интеграции целых модулей.

Структура нейронных сетей строится по аналогии с человеческим мозгом, являющимся ярким примером, доказывающим то, что параллельные отказоустойчивые вычисления не только физически реализуемы, но также обеспечит быстрые и мощные инструменты решения задач.

## **2.2 Область применения нейронных сетей**

Выполняемые сетью функции можно разделить на несколько основных групп: аппроксимации и интерполяции, распознавания и классификации образов, сжатия данных, прогнозирования, идентификации, управления, ассоциации.

Традиционно нейронные сети применяются для распознавания образов (в широком смысле этого слова, т. е., например, как для обнаружения лиц на фотографиях, так и для распознавания текстов) и для сжатия информации. В последнее время, однако, нейронные сети все чаще применяются в задачах прогнозирования и управления. На фондовых рынках успешно действуют боты, написанные с применением нейронных сетей, принимающие решения о купле и продаже на основании значений индексов.

## **2.3 Общие черты нейронных сетей**

Широкий круг задач, решаемых нейронными сетями, не позволяет создавать мощные универсальные сети из-за ограниченности вычислительных ресурсов. И хотя разработчики вынуждены создавать специализированные нейронные сети, функционирующие по различным алгоритмам, эти отдельные типы сетей обладают несколькими общими чертами.

Во-первых, общим для нейронных сетей является принцип параллельной обработки сигналов, который достигается путем объединения большого числа

нейронов в слое. Нейроны различных слоев определенным образом соединены, а также, в некоторых конфигурациях, соединены и нейроны одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

Во-вторых, основу каждой нейронной сети составляют относительно простые однотипные ячейки, имитирующие работу нервных клеток (нейронов) человеческого организма, которые в некоторых случаях объединяют в слои:

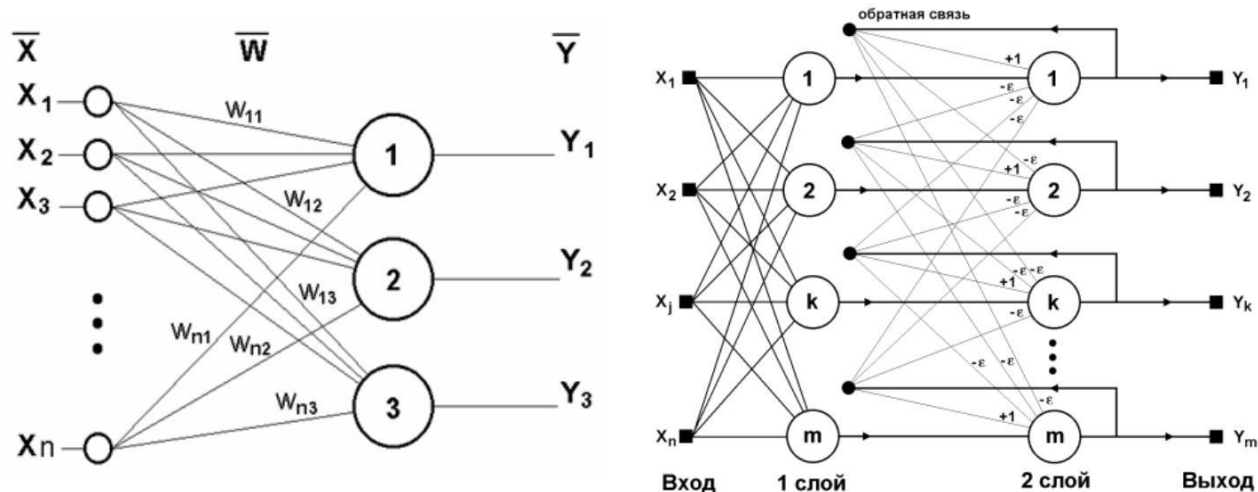


Рисунок 2.1 – Однослойная и двухслойная нейронные сети

Так как принцип работы искусственного нейрона основан на механизме функционирования нейронов настоящих, рассмотрим вначале их строение.

Нервная клетка (нейрон) имеет тело, называемое сомой, со стандартным набором органелл и ядром. Можно выделить два типа отростков:

- 1) дендриты, многочисленные тонкие ветвящиеся отростки, играющие ключевую роль во взаимодействии нейронов друг с другом;
- 2) аксон, длинный и толстый отросток, ветвящийся на конце.

Входные сигналы поступают в клетку через синапсы, а выходной сигнал отводится аксоном через его многочисленные нервные окончания, называемые коллатералиями. Коллатерали взаимодействуют с дендритами и сомами других клеток, создавая новые синапсы.





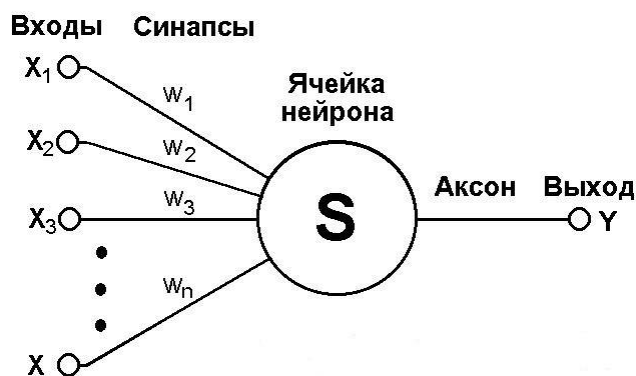


Рисунок 2.3 – Искусственный нейрон

Текущее состояние  $i$ -ого нейрона будем определять, как взвешенную сумму его входов:

(1)

а выход — как функцию его состояния .

Функция  $f(s)$  называется активационной, и, вообще говоря, может иметь различный вид.

## 2.4 Типы функций активации

Активационные функции, представленные в формулах как  $f(s)$ , определяют выход сигнала нейрона в зависимости от производного локального местоположения  $s$ . В основном существуют три типа функций активации.

1) Функция единичного скачка – функция Хевисайда – пороговая функция, или пороговая функция. Функция показана на рисунке 2.4(а), и описывается следующим выражением:

(2)

2) Кусочно-линейная функция, представленная на рисунке 2.4(б), описывается следующим образом:


$$(3)$$

Эту функцию активации можно рассматривать как аппроксимацию нелинейного усилителя. Следующие два варианта можно считать особой формой кусочно-линейной функции:

– если линейная область оператора не достигает порога насыщения, он превращается в линейный сумматор;

– если коэффициент усиления линейной области принять бесконечно большим, то кусочно-линейная функция вырождается в пороговую [3].

3) Сигмоидальная функция. График этой функции похож на букву S. Она, вероятно, является наиболее распространенной функцией, используемой для создания искусственных нейронных сетей. Это быстро растущая функция, которая поддерживает баланс между линейным и нелинейным поведением. Примером сигмоидальной функции может служить логистическая функция, задаваемая следующим выражением:


$$(4)$$

где  $a$  параметр наклона сигмоидальной функции.

Изменяя параметр  $a$ , можно построить функции с различной крутизной (рисунок 2.4(в)). Первый график соответствует величине параметра, равной,  $a/2$ . в пределе, когда параметр наклона достигает бесконечности, сигмоидальная

функция вырождается в пороговую. Если пороговая функция может принимать только значения 0 и 1, то сигмоидальная функция принимает бесконечное множество значений в этом же диапазоне. При этом следует заметить, что сигмоидальная функция является дифференцируемой, в то время как пороговая – нет.

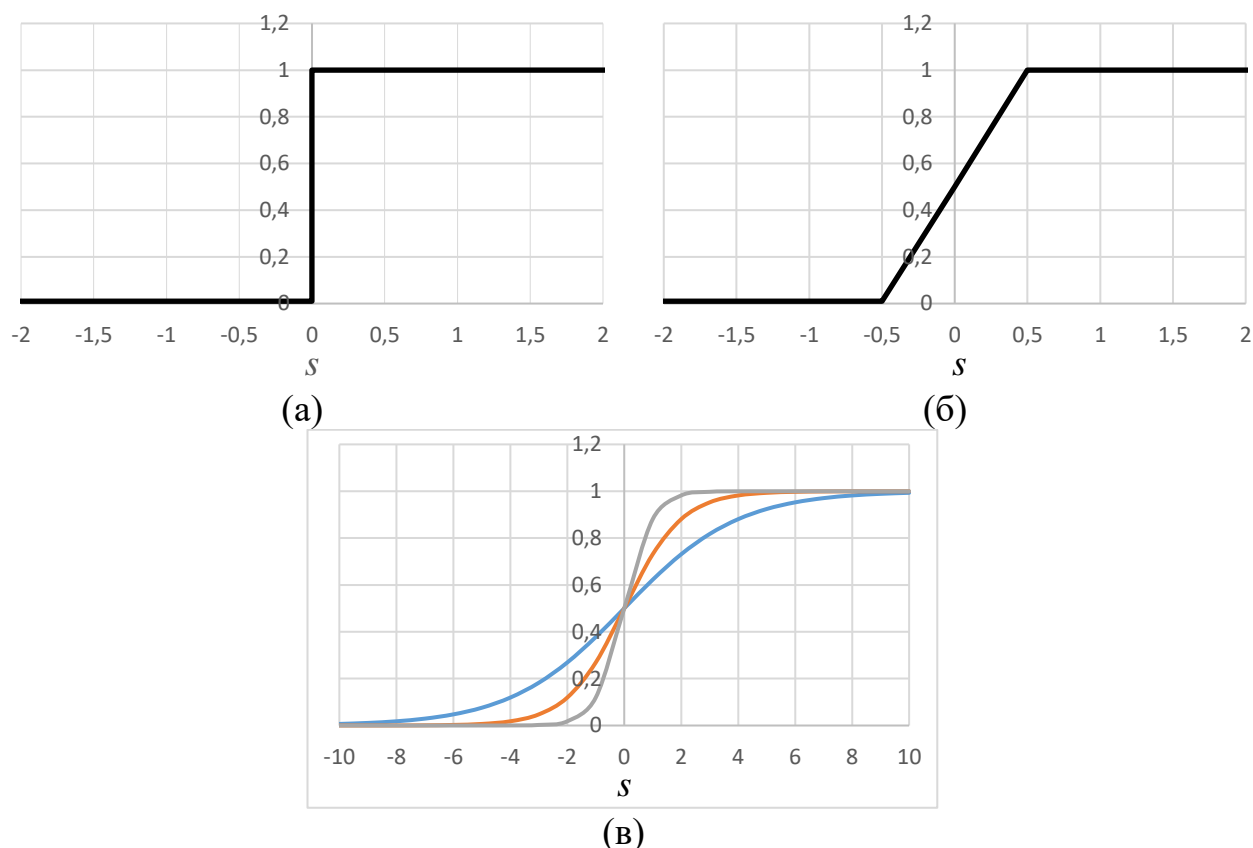


Рисунок 2.4 – Типы функции акций активации

а – функция единичного скачка, б – кусочно-линейная функция,  
 в – сигмоидальные функции

### 3 Методы обучения нейронной сети

Формирование весовых коэффициентов синапсов (ВКС) – главная задача по созданию нейронной сети (НС). От правильности решения напрямую будет зависеть вся работоспособность системы. Процесс поиска и структурирования ВКС условно называют обучением нейронной сети.

Способы обучения можно объединить в две общие группы:

1) Обучение с учителем. При наличии тестового массива входящих данных и установленного соответствия на подачу выходного сигнала можно провести обучение с учителем. Нейронной сети предоставляют уже готовую информацию, которая подчиняется строгим правилам и алгоритмам. За счет этого ей удастся сформировать последовательности и уже самостоятельно подстраивать веса синаптических связей. Сеть, на основе предоставленных примеров, устанавливает соответствие сигналов между потоками входных и выходных сигналов.

2) Обучение без учителя. Для поиска весовых коэффициентов синапсов здесь используются алгоритмы, которые зависят только от значений входных данных.

#### 3.1 Обучение с учителем

Для большего понимания изучим парадигмы обучения НС. На рисунке 3.1 показана блочная диаграмма, иллюстрирующая эту форму обучения. При использовании методики с учителем общая концепция строится на передачи знаний из окружающей среды, сформированных в пару вход-выход. Для нейронной сети в процессе обучения сама окружающая среда остается неизвестной. Если учителю и ученику в виде НС поступает обучающий вектор из внешней среды – учитель на основе своих знаний может передать и сформировать для сети необходимый отклик на поступающий сигнал. Это и

является функциональным решением задачи. В процессе обучения параметры нейронной сети могут корректироваться на основе входящих данных и сигналов ошибок (разности между требуемым результатом и текущим выходным откликом).

Корректировка производится поэтапно, чтобы сформировать в НС симуляцию поведения учителя. Это является неотъемлемым процессом формирования самостоятельной работы нейронной сети. Только при понимании реакции на сигналы ошибок удастся передать знания от учителя в полном объеме. В дальнейшем нейронную сеть можно переводить на самостоятельную работу с поступающими сигналами от окружающей среды. Здесь важно понимание того, что при работе с учителем система является замкнутой. Постоянная обратная связь исключает воздействие окружающей среды.

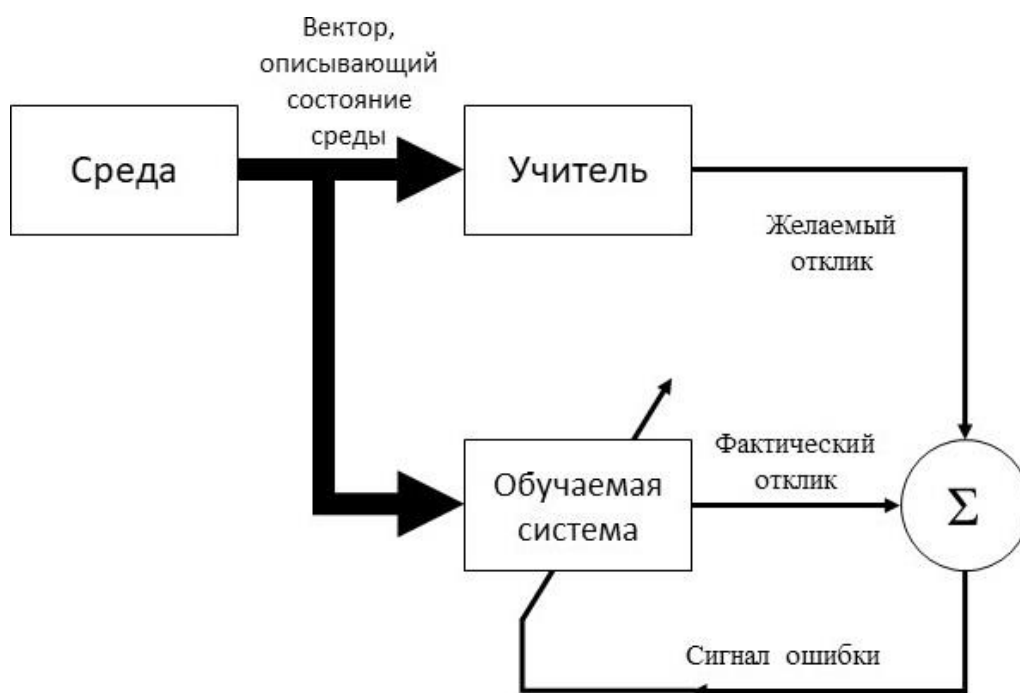


Рисунок 3.1 – Блочная диаграмма обучения с учителем

Производительность нейронной сети оценивается по терминам среднеквадратической ошибки или сумме квадратов ошибок. На основе значений можно выстроить многомерную поверхность ошибки на координатах.



блочная диаграмма одной из форм системы обучения с подкреплением, который включает блок критики, преобразующий первичный сигнал подкрепления, полученный из внешней среды, в сигнал более высокого качества, называемый эвристическим сигналом подкрепления [3].

Все эти сигналы являются скалярными. Эта система подразумевает обучение с задержанным подкреплением. Это означает, что система получает извне последовательность возбуждающих сигналов (то есть, векторов состояния), приводящих к генерации эвристического сигнала подкрепления. Цель обучения – минимизация функции цены перехода, определенной как математическое ожидание кумулятивной цены действий, которые предприняты за нескольких шагов, а не только текущей цены. Может стать, что некоторые совершённые раньше в этой последовательности действия были решающими в формировании общего поведения системы целиком. Функция обучаемой машины, которая составляет второй элемент системы, распознает такие действия и формирует на основе них сигнал обратной связи, который направляется ко внешней среде.

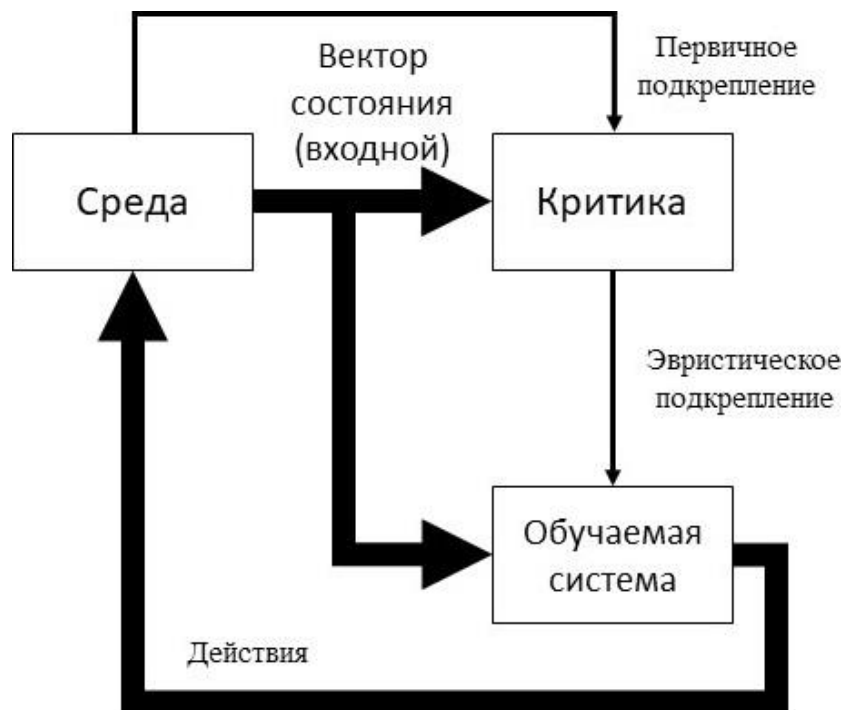


Рисунок 3.2 – Блочная диаграмма обучения с подкреплением



Реализация на практике обучения с отложенным подкреплением осложняется двумя причинами.

– Нет учителя, который формирует желаемый отклик на каждом этапе обучения.

– Из-за задержки формирования первичного сигнала подкрепления необходимо решить временную задачу присваивания коэффициентов доверия. Это означает, что обучаемая машина должна уметь присваивать коэффициенты недоверия и доверия действиям, которые выполняются на всех этапах, которые приводят к конечному результату. Первичный сигнал подкрепления формируется лишь на основании конечного результата.

Вопреки этим сложностям, системы обучения с отложенным подкреплением очень привлекательны. Они являются базисом систем, которые взаимодействуют с внешней средой, формируя тем самым способность самостоятельного разрешения появляющихся задач на основании только собственного результата взаимодействия со средой. Обучение с подкреплением тесно связано с динамическим программированием – методологией, созданной Беллманом в 1957 году в контексте теории оптимально управления [118]. Динамическое программирование реализует математический формализм последовательного принятия решений.

### **3.2.2 Обучение без учителя**

При использовании методики без учителя отсутствует контроль за процессами настройки весовых коэффициентов. Здесь нет конкретных примеров на действие любых входящих данных при обучении НС. Процесс обработки массивов информации строится на основе предоставленных параметров качества для решаемой задачи. Все статистические закономерности, которые поступают из внешней среды, формируются нейронной сетью и кодируются по

установленным признакам входящих сигналов с последующим созданием нужных классов.



Рисунок 3.3 – Блочная диаграмма обучения без учителя

Для такой методики обучения эффективным приемом становится правило конкурентного обучения. Как пример, нейронную сеть создают из двух слоев входных и выходных сигналов. На первый слой отправляются данные. На второй слой передается информация, а нейроны в составе сети конкурируют между собой на получение права отклика. Но конкуренция возможно только в том случае, если входной сигнал удовлетворяет всем признакам для отбора.

Говоря об обучении без учителя, стоит выделить алгоритм обучения *по Хеббу*. Он состоит в следующем.

1) На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.

2) На входы сети подается входной образ, и сигналы возбуждения распространяются по всем слоям обычным образом: для каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется активационная (передаточная) функция нейрона, в результате чего получается его выходное значение  $y_n$ , где  $n$  – число нейронов в слое  $p$ .

3) На основании полученных выходных значений нейронов по формулам Хебба производится изменение весовых коэффициентов.

4) Цикл повторяется с шага 2, пока выходные значения сети не застабилизуются с заданной точностью. Применение этого способа определения завершения обучения обусловлено тем, что подстраиваемые значения синапсов фактически не ограничены.

Обучение *по Кохонену* проводится так же, с тем лишь отличием, что на шаге 3 из всего слоя выбирается нейрон, значения синапсов которого максимально подходят на входной образ, и подстройка весов по формуле Кохонена проводится только для него.

Особый интерес представляют сети Хопфилда и Хэмминга, поскольку по приведенной классификации они не принадлежат, строго говоря, ни к одному из классов.

## 4 Разработка и анализ алгоритмов работы сети Хопфилда

### 4.1 Нейронная сеть Хопфилда

Сеть Хопфилда состоит из одного слоя нейронов. Число нейронов равно числу входов и выходов сети. Каждый нейрон связан синапсами со всеми остальными нейронами, а также с одним из входов. Структурная схема сети приведена на рисунке 4.1

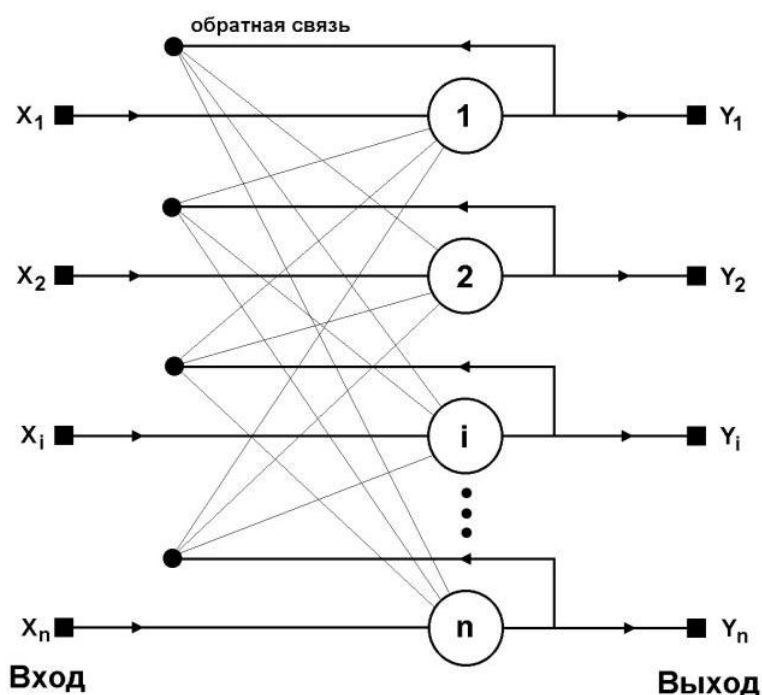


Рисунок 4.1 – Структурная схема сети Хопфилда

Сеть Хопфилда позволяет решать следующую задачу: пусть имеется набор двоичных сигналов, которые считаются образцовыми. Сеть должна уметь по неидеальному (зашумленному) сигналу распознать образцовый, либо же сообщить о том, что входные данные не соответствуют ни одному из образцов.

Представим такой сигнал в виде вектора

, где  $n$  — число нейронов сети (отсюда видно, что размерность входного и выходного сигнала должна быть равна числу нейронов сети).

Обозначим вектор, соответствующий  $k$ -ому образцу, за  $X_k$ , а его элемент — за  $x_{kj}$ ,

. Если сеть сможет распознать зашумленный сигнал, выходной вектор  $Y$  будет равен соответствующему образцу, в противном случае, выходной вектор не совпадет ни с одним из образцов (скорее всего, на выходе окажется сигнал, составленный из различных кусков образцовых).

На стадии инициализации сети весовые коэффициенты синапсов устанавливаются следующим образом:

(5)

где  $i$  и  $j$  — индексы, соответственно, предсинаптического и постсинаптического нейронов;

, —  $i$ -ый и  $j$ -ый элементы вектора  $k$ -ого образца.

Нулевая проводимость синапса при  $i = j$  обозначает на самом деле отсутствие синапса, ведущего на тот же нейрон, с которого он берет начало.

Далее сеть функционирует по следующему алгоритму:

- 1) на входы сети подается неизвестный сигнал;
- 2) рассчитывается новое состояние нейронов:

(6)

и новые значения аксонов:

(7)

где  $f$  — активационная функция в виде скачка (рисунок 2.4(а), справа);

3) если значения аксонов на за последнюю итерацию изменились — переход к пункту 2, иначе (если выходы застabilizировались) — конец.

При этом вектором выходных значений является образец, объективно сочетающийся с входными данными.

## 4.2 Методы нахождения кратчайшего пути

Обеспечение эффективного применения сетевых ресурсов считается одной из основных задач маршрутизации. При этом основой математического решения задачи маршрутизации сетевого трафика в имеющихся протоколах (EIGRP, RIP, BGP, OSPF, IS-IS и пр.) является модель нахождения кратчайшего пути на графах. Ряд предложенных эвристических решений приуступающий в некоторых по сфере управления сетевыми ресурсами в телекоммуникационной системе (ТКС), обладаем общим недостатком, которое выражается невысоком качестве получаемых решений. Это обстоятельство создает проблемы при реализации для работы в реальном времени для протоколов, а также необоснованное использование канальных и кэшируемых ресурсов [6].

Разработка и применение новых методов решения задач администрирования ТКС сетевых ресурсов и проблемы маршрутизации является выходом из сформировавшейся ситуации. Например, существующие протоколы обеспечивают периодическую или аperiodический (при необходимости) перерасчет управляющих воздействий (таблицы маршрутизации, используя порядок каналов и буферных ресурсов) в соответствии с текущим состоянием изменения ТКС. Как правило, при осуществлении этих протоколов считаются незначительным количеством системных параметров, которые установлены в административном или определенных пути статистического усреднения результатов измерений. Однако, следует иметь в виде, что ТКС – это сложная динамическая многопараметрическая и слабо детерминированная система, позволяющий для возможности реализации различных стратегий управления и

методы управления. Следовательно, для достижения необходимого уровня адекватности математического описания ТКС в оптимальном управлении своими ресурсами возможно лишь в рамках моделей, учитывающих динамические характеристики системы. В данной работе, в целях повышения адекватности математического описания ТКС предлагается рассмотреть возможность использования для решения задачи маршрутизации Хопфилда нейронной сети.

В типичном случае решение проблемы маршрутизации (определение кратчайшего пути) будет рассматриваться граф  $G$ :

(8)

где  $V$  – множество вершин, с количеством вершин равным  $N$ , а каждая вершина представляет собой узел (маршрутизатор) ТКС;

$L$  – множество рёбер графа, каждый из которых моделирует связь между узлами.

Число рёбер графа равно  $M$ .

(9)

где  $c_{ij}$  – стоимость передачи пакета между узлами  $i$  и  $j$ .

При этом предполагается, что  $c_{ij} = c_{ji}$  и  $c_{ii} = 0$ .

Путь  $P$  от узла  $s$  к узлу  $d$  через вершины графа  $G$ , подразумевается упорядоченный набор:

(10)

Тогда, чтобы найти кратчайший путь, критерий  $C$  должен определяться минимумом следующего выражения:

(11)

где  $\alpha_{ij}$  – доля интенсивности входящего в ТКС потока, протекающего между узлами  $i$  и  $j$ .

Как правило, должны быть выполнены правила, чтобы сохранить поток ТКС в узел, который может быть записан в следующем виде:

(12)

где

(13)

Решение задач поиска кратчайшего пути методами полиномиальных алгоритмов было продемонстрировано и описано в работе [7]. На основе этих алгоритмов является разработка современных полиномиальных-алгоритмических методов оптимизации.

Одним из способов решить задачи маршрутизации являются нейронные сети. Одним из первых научных трудов по решению маршрутизации написал Хопфилд [8]. В работе [4] рассматривается решение задач оптимизации на примере решения задачи коммивояжера. Решение данной задачи было предложено проводить с использованием нейронной сети с обратными связями, называемой в дальнейшем нейронной сетью Хопфилда [4]. Главным результатом, сделанным в работах [9-12], является применение функции Ляпунова как решающей функции задачи оптимизации. Однако представленная в них функция, давала возможность находить замкнутый путь на графе, длина которого достаточно близка или даже равна минимальному пути:





Это была первая функция Ляпунова, которая решала задачу нахождения кратчайшего пути из узла  $s$  в узел  $d$  и была нечувствительна к изменению топологии ТКС и к динамическому изменению весов . Однако при реализации данной функции оказалось, что с ростом числа узлов в ТКС сеть склонна находить пути неоптимальной длины, а также циклические пути (петли).

Для преодоления приведенных трудностей в работе [9] предложена следующая функция Ляпунова:

$$\begin{aligned} & \text{---} & & \text{---} & & \text{---} \\ & & & & & \\ & & & \text{---} & & \text{---} \end{aligned} \tag{17}$$

Эта функция позволяет избежать циклических петель при нахождении пути и обеспечивает быструю сходимость к оптимальному решению, составляющему маршрут наименьшей длины. Но для еще более быстрой сходимости в работе [10] предложена усовершенствованная функция Ляпунова:

$$\begin{aligned} & & & \text{---} & & \\ & & & & & \\ & & & \text{---} & & \end{aligned} \tag{18}$$

Как видно (18) – это развитие функции (17) путем ввода двух дополнительных слагаемых, обеспечивающих более быструю сходимость расчетной процедуры.

Последующее развитие идей маршрутизации с применением нейронной сети Хопфилда получено в работе [11]. Предложенная в данной статье функция Ляпунова (энергии нейронной сети) выглядит так:

$$\begin{aligned} & \dots \\ & \dots \\ & \dots \end{aligned} \tag{19}$$

где

Применение функции (19) позволяет получить большую скорость сходимости решений для довольно большого количества узлов в ТКС.

В работе [12] сделан анализ возможностей использования для решения задачи маршрутизации нейронных сетей, которые отличны от сетей Хопфилда. Полученные при этом данные говорят о том, что более приемлемыми для решения задач данного вида являются именно нейронные сети Хопфилда.

## 5 Практическая реализация искусственной нейронной сети

### 5.1 Выбор среды проектирования

Реализацию численного алгоритма, используемого для моего исследования, начато с прототипирования с помощью Matlab, но затем проект был переведен проект на C++, чтобы получить улучшение производительности в 10 до 100 раз.

Есть много тонкостей работы обоих языков программирования:

1) Matlab является языком сценариев, но C++ компилируется. Matlab использует jit-компилятор для перевода скрипта в машинный код, однако, можно улучшить скорость не более чем в 1,5-2 раза, используя компилятор, который предоставляет Matlab.

2) Matlab код может быть в состоянии – полностью векторизован, но существует возможность оптимизировать свой код вручную в C++. Полностью векторизованный код Matlab может вызывать библиотеки, написанные на C++/C/Assembly. Но простой C++ код может быть достаточно векторизован современными компиляторами.

3) Наборы инструментов и подпрограммы, которые предоставляет Matlab, должны быть очень хорошо настроены и иметь разумную производительность. Кроме процедур линейной алгебры, производительность обычно плохая.

Причины, по которым можно получить производительность 10x~100x в C++ по сравнению с векторизованным кодом Matlab:

1) Вызов внешних библиотек (MKL) в Matlab стоит времени.

2) Память в Matlab динамически выделяется и освобождается. Например, для умножения малых матриц требуется создание 2 временных матриц. А в C++, память выделяется вручную. И если это повторять 1000 раз, то происходит заметное увеличение производительности. Другое решение в C++ предоставляется C++11 Rvalue reference. Это одно из самых больших улучшений в C++ – код может быть так же быстр, как простой C код.





достижение указанной точки. Это происходит при расчете состояния нейронов через функцию активации, заданную сигмоидом. После того как весовые коэффициенты примут установившееся значение происходит проверка достижения узла. Если проверка пройдена, происходит переход на следующий отрезок маршрута, иначе происходит подсчет стоимости и вывод маршрута.

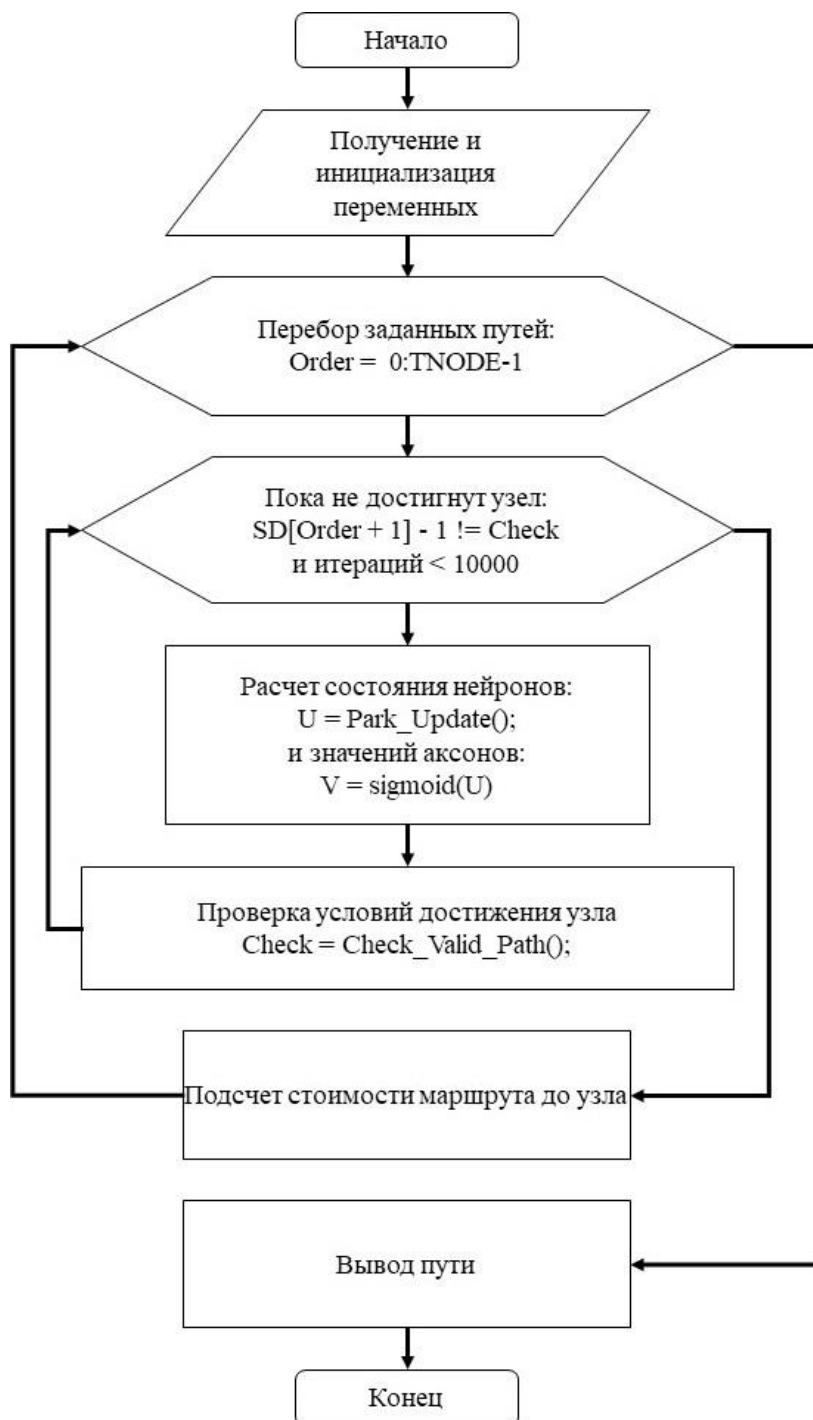


Рисунок 5.1 – Блок схема работы алгоритма

### 5.3 Анализ полученных данных

При проектировании искусственной нейронной сети Хопфилда для решения задачи маршрутизации и нахождения пути оптимальной длины будут использоваться ТКС с количеством узлов равным 40, представленной на рисунке 5.1. Чтобы найти путь минимальной длины и выразить процесс более полно, необходимо понять роль функции (19) и весовых коэффициентов

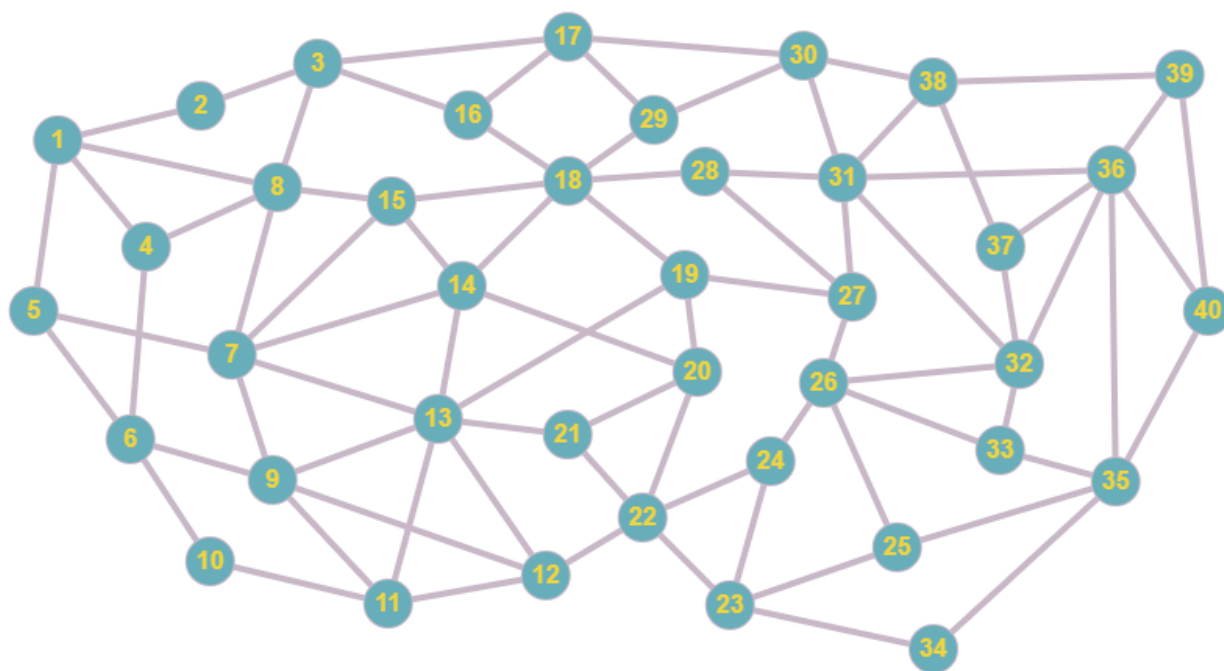


Рисунок 5.2 – Структура ТКС с 40 узлами

Первая составляющая функции (19), а также коэффициент отвечают за скорость поиска минимальной стоимости передачи пакетов по маршруту из узла  $s$  в узел  $d$ . Введение компонента функции при коэффициенте обусловлено использованием при поиске только существующих соединений. Например, не все дуги в графе могут существовать, т.е. не все маршрутизаторы подключены друг к другу, поэтому коэффициент будет равен бесконечности. В этом случае, отвечающий за связь между узлами, будет равен 1. При решении задач

Лит	Изм.	№ докум.	Подп.	Дата



маршрутизации это обеспечивает исключение несуществующих дуг за счет устранения внесения в маршрут несуществующих связей.

Составляющая с коэффициентом добавлено, чтобы гарантировать выполнение требования сохранения потока в узле в соотношении с условием (12). Чтобы убедиться, что условие выполнено, вводится член функции с коэффициентом, не позволяя получить преимущество какому либо нейрону. В процессе реализации поиска решения задачи маршрутизации нужно определить алгоритм изменения весов соединений в нейронной сети. Для этого определяется матрица весов в сети. В некоторых случаях изменение весов связей в ИНС может быть выполнено с использованием следующего выражения:

$$\text{---} \quad \text{---} \quad \text{---} \quad (20)$$

Используя метод градиентного спуска, можно минимизировать функцию активации и найти стабилизированное состояние сети Хопфилда, чтобы обеспечить кратчайший путь между узлами s и d.

### 5.3 Анализ полученных данных

Значения коэффициентов нейронной сети, использованных в опытах, приведены в таблице 5.1.

Результаты, полученные в результате работы, показанные в таблице 5.2, дают сведения о направлениях, проведенных в опытах, и процентной доле лучших путей.

Таблица 5.1 – Коэффициенты нейронной сети

Параметр							
Значение	550	2550	1500	250	1350	1	1

Таблица 5.2 – Результаты проектирования

Множество заданных	Соотношение оптимальных маршрутов, %
1, 7, 15	95
1, 7, 8	97
1, 10, 15, 26	87
1, 9, 15, 19, 23	94
1, 9, 15, 19, 23, 34	92
1, 22	73
1, 26	Маршрут не был найден за отведенное время

Результаты, полученные в результате работы, показанные на рисунках 5.3 и 5.4, дают сведения о направлениях и процентной доле лучших и оптимальных путей в общем множестве рассчитанных маршрутов при реализации всех опытов.

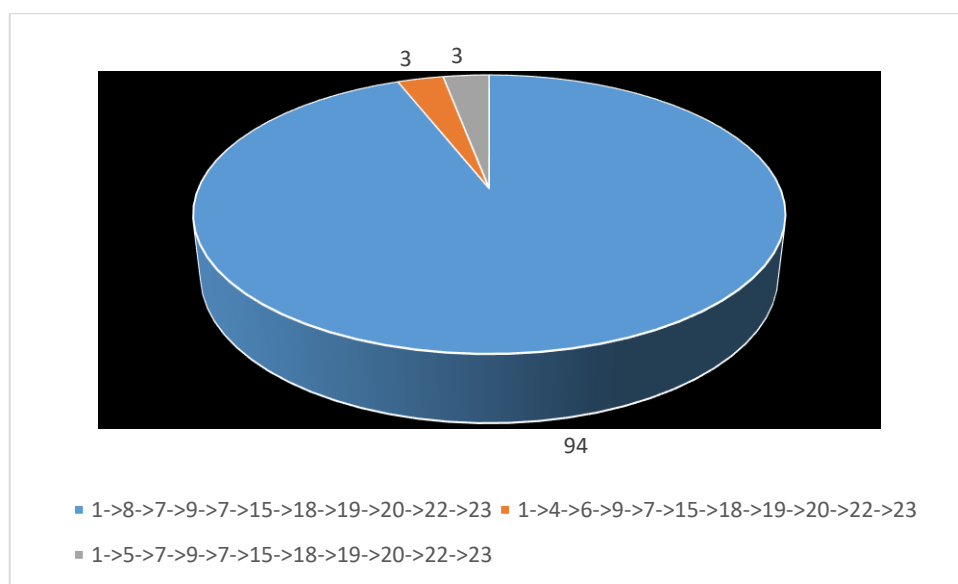


Рисунок 5.3 – Результаты для маршрутов {1, 9, 15, 19, 23}

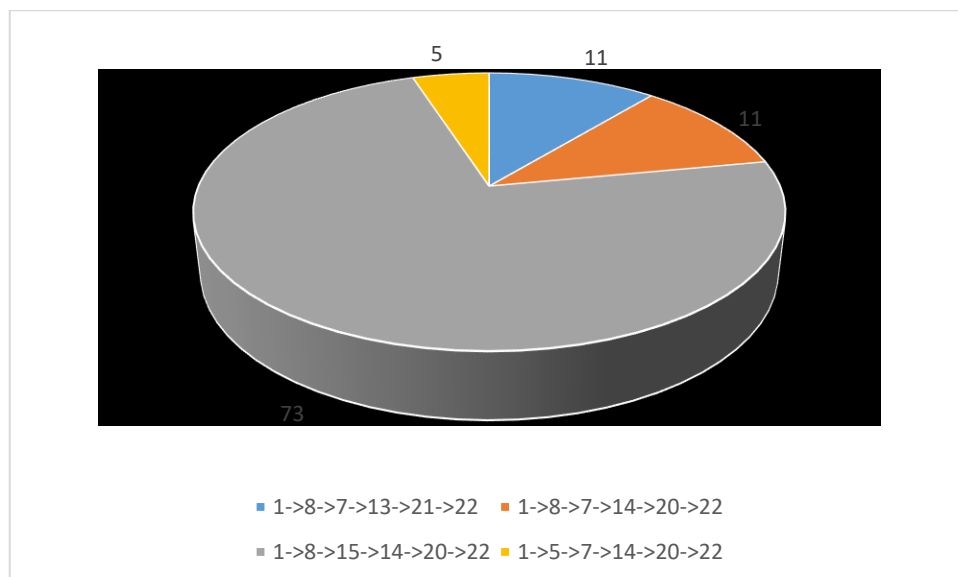


Рисунок 5.4 – Результаты для маршрутов {1, 22}

Из результатов видно, что, алгоритм применим для сетей малого объема (<20). Обеспечивает высокую оптимальность маршрута. Время затраченной на поиск всего маршрута варьируется от 10мс до 30мс. Также был проведен анализ работы сети при включении всего лишь одного цикла параллельной обработки полученной информации. Это дало прирост производительности в 2 раза, что еще раз подтверждает высокую продуктивность нейронных сетей при параллельном обрабатывании информации. Применение таких языков программирования как JAVA или Golang, могут обеспечить рост скорости работы алгоритма в десятки раз.

Функциональный код программы приведен в приложении А.

## ЗАКЛЮЧЕНИЕ

Сегодняшние искусственные нейросети представляют из себя устройства, которые используют огромное количество искусственных нейронов и нейронных связей. Невзирая на то, что цель разработки нейросетей – абсолютное моделирование человеческого мышления так и не достигнута, уже в наше время их применяют для решения огромного числа задач обработки изображений, управления робототехникой и непрерывными производствами, для синтеза и понимания речи, для диагностики болезней и технических неисправностях в приборах и машинах, для предсказания валютных курсов и пр. Если перейти на более прозаический уровень, то нейросети – это всего лишь сети, которые состоят из связанных простых элементов формальных нейронов. Центром используемых представлений выступает идея, что нейроны можно смоделировать относительно простыми автоматами, а сложность человеческого мозга, гибкость его работы и другие ключевые качества определяются нейронными связями.

В отличие от микропроцессорных цифровых систем, которые представляют из себя сложные комбинации запоминающих и процессорных блоков, основанные на нейронных сетях нейропроцессоры, содержат память, которая распределена в связях между простыми процессорами. Таким образом, основные нагрузки на реализацию функций процессорами лежит на архитектуре системы, а её детали в свою очередь определяют межнейронные связи. Созданные на подобной структуре прототипы нейронных компьютеров выдают стандартные решения многих нестандартных задач.

Необходимо отметить, что главное назначение сети Хопфилда – решать задачи обработки образов (ассоциативная память и фильтрация). Но, они позволяют находить решения комбинаторных задач оптимизации, если их можно сформулировать как задачи минимизации энергии. Если подобную сеть привести в случайное начальное состояние, то можно ждать, что результирующие

стабильное состояние предоставит субоптимальный путь, длиной не слишком превосходящим оптимальный, а может и со значительными отличиями от оптимального. Как следствие, для практического использования сеть необходимо запустить несколько раз, а потом выбрать лучший путь.

Решение этой задачи является интересным не столько качеством (есть алгоритмы, которые решают её эффективнее), сколько подходом к оптимизационным задачам: если реально перевести условия какой-либо задачи в параметры нейронных связей, то сеть может её относительно неплохо решить без дополнительного анализа.



12 Ahn C.W. Neural Network Based Near-Optimal Routing Algorithm / C.W. Ahn, R.S. Ramakrishna – Proc. of ICONIP Vol. 4, 2002. – 1771-1776 pp.

13 Dong-Chul P. A shortest path routing algorithm using Hopfield neural network with an improved energy function / P Dong-Chul, K. Kyo-Reen – International Journal of General Systems Vol. 38, No. 7, 2009. – 777-791 pp.

14 Mehmet Ali. Neural Networks for Shortest Path Computation and Routing in Computer Networks / Ali Mehmet, Mustafa K., Faouzi Kamoun – IEEE transactions on neural networks, Vol. 4, No 6, 1993. – 941-954 pp.

15 Kulvir Kaur. Shortest path routing problem in networks and optimization algorithms / Kaur Kulvir. – M.: LAP Lambert Academic Publishing, 2012. – 80 p.

16 Khadija Fatima. A Study on Migration and Labour Market Outcomes in Pakistan / Fatima Khadija. – M.: LAP Lambert Academic Publishing, 2012. – 176 p.

17 Лавренков Ю. Н. Исследование и разработка комбинированных нейросетевых технологий для повышения эффективности безопасной маршрутизации информации в сетях связи / Ю. Н. Лавренков – К.: МГТУ им. Н. Э. Баумана (калужский филиал), 2012. – 208 с.

18 Павленко М. А. Решение задачи маршрутизации на основе использования нейронной сети Хопфилда с разработкой функции Ляпунова с заданными свойствами / М. А. Павленко, А. А. Романюк, В. Ю. Яковлев // Электронное научное специализированное издание – журнал «Проблемы телекоммуникаций», 2012. – 43-57 с.

19 Комарцова Л. Г. Нейрокомпьютеры. / Л. Г. Комарцова, А.В. Максимов. – М.: МГТУ им. Н. Э. Баумана, 2004. – 400 с.

20 Ярушкина Н. Г. Основы теории нечетких и гибридных систем. / Н. Г. Ярушкина – М.: Финансы и статистика, 2009. – 320 с.

21 Советов Б. Я. Представление знаний в информационных системах / Б. Я. Советов, В.В. Цехановский, В.Д. Чертовской – М.: Академия, 2012. – 144 с.

22 Барский А. Б. Логические нейронные сети / А. Б. Барский – М.: Интернет-университет информационных технологий, Бином. Лаборатория

знаний, 2007. – 352 с.

23 Костров Б. В. Архитектура микропроцессорных систем. / Б. В. Костров, В. Н. Ручкин – М.: Диалог-МИФИ, 2007. – 304 с.

24 Яхьяева Г. Э. Нечеткие множества и нейронные сети / Г. Э. Яхьяева – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. – 320 с.

25 Романов В. П. Информационные технологии моделирования финансовых рынков / В. П. Романов, М. В. Бадрина – М.: Финансы и статистика, 2010. – 288 с.

					БР-02069964-11.03.02-17-18	Лист
Лит	Изм.	№ докум.	Подп.	Дата		56



## ПРИЛОЖЕНИЕ А

(обязательное)

### Функциональный код программы

Листинг А.1 – Главная функция программы.

```
1  #include "stdafx.h"
2  #include "cmath"
3  #include "iostream"
4  #include "conio.h"
5  #include "ctime"
6  #include <string>
7  #include <time.h>
8  #include <windows.h>
9  #include <ppl.h>
10
11 using namespace std;
12 using namespace concurrency;
13
14 // ----- Число узлов и количество назначений ----- //
15 #define NODE 20
16 #define TNODE 2
17
18 // ----- Общие параметры ----- //
19 #define DELTA 0.0001
20 #define TAU 1
21 #define LAMBDA 1
22
23 //----- Коэффициенты Park & Choi's -----//
24 #define PARK_A 550
25 #define PARK_B 2550
26 #define PARK_C 1500
27 #define PARK_D 250
28 #define PARK_F 1350
29
30 double U[NODE][NODE];
31 double V[NODE][NODE];
32 double Cost[NODE][NODE];
33 int Gamma[NODE][NODE];
34
35 int SD[TNODE];
36 int SD_order[TNODE][NODE];
37
38 int randomize() {
39     int a = 10;
40     int b = 1000;
41     return rand() % (b - a + 1) + a;
42 }
```

Лит	Изм.	№ докум.	Подп.	Дата

БР-02069964-11.03.02-17-18

Лист

57

```

43
44 //----- Сигмоид -----//
45 double Sigmoid_Function(double in)
46 {
47     return(1 / (1 + exp(-LAMBDA*in)));
48 }
49
50 // ----- Номера узлов источника и получателей ----- //
51 void Loading_Source_Desination(int Node, int Tnode)
52 {
53     // ----- Номера узлов источника и получателей ----- //
54     SD[0] = 1;
55     SD[1] = 7;
56
57     for (int i = 0; i<Tnode; i++) {
58         for (int j = 0; j<Node; j++) {
59             SD_order[i][j] = -10;
60         }
61     }
62 }
63
64 //----- Значение phi -----//
65 double PHI_Value(int a, int s, int d)
66 {
67     double Value;
68
69     if (a == s)
70         Value = 1.0;
71     else if (a == d)
72         Value = -1.0;
73     else
74         Value = 0.0;
75
76     return(Value);
77 }
78
79 // ----- Верность пути от источника до получателя ----- //
80 int Check_Valid_Path(int order, int number, int s, int d)
81 {
82     int Visit = -1, Count = 0, Position = 0, Value = 0;
83
84     for (int i = 0; i<number; i++) {
85         if (V[s][i]>0.8) {
86             SD_order[order][0] = s;
87             SD_order[order][1] = i;
88             Visit = i;
89             Position = i;
90             Count = 3;
91         }
92     }
93
94     if (d == Visit) {
95         Value = Visit;

```

```

96     }
97     else {
98         for (int i = 0; i<number - 2; i++) {
99             for (int j = 0; j<number; j++) {
100
101                 if (V[Position][j]>0.8) {
102                     if (d == j) {
103                         SD_order[order][Count - 1] = j;
104                         Visit = j;
105                         Value = Visit;
106                     }
107                     else {
108                         SD_order[order][Count - 1] = j;
109                         Count++;
110                         Position = j;
111                     }
112                 }
113             }
114         }
115     }
116
117     return(Value);
118 }
119
120 // ----- Подсчет количество путей из источника до получателя ----- //
121 int Path_Count(int order, int number)
122 {
123     int count = 0;
124
125     for (int i = 0; i<number; i++) {
126         if (SD_order[order][i] != -10)
127             count++;
128     }
129     return(count);
130 }
131
132 // ----- Стоимость между каждым источником и получателем ----- //
133 double Cost_Value_Save(int order, int count)
134 {
135     double Cost_Sum = 0.0;
136
137     for (int i = 0; i<count - 1; i++) {
138         Cost_Sum += Cost[SD_order[order][i]][SD_order[order][i + 1]];
139     }
140     return(Cost_Sum);
141 }
142
143 // ----- Получения пути следования ----- //
144 string Get_Path(int order, int count)
145 {
146     string path = "";
147
148     for (int i = 0; i<count - 1; i++) {

```

```

149     path += to_string(SD_order[order][i] + 1) + "->";
150     if (order == TNODE - 2 && i == count - 2) {
151         path += to_string(SD_order[order][i + 1] + 1);
152     }
153 }
154
155 return(path);
156 }
157
158 //-----Park & Choi часть мю1-----//
159 double Park_A_Term(int i, int j)
160 {
161     return(Cost[i][j]);
162 }
163
164 //-----Park & Choi часть мю2-----//
165 double Park_B_Term(int i, int j)
166 {
167     return((double)Gamma[i][j]);
168 }
169
170 //----- Park & Choi часть мю3.1 -----//
171 double Park_C_first_Term(int i, int s, int d, int Node)
172 {
173     double Sum = 0.0;
174
175     for (int k = 0; k<Node; k++) {
176         if (k != i)
177             Sum += (V[i][k] - V[k][i]);
178     }
179
180     return(Sum - PHI_Value(i, s, d));
181 }
182 //----- Park & Choi часть мю3.2 -----//
183 double Park_C_second_Term(int j, int s, int d, int Node)
184 {
185     double Sum = 0.0;
186
187     for (int k = 0; k<Node; k++) {
188         if (k != j)
189             Sum += (V[j][k] - V[k][j]);
190     }
191     return(Sum - PHI_Value(j, s, d));
192 }
193
194 //----- Park & Choi часть мю4 -----//
195 double Park_D_Term(int i, int j)
196 {
197     return(1.0 - 2.0*V[i][j]);
198 }
199 //----- Park & Choi часть мю5 -----//
200 double Park_F_Term(int i, int j)
201 {

```

```

202     return((double)Gamma[i][j] * V[j][i]);
203 }
204
205 //----- Park & Choi общая -----//
206 double Park_Update_Function(int i, int j, int s, int d, int Node)
207 {
208     double Value = 0.0;
209
210     Value = U[i][j] + DELTA * (-U[i][j] / (double)TAU
211         - 0.5*PARK_A*Park_A_Term(i, j)
212         - 0.5*PARK_B*Park_B_Term(i, j)
213         - PARK_C * Park_C_first_Term(i, s, d, Node)
214         + PARK_C * Park_C_second_Term(j, s, d, Node)
215         - 0.5*PARK_D*Park_D_Term(i, j)
216         - 0.5*PARK_F*Park_F_Term(i, j)
217
218         - 0.5*PARK_A*Cost[i][j]
219         - 0.5*PARK_B*(double)Gamma[i][j]
220         + PARK_C*PHI_Value(i, s, d)
221         - PARK_C*PHI_Value(j, s, d)
222         - 0.5*PARK_D
223         );
224
225     return(Value);
226 }
227
228 //-----Пересчет состояний неронов-----//
229 void Park_Hopfield_NN(int s, int d, int Node)
230 {
231     parallel_for(0, Node, [&](int x)
232     {
233         for (int i = 0; i<Node; i++) {
234             U[x][i] = Park_Update_Function(x, i, s, d, Node);
235             V[x][i] = Sigmoid_Function(U[x][i]);
236         }
237     });
238 }
239
240 void main()
241 {
242     __int64 start, end, tps;
243
244     QueryPerformanceFrequency((LARGE_INTEGER *)&tps);
245
246     string Path = "";
247
248     int Check = -1;
249     int Number, Epoch, Count;
250     double T_Cost = 0.0;
251
252     Loading_Source_Desination(NODE, TNODE);
253     Loading_Link_Information(NODE);
254

```

```

255 QueryPerformanceCounter((LARGE_INTEGER *)&start);
256
257 for (int Order = 0; Order < TNODE - 1; Order++) {
258     Number = NODE;
259
260     Initialize_Value(NODE);
261
262     Epoch = 0;
263
264     while (SD[Order + 1] - 1 != Check && Epoch <= 10005) {
265         Epoch++;
266         Park_Hopfield_NN(SD[Order] - 1, SD[Order + 1] - 1, Number);
267         Check = Check_Valid_Path(Order, Number, SD[Order]-1,SD[Order+1]-1);
268     }
269     if (Epoch >= 10000) {
270         cout << "Алгоритм не сходится при данных условиях" << endl;
271         break;
272     }
273
274     // --- Подсчитываем количество путей обхода для каждого S->D --- //
275     Count = Path_Count(Order, Number);
276
277     //---Общая длина---//
278     T_Cost += Cost_Value_Save(Order, Count);
279
280     Path += Get_Path(Order, Count);
281 }
282
283 QueryPerformanceCounter((LARGE_INTEGER *)&end);
284 double time = ((double)(end - start) / tps) * 1000.;
285
286 cout << "Time: " << Time << "ms" << endl;
287 system("pause");
288 }

```

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»  
(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

**ОТЧЕТ**

*о результатах проверки бакалаврской работы обучающегося  
на наличие заимствований*

Автор работы Потапкин Кирилл Олегович

Тема работы «Разработка искусственной нейронной сети для решения задачи маршрутизации»

Руководитель Маняев Игорь Витальевич

Представленная работа прошла проверку на наличие заимствований в системе «Антиплагиат. ВУЗ»

Результаты автоматической проверки:	оригинальность	82,77%
	цитирования	0,25%
	заимствования	16,98%

Результаты анализа полного отчета на наличие заимствований:  
правомерные заимствование: 16,98%

корректные цитирования: 0,25%

неправомерные заимствования: нет

признаки обхода системы: нет

Общее заключение об итоговой оригинальности работы и возможности ее допуска к защите: система показала, что оригинальный текст в проверяемом документе составляет 82,77%, остальные 16,98% присутствуют в 25 источниках. Анализ источников показал, что в исследуемом тексте имеются корректные совпадения в виде наименований публикаций, конференций, терминологии и устойчивых словосочетаний, принятых в данной сфере науки, нормативных и правовых документах.

Руководитель

(подпись)



И. В. Маняев

ст. преподаватель

(дата)

14.06.18

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»  
(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

**ОТЗЫВ**

*о бакалаврской работе*

студента Потапкина Кирилла Олеговича,

обучающегося по направлению подготовки

11.03.02 Инфокоммуникационные технологии и системы связи

на тему «Разработка искусственной нейронной сети для решения задачи маршрутизации»

Бакалаврская работа выполнена в полном объеме и представлена к сроку. Она состоит из пяти глав.

Целью выпускной работы является разработка искусственной нейронной сети для решения задачи маршрутизации.

Актуальность исследований в этом направлении подтверждается массой различных применений нейронных сетей. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие приложения.

В процессе работы Потапкин К. О. показал себя грамотным, способным самостоятельно решать поставленные перед ним задачи. При выполнении работы использовано значительное количество специальной литературы (периодические издания, патенты), использование лицензированных программ, что говорит об умении разбираться в достаточно сложных технических вопросах в области связи.

В процессе разработки были рассмотрены следующие вопросы: способы организации потоков информации, искусственные нейронные сети, методы обучения нейронной сети, алгоритмы работы сети Хопфилда. Также была произведена практическая реализация искусственной нейронной сети.

Бакалаврская работа выполнена и оформлена по государственным стандартам.

Считаю, что бакалаврская работа заслуживает высокой оценки, а студент Потапкин К. О. – присвоения квалификации бакалавра по направлению «Инфокоммуникационные технологии и системы связи».

Научный руководитель  
« 14 » июня 2018г.



И. В. Маняев