

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ  
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»  
(СПБГУТ)**

---

**В.С. Елагин, А.Г. Владыко, Р.И. Пупцев, Д.В. Окунева**

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ.  
ПРИКЛАДНЫЕ АСПЕКТЫ  
ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ  
В СИСТЕМАХ УПРАВЛЕНИЯ  
ИНФОКОММУНИКАЦИЯМИ**

**Учебное пособие**

**СПБ ГУТ)))**

**САНКТ-ПЕТЕРБУРГ  
2020**

УДК 004.7(075.8)

ББК 32.973-02я73

E44

Рецензент

д-р техн. наук Соколов Н.А. (ЛО ЦНИИС)

Рекомендовано к печати

редакционно-издательским советом СПбГУТ

Елагин В.С., Владыко А.Г., Пупцев Р.И., Окунева Д.В.

**E44 Елагин, В.С.** Искусственный интеллект. Прикладные аспекты применения нейронных сетей в системах управления инфокоммуникациями / В.С. Елагин, А.Г. Владыко, Р.И. Пупцев, Д.В. Окунева. – СПб: СПбГУТ, 2020. – 70 с.

Содержит учебный материал с описанием технологических основ применения математического аппарата нейронных сетей в инфокоммуникациях. Особое внимание уделено вопросу применения нейронных сетей в системах управления операционными процессами в сетях связи. Материал учебного пособия представлен в виде теоретической и практической частей, включает в себя методики проведения адаптируемого к слушателю обучения и практических занятий.

Предназначено для студентов профильных специальностей: 09.03.02, (09.04.02) «Информационные системы и технологии» 11.03.02 (11.04.02) «Инфокоммуникационные технологии и системы связи», а также специалистам работающих на сетях связи различного назначения.

УДК 004.7(075.8)

ББК 32.973-02я73

ISBN

- © В.С. Елагин, А.Г. Владыко, Р.И. Пупцев, Д.В. Окунева, 2020
- © Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2020

## СОДЕРЖАНИЕ

Перечень сокращений .....	4
Введение .....	5
1. История нейронных сетей .....	7
2. Введение в нейронные сети.....	11
2.1. Простейшая модель нейрона .....	11
2.2. Активационная функция.....	14
2.3. Архитектура нейронных сетей .....	15
2.4. Области применения нейронных сетей .....	24
3. Проектирование модели сети управления трафиком .....	26
3.1. Управление трафиком сети передачи данных .....	26
3.2. Моделирование структуры сети .....	29
3.3. Основные параметры модели управления .....	31
3.4. Инструменты для проектирования .....	36
3.5. Данные для обучения и их обработка .....	41
4. Практическая работа .....	57
4.1. Создание нейронной сети .....	57
4.2. Результаты первого эксперимента .....	58
4.3. Результаты второго эксперимента .....	62
4.4. Выводы по результатам экспериментов .....	66
Заключение.....	68
Литература .....	69

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

**ИНС** (ANN (Artificial Neural Networks)) – искусственная нейронная сеть

**МП** – многослойный перцептрон

**НМ** – нейронная модель

**НС** – нейронная сеть

**ПД** – передача данных

**СПД** – сеть передачи данных

**АРР** – автоматическое распознавание речи

**CNN** (Convolutional Neural Network) – свёрточная нейронная сеть

**DRL** (Deep Reinforcement Learning) – глубокое обучение с подкреплением

**FNN** – нечеткая нейронная сеть

**QoS** (Quality of Service) — технология предоставления различным классам трафика различных приоритетов в обслуживании

**OSS** (Operations Support Systems) – система поддержки операций/система поддержки бизнеса

**RNN** (Recurrent neural network) – рекуррентная нейронная сеть

**RRBFN** (Recurrent Radial Basis Function Networks) – повторяющаяся радиально-базисная функция

**SDN** (Software-Defined Networking) – программно-конфигурируемая сеть

## ВВЕДЕНИЕ

Еще на заре развития техники и технологий стояла проблема создания систем для различного рода интеллектуальной деятельности – так называемого искусственного интеллекта. Основным аспектом данной проблемы является то, что искусственная система должна не только функционировать по определённым алгоритмам, но и быть адаптивной, обучаемой, генерировать знания и алгоритмы решения задач. Развитие таких систем происходит за счет самоорганизации, в результате которой осуществляется адаптация к решаемой задаче.

Существует несколько основных направлений в приложениях искусственного интеллекта. Одни традиционные направления используют методы логических рассуждений и символьной обработки информации, другие связаны с интеллектуальными вычислениями, которые включают в себя генетические алгоритмы, эволюционное программирование, нечеткие системы и нейронные сети [1]. Последние исследуют сети, состоящие из нейронных элементов, опираются на биологические основы естественного интеллекта, что позволяет проектировать системы, способные к обучению и самоорганизации. В данном пособии основное внимание уделено рассмотрению истории и теории нейронных сетей (НС) и её использованию для решения различного рода задач в инфокоммуникациях. Нейронные сети способны обучаться, этот дает им главное преимущество перед традиционными алгоритмами. В процессе обучения НС способна выявлять сложные зависимости между входными и выходными данными. К наиболее известным применениям искусственных нейронных сетей (ИНС) относятся: распознавание образов, принятие решений и управление, кластеризация, прогнозирование, аппроксимация, сжатие данных и ассоциативная память, анализ данных, оптимизация. Благодаря использованию ИНС становится возможным решение проблемы затруднительного анализа инфраструктур и автоматизации систем поддержки эксплуатационных процессов.



## 1. ИСТОРИЯ НЕЙРОННЫХ СЕТЕЙ

Термин «нейронная сеть» появился в середине XX века. Первые работы, в которых были получены основные результаты в данном направлении, были проделаны Уорреном МакКаллоком и Уолтером Питтсом [2]. В 1943 году ими была разработана компьютерная модель нейронной сети на основе математических алгоритмов и теории деятельности головного мозга. Они выдвинули предположение, что нейроны можно упрощённо рассматривать как устройства, оперирующие двоичными числами, и назвали эту модель «пороговой логикой». Подобно своему биологическому прототипу нейроны МакКаллока–Питтса были способны обучаться путём подстройки параметров, описывающих синаптическую проводимость. Исследователи предложили конструкцию сети из электронных нейронов и показали, что подобная сеть может выполнять практически любые вообразимые числовые или логические операции. МакКаллок и Питтс предположили, что такая сеть в состоянии также обучаться, распознавать образы, обобщать, т. е. обладает всеми чертами интеллекта.

Данная модель заложила основы для двух различных подходов исследований нейронных сетей. Один подход был ориентирован на изучение биологических процессов в головном мозге, другой – на применение нейронных сетей как метода искусственного интеллекта для решения различных прикладных задач.

В 1949 году канадский физиолог и психолог Дональд Хебб высказал идеи о характере соединения нейронов мозга и их взаимодействии [3]. Он первым предположил, что обучение заключается в первую очередь в изменениях силы синаптических связей. Теория Хебба считается типичным случаем самообучения, при котором испытываемая система спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. В более поздних вариантах теория Хебба легла в основу описания явления долговременной потенциации.

В 1954 году в Массачусетском технологическом институте с использованием компьютеров Бельмонт Фарли и Уэсли Кларк разработали имитацию сети Хебба [4]. Также исследования нейронных сетей с помощью компьютерного моделирования были проведены Н. Рочестером, Д. Холландом, Л. Хебитом и У. Дудой в 1956 году [5].

В 1958 году Фрэнком Розенблаттом были разработаны математическая и компьютерная модели восприятия информации мозгом на основе двухслойной обучающейся нейронной сети [6]. При обучении данная сеть использовала арифметические действия сложения и вычитания. Розенблатт описал также схему не только основного перцептрона, но и схему логического сложения. Кроме этого, им была предложена модель электронного устройства, которое должно было имитировать процессы человеческого мышления, а два года спустя была продемонстрирована первая действующая машина, которая могла научиться распознавать некоторые из букв, написанных на карточках, которые подносили к его «глазам», напоминая кинокамеры.

Интерес к исследованию нейронных сетей угас после публикации работы по машинному обучению Марвина Минского и Сеймура Пейперта в 1969 году [7]. Ими были обнаружены основные вычислительные проблемы, возникающие при компьютерной реализации искусственных нейронных сетей. Первая проблема состояла в том, что однослойные нейронные сети не могли совершать «сложение по модулю 2», то есть реализовать функцию «исключающее ИЛИ». Второй важной проблемой было то, что компьютеры не обладали достаточной вычислительной мощностью, чтобы эффективно обрабатывать огромный объём вычислений, необходимый для больших нейронных сетей.

Исследования нейронных сетей замедлились до времени, когда компьютеры достигли больших вычислительных мощностей. Одним из важных шагов, стимулировавших дальнейшие исследования, стала разработка в 1975 году Полом Вербосом метода обратного



распространения ошибки, который позволил эффективно решать задачу обучения многослойных сетей и решить проблему со «сложением по модулю 2» [8].

В 1980 году Кунихикой Фукусимой был представлен неокогнитрон, который стал одной из первых многослойных нейронных сетей [9]. Фактическая структура сети и методы, используемые в неокогнитроне для настройки относительных весов связей, варьировались от одной стратегии к другой. Каждая из стратегий имела свои преимущества и недостатки. Сети могли распространять информацию только в одном направлении или перебрасывать информацию из одного конца в другой, пока не активировались все узлы и сеть не приходила в конечное состояние. Достиж двусторонней передачи информации между нейронами удалось лишь в ассоциативной нейронной сети, предложенной Джоном Хопфилдом в 1982 году, и специализация этих узлов для конкретных целей была введена в первых гибридных сетях [10].

Алгоритм параллельной распределённой обработки данных в середине 1980 годов стал популярен под названием коннективизма. В 1986 году в работе Дэвида Руммельхарта и Джеймса Макклелланда коннективизм был использован для компьютерного моделирования нейронных процессов [11].

Несмотря на большой энтузиазм, вызванный в научном сообществе разработкой метода обратного распространения ошибки, это также породило многочисленные споры о том, может ли такое обучение быть на самом деле реализовано в головном мозге. Отчасти это связывали с тем, что механизм обратного прохождения сигнала не был очевидным в то время, так как не было явного источника, обучающего и целевого сигналов. Тем не менее, в 2006 году было предложено несколько неконтролируемых процедур обучения нейронных сетей с одним или несколькими слоями с использованием алгоритмов глубокого обучения. Эти алгоритмы могут быть использованы для изучения промежуточных

представлений, как с выходным сигналом, так и без него, чтобы понять основные особенности распределения сенсорных сигналов, поступающих на каждый слой нейронной сети.

Как и во многих других случаях, задачи высокой сложности требуют применения не одного, а нескольких методов решения или их синтеза. Не исключение и искусственные нейронные сети. С начала нынешнего столетия в работах различных исследователей активно описываются нейро-нечёткие сети, ячеечно-нейросетевые модели. Также нейронные сети используются, например, для настройки параметров нечётких систем управления. Нет никаких сомнений в дальнейшей интеграции методов искусственного интеллекта между собой и с другими методами решения задач.

## 2. ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ

### 2.1. Простейшая модель нейрона

За основу построения искусственных нейронных сетей был взят принцип биологических нейронных сетей, которые являются сетью нервных клеток, выполняющих необходимые физиологические функции. Основными элементами нейронных сетей являются нейроны (рисунок 1).

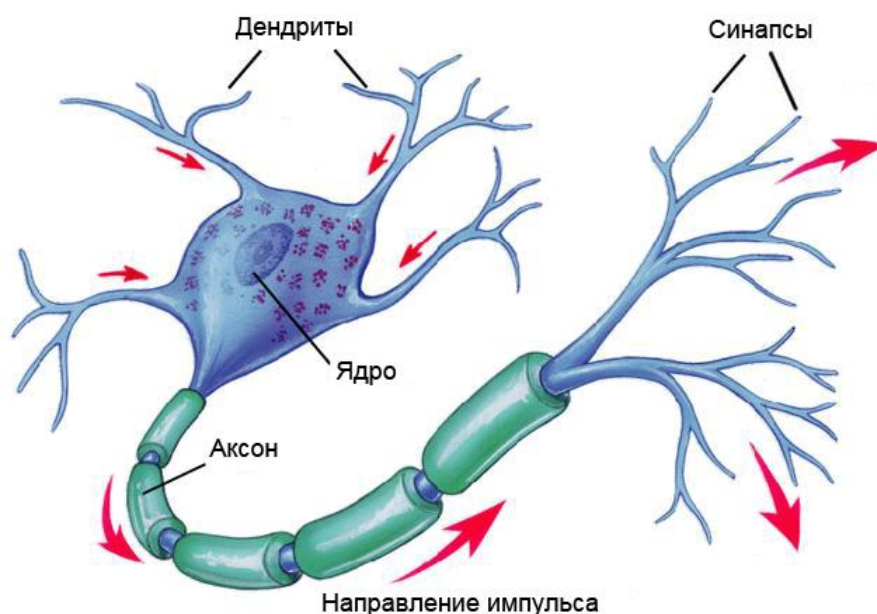


Рисунок 1. Биологическая модель нейрона.<sup>1</sup>

От центральной части нейрона (сомы) отходят древовидные отростки – дендриты. Они играют роль рецепторов – сигналов от других нейронов. Задача аксона, самого крупного отростка, заключается в том, чтобы передавать сигнал активности от сомы другим нейронам. Место соединения аксона с дендритами других нейронов разделено малым, порядка 200 нм, расстоянием. Этот промежуток принято называть синапсом.

Отсюда мы можем выделить несколько функций нейрона:

- функция приема: нейрон получает информацию через синапс;

<sup>1</sup> Использован рисунок из учебного пособия: Анисимов В.В., Ещенко Р.А. Интеллектуальные информационные системы. Хабаровск: Изд-во ДВГУПС, 2017.

- функция проводника: аксон передает сигнал активности от сомы другим нейронам;
- функция интеграции: на выходе нейрона имеется сумма всех сигналов, прошедших через него;
- функция передачи: импульс, достигший окончания аксона, заставляет медиатор передавать возбуждение следующему нейрону.

Синапсы — это связи между двумя соседними нейронами. У каждой связи есть своя характеристика - вес. Если вес положительный, то связь возбуждающая. Соответственно, если вес отрицательный, то связь тормозящая.

В искусственной нейронной сети нейрон представляет собой нелинейную функцию от единственного аргумента - линейной комбинации всех входных сигналов. Эту функцию принято называть активационной. Результат активационной функции посылается на выход нейрона, а собрав вместе такие нейроны, получают нейронную сеть.

Рассмотрим модель нейрона, связанную с первыми попытками формализовать описание функционирования нервной клетки. Введем следующие обозначения:

$u_1, \dots, u_n$  - входные сигналы данного нейрона, приходящие от других нейронов;

$w_1, \dots, w_n$  - синаптические веса;

$y$  - выходной сигнал нейрона;

$v$  - пороговое значение.

Формула, описывающая функционирование нейрона, имеет вид:

$$y = \begin{cases} 1, & \text{если } \sum_{i=1}^n w_i u_i \geq v, \\ 0, & \text{если } \sum_{i=1}^n w_i u_i < v \end{cases} \quad (1)$$

Модель (1) может быть представлена в виде:

$$y = f(\sum_{i=1}^n w_i u_i), \quad (2)$$

где:

$$f(x) = \begin{cases} 1, & \text{если } x \geq 0, \\ 0, & \text{если } x < 0 \end{cases} \quad (3)$$

$$w_0 = v, \quad u_0 = 1.$$

Формула (2) описывает модель нейрона, представленную на рисунке

2.

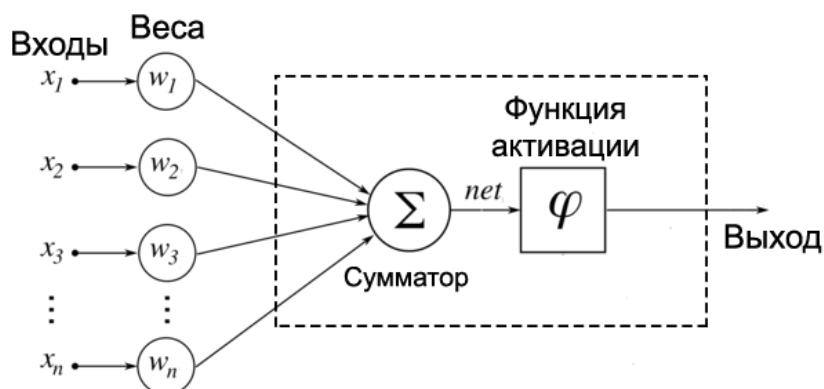


Рисунок 2. Модель нейрона.

Эта модель была предложена МакКаллоком и Питтсом [2]. В качестве функции  $f$  может приниматься не только единичная функция (3), но и другие пороговые функции вида:

$$f(x) = \begin{cases} 1 & \text{при } x \geq 0 \\ -1 & \text{при } x < 0 \end{cases} \quad (4)$$

или

$$f(x) = \begin{cases} 1 & \text{при } x > 1, \\ -1 & \text{при } x < -1, \\ x & \text{при } |x| \leq 1 \end{cases} \quad (5)$$

Нелинейную функцию  $f(x)$  называют активационной. Когда активационная функция одна и та же для всех нейронов сети, сеть называется однородной (гомогенной). При условии, что функция активации имеет зависимость еще от одного или нескольких параметров с меняющимися значениями от нейрона к нейрону, сеть называется неоднородной (гетерогенной).

Нейронные сети также можно классифицировать как бинарные и аналоговые. Бинарные работают с двоичными сигналами, на выходе

нейрона находится исключительно два значения: логический ноль («заторможенное» состояние) и логическая единица («возбужденное» состояние). Значения в выходе аналоговых сетей могут принимать непрерывные значения.

Существует классификация нейронных сетей на синхронные и асинхронные. В синхронных нейронных сетях только один нейрон изменяет свое состояние в каждый момент времени. В асинхронных - состояние изменяется у группы нейронов в один момент и, в основном, у всего слоя. Решение о синхронизации при использовании программных имитаторов нейронных сетей принимаются вычислительной системой.

## 2.2. Активационная функция

Чтобы понять суть этой функции, рассмотрим какое место она занимает в искусственном нейроне.

Слева на рисунке 2 обозначены входы нейрона, что является имитацией дендрита, входные значения  $x_1-x_n$  и их веса  $w_1-w_n$ . Вес – «ценность» или «важность» значения на входе. Затем значения умножаются на свои веса и поступают в сумматор, где складываются. После чего полученная сумма поступает на вход функции активации.

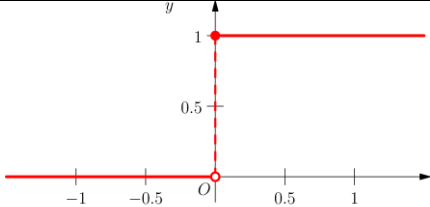
Функция активации принимает решение о возбуждении нейрона. Изначально все входные и выходное значения задумывались как бинарные. Следовательно «0» на выходе означал, что сумма весов не преодолела некий порог и нейрон не активировался.

Некоторые примеры активационных функций:

а) Пороговая функция (Таблица 1).

Если значение на входе меньше порогового, то значение функции активации равно минимально допустимому, иначе – максимальному.

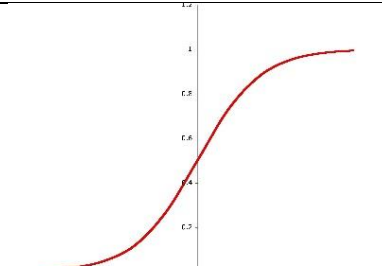
Таблица 1. Единичная функция активации

Единичная функция	$f(x) = \begin{cases} 1 & \text{при } x \geq 0 \\ 0 & \text{при } x < 0 \end{cases}$	
-------------------	--	--

б) Сигмоидальная функция (Таблица 2).

Благодаря этой функции можно получить непрерывные выходные значения в диапазоне  $[0, 1]$ .

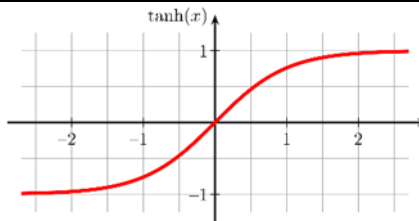
Таблица 2. Сигмоидальная функция активации

Сигмоидальная функция	$f(x) = \frac{1}{1 + e^{-ax}}$	
-----------------------	--------------------------------	--

в) Гиперболический тангенс (Таблица 3).

Гиперболический тангенс стоит использовать, когда необходимо получить отрицательные значения на выходе.

Таблица 3. Гиперболический тангенс, как функция активации

Гиперболический тангенс	$f(x) = \tanh\left(\frac{\alpha x}{2}\right) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}}$	
-------------------------	---	--

## 2.3. Архитектура нейронных сетей

### 2.3.1. Нейронные сети прямого распространения

Нейронные вычисления начинаются не с работы одного нейрона, что выполняет простейшие функции, а с нейронной сети. Задачи, решаемые нейронными сетями, вынуждают создавать узконаправленные нейронные сети, выполняющие различные алгоритмы, и не дают возможность к разработке универсальных сетей.

Решение о подборе архитектуры нейронной сети принимается на основе сложности задачи и ее спецификой. На данный момент есть типы задач, под которые уже существуют оптимизированные конфигурации нейронной сети. Основная часть представлена на рисунке 3 [12].

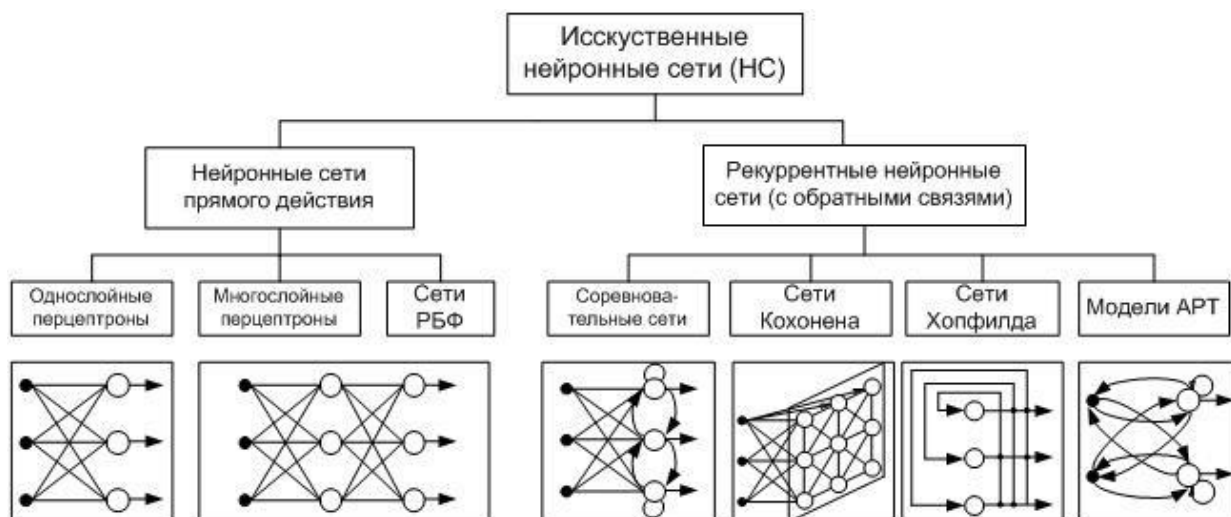


Рисунок 3. Основные архитектуры НС

Самая простая однослойная сеть (однослойный перцептрон), которая состоит из группы нейронов, представлена на рисунке 4.

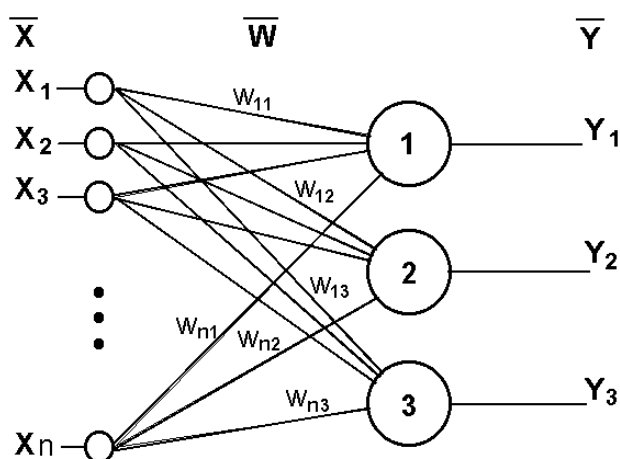


Рисунок 4. Однослойная сеть.



На  $n$  входов поступают сигналы, переходящие по синапсам на три нейрона, создающие единственный слой данной нейронной сети. На выходе получаются три сигнала:

$$y_j = f\left[\sum_{i=1}^n x_i w_{ij}\right], \quad (6)$$

где  $j = 1, 2, 3$ .

В матричной форме:  $Y = F(XW)$

На рисунке 5 представлена двуслойная нейронная сеть, образовавшаяся из однослойной посредством добавления второго слоя, строящегося из двух нейронов. Сеть, в которой два или более слоя, называют многослойным персептроном [8].

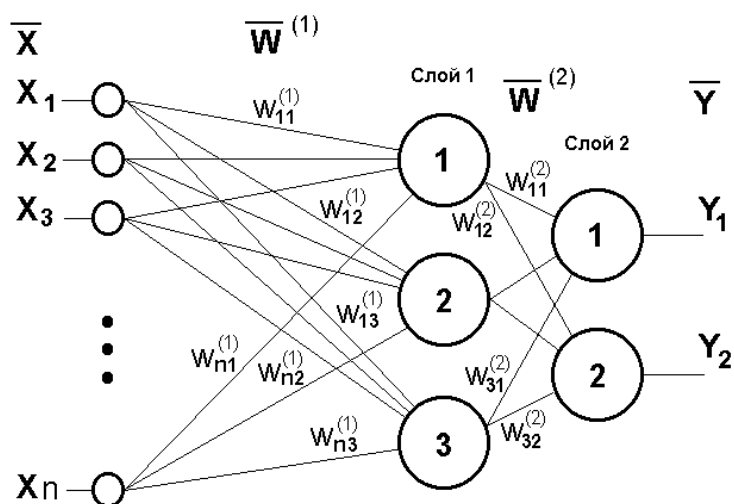


Рисунок 5. Двуслойная нейронная сеть.

В этом случае вычисления происходят по принципу того, что р-слойная нейронная сеть равнозначна однослойной нейронной сети с весовой матрицей единственного слоя  $W^{(\Sigma)}$ :

$$Y = XW^{(\Sigma)}, \text{ где } W^{(\Sigma)} = W^{(1)}W^{(2)} \dots W^{(p)}, \quad (7)$$

### 2.3.2. Типы архитектур НС и примеры их применения.

Рассмотрим наиболее популярные архитектуры нейронных сетей.

1) Свёрточная нейронная сеть (Convolutional Neural Network, CNN) - специальная архитектура искусственных нейронных сетей. Идея

свёрточных нейронных сетей заключается в чередовании свёрточных слоёв и слоёв подвыборки. Структура сети - многослойная с прямыми связями. Своё название архитектура сети получила из-за наличия операции свёртки, суть которой заключается в том, что каждый фрагмент изображения умножается на ядро свёртки поэлементно, результат суммируется и записывается в аналогичную позицию выходного изображения.

Свёрточная нейронная сеть в основном используется для работы с аудио и графической информацией. Эта архитектура обрабатывает данные не целиком, а фрагментами, но при этом данные не дробятся на части, а осуществляется своего рода последовательный прогон. После чего данные передаются дальше по слоям. Кроме свёрточных слоев (С) используются также слои объединения (P-pooling).

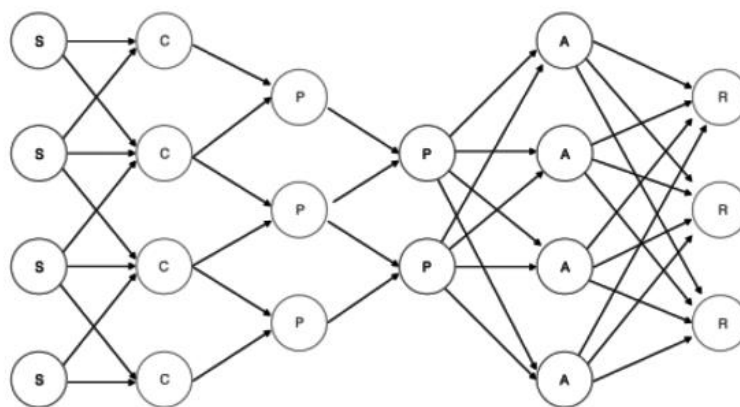


Рисунок 6. Архитектура сверточной нейронной сети.

Свёрточная нейронная сеть нуждается в настройке большого количества варьируемых параметров, таких как количество слоев, количество ядер в каждом слое, функции активации каждого нейрона и многие другие.

Ниже представлены наиболее известные примеры применения сверточной нейронной сети.

- LeNet. Яну Лекуну в 1990-е годы удалось разработать первое успешное применение свёрточной нейронной сети [13]. Архитектура LeNet применялась для считывания почтовых индексов и цифр. Последняя

версия сети LeNet-5 демонстрировала на тот момент впечатляющие результаты из-за реализации нескольких новаторских идей. Основным отличием от неокогнитрона было включение субдискретизирующего max-pooling слоя после каждого свёрточного слоя и включение полносвязных слоёв на выходе сети. С этого момента свёрточные и max-pooling-слои становятся сердцем современных глубоких сетей.

- AlexNet. Это совместная работа Алекса Крижевского, Ильи Суцкевера и Джеффа Хинтона, которая сыграла значительную роль в популяризации CNN в области компьютерного зрения [14]. Архитектура включает свёрточные слои, max-pooling-слои, полносвязные слои на выходе сети. Использовались методы исключения (dropout) и локальной нормализации. В качестве функции активации использовалась функция ReLU  $f(x) = \max(0, x)$ .

- ZF Net. Свёрточная нейронная сеть, созданная Мэтью Зейлером и Робом Фергюсом [15]. Данная архитектура была улучшенной версией AlexNet - в ней увеличили размеры средних свёрточных слоев, уменьшили шаг и размер фильтра на первом слое.

- GoogLeNet. Свёрточная нейронная сеть, разработанная сотрудниками корпорации Google [16]. Основная заслуга данной архитектуры состоит в разработке и внедрении входного модуля (Inception Module), что позволило резко сократить число параметров с 60 миллионов до 4 миллионов. Также сокращение параметров происходит благодаря замене полносвязных слоев в верхней части сети слоями среднего пулинга. В архитектуре был уменьшен размер свёртки, были параллельно включены небольшие свёртки разного масштаба, удалены полносвязные слои на выходе сети и вместо них включены слои, названные global average pooling. Активное развитие архитектура достигла благодаря своей эффективности. *Inception-v3* – последняя реализация, включающая несколько десятков слоёв.

- VGGNet. Сеть является совместной работой Карена Симоняна

и Эндрю Циссермана [17]. Разработчикам удалось наглядно продемонстрировать, что глубина является ключевым фактором для производительности. Их сеть содержит 16 свёрточных и полносвязных слоев и имеет чрезвычайно однородную архитектуру, которая выполняет свёртывание  $3 \times 3$  и пулинг  $2 \times 2$  от начала до конца. Исходная модель доступна в режиме Plug and Play во фреймворке Caffe [18].

- ResNet (Residual Network). Сеть разработана Каймингом Хе [19]. Ключевыми особенностями являются: интенсивное использование пакетной нормализации и специальные скип-соединения. Данная архитектура получила дальнейшее развитие, например, в виде высокоэффективной гибридной сети Inception-ResNet. ResNet по состоянию на сегодняшний день является настоящим произведением искусства в мире свёрточных нейронных сетей и часто используется.

Основные преимущества свёрточной НС:

- является одним из лучших алгоритмов по распознаванию и классификации изображений;
- обладает относительной устойчивостью к повороту и сдвигу распознаваемого изображения;
- обучение осуществляется при помощи классического метода распознавания ошибки.

К основным недостаткам НС относятся большое количество переменных параметров сети, к которым относятся: количество слоёв, размерность ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, функция по уменьшению размерности (выбор максимума, среднего и т.п.), передаточная функция нейронов. Важно понимать, что все эти параметры существенно влияют на результат.

2) Рекуррентные нейронные сети (Recurrent Neural Network, RNN) — вид нейронных сетей, характеризующийся наличием обратных связей. При этом под обратной связью подразумевается связь от логически более удалённого элемента к менее удалённому. Благодаря наличию обратных

связей, НС способна запоминать и воспроизводить целые последовательности реакций. Возможно рассмотрение RNN с различных сторон: с точки зрения программирования в таких сетях появляется аналог циклического выполнения, а с точки зрения систем, такая сеть эквивалентна конечному автомату.

Данная архитектура похожа на сеть прямого распространения, но со смещением времени. Информация к нейронам приходит не только с предыдущих слоев, но и от самих себя с предыдущей итерации. К сожалению, информация в этих сетях со временем теряется со скоростью, зависящей от активационных функций.

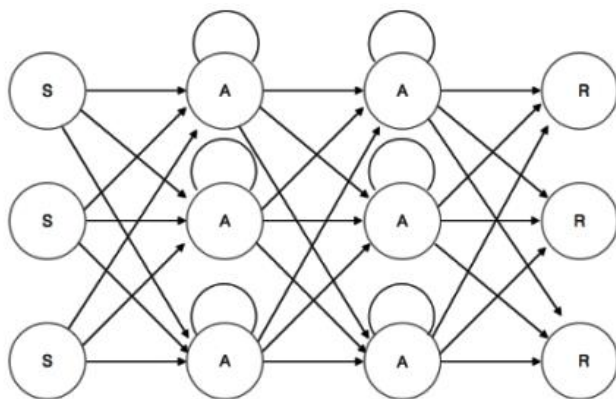


Рисунок 7. Архитектура рекуррентной нейронной сети.

Данную архитектуру можно использовать в самых различных случаях, ведь любые данные можно представить в виде последовательности.

Наиболее ярким примером применения рекуррентных НС является приложение для виртуальной клавиатуры SwiftKey. Новая SwiftKey Neural использует уже не просто шаблоны, а искусственные нейронные сети для предсказания и исправления ввода. НС представляют часть более широкой области машинного обучения и искусственного интеллекта, они основаны больше не на шаблонах, а на структурах и принципах работы человеческого мозга. Для SwiftKey и миллионов его пользователей, нейронные сети должны стать способом печатать более быстро и

аккуратно, независимо от того, что уже система знает о конкретном пользователе или других текстах.

Трудность рекуррентной сети заключается в том, что если учитывать каждый шаг времени, то становится необходимым для каждого шага времени создавать свой слой нейронов, очевидно, что это вызывает серьёзные вычислительные сложности. Кроме того, многослойные реализации оказываются вычислительно неустойчивыми, так как в них как правило исчезают или «зашкаливают» веса. Если ограничить расчёт фиксированным временным окном, то полученные модели не будут отражать долгосрочных трендов.

### 3) Комбинированные нейронные сети

Комбинированные нейронные сети объединяют возможности разных типов НС. Рассмотрим некоторые приложения на базе комбинированных нейронных сетей:

- Skype внедрил НС для возможности синхронного перевода для нескольких языков, среди которых есть русский и японский — одни из самых сложных в мире.
- Яндекс на базе нейронных сетей создал два поисковых алгоритма: «Палех» и «Королев». Первый помогает найти максимально релевантные сайты для низкочастотных запросов путем изучения заголовков страниц и сопоставления их смысла со смыслом запросов. На основе «Палеха» появился алгоритм «Королев». Этот алгоритм оценивает не только заголовок, но и весь текстовый контент страницы.
- Google Inbox с помощью нейросетей осуществляется ответ на сообщение. Сеть изучает переписку и генерирует возможные варианты ответа.
- YouTube использует нейронные сети для ранжирования роликов, причём сразу по двум принципам: одна нейронная сеть изучает

ролики и реакции аудитории на них, другая проводит исследование пользователей и их предпочтений.

- Facebook активно работает над DeepText AI — программой для коммуникаций, которая понимает жаргон и ненормативную лексику.

## 2.4. Области применения нейронных сетей

Нейронные сети применимы практически в любой области жизни. Среди основных областей применения нейронных сетей — прогнозирование, принятие решений, управление, распознавание образов, оптимизация, анализ данных. Рассмотрим некоторые из них подробнее.

1) Анализ данных. Данная область занимается построением и исследованием наиболее общих математических методов и вычислительных алгоритмов извлечения знаний из экспериментальных данных. Можно сказать, что это процесс исследования, фильтрации, преобразования и моделирования данных, целью которого является извлечение полезной информации и принятие решений.

Автоматическое распознавание речи (АРР) для центра управления запуском NASA. Передача голосовых сообщений в центр управления запуском NASA предельно важна, однако, автоматическое распознавание речи довольно сложный процесс из-за нескольких факторов: отсутствие данных для обучения, незнакомые слова в аббревиатурах, несколько разных ораторов и акцентов, а также разговорные характеристики речи. Для решения данной проблемы были использованы двунаправленные глубокие рекуррентные нейронные сети для обучения и тестирования производительности распознавания речи. Данные расширенные и пользовательские языковые модели могут улучшить точность распознавания речи. Транскрипция сообщений из центра управления запуском поможет машине проанализировать информацию и ускорить создание знаний.

Автоматическое распознавание речи имеет три основных приложения, включая устройства ввода/вывода, средства связи и поиск информации. Эффективность АРР была значительно улучшена за счет использования глубоких нейронных сетей, в основном для устройств ввода/вывода. Ведущие отрасли, такие как Amazon, Apple, Baidu, Google, Microsoft и IBM развиваются, чтобы помочь общественности легко



использовать свои продукты посредством автоматического распознавания речи.

Для обучения и тестирования производительности распознавания речи используются двунаправленные рекуррентные нейронные сети. Но обычные RNN все еще имеют существенные практические проблемы, вызванные экспоненциальным распадом градиентного спуска, который препятствует обучению долгосрочных отношений между словами. LSTM — это особый тип рекуррентной нейронной сети, которая может изучать долгосрочные зависимости посредством выборочной консолидации памяти.

2) Управление. НС производит расчет входного воздействия, при котором система будет следовать желаемой траектории или соответствовать требуемым параметрам.

Моделирование и управление с использованием модели на основе нейронной сети Predictive Control. Одной из основных целей в отрасли является снижение эксплуатационных расходов. Это подразумевает улучшение качества конечного продукта, а также более эффективное использование энергетических ресурсов. Усовершенствованные системы управления призваны справиться с этими требованиями. В последние годы использование нейронных сетей для идентификации нелинейной системы оказалось чрезвычайно успешным. Целью использования НС Predictive Control является управление для обеспечения высококачественного контроля при наличии нелинейностей, а также лучшего понимания процесса проектирования при использовании новых технологий, то есть алгоритма управления нейронной сетью. Сочетание нейронных сетей и прогнозирующего контроля на основе моделей, по-видимому, является прекрасным выбором для достижения хороших результатов.

### **3. ПРОЕКТИРОВАНИЕ МОДЕЛИ СЕТИ УПРАВЛЕНИЯ ТРАФИКОМ**

#### **3.1. Управление трафиком сети передачи данных**

На сегодняшний день проведены обширные исследования для разработки алгоритмов и протоколов сетей связи для эффективного использования и управления ресурсами [20-25]. Существует множество проблем, связанных с усложнением телекоммуникационных технологий. Например, трудности встречаются при моделировании, прогнозировании и контроле трафика в сетях передачи данных (СПД).

С целью максимизации эффективности управления трафика данных, необходимо решить проблему маршрутизации трафика по кратчайшим путям (например, Open Shortest Path First (OSPF)) или через несколько доступных (например, Valiant Load Balancing (VLB)). Однако ни одно из этих решений не является оптимальным. Более эффективный способ решения можно разработать при использовании точных математических моделей для сетей связи, взяв во внимание определенные требования пользователей. Необходимо помнить, что данная модель может не сработать для тех сетевых проблем, которые связаны с задержкой и маршрутизацией с несколькими переходами. Среди причин этого можно отметить следующие:

а) Очереди. Сеть связи с сетчатой топологией представляет собой довольно сложную топологию с многоточечной сетью очередей, где пакеты данных от очереди могут быть распределены по нескольким очередям, расположенным ниже по потоку, а также очередь может принимать пакеты из нескольких разных очередей.

б) Теория массового обслуживания способна давать только точные оценки задержке в очередях, однако в сложной сети связи данный процесс может не выполняться.

А также для повышения результативности управления трафиком в сети передачи данных, необходимо принимать и обрабатывать

информацию о трафике в режиме реального времени и динамически корректировать сигналы. С точки зрения динамической корректировки длительности сигналов трафика, существующие программы либо разделяют сигнал трафика на равную продолжительность, либо извлекают ограниченную информацию о трафике из реальных данных. Управление трафиком похоже на управление перекрестком, основываясь на данном сравнении, разберём проблему контроля подробнее.

При определённых условиях (входных данных, состоянии окружающей среды) существующие системы управления светофором работают исправно и достаточно эффективно. Однако в ситуациях высокого потока транспорта, например в час пик, или же в период проведения спортивных мероприятий, системы контроля светофора становятся парализованными. Так и в сетях передачи данных, при высоком потоке данных системы управления могут не справляться со своими задачами. Данный пример мотивирует создать систему управления потоком, которая в режиме реального времени способна обрабатывать входные данные и управлять движением подобно регулировщику.

Работы по изучению искусственных нейронных сетей привнесли методы, обеспечивающие эффективное управление трафиком (маршрутизации, предсказания тренда трафика и пр.) в сетях передачи данных [26-36].

В классическом виде задачи маршрутизации решаются с помощью нейронных сетей на основе задачи коммивояжера, в таких задачах используется модель нейронной сети Хопфилда. Такую НС можно успешно применять на сетях связи с коммутацией каналов, однако сеть Хопфилда имеет ряд ограничений, в частности по числу запоминаемых образцов, по появлению перекрестных ассоциаций (в случае, когда два образца подаваемые в сеть сильно похожи), а также ввиду стабилизации в локальном, а не в глобальном минимуме. Таких недостатков лишена нейронная сеть Хэмминга. Ее используют в случаях, где нет

необходимости получать от НС явный образец, а требуется лишь номер образца. Сеть Хэмминга затрачивает меньше памяти и объема вычислений.

Сети передачи данных на основе стека протоколов TCP/IP имеют особенность ускорения и оптимизации громоздких блоков данных при передаче данных и методику управления потоком. Такие функции выполняет протокол TCP и метод называется методом плавающего окна. Он дает возможность пересылать блоки данных без подтверждения. В случае, когда есть возможность заранее иметь информацию о перегруженности буфера или же об увеличивающихся задержках, качественно увеличивается управление сетью в особенностях пресечения потери дейтаграмм. Такой возможностью располагает нейронная сеть Хэмминга, позволяющая прогнозировать тенденцию трафика, используя собранные статистические данные.

Также в задачу предсказания тренда трафика входит предсказание самоподобного трафика, что позволяет решать вопросы качества обслуживания и долгосрочных прогнозов загрузки каналов [34, 35]. Особенностью такой задачи является факт того, что сетевой трафик это самоподобный случайный процесс, который можно назвать детерминированным хаосом. Однако решение такой задачи можно свести к прогнозу временного ряда, с чем могут справиться нейронные сети, способные к обобщению. Вариантом модели нейронной сети для такой проблемы может быть многослойный персептрон, который имеет возможность обобщать информацию, строить нелинейные отображения и способен к адаптации.

Глубокое обучение с подкреплением (Deep Reinforcement Learning, DRL) возможно использовать для решения проблемы распределения ресурсов в сложных сетях связи, которые обычно характеризуются огромными или непрерывными пространствами действий. Метод Deep Reinforcement Learning считается особенно перспективным для управления так как способен иметь дело с изменяющимися со временем состоянием

системы и пользовательскими требованиями и обрабатывает сложные состояния.

### 3.2. Моделирование структуры сети

Для того чтобы продемонстрировать сценарий контроля потоком трафика, вновь воспользуемся сравнением с управлением дорожным перекрестком. Модель представлена на рисунке 8. Левая часть демонстрирует сбор информации о дорожном движении через транспортную сеть. Анализируется текущее состояние трафика и выбирается действие, используя глубокую нейронную сеть, которая изображена справа.

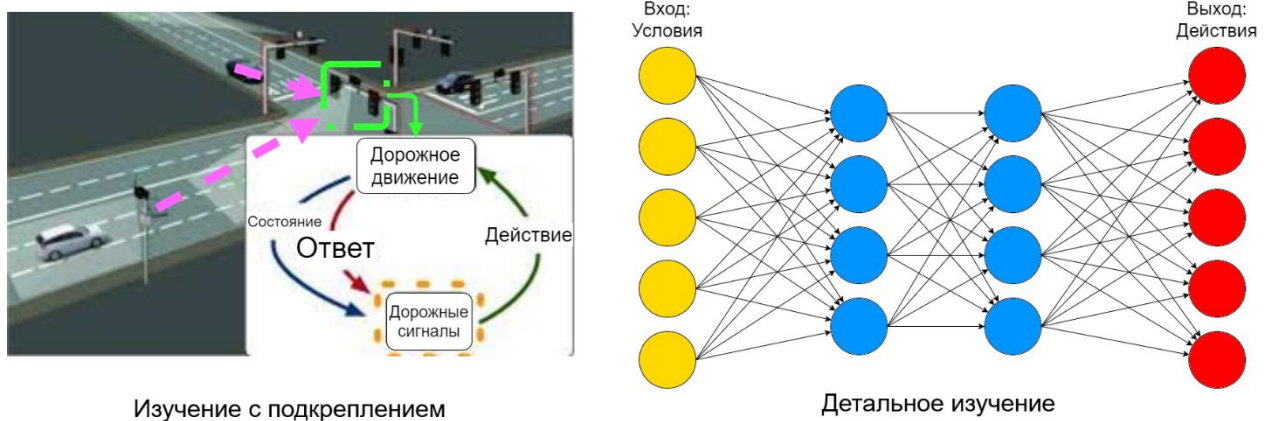


Рисунок 8. Глубокое обучение с подкреплением.

Состояние системы определяется на основе двух частей информации: положение и скорость транспортных средств на перекрёстке. Светофор может извлечь виртуальный образ текущего пересечения, который разделен на маленькие квадраты одинакового размера (рисунок 9). Размер ячейки должен гарантировать следующее:

- одно транспортное средство может быть размещено в одном квадрате;
- не может быть размещено два транспортных средства в одной и той же ячейке.

Эти условия введены для уменьшения вычислений. В каждой ячейке значение состояния является комбинацией значений «позиция» и «скорость» одного автомобиля.

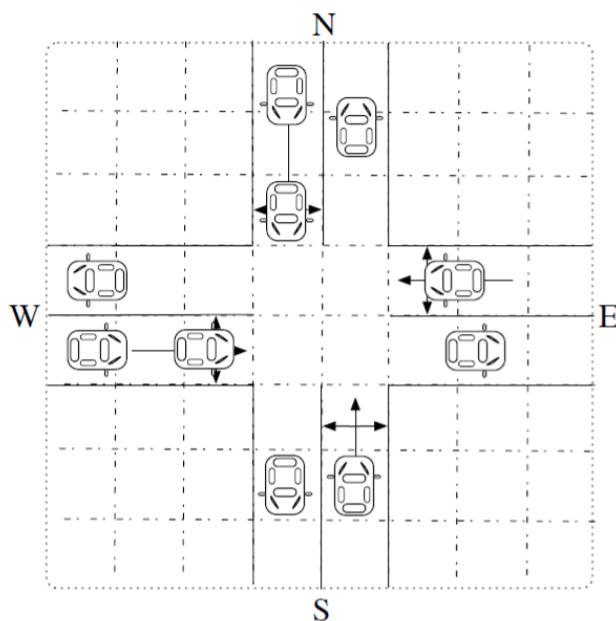


Рисунок 9. Состояние трафика на перекрёстке.

Матрица положения (Таблица 4) имеет такой же размер сети. Размерность позиции является двоичным значением. Если в ячейке есть транспортное средство, то значение равно 1; в противоположном случае оно равно 0. Размер скорости представляет собой целочисленное значение, обозначающее текущую скорость автомобиля.

Таблица 4. Матрица положения транспортных средств

			1.0				
				1.0			
			1.0				
1.0					1.0		
1.0		1.0				1.0	
			1.0	1.0			

Основываясь на состоянии трафика, система управления должна выбрать подходящее действие для регулирования потоков данных. Как

известно, существует три световых сигнала: зелёный, жёлтый и красный. Эти сигналы являются рычагами управления. Статус – одна из комбинаций сигналов светофора. Продолжительность пребывания в одном статусе называется фазой. Все фазы циклически изменяются в фиксированной последовательности для направления транспортных средств при прохождении перекрестка. Повторение фазы один раз является циклом. Последовательность фаз в цикле фиксирована, но продолжительность каждой фазы является переменчивой. Если один этап необходимо пропустить, то его продолжительность можно установить равной 0 секунд. Возможна корректировка продолжительности на каждом этапе для решения различных транспортных ситуаций на перекрестке. Проблема заключается в том, как оптимизировать эффективность использования перекрестка путем динамического изменения длительности каждой фазы светофора, опираясь на историю данных. Обучение нейронной сети с подкреплением – способ освободить регулировщика из процесса управления потоком трафика.

### **3.3. Основные параметры модели управления**

Для реализации модели управления трафиком в СПД необходимы следующие условия.

1) Наличие «глаз», просматривающих состояние сети в режиме реального времени. В ситуации с управлением перекрестком такими «глазами» являются камеры, а также датчики, например, подземные индуктивные петлевые детекторы для обнаружения наличия транспортных средств перед светофорами. С помощью данных устройств можем определить нахождение автомобиля и его скорость. Что касается управления трафиком сети передачи данных, то необходимо следить за пропускной способностью сети, емкостью канала и т.п. В список параметров, выполняющих функции «глаза» входят:

а) Задержка. Выделяют три типа задержек: задержка обработки,

очереди, распространения. Данный параметр отражает затраченное время на обработку, постановку и нахождение в очереди, а также доставку пакета.

б) Очередь. Данный параметр позволит определить количество и размер пакетов, которые уже находятся в очереди, а также величину полосы пропускания интерфейса.

в) Потери. Потеря пакетов происходит при условии переполнения буфера маршрутизатора. С помощью данного параметра можно определить недостаток мощности маршрутизатора, переполнение буфера, наличие аппаратных ошибок во фреймах.

г) Нагрузка. Используя количество пакетов данных вместе со значением среднего времени обработки, можно вычислить количество трафика в час наибольшей нагрузки и использовать полученное значение для анализа трафика.

д) Маршрутизация. Основной целью маршрутизации является обеспечение наилучшего пути следования информации с точки зрения ее минимально возможной задержки и максимальной пропускной способности сети. Также данный параметр позволяет обеспечивать защиту и надежность передачи информации.

е) Метрика. Метрика представляет собой оценку качества связи в данной сети. С помощью этого параметра можно оценить пропускную способность, надежность и задержку.

Следовательно, используя перечисленные выше параметры, можем анализировать и производить контроль над задержками обработки, очереди и распространения, над количеством передаваемых пакетов, очередями в сети, над выбором наилучшего пути следования данных.

2) Наличие «мозга», который будет обрабатывать полученную информацию. Примером «мозга», используемого при управлении трафиком СПД, является нейронная сеть Deep Reinforcement Learning. НС DRL управляет маршрутизацией информации по-разному в зависимости от



сетевой архитектуры. В рамках концепции SDN (Software-Defined Networking) применимо централизованное управление маршрутизацией. Модель сети SDN делится на три уровня: уровень передачи данных (коммутаторы SDN), уровень управления (контроллеры SDN) и уровень сетевых приложений [37]. В данном случае интересует уровень управления, на котором находятся управляющие устройства – контроллеры. На контроллерах организованы все жизненно важные функции для жизнедеятельности сети, такие как: функции по управлению трафиком, мониторинг состояния сети в целом. Данные, полученные от нейронной сети, попадают на контроллер SDN, с помощью которого организовывается централизованная маршрутизация, следовательно, в данном типе архитектуры используется один рычаг управления. В традиционной архитектуре сети рычаг управления используется на каждом устройстве, используется децентрализованное управление маршрутизацией.

3) Наличие рычагов управления. Примером в управлении перекрестком являются зелёный, жёлтый и красный цвета светофора. В ситуации с управлением трафика СПД в роли «рычага управления» выступают:

а) Полоса пропускания. Установление диапазона частот для каждого сигнала позволяет контролировать требования, предъявляемые качеству.

б) Пропускная способность. Данный параметр способен устанавливать максимально возможную скорость передачи данных, которая может быть достигнута на данном участке сети. Управление скоростью расширяет возможности обеспечения требуемого качества на основе организации очередей при значительном повышении эффективности управления и детализации контроля за входными данными.

в) Приоритет потоков. Задавая приоритет трафика, появляется возможность контролировать, чтобы ни один пакет из обычной очереди не был передан при существовании более приоритетной очереди.

В сети передачи данных с помощью определенных параметров сети можно узнать о реальном состоянии сети. На основе данных о состоянии этих параметров можно предположить, существуют ли проблемы с управлением трафиком на данный момент или нет. Конкретной сетью или сценарием можно управлять, изменяя параметры сети.

Для того чтобы использовать Deep Reinforcement Learning, необходимо спроектировать пространство состояний и пространство действий:

- Пространство состояний (State Space): состояние содержит два компонента: пропускную способность и задержку каждого сеанса связи. Формально вектор пространства состояний можно представить:

$$s = [(x_1, z_1), \dots, (x_i, z_i), \dots, (x_k, z_k)] \quad (8)$$

- Пространство действий (Action Space): действие определяется как решение проблемы, то есть набор коэффициентов разделения для сеансов связи. Формально вектор пространства действия можно представить:

$$a = [w_1, 1, \dots, w_{kj}, \dots, w_k, |P_k|] \quad (9)$$

Далее представлен предлагаемый контроль над управлением трафиком на основе DRL. Ядро предлагаемой структуры управления является агентом, который запускает алгоритм «Actor-Critic» [38]:

Шаг 01. Случайная инициализация сети Critic  $Q$  и сети Actor  $\pi$ .

Шаг 02. Инициализация приоритетного буфера воспроизведения  $B$ .

Шаг 03. Получение начального наблюдаемого состояния  $S_1$ .

Шаг 04. Применение метода разведки, соответствующего проблеме.

Шаг 05. Сохранение переходного образца  $(st, at, rt, st+1)$ .

с максимальным приоритетом  $pt = \max_j < t pj$ .

Шаг 06. Приоритетная выборка перехода  $(si, ai, ri, si+1)$ .

Шаг 07. Вычисление веса важной выборки  $w = (|B| \cdot P(i)) / \max w$ .

Шаг 08. Вычисление TD-ошибки  $\delta_i = y_i - Q(si, ai)$ .

Шаг 09. Вычисление градиента.

Шаг 10. Обновление приоритета перехода.

Шаг 11. Накопление изменения веса для сети Critic:  $\Delta \theta Q := \Delta \theta Q + w$ .

Шаг 12. Накопление изменения веса для сети Actor:  $\Delta \theta \pi := \Delta \theta \pi + w$ .

Шаг 13. Обновление веса сети Critic:  $\theta Q := \theta Q + \eta Q$ .

Шаг 14. Обновление веса сети Actor:  $\theta \pi := \theta \pi + \eta \pi$ .

Шаг 15. Обновление веса цели сети Critic:  $\theta Q := \tau \theta Q + (1 - \tau) \theta Q$ .

Шаг 16. Обновление веса цели сети Actor:  $\theta \pi := \tau \theta \pi + (1 - \tau) \theta \pi$ .

Необходимо отметить, что дизайн пространства состояний и пространства действий имеет решающее значение для успеха метода Deep Reinforcement Learning. Ядро предлагаемой структуры управления является агентом, который запускает алгоритм DRL, чтобы найти лучшее действие для каждой решающей эпохи, а также наблюдает за состоянием сети. В качестве Actor в данном дизайне и реализации используется двухслойная нейронная сеть с прямыми связями, которая включает в себя 64 нейрона в первом слое и 32 нейрона во втором. Для сети Critic используется двухслойная НС с обратной связью, с 64-мя и 32-мя нейронами в первом и втором слое соответственно.

Существует множество проблем, которые напрямую связаны с усложнением телекоммуникационных технологий. Трудности встречаются при моделировании, прогнозировании и контроле трафика в сетях передачи данных. Для решения проблемы управления трафиком в сетях передачи данных предложено применение модели нейронной сети, основанной на глубоком обучении с подкреплением.

DRL демонстрирует превосходную способность хорошо контролировать сеть связи, грамотно распределять ресурсы, а также обеспечивает высокую пропускную способность. На основании параметров, нейронной сети и «рычагов» управления сетью связи, возможно создание модели для управления трафиком в сети передачи данных.

### 3.4. Инструменты для проектирования

#### 3.4.1. Библиотеки нейронных сетей

Существует большое количество библиотек и программных решений для машинного обучения, построения и исследования нейронных сетей.

Рассмотрим некоторые из них:

##### 1. Theano [39].

Theano — это расширение языка Python, позволяющее эффективно вычислять математические выражения, содержащие многомерные массивы. Theano разработана в лаборатории LISA для поддержки быстрой разработки алгоритмов машинного обучения.

Библиотека поддерживается на операционных системах Windows, Linux и Mac OS. В состав Theano входит компилятор, который переводит математические выражения, написанные на языке Python в эффективный код на языках C++ или CUDA.

Theano предоставляет базовый набор инструментов для конфигурации нейросетей и их обучения. Возможна реализация многослойных полностью связанных сетей (Multi-Layer Perceptron), сверточных нейросетей, рекуррентных нейронных сетей, автокодировщиков и ограниченных машин Больцмана. Также предусмотрены различные функции активации, в частности, сигмоидальная, softmax-функция, кросс-энтропия.

##### 2. TensorFlow [40].

TensorFlow — открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия. Применяется как для исследований, так и для разработки собственных продуктов Google. Основное API для работы с библиотекой реализовано для Python, также существуют реализации для C++, Haskell, Java и Go.

Является продолжением закрытого проекта DistBelief. Изначально TensorFlow была разработана командой Google Brain для внутреннего использования в Google, в 2015 году система была переведена в свободный доступ с открытой лицензией Apache 2.0.

### 3. MATLAB [41].

Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

Программы, написанные на MATLAB, бывают двух типов — функции и скрипты. Функции имеют входные и выходные аргументы, а также собственное рабочее пространство для хранения промежуточных результатов вычислений и переменных. Скрипты же используют общее рабочее пространство. Как скрипты, так и функции сохраняются в виде текстовых файлов и компилируются в машинный код динамически. Существует также возможность сохранять так называемые pre-parsed программы — функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются быстрее обычных, особенно если функция содержит команды построения графиков.

Основной особенностью языка MATLAB являются его широкие возможности по работе с матрицами.

Для MATLAB имеется возможность создавать специальные наборы инструментов, расширяющие его функциональность. Наборы инструментов представляют собой коллекции функций, написанных на языке MATLAB для решения определённого класса задач. В данном случае понадобится инструмент Deep Learning Toolbox (ранее Neural Network Toolbox) [42].

Toolbox обеспечивает основу для проектирования и реализации глубоких нейронных сетей, используя как предварительно обученные модели, так и приложения и инструменты для проектирования разработки архитектур нейронных сетей. Toolbox предоставляет функции и приложения для моделирования сложных нелинейных систем, которые сложно описываются уравнениями, поддерживает обучение с учителем и прямым распространением, с радиальными базисными функциями и динамическими сетями. Также есть поддержка обучения без учителя с самоорганизующимися картами и конкурентными слоями. С данным инструментом вы можете создавать, обучать, визуализировать и моделировать нейронные сети. Toolbox можно использовать для таких задач, как аппроксимация данных, распознавание образов, кластеризация, прогноз временных рядов, моделирование динамических систем и их управление. Toolbox позволяет использовать методику переноса обучения, для уже обученных сетей и не начинать обучение сети с нуля, например можно использовать сети SqueezeNet, Inception-v3, ResNet-101, GoogLeNet, VGG-19 и т.д. или импортировать сеть из TensorFlow и Caffe.

### 3.4.2. Настройка программного обеспечения

Настройка Deep Learning Toolbox (Neural Network Toolbox) не представляет сложности для пользователя, библиотека готова к использованию уже после того, как MATLAB был установлен на ПК. Для интеграции данных будем использовать широко известный инструмент компании Microsoft – Excel. В нем составляется таблица с исходными данными для подачи на вход НС и ее обучения, а Toolbox позволяет загружать эти данные из файла Excel.

Чтобы начать создание НС в Toolbox нужно воспользоваться командой *nstart*. Выбираем кнопку *Fitting app* (рисунок 10).

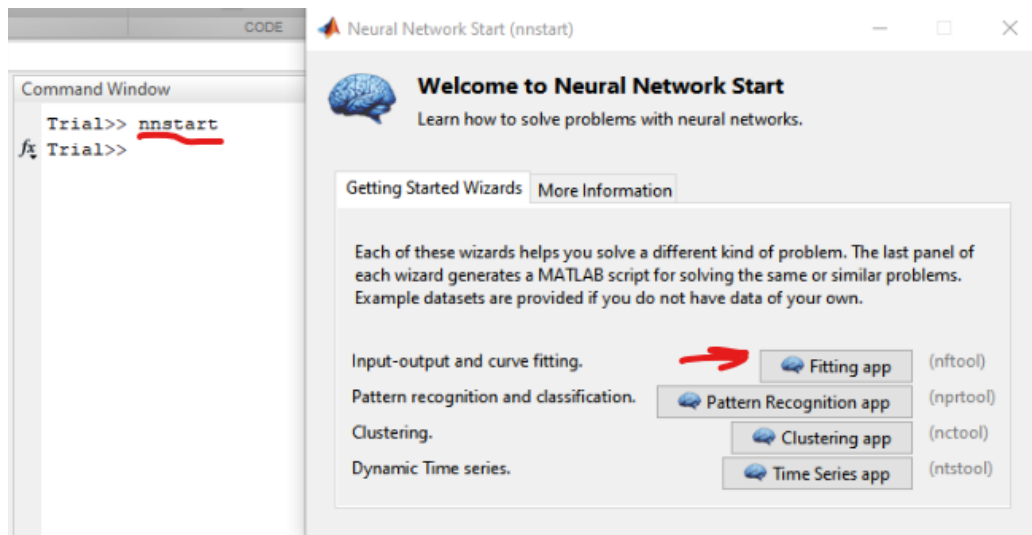


Рисунок 10. Стартовое окно.

Далее выбираем откуда следует брать входные данные и куда направить выход (Рисунок 11). А также, в каком виде подаются данные (строками или столбцами).

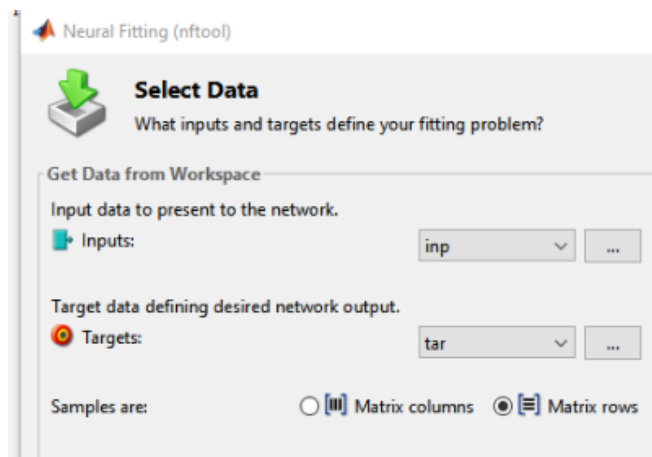


Рисунок 11. Выбор данных.

В окне, изображенном на рисунке 12, предстоит выбрать соотношение между обучающим (*Training*), проверочным (*Validation*) набором данных, которые используются, чтобы оценить обобщающие свойства сети и остановить обучение, когда обобщение прекращает улучшаться, и тестовым множеством (*Testing*), не оказывающим влияния на обучение, но служащим для проверки качества обучения на данных, которые не использовались в обучении сети.

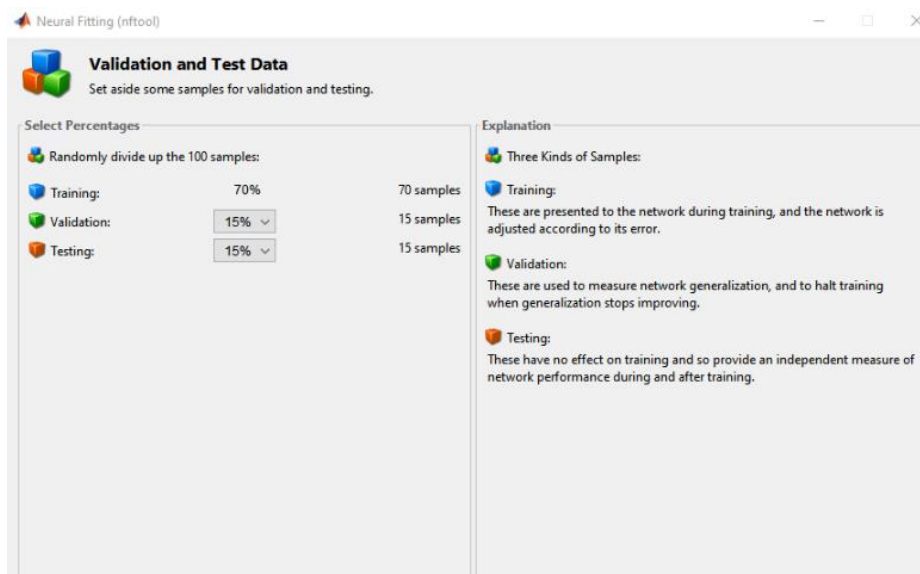


Рисунок 12. Выбор соотношения данных.

В следующем окне (Рисунок 13) представлена стандартная архитектура нейронной сети, которая по сути является многослойным персептроном с одним скрытым слоем и одним выходным слоем, сигмоидальной активационной функцией в первом слое и линейной функцией во втором. Здесь же можно задать количество нейронов в скрытом слое.

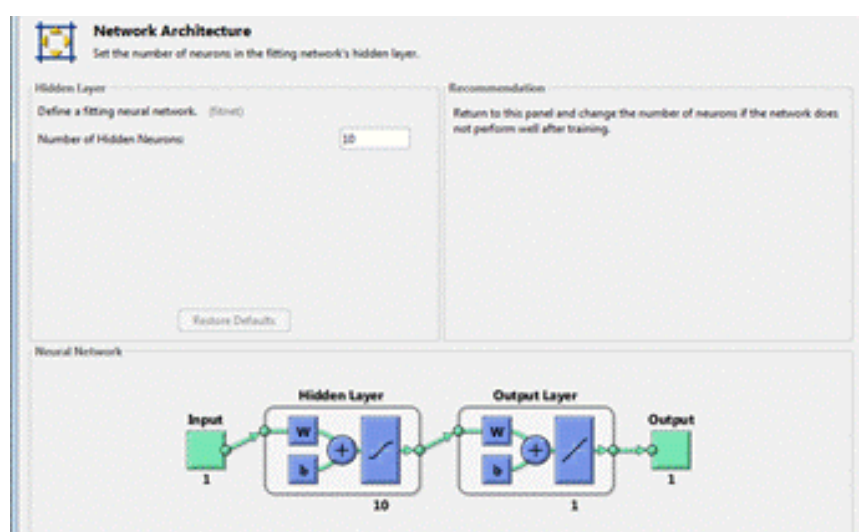


Рисунок 13. Архитектура НС.



Конечным этапом перед эксплуатацией нейронной сети является выбор метода обучения (Рисунок 14). В интерфейсе предоставлены только три метода обучения: Левенберга-Марквардта (Levenberg-Marquardt) [43-44], байесовской регуляризации (Bayesian Regularization) [45] и сопряженных градиентов (Scaled Conjugate Gradient) [46].

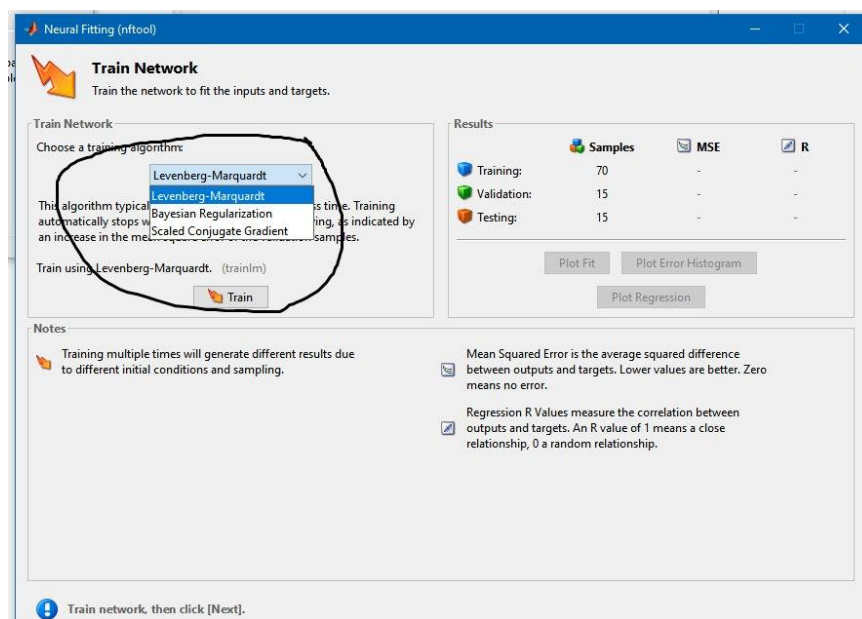


Рисунок 14. Выбор метода обучения.

### 3.5. Данные для обучения и их обработка

При подготовке данных для обучения нейронной сети необходимо обращать внимание на следующие существенные моменты.

- Количество наблюдений в наборе данных. Следует учитывать тот фактор, что чем больше размерность данных, тем больше времени потребуется для обучения сети.
- Обучающая выборка должна быть представительной (репрезентативной).
- Обучающая выборка не должна содержать противоречий, так как нейронная сеть однозначно сопоставляет выходные значения входным.

- Нейронная сеть работает только с числовыми входными данными, поэтому важным этапом при подготовке данных является преобразование и кодирование данных.
- На вход нейронной сети следует подавать значения из того диапазона, на котором она обучалась. Например, если при обучении нейронной сети на один из ее входов подавались значения от  $0$  до  $10$ , то при ее применении на вход следует подавать значения из этого же диапазона или близлежащие значения.

После того как данные собраны, их необходимо подготовить. Этот этап называется препроцессинг. Основная задача препроцессинга - отображение данных в формат пригодный для обучения модели. Можно выделить три основных манипуляции над данными на этапе препроцессинга.

1) Создание векторного пространства признаков, где будут жить примеры обучающей выборки. По сути, это процесс приведения всех данных в числовую форму. Это избавляет нас от категориальных, булевых и прочих не числовых типов.

2) Нормализация данных. Процесс, при котором мы добиваемся, например, того, чтобы среднее значение каждого признака по всем данным было нулевым, а дисперсия - единичной.

3) Изменение размерности векторного пространства. Если векторное пространство признаков слишком велико (миллионы признаков) или мало (менее десятка), то можно применить методы повышения или понижения размерности пространства:

- для повышения размерности можно использовать часть обучающей выборки как опорные точки, добавив в вектор признаков расстояние до этих точек - этот метод часто приводит к тому, что в пространствах более высокой размерности множества становятся линейно разделимыми, и это упрощает задачу классификации;

- для понижения размерности чаще всего используют метод главных компонент, основная задача которого - поиск новых линейных комбинаций признаков, вдоль которых максимизируется дисперсия значений проекций элементов обучающей выборки.

Для сбора «сырых» данных была использована статистика по 100 эхо запросов, где была отобрана информация о переданных в обоих направлениях пакетах за секунду и соответствующей задержки на конкретном интерфейсе, с которого были переданы пакеты. Получается, что нужная нам информация представлена в разных размерностях и в форме неудобной для понятия машиной и нейронной сетью. Чтобы это исправить следует использовать методы препроцессинга и постобработки. Ведь помимо того, что необходимо собрать сырые данные, сделать их понятными для человека, нужно сделать их понятным для машины и НС. Далее после работы нейронной сети необходимо предоставить выходные данные в понятный для человека вид.

Для предобработки количественных величин чаще всего применяют линейный сдвиг интервала значения признака, например, в интервал  $[-1, 1]$ . Формула пересчета значения признака  $x$  для  $i$ -го примера выборки в интервал  $[a, b]$  будет иметь вид [47]:

$$\tilde{x}_i = \frac{x_i - x_{\min}(b-a)}{x_{\max} - x_{\min}} + a, \quad (10)$$

где  $x_{\min}$  и  $x_{\max}$  - минимальное и максимальное выборочные значения признака.

Для улучшения работоспособности нейронной используем метод масштабирования на обучающей выборке в виде времени задержки. Выберем интервал  $[0; 1]$ . Значение  $x_{\max}$  будет 4000 мс, так как это время задержки, после которого пакет отбрасывается. Так же будем обработать данные подаваемые на вход сети, то есть количество переданных пакетов в секунду.

Далее для работы с нейронной сетью следует перевести данные в матричную форму:

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{1m} & \cdots & x_{nm} \end{pmatrix} \quad (11)$$

После того как получены результаты работы нейронной сети, выходные значения следует выразить обратно в исходную форму для репрезентативности. Для этого воспользуемся той же формулой сдвига, однако рассчитаем не  $\tilde{x}_i$ , а  $x_i$ :

$$x_i = \tilde{x}_i(x_{\max} - x_{\min}) + x_{\min}, \quad (12)$$

Процесс постобработки данных также включает в себя составление и анализ полученных данных в графической форме. В Toolbox автоматический препроцессинг данных после работы сети предлагает построить необходимые для анализа графики. Представленная выше вариация обработки данных универсальна, подходит для любого вида работ с нейронными сетями. Так как библиотека выполняет эти этапы в штатном режиме, далее не будем рассматривать их подробно, а перейдем к непосредственному эксперименту.

#### *Сбор данных.*

Использовалась программа *Wireshark* и утилита *ping*. Объектом *ping* стал Копенгагенский университет (<https://www.ku.dk>). Запускаем захват пакетов с помощью *Wireshark* и одновременно с этим утилиту *ping* для IP-адреса *130.226.237.173* на *100* запросов. Для моделирования реального режима работы сети параллельно с этим, запускаем *4k* видео на сайте *YouTube*. Получаем статистику, представленную на рисунке 15.

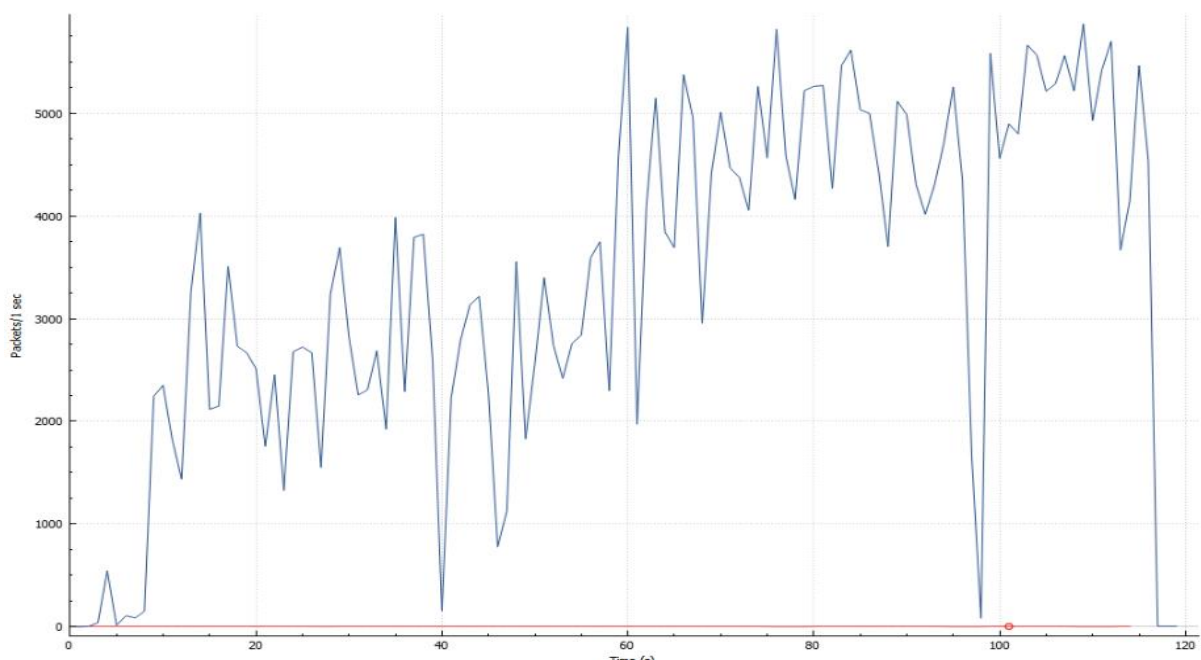


Рисунок 15. График полученной задержки.

Сопоставление данных о задержке в конкретную секунду и данных о количестве пакетов в данную секунду представлено в таблице 5.

Таблица 5. Данные для обучения.

№, с	Задержка, мс	Кол-во пакетов / с
1	28	1
2	27	3
3	33	38
4	27	545
5	27	12
6	26	105
7	26	80
8	30	150
9	27	2230
10	31	2351
11	29	1815
12	31	1430
13	34	3200
14	27	4030
15	29	2115
16	27	2135
17	27	3500

18	27	2750
19	29	2650
20	28	2520
21	28	1750
22	37	2460
23	35	1320
24	28	2650
25	28	2750
26	26	2650
27	27	1560
28	29	3180
29	28	3680
30	31	2820
31	28	2250
32	28	2300
33	38	2700
34	32	1920
35	31	4000
36	30	2300
37	30	3800
38	34	3820
39	26	2580

*Разработка модели.*

Выбор сети выполнен на основе испытаний с различными структурами многослойного перцептрона. Самая низкая ошибка соответствует 7 нейронам в скрытом слое. Следовательно, он выбирается как оптимальная архитектура ИНС. Выбранная сеть состоит из 4 нейронов во входном слое, 7 нейронов в скрытом слое и одного нейрона в выходном слое. Архитектура ИНС, используемая в данной ситуации, представлена на рисунке 16. Алгоритм обучения, используемый в данном моделировании, является алгоритмом обратного распространения. Перед тренировкой веса процесс инициализируются небольшими случайными числами. Веса корректируются до тех пор, пока ошибка не будет сведена к минимуму для

всех наборов тренировок. Когда ошибка для всего набора приемлемо низкая, обучение прекращается.

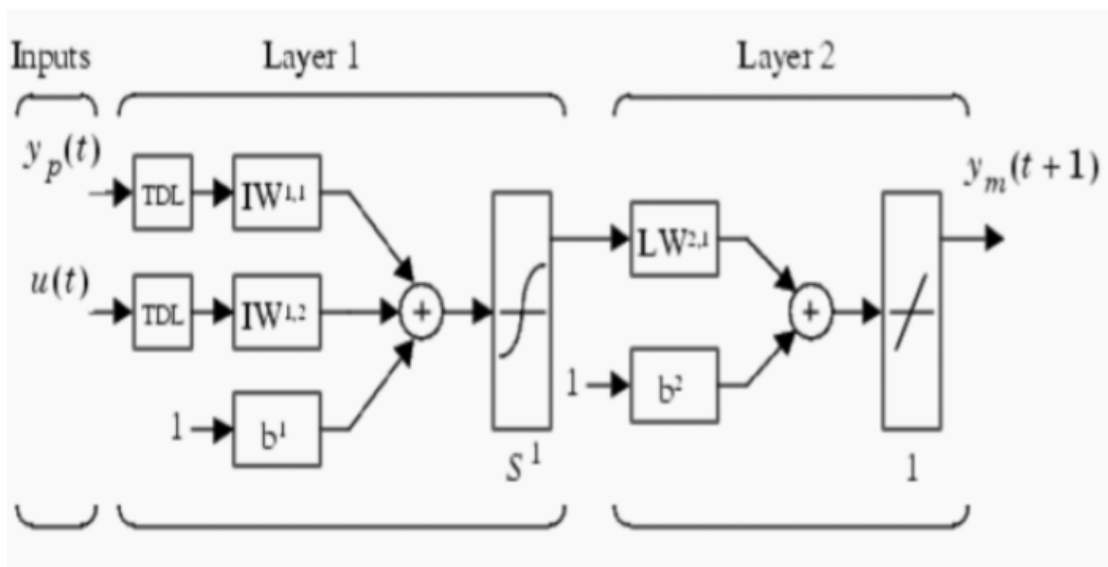


Рисунок 16. Архитектура ИНС

В таблице 6 продемонстрированы параметры, используемые при разработке модели ИНС.

Таблица 6. Параметры, используемые при моделировании

№	Параметры	Значения
1	Входные нейроны	4
2	Выходные нейроны	1
3	Скрытые слои нейронов	7
4	Активационная функция	Сигмоидальная
5	Алгоритм обучения	Алгоритм Левенберга-Марквардта
6	Архитектура	Прогнозирование
7	Начальные веса	1

Последним шагом в разработке модели является проверка модели, выполняющаяся путем оценки производительности модели с использованием подготовленных числовых и текстовых данных. Сеть обучена с использованием алгоритма Левенберга-Марквардта.

### Принятие решений.

Процесс, результатом которого является выбор мнения или направление действий среди нескольких альтернативных возможностей. Каждый процесс принятия решения производит окончательный выбор, за которым остаётся право возбуждать или не возбуждать действие.

*Задача выбора модели нейронной сети для системы принятия решений.* Наиболее перспективная область применения НС - это система принятия решений при управлении различными техническими средствами и технологическими процессами. В ней формируются наборы управляющих воздействий, минимизирующих отклонение реальной траектории системы от желаемой, основываясь на базе анализа прошлых управляющих неконтролируемых воздействий и отклонений траектории системы. Дальнейший анализ объектов управления позволяет выявить два вида системы управления с использованием модели нейронной сети: управление параметрами системы и управление реакцией системы (Рисунок 17).

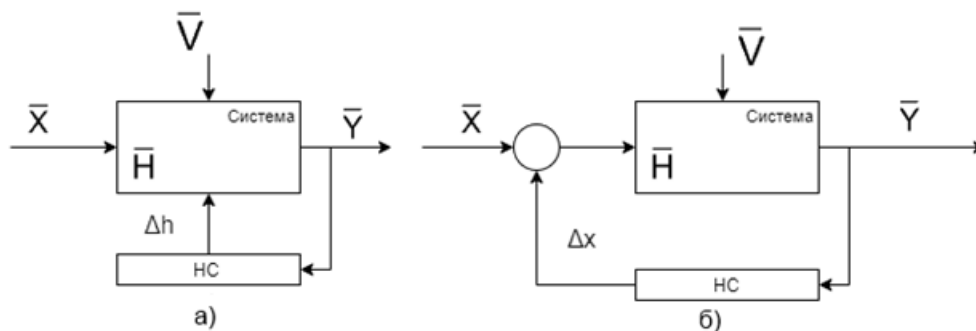


Рисунок 17. Исследуемые схемы управления

Таким образом, модель НС может корректировать как внутренние параметры самого объекта, так и управляющие воздействия.

С увеличением предлагаемых услуг и сетей доступа эффективное управление доступными радиоресурсами становится решающим фактором в обеспечении стабильности сети и предоставлении необходимого QoS. Распределение ресурсов является сложной задачей, так как ресурсы всегда



находятся в дефиците в беспроводной среде. В [48] предлагается сочетание нечетких и нейронных сетей, образующих гибридную нечеткую нейронную сеть, которая используется для управления радиоресурсами. Данный метод демонстрирует лучшую эффективность, что приводит к повышению удовлетворения пользователей. Модель разработана с использованием нейронной сети, основанной на повторяющейся радиально-базисной функции (Recurrent Radial Basis Function Networks, RRBFN). Данная НС сочетает в себе лучшее обучение и адаптивность, которая может быть использована в системе для обработки входящего трафика. Предлагаемая модель способна обрабатывать три типа приложений, включая в себя разговорный трафик, интерактивный трафик и фоновый трафик. Весь этот трафик представляет собой различные классы со следующими параметрами QoS.

1) Разговорный трафик чувствителен к задержкам передачи и дрожанию. Он требует гарантированной скорости передачи данных и низкой битовой ошибки. Примеры приложений, принадлежащих данной категории - видеоконференции и аудиоконференции.

2) Интерактивный трафик - это класс QoS, который не чувствителен к задержкам и дрожанию, но требует низкой скорости передачи. Примерами являются: просмотр веб-страниц, интерактивные чаты и игры.

3) Класс QoS с фоновым трафиком не является чувствительным к задержке передачи и дрожанию, не зависит от гарантированной скорости передачи. Примеры, принадлежащие этой группе - электронная почта, SMS-приложения.

Предлагаемая архитектура сети представлена на рисунке 18.

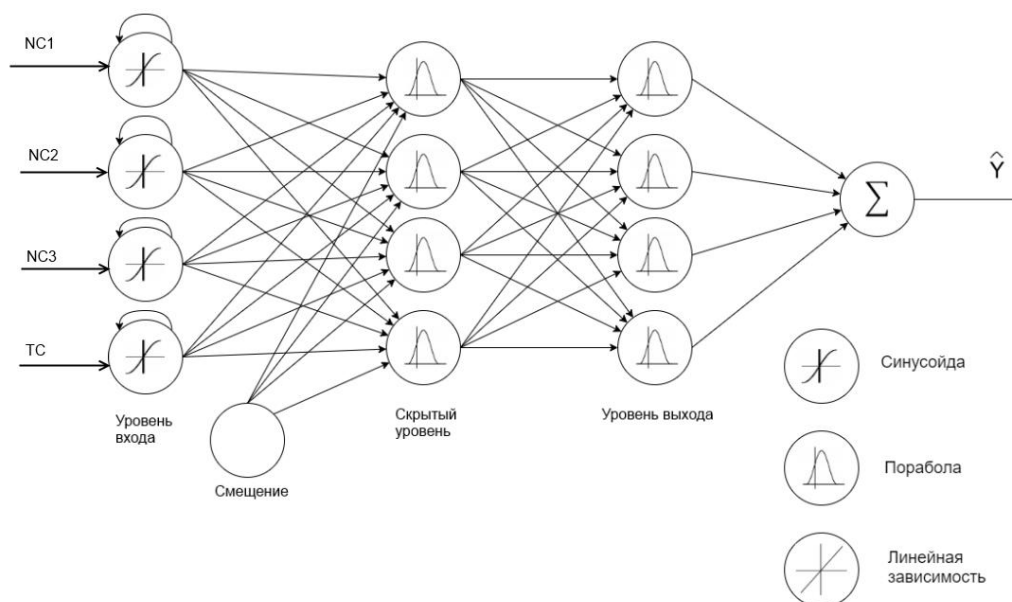


Рисунок 18. Модель RRBFN сети.

Требования входящего трафика принимаются как входы. Управление доступом к сети включает в себя обучение и тестирование контроллера на основе RRBFN. Тренировка и тестовые образцы выбираются произвольно. Сеть RRBFN имеет четыре уровня: вход, два скрытых слоя и выходной слой. Диапазон возвращаемых весов -  $[-1, 1]$ . Функциональный и выходной слой имеет один нейрон с линейной активацией.

#### *Прогнозирование.*

НС характеризуется способностью к обобщению и выделению скрытых зависимостей между входными и выходными данными. После обучения НС способна предсказать будущее значение некой последовательности на основе нескольких предыдущих значений и (или) каких-то существующих в настоящий момент факторов.

1. Создание надежных моделей прогнозирования. Предсказанию работоспособности компонентов в сложных динамических системах создают многочисленные затруднения. Главной целью интеллектуальных систем является использование данных, полученных от различных датчиков для предсказания поведения. Точные и надежные прогнозы имеют большой потенциал для предотвращения непредвиденного отказа

оборудования и повышения производительности. Одной из сложностей является сбор данных с нескольких сотен датчиков и других расчетных значений. Например, в автомобиле можем предсказать износ системы двигателя, используя сигналы от различных датчиков, таких как крутящийся момент, давление, температура и скорость.

Проблемы, которые должны быть решены моделью прогнозирования:

- Модель должна быть устойчива к отсутствию атрибутов, т.е. условие «застревание в нуле».
- Модель должна быть устойчива к шумовым признакам.

В НС используются алгоритмы машинного обучения, основанные на биологической нервной системе и способные идентифицировать сложные нелинейные отношения. Информация обрабатывается с использованием набора высоко взаимосвязанных узлов, также называемыми нейронами. Сеть взвешенных узлов складывается в несколько слоев. Это может эффективно ограничивать распространение сигнала на следующие слои. Таким образом, усиливается или блокируется активация узлов сети. Этот процесс сопоставим с выбором функции.

Следовательно, при выборе встроенных функций и способности идентифицировать различные шаблоны с минимальной предварительной обработкой, НС являются идеальным вариантом для данной задачи.

НС активно используются во многих областях, таких как прогнозирование медицинских результатов, экологические проблемы, прогнозы акций рыночного индекса и износ машин.

2. Понимание прогнозирования модели НС с точки зрения сетевого потока. Рассмотрим механизм прямой нейронной сети с точки зрения сетевого потока, который состоит из множества направленных классов-путей. Входная информация распространяется по слоям вдоль «пути» от входных узлов (нейронов во входном слое) к определенному выходному узлу (один нейрон в выходном слое). В данном случае рассматривается

каждый нейрон в сети как узел. Нейроны во входном слое рассматриваются как начальные узлы, а нейроны на выходном уровне - выходные узлы. Выходные узлы представляют собой метки классов для НС-классификаторов. С этой точки зрения обучение НС-классификатору рассматривается как процесс формулировки классовых путей для разных классов. В соответствии с основными идеями существующих алгоритмов, возможно классифицировать их по двум категориям: визуализация функции и визуализация класса. Первая категория фокусируется на визуализирующих функциях, которые узнала сеть, вторая направлена на выделение важных областей в заданном входном изображении для определенного класса.

3. Визуализация полученных функций. Данная категория представлена, например, для измерения чувствительности точности классификации для небольших вариаций значений пикселей изображения. Используется технология Deconvolutional Network (deconvnet) для визуализации промежуточных функциональных слоев, для уменьшения размеров SIFT, для сжатия дескрипторов уровня изображения. Такие подходы могут удалять избыточную информацию, требующую лишь нескольких бит для точки интереса. Тем не менее, данные методы требуют выделения признаков и векторного квантования, что связано с большими вычислительными затратами для мобильных устройств с ограниченной памятью. Кроме того, отправка сжатых локальных дескрипторов в облачный терминал способна обеспечить доставку запросов с низкой скоростью передачи битов, но также теряет глобальные функции, такие как цвета, текстуры, формы объектов и пространственные отношения между локальными функциями.

Можно представить, будто каждый нейрон аналогичен контейнеру для воды, связи между нейронами аналогичны трубопроводам, а значения весов соединения точно такие же, как у разных размеров трубопроводов. Чем больше размер трубопровода, тем больше объем воды через

трубопровод. Положительное значение весов соединения аналогично тому, что вода течет в контейнер. Отрицательное значение весов аналогично тому, что вода вытекает из контейнера. После форвардного вычисления НС-классификатора, как вода, втекающая в или из контейнеров сверху вниз по трубопроводам, возможно изменить размер трубопровода слой за слоем, чтобы изменять направления потока воды, которые образуют различные пути потока из верхних контейнеров в нижние контейнеры. Обучение НС-классификатору может рассматриваться как процесс формулировки конкретного пути для каждого класса (класс-путь). Каждый class-pathway (информация, поступающая из контейнера во входном слое в конкретный контейнер на выходе слой) соответствует классу, а класс-путь представлен конкретным активированным нейронным узлом в каждом слое и их соединениях. Как показано на рисунке 19, два класса-пути для двух классов, сформированные после обучения.

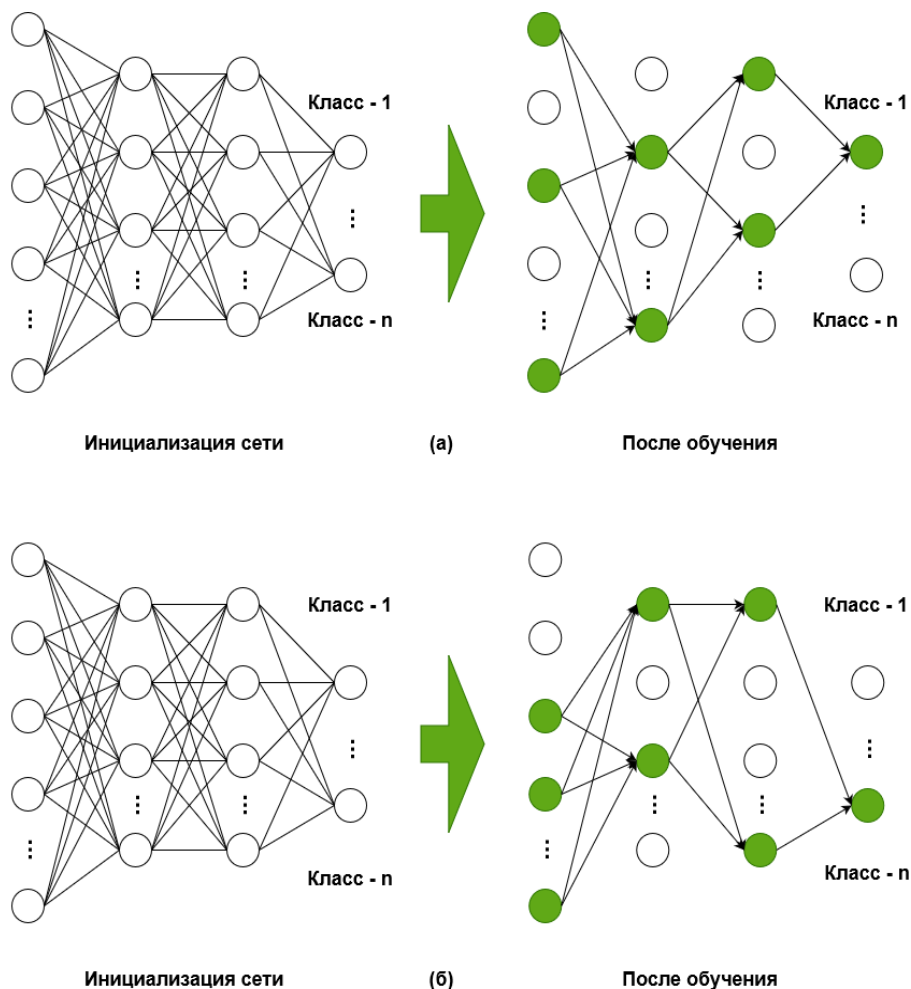


Рисунок 19. Два класса-пути после обучения.

Во-первых, необходимо взять каждый нейрон как нейронный узел и принять нейронные соединения в качестве однонаправленного пути. Затем нейронная сеть может быть аналогией ориентированного графика, включающего начальные узлы, скрытые узлы и выходные узлы, которые распространяют информацию от входных узлов на выходные узлы. В-третьих, рассматриваем направленную искусственную нейронную сеть как модель сетевого потока, которая состоит из множества направленных классов-путей.

В ходе изучения искусственных нейронных сетей проведен анализ возможностей использования типов архитектур НС в различных классах задач, результаты сравнения приведены в таблице 7 (знак «+» означают возможность применения нейронной сети данного типа к решению соответствующей задачи).

Таблица 7. Сравнение различных типов нейронных сетей

Тип нейронной сети	Ассоциативная память и распознавание образов	Сжатие информации	Прогнозирование	Оптимизация	Классификация, экспертные системы и диагностика	Кластеризация	Аппроксимация функции	Сглаживание (обобщение)
Многослойный персептрон (MLP)	+	+	+		+		+	+
Сеть радиального базиса (RBF)			+		+		+	+
Карта Кохонена				+	+	+		
Сеть адаптивного резонанса ART					+	+		

Если говорить об анализе данных, необходимо понимать, что данная область занимается исследованием и построением наиболее общих математических методов и вычислительных алгоритмов извлечения знаний из экспериментальных данных. Другими словами, можно сказать, что это процесс исследования, фильтрации, преобразования и моделирования данных, целью которого является извлечение полезной информации и принятие решений.

Изучая следующую область применения – управление, отметим, что главной задачей является расчет входного воздействия, при котором система будет следовать желаемой траектории или соответствовать требуемым параметрам. Одной из основных целей в отрасли является снижение эксплуатационных расходов. Примером послужило моделирование и управление с использованием модели на основе нейронной сети Predictive Control. Целью использования НС Predictive Control является управление для обеспечения высококачественного контроля при наличии нелинейностей, а также лучшего понимания алгоритма управления нейронной сетью. Сочетание НС и прогнозирующего контроля является прекрасным выбором для достижения хороших результатов в контроле.

Как следует из результатов анализа, приведенных в таблице 7, многослойный персептрон весьма хорошо справляется с задачами, в которых ответ действительно зависит только от того, что дается на вход сети, но никак не зависит от истории входов. Но не стоит забывать, что в наличии должно быть достаточно много примеров (желательно иметь не менее сотни примеров на каждую связь сети). Это связано с тем, что имея много коэффициентов, сеть может запомнить много примеров и может выдавать на них отличный результат, но ее прогнозы не будут иметь ничего общего с реальностью в случае, если дать на вход примеры не из обучающей выборки. Из этого следует вывод, что, несмотря на

возможность применения данного типа нейронной сети при решении различных классов задач (прогнозирование, ассоциативная память, классификация, диагностика, сглаживание), такой тип архитектуры не актуален в сфере телекоммуникаций.

Рассмотренные типы архитектур успешно справляются с обширным комплексом задач. Что касается выбора архитектуры нейронных сетей для решения поставленных задач в области управления, то рекомендуется использовать комбинирование различных типов архитектур НС. Данный способ моделирования сети открывает удивительные возможности, которые не доступны при использовании лишь одного типа архитектуры.

Развитие систем поддержки операций (Operations Support Systems, OSS) прямо пропорционально усложнению телекоммуникационных технологий [49-51]. Благодаря применению искусственных нейронных сетей автоматизация и управление инфраструктурой стали удобнее и проще. ИНС способны объединять между собой простые процессоры (искусственные нейроны) в достаточно большую сеть с управляемым взаимодействием, следовательно, способны выполнять довольно сложные задачи. Нейронные сети способны обучаться, этот факт дает им главное преимущество перед традиционными алгоритмами. В процессе обучения НС способна выявлять сложные зависимости между входными и выходными данными. К наиболее известным применениям искусственных нейронных сетей относятся: распознавание образов, принятие решений и управление, кластеризация, прогнозирование, аппроксимация, сжатие данных и ассоциативная память, анализ данных, оптимизация. Благодаря использованию ИНС становится возможным решение проблемы затруднительного анализа инфраструктур и автоматизации систем поддержки эксплуатационных процессов.



## 4. ПРАКТИЧЕСКАЯ РАБОТА

### 4.1. Создание нейронной сети

Моделью нейронной сети, исходя из результатов анализа архитектур, был выбран многослойный персептрон. Neural Network Toolbox предоставляет на выбор три метода обучения: Левенберга-Марквардта, байесовской регуляризации и сопряженных градиентов. Обучим нейронную сеть всеми тремя методами и проведем сравнительный анализ.

Изначально количество нейронов выбиралось равным 100 нейронам. Также было выбрано количество эпох равным 1000 (будет произведено 1000 предъявлений сети обучающего множества и проверок качества обучения после каждого множества). На рисунках 20-22 показан ход обучения нейронной сети с 100 нейронами в скрытом слое по трем методам обучения.

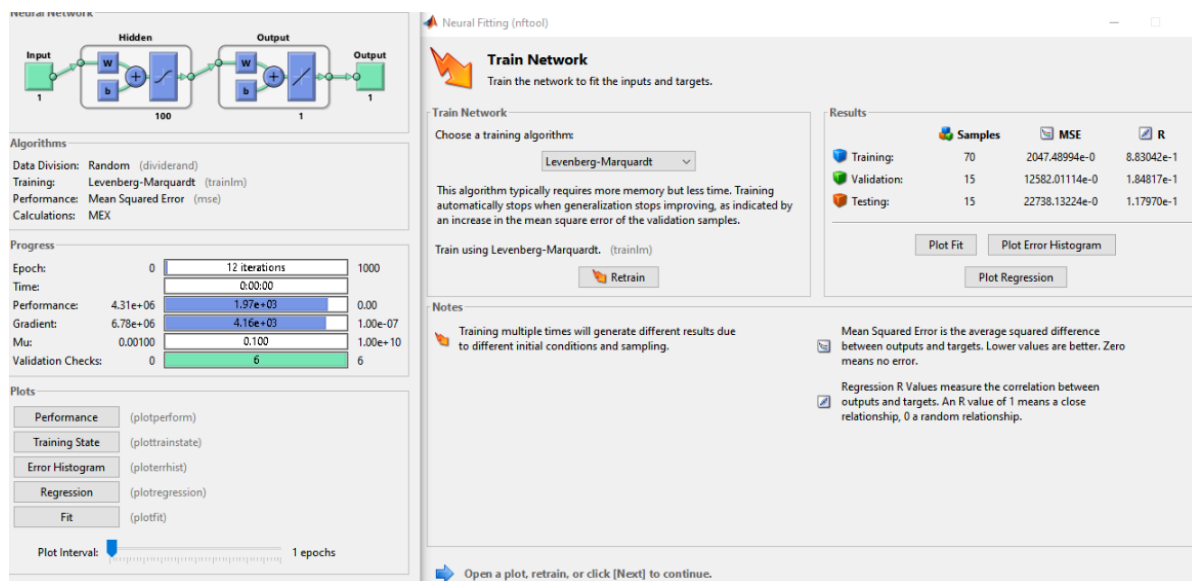


Рисунок 20. Метод Левенберга-Марквардта.

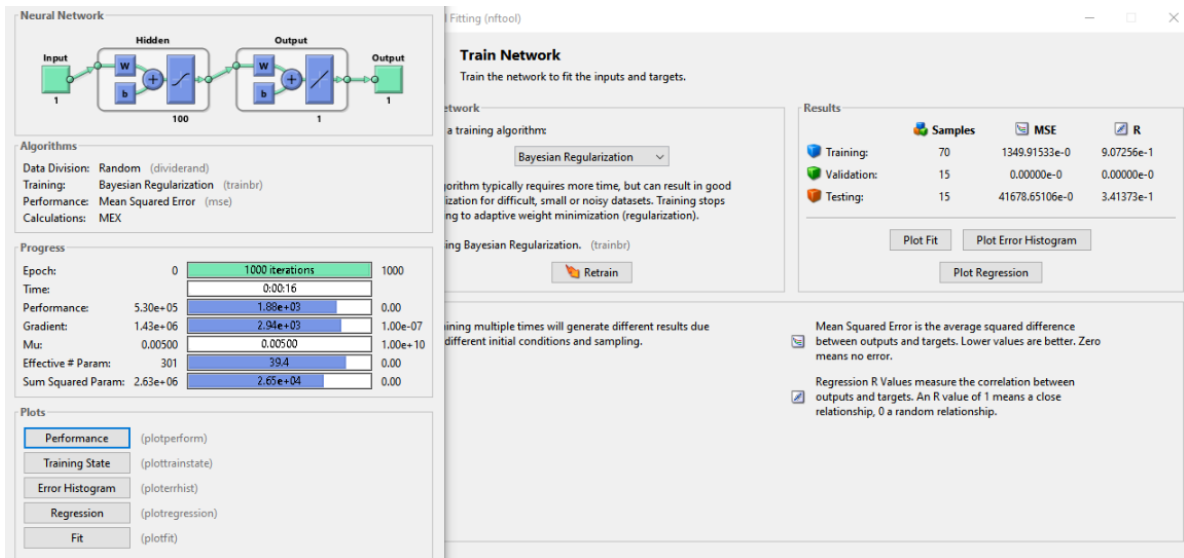


Рисунок 21. Метод байесовской регуляризации

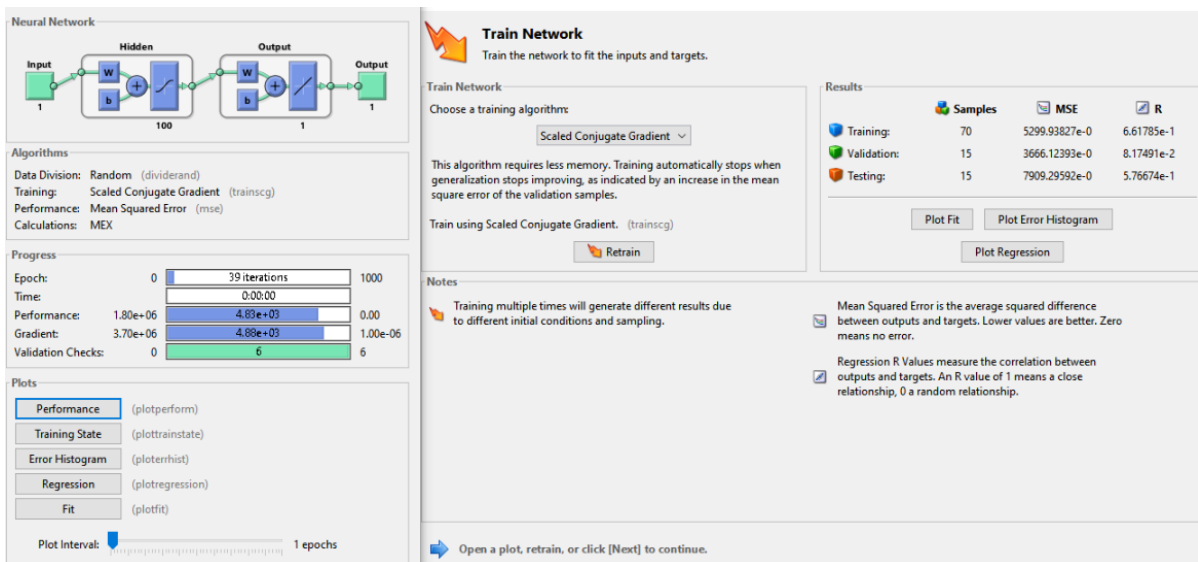


Рисунок 22. Метод сопряженных градиентов.

## 4.2. Результаты первого эксперимента

По итогу обучения были получены результаты, не соответствующие ожиданиям. Регрессия со всех обучающих алгоритмов была неудовлетворительной. На рисунках 23-25 представлена регрессия относительно обучающей (синим), тестовой (зеленым) и проверочной (красным) выборок.

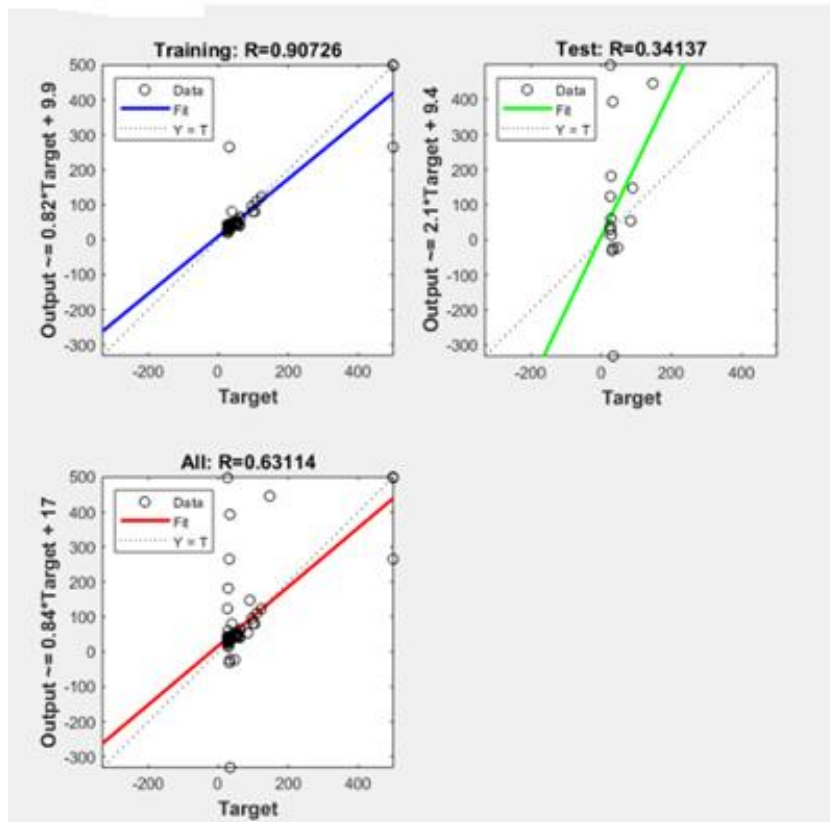


Рисунок 23. Регрессия. Метод байесовской регуляризации.

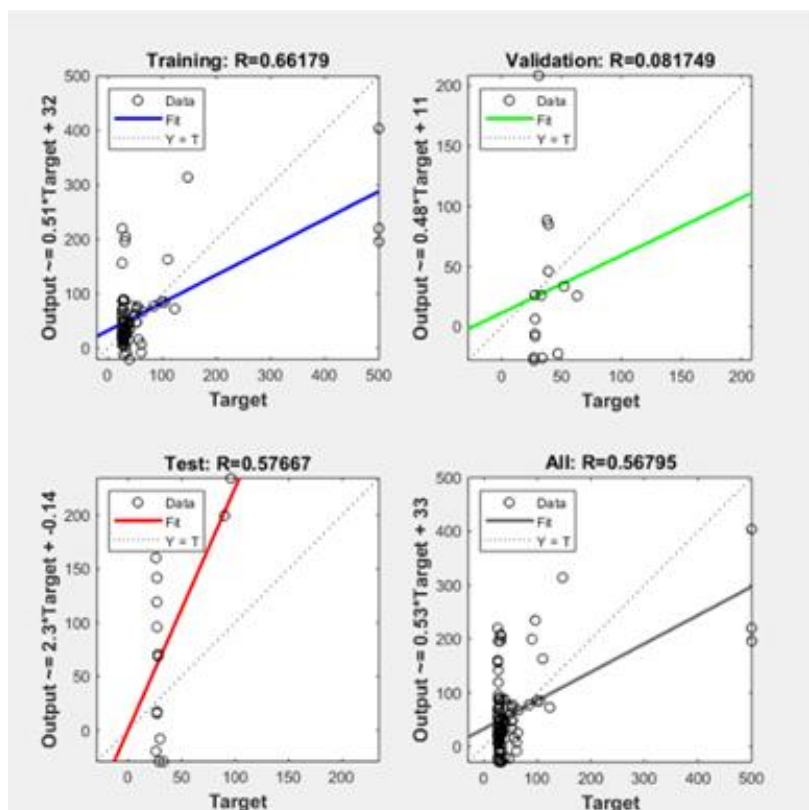


Рисунок 24. Регрессия. Метод сопряженных градиентов.

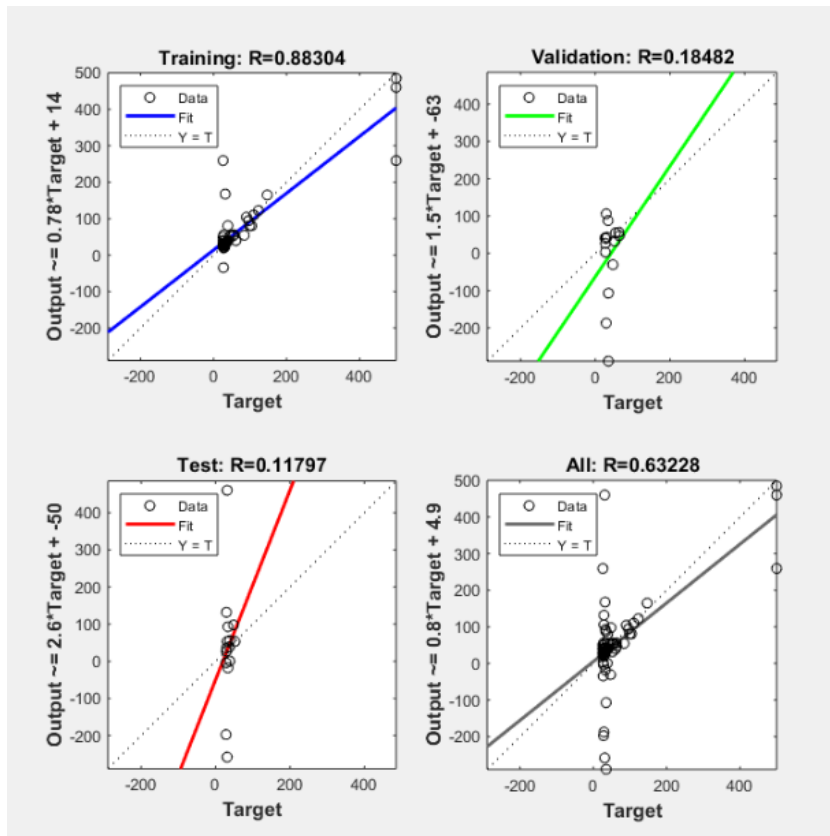


Рисунок 25. Регрессия. Метод Левенберга-Марквардта.

По итогу работы нейронной сети должны были получиться предсказания задержек. На рисунках 26-28 отображены графики задержки из изначальных данных по отношению к времени и значения задержки, полученной нейронной сетью.

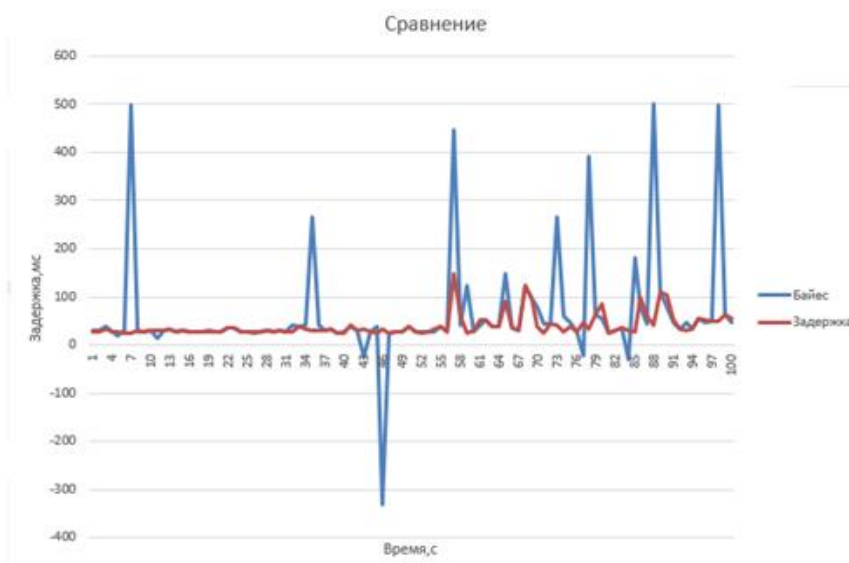


Рисунок 26. Сравнение изначальной и полученной задержки. Метод байесовской регуляризации.

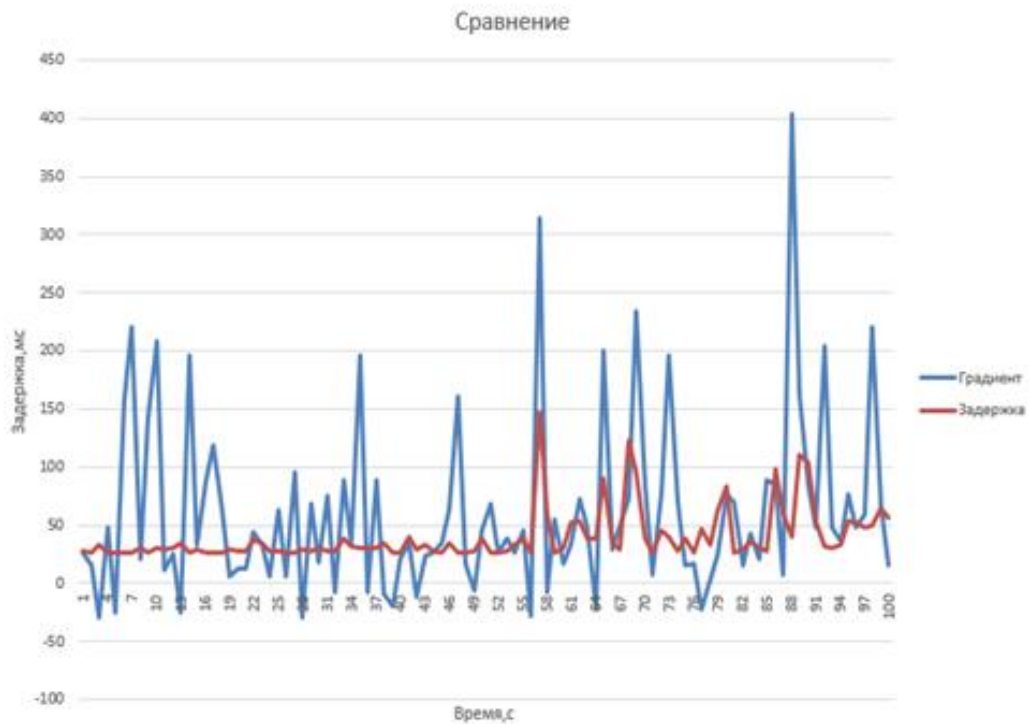


Рисунок 27. Сравнение изначальной и полученной задержки.  
Метод сопряженных градиентов.

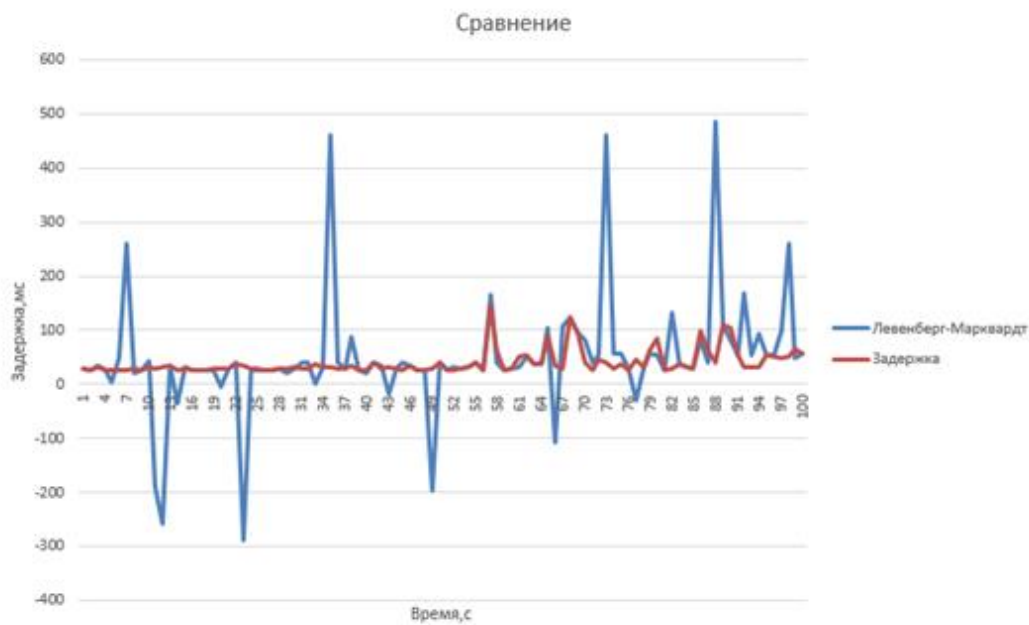


Рисунок 28. Сравнение изначальной и полученной задержки.  
Метод Левенберга-Марквардта.

Задержка, полученная нейронной сетью в каждом варианте обучения, не только сильно отличается, но и становится отрицательной,

что невозможно. Тем не менее, отметим, что в некоторых моментах нейронная сеть выдала ожидаемое значение задержки.

### 4.3. Результаты второго эксперимента

Для улучшения показателей нейронной сети было решено использовать в скрытом слое 200 нейронов вместо 100. На рисунке 29 представлена получившаяся регрессия нейронной сети, обученной методом байесовской регуляризации относительно обучающей (синим), тестовой (зеленым) и проверочной (красным) выборок.

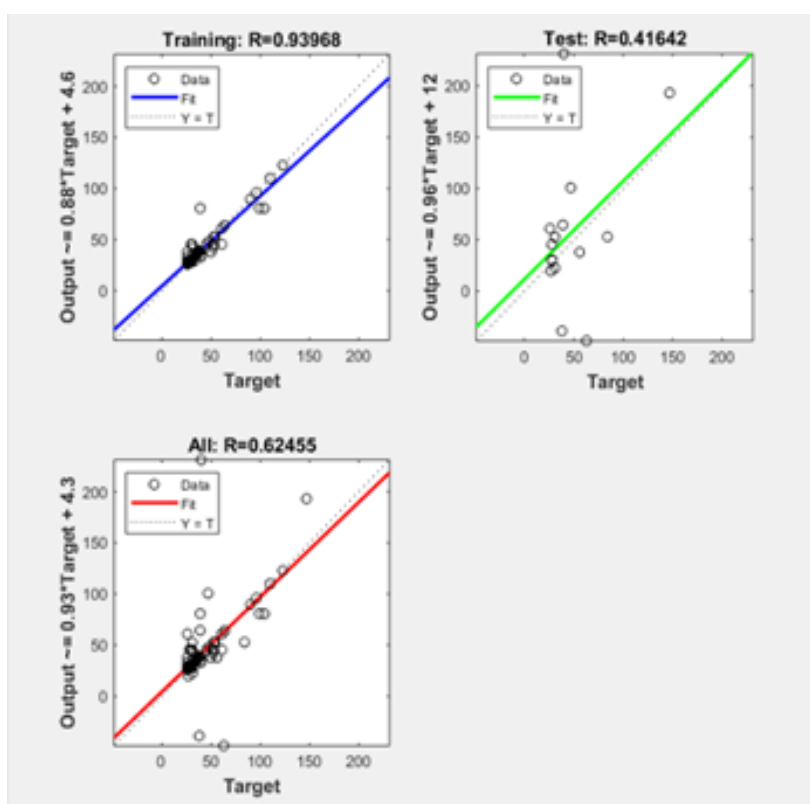


Рисунок 29. Регрессия. Метод байесовской регуляризации.

Метод Байесовской регуляризации показал достаточно хороший результат по отношению к предыдущему эксперименту. Далее на рисунке 30 показана регрессия при обучении НС методом сопряженных градиентов.

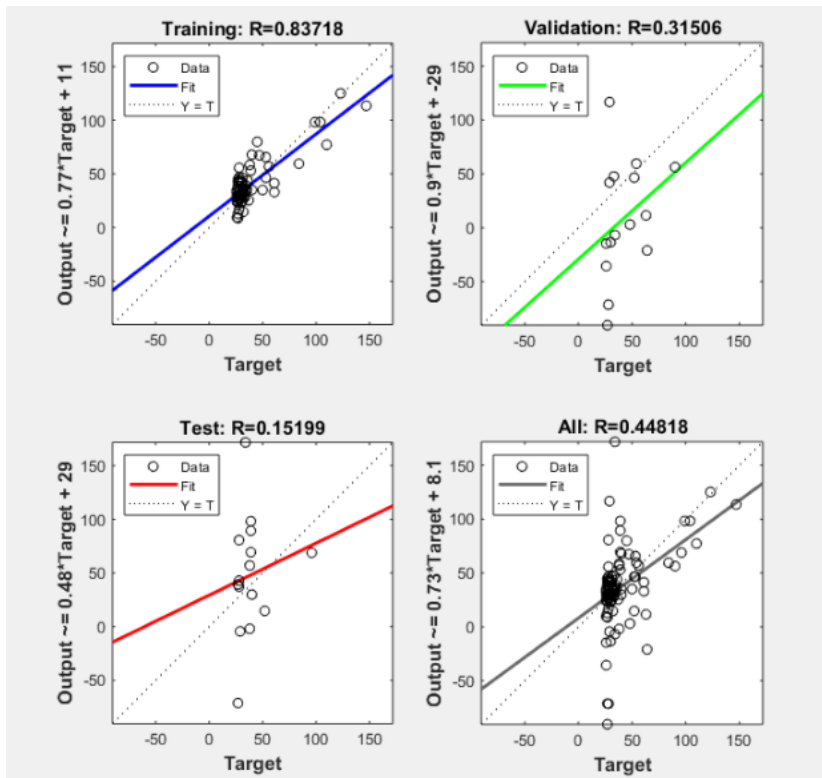


Рисунок 30. Регрессия. Метод сопряженных градиентов.

Метод сопряженных градиентов в очередной раз показывает самый неудовлетворительный результат. Немного лучше результат при обучении сети методом Левенберга-Марквардта (рисунок 31).

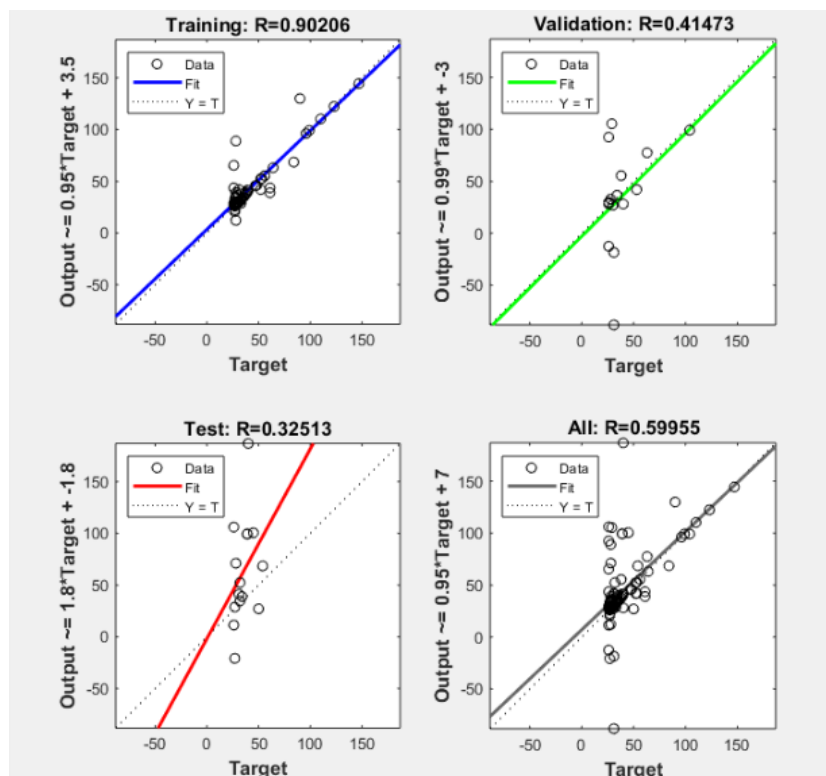


Рисунок 31. Регрессия. Метод Левенберга-Марквардта.

Рассмотрим графики изначальной и полученной с помощью нейронной сетью задержки. Нейронная сеть обучалась методом сопряженных градиентов (рисунок 32).

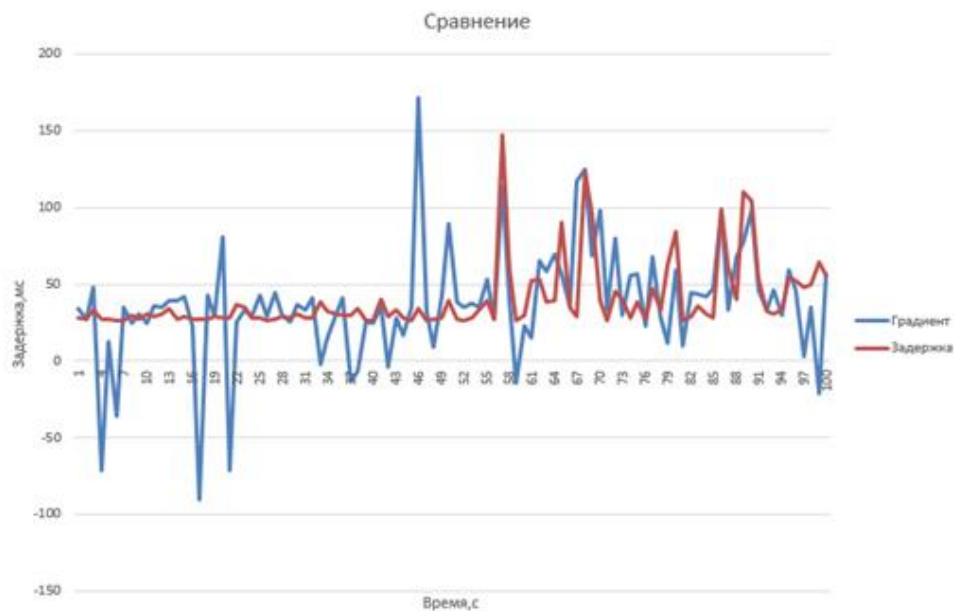


Рисунок 32. Сравнение изначальной и угаданной задержки. Метод сопряженных градиентов.

Метод сопряженных градиентов в этот раз показывает лучшие результаты, однако результаты полученные при обучении НС методом Левенберга-Марквардта превосходят метод сопряженных градиентов (Рисунок 33).



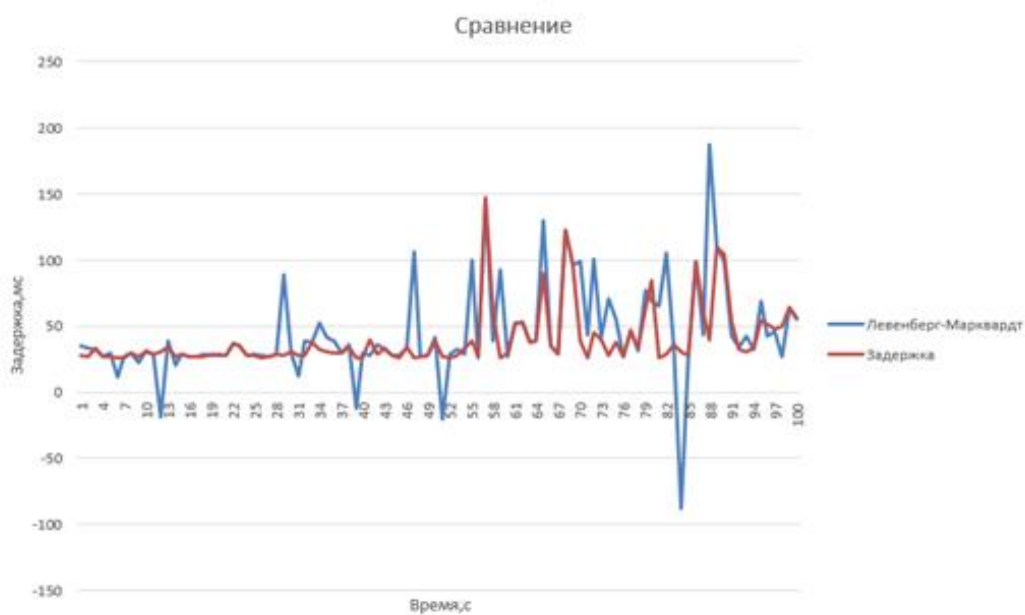


Рисунок 33. Сравнение изначальной и полученной задержки.  
Метод Левенберга-Марквардта

Наконец НС, обученная алгоритмом байесовской регуляризации, несмотря на то, что выдает отрицательное значение задержки, лучше всего определяет начальные примеры (рисунок 34).

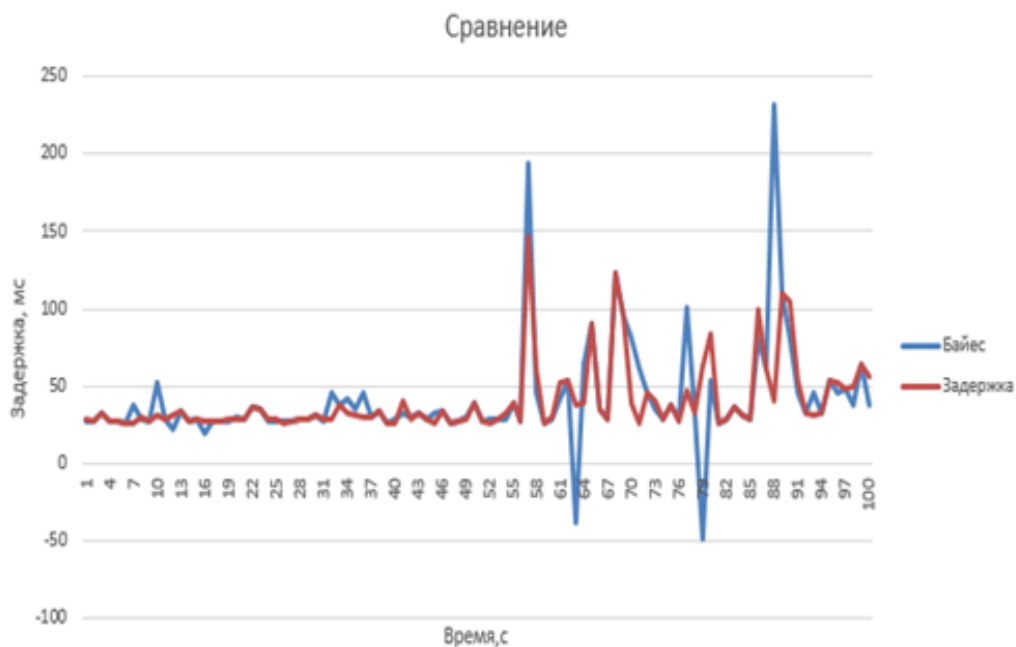


Рисунок 34. Сравнение изначальной и полученной задержки.  
Метод байесовской регуляризации

#### 4.4. Выводы по результатам экспериментов

Представлен общий метод препроцессинга данных для представления, как на вход нейронной сети, так и для обучающей выборки. Показан метод линейного сдвига и то, в какой форме стоит подавать нейронной сети данные. Далее было показано, как можно выполнить постобработку полученных данных для представления их в форму, удобную для человеческого восприятия. Это включает в себя не только возврат данных в начальное состояние, но и графического представления полученных результатов. Следующим пунктом было проведение эксперимента, где нужно было построить нейронную сеть и обучить ее несколькими алгоритмами, чтобы произвести сравнение и сделать выводы о наиболее подходящем алгоритме обучения. Нейронная сеть обучалась статистическими данными о задержках в сети и должна была при подаче на вход данных о переданных пакетах в секунду определить соответствующую задержку в сети TCP/IP. А значит, прежде всего, определить («угадать») задержку при подаче уже известной информации о переданных данных. Для этого была создана нейронная сеть в программной библиотеке Neural Network Toolbox являющаяся многослойным персептроном и обучена тремя методами обучения: Левенберга-Марквардта, байесовской регуляризации и сопряженных градиентов. Первоначально задумывалось установить число нейронов в скрытом слое нейронной сети равным *100*, что и было выполнено. Однако результаты работы такой нейронной сети были неудовлетворительными, регрессия была слишком большой, и появились отрицательные значения задержки, такой результат не только не предполагался, но и нереален. Тем не менее, при обучении методом Левенберга-Марквардта и методом байесовской регуляризации прослеживалась некая тенденция к правильным ответам.

После того как работоспособность нейронной сети со *100* нейронами в скрытом слое была проверена, эмпирическим путем было выявлено

улучшение результатов работы при добавлении еще 100 нейронов. Соответственно улучшилась регрессия и сами значения определялись лучше, однако это не решило проблему отрицательных значений задержки. Способом решения этой проблемы предлагается смена активационной функции в нейронной сети с линейной на сигмоидальную, которая выдает значения больше нуля. Для этого понадобятся более углубленные знания языка MATLAB.

Проведем сравнение полученных результатов о методах обучения представленных в Neural Network Toolbox в таблице 8. Значения из диапазона  $[0, 5]$  показывают степень удовлетворительности работы нейронной сети от 0 – неудовлетворительно, 5 – удовлетворительно.

Таблица 8. Сравнение методов обучения после эксперимента

Количество нейронов в скрытом слое/ метод обучения	Сопряженных градиентов	Левенберга-Марквардта	Байесовской регуляризации
100	0	2	3
200	1	2	4

Из результатов сравнения можно сделать вывод о том, что метод обучения байесовской регуляризации больше всего из представленных подходит в решении задач на предсказание, а в рамках данной задачи, предсказание задержек в сетях на базе протокола TCP/IP. Следовательно, вместе с архитектурой многослойного персептрона, метод обучения байесовской регуляризации может быть использован для реализации решений оператора связи по управлению TCP/IP сетями.

## ЗАКЛЮЧЕНИЕ

За множество лет существования сетей связи накопилось большое количество технологий построения сетей, образовались громоздкие архитектуры сетей. На данный момент их всех объединяет концепция конвергенции, в том числе и по схожим проблемам и задачам в управлении сетями, которые с определенным успехом решают операторы связи.

Проблемы и задачи связаны, прежде всего, с тем, что на сегодняшний день в сетях передается и хранится огромное количество информации, из этого вытекает задача нахождения пика трафика, предсказание самоподобного трафика и предсказание загрузки канала.

Пособие направлено на исследование основных моделей нейронных сетей из двух видов: прямого распространения и с обратными связями. И методов их обучения, представленных в двух парадигмах: с учителем и без учителя. Для представления одной из возможности решения задач управления сетями TCP/IP в рамках автоматизированного подхода. Было проведено сравнение представленных архитектур нейронных сетей и алгоритмов обучения с выводом о том, какая из моделей и методов больше всего подходит для общего применения. Также был показан общий метод подготовки данных для работы с нейронной сетью. Была построена нейронная сеть с представленными в программной библиотеке методами обучения. По итогу проведения эксперимента были сделаны выводы о том, на сколько верно была решена задача, поставленная перед нейронной сетью, какая конфигурация предпочтительнее при решении такой задачи, что можно улучшить для более корректного решения и какой из возможных методов обучения нейронной сети из имеющегося в программной библиотеке лучше справился с задачей.

## ЛИТЕРАТУРА

1. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия–Телеком, 2013. 384 с.
2. McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity // The bulletin of mathematical biophysics, 1943, vol. 5, pp.115–133.
3. Hebb D.O. The organization of behavior: a neuropsychological theory. John Wiley & Sons, 1949.
4. Farley B., Clark W. Simulation of self-organizing systems by digital computer // Transactions of the IRE Professional Group on Information Theory, 1954, vol. 4, iss. 4, pp. 76-84.
5. Rochester N., Holland J., Haibt L., Duda W. Tests on a cell assembly theory of the action of the brain, using a large digital computer // IRE Transactions on Information Theory, 1956, vol. 2, iss. 3, pp. 80-93.
6. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain// Psychological Review, 1958, vol. 65, iss. 6, pp. 386–408.
7. Minsky M.L., Papert S.A. Perceptrons. MIT Press, Cambridge, MA, 1969.
8. Werbos P.J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
9. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position // Biological Cybernetics, 1980, vol. 36, pp. 193–202.
10. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proceedings of the National Academy of Sciences, 1982, vol. 79, iss. 8, pp. 2554-2558.
11. Rumelhart D.E., McClelland J.L. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge, MA: MIT Press, 1986.
12. Jain A.K., Mao J., Mohiuddin K.M. Artificial neural networks: A tutorial // Computer, 1996, vol. 29, iss. 3, pp. 31-44.
13. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE, 1998, vol. 86, iss. 11, pp. 2278-2324.
14. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012.
15. Zeiler M.D., Fergus R. Visualizing and understanding convolutional networks // Lecture Notes in Computer Science, 2014, vol 8689, pp. 818-833.

16. Szegedy C., Liu W., Jia Y., et al. Going deeper with convolutions // IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
17. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
18. Jia Y. Caffe - deep learning framework. 2017. Режим доступа: <http://caffe.berkeleyvision.org> (дата обращения 01.08.2020).
19. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
20. Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи. Учебник для вузов. - СПб.: БХВ-Петербург, 2010.
21. Гольдштейн Б.С. Инфокоммуникационные сети и системы. - СПб.: БХВ-Петербург, 2019.
22. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Интеллектуальные сети. - М.: Радио и связь, 2000.
23. Гольдштейн Б.С. Протоколы сети доступа. Том 2. 3-е издание. - СПб.: БХВ-Петербург, 2005.
24. Гольдштейн Б.С., Елагин В.С., Сенченко Ю.Л. Протоколы AAA: RADIUS и Diameter. Книга 9. Сер. Телекоммуникационные протоколы ЕСЭ РФ. - СПб.: БХВ-Петербург, 2010.
25. Елагин В.С., Спиркина А.В., Владыко А.Г., Иванов Е.И., Помогалова А.В., Аптриева Е.А. Основные сетевые характеристики blockchain трафика и подходы к моделированию. Т-Comm: Телекоммуникации и транспорт. 2020. Т. 14. № 4. С. 39-45.
26. Yuhas B., Ansari N. Neural networks in telecommunications. Springer US, 1994.
27. Rauch H.E., Winarske T. Neural networks for routing communication traffic // IEEE Control Systems Magazine, 1988, vol. 8, iss, 2, pp. 26-31.
28. Zhang L., Thomopoulos S.C.A. Neural network implementation of the shortest path algorithm for traffic routing in communication networks // International 1989 Joint Conference on Neural Networks, 1989.
29. Fischer M.M., Gopal S. Artificial neural networks: a new approach to modeling interregional telecommunication flows // Journal of regional Science, 1994, vol. 34, iss. 4, pp. 503-527.
30. Khoshgoftaar T.M., Allen E.B., Hudepohl J.P., Aud S.J. Application of neural networks to software quality modeling of a very large telecommunications system // IEEE Transactions on Neural Networks, 1997, vol. 8, iss. 4, pp. 902-909.
31. Khotanzad A., Sadek N. Multi-scale high-speed network traffic prediction using combination of neural networks // Proceedings of the International Joint Conference on Neural Networks, 2003.

32. Комашинский В.И., Смирнов Д.А. Нейронные сети и их применение в системах управления и связи. - М.: Горячая линия-Телеком, 2003.
33. Семейкин В.Д., Скупченко А.В., Метод управления компьютерной сетью на базе нейронных сетей // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2009. № 2. С. 161-165.
34. Султанов А.Х., Габдрахманов А.А., Габдрахманова Н.Т. Использование нейросетевых моделей для прогнозирования интенсивностей самоподобного трафика телекоммуникационной сети// Вестник уфимского государственного авиационного технического университета. 2011. Т.15. №2. С. 93-98.
35. Васылив Н.И., Есалов К.Э., Кисляков С.В., Пупцев Р.И. Сравнительный анализ нейросетевых моделей применительно к задаче прогнозирования нагрузки контакт-центра // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). Сборник научных статей VIII Международной научно-технической и научно-методической конференции. 2019. С. 201-205.
36. Есалов К.Э., Маслюхин С.М., Павленко М.Е. Анализ архитектур нейронных сетей для решения задач классификации трафика // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2018). VII Международная научно-техническая и научно-методическая конференция. Сборник научных статей. 2018. С. 367-372.
37. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg S.E., Azodolmolky S., Uhlig S. Software-Defined Networking: A Comprehensive Survey // Proceedings of the IEEE, 2015, vol. 103, iss. 1, pp. 14-76.
38. Konda V., Tsitsiklis J. Actor-critic algorithms // Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99), 1999, pp. 1008-1014.
39. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. 2017. Режим доступа: <http://www.deeplearning.net/software/theano/> (дата обращения 01.08.2020).
40. TensorFlow. 2020. Режим доступа: <https://www.tensorflow.org> (дата обращения 01.08.2020).
41. MATLAB. The MathWorks, Inc. 2020. Режим доступа: <https://www.mathworks.com/products/matlab.html> (дата обращения 01.08.2020).
42. Deep Learning Toolbox™. The MathWorks, Inc. 2020. Режим доступа: <https://www.mathworks.com/products/deep-learning.html> (дата обращения 01.08.2020).

43. Levenberg K. A Method for the Solution of Certain Non-Linear Problems in Least Squares // Quarterly of Applied Mathematics, 1944, vol. 2, iss. 2, pp. 164–168.
44. Marquardt D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters // SIAM Journal on Applied Mathematics, 1963, vol. 11, iss. 2, pp. 431–441.
45. Burden F., Winkler D. Bayesian Regularization of Neural Networks Artificial Neural Networks, 2008, vol 458, pp 23-42.
46. Møller M.F. A scaled conjugate gradient algorithm for fast supervised learning // Neural Networks, 1993, vol. 6, iss. 4, pp. 525-533.
47. Царегородцев В.Г. Оптимизация предобработки данных: константа Липшица обучающей выборки и свойства обученных нейронных сетей // Нейрокомпьютеры: разработка, применение. 2003. № 7. С. 3-8.
48. Çalhan A., Çeken C. An adaptive neuro-fuzzy based vertical handoff decision algorithm for wireless heterogeneous networks // 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2010.
49. Elagin V.S., Goldshtein A.B., Onufrienko A.V., Zarubin A.A., Belozertsev I.A. Synchronization of delay for OTT services in LTE // Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), 2018, pp. 1-4.
50. Elagin V.S., Goldshtein B.S., Onufrienko A.V., Zarubin A.A., Savelieva A. A. The efficiency of the DPI system for identifying traffic and providing the quality of OTT services// Systems of Signals Generating and Processing in the Field of on Board Communications, 2018, pp. 1-5.
51. Elagin V.S., Belozertsev I.A., Goldshtein B.S., Onufrienko A.V., Vladyko A.G. Models of QOE ensuring for OTT services // Systems of Signals Generating and Processing in the Field of on Board Communications, 2019, pp. 1-4.



Елагин Василий Сергеевич  
Владыко Андрей Геннадьевич  
Пупцев Ринат Игоревич  
Окунева Дарина Владимировна

# ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ. ПРИКЛАДНЫЕ АСПЕКТЫ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ В СИСТЕМАХ УПРАВЛЕНИЯ ИНФОКОММУНИКАЦИЯМИ

Учебное пособие

План издания 2020 г., п. 85

Подписано к печати 01.08.2020  
Объем 4,4 усл.-печ. л. Тираж 500 экз. Заказ 956  
Редакционно-издательский отдел СПбГУТ  
191186 СПб., наб. р. Мойки, 61, литера А  
Отпечатано в \_\_\_\_\_