

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа Информационных технологий и робототехники  
Направление подготовки 09.03.01 «Информатика и вычислительная техника»  
Отделение школы (НОЦ) Отделение информационных технологий

**БАКАЛАВРСКАЯ РАБОТА**

Тема работы
<b>Разработка библиотеки обучения нейронных сетей для решения задачи прогнозирования временных рядов</b>

УДК 004.85.032.26:519.246.8

Студент

Группа	ФИО	Подпись	Дата
8В4А	Урываев К. А.		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Болотова Ю. А.	к.т.н.		
Аспирант	Савицкий Ю.В.			

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Старикова Е. В.	к.ф.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева И. И.			

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
09.03.01 Информатика и ВТ	Погребной А. В.	к.т.н.		

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ПО ОСНОВНОЙ  
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ  
09.03.01 «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА», ИК  
ТПУ, ПРОФИЛЬ «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, КОМПЛЕКСЫ,  
СИСТЕМЫ И СЕТИ»**

Код результата тов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа информационных технологий и робототехники  
Направление подготовки 09.03.01 «Информатика и вычислительная техника»  
Отделение школы (НОЦ) информационных технологий

УТВЕРЖДАЮ:  
Руководитель ООП  
\_\_\_\_\_ Погребной А.В.  
(Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8В4А	Урываев Кирилл Антонович

Тема работы:

<b>Разработка библиотеки обучения нейронных сетей для решения задачи прогнозирования временных рядов</b>	
Утверждена приказом директора (дата, номер)	от 20.03.2018 № 1970/с

Срок сдачи студентом выполненной работы:	31.05.18
--	----------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Техническое задание для разработки библиотеки обучения нейронных сетей для решения задачи прогнозирования временных рядов
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	<ol style="list-style-type: none"><li>1 Обзор методов прогнозирования и конкурентных решений;</li><li>2 Проектирование библиотеки прогнозирования, приложения для взаимодействия с ней и приложения нормализации;</li><li>3 Реализация и тестирование приложений и библиотеки на Proben1;</li><li>4 Решение задачи прогнозирования кассовых сборов фильмов</li></ol>

	5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; 6 Социальная ответственность.
<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>	– UML диаграмма классов приложений – Алгоритм прунинга
<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Старикова Екатерина Васильевна
Социальная ответственность	Авдеева Ирина Ивановна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	1.02.18
---	---------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Болотова Юлия Александровна	кандидат технических наук		1.02.18

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8В4А	Урываев Кирилл Антонович		1.02.18

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа Информационных технологий и робототехники

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Уровень образования бакалавриат

Отделение школы (НОЦ) Отделение информационных технологий

Период выполнения \_\_\_\_\_ (осенний / весенний семестр 2017/2018 учебного года)

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	31.05.2018
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
1.02.18	Аналитический обзор	15
2.03.18	Проектирования библиотеки и приложения	15
18.04.18	Реализация и тестирование приложений и библиотеки	30
4.05.18	Прогнозирование кассовых сборов фильмов	20
16.05.18	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
25.05.18	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Болотова Ю. А.	к.т.н.		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
09.03.01 Информатика и ВТ	Погребной А. В.	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8В4А	Урываев Кирилл Антонович

Школа	ИШИТР	Отделение	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.01 Информатика и вычислительная техника

<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Объект исследования – приложение, реализующее прогнозирование временных рядов с помощью нейронной сети, реализованное на персональном компьютере.</p> <p>Рабочая зона – аудитория, оборудованная системой отопления, кондиционирования воздуха, с естественным и искусственным освещением. Рабочее место – стационарное, оборудованное компьютером.</p> <p>Область применения – любая аналитическая деятельность, обладающая достаточным числом данных, для достижения допустимой погрешности.</p>
<b>Перечень вопросов, подлежащих исследованию, проектированию и разработке:</b>	
<p><b>1. Производственная безопасность</b></p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p>	<p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации включает: электромагнитное излучение, слабая освещенность, плохой микроклимат, умственное перенапряжение, перегрузки из-за монотонности труда. Требования к помещению описаны в СанПиН 2.2.4.3359-16.</p> <p>1.2. Анализ выявленных опасных факторов</p>

	при разработке и эксплуатации проектируемого решения включает: удар электрическим током и статическим электричеством, возможность короткого замыкания.
<b>2. Экологическая безопасность:</b>	Утилизация используемой оргтехники, компонентов компьютера и других компонентов аппаратно-программного комплекса, люминесцентных ламп, а так же канцелярских принадлежностей и бумаги.
<b>3. Безопасность в чрезвычайных ситуациях:</b>	Чрезвычайная ситуация техногенного характера для данного помещения – пожар в результате возгорания электропроводки, перегрева рабочих частей ПК. Установка общих правил поведения и рекомендаций во время пожара, план эвакуации, наличие исправного огнетушителя.
<b>4. Правовые и организационные вопросы обеспечения безопасности:</b>	Регулирование организации рабочих мест при выполнении работ сидя согласно ГОСТ 12.2.032-78. Анализ правильного расположения и компоновки рабочих мест, режима работы. Основные проводимые правовые и организационные мероприятия по обеспечению безопасности трудящихся в учебных аудиториях.

<b>Дата выдачи задания для раздела по линейному графику</b>	01.03.2018
---	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева И. И.			01.03.2018

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8В4А	Урываев Кирилл Антонович		01.03.2018

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8В4А	Урываев Кирилл Антонович

Школа	ИШИТР	Отделение	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.01 «Информатика и вычислительная техника»

### Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Энергетические (стоимость в рублях на 1 кВт/ч для юрид. лиц), информационные (час работы в интернете) и человеческие (согласно окладам научного руководителя и инженера-программиста)
2. Нормы и нормативы расходования ресурсов	Нормы из реальных осуществляемых затрат: потребление технических ресурсов, норма потребления электроэнергии
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Для юридических лиц в области образования социальные отчисления – 27,1%

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценка ресурсной, социальной эффективности НИ и потенциальных рисков. На основании информации, представленной в научных статьях и публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах, определить методику расчета экономической эффективности.
2. Планирование и формирование бюджета научных исследований	
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	

Дата выдачи задания для раздела по линейному графику	18.03.2018
--	------------

### Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Старикова Екатерина Васильевна	Кандидат философских наук		18.03.2018

### Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В4А	Урываев Кирилл Антонович		18.03.2018



## РЕФЕРАТ

Выпускная квалификационная работа 85с., 17 таб., 18 рис., 33 источников.

Ключевые слова: искусственные нейронные сети, прогнозирование, экстраполяция, программирование, машинное обучение.

Объект исследования: нейронные сети и их использование в прогнозировании временных рядов.

Цель работы – разработка библиотеки обучения искусственной нейронной сети.

Область применения: аналитика, прогнозирование, расчёты.

В процессе исследования изучены методы прогнозирования, принцип работы нейронных сетей, конкурентные решения, методы повышения результатов.

В результате исследования была разработана библиотека обучения нейронной сети, приложение для работы с ней и приложение для нормализации данных. Работоспособность была проверена на Proben1 и на задаче прогнозирования кассовых сборов фильмов.

## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ**

**ИНС** – искусственная нейронная сеть

**Бенчмарк** – набор данных для проверки производительности и работоспособности

**C#** – объектно-ориентированный язык программирования

**GPU** – graphics processing unit, графический процессор

**Экстраполяция** – особый тип аппроксимации, при котором функция аппроксимируется вне заданного интервала

**Сигмоида** — это гладкая монотонная нелинейная функция, имеющая форму буквы "S"

## Оглавление

РЕФЕРАТ .....	9
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ .....	10
ВВЕДЕНИЕ.....	15
1 АНАЛИТИЧЕСКИЙ ОБЗОР .....	17
1.1. Описание задачи прогнозирования временных рядов .....	17
1.1.1 Общие положения.....	17
1.1.2 Формулировка задачи и методы её решения .....	17
1.1.3 Применение нейронных сетей в решении задач экстраполяции.....	19
1.2. Описание работы искусственной нейронной сети .....	21
1.2.1 Общие положения.....	21
1.2.2 Принцип работы искусственных нейронных сетей .....	22
1.2.3 Обучение обратным распространением ошибки.....	24
1.2.4 Метрики для определения результата .....	27
1.3. Обзор существующих решений.....	29
1.3.1 Lasagne .....	29
1.3.2 Caffe.....	29
1.3.3 Keras .....	31
1.4. Исследование методов улучшения результата прогнозирования.....	32
1.4.1. Прунинг.....	32
1.4.2. Разреженные матрицы.....	33
1.4.3. Динамический коэффициент обучения .....	34
1.4.4. Инициализация Ксавьера .....	35

1.4.5. Перемешивание данных в ходе обучения .....	35
2 ПРОЕКТИРОВАНИЕ .....	36
2.1 Проектирование структуры библиотеки.....	37
2.2 Проектирование структуры приложения по работе с библиотекой.....	38
2.3 Проектирование приложения нормализации данных .....	39
3 РЕАЛИЗАЦИЯ.....	41
3.1. Реализация библиотеки обучения нейронной сети .....	41
3.2. Реализация приложения по работе с библиотекой .....	42
3.3. Реализация приложения по нормализации данных .....	44
3.4. Тестирование библиотеки на Proben1 .....	45
4 РЕЗУЛЬТАТЫ РАБОТЫ .....	45
4.1 Постановка задачи прогнозирования временного ряда для кассовых сборов фильмов .....	45
4.2 Подготовка данных .....	46
4.3 Результаты обучения.....	48
5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И ЭНЕРГОСБЕРЕЖЕНИЕ .....	50
5.1 Введение.....	50
5.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	50
5.2.1 Потенциальные потребители результатов исследования .....	50
5.2.2 Анализ конкурентных технических решений.....	51
5.2.3 Технология QuaD .....	53
5.2.4 SWOT-анализ.....	54

5.3	Определение возможных альтернатив проведения научных исследований .....	56
5.4	Планирование научно-исследовательских работ.....	57
5.1	Структура работ в рамках научного исследования .....	57
5.2	Определение трудоемкости выполнения работ .....	58
5.3	Разработка графика проведения научного исследования.....	59
5.4	Бюджет научно-технического исследования (НТИ) .....	61
5.5	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .....	64
6	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ .....	68
6.1	Производственная безопасность.....	68
6.1.1	Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.....	68
6.1.2	Монотонность работы .....	68
6.1.3	Микроклимат .....	69
6.1.4	Освещение .....	70
6.1.5	Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности...	72
6.1.6	Статическое электричество.....	72
6.1.7	Высокое напряжение электрической сети.....	72
6.2	Экологическая безопасность.....	74
6.3	Безопасность в чрезвычайных ситуациях.....	75
6.4	Правовые и организационные вопросы обеспечения безопасности.....	76
6.4.1	Правовые нормы трудового законодательства .....	76

6.4.2 Организационные мероприятия при компоновке рабочей зоны.....	77
ЗАКЛЮЧЕНИЕ .....	79
СПИСОК ПУБЛИКАЦИЙ СТУДЕНТА.....	81
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	82

## **ВВЕДЕНИЕ**

Прогнозирование нелинейная и часто достаточно труднореализуемая задача, находящая своё применение как в чисто научных областях (физика, химия, биология и т.д.), так и в непосредственно прикладных (маркетинг, статистика и т.д.). Для различных функций существуют различные математические методы прогнозирования, но они могут давать большие погрешности и иногда требуют больших вычислений. В связи с этим, есть необходимость искать новые методы решения задачи прогнозирования.

Искусственные нейронные сети, получившие в последнее время большое распространение, являются инструментом, с помощью которого можно эффективно решать поставленные задачи. Для ИНС прогнозирование временных рядов – это задача нелинейной регрессии.

Целью работы является разработка библиотеки нейронных сетей для решения задач прогнозирования. Для этого необходимо изучить существующие методы прогнозирования временных рядов. Также уже есть варианты реализации ИНС, и необходимо рассмотреть их, и выявить, какие у них есть достоинства и недостатки.

Помимо самого алгоритма ИНС, существуют различные способы улучшения результата обучения ИНС. Такими способами являются, например, динамическое изменение коэффициента обучения (learning rate), которое должно приблизить нейросеть к локальному минимуму, и инициализация Ксавьера, которая ускорит обучение за счёт того, что начальные веса будут уже инициализированы случайными значениями, сгенерированными из равномерного случайного распределения, имеющего определённую дисперсию. Менее однозначным способом является прунинг, т.е. удаление (обнуление) связей с маленьким по отношению к остальным весом. Также для реализации прунинга необходимо перейти на разреженные матрицы, т.е. матрицы с большим количеством 0.

Для взаимодействия с библиотекой необходим какой-либо способ, какая-то оболочка. Большинство существующих решений использует python в

качестве языка взаимодействия. Но в данной работе для большего удобства будет реализован графический интерфейс. Помимо приложения для самого взаимодействия с сетью, необходимо приложение, которое будет подготавливать данные для обучения. Для этого необходимо разработать алгоритмы нормализации для каждого типа данных.

Для того, чтобы убедиться в работоспособности библиотеки, необходимо проверить её на тестовом задании. Набор таких заданий содержится в бенчмарке Proben1, поэтому перед решением реальной задачи, необходимо будет решить хотя бы пару заданий из Proben1.

И для подтверждения результата работы, т.е. для проверки библиотеки обучения ИНС, необходимо решить реальную задачу. В качестве такой задачи было выбрано прогнозирование кассовых сборов фильмов. Данная задача уже решалась в другом исследовании, и там был достигнут результат около 61.28% [1]. Он и будет опорой при учёте результата полученным нашей ИНС.

Таким образом, для достижения поставленной цели необходимо решить следующие задачи:



# **1 АНАЛИТИЧЕСКИЙ ОБЗОР**

## **1.1. Описание задачи прогнозирования временных рядов**

### **1.1.1 Общие положения**

Главным инструментом прогнозирования является экстраполяция. И задача что-либо спрогнозировать встречается достаточно часто, это и прогноз погоды, и котировок на бирже, и прогнозирование следующего хода противника в шахматах, и т.д.

Для решения некоторых задач данного типа подходят и обычные алгоритмы. Особенно они хороши в типах задач, где нет случайных факторов и число комбинаций не слишком велико, например в крестиках-ноликах. Можно написать алгоритм, который будет разыгрывать самые выигрышные тактики в зависимости от ситуации.

Но в шахматах, го или даже шашках уже сложно просчитать все варианты, так как их число слишком велико, и не всегда эвристические методы не всегда дают хороший результат. Это относится и ко многим другим процессам, в которых найти закономерность уже сложнее, чем в перечисленных выше играх. Например, в случае с биржевыми котировками или предсказанием погоды. Реализовать алгоритм, учитывающий многие нюансы и обеспечивающий большую точность сложно и дорого.

Решением задач экстраполяции является временной ряд, набор значений на некотором временном промежутке. Но для получения временного ряда для будущего времени часто необходим анализ предыдущих временных рядов. Анализ временных рядов основывается на предположении, что значения в ряде данных имеют общую закономерность [2].

### **1.1.2 Формулировка задачи и методы её решения**

Экстраполяция – это операция построения функции за пределами интервалов, на которых эта функция определена. Следовательно, продолжение функции и будет решением этой задачи. Но одно дело просто продолжить функцию, а другое сделать это достаточно достоверно.

Так как функции бывают разных видов, то соответственно существует несколько методов экстраполяции. Для лучшего результата необходимо проанализировать функцию, и выбрать подходящий метод:

### *Линейная*

Линейная экстраполяция подразумевает построение касательной к крайней точке известных данных. Данный метод может обеспечить хороший результат только для линейной функции и обычно на небольшом промежутке.

Если в окрестностях точки  $x^*$  экстраполировать две точки  $(x_{k-1}, y_{k-1})$  и  $(x_k, y_k)$ , то функция экстраполяции будет иметь вид (1).

$$y(x^*) = y_{k-1} + \frac{x^* - x_{k-1}}{x_k - x_{k-1}} * (y_k - y_{k-1}) \quad (1)$$

### *Полиномиальная*

Полиномиальная экстраполяция подразумевает построение кривой на основе многочлена, который может быть определён на конце известной функции или на основе всей функции. Осуществляется полиномиальная экстраполяция с помощью интерполяционного многочлена Лагранжа или методом Ньютона для получения рядов Ньютона.

Многочлен Лагранжа получается из суммы базисных полиномов разных степеней (2).

$$L(x) = \sum_{i=0}^n y_i l_i(x) \quad (2)$$

где базисные полиномы определяются по формуле (3).

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} \quad (3)$$

Но чем выше степень полинома, тем хуже он сходится, поэтому обычно не целесообразно использовать полиномы высоких порядков [3].

### *Коническая*

В этом методе, с помощью пяти точек, которые находятся рядом с границей определённых значений функции, строится конусное сечение. Если конусное сечение замкнутое, то будет экстраполирован эллипс или круг. В случае других функций, экстраполированные значения не обязательно замкнутся, но могут вернуться по оси X назад к определённой части функции.

### Лекальная экстраполяция

Данный метод экстраполяции обеспечивает хорошие результаты в случае, когда функция имеет экспоненциальный или похожий вид, с ускорениями или замедлениями.

#### 1.1.3 Применение нейронных сетей в решении задач экстраполяции

ИНС по ряду причин очень хорошо подходят для решения задач экстраполяции. Во-первых, нейронные сети обрабатывают информацию параллельно. Во-вторых, нейронная сеть сама способна находить закономерности, обобщать информацию. В-третьих, нейронная сеть обучаема. Поэтому, при каком-либо существенном изменении ситуации, нейронная сеть может изменить модель прогнозирования при переобучении, в то время как при заранее определённом методе экстраполяции придётся перестраивать алгоритм вручную.

Для решения задачи прогнозирования необходимо для начала построить нейронную сеть таким образом, чтобы она корректно выполняла свою задачу (Рисунок 1).

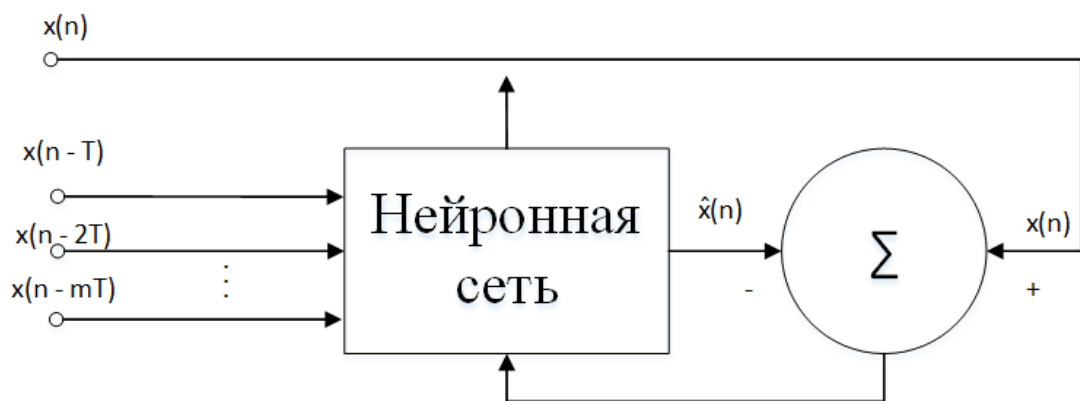


Рисунок 1 – Блочная диаграмма нелинейного прогнозирования

Для прогнозирования процесса  $x(n)$  необходимо иметь информацию об этом процессе в ограниченное дискретное число времени. Входами схемы будут  $m$  значений функции  $x(n)$  с интервалом времени между ними  $T$ . Поэтому ширина окна (т.е. длина интервала функции, который будет использоваться для предсказания) определяется числом входов  $m$  и расстоянием между ними  $T$ . Для

определения функции  $x$  в точке  $n$ , нейросеть будет использовать данные о значениях функции  $x$  в точках  $n-T, n-2T, \dots, n-mT$ .

Выходом ИНС будет предсказанное значение функции  $x(n)$ , которое будет идти как на выход, так и в саму ИНС для обучения. Данная ИНС может обучаться без учителя, т.к. она будет получать на вход реальные значения функции в прошлом, и, сравнивая со спрогнозированным результатом, корректировать свои ошибки (менять синаптические веса), т.е. со временем нейронная сеть, совершая ошибки, найдёт закономерность. Это позволяет данной ИНС быть очень эффективным инструментом нелинейного предсказания [4].

Очень часто хорошо обученная сеть обеспечивает меньшую погрешность, чем прямые методы. Например, в статье [5] описывается проблема загрязнения почв тяжёлыми металлами. Для прогнозирования, какой участок земли заражён, авторы решили использовать нейронную сеть и математический алгоритм. Задачей было предсказание концентрации свинца ( $Pb$ ) при условии наличия кадмия ( $Cd$ ). Результат испытаний приведён ниже (Рисунок 2)

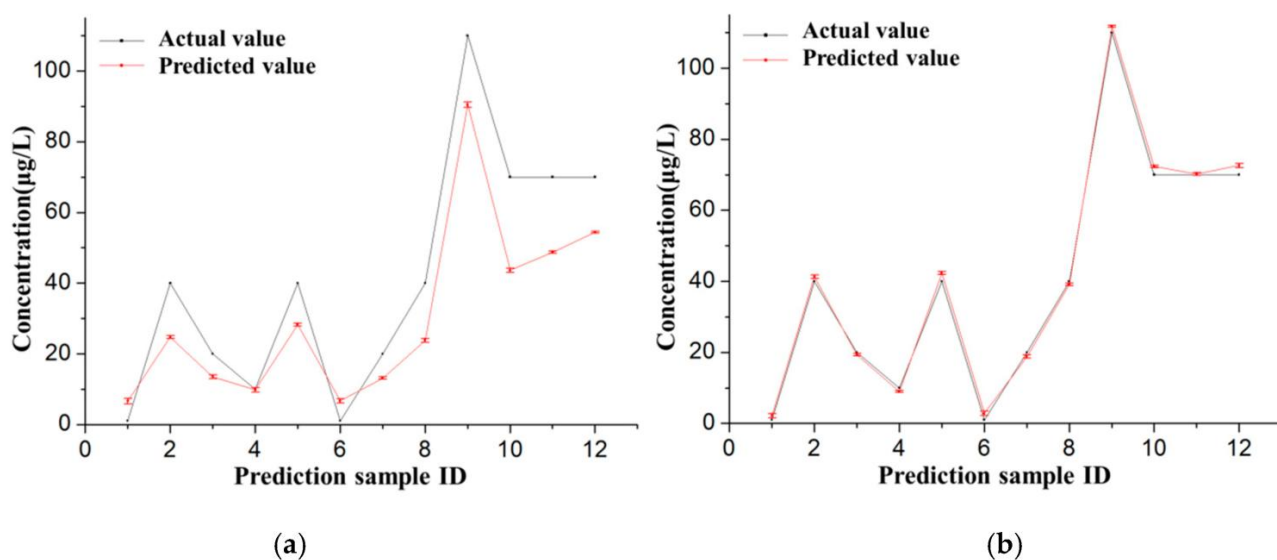


Рисунок 2 – Результаты предсказания прямым методом и ИНС наличия Pb в присутствии Cd

В качестве результата представлены два графика. Серым на графиках указано реальное значение, а красным предсказанное значение. На вертикальной оси указана концентрация Pb, а на горизонтальной - номер теста.

Как видно на графике слева, математические алгоритмы следовали тенденции, но давали погрешность в пределах десятков  $\mu\text{g/L}$ . В то время как решение, полученное с помощью нейронной сети на графике справа, имеет погрешность меньше единицы. Из этого следует, что в решении данной задачи нейронные сети справились гораздо лучше математических алгоритмов, ввиду того, что погрешность у ИНС в разы меньше, чем у прямого метода.

## 1.2. Описание работы искусственной нейронной сети

### 1.2.1 Общие положения

Искусственная нейронная сеть (ИНС) – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Элементарной единицей ИНС является искусственный нейрон [4].

Искусственный нейрон (Рисунок 3) получает на вход произведения признаков ( $X_i$ ) на веса их связей ( $W_i$ ), затем суммирует их на элементе  $\Sigma$ . Эта сумма поступает в качестве аргумента на функцию активации  $f$ . Значение этой функции и есть выход нейрона ( $Y$ ).

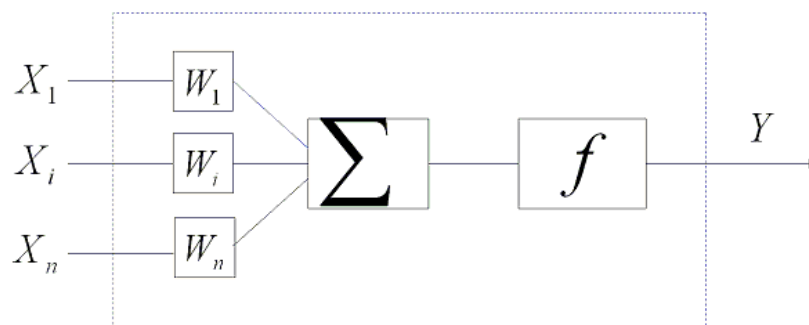


Рисунок 3 – Модель искусственного нейрона

В данной работе будут рассматриваться сети прямого распространения. ИНС прямого распространения – это ИНС, в которых сигнал распространяется строго от входного слоя к выходному. В обратном направлении сигнал не распространяется.

ИНС прямого распространения состоит из трёх типов слоёв нейронов, входных (input), скрытых (hidden) и выходных (output). Для корректной работы ИНС необходимо получать какие-либо признаки на вход, и выдавать какой-либо результат, соответственно ИНС может иметь два и больше слоёв. Но, как правило, ИНС содержит необходимое для решения задачи число скрытых слоёв, и каждый скрытый слой производит переход из одного пространства признаков в другой.

Каждый нейрон (кроме выходных) связан с другим нейроном, и эта связь имеет определённый вес, который может быть как положительным, так и отрицательным. Обучение ИНС подразумевает определение этих весов, а делается это с помощью различных алгоритмов, один из наиболее популярных алгоритмов это обратное распределение ошибки (backpropagation).

### 1.2.2 Принцип работы искусственных нейронных сетей

Как было сказано выше, ИНС состоит из трёх типов слоёв (Рисунок 4). ИНС может иметь различное количество входов, выходов и скрытых слоёв. Архитектура сети зависит от решаемой задачи.

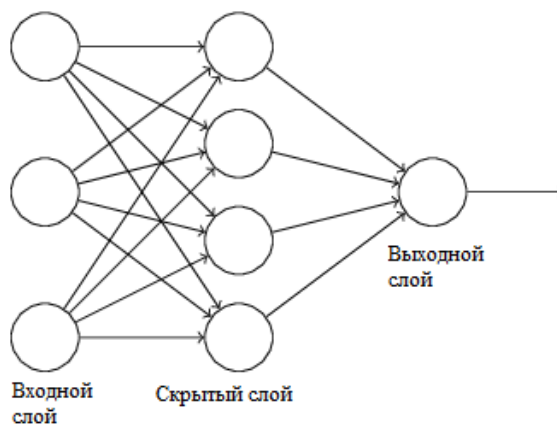


Рисунок 4 – Изображение искусственной нейронной сети

При обработке информации ИНС, данные подаются на входной слой, и с него идут на первый скрытый слой и при этом умножаются на вес связи. На первом скрытом слое каждый нейрон суммирует пришедшие на него произведения признаков и весов. Затем данная сумма подаётся как аргумент в функцию активации, и значение функции и будет являться выходным

значением нейрона. В случае если функцией активации будет являться сигмоида (Рисунок 5), то выходным значением будет число от 0 до 1.

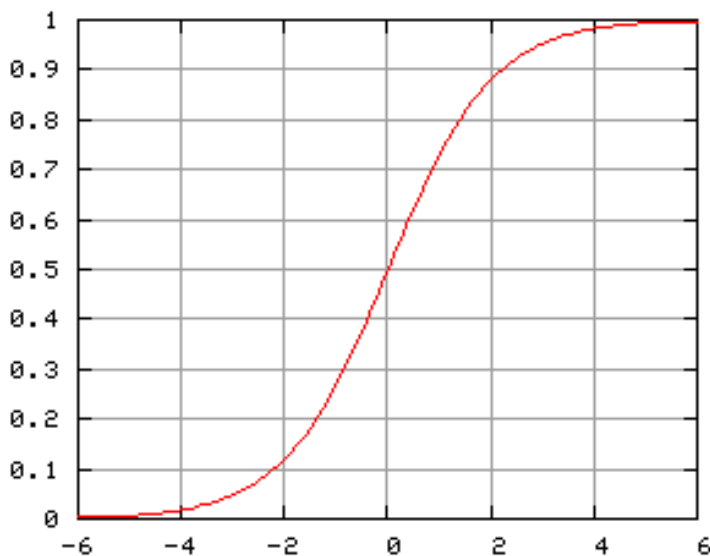


Рисунок 5 – График значений сигмоиды

Выдача нейроном выходного значения называется активацией нейрона. После того, как все нейроны первого скрытого слоя будут активированы, они так же передадут признаки, но уже новые, сформированные ими, следующему слою, и так вплоть до выходного.

Для корректной работы ИНС, её необходимо обучать. Обучение бывает с учителем и без учителя. В случае обучения с учителем предполагается, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Входной и целевой вектора называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть, и веса связей изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Обучение происходит до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня [4].

### 1.2.3 Обучение обратным распространением ошибки

Главная цель обучения искусственной нейронной сети (ИНС) состоит в том, чтобы при решении задач она совершала минимальное число ошибок. Одной из характеристик, позволяющих оценивать точность нейронной сети, является функция стоимости (cost function)  $J$ . Она оценивает то, насколько гипотезы ИНС ( $h$ ) отличаются от ожидаемых ответов ( $y$ ).

При решении задач линейной регрессии, т.е. задач, в которых требуется дать ответ в виде непрерывной величины (стоимость квартиры, число лет и т.д.),  $J$  высчитывается по следующей формуле (4).

$$J(W) = \frac{1}{2m} * \left[ \sum_{i=1}^m (h_W(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n W_j^2 \right], \quad (4)$$

где  $W$  – это матрица весов ИНС,  $m$  – число поданных для ИНС примеров,  $x$  – входные данные,  $\lambda$  – коэффициент регуляризации (будет рассмотрено ниже), а  $n$  это число слоёв (причём веса связей с *bias* не учитываются).

При решении задач логической регрессии, т.е. задач, в которых требуется распределить данные по каким либо классам (злокачественная опухоль или нет, определить марку автомобиля и т.д.),  $J$  высчитывается следующим образом (5).

$$J(W) = - \left[ \frac{1}{m} * \sum_{i=1}^m y^{(i)} \log(h_W(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_W(x^{(i)})) \right] + \quad (5)$$
$$+ \frac{\lambda}{2m} \sum_{j=1}^n W_j^2$$

Чем меньше  $J$ , тем лучше обучена сеть для данной выборки, соответственно целью обучения будет уменьшение  $J$ . Если  $J$  во время обучения увеличилась, значит, обучение прошло неверно и необходимо изменить параметры сети.

Если целью обучения является минимизация  $J$ , т.е. минимизация ошибки, то задача обучения состоит в поиске экстремума, точки минимума. Для нахождения точки минимума необходимо найти производную, а так как



функция  $J$  зависит от весов связей ( $W$ ), то необходимо найти частную производную для каждой связи. Одним из методов нахождения этих частных производных и является метод обратного распространения ошибки. После нахождения этих частных производных, необходимо изменить значения  $W$ , для этого используется алгоритм градиентного спуска, который позволяет менять значения весов  $W$  (6) [6].

$$W_j^{(iter)} = W_j^{(iter-1)} - \alpha \left( \frac{\partial}{\partial W_j^{(iter-1)}} + W_j^{(iter-1)} * \frac{\lambda}{m} \right), \quad (6)$$

где  $W_j$  слева – это новое значение весов, полученное на данной итерации из значений весов предыдущей итерации, т.е.  $W_j$  расположенных справа.

Значение  $\alpha$  отвечает за скорость обучения сети, т.е. за размер градиентного шага. Если шаги будут слишком малыми, то сеть будет долго обучаться, если шаги будут слишком большими, то сеть может начать расходиться [7].

Результат обучения может получиться разным, т.к. локальных минимумов у функции может быть несколько, соответственно результат обучения также зависит от начальных значений (Рисунок 6).

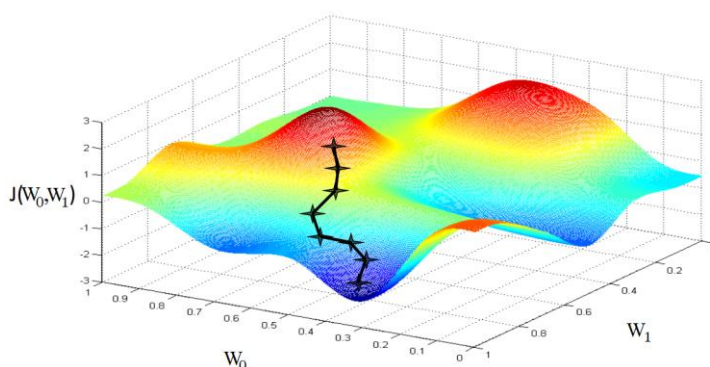


Рисунок 6 – Пример градиентного спуска

Результатом работы алгоритма обратного распространения ошибки будут частные производные весов. Суть работы алгоритма в следующем: на ИНС подаётся набор обучающих примеров в количестве  $m$ , с входными данными  $X$  и выходными данными  $Y$ . Нейронная сеть на основе полученных параметров  $X$  получает  $m$  наборов значений активаций каждого нейрона. Затем начинается

расчёт ошибок. Вычисления начинаются с конца, т.е. вычисляется разница между  $h$  и  $Y$  в данном примере. И далее разница, т.е.  $\delta$  вычисляется для каждого слоя, вплоть до входного слоя.

Данные вычисления удобнее производить в матричной форме, что значительно упрощает реализацию. Поэтому сначала рассчитываются значения активации на нейронах (7).

$$A_i = g(W_{i-1} * A_{i-1}), \quad (7)$$

где  $A$  – значения активаций на слое,  $i$  номер слоя,  $g$  – функция активации (в данном случае сигмоида), а  $n$  – число нейронов в слое. Как можно понять по формуле, она не подходит для первого слоя, поэтому в нём вместо  $A$  используют  $X$ .

Далее начинаются непосредственно вычисления самого алгоритма обратного распространения ошибки. Первым делом вычисляется ошибка выходного слоя (8).

$$\delta = h - Y \quad (8)$$

Как видно из формулы (8) ошибка последнего слоя это разность между ожидаемым результатом и полученным. После расчёта ошибки последнего слоя необходимо рассчитать ошибки для остальных слоёв (9).

$$\delta^i = (W^{(i)})^T \delta^{(i+1)} .* g(A^{(i)}), \quad (9)$$

где  $.*$  поэлементное умножение,  $(W^i)^T$  транспонированная матрица весов данного слоя,  $i$  номер слоя, а производная сигмоиды  $g(A^i)'$  равна (10).

$$g(A^{(i)})' = A^{(i)} * (1 - A^{(i)}) \quad (10)$$

В результате, после вычисления всех ошибок в примере  $m$ , начинают рассчитываться частные производные  $\Delta$ , где новое значение (с левой стороны) получается из суммы старого значения (с правой стороны) с произведением вектора ошибок на транспонированный вектор значений активаций нейронов (11) [8].

$$\Delta^{(i)} = \Delta^{(i)} + \delta^{(i+1)} (A^{(i)})^T \quad (11)$$

После выполнения данных расчётов для всех примеров  $m$ , значения  $\Delta$  делятся на  $m$ . Это и есть искомые частные производные  $\frac{\partial}{\partial w_i}$ , где  $i$  номер слоя. Данные частные производные меняют веса  $W$  в соответствии с формулой (6), на этом одна итерация обучения закончена, после этого проверяется функция  $J$ , и как было сказано в предыдущем пункте, если её значение увеличилось, то обучение прошло неправильно. Иначе обучение прошло правильно, и его можно продолжать, если необходимо улучшить результаты.

Для того чтобы избежать переобучения, т.е. подстройки значений нейросети применительно только для обучающей выборки (Рисунок 7), применяется регуляризация [9].

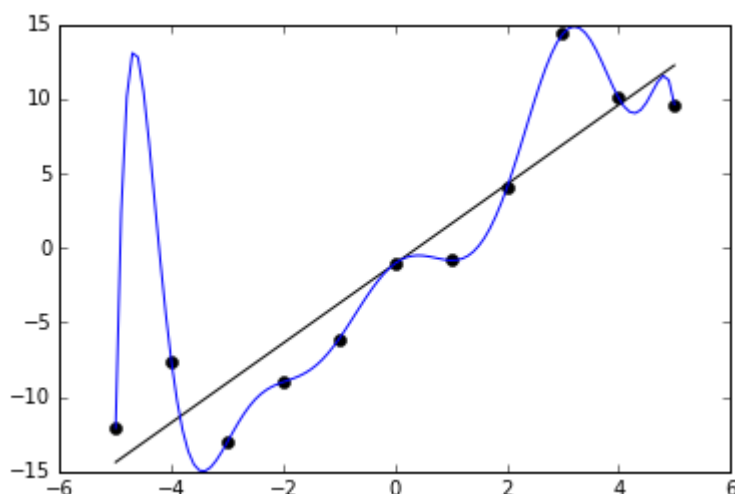


Рисунок 7 – Пример переобучения

В данном примере видно, что синяя кривая (т.е. обученная нейросеть) сильно привязана к обучающим примерам (чёрным точкам) и поэтому если брать входные данные не из обучающих примеров, ответ может сильно отличаться. Регуляризация уменьшает влияние самых сильных коэффициентов, для того, чтобы избежать переобучения.

#### 1.2.4 Метрики для определения результата

Две задачи, которые обычно решает искусственная нейронная сеть, это классификация и регрессия. И при решении задач, необходимо определить, верно ли нейросеть её решила или нет. Но для этих задач используются разные

метрики. Поэтому выберем одну метрику для классификации, одну для регрессии [10].

Для классификации возьмём F-метрику. Данная метрика лучше процента правильных ответов тем, что позволит избежать проблемы, при дисбалансе классов. Первым главным критерием в ней являются полнота (recall) (12).

$$recall = \frac{TP}{TP+FN}, \quad (12)$$

где TP (true positive) – число верных ответов при равенстве признака 1, а FN (false negative) – число неверных ответов при верном ответе 0.

Вторым главным критерием является точность (precision) (13).

$$precision = \frac{TP}{TP+FP}, \quad (13)$$

где FP (false positive) – число неверных ответов, при верном ответе 1. F-мера, или ещё её называют просто F-1 рассчитывается следующим образом (14).

$$F1 = \frac{2*precision*recall}{precision+recall}, \quad (14)$$

Чем ближе данный показатель к 1, тем более полно классификатор описывает модель.

В качестве метрики для определения результата обучения в случае задачи регрессии, будет использована нормированная среднеквадратичная ошибка, она же коэффициент детерминации (15).

$$R^2 = 1 - \frac{\sum_{i=1}^l (h(x_i) - y_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2}, \quad (15)$$

где  $l$  – число примеров,  $h(x_i)$  - гипотеза,  $y_i$  - истинный результат,  $\bar{y}$  - среднее значение истинного результата.

Если значение  $R^2$  близко к 1, значит модель прогнозирования покрывает почти всю систему. Значение от 0.5 считается уже приемлемой моделью, а выше 0.8 хорошей моделью. Если значение близко к 0, то прогнозирование не отличается от константного. А если значение отрицательное, то это говорит о неадекватности модели, среднее является лучшим ответом, чем то, что даёт модель.

## 1.3. Обзор существующих решений

### 1.3.1 Lasagne

Lasagne – библиотека для создания и обучения нейронных сетей с помощью Theano (библиотеки численного вычисления, представляющей собой препроцессор на языке python для системы вычислений с многомерными массивами данных (тензорами), сочетающей в себе математические пакеты Mathematica и MATLAB). Для её работы также требуется среда для python, т.к. именно этот язык является средством взаимодействия с нейросетью. Также требуется компилятор C и библиотека для расчётов numpy/scipy и BLAS.

В Lasagne сеть настраивается непосредственно в .py файле, т.е. например входной слой задаётся следующим образом: задаётся форма слоя:

Batchsize – число наборов, которые будут взяты для одного обучения, например при значении 100 и при 1000 примерах, нейросеть будет обучаться наборами по 100 примеров и так 10 раз, пока не кончатся данные. Это, как правило, позволяет ускорить обучение и использовать меньше памяти во время работы, но может отрицательно сказаться на результате обучения.

Channels – число каналов. Как правило, значение больше 1 используется при работе с изображениями, когда нужно 1 пиксель разбить на 3 канала: красный, зелёный и синий.

Rows и Columns – число строк и столбцов матрицы, собственно размер слоя.

А так же задаётся переменная, куда будут поступать результаты работы слоя. В скрытом и выходном слое, так же задаётся тип функции активации.

Преимуществом данной библиотеки является простота модификации сети и влиять на процесс обучения. Недостатком же является меньшая скорость обучения, чем у не символьных фреймворков.

### 1.3.2 Caffe

Caffe (Convolution Architecture For Feature Extraction) - это среда для глубокого обучения. Она разработана Berkeley AI Research (BAIR) Янцинем

Цзя, во время получения PhD в Калифорнийском университете в Беркли. Caffe выпускается по лицензии BSD 2-Clause. Данный фреймворк бесплатен, написан на c++, а интерфейс реализован с помощью языка python.

Для работы нейросети в данном фреймворке необходимо описать её в конфигурационных файлах: solver (где указываются общие характеристики, как число итераций, алгоритм обучения и прочее) и net (где настраиваются слои нейросети).

Структура слоёв сети представлена блобами (Рисунок 8) [11].

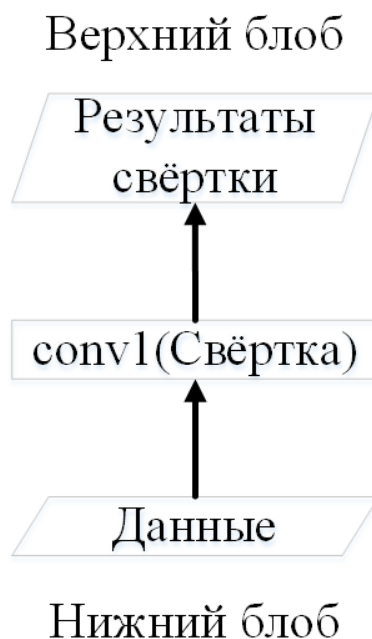


Рисунок 8 – Структура слоя в Caffe

Где нижний блоб – это входные данные, а верхний – выходные, т.е. в нижнем блобе хранятся данные до свёртки, а в верхнем после неё.

Данный фреймворк так же отличается от нашего тем, ориентирован на работу со свёрточными сетями, и соответственно ожидается, что будут использоваться слои свёртки и подвыборки. Данные особенности продиктованы тем, что он разработан для работы с изображениями, и соответственно с большим числом признаков и данных. Для текстовых данных используется формат hdf5.

В Caffe так же есть полносвязный слой, он обозначается в графе type как INNER\_PRODUCT. Данный слой принимает на вход вектор значений, и его

выход так же является вектором значений, где вход и выход также устанавливаются в графах `bottom` и `top` соответственно. Как и в остальных слоях, для полносвязного слоя устанавливается тип функций активаций и скорость обучения. Но в отличие от свёрточных слоёв, у полносвязного нет настроек свёртки, т.е. размеров фильтра и интервалов между ними.

Достоинствами Caffe являются высокая производительность, большое число поддерживаемых технологий. Одним из самых больших недостатков является невозможность влиять на ход обучения, т.е. пользователь задаёт параметры обучения и сети в конфигурационных файлах, и не имеет возможности динамического программирования, т.е. совершать какие либо действия с данными в ходе обучения, как у Lasagne или Keras.

### 1.3.3 Keras

Keras - это высокоуровневый API для работы с нейронными сетями, написанный на Python и способный работать поверх TensorFlow, CNTK или Theano. Приоритетом является компактность, модульность и расширяемость сетей.

Согласно концепции, Keras является скорее интерфейсом, чем сквозной системой машинного обучения. Keras предоставляет высокоуровневый, более интуитивный набор абстракций, который делает простым формирование нейронных сетей не зависимо от используемой на нижнем уровне библиотеки научных вычислений.

Параметры для данного API настраиваются в конфигурационных файлах `json`, т.е. с какой точностью будут числа с плавающей запятой, метод преобразования изображений и т.д.

Далее на python описывается модель нейронной сети. Сначала импортируются необходимые функции из Keras, а затем уже последовательно описывается модель ИНС, т.е. число входов и их "форма", последовательно описываются слои, и переменная, куда передать их результат, и таким же образом описываются выходы.

Производительность Keras зависит в основном от того, функции какой библиотеки он будет реализовывать, но за счёт высокоуровневого API обучение сетей становится проще и управляемее для пользователя, но снижается производительность.

#### 1.4. Исследование методов улучшения результата прогнозирования

##### 1.4.1. Прунинг

Во всех приведённых выше нейронных сетях, отсутствуют некоторые технологии, которые могут улучшить результаты. Первая это прунинг.

Прунинг – это удаление связей с весами, близкими к 0. Данный метод позволяет избавиться от лишних связей и тем самым оптимизирует работу нейронных сетей [12]. Но данную процедуру стоит проводить осторожно, т.к. она призвана убирать избыточные связи, но можно таким образом и повредить нейросеть. Близкие к 0 веса определяются в соответствии со средним значением на слое. Прунинг позволяет ускорять работу сетей и уменьшать их размер, однако данная процедура не получила ещё широкого распространения.

Алгоритм прунинга достаточно прост (Рисунок 9).

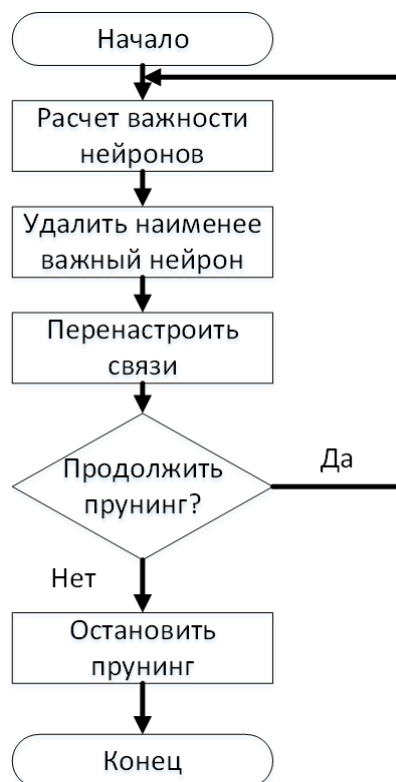


Рисунок 9 – Алгоритм прунинга



Вероятно, прунинг не так распространён из-за уменьшения точности при удалении всё-таки нужной связи, и так же из-за трудности включения в уже существующие фреймворки. Наибольшую эффективность прунинг показывает в свёрточных сетях при глубоком обучении. Но для применения данного метода необходимо, чтобы использовались разреженные матрицы, т.к. в полных матрицах невозможно удаление каких-либо связей.

### 1.4.2. Разреженные матрицы

Использование разреженных матриц, позволит создавать не только ИНС с плотными матрицами (где большинство элементов не 0), но так же позволит использовать прунинг. Но недостатком является то, что на аппаратном уровне во многих GPU не поддерживаются вычисления разреженных матриц, в то время как расчёты обычных матричных операций поддерживаются [13].

Использование разреженных матриц при этом на точность не влияет отрицательно. Особенно актуально использование разреженных матриц в больших системах. Как видно из исследования [14], разреженные свёрточные сети (синяя сплошная линия), сходятся даже быстрее плотных (зелёная штриховая линия) (Рисунок 10).

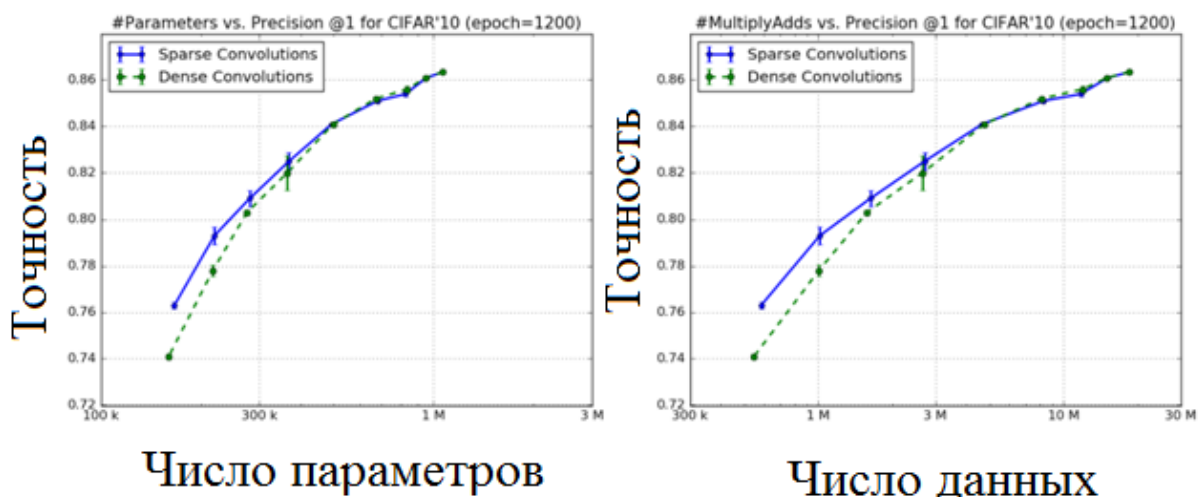


Рисунок 10 – Результаты обучения плотных и разреженных сетей на наборе CIFAR-10

Поэтому использование разреженных матриц и прунинга оправданы как при большом количестве параметров и данных, так и при малом. При большом

количестве параметров или данных ИНС меньше в размерах, и быстрее производит вычисления, а при малом количестве данных или параметров, обучение идёт быстрее.

### 1.4.3. Динамический коэффициент обучения

Динамическое снижение коэффициента – простой, но достаточно эффективный метод. Если с обучение с крупным шагом попало в минимум, и сеть расходиться при дальнейшем обучении, то с помощью динамичного снижения коэффициента обучения можно более мелкими шагами приблизиться к минимуму (Рисунок 11) [15].

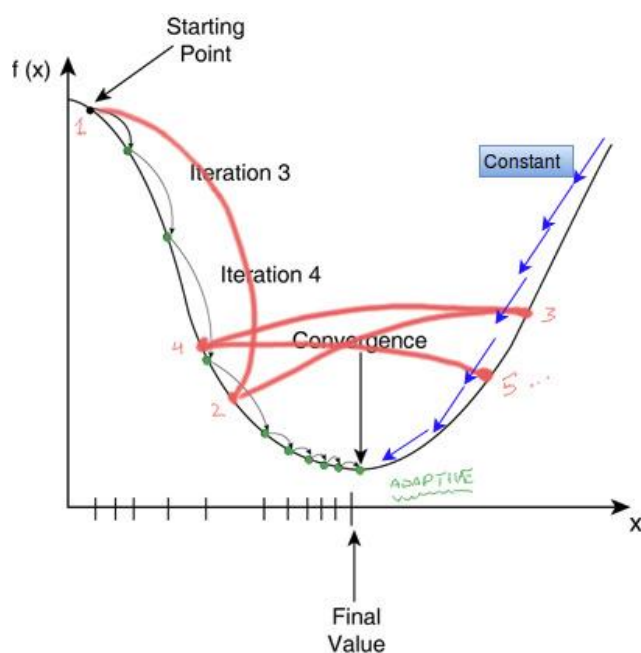


Рисунок 11 – Принцип работы динамического коэффициента обучения

На данном рисунке описано 3 ситуации:

- Первая ситуация отображена красным, это происходит, если коэффициент обучения слишком велик. В данном случае сеть будет пытаться приблизиться к минимуму, но будет постоянно перескакивать его, из-за слишком большого шага.
- Вторая ситуация отображена синим, это происходит, если коэффициент обучения слишком мал. В данном случае сеть будет слишком маленькими шагами приближаться к минимуму, и медленнее достигнет минимума.

- Третья ситуация отображена зелёным, это адаптивный коэффициент обучения. В случае, когда шаг приведёт к уходу от минимума, шаг уменьшается. И таким образом в начале прогресс быстро достигается большими шагами, а затем ближе к минимуму шаги начинают уменьшаться.

#### 1.4.4. Инициализация Ксавьера

Инициализация Ксавьера – это случайная генерация начальных значений весов по следующей формуле (16)[16].

$$W_i = U\left(-\frac{\sqrt{6}}{\sqrt{Count_{in} + Count_{out}}}, \frac{\sqrt{6}}{\sqrt{Count_{in} + Count_{out}}},\right) \quad (16)$$

где  $W_i$  – Вес связи нейрона,  $U$  – равномерное распределение,  $Count_{in}$  – число входов у нейрона,  $Count_{out}$  – число выходов у нейрона.

Данная инициализация позволяет получать не 0 начальные веса, что ускоряет обучение [17]. Данная инициализация так же позволяет получать различные результаты каждый раз при инициализации новой сети, т.к. из-за разных начальных условий, сеть может найти другой локальный минимум, хуже или лучше.

#### 1.4.5. Перемешивание данных в ходе обучения

Перемешивание данных позволит ускорить обучение, особенно при большом числе эпох, т.к. подряд идущие множество раз одинаковые данные менее эффективны, чем подающиеся случайным образом те же самые данные [18]. Но при этом данный подход может немного замедлить скорость вычислений, т.к. на операцию перемешивания тоже тратятся ресурсы.

### 3 РЕАЛИЗАЦИЯ

#### 3.1. Реализация библиотеки обучения нейронной сети

Как было определено выше, работа с нейронной сетью будет производиться на уровне слоёв. Соответственно каждый слой хранит свои веса связей, которые представлены в виде матрицы размером число выходов на число входов плюс один (смещение). При прямом прохождении нейронной сети (т.е. когда она решает задачу) производится только расчёт значений активации нейронов, т.е. производится умножение матрицы весов на вектор значений активации, или в случае с входным слоем, на вектор входных значений.

Обучение идёт то число итераций, которое задал пользователь, либо до тех пор, пока сеть не начнёт расходиться. Расходиться сеть или нет, определяется функцией ошибки, если она стала больше, чем в прошлой итерации, то библиотека вернёт приложению 1, где может быть решено, закончить обучение, или например уменьшить коэффициент обучения. Само обучение начинается с расчёта значений активаций на слоях, и затем начиная с последнего слоя, идёт расчёт ошибок, за счёт которых будут получены частные производные. Далее, с помощью данных производных изменяются значения весов связей. Все вычисления выполняются с помощью разреженных матриц.

В вычислении функции стоимости и при перерасчёте значение весов связей используется коэффициент регуляризации, который помогает уменьшить влияние переобучения. Нейросеть ещё содержит возможность перемешивать данные в ходе обучения между итерациями, для этого необходимо задать число итерации и размер данных, которые будут перемешиваться. Также библиотека содержит в себе функции по вычислению ошибок 2 метриками: нормированной среднеквадратичной ошибкой, и F1 мерой.

### 3.2. Реализация приложения по работе с библиотекой

В результате, было разработано приложение, реализующее указанные в проектировании функции. Начальное окно выглядит следующим образом (Рисунок 12).

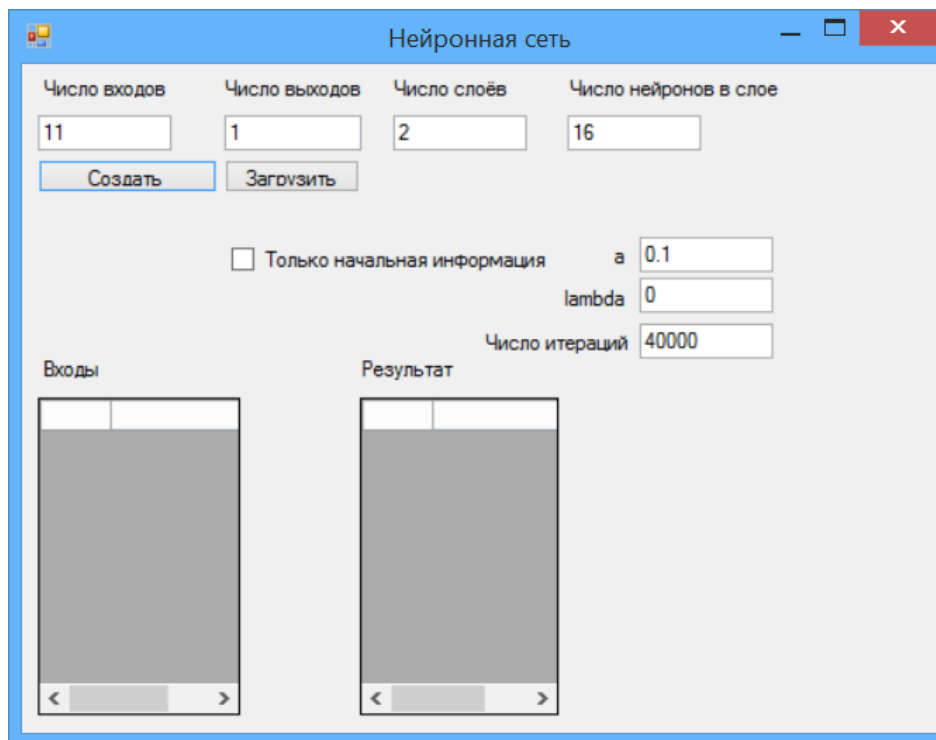


Рисунок 12 – Начальное окно приложения

Наверху задаются параметры для новой нейронной сети, либо можно загрузить уже готовую. После её загрузки открываются новые опции (Рисунок 13).

Во-первых, открывается возможность получения ответа от сети ручным вводом значений. Во-вторых открывается возможность обучения сети. Параметры для обучения задаются в правой стороне.

Среди параметров есть:  $a$  – коэффициент скорости обучения,  $\lambda$  – коэффициент регуляризации (защита от переобучения). И так же задаётся число итераций.

Следующей функцией является проверка на 1 примере с построением графика, она больше нужна для наглядности. Ещё можно спрогнозировать результат на основе только начальной информации.

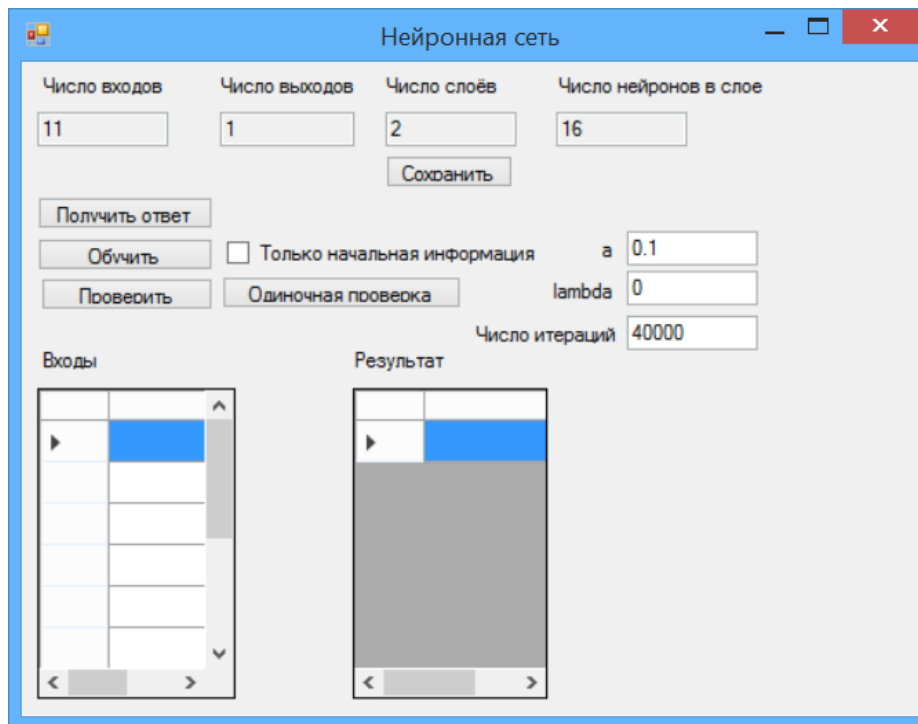


Рисунок 13 – Состояние приложения после создания сети

При обучении пользователь теряет возможность воздействия на программу, кроме остановки обучения (Рисунок 14).

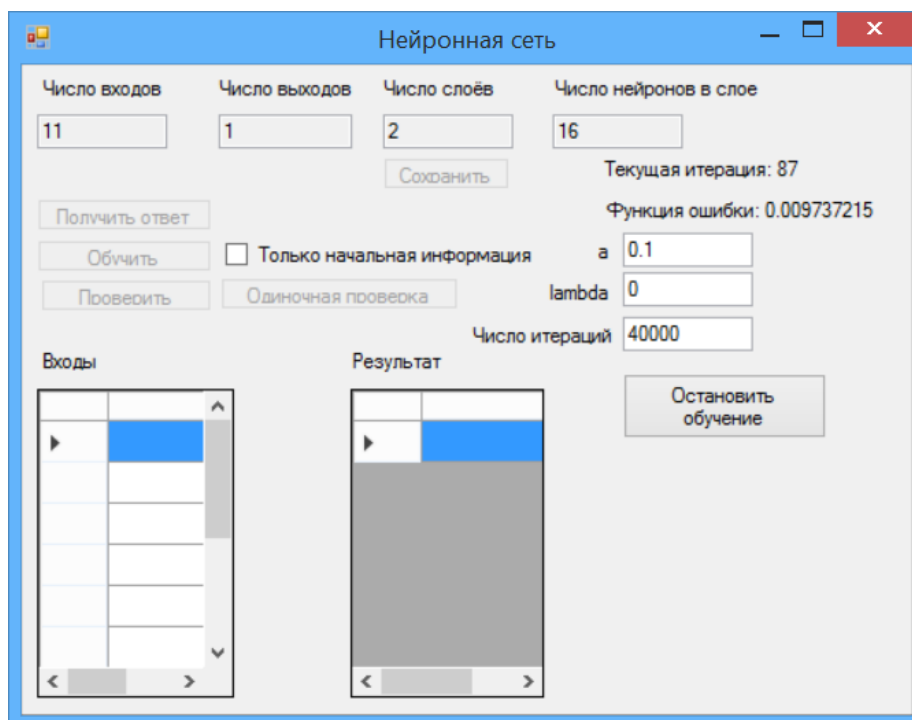


Рисунок 14 – Приложение во время обучения

Во время обучения, пользователь также видит, какая сейчас итерация обучения, и чему равна функция стоимости, что позволит понять, на какой стадии обучение и с какой скоростью оно идёт.

### 3.3. Реализация приложения по нормализации данных

При запуске приложения, пользователь может загрузить конфигурацию, либо настроить её вручную (Рисунок 15).

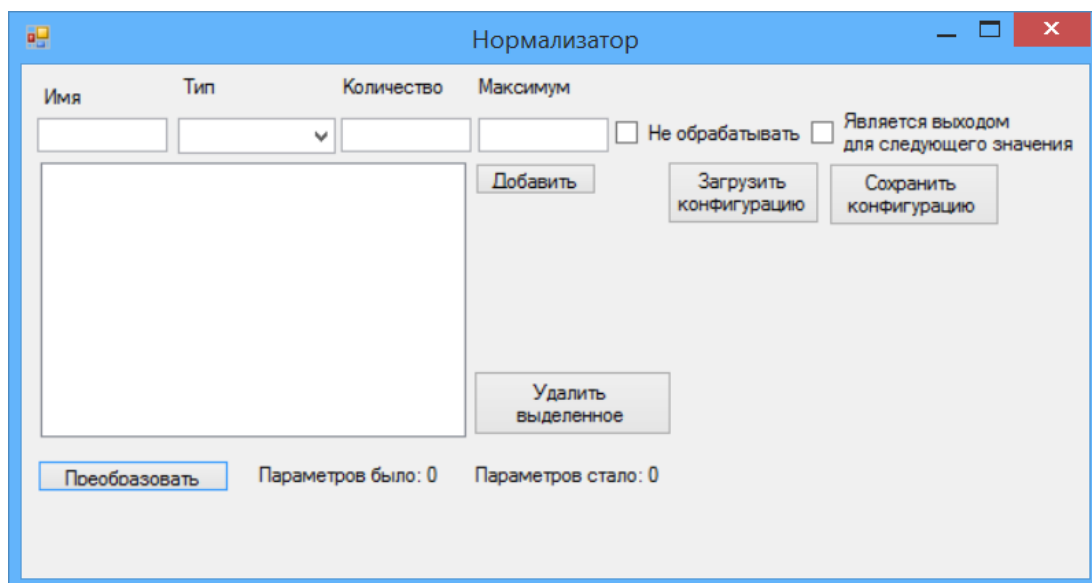


Рисунок 15 – Начальное состояние приложения по нормализации

Данные о параметрах заносятся в список, откуда их можно удалить, выделив параметр для удаления и нажав кнопку удалить выделенное (Рисунок 16).

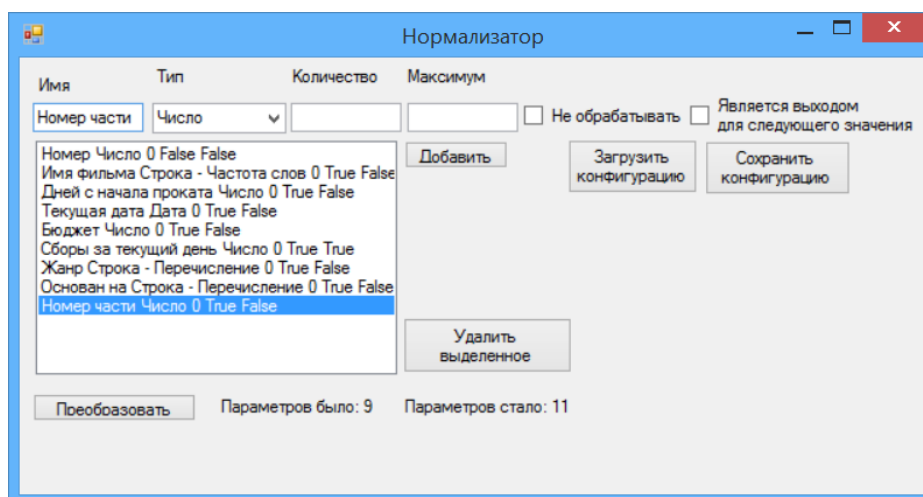


Рисунок 16 – Заполненные параметры

Поля имя и тип являются обязательными, остальные поля обрабатываются по умолчанию следующим образом: количество если не задано, то создаётся 1 запись. Максимум если не задан, то он определяется на основе входных данных. Если максимум задан, а данные превышают его, то

они берутся за 1. Также видно, что приложение информирует, сколько нужно подать параметров, и сколько будет в новом файле.

### **3.4. Тестирование библиотеки на Proben1**

Работоспособность библиотеки проверим с помощью бенчмарка Proben1 [20]. Первый тест Cancer1, задача классификации, 9 входов, 2 выхода. Будем решать однослойной сетью размером 10 нейронов. Сначала обучим, коэффициент обучения 0.1, без регуляризации, 40000 итераций. В результате задача была успешно решена с результатом по F-1 метрике: 0.93, что близко к 1.

Следующая задача из данного набора, которую попробуем решить, это Card1. Параметры возьмём идентичные, кроме того, что в данной задаче 51 вход, а выходов так же 2.

Времени на обучение задаче Card1 было затрачено гораздо больше. Результат составил 0.72 по метрике F-1 данный результат является приемлемым. В Proben1 есть ещё больше задач, но решив 2, было доказано, что нейронная сеть работает.



## **5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И ЭНЕРГОСБЕРЕЖЕНИЕ**

### **5.1 Введение**

Искусственные нейронные сети, не смотря на давнее появление, широкое коммерческое развитие получили недавно. Особенно широко они распространились в области распознавания (голоса, изображения) и в области прогнозирования временных рядов. Крупные компании вроде Google, Amazon, Microsoft и Яндекс имеют собственные подразделения и разработки в данной области.

Фирмам поменьше приходится искать готовые решения или заказывать собственную разработку. А спрос на нейронные сети высок, т.к. они позволяют автоматизировать многие процессы, которые автоматизировать с помощью обычных алгоритмов сложно и дорого [22].

Целью данного раздела является разработка оптимального решения для разработки приложения, при котором эффективно будут использоваться ресурсы пользователей, а на разработку будет потрачено наименьшее количество средств.

Для этого необходимо проанализировать конкурентные решения, понять как сделать эффективнее, найти те характеристики, изменение которых приведёт к повышению конкурентоспособности. В результате должна быть достигнута максимальная финансовая и ресурсная эффективность.

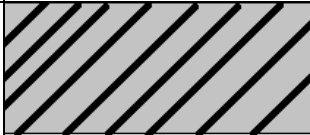



### **5.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **5.2.1 Потенциальные потребители результатов исследования**

Сперва необходимо провести анализ потенциальных потребителей нашей разработки, т.е. определить целевой рынок. Для этого необходимо найти заинтересованные сегменты. А для поиска сегментов нужно сегментировать

рынок. Произведём сегментирование по 2 критериям, отрасль и размер компании. Наиболее перспективными отраслями в нейронных сетях являются медицина, финансы и развлечения (Таблица 1).

Таблица 1 – Карта сегментирования рынка

		Отрасль		
		Медицина	Развлечения и искусство	Финансы
Размер компании	Крупные			
	Средние			
	Мелкие			



– АКК



– Nvidia



– Paypal

АКК – Американская коллегия кардиологии.

В среде средних компаний Nvidia представлена её дочерними компаниями.

Оптимальной стратегией будет выбрать сектор, куда сосредоточить основные усилия. Крупные компании уже имеют поставщиков услуг в своих крупных проектах. Поэтому выберем сегмент рынка средних компаний, в области развлечений и искусств, т.к. данная ниша имеет пока что низкую конкуренцию и спрос на нейросетевые услуги начинает нарастать, т.к. остальные методы не могут решать те задачи, которые решают нейронные сети.

### 5.2.2 Анализ конкурентных технических решений

В качестве конкурентных решений задачи прогнозирования временных рядов были выбраны следующие варианты:

- Прогнозирование с помощью нейронных сетей (данный проект)
- Прогнозирование на основе генетических алгоритмов

- Прогнозирование с помощью полиномиальной экстраполяции

Построим оценочную карту нашего и конкурентных решений (Таблица 2).

Таблица 2 – Оценочная карта

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
Удобство в эксплуатации	0.05	5	4	5	0.25	0.15	0.25
Потребность в ресурсах памяти	0.15	3	4	4	0.45	0.45	0.6
Функциональная мощность	0.1	5	4	1	0.5	0.4	0.1
Простота эксплуатации	0.05	4	5	5	0.2	0.15	0.25
Потребность в мощности ЭВМ	0.15	2	2	3	0.3	0.3	0.45
Скорость работы	0.15	3	2	4	0.45	0.3	0.6
<b>Экономические критерии оценки эффективности</b>							
Уровень проникновения на рынок	0.1	4	2	5	0.4	0.2	0.5
Послепродажное обслуживание	0.15	5	5	2	0.75	0.75	0.3
Готовность продукта	0.1	3	2	5	0.3	0.2	0.5
Итого	1	34	30	34	3.4	2.9	3.55

По результатам теперь рассчитаем конкурентоспособность нашего решения по отношению к двум другим. Сначала посчитаем по отношению ко 2 (17).

$$K_{12} = \frac{K_1}{K_2} = \frac{3.4}{2.9} = 1.17 \quad (17)$$

Теперь по отношению к 3 (18).

$$K_{13} = \frac{K_1}{K_3} = \frac{3.4}{3.55} = 0.96 \quad (18)$$

Как видно, по большинству параметров наше решение превосходит второе по таким важным показателям, как скорость работы, удобство и функциональная мощность, но по показателям потребность в ресурсах памяти и скорости работы уступает третьему решению.

### 5.2.3 Технология QuaD

По технологии QuaD построим оценочную карту нашей новой разработки (Таблица 3).

Таблица 3 – Оценочная карта

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Среднее взвешенное (5*2)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
Удобство эксплуатации	0.05	90	100	0.9	0.045
Потребность в ресурсах памяти	0.15	60	100	0.6	0.09
Функциональная мощность	0.1	100	100	1	0.1
Простота эксплуатации	0.05	70	100	0.7	0.035
Потребность в	0.15	30	100	0.3	0.045

мощности ЭВМ					
Скорость работы	0.15	55	100	0.55	0.0825
<b>Показатели оценки коммерческого потенциала разработки</b>					
Уровень проникновения на рынок	0.1	70	100	0.7	0.07
Послепродажное обслуживание	0.15	90	100	0.9	0.135
Готовность продукта	0.1	60	100	0.6	0.06
Итого:	1	625	900	6,25	0,6625

Оценка качества и перспективности по QuaD следующим образом (19).

$$P_{cp} = \sum V_i B_i = 0.6625 \quad (19)$$

Исходя из этой цифры, делаем вывод, что перспективность данного проекта выше среднего.

#### 5.2.4 SWOT-анализ

	<p><b>Сильные стороны:</b></p> <ol style="list-style-type: none"> <li>1. Удобство встраивания в систему</li> <li>2. Широкая область применения</li> <li>3. Простота масштабирования и обслуживания</li> <li>4. Отлаженность технологии</li> <li>5. Простота использования</li> </ol>	<p><b>Слабые стороны:</b></p> <ol style="list-style-type: none"> <li>1. Большие требования к производительности ЭВМ</li> <li>2. Не является распространённым решением на рынке</li> <li>3. Низкая скорость обучения</li> <li>4. Чрезмерное потребление памяти при большом числе данных</li> </ol>
<p><b>Возможности</b></p> <ol style="list-style-type: none"> <li>1. Повышение популярности нейронных сетей в</li> </ol>	<p><b>В1Сил1Сил2Сил5</b></p> <p>Благодаря простоте интеграции и использования, а так же широкой области</p>	<p><b>В1Сл2</b></p> <p>Со временем недоверие к неизвестной технологии пропадёт.</p> <p><b>В2Сл1Сл3</b></p>

<p>медиа</p> <p>2. Автоматизация сложных процессов.</p> <p>3. Увеличение мощностей ЭВМ</p> <p>4. Большая потребность в улучшении текущих результатов</p>	<p>применений, среди компаний, желающих быть передовыми (Сбербанк[23], Магнит).</p> <p>V2Сил1Сил2Сил3</p> <p>Ранее не автоматизированные процессы, теперь автоматизируются с помощью нейронных сетей, т.к. ранее алгоритмы не могли с ними справиться</p> <p>V3Сил2Сил4</p> <p>Благодаря увеличению мощностей ЭВМ расширяется спектр решаемых задач и появляется манёвр для улучшения технологии.</p> <p>V4Сил1Сил2Сил4</p> <p>Даже при наличии уже работающего решения, компании предпочтут выбрать оптимальный проект.</p>	<p>Так как обучение происходит не всё время, а только при изменении системы, то компании это не составит проблемы.</p> <p>V3Сл1Сл3Сл4</p> <p>За счёт улучшения характеристик ЭВМ, время затрачиваемое на обучение уменьшится.</p> <p>V4Сл2</p> <p>По мере распространения нейронных сетей, компании будут переходить на оптимальные решения.</p>
<p><b>Угрозы</b></p> <p>1. Высокая распространенность конкурентных решений</p> <p>2. Неблагоприятная</p>	<p>У1Сил1Сил4</p> <p>При наличии уже существующего решения, компания будет с меньшим желанием переходить на</p>	<p>У1Сл2</p> <p>Ввиду не распространённости решения, компания может отказаться от работы из-за недоверия</p>

экономическая обстановка	новое. У2Сил1Сил2Сил3	У2Сл1Сл4 В неблагоприятной обстановке,
3. Растущие требования компаний	В условиях экономических проблем, компании менее	компании не будут увеличивать мощности своего оборудования.
4. Ужесточение законодательства	склоны к риску, связанному с переходом на новое решение	У3Сл1Сл3 При небольшом числе данных и слабом оборудовании обучение
	У3Сил2Сил5 Решения конкурентов тоже совершенствуются, и всегда	будет идти недостаточно быстро и качественно.
	есть риск потерять клиента. У4Сил4	У4Сл2 Из-за небольшого наработанного
	Для нейронных сетей необходимы массивы	опыта в использовании, ещё не нормализована юридическая
	данный, а работа с личными данными требует	сторона использования данных.
	определённых процедур, и при расширении понятия	
	персональные данные обучение будет затруднено.	

Из анализа можно сделать вывод, что необходимо выходить на рынок, пока там господствуют более слабые решения, и находить компании, заинтересованные в автоматизации аналитических процессов.

### **5.3 Определение возможных альтернатив проведения научных исследований**

Для оценки ресурсоэффективности разрабатываемого проекта, необходимо проанализировать возможные варианты проекта. Воспользуемся методом синтеза вариантов из особенностей морфологии проекта. Основные характеристики системы и их варианты приведены ниже (Таблица 4).

Таблица 4 – Морфологическая матрица

	1	2	3
А. Фреймворк	Собственный	Theano	СЕТК
Б. Язык программирования	Java	C++	C#
В. Тип сети	Прямого распространения	Сверочная	Рекуррентная
Г. Реализация математики	Собственная	MathNet	OpenBlas

В качестве фрейворка было решено написать собственный, т.к. это позволит получить больший контроль над процессом работы и упростит внедрение новых технологий.

В качестве языка программирования был выбран C#, так как он позволяет создавать приложения с удобным пользовательским интерфейсом, и поддерживает множество уже готовых решений для разных задач.

В качестве типа нейронной сети была выбрана прямого распространения, т.к. в нашей задаче, число входов недостаточно велико, для использования сверочных и рекуррентных.

Реализация математики была выбрана MathNet, за счёт простоты внедрения и упрощения логики программы, в сравнении с ручным написанием математики.

## 5.4 Планирование научно-исследовательских работ

### 5.1 Структура работ в рамках научного исследования

Процесс проведения работ в рамках научного исследования имеет определённую структуру и может быть разбит на этапы, которые включают в себя список работ и их исполнителей. Перечень этапов, работ и распределение исполнителей приведены ниже (Таблица 5).

Таблица 5 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№	Содержание работ	Должность
----------------	---	------------------	-----------



			исполнителя
Определения задач для достижения цели	1	Определение и формулировка задач	Руководитель
	2	Календарное планирование работ по теме	
Аналитический обзор	3	Подбор и изучение материалов по теме	Инженер
	4	Изучение существующих продуктов	
	5	Выбор технических решений	Руководитель, инженер
Проектирования библиотеки и приложения	6	Составление общей структуры программ	Инженер
	7	Проектирование структуры библиотеки	
	8	Проектирование структуры приложения по работе с библиотекой	
	9	Проектирование структуры приложения нормализации данных	
Реализация	10	Программирование спроектированных программ	Инженер
Тестирование	11	Проверка результатов на бенчмарке Proben1	Инженер
	12	Проверка результатов на кассовых сборах фильмов	Инженер
Оформление работы	13	Описание проделанной работы.	Руководитель, инженер

## 5.2 Определение трудоемкости выполнения работ

Для определения трудоёмкости работ, необходимо оценить минимальное и максимальное затраченное на работу время. Произведём расчёт ожидаемой трудоёмкости это с помощью формулы (20).

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5} \quad (20)$$

Учитывая, что исполнитель один, ускорить работу за счёт распараллеливания не представляется возможным.

## 5.3 Разработка графика проведения научного исследования

Построим таблицу временных показателей научного исследования (Таблица 6). Для расчёта длительности работ в календарных днях рассчитаем коэффициент календарности (21)[24].

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}} = \frac{365}{365 - 118} = 1.48 \quad (21)$$

Таблица 6 – Временные показатели научного исследования

Название работы	Трудоёмкость работ									Исполнители	Длительность работ в рабочих днях $T_{pi}$	Длительность работ в календарных днях $T_{ki}$							
	$t_{min}$ , чел-дни			$t_{min}$ , чел-дни			$t_{ож\ i}$ , чел-дни					Рук	Инж	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3										
Определение задач	1	1	1	2	2	2	1.4	1.4	1.4	+	-	1	1	1	1	1	1		
Календарное планирование	1	1	1	2	2	2	1.4	1.4	1.4	+	-	1	1	1	1	1	1		
Подбор и изучение	2	2	2	5	5	5	3.2	3.2	3.2	-	+	3	3	3	4	4	4		

материалов																	
Изучение существующих продуктов	1	1	1	3	3	3	1.8	1.8	1.8	-	+	2	2	2	3	3	3
Выбор технических решений	2	2	2	3	3	3	2.4	2.4	2.4	+	+	2	2	2	3	3	3
Составление общей структуры	2	2	1	4	5	3	2.8	3.2	1.8	-	+	3	3	2	4	4	3
Проектирование библиотеки	3	3	2	5	6	4	3.8	4.2	2.8	-	+	4	4	3	6	6	4
Проектирование приложения	2	3	2	4	5	4	2.8	3.8	2.8	-	+	3	4	3	4	6	4
Проектирование приложения нормализации	3	3	1	4	4	2	3.4	3.4	1.4	-	+	3	3	1	4	4	1
Программирование	10	15	20	15	18	25	12	16.2	22	-	+	12	16	22	18	24	33
Проверка результатов на Proben1	3	3	5	5	5	6	3.8	3.8	5.4	-	+	4	4	5	6	6	7
Проверка результатов на кассовых сборах	10	11	6	15	15	10	12	12.6	7.6	-	+	12	13	8	18	19	12
Описание проделанной	7	7	7	14	14	14	9.8	9.8	9.8	+	+	10	10	10	15	15	15



где  $Z_m$  – месячный оклад сотрудника,  $M$  – количество месяцев без отпуска в году,  $F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (Таблица 7).

Таблица 7 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней (выходные дни + праздничные дни)	118	118
Потери рабочего времени (отпуск + невыходы по болезни)	24	24
Действительный годовой фонд рабочего времени	223	223

Расчёт основной заработной платы приведён ниже,  $Z_{осн}$  рассчитана с учётом районного коэффициента (1.3) (Таблица 8).

Таблица 8 – Расчёт основной заработной платы исполнителей

Исполнитель	$Z_m$ , руб	$F_d$ , дн и	$M$ , мес	$Z_{дн}$ , руб	$T_p$ , раб. дни			$Z_{осн}$ , руб		
					Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Дипломник	2000 0	22 3	11. 2	100 4	85	94	89	11094 2	122688. 8	116162. 8
Руководитель	4000 0	22 3	11. 2	200 9	20	20	20	52234	52234	52234

Затраты по дополнительной заработной плате исполнителей учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от

нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.). Дополнительная заработная плата начисляется по формуле, приведённой ниже (18).

$$Z_{\text{доп}} = Z_{\text{осн}} * k_{\text{доп}}, \quad (18)$$

где  $k_{\text{доп}}$  – коэффициент дополнительной з/п. Принимаем его равным 0.15.

После расчёта дополнительной заработной платы получаем следующие цифры (Таблица 9).

Таблица 9 – Дополнительная заработная плата исполнителей

Исполнитель	Исп.1	Исп.2	Исп.3
Дипломник	16641.3	18403.32	17424.42
Руководитель	7835.1	7835.1	7835.1

Теперь необходимо рассчитать отчисления во внебюджетные фонды. Они рассчитываются следующим образом (24).

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}), \quad (24)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.) Для образовательных и научных учреждений он составляет 27.1%.

Результат вычислений приведён ниже (Таблица 10).

Таблица 10 – Отчисления во внебюджетные фонды

Исполнитель	Основная з/п, руб			Дополнительная з.п., руб		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	52234	52234	52234	7835.1	7835.1	7835.1
Дипломник	110942	122688.8	116162.8	16641.3	18403.3 2	17424.4 2
$k_{\text{внеб}}$	0.271					
<b>Итого</b>						
<b>Исполнение 1</b>	50854					

<b>Исполнение 2</b>	54515
<b>Исполнение 3</b>	52481

На основе предыдущих расчётов, можно вывести итоговый бюджет НТИ (Таблица 11).

Таблица 11 – Расчёт бюджета НТИ

Наименование статьи	Сумма, руб		
	Исп.1	Исп.2	Исп.3
Затраты по основной заработной плате исполнителей	163176	174503	168397
Затраты по дополнительной заработной плате исполнителей	24476	26238	25260
Отчисления во внебюджетные фонды	50854	54515	52481
Накладные расходы (16%)	38161	40841	39382
<b>Итого:</b>	<b>276667</b>	<b>296097</b>	<b>285520</b>

### **5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования**

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Сначала рассчитаем финансовую эффективность нашего исполнения (25).

$$I_{\text{финр}}^{\text{исп}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (25)$$

где  $I_{\text{финр}}^{\text{исп}i}$  – интегральный финансовый показатель разработки,  $\Phi_{pi}$  – стоимость  $i$ -того исполнения,  $\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Рассчитаем для всех 3-ёх исполнений:

Для первого результат равен 0.93.

Для второго результат равен 1.

Для третьего результат равен 0.96.

Теперь рассмотрим интегральный показатель ресурсоэффективности, он рассчитывается по следующей формуле (26).

$$I_{pi} = \sum a_i * b_i, \quad (26)$$

где  $a_i$  –весовой коэффициент  $i$ -го варианта исполнения разработки,  $b_i$  – бальная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Для расчётов построим таблицу (Таблица 12).

Таблица 12 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерий	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
Удобство в эксплуатации	0.075	5	4	5
Потребность в ресурсах памяти	0.2	3	4	4
Функциональная мощность	0.15	5	4	1
Простота эксплуатации	0.075	4	5	5



Потребность в мощности ЭВМ	0.2	2	2	3
Скорость работы	0.3	3	2	4
<b>Итого</b>	1	3.3	3.1	3.5

Теперь необходимо сравнить исполнения между собой. Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле (27).

$$I_{испi} = \frac{I_{pi}}{I_{финr}} \quad (27)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта определяется следующим образом (28).

$$\mathcal{E}_{ср} = \frac{I_i}{I_j} \quad (28)$$

Построим сравнительную таблицу (Таблица 13).

Таблица 13 – Сравнительная эффективность разработки

Показатель	Исп.1	Исп.2	Исп.3
Интегральный финансовый показатель разработки	0.93	1	0.96
Интегральный показатель ресурсоэффективности разработки	3.3	3.1	3.5
Интегральный показатель эффективности	3.55	3.1	3.65

Сравнительная эффективность вариантов исполнения	0.97	0.85	1
--	------	------	---

В итоге, был проведён анализ данной работы с точки зрения финансовой и ресурсоэффективности, и рассмотрено 3 различных исполнения. Третье исполнение оказалось самым эффективным, но при этом наименее технологичным, преимущества данное исполнение несмотря на это, теряет свои позиции, и предпочтительным является первое исполнение, так как оно гораздо более гибкое. Второе же исполнение, несмотря на также большую технологичность, дороже и сложнее в реализации, а результат будет не в той же мере лучше. Поэтому было выбрано первое исполнение.

В результате проведения SWOT анализа, были выявлены слабые и сильные стороны проекта. Для успеха на рынке, необходимо воспользоваться имеющейся популярностью технологии, и убедить клиентов, в том, что переход с имеющихся методов на предлагаемы даст им улучшение точности прогнозирования, и более гибкую к изменениям технологию. Но при этом следует опасаться нестабильной ситуации на рынке, т.к. тогда клиенты не будут менее склонны к перестройке своей системы и закупке оборудования для неё.

Так же, как видно из итоговых оценок всех решений, они все являются не оптимальными, т.к. их оценка не превышает даже 4, что означает, что в данной области (прогнозирование временных рядов) ещё есть множество путей повышения эффективности, и для дальнейшего повышения конкурентоспособности данного проекта, необходимо оптимизировать алгоритмы, для снижения потребления ресурсов, а так же повышать удобность использования для конечных пользователей.

## **СПИСОК ПУБЛИКАЦИЙ СТУДЕНТА**

1. Урываев К.А. Прогнозирование временных рядов с помощью искусственных нейронных сетей на примере кассовых сборов фильмов [Электронный ресурс] // Материалы 56-й Международной научной студенческой конференции. г. Новосибирск, 22–27 апреля 2018. – Новосибирск: НГУ, 2018. – ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. – С. 250.