TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.01 «Информатика и вычислительная техника»
Отделение школы – Отделение информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Распознавание образов на изображениях с использованием инструментов машинного обучения

УДК 004.932.2:004.85.032.26

Студент

Группа	ФИО	Подпись	Дата
8ВМ82	Вторушина А.С		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А	к.т.н.,доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Горбенко М.В.	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н.		

Томск – 2020 г.

Планируемые результаты обучения по ООП 09.04.01 Информатика и вычислительная техника

Код	Результаты обучения	**Требования ФГОС 3++ ВО, СУОС ТПУ, критерии ассоциации инженерного образования России и международных стандартов, требования профессиональных стандартов России**
Р1	Самостоятельно приобретать и применять математические, естественнонаучные, социально-экономические и профессиональные знания в области современных информационно-коммуникационных технологий для решения междисциплинарных инженерных задач.	Требования ФГОС 3++ ВО (ОПК-1, ОПК-4), СУОС ТПУ (УК-1, УК-4, УК-5), критерий 5 АИОР (п. 1.1), требования профессионального стандарта 06.014 (ПК-1).
Р2	Разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.	Требования ФГОС 3++ ВО (ОПК-2), СУОС ТПУ (УК-1), критерий 5 АИОР (п. 1.1, п. 1.2), соответствующий международным стандартам EUR-ACE и FEANI, требования профессиональных стандартов 06.015 (ПК-2), 06.016 (ПК-3), 06.041 (ПК-11).
Р3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.	Требования ФГОС 3++ ВО (ОПК-3), СУОС ТПУ (УК-1, УК-6), критерий 5 АИОР (п. 1.2), соответствующий международным стандартам EUR-ACE и FEANI, требования профессионального стандарта 01.004 (ПК-12, ПК-13, ПК-14).
Р4	Демонстрировать способность к практическому использованию полученных новых знаний, новых научных принципов и новых методов исследований.	Требования ФГОС 3++ ВО (ОПК-3, ОПК-4), СУОС ТПУ (УК-4, УК-6), критерий 5 АИОР (п. 1.6, п. 2.2,2.6.), соответствующий международным стандартам EUR-ACE и FEANI, требования профессиональных стандартов 06.027 (ПК-7), 06.036 (ПК-8), 06.037 (ПК-9), 06.040 (ПК-10).

Р5	Разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем, адаптировать зарубежные комплексы обработки информации и информационно-коммуникационные системы к нуждам отечественных предприятий.	Требования ФГОС 3++ ВО (ОПК-5, ОПК-6, ОПК-7), СУОС ТПУ (УК-2, УК-3, УК-6), критерий 5 АИОР (п. 2.1, п. 2.3, п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI, требования профессиональных стандартов 06.015 (ПК-2), 06.026 (ПК-6).
Р6	Осуществлять эффективное управление разработкой программных средств и проектов, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре.	Требования ФГОС 3++ ВО (ОПК-8), СУОС ТПУ (УК-2), требования профессиональных стандартов 06.017 (ПК-4), 06.022 (ПК-5).
Р7	Осуществлять авторское сопровождение процессов проектирования, внедрения, эксплуатации и модернизации программно-аппаратного обеспечения информационно-телекоммуникационных систем на всех этапах жизненного цикла.	Требования СУОС ТПУ (УК-2, УК-3, УК-4), критерий 5 АИОР (п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI. Требования профессиональных стандартов 06.026 (ПК-6), 06.036 (ПК-8), 06.037 (ПК-9).
Р8	Критически анализировать современные проблемы информатики и вычислительной техники, ставить задачи и разрабатывать программу исследования в индустрии новых информационных технологий, выбирать соответствующие методы решения экспериментальных и теоретических задач, критерии эффективности и ограничения их применимости, прогнозировать тенденции научно-технического развития.	Требования ФГОС 3++ ВО (ОПК-3), СУОС ТПУ (УК-1), требования профессионального стандарта 01.036 (ПК-8).
Р9	Способность к профессиональной коммуникации в устной и письменной формах на русском и иностранном языках для решения задач профессиональной деятельности на основе истории и философии нововведений, математических методов и моделей для управления разработкой программных средств и проектов; способность руководить коллективом в сфере профессиональной деятельности, толерантно воспринимая социальные, этнические,конфессиональные и культурные различия; способность публично выступать и отстаивать свою точку зрения.	Требования ФГОС 3++ ВО (ОПК-1), СУОС ТПУ (УК-5, УК-6), требования профессионального стандарта 01.004 (ПК-12, ПК-14).

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки Информатика и вычислительная техника
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ Ботыгин И.А.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

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

Студенту:

Группа	ФИО
8ВМ82	Вторушина А.С.

Тема работы:

Распознавание образов на изображениях с использованием инструментов машинного обучения	
Утверждена приказом директора (дата, номер)	№ 59-42/с от 28.02.2020 г.

Срок сдачи студентом выполненной работы:	09.06.20

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	Необходимо создать программное обеспечение, в основе которого лежит нейронная сеть, способное распознавать рукописные числовые символы. Провести программную оптимизацию продукта, обеспечив достаточное быстродействие и снижение аппаратной нагрузки на ЭВМ.

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	В ходе исследования провести поиск и структурирование литературного материала по теме, проработать основные вопросы в области научных инноваций среди российских и зарубежных исследований. Провести анализ нескольких алгоритмов работы нейронной сети, выбрав при этом оптимальный, исходя из задач исследования. Провести моделирование выбранного алгоритма с последующей оптимизацией процесса.
Перечень графического материала *(с точным указанием обязательных чертежей)*	

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Основная часть	Ботыгин И.А.
Социальная ответственность	Горбенко М.В.
Финансовый менеджмент	Конотопский В.Ю.
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Экспериментальная часть	Аксенова Н.В.
Заключение	Аксенова Н.В.

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	27.01.2020

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н.,доцент		
Доцент	Горбенко М.В.	к.т.н.,доцент		
Доцент	Ботыгин И.А.	к.т.н., доцент		27.01.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ82	Вторушина А.С		27.01.2020

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки 09.04.01 Информатика и вычислительная техника
Уровень образования магистратура
Отделение школы (НОЦ) информационных технологий
Период выполнения осенний / весенний семестр 2019 /2020 учебного года

Форма представления работы:

Магистерская диссертация

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	02.06.2020

Дата контроля	Название раздела / вид работы (исследования)	Максимальный балл раздела
28.02.2020	Модель нейронной сети	
16.03.2020	Выбор программных средств	
20.04.2020	Поведенческое моделирование нейронной сети	
16.05.2020	Оптимизация аппаратных средств	
17.05.2020	Финансовый менеджмент	
27.05.2020	Социальная ответственность	
02.06.2020	Приложение A. Chapter 4. Experimental part	

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н., доцент		

СОГЛАСОВАНО:
Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н., доцент		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ВМ82	Вторушина А.С.

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	Информатика и ВТ

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах; анкетирование; опрос
2. Нормы и нормативы расходования ресурсов	—
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Действующие ставки единого социального налога и НДС, ставка дисконтирования = 0,1 (см. МУ)
1) Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
2) Оценка коммерческого и инновационного потенциала НТИ	Потенциальные потребители результатов исследования; анализ конкурентных технических решений; оценка готовности проекта к коммерциализации.
3) Разработка устава научно-технического проекта	Цели и результат проекта; организационная структура проекта.
4) Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	План проекта; бюджет научного исследования
5) Определение ресурсной, финансовой, экономической эффективности	Оценка экономической эффективности использования результатов ВКР.
Перечень графического материала (с точным указанием обязательных чертежей):	
1. График проведения и бюджет НТИ	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ82	Вторушина А.С.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ВМ82	Вторушина А.С.

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	Информатика и ВТ

Тема ВКР:

Распознавание образов на изображениях с использованием инструментов машинного обучения	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является рабочее место студента. Рабочее место состоит из стола, стула и персонального компьютера. Область применения: машинное обучение
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	Основные проводимые правовые и организационные мероприятия по обеспечению безопасности трудящихся на рабочем месте согласно СанПиН 2.2.2/2.4.1340-03, ФЗ – 197.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Анализ выявленных вредных факторов: – недостаточная освещенность рабочей зоны; – отклонение параметров микроклимата в помещении; – повышенный уровень шума; – повышенный уровень излучения электромагнитных полей. Психофизические факторы: – повышенная нагрузка на органы зрения – длительные статические нагрузки; – монотонность труда; – нервно-эмоциональное напряжение. Анализ выявленных опасных

	производственных факторов рабочей среды, влияющих на организм человека при работе с программным обеспечением в рабочем помещении, а именно: – опасность поражения электрическим током, – опасность поражения статическим электричеством, – короткое замыкание.
3. Экологическая безопасность:	Утилизация используемой орг.техники, макулатуры и люминесцентных ламп.
4. Безопасность в чрезвычайных ситуациях:	Чрезвычайная ситуация техногенного характера для места– пожар. Установка общих правил поведения и рекомендаций во время пожара, план – эвакуации, огнетушитель.

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Горбенко М.В.	к.т.н, доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ82	Вторушина А.С.		

Реферат

Выпускная квалификационная работа 123 с., 31 рис., 27 табл., 44 источников, 2 прил.

Ключевые слова: нейронная сеть, сверточный слой, оптимизация, python, tenzorflow, keras, переобучение, aws, azure, google collaborate.

Объектом исследования является созданная нейронная сеть, по классификации относящаяся к типу сверточных нейросетей.

Цель работы: разработка автоматического метода распознавания рукописных чисел, построенного на основе нейросетевого алгоритма, а также синтез архитектуры нейронной сети и ее оптимизация, с точки зрения ускорения и повышения точности распознавания цифровых рукописных символов.

В процессе исследования проводился выбор оптимальных средств и инструментов синтеза и моделирования нейронной сети, выполнялось обучение и оптимизация разработанного алгоритма.

В результате исследования создано программное обеспечение на основе нейронной сети, по топологии, относящейся к типу сверточных, которое решает задачи распознавания рукописных числовых символов.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: точность 99,2%, мини пакеты 200 шт., соотношение тренировочного и обучающего множеств 0,2, количество эпох – 10.

Созданное программное обеспечение может быть использовано в областях визуального анализа данных бумажной документации какого-либо предприятия, где необходим перевод данных с бумажного носителя в электронный вид.

В будущем планируется использовать результаты исследования для расширения функционала нейронной сети вплоть до возможности распознавания буквенных и других символов.

Определения, обозначения, сокращения, нормативные ссылки

В настоящей работе использованы ссылки на следующие стандарты:

1. СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

2. СанПин 52.13330.2011 «Естественное и искусственное освещение. Актуализированная редакция СанПин 23-05-95*».

3. ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования».

4. ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».

5. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

6. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений.

7. ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.

8. СанПиН 2.2.1/2.1.1.1278-12. «Электромагнитные поля в производственных условиях»

9. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.

10. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.

В данной работе приведены следующие термины с соответствующими определениями:

искусственный нейрон (ИН): Самый простой аналоговый элемент преобразования, который моделирует основные идеи о работе живого нейрона.

нейронная сеть (НС): Представляет собой серию нейронов, которые связаны синапсами.

rectified linear unit (ReLU): Выпрямленная линейная функция активации.

сверточная нейронная сеть (СНС): Специальная архитектура нейронных сетей, предложенная Яном Лекуном.

градиентный спуск: Метод нахождения локального экстремума (минимального или максимального) функции путем перемещения по градиенту.

слой: Модуль обработки данных, принимающий на входе и возвращающий на выходе один или несколько тензоров.

сверточный слой: Набор карт (другое название - карты атрибутов, в повседневной жизни это обычные матрицы), каждая карта имеет синаптическое ядро (сканирующее ядро или фильтр).

дропаут: Способ регуляризации искусственных нейронных сетей. Такой способ предназначен для предотвращения переобучения нейронных сетей.

google colaboratory: Бесплатный облачный сервис, предоставляемый компанией Google

microsoft Azure: Облачная платформа от компании Microsoft.

amazon Web Services (AWS): Коммерческое публичное облако, поддерживаемое и развиваемое компанией Amazon с 2006 года.

предварительно обученная сеть: Сохраненная сеть, прежде обученная на большом наборе данных, обычно в рамках масштабной задачи классификации изображений.

Оглавление

Введение

Данная работа посвящена разработке методов и алгоритмов, входящих в программный комплекс, осуществляющий распознавание изображений и рукописных символов с использованием машинного обучения и построенного по архитектуре искусственных нейронных сетей. В работе поднимаются вопросы сложности обучения сети, ее возможного переобучения, а также предлагаются варианты решения проблемы низкой производительности путем оптимизации архитектуры сети и аппаратных ресурсов с применением облачных сервисов обработки данных. В процессе исследования проведен сравнительный анализ результатов обучения нейронной сети с применением открытой библиотеки машинного обучения TensorFlow, библиотек Keras и NumPy, а также набором данных из базы MNIST.

Актуальность данной темы исследования подтверждается массовым внедрением компьютерных технологий и систем искусственного интеллекта практически во все сферы человеческой деятельности такие как: системы видео и аудио фиксации, поиск и обработка нецифровой информации, контроль качества и другие, где требуется полная автоматизация процесса, повышение качества, скорости выполнения задач. Фундаментальными исследованиями в области нейронных сетей и распознавания образов занимался С. Хайкин. В его трудах приводятся математическое обоснование нейросетевых алгоритмов, примеров и описание компьютерных экспериментов по распознаванию образов, управлению и обработке сигналов. Над задачами по визуальному анализу данных работают научный национальный институт стандартов и технологий (NIST), подразделение «Microsoft Research» и многие другие, которые используют большое количество различных методов и практик для создания новых нейросетевых алгоритмов, построенных в том числе и на сверточных нейнонных сетях, так как они лучше подходят для задач визуального анализа данных. Результаты таких исследований широко внедряются в современные

технологии оптико-электронных приборов и комплексов, ориентированных на формирование и обработку цифровых изображений.

Одна из центральных проблем, которая должна быть разрешена с помощью настоящего исследования — это проблема определения методов и алгоритмов обработки информации на основе которых могли бы производиться создание и направленная оптимизация инструментов решения поставленной задачи. Таким образом, исходя из актуальности темы, задача настоящего исследования направлена на разработку программного обеспечения для распознавания рукописных цифр с использованием методов машинного обучения.

Целью исследования является разработка автоматического метода распознавания рукописных чисел, построенного на основе нейросетевого алгоритма, а также синтез архитектуры нейронной сети и ее оптимизация, с точки зрения ускорения и повышения точности распознавания цифровых рукописных символов.

Предметом исследования в данной работе выступает оптимизация, ускорение и повышение точности машинных систем распознавания образов.

Объектом исследования является созданная нейронная сеть, по классификации относящаяся к типу сверточных нейросетей.

Научная новизна исследования заключается в создании оригинальной архитектуры нейронной сети, построенной по принципу многослойности, и относящаяся к типу сверточных, являющаяся вариацией многослойного персептрона, где каждый слой содержит определенное количество рецептивных полей.

Практическая значимости результатов исследования подтверждается множеством публикаций по данной теме, а также повсеместным использованием в повседневной жизни. Результаты работы могут быть использованы в областях визуального анализа данных бумажной документации какого-либо предприятия, где необходим перевод данных с бумажного носителя в электронный вид.

Апробация работы проведена путем публикации научной работы «Распознавание образов с использованием инструментов машинного обучения» в журнале «Молодежь и современные информационные технологии».

1 Объект и методы исследования

Согласно техническому заданию на выполнение диссертационной работы суть исследования заключается в разработке методов и алгоритмов, входящих в программный комплекс, осуществляющий распознавание изображений и рукописных символов с использованием машинного обучения и построенного по архитектуре искусственных нейронных сетей.

В ходе прохождения производственной практики определены объекты и предметы исследования, которыми здесь выступают вновь созданная нейронная сеть и ее оптимизация соответственно. В рамках исследования был разработан алгоритм нейронной сети, ее архитектура, на основе которых созданы программное обеспечение для выполнения задач, поставленных в техническом задании. По результатам моделирования нейронной сети произведена оптимизация алгоритма, направленная на ускорение и оптимальную обучаемость сети.

В качестве инструментов использовались интегрированная среда программирования Visual Studio, язык программирования Python, а также дополнительные пакеты Keras, Фреймворк «TensorFlow», Numpy и различные общедоступные облачные сервисы по оптимизации процесса.

2 Модель нейронной сети

2.1 Обзор литературы

В статье Маринчука А. С. И Баженова Р. И. «Распознавание цифр на основе нейронных сетей в Oktave» [1] описан способ создания инструмента для распознавания числовых символов при помощи нейронной сети в программной среде Oktave. Основной целью работы в статье является разработка оптимального алгоритма нейронной сети на языке высокого уровня совместного с MATLAB. Среди задач исследования можно выделить повышение точности распознавания числовых символов, поступающих на вход нейросети, оптимизация количества итераций внутри алгоритма и достижение определенного времени обучения. Автору удалось достигнуть точности предсказания в распознавании рукописных цифр 75%, ч в свою очередь по словам авторов статьи приемлемо для использования такого алгоритма для распознавания рукописных документов.

В статье «Алгоритмы распознавания символов» авторов А. А. Вальке и Д. Г. Лобова [2] рассматриваются различные алгоритмы распознавания символов, проводится их сравнение и анализ. На основе метода шаблонов предложен алгоритм по распознаванию символов и представлены результаты его тестирования. Основной целью работы является создание программы по распознаванию рукописных символов, построенного на основе уже существующих алгоритмов. Авторами статьи приведены теоретические основы построения алгоритмов распознавания символов, а также описаны и проанализированы наиболее часто встречающиеся методы распознавания символов, к которым относятся шаблонный, признаковый, структурный и основанный на искусственных нейроных сетях.

В книге «Глубокое обучение на Python» Ф. Шолле [3] рассматривается реальное практическое исследование глубокого обучения с объяснением количественных понятий с помощью фрагментов кода, сформированными

практическими идеями машинного и глубокого обучения. В примерах данной книги используется фреймворк глубокого обучения Keras, написанный на языке программирования Python и библиотека TensorFlow в качестве внутреннего механизма. Кроме того, здесь описывается теория для освоения машинного обучения: градиентный спуск, обратное распространение ошибки обучения. Такая теория применима для распознавания образов с особым вниманием к классификации изображений.

В работе Х. С. Исрафилова «Применение нейронных сетей в распознавании рукописного текста», посвященной проблемам распознавания почерка, проведен анализ методологии решения задачи распознавания с использованием нейронных сетей, рассмотрены основные свойства почерка, а также выявлены основные достоинства и недостатки нейросетей с «учителем» и сетей «без учителя» [4].

Над проблемой распознавания образов работали авторы статьи «Распознавание рукописных символов с применением нейросетевой технологии» В. В. Андреев и М. С. Журавлев. В работе описано созданное программное обеспечение, способное сканировать символы определенной группы из графического файла. Предполагаемый метод распознавания символов из файла основан на выделении точек, несущих наибольшую информацию с последующей обработки функций MATLAB и нейронной сетью.

В статье «Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis» описаны работы некоторых исзвестных алгоритмов, проведен их анализ, выявлены их преимущества и недостатки, что является основой для создания нового алгоритма распознавания рукописных цифр, на основе сверточной нейронной сети с целью улучшения оптимизации и точности распознавания [5].

В статье «Understanding Batch Normalization for Neural Network» приводится алгоритмы, и архитектура созданной нейронной сети для распознавания цифр с помощью базы MNIST. В нейронной сети рассматривается работа со сверточными и полносвязными слоями , а также

зависимость сетов эпох от общего объема данных в эпохе в задаче распознавания рукописных чисел [6].

2.2 Математическая концепция нейронной сети

Искусственный нейрон (ИН) - это самый простой аналоговый элемент преобразования, который моделирует основные идеи о работе живого нейрона. Некоторые сигналы принимаются на входе ИН. Каждый вход взвешивается - умножается на определенный коэффициент (синаптическая мощность). Сумма всех продуктов определяет степень активации нейрона.

Активационная функция F должна быть монотонной. Обычно $F(y)$ принадлежит интервалу [0,1] или [–1,1]. Чаще используют множество вариантов активационных функций. Таким образом, ИН выполняет две операции. Сначала вычисляется сумма скалярного произведения вектора весов W и входного вектора X:

$$y = X^t W + b \tag{1}$$

затем срабатывает активационная функция, определяющая значение выходного сигнала:

$$z = f\left(y\right) \tag{2}$$

Таким образом, ИН реализует следующую структуру:

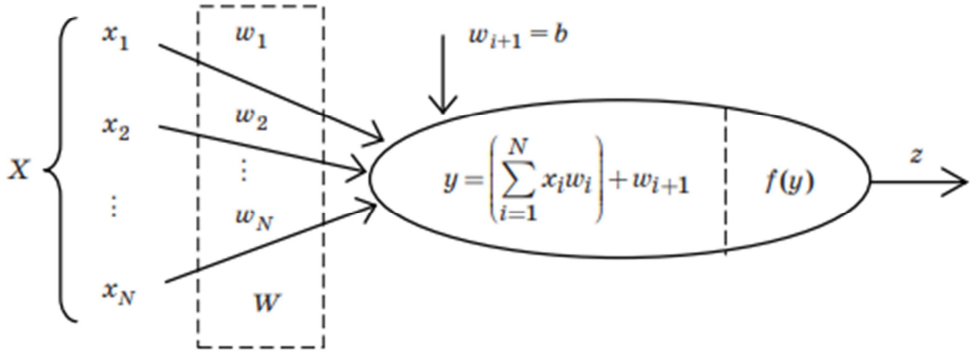


Рисунок 2.1 – Структура искусственного нейрона

Нейронная сеть (НС) представляет собой серию нейронов, которые связаны синапсами. Структура нейронной сети пришла непосредственно из биологии в мир программирования. Благодаря этой структуре, машина может анализировать и даже хранить различную информацию. Нейронные сети могут не только анализировать поступающую информацию, но и воспроизводить ее из своей памяти.

Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений. Наиболее распространенные виды использования нейронных сетей:

- Классификация - распределение данных по параметрам;
- Предсказание - способность предсказать следующий шаг;
- Распознавание. В настоящее время является самым широким применением нейронных сетей [7].

2.3 Свойства нейронных сетей

2.3.1 Выпрямленная линейная функция активации (rectified linear unit, ReLU)

Известно, что нейронные сети могут аппроксимировать произвольно сложную функцию, если они содержат достаточно слоев и функция активации является нелинейной. Функции активации, такие как сигмоидальная или тангенциальная, не являются линейными, но приводят к проблемам с демпфированием или увеличением градиентов в процессе обучения. Однако можно использовать гораздо более простой вариант - прямую линейную функцию активации.

Функция ReLU является линейной функцией и в настоящее время считается гораздо более простым и эффективным вариантом передаточной функции с точки зрения сложности вычислений. Это один из последних успехов в области методов оптимизации глубоких нейронных сетей.

Данная функция активации описывается следующей формулой:

$$f(s) = \max(0, s)$$

$$f(s) = \begin{cases} 1, s > 0 \\ rand(0.01, 0.05), s \le 0 \end{cases} \quad (3)$$

и графически выглядит следующим образом:

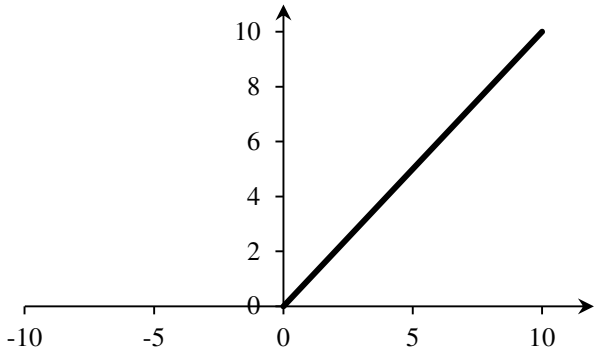


Рисунок 2.2 - График функции активации

Ее производная равна единице или нулю, и поэтому градиентный рост или ослабление не могут произойти. Кроме того, использование этой функции приведет к уменьшению масштаба. Сегодня существует семейство различных модификаций ReLU, которые решают проблемы надежности этой передаточной функции, когда большие градиенты проходят через нейрон: Leaky ReLU, Parametric ReLU, Randomized ReLU [8]. Достоинства такой функции активации, следующие:

- Лишена ресурсоёмких операций;
- Отсекает не нужные детали;
- Отсутствует разрастание/затухание градиента;
- Быстрое обучение.

2.3.2Функция активации Softmax

Эта функция активации служит для преобразования каждого вектора с действительными значениями в вектор вероятности и определяется для *i*-го нейрона. Для полносвязного слоя будет другая функция активации softmax.

Softmax будет использоваться в качестве функции активации при решении задачи классификации [9]. Данная функция активации задается следующей формулой:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \quad (4)$$

где z_i – значение на выходе из i-го нейрона до активации,

N – общее количество нейронов в слое.

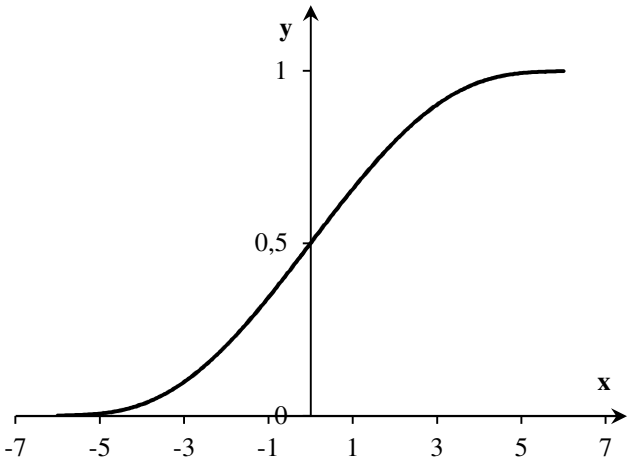


Рисунок 2.3 – Функция активации Softmax

2.3.3Метод градиентного спуска

Градиентный спуск - это метод нахождения локального экстремума (минимального или максимального) функции путем перемещения по градиенту. Градиент представляет собой вектор, который определяет крутизну склона функции и указывает его направление относительно точки на поверхности или графике. Чтобы найти градиент нужно взять производную от функции в данной точке – *f'(w1)* на рисунке ниже.

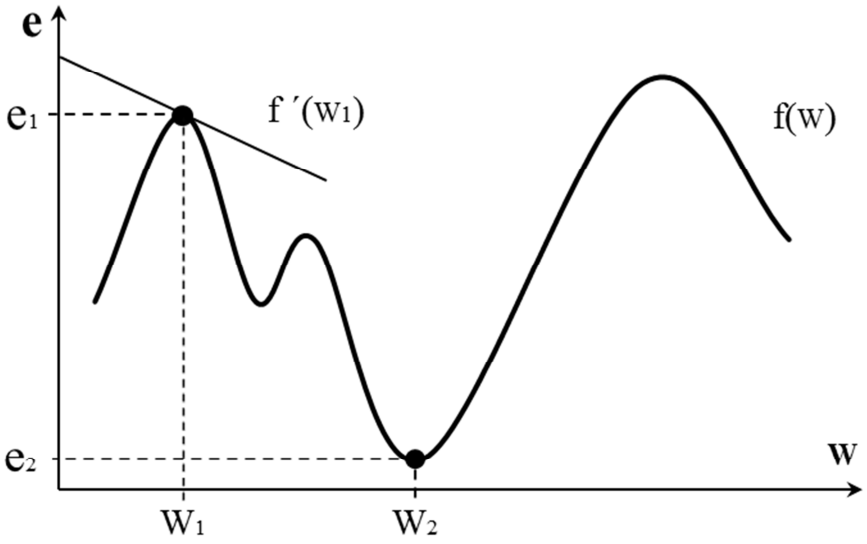


Рисунок 2.4 – Производная функции

Метод градиентного спуска в чистом виде может «застревать» в локальных минимумах функции. Для преодоления проблемы метода градиентного спуска используется метод моментов. Идея метода в сохранении части скорости спуска в каждой итерации. Фактически, происходит добавление к приращению веса приращения веса из предыдущей итерации, умноженной на коэффициент момента

$$\Delta W_t = \eta \cdot \nabla E + \mu \cdot \Delta W_t - 1 \qquad (5)$$

где η – коэффициент скорости обучения,

∇E – градиент функции потери,

μ – коэффициент момента,

ΔW_t - 1 – изменение весов на предыдущей итерации.

В дополнение к моменту метод обратного распространения также использует такой параметр, как скорость обучения. При этом можно контролировать размер корректирующих весов на каждой итерации [10].

2.3.4Алгоритм обратного распространения ошибки

Основная идея обратного распространения - получить оценку ошибки для нейронов скрытого слоя. Следует обратить внимание, что известные

ошибки, сделанные нейронами выходного слоя, происходят из-за неизвестных ошибок нейронов скрытого слоя. Чем больше значение синаптической связи между нейроном скрытого слоя и выходным нейроном, тем больше ошибка первого влияет на ошибку второго. Следовательно, оценка погрешности элементов скрытых слоев может быть получена как взвешенная сумма ошибок последующих слоев.

Смысл алгоритма заключается в том, что при обучении сети сначала отображается изображение, для которого вычисляется выходная ошибка. Кроме того, эта ошибка распространяется в противоположном направлении по сети и изменяет веса межнейронных соединений. Алгоритм содержит ту же последовательность действий, что и при обучении персептрона. Сначала весам межнейронных связей задаются случайные значения, затем выполняются следующие шаги:

1) выбирается обучающая пара (X, Z*), X подается на вход;
2) вычисляется выход сети Z = F(Y);
3) рассчитывается ошибка выхода E;
4) веса сети корректируются с целью минимизации ошибки;
5) возврат к первому шагу и т. д., пока не будет минимизирована ошибка по всем обучающим парам. Шаги 1 и 2 – это прямое распространение по сети, а шаги 3 и 4 – обратное.

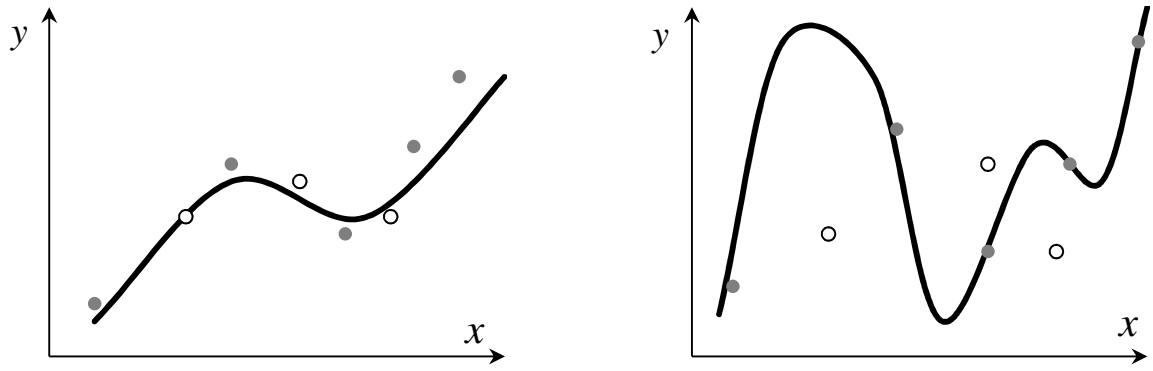
Перед обучением существующие пары ввода-вывода должны быть разделены на две части: обучение и тестирование. Тестовые пары используются для проверки качества обучения: НС хорошо обучена, если выдает выходные данные для конкретной тестовой пары, близкие к выходным данным теста. Во время обучения возможна ситуация, когда НС показывает хорошие результаты для данных обучения и плохие результаты для данных теста.

Это может быть обусловлено двумя причинами:

− Тестовые данные сильно отличаются от обучающих, т. е. обучающие пары охватывали не все области входного пространства.

– Возникло явление «переобучения» (overfitting), при котором поведение НС оказывается более сложным, чем решаемая задача.

Примером хорошо обученной и переобученной НС могут служить функции аппроксимации тестовых и обучающих данных, которые представлены на рисунке (Рисунок 2.5) светлыми и темными кружками соответственно.



а) б)

Рисунок 2.5 - а) – нормально обученная НС; б) – переобученная НС

Во избежание переобучения НС следует добавить второй слой к существующей однослойной НС. При этом такой слой можно считать скрытым, а следовательно формула для коррекции весов данного скрытого слоя задается следующей формулой:

$$W_{ij}(t+1) = W_{ij}(t) - \eta\left(\sum_{k=1}^{p} \Delta_k z_k (1 - z_k) v_{ik}\right) u_j (1 - u_j) x_i \qquad (6)$$

где u_j – выход j-го нейрона скрытого слоя

вес v_{jk}, связывающий j-й нейрон скрытого слоя и k-й нейрон выходных весов скрытого слоя ошибки для скрытого слоя Δ_j.

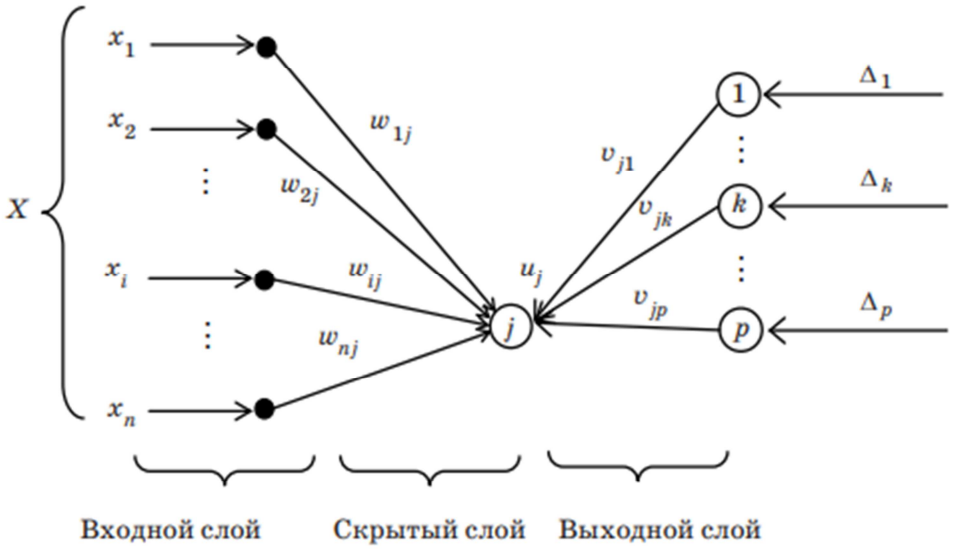


Рисунок 2.6 – Обратное распространение ошибки

Алгоритм перераспределения ошибок может быть реализован тремя способами:

– Последовательный режим. Режим последовательной тренировки иногда называют стохастическим градиентным спуском.

– Пакетный режим. В режиме пакетного обучения веса соединений корректируются после входа в сеть всех обучающих примеров эпохи обучения.

– Мини партия. Существует компромисс между двумя типами распространения ошибок (градиентный спуск), иногда называемый «мини-пакет».

В этом случае синоптические веса сети корректируются после небольшого количества шаблонов обучения. С точки зрения производительности, режим последовательного обучения предпочтительнее пакетного режима, поскольку для хранения каждой синоптической ссылки требуется меньше внутренней памяти. Кроме того, представление обучающих примеров в случайном порядке в процессе обучения для последовательного режима делает поиск в пространстве весов стохастическим. Это уменьшает вероятность того, что алгоритм останавливается в точке локального минимума [7, 11].

2.4 Описание сверточной нейронной сети

Сверточная нейронная сеть (CHC) — это специальная архитектура нейронных сетей, предложенная Яном Лекуном. Сверточные нейронные сети отличаются от множества других видов нейронных сетей. Первоначально такая сеть нацелена на эффективное распознавание и классификация изображений

В сверточной нейронной сети происходит чередование двух типов слоев: сверточных слоев и слоев подвыборки. После каждого этапа свёртки следует этап подвыборки, формируются несколько карт признаков, каждая карта признаков проходит этап подвыборки для уменьшения размерности и фильтрации незначимых деталей. После нескольких циклов свёртки и подвыборки, конечные карты признаков разворачиваются в вектор признаков.

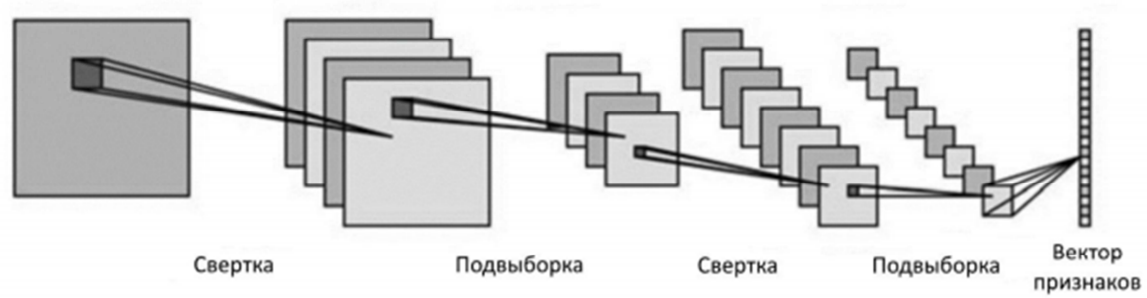


Рисунок 2.7 – Сверточные слои

Такое название сеть получила из-за того, что присутствуют операции свёртки. Суть операций свертки заключается в вычислении совершенно нового значения текущего пикселя, полностью учитывая значения соседних пикселей. Чтобы вычислить такие значения используется ядро свертки.

Сверточные нейронные сети могут обеспечивать устойчивость к следующим изменениям:

- масштаба;
- смещения;
- поворота;
- смене ракурса;
- прочих искажений.

На сегодняшний день сверточная нейронная сеть и прочие модификации сети считаются лучшими алгоритмами по точности и скорости нахождения объектов на изображениях.

После конкурса ImageNet сверточные нейронные сети получили особое внимание и популярность. Данный конкурс был посвящён распознаванию объектов на фотографиях [12].

СНС состоит из следующих слоёв:

- входной слой;
- сверточные слои (convolutional);
- Pooling-слои (Subsampling);
- полносвязный слой;
- выходной слой.

2.4.1 Сверточный слой

Слой – это модуль обработки данных, принимающий на входе и возвращающий на выходе один или несколько тензоров. Сверточный слой - это набор карт (другое название - карты атрибутов, в повседневной жизни это обычные матрицы), каждая карта имеет синаптическое ядро (сканирующее ядро или фильтр). Ядро свертки представляет из себя фильтр или окно.

Свертки определяются двумя ключевыми параметрами:

- Размер шаблонов, извлекаемых из входных данных. Обычно 3x3 или 5x5;
- Глубина выходной карты признаков – количество фильтров, вычисляемых сверткой.

Свертка работает методом скользящего окна: она двигает окно 3x3 или 5x5 по трехмерной входной карте признаков, останавливается в каждой возможной позиции и извлекает трехмерный шаблон окружающих признаков (с формой высоты окна, ширины окна и глубины входа). Каждый такой шаблон затем преобразуется путем умножения тензора на матрицу весов, получаемую в

ходе обучения, которая называется ядром свертки. Все эти векторы собираются в трехмерную выходную карту с формой высоты окна, ширины окна и глубины входа. Каждое пространственное местоположение в выходной карте признаков соответствуют тому же местоположению во входной карте признаков.

Например, сеть была обучена на нескольких лицах, одно из ядер в процессе обучения может давать один из самых больших сигналов в области глаза, рта, брови или носа. В тоже время другое ядро может видеть и другие признаки. Обычно размер ядра берут в пределах от 3x3 до 7x7. Если ядро маленькое, оно не может обнаружить никаких признаков. Если оно слишком велико, количество связей между нейронами увеличивается. Кроме того, размер ядра выбирается таким образом, чтобы размер изображений слоя свертки был равномерным, чтобы не терялась информация при уменьшении размерности в слое недостаточной дискретизации, описанном ниже.

Такое ядро представляет собой систему общих весов или синапсов. Это является главной особенностью такой нейронной сверточной сети. В обычной многоуровневой сети существует множество связей между нейронами, то есть синапсами, что сильно замедляет процесс распознавания. С другой стороны, в сверточной сети общий вес уменьшает количество соединений и позволяет нам находить одну и ту же функцию по всей области изображения.

Первоначально значения каждой карты слоя свертки равны 0. Значения весов ядер устанавливаются случайным образом в диапазоне от -0,5 до 0,5. Ядро скользит по предыдущей карте и выполняет операцию свертки, которая обычно используется для обработки изображений [3]. Формула, по которой вычисляется карта слоя свертки, представлена ниже:

$$f \cdot g[m,n] = \sum_{k,i} f[m-k, n-l] \cdot g[k,l] \qquad (7)$$

Принцип действия свертки представлен на рисунке ниже:

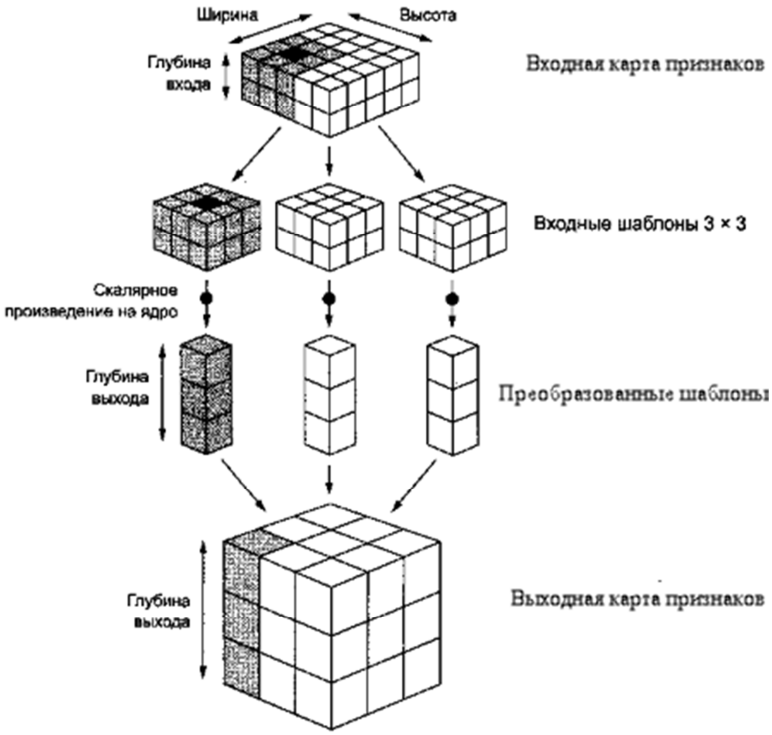


Рисунок 2.8 – Принцип действия свертки

2.4.2Шаг свертки

Другой фактор, который может влиять на размер выходной карты признаков – шаг свертки. До сих пор в объяснениях выше предполагалось, что центральная клетка окна свертки последовательно перемещается в смежные клетки входной карты. Однако в общем случае расстояние между двумя соседними окнами является настраиваемым параметром, который называется шагом свертки и по умолчанию равен единице. Также имеется возможность определять свертки с пробелами – свертки с шагом больше единицы. На рисунке ниже можно видеть, как извлекаются шаблоны с 3x3 сверткой с шагом 2 из входной карты 5x5 (без дополнения).

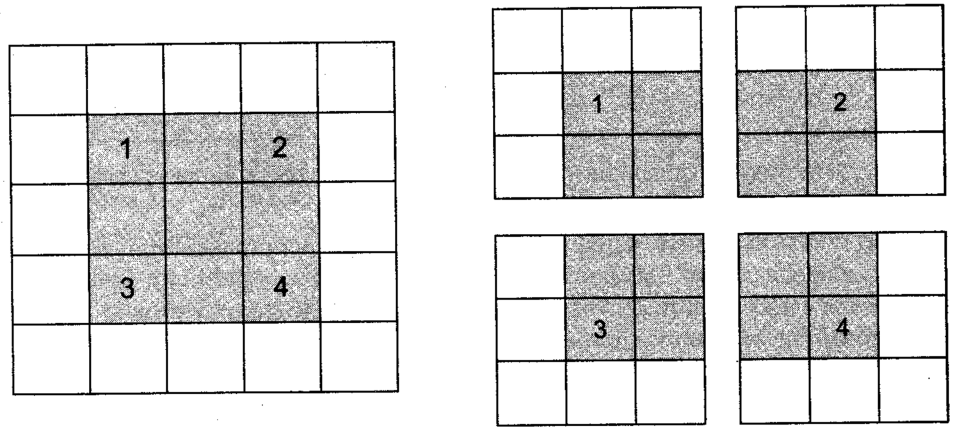


Рисунок 2.9 - Шаблоны 3x3 с шагом свертки 2x2

Использование шага 2 означает уменьшение ширины и высоты карты признаков за счет уменьшения разрешения в два раза (в дополнение к любым изменениям, вызванным эффектами границ). Для уменьшения разрешения карты признаков вместо шага часто используется операция выбора максимального значения из соседних [3].

2.4.3Выбор максимального значения из соседних

Операция выбора максимального значения из соседних заключается в следующем: из входной карты признаков извлекается окно, и из него выбирается максимальное значение для каждого канала. Концептуально, это напоминает свертку, но вместо преобразования локальных шаблонов с обучением на линейных преобразованиях (ядро свертки) они преобразуются с использованием жестко заданной тензорной операции выбора максимального значения. Главное отличие от свертки состоит в том, что выбор максимального значения из соседних обычно производится с окном 2x2 и шагом 2, чтобы уменьшить разрешение карты признаков в два раза.

Уменьшение разрешения используется для уменьшения коэффициентов в карте признаков для обработки, а также для внедрения иерархии пространственных фильтров путём создания последовательных слоев свертки для просмотра все более крупных окон [12].

2.4.4Подвыборочный слой

Подвыборочный слой также, как и сверточный имеет карты, но их количество соответствует предыдущему (сверточному) уровню. Назначение слоя - уменьшить размер карт предыдущего слоя. Если некоторые символы уже были идентифицированы в предыдущей операции свертки, такое детальное изображение больше не требуется для дальнейшей обработки и сжимается до меньшего количества деталей. Кроме того, фильтрация ненужных частей помогает предотвратить переподготовку. Во время сканирования через ядро подсканирующего слоя (фильтра) карты предыдущего уровня ядро сканирования не пересекается, в отличие от слоя свертки. Обычно каждая карта имеет ядро 2x2, что может уменьшить количество предыдущих карт в таблице сгибов в два раза. Все отображение атрибутов разделено на ячейки 2x2, из которых выбираются максимальные значения

Рисунок 2.10 – Подвыборочный слой

Подвыборочный слой идет за свёрточным слоем. Слой состоит из плоскостей и, обычно, имеет такое же количество плоскостей, что и в предыдущем сверточном слое [13].

2.4.5Визуализация промежуточных активаций

Визуализация промежуточных активаций заключается в отображении карт признаков, которые выводятся разными сверточными и объединяющими слоями в ответ на определенные входные данные (вывод слоя, результат функции активации, часто называют его активацией). Этот прием позволяет увидеть, как входные данные разлагаются на различные фильтры, полученные сетью в процессе обучения. Обычно для визуализации используются карты признаков с тремя измерениями: шириной, высотой и глубиной (каналы цвета). Каналы кодируют относительно независимые признаки, поэтому для визуализации этих карт признаков предпочтительнее строить двумерные изображения для каждого канала в отдельности [3].

2.4.6Полносвязанный слой

В полносвязанном слое каждый нейрон соединён со всеми нейронами на предыдущем уровне. Каждая такая связь имеет свой весовой коэффициент.

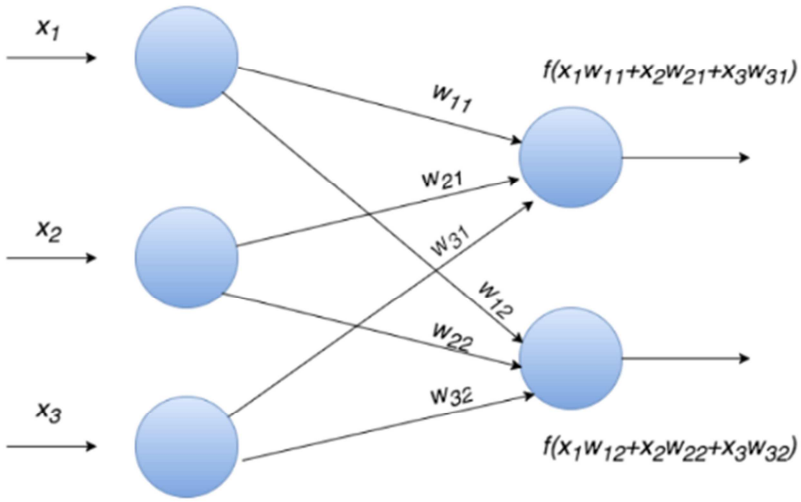


Рисунок 2.11 – Полносвязный слой

где x_i – входные данные

W_{ij} – весовые коэффценты

$f()$ – функция активаци

2.4.7DropOut слой

Дропаут (от англ. dropout) - способ регуляризации искусственных нейронных сетей. Такой способ предназначен для предотвращения переобучения нейронных сетей. Суть метода заключается в том, что при обучении выбирается слой, из которого произвольно выбрасывается определенное количество нейронов (например, 30%), которые отключаются от дальнейших расчетов. Эта техника повышает эффективность обучения и качество результата. Более обученные нейроны приобретают больший вес в сети.

- Дропаут существует в двух версиях: прямая (редко используется) и обратная.
- Dropout на отдельном нейроне можно представить, как случайную величину с распределением Бернулли.
- Dropout может быть представлен как случайная величина с биномиальным распределением на множестве нейронов.
- Невзирая на факт вероятности того, что np нейроны будут отсоединены от сети, np - это среднее число нейронов, отсоединенных в слое n нейронов.
- Обратный Dropout может использоваться для увеличения скорости обучения.
- Обратный Dropout необходимо использовать в сочетании с другими методами нормализации, которые ограничивают значения параметров. Это нужно чтобы упростить процесс выбора скорости обучения.

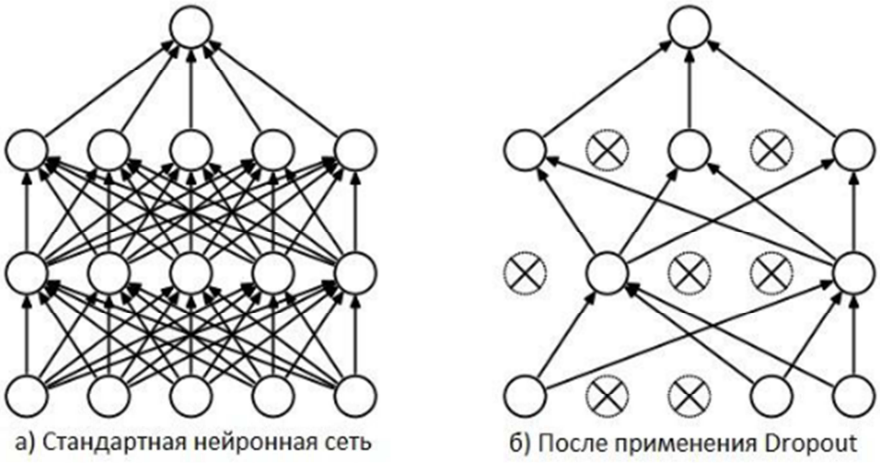


Рисунок 2.12 – Топология НС

К преимуществам НС можно отнести:

- сверточные сети – лучший тип моделей машинного обучения для задач распознавания образов. Вполне можно обучить такую сеть с нуля на очень небольшом наборе данных и получить хороший результат;
- существующую сверточную нейронную сеть с легкостью можно повторно использовать на новом наборе данных, применив выделения признаков. Этот прием особенно ценен при работе с небольшими наборами изображений;
- возможность использовать прием дообучения, который адаптирует к новой задаче некоторые представления, ранее полученные существующей моделью. Он еще больше повышает качество модели;
- возможность воспользоваться методом распараллеливания вычислений;

Основным недостатком НС является следующее:

- Когда объем данных ограничен, главной проблемой становится переобучение;
- Множество переменных, от которых зависят параметры сверточной нейронной сети, например:

1) количество слоев;
2) размер сверточного ядра каждого слоя;
3) количество ядер каждого слоя;

4) шаг сдвига ядра при обработке слоя;

5) необходимость в уровнях подвыборки, степень уменьшения их размерности;

6) функция уменьшения размерности, передаточная функция нейронов;

7) наличие и параметры выхода полностью связанной нейронной сети на выходе свертки.

Эти параметры оказывают большое влияние на результат и подбираются опытным путем. Для популярных задач существуют определенные конфигурации сети, но для новых задач выбор параметров выполняется опытным путем [13].

3 Выбор программных средств

Для реализации алгоритма сверточной нейронной сети требуется:

- Выбор среды программирования;
- Выбор языка программирования;
- Выбор дополнительных инструментов разработки;
- Выбор облачных сервисов для оптимизации процесса.

3.1 Выбор языка программирования

Для разработки программного обеспечения можно использовать любые существующие языки программирования. Один из популярных языков программирования на сегодняшний день – Python. В процессе изучения разработки нейронных сетей на языке Python был выбраны дополнительные пакеты как:

- Keras
- Фреймворк «TensorFlow»
- Numpy

3.1.1 Python

Python — мощный и простой для изучения язык программирования. Он позволяет использовать эффективные высокоуровневые структуры данных и предлагает простой, но эффективный подход к объектно-ориентированному программированию. Сочетание изящного синтаксиса, динамической типизации в интерпретируемом языке делает Python идеальным языком для написания сценариев и ускоренной разработки приложений в различных сферах и на большинстве платформ.

Интерпретатор Python и разрастающаяся стандартная библиотека находятся в свободном доступе в виде исходников и двоичных файлов для всех

основных платформ на официальном сайте Python http://www.python.org и могут распространяться без ограничений. Кроме этого на сайте содержатся дистрибутивы и ссылки на многочисленные модули сторонних разработчиков для языка Python, различные программы и инструменты, а также дополнительная документация.

Интерпретатор Python может быть легко расширен с помощью новых функций и типов данных, написанных на C/C++ (или других языках, к которым можно получить доступ из C). Также Python можно применять как язык расширений для настраиваемых приложений [14].

3.1.2Описание используемых библиотек и фреймворков

В качестве основного вычислительного ядра программы выступает открытый программный фреймворк для машинного обучения «TensorFlow». Фреймворк «TensorFlow» [15] разработан компанией Google для решения задач, связанных с построением и тренировкой нейронной сети. Целью разработки будет автоматическое нахождение и классификация образов, для достижения качества человеческого восприятия. Фреймворк применяется как для исследований, так и для собственных задач связанных с разработкой продуктов Google. Основной API для работы с библиотекой реализован для Python, также существуют реализации для следующих языков программирования: C++, Haskell, Java, Go и Swift.

Для удобной работы с TensorFlow воспользуемся открытой нейросетевой библиотекой «Keras», написанная на языке Python. Библиотека представляет собой надстройку над фреймворком TensorFlow[16]. Keras нацелен на эффективную работу с нейросетями машинного обучения. Наряду с этим keras спроектирован для того, чтобы быть компактным, но в тоже время, модульным и расширяемым. Эта библиотека содержит многочисленные реализации широко применяемых блоков для работы с нейронными сетями.

Например, слои, целевые и передаточные функции, оптимизаторы, и другие инструменты для упрощения работы с изображениями и текстом

Преимущества библиотеки Keras:

- Простота в использовании;
- Модульность;
- Легко расширяемая модель.

В качестве основного языка разработки был выбран язык Python. По этой причине в нашей работе не обойтись без использования фундаментального пакета для научных вычислений «NumPy», адаптированного специально для этого языка. Данный пакет содержит в себе [17]:

- мощный N-мерный массив объектов
- сложные (вещательные) функции
- инструменты для интеграции C / C ++ и кода Fortran
- полезная линейная алгебра, преобразование Фурье и возможности случайных чисел.

3.2 Выбор среды программирования

Одно из основных этапов разработки программного продукта является выбор интегрированной среды программирования [18]. Для выбора среды программирования необходимо учитывать следующие требования:

Запуск кода из среды программирования

- Поддержка отладки;
- Подсветка синтаксиса для быстрого и удобного понимания кода;
- Автоматическое форматирование кода для сокращения времени разработки;
- Сохранение ранее открытых файлов в IDE.

3.2.1Pycharm

Одной из лучших полнофункциональных IDE, предназначенных именно для Python, является PyCharm. Существует как бесплатный open-source (Community), так и платный (Professional) варианты IDE. PyCharm доступен на Windows, Mac OS X и Linux.

PyCharm поддерживает разработку на Python напрямую — открыв новый файл есть возможность сразу писать код. Реализована возможность запускать и отлаживать код прямо из PyCharm. Кроме того, в IDE есть поддержка проектов и системы управления версиями.

Особенности:

- Мощный и функциональный редактор кода с подсветкой синтаксиса, авто-форматированием и авто-отступами для поддерживаемых языков.
- Простая и мощная навигация в коде.
- Помощь при написании кода, включающая в себя автодополнение, авто-импорт, шаблоны кода, проверка на совместимость версии интерпретатора языка, и многое другое.
- Быстрый просмотр документации для любого элемента прямо в окне редактора, просмотр внешеней документации через браузер, поддержка docstring – генерация, подсветка, автодополнение и многое другое.
- Большое количество инспекций кода.
- Мощный рефакторинг кода, который предоставляет широкие возможности по выполнению быстрых глобальных изменений в проекте.
- Полная поддержка свежих версий Django фреймворка.
- Поддержка Google App Engine.
- Поддержка IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK и др.
- Поддержка Flask фреймворка и языков Mako и Jinja2.
- Редактор Javascript, Coffescript, HTML/CSS, SASS, LESS, HAML.

– Интеграция с системами контроля версий (VCS).

– UML диаграммы классов, диаграммы моделей Django и Google App Engine.

– Интегрированное Unit тестирование.

– Интерактивные консоли для Python, Django, SSH, отладчика и баз данных.

– Полнофункциональный графический отладчик (Debugger).

– Поддержка схем наиболее популярных IDE/редакторов. таких как Netbeans, Eclipse, Emacs, эмуляция VIM редактора.

– Поддерживаемые языки: Python (Versions: 2.x, 3.x), Jython, Cython, IronPython, PyPy, Javascript, CoffeScript, HTML/CSS, Django/Jinja2 templates, Gql, LESS/SASS/SCSS/HAML, Mako, Puppet, RegExp, Rest, SQL, XML, YAML.

– PyCharm имеет несколько цветовых схем, а также настраиваемую подсветку синтаксиса кода.

– Огромная, постоянно пополняемая коллекция плагинов.

– Кросс-платформенность (Windows, Mac OS X, Linux).

Преимущества: это среда разработки для Python с мощной поддержкой и хорошим коммьюнити.

Недостатки: PyCharm может медленно загружаться, а настройки по умолчанию, возможно, придётся подкорректировать для существующих проектов [19].

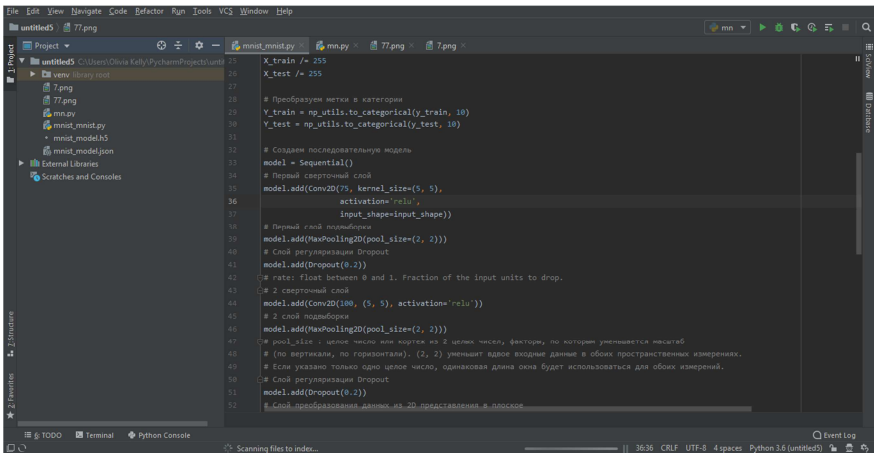


Рисунок 3.1 - PyCharm

3.3 Выбор облачных сервисов для оптимизации процесса

3.3.1 Google Colaboratory

Google Colaboratory [20] – это бесплатный облачный сервис, предоставляемый компанией Google. Данный сервис направлен для упрощения исследований науки о данных и машинного обучения. Google Colaboratory, безусловно, один из самых популярных облачных сервисов. Его главный плюс в том, что он бесплатный и предоставляет необходимый набор пакетов и библиотек [21]. В Google Colaboratory не надо устанавливать пакеты и библиотеки вручную, достаточно просто их импортировать. В то время, как в обычной IDE нужно вручную устанавливать все библиотеки для работы [22]. Google Colaboratory предоставляет бесплатно 12.72 ГБ RAM и 48.98ГБ дискового пространства.

3.3.2 Microsoft Aruze

Microsoft Azure – это облачная платформа от компании Microsoft. Данная платформа представляет возможность разработки, выполнения приложений и хранения данных на серверах, расположенных в распределённых дата-центрах. Microsoft Azure реализует облачные модели платформы как сервис и инфраструктуры как сервис.

Azure предоставляет такие службы для разработчиков Python, как решения для размещения приложений, хранилище, базы данных с открытым исходным кодом как Cosmos DB, Redis, SQL Azure, PostgreSQL и MySQL [23]. А также для создания алгоритмов искусственного интеллекта и машинного обучения, обеспечения безопасности инфраструктуры и многое другое [24].

Для использования Microsoft Azure предоставляет следующие тарифы : бесплатный тариф, общий и стандартный.

	Бесплатный	Общий	Стандартный
Цена	Бесплатно	$0.013 / час	$0.10 / час за CPU (PAYG)
Число сайтов	До 10 в регионе (всего 60)	До 100	До 500
Макс. масштаб	1 экземпляр	6 экземпляров	10 экземпляров
Хранилище	1GB (разделено между всеми сайтами)	1GB (разделено между всеми сайтами)	10GB (разделено между всеми сайтами)
CPU	1 ч/день. Разделяется между всеми сайтами в регионе	4 часа / день	Полные ресурсы
Память	1 GB для всех сайтов в регионе	512MB на сайт	Полные ресурсы
SQL	Включает 20MB SQL БД	Включает 20MB SQL БД	Включает 20MB SQL БД
MySQL	Включает 20MB БД	Включает 20MB БД Можно докупить в Azure Store.	Включает 20MB БД Можно докупить в Azure Store.
Сеть	Входящий – не ограничено Исходящий - 165MB/день (5GB/месяц)	Входящий – не ограничено Исходящий – стандартные цены Azure (при превышении 5GB/месяц)	Входящий – не ограничено Исходящий – стандартные цены Azure (при превышении 5GB/месяц)
SSL	Нет	Нет	Доступно
SLA	Нет	Нет	Доступно
Поддержка	Отсутствует	Отсутствует	99.9% ежемесячно

Рисунок 3.2 - Тарифы Microsoft Azure

Тарифы различаются разной ценой и доступными для разработчиков мощностями, предлагая гибко подходить к применению облака для размещения веб-приложений. Например, получить за несколько минут и использовать его бесплатно, масштабировать веб-сайт вверх и вниз по мере изменения его популярности [25,26,27].

3.3.3AWS

Amazon Web Services (AWS) — коммерческое публичное облако, поддерживаемое и развиваемое компанией Amazon с 2006 года. Предоставляет подписчикам услуги как по инфраструктурной модели (виртуальные серверы, ресурсы хранения), так и платформенного уровня (облачные базы данных, облачное связующее программное обеспечение, облачные бессерверные вычисления, средства разработки) [28].

Для машинного обучения Amazon предоставляет AWS Deep Learning AMI. AWS Deep Learning AMI предоставляет инфраструктуру и инструменты для ускорения глубокого обучения в облаке в любых масштабах. Возможность быстрого запуска в Amazon EC2 с предварительно установленными популярными платформами и интерфейсами глубокого обучения, такими как TensorFlow, Keras, что позволяет обучить сложные модели ИИ, экспериментировать с новыми алгоритмами или изучать новые навыки и методы [29,30,31].

4 Поведенческое моделирование нейронной сети

Перед передачей в сеть данные должны быть преобразованы в тензоры с вещественными числами. В настоящее время данные хранятся в виде файлов JPEG, поэтому их нужно подготовить для передачи в сеть, выполнив следующие шаги:

- Прочитать файлы с изображениями
- Декодировать содержимое из формата JPEG в таблицы пикселов RGB
- Преобразовать их в тензоры с вещественными числами.
- Масштабировать значения пикселов из диапазона [0; 255] в диапазон [0,1]

Для нейронных сетей предпочтительно передавать небольшие значения. Поэтому в пункте 4 такой указан масштаб.

В рамках проектирования был выбран язык программирования Python и фреймворк Keras. Keras обладает уже теми утилитами, которые позволят автоматизировать все эти четыре пункта. Keras.preprocessing.image без труда работает с изображениями. А класс ImageDataGenerator, позволит настроить быстро генераторы для автоматического преобразования файлов с изображениями в пакеты готовых тензоров.

Для сравнения рассмотрим сверточную нейронную сеть с генераторами и расширением данных и предварительно обученную нейронную сеть.

4.1 Генераторы

Генератором в языке Python называется объект, действующий как итератор.

Рассмотрим вывод нейронной сети с генератором. Нейронная сеть будет возвращать пакеты изображений с 28x28. В каждом пакете имеется 20 образцов. Исходные данные в модель будут передаваться с помощью

генератора, метода fit_generator (эквивалент метода fit). С помощью генератора данные будут генерироваться бесконечно, поэтому необходимо определить сколько образцов необходимо будет извлечь. В данном случае, как ранее описано, пакеты содержат по 20 образцов, поэтому для получения 2000 образцов потребуется извлечь 100 пакетов.

Создадим графики изменения точности и потерь модели по обучающим и проверочным данным в процессе обучения.

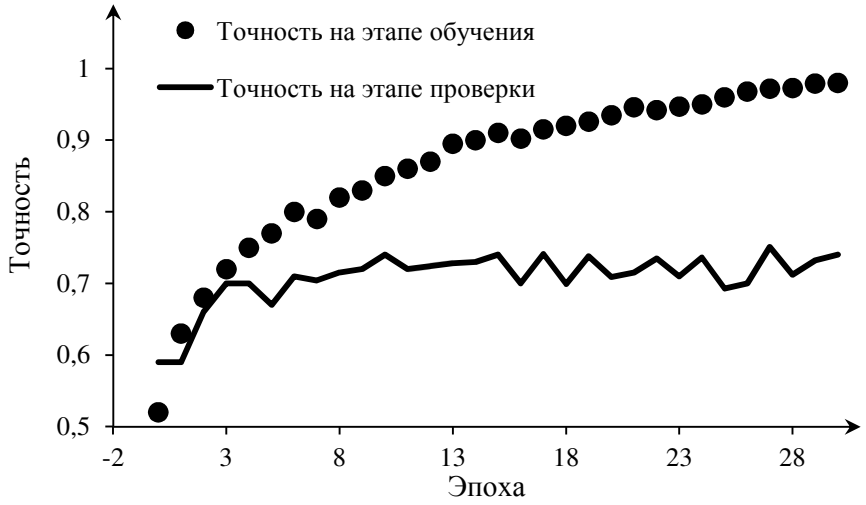


Рисунок 4.1 – График точности на этапах обучения и проверки

Рисунок 4.2 – График потерь на этапах обучения и проверки

На графиках четко наблюдается эффект переобучения. Точность на обучающих данных линейно растет и приближается к 100%, тогда как точность на проверочных данных останавливается на отметке 70-72%. Потери на этапе

проверки достигают минимума всего после пяти эпох и затем замирают, а потери на этапе на этапе обучения продолжают линейно уменьшаться, почти достигая 0.

Поскольку в данном сравнительном эксперименте относительно не много обучающих образцов. Поэтому проблема переобучения становится главной проблемой.

4.2 Расширение данных

Причиной переобучения является недостаточное количество образцов для обучения модели, способной обобщать новые данные. Имея бесконечный объем данных, можно было бы получить модель, учитывающую все аспекты распределения данных: эффект переобучения данных никогда бы не наступил.

Прием расширения данных реализует подход создания дополнительных обучающих данных из имеющихся путем трансформации образцов множеством случайных преобразований, дающих правдоподобные изображения. Цель состоит в том, чтобы на этапе обучения модель никогда не увидела одно и то же изображение дважды.

Даже если обучить новую сеть с использованием этих настроек расширения, то входные данные по-прежнему будут тесно связаны между собой. Причина: они получены из небольшого количества оригинальных изображений. Обучим сеть, задействовав расширение данных и прореживание. Выведем графики с результатами обученной нейронной сети.

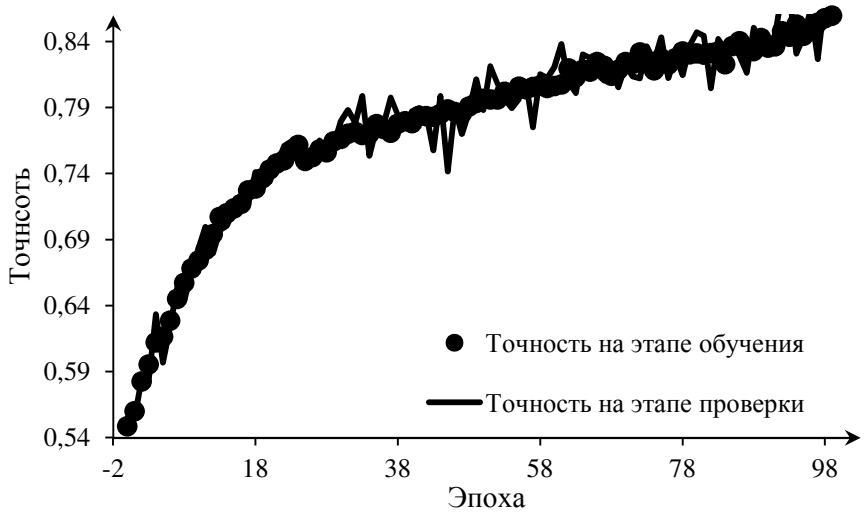


Рисунок 4.3 – График точности на этапах обучения и проверки

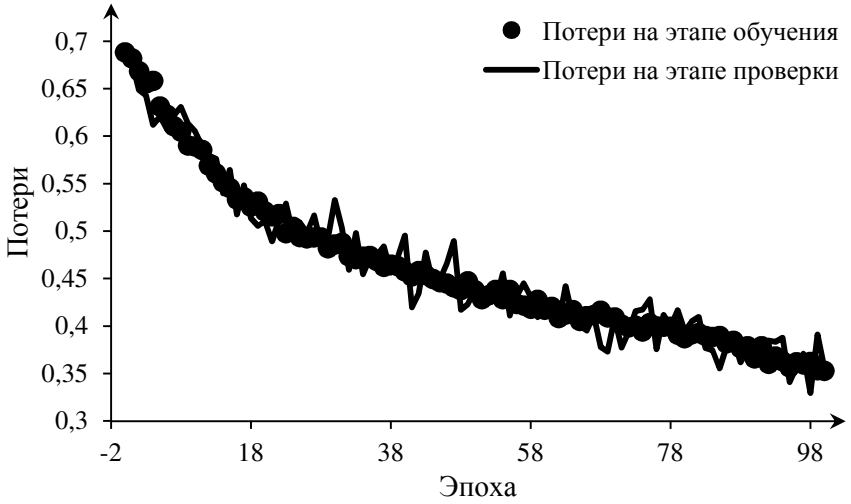


Рисунок 4.4 – График потерь на этапах обучения и проверки

Применив расширение данных и прореживание в практических результатах, произошло избавление от переобучения: кривые точности и потерь на этапе обучения близки к аналогичным кривым на этапе проверки. Достижение точности 82%. В сравнении алгоритма с использованием генератора улучшение произошло на 15%.

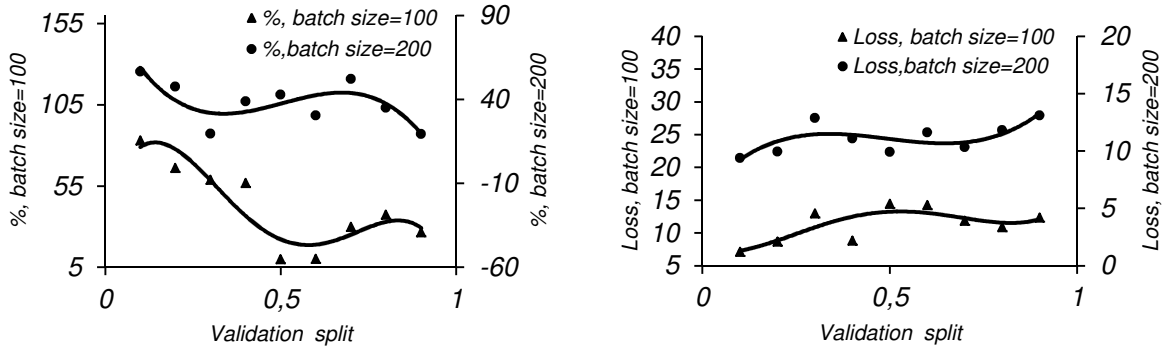
Для увеличения точности будет использоваться третий алгоритм: предварительно обученная модель.

4.3 Предварительно обученная нейронная сеть

Типичным и эффективным подходом к обучению на небольших наборах изображений является использование предварительно обученной сети.

Предварительно обученная сеть – это сохраненная сеть, прежде обученная на большом наборе данных, обычно в рамках масштабной задачи классификации изображений. Если этот исходный набор данных достаточно велик и достаточно обобщён, тогда пространственная иерархия признаков, изученных сетью может эффективно выступать в роли обобщенной модели видимого мира и быть полезной во многих разных задачах распознавания образов, даже если эти новые задачи будут связаны с совершенно иными классами, отличных от классов в оригинальной задаче. Такая переносимость изученных признаков между разными задачами – главное преимущество машинного обучения перед многими более старыми приемами поверхностного обучения, которое делает машинное обучение очень эффективным инструментом для решения задач с малым объемом данных.

Построим графики изменения.



а) б)

Рисунок 4.5 – а) – Точность; б) - Ошибка обучения

На графиках видно, что достижение точности ближе к 90%. Для достижения точности к 100% воспользуемся предварительно обученной нейронной сетью для распознавания рукописных цифр. Ниже рассмотрим структуру нейросети.

4.4 Архитектура программного обеспечения

Программа состоит из следующих модулей:

- mnist_nw.py – Файл со предварительно обученной сверточной нейронной сетью.
- mnist_model.json - содержит в себе структуру сети в файле. Этот файл создается в процессе обучения нейронной сети.
- mnist_model.h5 – файл с данными о весах. Создается в процессе обучения нейронной сети.
- mnist.py – точка входа в программу.

Взаимодействие модулей между собой представлена в блок-схеме ниже:

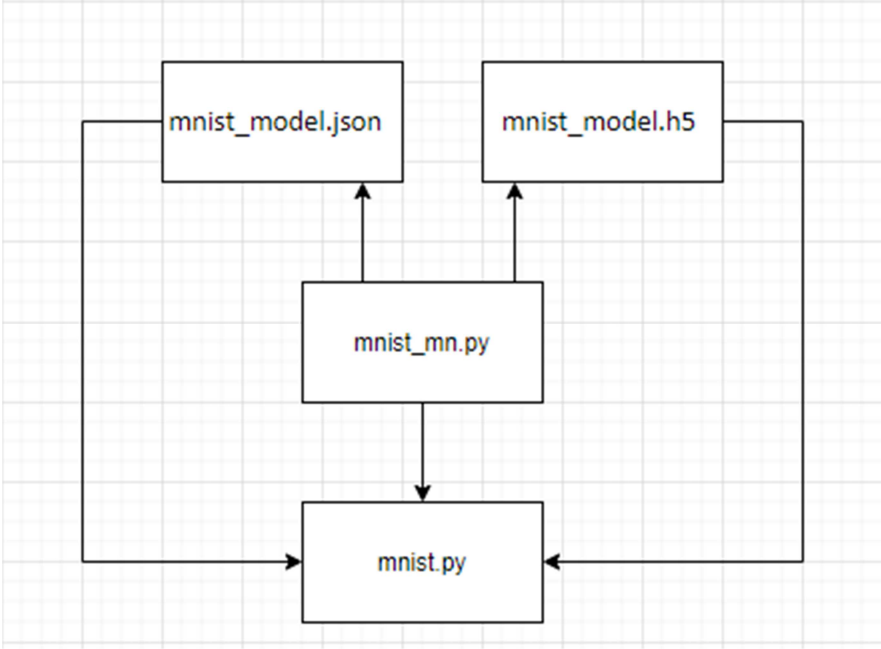


Рисунок 4.6 - Взаимодействие модулей в программе

Описание методов, используемых в файлах:

numpy.random.seed – Число повторяемости результатов.

mnist.load_data()- Загрузка ранее полученных данных. В данном случае данных MNIST. Эта же функция будет содержать в себе тренировочный набор, который будет необходим для обучения

reshape – метод для преобразования полученных изображений, который позволяет изменить форму тензора. Метод преобразует данные в трехмерный

массив, значениями в котором являются числа в интервале [0,255]. Который затем будет преобразован в другой тип данных в интервале [0,1].

To_categorical - прямое кодирование для форматирования категорий.

Conv2D - определяет размер шаблонов, извлекаемых из входных данных и глубину выходной карты признаков.

MaxPooling2D – указание слоя подвыборки. Выбирает максимальное значение из соседних.

Dropout – слой регуляризации, который позволяет нейросети исключить переобучение

Flatten() – слой для преобразования данных, который объединяет все тензоры.

Dense() – последовательность из двух слоев, которые являются тесно связанными нейронными слоями. Первый слой использует функцию активации relu; второй (и он же последний) слой – это 10-переменный слой потерь (softmax), возвращающий массив с 10 оценками вероятности. Каждая оценка определяет вероятность принадлежности к одному из 10 классов цифр. Аргумент, передающийся каждому слою, является числом скрытых нейронов слоя.

Batch size. Пакет или мини-пакет – небольшой набор образцов, обрабатываемых моделью одновременно. В процессе обучения один мини-пакет используется в градиентном спуске для вычисления одного изменения весов модели.

Fit. Метод, который пытается адаптировать модель под обучающие данные и начинает перебирать обучающие данные по мини-пакетам по 200 образцов.

Categorical crossentropy - функция потерь, которая используется в качестве сигнала обратной связи для обучения весовых тензоров, и которую этап обучения стремится свести к минимуму. Точные правила, управляющие конкретным применением градиентного спуска, определяются оптимизатором adam, который передается во втором аргументе и выполняет 10 итераций.

Для каждого мини-пакета сеть вычисляет градиенты весов с учетом потерь в пакете и изменяет значения весов в соответствующем направлении.

Prediction – предсказание. Результат работы модели.

4.5 Результаты эксперимента

Целью экспериментальной части работы является обеспечение максимальной точности распознавания нейронной сетью рукописных числовых знаков в диапазоне от 0 до 9. При этом в процессе работы нейросетевого алгоритма нагрузка на аппаратную часть вычислительной машины должна быль адекватной и равномерной. К аппаратной части машины в данном случае стоит отнести центральный и графический процессоры.

В процессе эксперимента основными задачами ставились исследование влияния на точность выходных данных таких параметров как:

- Количество слоев сверточной нейросети;
- Количество циклов обучения (эпох);
- Количество сетов эпохи (Batch size);
- Соотношение тренировочных и обучающих объектов обучающего множества (Validation split).

При построении модели нейронной сети следует обратить внимание на одну из основных проблем – переобучение. Переобучение возникает в случае слишком долгого обучения, недостаточного числа обучающих примеров или переусложненной структуры нейронной сети.

Первым этапом эксперимента является определение зависимости ошибки обучения от параметра Validation split. Варьирование данного параметра служит одним из вариантов борьбы с переобучением сети. Также он отвечает за деление обучающего множества на два множества – обучающее и тестовое.

Количество эпох и слоев сети следует выбрать минимальное для исключения влияния этого параметра на функцию ошибок. Количество сетов

эпохи следует выбрать исходя из возможностей аппаратной части вычислительной машины. Для примера достаточно выбрать 2 параметра Batch size: 100 и 200. Диапазон варьирования параметра Validation split принят в пределах от 0,1 до 0,9.

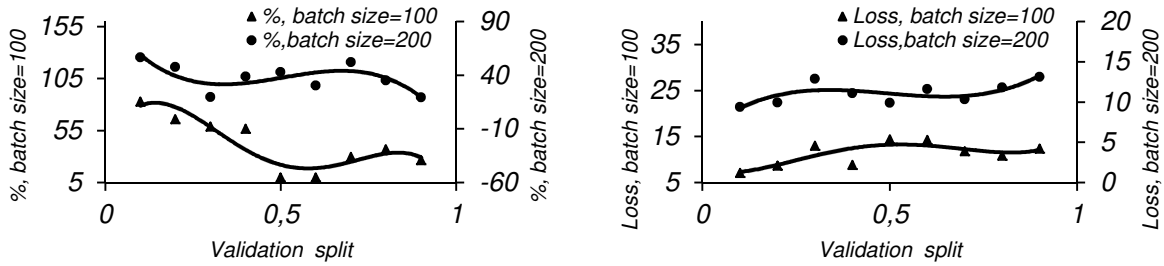
Таблица 4.1 - Результаты варьирования Validation split однослойной сети

Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	точность, %
1	100	0,1	207	7,1487	0,5522	2,615	0,8355	83,26
	100	0,2	160	8,7339	0,4549	5,4367	0,6607	66,26
	100	0,3	174	13,0207	0,1919	12,9488	0,5645	58,98
	100	0,4	152	8,8789	0,4462	6,9959	0,5645	56,98
	100	0,5	151	14,4622	0,1025	14,4751	0,1019	10,1
	100	0,6	118	14,3056	0,1119	14,5014	0,1003	10,32
	100	0,7	109	11,8767	0,2621	11,384	0,2933	29,94
	100	0,8	80	10,8949	0,3209	10,0635	0,3746	37,39
	100	0,9	61	12,397	0,2284	11,8742	0,2622	26,71

Таблица 4.2 - Результаты варьирования Validation split однослойной сети

Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	точность, %
1	200	0,1	169	9,3948	0,4147	6,9473	0,5673	56,73
	200	0,2	155	9,9526	0,381	8,3889	0,4782	47,77
	200	0,3	141	12,8819	0,2002	13,011	0,1927	19,63
	200	0,4	131	11,1136	0,3089	9,8834	0,386	38,96
	200	0,5	114	9,9274	0,3822	9,1189	0,4333	42,93
	200	0,6	100	11,6483	0,2756	11,2328	0,3026	30,51
	200	0,7	86	10,3461	0,3534	7,7905	0,5133	52,22
	200	0,8	70	11,8339	0,2626	10,4645	0,3489	35,19
	200	0,9	56	13,1281	0,1815	12,9804	0,1944	19,46

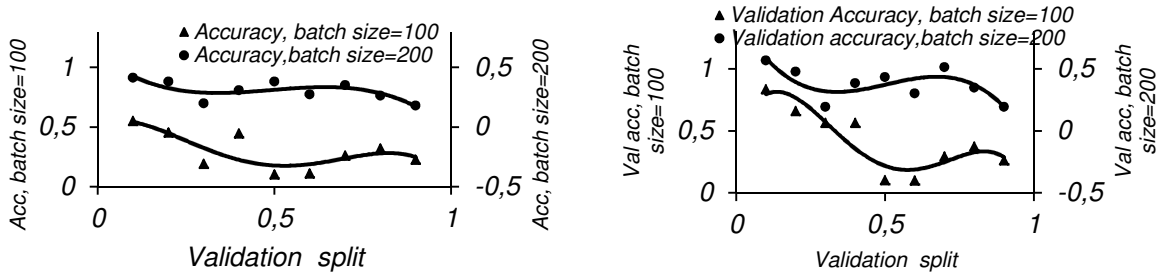
Результаты сводной таблицы наглядно демонстрируют графические зависимости, приведенные ниже.



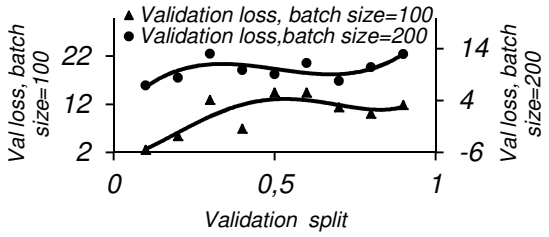
а) б)

Рисунок 4.7 – а) – Точность; б) - Ошибка обучения

При разных количествах сетов в эпохе можно заметить снижение точности выходных данных при Validation split, стремящемуся к 1. Это говорит о том, что чем больше объектов обучения общего множества выделено на обучение сети, тем точность обучения выше. Функция ошибок Loss ведет себя несколько иначе. При малых значениях Batch size четкой корреляции и тенденции не прослеживается, в то время как при увеличении Batch size заметно увеличение точности выходных данных при соотношении обучающего и тестового множеств 50/50. При других соотношениях точность обучения имеет тенденцию к снижению.



а) б)



в)

Рисунок 4.8 – а) - Точность обучения; б) - Точность проверки; в) - Ошибка проверки

Построенная нейронная сеть, основанная на 1 сверточном слое, показывает недостаточную точность выходных данных при оптимальном количестве сетов одной эпохи во всем диапазоне Validationsplit. Увеличить точность алгоритма возможно путем увеличения слоев нейронной сети.

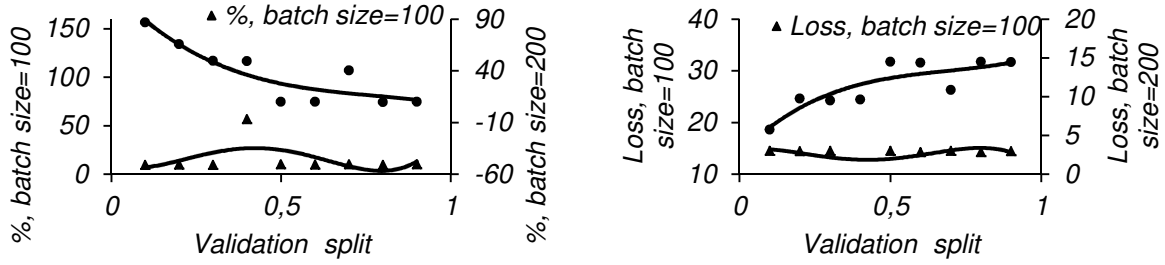
Для сравнения достаточно внедрить в сеть еще один слой и выбрать оптимальные параметры сети.

Таблица 4.3 - Результаты варьирования Validation split двухслойной сети

Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	точность, %
1	100	0,1	535	14,5482	0,0972	14,5197	0,0992	9,82
	100	0,2	418	14,4952	0,1005	14,4499	0,1035	10,1
	100	0,3	388	14,5323	0,092	14,5573	0,0968	9,82
	100	0,4	352	9,6922	0,395	6,9544	0,5672	56,96
	100	0,5	314	14,5329	0,0982	14,5009	0,1003	10,32
	100	0,6	272	14,3135	0,1108	14,534	0,0983	10,09
	100	0,7	236	14,5619	0,0964	14,4994	0,1004	10,32
	100	0,8	197	14,3308	0,1093	14,5318	0,0984	9,74
	100	0,9	160	14,5277	0,0975	14,5132	0,0996	10,32

Таблица 4.4 - Результаты варьирования Validation split двухслойной сети

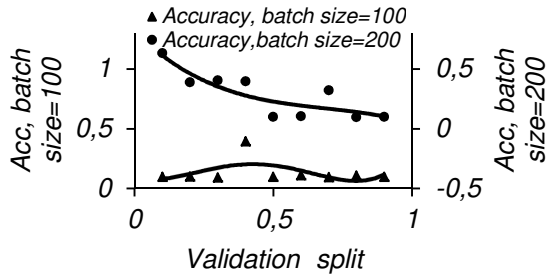
Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	точность, %
1	200	0,1	445	5,7497	0,6329	1,8936	0,8757	86,83
	200	0,2	412	9,7557	0,3898	5,5786	0,6488	65,88
	200	0,3	358	9,5275	0,4057	8,2191	0,4893	49,89
	200	0,4	338	9,6307	0,3979	8,1579	0,4925	49,44
	200	0,5	298	14,5042	0,0995	14,5009	0,1003	10,32
	200	0,6	269	14,4049	0,1052	14,4275	0,1049	10,28
	200	0,7	231	10,8672	0,3217	9,7025	0,3973	40,54
	200	0,8	193	14,5131	0,0986	14,554	0,097	9,82
	200	0,9	163	14,483	0,0988	14,5132	0,0996	10,32



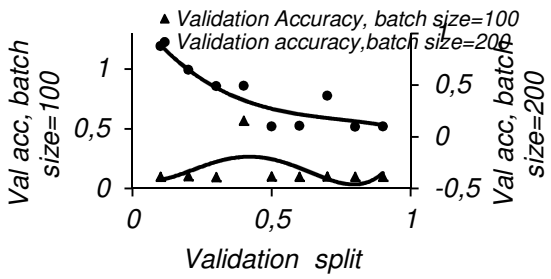
а) б)

Рисунок 4.9 – а) - Точность обучения; б) - Точность проверки

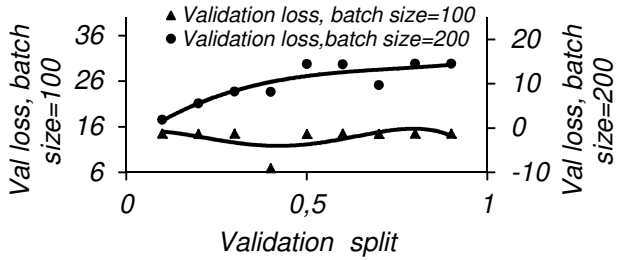
По аналогии с предыдущей однослойной сетью более выраженный спад точности имеет нейронная сеть с большим количеством Batchsize.



а)



б)



в)

Рисунок 4.10 – а) - Точность обучения; б) - Точность проверки; в) - Ошибка проверки

В сравнении с однослойной нейросетью, в двухслойной наблюдается наиболее ярко выраженные тенденции кривых. В то же время одни и те же показатели качества обеих сетей коррелируют между собой. К примеру, показатель точности сети, содержащей один сверточный слой, имеет тенденцию к снижению качественной характеристики практически независимо от количества сетов в одной эпохе. Этот же показатель точности имеет более выраженную тенденцию к спаду в нейронной сети с двумя слоями. При этом наибольшую крутизну кривой имеет конфигурация сети с большим числом сетов.

Величина Batchsize в значительной степени влияет на производительность сети и загруженность локальной машины. Конфигурация с Batch size = 200 для данной машины вполне приемлема, однако, такая нейросеть не обладает достаточной точностью выходных данных. Поэтому

следует повысить количество эпох вплоть до десяти, зафиксировав при этом величину Batchsize в размере двухсот объектов.

Таблица 4.5 - Сводная таблица двухслойной сети

Epoch	batch_size	Valid_split	time	loss	acc	val_loss	val_acc	Точность, %
10	200	0,1	446,4	2,8768	0,81942	1,92426	0,87955	87,38
	200	0,2	332,1	0,05536	0,98267	0,03438	0,98985	99,17
	200	0,3	342,3	3,50257	0,78082	2,7259	0,8328	95,66
	200	0,4	333,7	0,7712	0,93217	0,13884	0,97466	99,08
	200	0,5	218,8	14,52574	0,09878	14,51122	0,09967	10,09
	200	0,6	210,8	5,06448	0,68337	4,37316	0,72703	75,84
	200	0,7	174,3	8,05972	0,49921	7,33944	0,54388	64,03
	200	0,8	154,2	14,52728	0,09863	14,554	0,097	9,82
	200	0,9	126	12,45534	0,22568	12,03594	0,25205	27,88

Как видно из предыдущих таблиц наибольшей точностью обладает сеть с максимальным количеством (десятью) сверточных слоев.

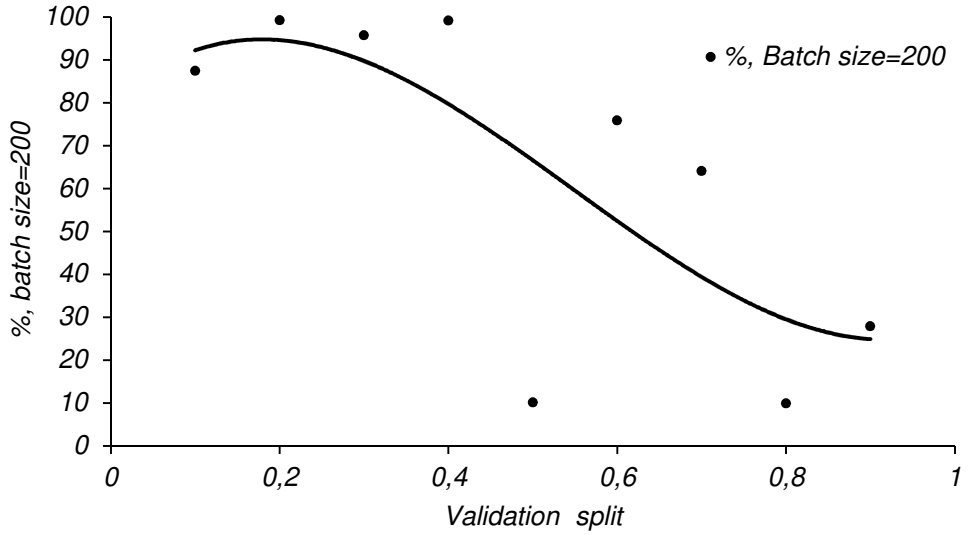


Рисунок 4.11 – Точность

В зависимости от соотношения обучающего и тестового множества точность выходных данных имеет тенденцию преимущественно к падению. Однако максимальная точность достигается при соотношении 20% - тестовых данных и 80% - обучающих.

В итоге структура окончательной нейросети содержит в себе:

- Входной слой из 784 нейронов;
- Скрытый слой из 100 нейронов;

– Второй скрытый слой состоящий из 75 нейронов;

– И выходной слой, состоящий 10 нейронов. Это будут цифры в диапазоне от 0 до 9.

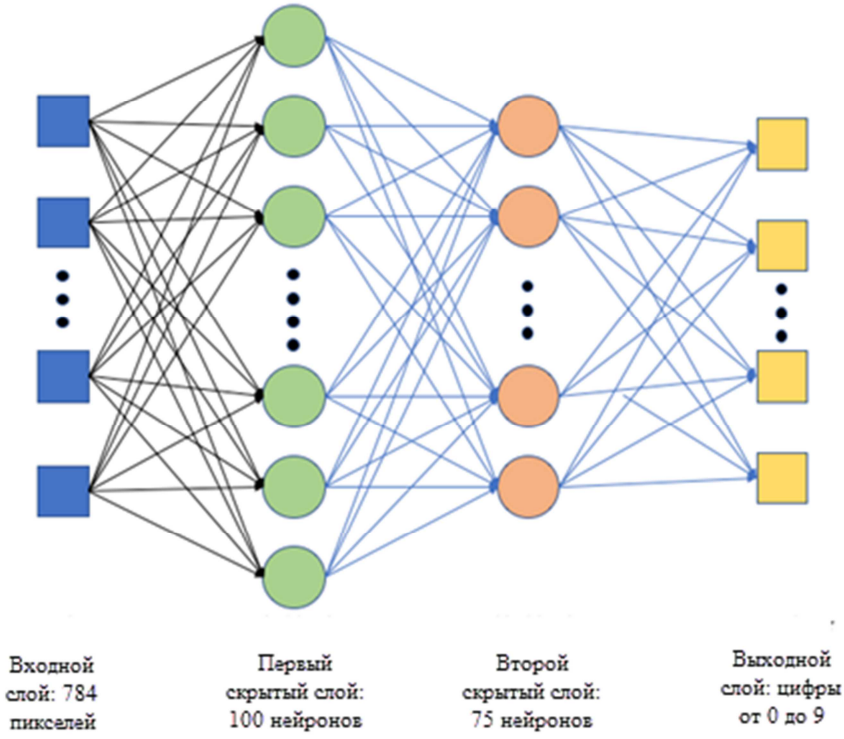


Рисунок 4.12 - Структура сети

Входные и выходные данные не изменились по причине того, что на вход стандартно подается изображение размером 28x28, что будет равно 784 нейронам. Выход будет стандартным – это цифры от нуля до девяти.

Данная нейросеть со своей структурой будет использоваться в дальнейших экспериментах с оптимизацией аппаратных ресурсов.

5 Оптимизация аппаратных ресурсов

5.1 Google Colaboratory

Оптимизация алгоритма нейронной на аппаратном уровне предполагает использование дополнительных ресурсов машины. Для сравнения точности и времени исполнения кода нейросетевого алгоритма двух вычислительных машин, была использована бесплатная система облачной обработки данных Google Colaboratory.

Таблица 5.1 – Результаты работы облачных вычислений

Epoch	batch_size	Valid_split	time	loss	acc	val_loss	val_acc	точность, %
1	200	0,2	158	0,263	0,9164	0,0598	0,982	99,37
2	200		156	0,0673	0,9792	0,0434	0,9874	
3	200		155	0,0452	0,9859	0,0339	0,9893	
4	200		155	0,0393	0,9874	0,0308	0,9902	
5	200		155	0,0307	0,9906	0,0296	0,9916	
6	200		155	0,0274	0,9914	0,0296	0,9909	
7	200		154	0,0324	0,9925	0,0268	0,9931	
8	200		154	0,0182	0,994	0,0259	0,9926	
9	200		154	0,0203	0,9932	0,0265	0,9931	
10	200		154	0,0161	0,9949	0,0254	0,9926	

Стоит отметить, что в пределах десяти эпох обучения нейронной сети показатели качества работы алгоритма практически идентичны показателям локальной машины. Однако разница заключается во времени исполнения кода. Благодаря тому, что аппарат облачных вычислений предоставляет возможность использования большего объема оперативной памяти, время на исполнение кода сокращается примерно в 2 – 3 раза.

Аппаратная оптимизация процесса исполнения кода на локальной машине может включать в себя как применение многопоточности, а также распараллеливание на графический процессор CUDA.

Результаты работы сети с применением многопоточности приведены в таблице ниже:

Таблица 5.2 - Результаты работы многопоточности

Epoch	time	loss	acc	val_loss	val_acc	Точность, %
10	287	0.2626	0.9161	0.0652	0.9803	99,43
	283	0.0706	0.9788	0.0430	0.9871	
	287	0.0475	0.9857	0.0367	0.9898	
	282	0.0407	0.9871	0.0314	0.9907	
	283	0.0331	0.9892	0.0317	0.9910	
	281	0.0291	0.9904	0.0312	0.9909	
	282	0.0253	0.9921	0.0335	0.9908	
	284	0.0203	0.9931	0.0302	0.9915	
	288	0.0212	0.9927	0.0282	0.9933	
	287	0.0164	0.9948	0.0268	0.9923	

В отличие от облачных вычислений, локальная машина справилась с алгоритмом не так быстро, однако значительно быстрее, чем без применения многопоточности.

5.2 Azure Machine Learning

Для использования Azure Machine Learning Notebook VM нужно:

- Установить Azure Machine Learning SDK;
- Создать конфигурационный файл;
- Инициализировать рабочую область в скрипте.

Использовался стандартный тариф. Результаты представлены в таблице ниже.

Таблица 5.3 - Результат работы в облачном сервисе

Epoch	batch_size	Valid_split	time	loss	acc	val_loss	val_acc	точность, %
1	200	0,2	102	0,3529	0,8883	0,0719	0,9785	99,32
2	200		102	0,0815	0,9746	0,0486	0,9862	
3	200		100	0,0561	0,9821	0,0404	0,9878	
4	200		100	0,0438	0,9863	0,0361	0,9887	
5	200		98	0,0374	0,9881	0,0364	0,9893	
6	200		98	0,03	0,9909	0,0297	0,9903	
7	200		98	0,0265	0,9914	0,0277	0,9916	
8	200		96	0,0229	0,9931	0,029	0,9918	
9	200		97	0,0213	0,9933	0,0267	0,9921	
10	200		96	0,0184	0,994	0,0248	0,9929	

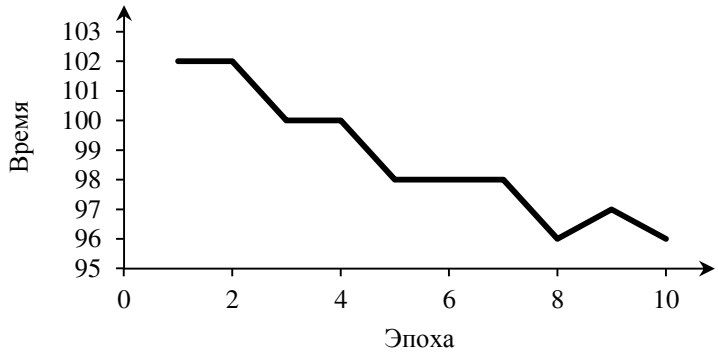


Рисунок 5.1 – График оптимизации Google Colaboratory

При использовании полных ресурсов в тарифе Microsoft Azure были достигнуты лучше результаты, чем при использовании Google Colaboratory. Самый главный результат эксперимента – это уменьшение времени вычисления. Сейчас в среднем с использованием Microsoft Azure время вычисления составляет 97 секунд.

5.3 AWS

Amazon также предоставляет сервис для глубокого обучения - amazon sagemaker. Но для начала воспользуемся обычными вычислительными мощностями, которые AWS предоставляют. Т.к. необходимо было ускорить вычисления, для этого было использован тип инстанса ускоренные вычисления (inf1). Данный инстанс отлично подходит для распознавания текста.

Размер инстанса	Виртуальные ЦПУ	Память (ГиБ)	Хранилище	Микросхемы Inferentia	Межсоединение микросхем Inferentia	Пропускная способность сети	Пропускная способность EBS
inf1.xlarge	4	8	Только EBS	1	н/п	До 25 Гбит/с	До 4,75 Гбит/с
inf1.2xlarge	8	16	Только EBS	1	н/п	До 25 Гбит/с	До 4,75 Гбит/с
inf1.6xlarge	24	48	Только EBS	4	Да	25 Гбит/с	4,75 Гбит/с
inf1.24xlarge	96	192	Только EBS	16	Да	100 Гбит/с	19 Гбит/с

Рисунок 5.2 – Характеристики инстанса

Результат использования представлен в таблице ниже.

Таблица 5.4 - Результат работы в облачном сервисе

Epoch	**batch_size**	**Valid_split**	**time**	**loss**	**acc**	**val_loss**	**val_acc**	**точность, %**
1	200	0,2	68	0,0398	0,9879	0,0838	0,9898	99,33
2	200		68	0,0335	0,9895	0,0518	0,9907	
3	200		68	0,0286	0,9911	0,0411	0,9917	
4	200		65	0,0253	0,9924	0,0355	0,9917	
5	200		66	0,0218	0,9931	0,0326	0,9922	
6	200		63	0,02	0,9937	0,0305	0,9922	
7	200		60	0,0188	0,9941	0,03	0,993	
8	200		59	0,0172	0,9942	0,0277	0,9933	
9	200		64	0,0152	0,9952	0,0277	0,992	
10	200		63	0,0145	0,9951	0,0281	0,994	

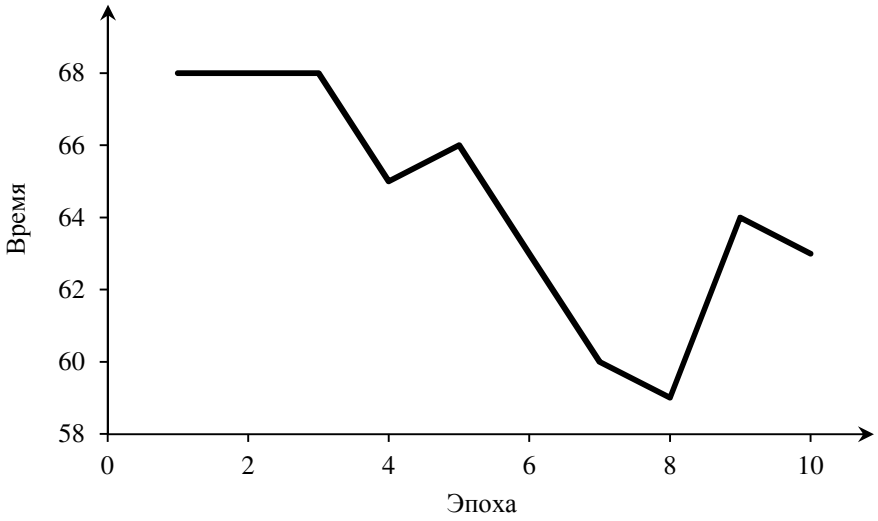


Рисунок 5.3 – График оптимизации AWS

При использовании выбранного инстанса в AWS были достигнуты наивысшие результаты, чем при использовании двух ранее описанных облачных сервиса. Целью эксперимента было уменьшение времени и при использовании мощностей AWS было достигнуто среднее время обучения равное 64 секунды.

6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

6.1 Организация и планирование работ

В процессе выполнения работ по проектированию для конкретной задачи необходимо определить список проводимых работ, количество исполнителей, а также продолжительность данных работ. Так как число исполнителей не превышает двух, линейный график работ является наиболее удобным и компактным способом представления данных планирования. Список исполнителей включает в себя научного руководителя работ (НР) и непосредственного исполнителя (И). График выполнения работ с указанием перечня задач приведен в (Таблица 6.1).

Таблица 6.1 - Перечень этапов работ и распределение исполнителей

Этапы работы	**Исполнители**	**Загрузка исполнителей**
Составление и утверждение технического задания, постановка целей и задач исследования	НР, И	НР – 90% И – 10%
Изучение литературы, подготовка материалов по теме	НР, И	НР – 5% И – 95%
Составление календарного плана	НР, И	НР – 20% И – 80%
Выбор и сравнение рабочих платформ	НР, И	НР – 10% И – 90%
Разработка алгоритма и синтез программного кода	И	И – 100%
Тестирование и получение результатов	И	И – 100%
Оформление расчетно-пояснительной записки	И	И – 100%
Подведение итогов	НР, И	НР – 10% И – 90%

6.1.1Продолжительность этапов работ

Для определения продолжительности выполнения этапов работ следует воспользоваться опытно-статическим методом.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и рассчитывается с помощью длительности работ (ожидаемой длительности $t_{ожс}$) рабочих в календарных днях:

$$t_{ожс} = \frac{3t_{\min} + 2t_{\max}}{5}$$

где t_{min} – минимальная продолжительность работы, дн.;

t_{max} – максимальная продолжительность работы, дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчёт продолжительности выполнения каждого этапа в рабочих днях ($T_{РД}$) выполняется по формуле:

$$T_{РД} = \frac{t_{ожс}}{K_{ВН}} \cdot K_{Д}$$

где $K_{ВН}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{ВН} = 1$;

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д}$ = 1–1,2; в этих границах конкретное значение принимает сам исполнитель).

Для построения линейного графика необходимо перевести длительность этапов в рабочих днях в календарные дни. Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях.

Для расчёта длительности каждого из этапов работ в календарных днях необходимо рассчитать коэффициент календарности T_K используя формулу:

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ}$ = 365);

$T_{ВД}$ – выходные дни ($T_{ВД}$ = 52);

$T_{ПД}$ – праздничные дни ($T_{ПД}$ = 10).

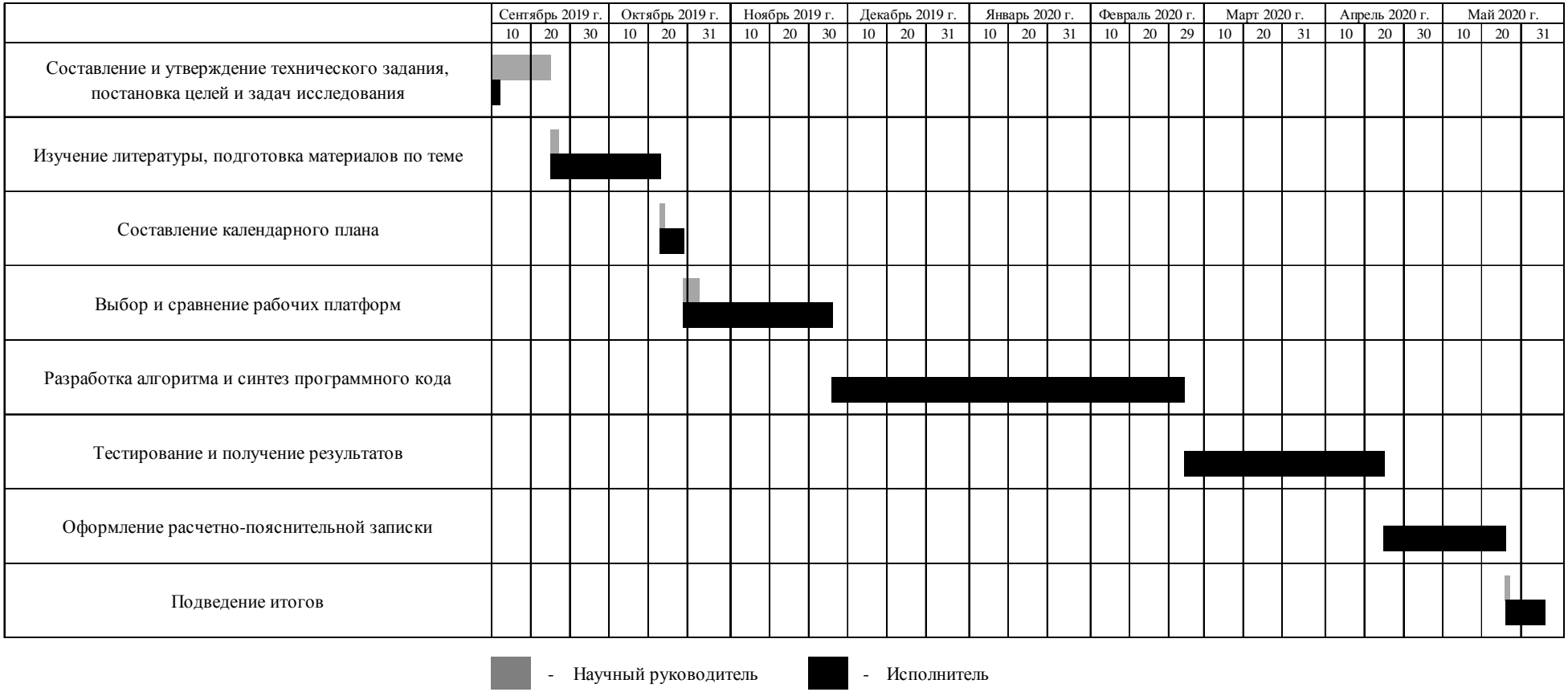
Воспользовавшись данными из (Таблица 6.1), приведенными выше формулами, произведем расчет продолжительности выполнения работ научным руководителем и исполнителем в календарных днях. Результаты расчетов представлены в (Таблица 6.2).

Таблица 6.2 - Расчет трудозатрат на выполнение проекта

Этапы работы	**Испол-нители**	**Продолжитель-ность работ, дни**			**Трудоемкость работ по исполнителям, чел.- дн.**			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Составление и утверждение технического задания, постановка целей и задач исследования	НР, И	7	20	12,2	12,078	1,342	14,549	1,6166
Изучение литературы, подготовка материалов по теме	НР, И	20	25	22	1,21	22,99	1,4576	27,694
Составление календарного плана	НР, И	5	7	5,8	1,276	5,104	1,5371	6,1484
Выбор и сравнение рабочих платформ	НР, И	30	35	32	3,52	31,68	4,2403	38,162
Разработка алгоритма и синтез программного кода	И	60	80	68	0	74,8	0	90,106
Тестирование и получение результатов	И	30	50	38	0	41,8	0	50,353
Оформление расчетно-пояснительной записки	И	20	30	24	0	26,4	0	31,802
Подведение итогов	НР, И	7	10	8,2	0,902	8,118	1,0866	9,7791
Итого				210,2	18,986	212,23	22,871	255,66

Для наглядного отображения графика и распределения работ между участниками проекта использована диаграмма Ганта (Таблица 6.3).

Таблица 6.3 - Календарный план-график проведения работ



6.2 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса мероприятий, составляющих содержание разработки. При формировании бюджета используется группировка затрат по следующим статьям:

- материалы и покупные изделия;
- заработная плата;
- отчисления в социальные фонды;
- амортизационные отчисления;
- накладные расходы.

6.2.1 Расчет затрат на материалы

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ над объектом проектирования.

На разных этапах работ требуется использование ряда программных продуктов, таких как Microsoft Word, Excel, Visual Studio и т.д. Большинство данных продуктов предоставляются ТПУ бесплатно для студентов. Для проведения тестирования разработанного программного обеспечения необходимо использование облачных сервисов Microsoft Azure, AWS, Google Colaboratory и др. Некоторые сервисы оказывают только платные услуги. Кроме того имеются расходы на канцелярские принадлежности и услуги копировальных центров. В статьи расходов не входит компьютерная и другая офисная техника, так как к ним имеется свободный доступ в лабораториях ТПУ.

Таблица 6.4 – Материальные затраты

Наименование материалов	**Цена за ед, руб.**	**Количество**	**Сумма, руб**
Microsoft Azure	5850	1	5850
AWS	10,05	120	1206
Бумага для принтера	240	1	240
Картридж для принтера	2500	1	2500
Услуги копировального центра	60,20	1	60,20
Итого			9856,20

6.2.2Расчет заработной платы

Расчет заработной платы включает в себя доходы научного руководителя и исполнителя научно-исследовательской работы. Помимо заработанной платы доходы исполнителя и научного руководителя включают в себя премии. Расчет заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Для расчета основной заработанной платы необходимо знать величину месячного оклада работника. Такие данные предоставлены ТПУ. Для научного руководителя, имеющего звание доцента и ученую степень к.т.н. месячный оклад без учета районного коэффициента установлен в пределах 33664.00 руб. оклад для инженера-исследователя установлен на уровне 9489 руб. без учета районного коэффициента.

Среднедневная заработная плата рассчитывается по формуле:

$$ЗП_{ДН-Т} = \frac{МО}{25,083}$$

где $МО$ – месячный оклад сотрудника. Учитывается, что в году 301 рабочий день и, следовательно, в месяце в среднем 25,083 рабочих дня (при шестидневной рабочей неделе).

Также был принят во внимание коэффициент, учитывающий коэффициент по премиям $K_{ПР}$ = 1,1, районный коэффициент $K_{РК}$ =1,3 и коэффициент дополнительной заработанной платы $K_{доп\ ЗП}$ =1,188. Общий коэффициент равен:

$$K = K_{ПР} \cdot K_{РК} \cdot K_{допЗП} = 1{,}1 \cdot 1{,}3 \cdot 1{,}188 = 1{,}699$$

Таблица 6.5 - Затраты на заработную плату

Исполнитель	**Оклад, руб./мес.**	**Среднедневная ставка, руб./раб. день**	**Затраты времен, раб. дни**	**Коэффициент**	**Фонд з/платы, руб**
НР	33 664	1342,10	19	1,699	43288,46
И	9489	378,30	212	1,699	136398,12
Итого:					179686,59

6.2.3Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30,2 % от полной заработной платы по проекту:

$$C_{соц} = C_{ЗП} \cdot K_{СОЦ}$$

где $K_{СОЦ}$ – коэффициент, учитывающий размер отчислений из заработной платы. Данный коэффициент включает в себя:

- отчисления в пенсионный фонд;
- на социальное страхование;
- на медицинское страхование.

Таблица 6.6 – Социальные отчисления

Работник	**Заработанная плата ($C_{ЗП}$), руб**	**Коэффициент отчислений ($K_{СОЦ}$)**	**Величина отчислений ($C_{СОЦ}$), руб.**
НР	43288,47	0,32	13852,31
И	136398,13		43647,40
Итого			57499,71

6.2.4Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{эл.об} = P_{об} \cdot t_{об} \cdot Ц_э$$

где $P_{об}$ – мощность, потребляемая оборудованием, кВт;

$Ц_э$ – тариф на 1 кВт·час;

$t_{об}$ – время работы оборудования, час.

Для ТПУ $Ц_э$ = 6,59 руб./кВт·час (с НДС), для жилого помещения $Ц_э$ = 2,45 руб./кВт·час.

Время работы оборудования определяется по формуле:

$$t_{об} = T_{РД} \cdot K_t$$

где $K_t \leq 1$– коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{РД}$, определяется исполнителем самостоятельно. В ходе выполнения работы были задействованы три компьютера: личный компьютер исполнителя, компьютер лаборатории ТПУ, компьютер научного руководителя, а также лазерный принтер.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном} \cdot K_c$$

где $P_{ном}$. – номинальная мощность оборудования, кВт;

$K_C \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_C = 1$.

В соответствие с данными формулами определим затраты на потребленную оборудованием электроэнергию и сведем результаты расчетов в таблицу (Таблица 6.7).

Таблица 6.7 - Затраты на электроэнергию

Наименование оборудования	**Время работы оборудования $t_{ОБ}$, час**	**Потребляемая мощность $P_{ОБ}$, кВт**	**Затраты $Э_{ОБ}$, руб.**
Персональный компьютер ТПУ	245	0,45	727,84
Персональный компьютер HP	150	0,45	445,92
Персональный компьютер исполнителя	2025	0,33	1637,08
Струйный принтер	0,8	0,15	0,28
Итого:			2811,12

6.2.5Расчет амортизационных расходов

Амортизационные расходы технических средств за весь период их использования может быть рассчитан по формуле:

$$C_{АМ} = \frac{H_A \cdot Ц_{ОБ} \cdot t_{РФ} \cdot n}{F_Д}$$

где H_A – годовая норма амортизации единицы оборудования;

$Ц_{ОБ}$ – балансовая стоимость единицы оборудования с учетом ТЗР либо величина действующей цены, содержащейся в ценниках, прейскурантах и т.п.;

$F_Д$ – действительный годовой фонд времени работы соответствующего оборудования.

Персональный компьютер и принтер входят в группу – вычислительная техника, следовательно, они имеют срок полезного использования 2-3 года. Так как к сроку начала работ компьютер исполнителя и принтер эксплуатировались более 6, то срок их полезного использования истек, а следовательно, амортизационные расходы на ПК и принтер равны нулю. Результаты расчетов приведены в таблице (Таблица 6.8).

Таблица 6.8 - Затраты на амортизацию оборудования

Наименование оборудования	**Годовая норма амортизации (H_A), 1/год**	**Балансовая стоимость ($Ц_{ОБ}$), руб**	**Фактическое время работы($t_{рф}$), дни**	**Действительный годовой фонд времени работы ($F_Д$), час**	**$C_{АМ}$, руб**
Персональный компьютер ТПУ	0,33	32000	245	2424	1080,02
Персональный компьютер HP	0,4	29000	150		719,59
Персональный компьютер исполнителя	0	24000	2025		0
Струйный принтер	0	14000	0,76		0
Итого					1799,61

6.2.6Расчет прочих расходов

Накладные расходы учитывают все затраты, не вошедшие в предыдущие статьи расходов: оплата связи, транспорта и т. д. Величину коэффициента накладных расходов можно принять в размере 10% от суммы всех предыдущих расходов:

$$C_{ПРОЧ} = \left(C_{МАТ} + C_{ЗП} + C_{СОЦ} + C_{ЭЛ.ОБ} + C_{АМ} \right) \cdot 0{,}1 = 25165{,}23 \text{ руб.}$$

6.2.7Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта. Смета затрат на разработку проекта приведена в таблице (Таблица 6.9).

Таблица 6.9 – Общая стоимость разработки

Статья затрат	**Условное обозначение**	**Сумма, руб.**
Материалы и покупные изделия	$С_{мат}$	9856,20
Основная заработная плата	$С_{зп}$	179686,59
Отчисления в социальные фонды	$С_{соц}$	57499,71
Расходы на электроэнергию	$С_{эл.}$	2811,12
Амортизационные отчисления	$С_{ам}$	1799,61
Прочие расходы	$С_{проч}$	25165,32
Итого:	$С$	276818,55

Таким образом, затраты на разработку составили $С$ = 276818,55 руб.

- Расчет прибыли, НДС и цены разработки НИР

Прибыль в размере 20% от полной себестоимости проекта составляет:

$$П = С \cdot 0,2 = 55363,71 \text{ руб.}$$

НДС составляет 20% от суммы затрат на разработку и прибыли. В нашем случае это:

$$С_{НДС} = (С + П) \cdot 0,2 = 66436,45 \text{ руб.}$$

Сумма полной себестоимости, прибыли и НДС соответствует цене разработки НИР, в нашем случае:

$$С_{НИР} = С + П + С_{НДС} = 398618,72 \text{ руб.}$$

6.3 Оценка экономической эффективности проекта

Данный проект по разработке методов и алгоритмов, входящих в программный комплекс, осуществляющий распознавание изображений и рукописных символов с использованием машинного обучения планируется

использовать в областях визуального анализа данных бумажной документации какого-либо предприятия, где необходим перевод данных с бумажного носителя в электронный вид. Ожидаемый экономический эффект проекта может носить как коммерческий, так и бюджетный характер, так как разработанный метод является составной частью продуктов предприятия, повышает эффективность и надёжность их работы, и, следовательно, влияет на стоимость и объёмы продаж. Это выражается в дополнительной прибыли предприятия. Экономический эффект бюджетного характера связан с последствиями осуществления проекта для федерального, регионального и местного бюджетов, коммерческий эффект может быть достигнут путем внедрения проекта в коммерческую деятельность того или иного предприятия с дальнейшим получением прибыли.

Актуальным аспектом качества выполненного проекта является соотношение обусловленного им экономического результата (эффекта) и затрат на разработку проекта. Следовательно, здесь мы имеем дело с частным случаем задачи оценки экономической эффективности инвестиций, т.е. вложением денежных средств в предприятие. Посредством правильной инвестиционной политики организации достигают своих стратегических и тактических целей, таких как проникновение на рынок, увеличение доли рынка, рост доходности и т.д.

Для получения количественной оценки экономической эффективности разработанного проекта – конкретных значений дополнительной прибыли предприятия и срока окупаемости инвестиций необходимо проведение специального комплексного исследования, которое выходит за рамки представленной работы.

6.4 Оценка научно-технического уровня НИР

В данном разделе проводится оценка научно-технического уровня разработки при помощи вычисления интегрального коэффициента научно-технического уровня $K_{НТУ}$. Научно-технический уровень характеризует, в какой мере выполнены работы и обеспечивается научно-технический прогресс в данной области.

Коэффициент научно-технического уровня проекта определяется по формуле:

$$K_{НТУ} = \sum_{i=1}^{3} R_i \cdot n_i$$

где $K_{НТУ}$ – коэффициент научно-технического уровня;

R_i – весовой коэффициент i-го признака научно-технического эффекта,

n_i – количественная оценка i-го признака научно-технического эффекта, в баллах.

Для определения коэффициентов следует воспользоваться таблицами, приведенными ниже.

Таблица 6.10 - Весовые коэффициенты признаков НТУ

Признак НТУ	**Характеристика признака**	$\boldsymbol{R_i}$
Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	0.4
Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	0.1
Возможность реализации	Время реализации в течение первых лет	0.5

Таблица 6.11 - Баллы для оценки уровня новизны

Уровень новизны	**Характеристика уровня новизны**	**Баллы**
Принципиально новая	Новое направление в науке и технике, новые факты и закономерности, новая теория, вещество, способ	8-10
Новая	По-новому объясняются те же факты, закономерности, новые понятия дополняют ранее полученные результаты	5-7
Относительно новая	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	2-4
Не обладает новизной	Результат, который ранее был известен	0

Таблица 6.12 - Баллы значимости теоретических уровней

Теоретический уровень полученных результатов	**Баллы**
Установка закона, разработка новой теории	10
Глубокая разработка проблемы, многоспектральный анализ, взаимодействия между факторами с наличием объяснений	8
Разработка способа (алгоритм, программа и т. д.)	6
Элементарный анализ связей между фактами (наличие гипотезы, объяснения версии, практических рекомендаций)	2
Описание отдельных элементарных факторов, изложение наблюдений, опыта, результатов измерений	0,5

Таблица 6.13 - Возможность реализации по времени

Время реализации	**Баллы**
В течение первых лет	10
От 5 до 10 лет	4
Свыше 10 лет	2

Определив количественные оценки для каждого признака можно вычислить показатель научно-технического уровня для данного проекта, который составил:

$$K_{НТУ} = 0,4 \cdot 4 + 0,1 \cdot 6 + 0,5 \cdot 9 = 6,7$$

Воспользовавшись таблицей ниже определим уровень НТЭ.

Таблица 6.14 - Оценка уровня научно-технического уровня

Уровень НТУ	**Показатель НТУ**
Низкий	1-4
Средний	4-7
Высокий	8-10

Таким образом, исходя из таблицы, воспользовавшись таблицей ниже определим уровень НТЭ. Данный проект имеет средний уровень научно-технического эффекта.

6.5 Выводы по разделу

В результате работы над данным разделом определены основные показатели эффективности научно-технического проекта: определен бюджет исследования, а также проведена оценка научно-технического уровня разработки. В результате данное исследование можно классифицировать как среднее, с точки зрения научно-технического уровня.

Анализ конкурентных технических решений показал, что предложенный метод имеет высокую конкурентоспособность за счет высокой точности распознавания, дешевизны и уровня проникновения на рынок.

Разграничение этапов работ позволило структурировать план работы над исследованием, определить ответственных за его этапы. На основе созданного перечня этапов и работ был создан календарный план-график, согласно которому общая длительность работ составляет 255 дней. Бюджет затрат на исследование составляет 398618,72 руб.

7 Социальная ответственность

7.1 Введение

Магистерская диссертация посвящена разработке методов и алгоритмов, входящих в программный комплекс, осуществляющий распознавание изображений и рукописных символов с использованием машинного обучения и построенного по архитектуре искусственных нейронных сетей. Применения такой технологии можно найти в системах видео и аудио фиксации, поиска и обработки нецифровой информации, а также для задач визуального анализа данных. Более конкретная сфера применения данной технологии — это оцифровка рукописных документов с целью внедрения электронного документооборота на различных предприятиях.

С точки зрения актуальности исследования в рамках социальной направленности результаты настоящего исследования позволят в значительной степени ускорить процесс оцифровки рукописных документов и снизить ошибки, вызванные проблемами ручного ввода оператором.

Данный раздел диссертации направлен на выявление опасных и вредных факторов, которые могут присутствовать при проектировании на рабочем месте оператора ПЭВМ. Рассмотрены меры по снижению и предупреждению вредных воздействий на окружающую среду, исследованы правовые и организационные вопросы обеспечения безопасности при чрезвычайных ситуациях.

7.2 Правовые и организационные вопросы обеспечения безопасности

Правовые вопросы организации труда работника, обеспечение его безопасности регулируется трудовым кодексом РФ, а также санитарными номами и инструкциями.

В соответствии со ст.100 Трудового кодекса РФ режим рабочего времени устанавливается правилами внутреннего трудового распорядка, которые, в свою очередь, утверждаются работодателем с учетом мнения представительного органа работников. В данном документе также регламентируется продолжительность рабочего времени, которая не должна быть меньше указанного времени в трудовом договоре, но, в свою очередь, не должна превышать 40 часов в неделю. Также законодательством предусмотрено установление в течение рабочего дня перерывов для питания и отдыха [32].

К актам, устанавливающим количество и продолжительность технологических перерывов, обязательным для исполнения, относятся санитарные нормы и правила [33], которыми установлены гигиенические требования к персональным электронно-вычислительным машинам (ПЭВМ) и организации работы. В них указывается, что в случаях, когда характер работы требует постоянного взаимодействия с видеодисплейными терминалами, рекомендуется организация перерывов на 10-15 мин. через каждые 45-60 мин. работы.

Согласно ФЗ «Об обязательном социальном страховании от несчастных случаев на производстве» установлены правовые основы обязательного социального страхования и определен порядок возмещения вреда, причиненного жизни и здоровью работника при исполнении им обязанностей по трудовому договору [34]. Обязательное социальное страхование предусматривает обеспечение социальной защиты застрахованных, в качестве которых выступают работники, возмещение вреда, причиненного жизни и здоровью застрахованного при исполнении им обязанностей, обеспечение предупредительных мер по сокращению производственного травматизма и профессиональных заболеваний. Средства на осуществление обязательного социального страхования формируются в основном за счет обязательных страховых взносов страхователей (работодателя).

В качестве поощрения работника за выполнение им трудовых обязанностей предусмотрена система оплаты труда, включающая размеры окладов, доплат и надбавок компенсационного характера и стимулирующего характера, системы премирования, установленные трудовыми договорами и соглашениями в соответствии с трудовым законодательством и иными нормативными правовыми актами.

7.2.1 Требования к организации рабочих мест пользователей

Рабочее место должно быть организовано с учетом эргономических требований согласно [35] и [36].

Конструкция рабочей мебели (рабочий стол, кресло, подставка для ног) должна обеспечивать возможность индивидуальной регулировки соответственно росту пользователя и создавать удобную позу для работы. Вокруг ПК должно быть обеспечено свободное пространство не менее 60-120см;

На уровне экрана должен быть установлен оригинал-держатель. На рисунке 7.1. схематично представлены требования к рабочему месту.

Рисунок 7.1 - Организация рабочего места

Работа программиста связана с постоянной работой за компьютером, следовательно, могут возникать проблемы, связанные со зрением. Также неправильная рабочая поза может оказывать негативное влияние на здоровье.

Таким образом, неправильная организация рабочего места может послужить причиной нарушения здоровья и появлением психологических расстройств.

Согласно [37]:

- яркость дисплея не должна быть слишком низкой или слишком высокой;
- размеры монитора и символов на дисплее должны быть оптимальными;
- цветовые параметры должны быть отрегулированы таким образом, чтобы не возникало утомления глаз и головной боли.
- опоры для рук не должны мешать работе на клавиатуре;
- верхний край монитора должен находиться на одном уровне с глазом, нижний – примерно на 20° ниже уровня глаза;
- дисплей должен находиться на расстоянии 45-60 см от глаз;
- локтевой сустав при работе с клавиатурой нужно держать под углом 90°;
- каждые 10 минут нужно отводить взгляд от дисплея примерно на 5-10 секунд;
- монитор должен иметь антибликовое покрытие;
- работа за компьютером не должна длиться более 6 часов, при этом необходимо каждые 2 часа делать перерывы по 15-20 минут;
- высота стола и рабочего кресла должны быть комфортными.

7.3 Производственная безопасность

Компьютерное моделирование и проектирование на ПЭВМ как правило производится в чистых офисных помещениях, компьютерных лабораториях, где присутствуют такие вредные производственные факторы как: повышенный уровень шума и температуры вследствие работы системных блоков ЭВМ и другой аппаратуры, отсутствие либо недостаток естественного и искусственного освещения, что может быть связано с неудачным

расположением окон помещения, а также недостаточная вентиляция. Кроме того, многие сотрудники подвержены воздействию таких психофизических факторов, как: умственное напряжение, перенапряжение зрительных и слуховых органов чувств, монотонность труда, эмоциональные перегрузки.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме. Возможные опасные и вредные факторы согласно ГОСТ 12.0.003-2015 приведены в таблице ниже.

Таблица 7.1 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003- 2015)	**Этапы работ**			**Нормативные документы**
	Разработка	**Изготовление**	**Эксплуатация**	
Опасные и вредные производственные факторы, обладающие свойствами психофизиологического воздействия	+	+	+	ГОСТ 12.0.002-2014 Система стандартов безопасности труда (ССБТ). Термины и определения
Опасные и вредные производственные факторы, связанные с аномальными микроклиматическими параметрами	+	+	+	СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений; ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
Опасные и вредные производственные факторы, связанные с повышенным уровнем характеристик шумового воздействия	+	+	+	ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности.

Опасные и вредные производственные факторы, связанные с электрическим током	+	+	+	ГОСТ 12.4.011-89 ССБТ «Средства защиты работающих. Классификация».
Опасные и вредные производственные факторы, связанные с электромагнитными полями	+	+	+	СанПиН 2.2.4.1340-03 «Гигиенические требования к персональным электронновычислительным машинам и организации работы»; СанПиН 2.2.4.3359-16 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах"; ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности.
Опасные и вредные производственные факторы, связанные со световой средой	+	+	+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95; СанПиН 2.2.1/2.1.1.1278–03. Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий.

7.3.1Анализ опасных и вредных производственных факторов Опасные и вредные производственные факторы, обладающие свойствами психофизиологического воздействия

Основную часть времени разработчик-программист проводит за работой на персональном компьютере. Для минимизации нагрузок психофизиологического характера на работника, особое внимание следует уделить требованиям организации рабочего пространства. Рабочее место должно обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования, например, компьютера или другого офисного

оборудования, с учетом его количества и конструктивных особенностей, характера выполняемой работы. Для снижения психофизиологических нагрузок на работника необходимо соблюдать требования к режиму труда и отдыха. В частности, соблюдение данных требований позволит минимизировать нервно-психические, нервно-эмоциональные перегрузки, а также утомление глаз, повышенную нагрузку на зрение.

Согласно СанПиН 2.2.2/2.4.1340-03 рекомендуется организовывать перерывы на 10-15 минут через каждые 45-60 минут работы. При этом продолжительность непрерывной работы с компьютером не должна превышать 2 часов. Во время перерывов следует выполнять комплекс упражнений для снятия утомления зрительного анализатора, напряжения в позвоночнике, а также общего эмоционального напряжения. Несоблюдение вышеуказанных правил может привести к получению работником травмы или развития заболевания.

7.3.1.1 Опасные и вредные производственные факторы, связанные с аномальными микроклиматическими параметрами

Производственный микроклимат, определяются состоянием температуры, влажности и движения воздуха производственных помещений, а также тепловым излучением от нагретого оборудования и обрабатываемых материалов [38]. В производственных помещениях с использованием ЭВМ, где работа связана с нервноэмоциональным напряжением обеспечивают оптимальные параметры микроклимата для категории работ 1а и 1б в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений.

Допустимые микроклиматические условия установлены по критериям

допустимого теплового и функционального состояния человека на период 8-часовой рабочей смены. Они не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и

локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности. Санитарные правила обязательны для соблюдения всеми государственными органами, предприятиями и другими организациями. Санитарные нормы устанавливают гигиенические требования к показателям микроклимата рабочих мест с учетом интенсивности энергозатрат работников. Допустимые величины показателей микроклимата на рабочих местах соответствуют значениям, приведенным в (Таблица 7.2) применительно к выполнению работ различных категорий в холодный и теплый периоды года. Холодным периодом года считается период года с среднесуточной температурой наружного воздуха равной +10 ^{o}C и ниже. Теплым периодом считается период года при среднесуточной температуре наружного воздуха выше +10 ^{o}C.

На основе интенсивности общих энергозатрат осуществляется разграничение работ по категориям СанПиН 2.2.4.548-96. Для производственных помещений с низким уровнем физической активности работников и энергозатратами организма человека до 120 ккал/ч или 139 Вт предусмотрена категория помещений Ia.

Таблица 7.2 - Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	**Категория работ по уровню энергозатрат, Вт**	**Температура воздуха, ^{o}C**	**Температура поверхностей, ^{o}C**	**Относительная влажность воздуха, %**	**Скорость движения воздуха, м/с**
Холодный	Ia (до 139 Вт)	22-24	21-25	40-60	0,1
Теплый	Ia (до 139 Вт)	23-25	22-26	40-60	0,1

В производственных помещениях, в которых допустимые нормативные величины показателей микроклимата невозможно установить, условия микроклимата рассматривают как вредные и опасные. В целях профилактики неблагоприятного воздействия микроклимата используют системы местного

кондиционирования воздуха, компенсацию неблагоприятного воздействия одного параметра микроклимата изменением другого, спецодежду и другие средства индивидуальной защиты, помещения для отдыха и обогревания, регламентацию времени работы, в частности, перерывы в работе, сокращение рабочего дня, увеличение продолжительности отпуска, уменьшение стажа работы и др.

7.3.1.2 Опасные и вредные производственные факторы, связанные с повышенным уровнем характеристик шумового воздействия

Повышенный шум влияет на нервную и сердечнососудистую системы, репродуктивную функцию человека, вызывает раздражение, нарушение сна, утомление, агрессивность, способствует психическим заболеваниям. При постоянном нахождение в помещение где уровень шума более 85 децибел, могут наблюдаться нарушения слуха.

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ЭВМ уровни шума на рабочих местах не превышают предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарноэпидемиологическими нормативами. Для лаборатории, в которой велась разработка, основными источниками шума являются расположенные в помещении компьютеры и кондиционер. В соответствии с нормами, регламентируемыми СН 2.2.4/2.1.8.562-96, предельно допустимый уровень звукового давления в лабораторных помещениях, где проводится проектирование, конструирование, научная деятельность не превышает 50 дБ.

Защита от шума регламентируется СНиП 11–12–77. К мероприятиям по защите от шума относятся: уменьшение шума в источнике возникновения; применение специальных глушителей, звукоизоляция помещений и оборудования, звукопоглощение, проведение предварительных и периодических осмотров рабочих мест и помещений.

7.3.1.3 Опасные и вредные производственные факторы, связанные с электрическим током

Электрическая безопасность включает в себя правовые, социальноэкономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия. Электрические установки, к которым относится практически все оборудование ЭВМ, представляют для человека потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ может произойти случайный контакт человека с электрически неизолированными либо незаземленными частями ЭВМ. Электрический ток оказывает на организм человека термическое, электролитическое и биологическое действия, приводящие к ожогам тела, нарушению физико-химического состава крови, а также к раздражению и возбуждению живых тканей организма, сопровождающиеся непроизвольными судорожными сокращениями мышц (сердца, легких).

Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ 12.1.019-2017 ССБТ [39].

Мероприятия, направленные на предотвращение возможности поражения электрическим током, включают в себя следующее:

- при выполнении монтажных работ необходимо использовать только исправно работающий инструмент, аттестованный службой КИПиА;
- запрет на выполнение работ на задней панели при включенном сетевом напряжении;
- выполнение работ по устранению неисправностей должно производиться компетентными людьми;

− нужно постоянно наблюдать за исправностью электропроводки и в случае обнаружения неисправностей незамедлительно принимать действия по их устранению.

7.3.1.4 Опасные и вредные производственные факторы, связанные с электромагнитными полями

Основным источником электромагнитных излучений является излучение от мониторов ЭВМ. Временные допустимые уровни электромагнитного излучения малы и отвечают требованиям СанПиН 2.2.1/2.1.1.1278-12. «Электромагнитные поля в производственных условиях» [40] которые приведены в (Таблица 7.3).

Таблица 7.3 - Временные допустимые уровни электромагнитных помех, создаваемых ЭВМ

Наименование параметра	**Диапазон частот, кГц**	**Временные допустимые уровни электромагнитных полей**
Напряженность электрического поля	В диапазоне частот 0,005 - 2	25 В/м
	В диапазоне частот 2 - 400	2,5 В/м
Плотность магнитного потока	В диапазоне частот 0,005 - 2	250 нТл
	В диапазоне частот 2 - 400	25 нТл
Напряженность электростатического поля		15 кВ/м

Мероприятия по снижению излучений включают:

− мероприятия по сертификации ЭВМ (ПК) и аттестации рабочих мест;

− применение экранов и фильтров;

− применение средств индивидуальной защиты путем экранирования пользователя ЭВМ (ПК) целиком или отдельных зон его тела;

− использование иных технических средств защиты от патогенных излучений.

7.3.1.5 Опасные и вредные производственные факторы, связанные со световой средой

В помещениях, где происходит работа за ЭВМ используется естественное и искусственное освещение. Естественное освещение предполагает проникновение внутрь зданий солнечного света через окна и различного типа светопроемы. Нормы естественного освещения для разных зданий и помещений разрабатываются с учетом их назначения. В помещениях общественных зданий, лабораторий применяют систему комбинированного освещения в помещениях общественных зданий, где выполняется напряженная зрительная работа. Общее освещение в лабораторных помещениях поддерживают равномерным.

Согласно санитарным нормам СНиП 23-05-95 [41] производственные помещения классифицируются на несколько категорий, для которых предусмотрены нормы освещенности. Лабораторное помещение для работы на ЭВМ относится к категории зрительной работы малой точности. Характеристики освещенности для данного класса помещений приведены в (Таблица 7.4).

Таблица 7.4 - Характеристики освещенности

Характеристика зрительной работы	**Наименьший размер объекта различения, мм**	**Разряд зрительной работы**	**Подразряд зрительной работы**	**Контраст объекта с фоном**	**Характеристика фона**	**Освещенность, лк**	
						Комбинированное освещение	Общее освещение
Малой точности	Св. 1 до 5	V	В	Малый Средний Большой	Светлый Средний Темный	-	200
			Г	Средний Большой	Светлый Средний	-	200

Безопасность и здоровье условия труда в большой степени зависят от освещенности рабочих мест и помещений. Неудовлетворительное освещение утомляет не только зрение, но и вызывает утомление организма в целом. Неправильное освещение может быть причиной травматизма: плохо освещенные опасные зоны, слепящие лампы, резкие тени ухудшают или вызывают полную потерю зрения, ориентации. Правильное освещение уменьшает количество несчастных случаев, повышает производительность труда.

Согласно требованиям СанПиН необходимо при проведении испытаний соблюдать определенные правила:

- применять комбинированную освещенность;
- естественный свет преимущественно должен падать слева;
- освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 – 500 лк;
- освещенность поверхности экрана не должна быть более 300 лк;
- в качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы либо компактные светодиодные лампы;

– для обеспечения нормируемых значений освещенности в помещениях для использования персональных электронно-вычислительных машин следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

Проведем расчет общего освещения в рабочем помещении, где установлено несколько рабочих компьютеров и присутствуют более 1 человека. Рабочим помещением служит учебный компьютерный класс в 407 ауд. кибернетического центра ТПУ. Данная аудитория имеет площадь 33,3 м^2 и вмещает в себя 10 рабочих мест. Расчет общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отраженный от потолка и стен. Световой поток лампы накаливания или группы люминесцентных ламп светильника определяется по формуле:

$$n = \frac{E_н \cdot S \cdot K_S \cdot z}{Ф_л \cdot \eta \cdot n}$$

где, $E_н$ – нормируемая минимальная освещенность по СНиП 23-05-95, 200 лк;

S – площадь освещаемого помещения – 33,3 м^2;

K_s – коэффициент запаса, учитывающий загрязнение светильника (источника света, светотехнической арматуры, стен и пр., т.е. отражающих поверхностей), пыли 1,4;

z – коэффициент неравномерности освещения, отношение $E_{ср}./E_{min}$. Для люминесцентных ламп при расчетах берется равным 1,1;

n – число ламп в светильнике;

η - коэффициент использования светового потока, %.

Коэффициент использования светового потока показывает, какая часть светового потока ламп попадает на рабочую поверхность. Он зависит от индекса помещения i, типа светильника, высоты светильников над рабочей поверхностью h и коэффициентов отражения стен ρ_c и потолка ρ_n.

Индекс помещения определяется по формуле

$$i = \frac{S}{h \cdot (A + B)}$$

где, h - допустимая высота подвеса светильников с люминесцентными лампами;

A – ширина помещения;

B – длина помещения.

Помещение имеет длину A= 6,4м, ширина B=5,2м, высота H= 2,7м. Высота рабочей поверхности $h_{рп}$ =0,7м. Требуется создать освещение E=200 лк.

Стены в рассматриваемом помещении окрашены светло-бежевой краской, потолок свежепобеленный, согласно СНиП 23-05-95 Естественное и искусственное освещение, коэффициенты отражения: стен ρ_c = 50%, потолка $\rho_п$ =70%. Коэффициент запаса $K_з$=1,4, коэффициент неравномерности Z = 1,1 согласно СНиП 23-05-95 Естественное и искусственное освещение.

Рассчитываем систему общего люминесцентного освещения. Выбираем светильники типа ЛВО 4×18.

Приняв h_c= 0 м, получаем h=2,7-0-0,7=2 м. – высота светильников над рабочей поверхностью.

Находим индекс помещения:

$$i = \frac{S}{h \cdot (A + B)} = \frac{33,3}{2 \cdot (6,4 + 5,2)} = 1,44$$

Для определения коэффициента использования светового потока используем табличные данные из [44], а также коэффициенты отражения стен и потолка согласно СНиП 23-05-95.

Коэффициент использования светового потока η=0,54.

Необходимое число светильников равно:

$$n = \frac{E_н \cdot S \cdot K_S \cdot z}{Ф_л \cdot \eta \cdot n} = \frac{200 \cdot 33,3 \cdot 1.4 \cdot 1.1}{1300 \cdot 0,54 \cdot 4} = 3,75 \sim 4$$

Размещаем светильники в два ряда. В каждом ряду можно установить 2 светильника типа ЛВО 4×18 мощностью 18 Вт. Учитывая, что в каждом

светильнике установлено 4 лампы, общее число ламп в помещении N=16. Размеры светильников ЛВО 4×18 составляют 595×595×85 мм.

Схема размещения светильников в компьютерной лаборатории для люминесцентных ламп представлена на рисунке ниже.

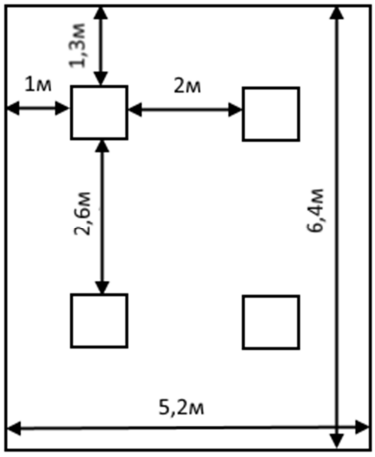


Рисунок 7.2 - Схема расположения светильников в компьютерной лаборатории

7.4 Экологическая безопасность

Работа за ПЭВМ не являются экологически опасными работами, потому объект, на котором производилась разработка продукта и объекты, на которых он будет использован операторами ПЭВМ относятся к предприятиям пятого класса, размер селитебной зоны для которых равен 50 м. Следовательно, создание санитарно-защитной зоны и принятие мер по защите атмосферы, гидросферы, литосферы не являются необходимыми.

Компьютер и другая оргтехника в своем составе содержит токсичные вещества. При завершении срока службы такого оборудования, его можно классифицировать, как отходы электронной промышленности. Утилизация, как электронно-вычислительных машин, так и другой оргтехники включает в себя работы по: погрузке, транспортировке, разгрузке, демонтажу и извлечению различных материалов из исписанных технических средств, а также сдачу на материалы специализированным организациям для дальнейшей переработки.

В нормативном документе СанПиН 2.2.1/2.1.1.1278-12, даются следующие общие рекомендации по снижению опасности для окружающей среды, исходящей от компьютерной техники:

- применять оборудование, соответствующее санитарным нормам и стандартам экологической безопасности;
- применять расходные материалы с высоким коэффициентом использования и возможностью их полной или частичной регенерации;
- отходы в виде компьютерного лома утилизировать;
- использовать экономные режимы работы оборудования.

Основной проблемой охраны окружающей среды в компьютерных лабораториях является утилизация люминесцентных ламп. Все люминесцентные лампы содержат ртуть (в дозах от 1 до 70 мг), ядовитое вещество 1-го класса опасности. Такая доза может причинить вред здоровью при повреждении лампы. Хранение и удаление отходов (в данном случае - люминесцентных ламп) осуществляются в соответствии с требованиями экологической безопасности согласно СанПиНу 2.2.7.029-99. Наполненную тару с отходами закрывают герметически стальной крышкой, при необходимости заваривают и передают по договору специализированным предприятиям, имеющим лицензию на их утилизацию.

7.5 Безопасность в чрезвычайных ситуациях

Чрезвычайная ситуация — это состояние, при котором в результате возникновения источника ЧС на объекте нарушаются нормальные условия жизни и деятельности людей, возникает угроза их жизни и здоровью, наносится ущерб имуществу населения, народному хозяйству и природной среде.

Наиболее распространенными источниками возникновения чрезвычайных ситуаций техногенного характера в лабораторных помещениях являются пожары. Пожары на предприятиях могут возникать в результате повреждения электропроводки и электрооборудования, находящегося под

напряжением, нарушение правил эксплуатации электрического оборудования, эксплуатация его в неисправном состоянии, перегрузка электрических сетей, применение неисправных осветительных приборов.

В число превентивных мероприятий могут быть включены мероприятия, направленные на устранение причин, которые могут вызвать пожар. Здание лаборатории оснащаются системами автоматической пожарной защиты. Они быстро обнаруживают очаг загорания, автоматически отключают электропитание ЭВМ, локализируют и тушат пожар. В помещении должен быть установлен углекислотный огнетушитель типа ОУ-5 для тушения пожаров. В случае угрозы возникновения ЧС необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения согласно плану эвакуации. Каждый сотрудник обязан соблюдать меры пожарной безопасности на предприятии и следить за их соблюдением другими.

В целях профилактики пожара предлагается не использовать открытые обогревательные приборы в помещении лаборатории. В целях уменьшения вероятности возникновения пожара вследствие короткого замыкания необходимо, чтобы электропроводка была скрытой. Еще одним фактором возникновения пожара может стать курение в помещении. Поэтому курение в помещении лаборатории категорически запрещено.

7.6 Выводы по разделу

В данном разделе проведен анализ вредных и опасных факторов, возникающих на предприятиях при работе на ПЭВМ. Предложены методы минимизации и устранения последствий воздействия таких факторов на организм человека. Подобные меры регламентируются законодательством РФ и носят, как рекомендательный, так и обязательный характер. Исполнение изученных норм и правил призвано обеспечить безопасность и ресурсоэффективность трудовой деятельности.

Заключение

В результате проделанной работы было проведено исследование принципа работы сверточной нейронной сети и ее прикладное применения для распознавания образов, в частности – рукописных цифр при помощи современных методов машинного обучения. Разработано консольное приложение на языке программирования Python для работы с данными из открытой базы MNIST.

Архитектура сети для нейронной сети будет следующая:

- Чередующиеся слои свертки и подвыборки;
- Полносвязные слои для классификации;
- Техника с борьбы с переобучением(Dropout).

Разработанное приложение впоследствии может послужить отправной точкой для разработки ядра более мощной программы распознавания рукописного ввода, а именно связки цифр, символов (номера автомобилей, почтовые индексы и т.д.).

Список публикаций

1. А.С. Вторушина, И.А. Ботыгин Распознавание рукописных цифр на изображениях с использованием инструментов машинного обучения / Молодежь и современные информационные технологии: сборник трудов XVII Международной научно-практической конференции студентов, аспирантов и молодых ученых (Томск, 17–20 февраля 2020 г.) / Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2020. – 458 с.

Список используемой литературы

1. Маринчук А. С., Баженов Р. И. Распознавание цифр на основе нейронных сетей в octave.: Постулат. - № 12-1 (38). – 2018. -102 с.
2. Вальке А. А., Лобов Д. Г. Алгоритмы распознавания символов.: Динамика систем, механизмов и машин. – Т. 6, № 4. – 2018. - 164 – 168 с.
3. Глубокое обучение на Python. – СПб.: Питер, 2018. – 400с.: Ил. – (серия «Библиотека программиста»). ISBN 978-5-4461-0770-4.
4. Исрафилов Х. С. Применение нейронных сетей в распознавании рукописного текста // Молодой ученый. — 2016. — №29. — С. 24-27. — URL https://moluch.ru/archive/133/37372/ (дата обращения: 15.03.2020).
5. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis / Patrice Y. Simard, Dave Steinkraus, John C. Platt // Seventh International Conference on Document Analysis and Recognition. -2003. Proceedings.
6. Understanding Batch Normalization/ Johan Bjorck, Carla Gomes, Bart Selman, Kilian Q. Weinberger// 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
7. Бураков, М.В. Нейронные сети и нейроконтроллеры: учебное пособие / М. В. Бураков. - СПб.: ГУАП, 2013. - 284 с.
8. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks [Электронный ресурс]. – URL: https://arxiv.org/abs/1312.6082. (Дата обращения 20.04.2019).
9. Krizhevsky, A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // Advances in neural information processing systems. – 2012. – P. 1097–1105.
10. Градиентный спуск [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Градиентный_спуск. (Дата обращения 15.03.2020).

11. Хайкин С. Нейронные сети : полный курс : пер. с англ. / С. Хайкин. – 2- е изд., испр.. – М. [и др.]: Вильямс, 2006. – 1103 с.
12. LeCun, Y. Efficient BackProp in Neural Networks: Tricks of the trade / Y.LeCun, L. Bottou, G. Orr, K. Muller – Springer, 1998
13. Классификатор изображений на основе сверточной сети [Электронный ресурс] URL: http://mechanoid.kiev.ua/ml-lenet.html
14. Python/Учебник Python 2.6. [Электронный ресурс]. – Режим доступа: https://ru.wikibooks.org/wiki/Python/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_Python_2.6. Дата обращения: 9 февраля 2020.
15. TensorFlow. Информационный портал.[Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/
16. TensorFlow.Guide Обучающие материалы [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/guide/keras. Дата обращения: 28 февраля 2020.
17. PyProg. Справочная информация по NumPy. [Электронный ресурс]. – Режим доступа: https://pyprog.pro/introduction.html.
18. Лучшие IDE и редакторы кода для Python. [Электронный ресурс]. – Режим доступа: https://tproger.ru/translations/python-ide/. Дата обращения: 3 февраля 2020.
19. 10 причин, почему мы перешли на PyCharm. [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/122018/. Дата обращения: 1 ферваля 2020.
20. Welcome to colaboratory. Облачный сервис. [Электронный ресурс]. – Режим доступа: https://colab.research.google.com.
21. Справочная информация по Google Colaboratory. [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/mastering-the-features-of-google-colaboratory-92850e75701.
22. Справочник по работе с Google Colaboratory. [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/how-to-use-google-colab.

23. Руководство Azure для разработчиков Python. [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/azure/python.
24. Microsoft Azure. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft_Azure.
25. Единая облачная PaaS-платформа для ASP.NET, PHP, Node.js и Python. [Электронный ресурс].
26. Install the Azure Machine Learning SDK for Python. [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/en-us/python/api/overview/azure/ml/install?view=azure-ml-py.
27. Настройка среды разработки для Машинного обучения Azure. [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/azure/machine-learning/service/how-to-configure-environment#workspace.
28. Amazon Web Services. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Amazon_Web_Services.
29. AWS Deep Learning AMIs developer resources. [Электронный ресурс].
30. Типы инстансов Amazon EC2. [Электронный ресурс]. – Режим доступа: https://aws.amazon.com/ru/ec2/instance-types/#Memory_Optimized.
31. Инстансы Amazon EC1 Inf1. [Электронный ресурс]. – Режим доступа: https://aws.amazon.com/ru/ec2/instance-types/inf1.
32. «Трудовой кодекс Российской Федерации» от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020)
33. СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
34. Федеральный закон от 24.07.1998 N 125-ФЗ (ред. от 01.04.2020) «Об обязательном социальном страховании от несчастных случаев на производстве и профессиональных заболеваний».
35. СанПин 52.13330.2011 «Естественное и искусственное освещение. Актуализированная редакция СанПин 23-05-95*».
36. ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования».

37. ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».
38. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
39. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений.
40. ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.
41. СанПиН 2.2.1/2.1.1.1278-12. «Электромагнитные поля в производственных условиях»
42. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95.
43. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.
44. Справочная книга по светотехнике/Под ред. Ю.Б. Айзенберга.-М.: Энергоиздат, 1983-472 с.

Приложение А

(справочное)

Chapter 4. Experimental part

Студент:

Группа	ФИО	Подпись	Дата
8ВМ82	Вторушина Анна Сергеевна		

Консультант школы отделения (НОЦ) (аббревиатура школы, отделения (НОЦ)) :

Должность	ФИО	Ученая степень, звание	Подпись	Дата

Консультант – лингвист отделения (НОЦ) школы (аббревиатура отделения (НОЦ) школы):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ ШБИП	Аксёнова Наталия Валерьевна	К.филол.н		

4 Data preprocessing

Before sending the network data, it must be converted to floating point tensors. Since the data are presented as JPEG files, the files must be prepared for transferring to the neural network. To do this, the following steps must be completed:

- Read files containing images;
- Decode the contents of files from JPEG format to RGB pixel tables;
- Convert the resulting tables into arrays (tensors) of real numbers;
- Convert the scale of pixel values from the range [0; 255] to the range [0,1].

Neural networks prefer transmitting small values. As part of the design Python programming language and framework Keras have been chosen. Keras has already got utilities that will automate all these four points. Keras.preprocessing.image easily works with images. ImageDataGenerator type, allows you to quickly set up generators for automatic conversion of image files into ready-made tensor packages.

As an experiment, we consider a convolutional neural network with generators, a neural network with data expansion and a pre-trained neural network.

4.1 Python generators

A Python generator is an object that acts as an iterator. Let us look at the output of generators.

Let us fit the model to the data using the generator. The neural network will return 28x28 image packages. Each packet has 20 samples. Input data to the model will be transmitted using the generator, the fit_generator method (the equivalent of the fit method). With the help of the generator, data will be generated infinitely, so it is necessary to define how many samples will have to be extracted. In this case, as was previously described, the packets contain 20 samples, therefore, to obtain 2000

samples we need to extract 100 packets. Let us create a change of graphics and accuracy loss model for training and verification data in the learning process.

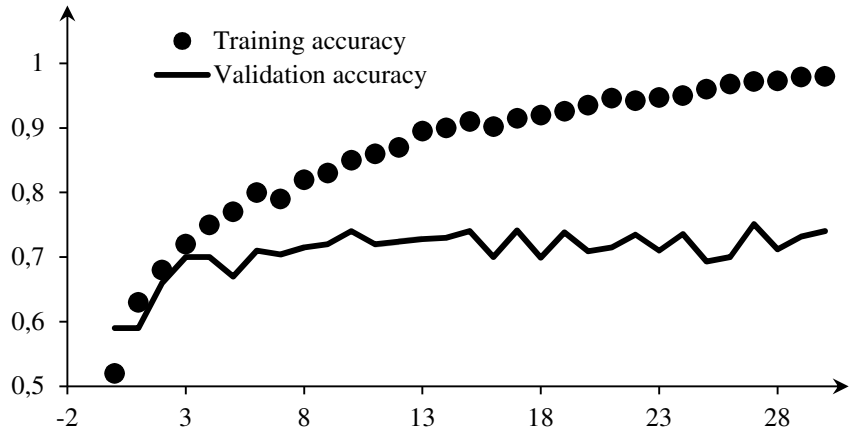


Figure 4.1 – Accuracy in the testing and training phases

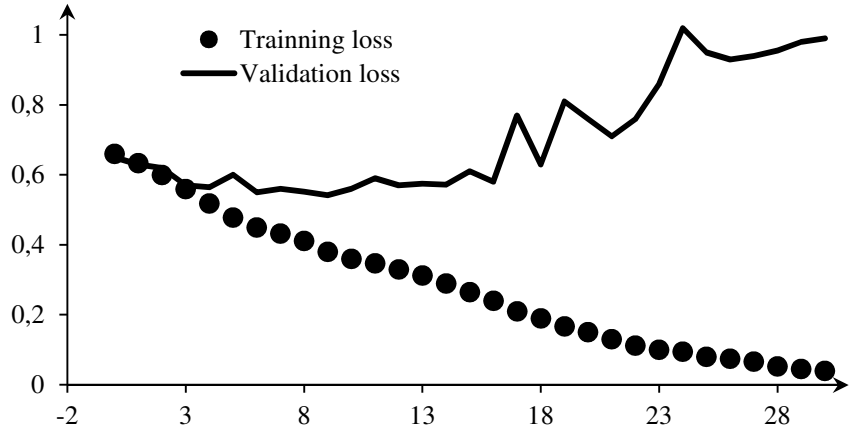


Figure 4.2 – Losses at the validation and training stages

In the graphs, overfitting effect is clearly observed. The accuracy on training data grows linearly and approaches 100%, while the accuracy on validation data stops at 70-72%. Losses at the validation stage reach a minimum after only five epochs and then stop, while losses at the training stage continue to decrease linearly, almost reaching 0 as there are relatively few training samples in this comparative experiment. Therefore, the problem of overfitting is becoming the main problem.

4.2 Data augmentation

Insufficient data for model training can be identified as one of the reasons for overfitting. With an infinite amount of data we would be able to obtain a model that will take into account all distribution of data aspects: the effect of overfitting would never have occurred. Data augmentation implements the approach of creating additional training data from the available by transforming samples with a set of random transformations that give plausible images. The target is that at training time, your model will never see the exact same picture twice. This helps expose the model to more aspects of the data and generalize better. Even if you train a new network using these extension settings, the input data will still be closely related. The cause is they are derived from a small number of original images.

We will use the data extension and thinning to train the network. We display graphs with the results of a trained neural network.

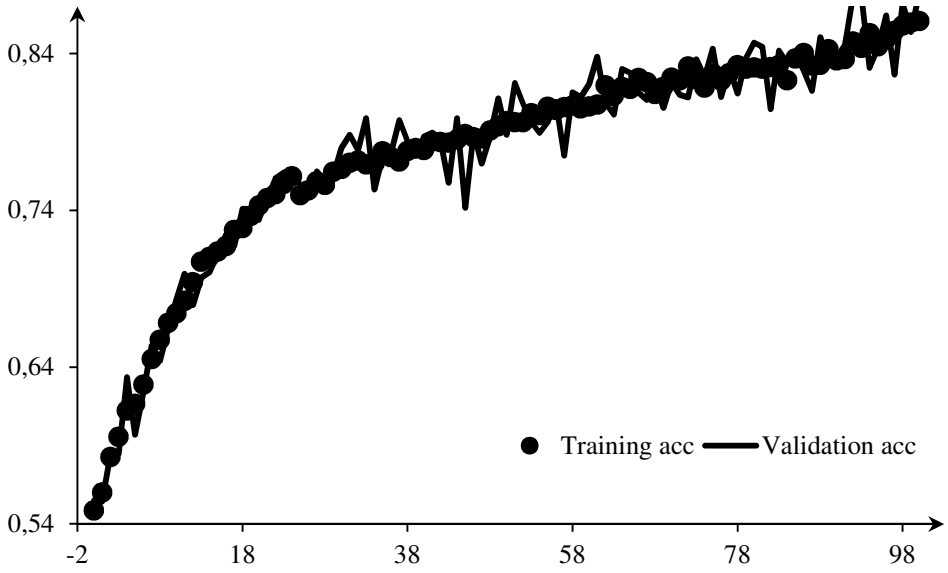


Figure 4.3 – Accuracy in the testing and training phases

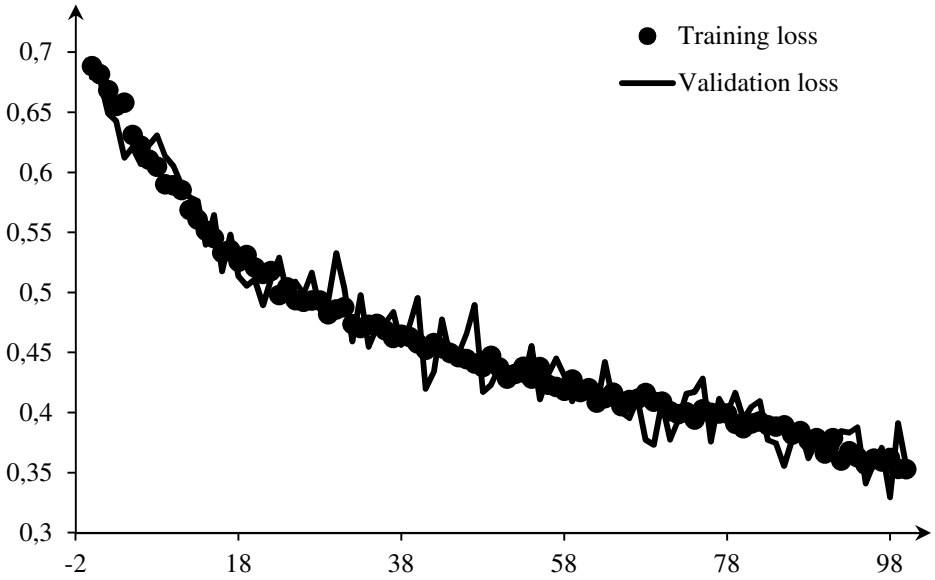


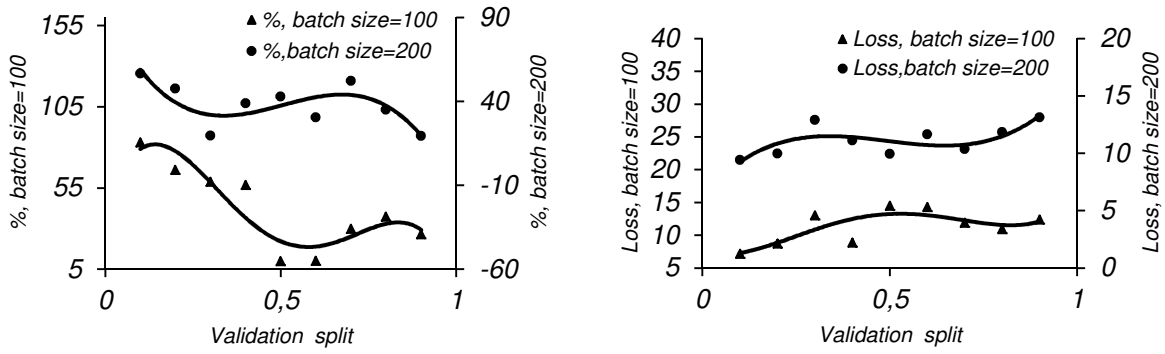
Figure 4.4 – Losses at the validation and training stages

Thanks to data augmentation and dropout, we are no longer overfitting: the training curves are closely tracking the validation curves. Now we reach an accuracy of 82%, a 15% relative improvement over the non-regularized model.

A third algorithm will be used for increasing accuracy: a pre-trained model.

4.3 A pretrained network

A typical and effective approach to learning on small sets of images is using a pre-trained network. A pre-trained network is a stored network previously trained on a large data set, usually as a part of a large-scale image classification task. If this initial dataset is large enough and sufficiently generalized, then the spatial hierarchy of features studied by the network can effectively act as a generalized model of the visible world and be useful in many different image recognition tasks, even if these new tasks are associated with completely different classes from those in the original task. We will draw a change of graphics and accuracy loss model for training and verification data in the learning process.



a) b)

Figure 4.5 – Graphics

To achieve accuracy of 100%, we will use a pre-trained neural network to recognize handwritten numbers.

4.4 Software architecture

The program consists of the following modules:

- **mnist_nw.py**. This is a file with a pre-trained CNN;
- **mnist_model.json.** This file contains the entire structure of the neural network;
- **mnist_model.h5.** This file contains data on the weights of neurons. The file is created after the program runs;
- **mnist.py.** The main program file.

The interaction of the modules with each other is presented in the block diagram below:

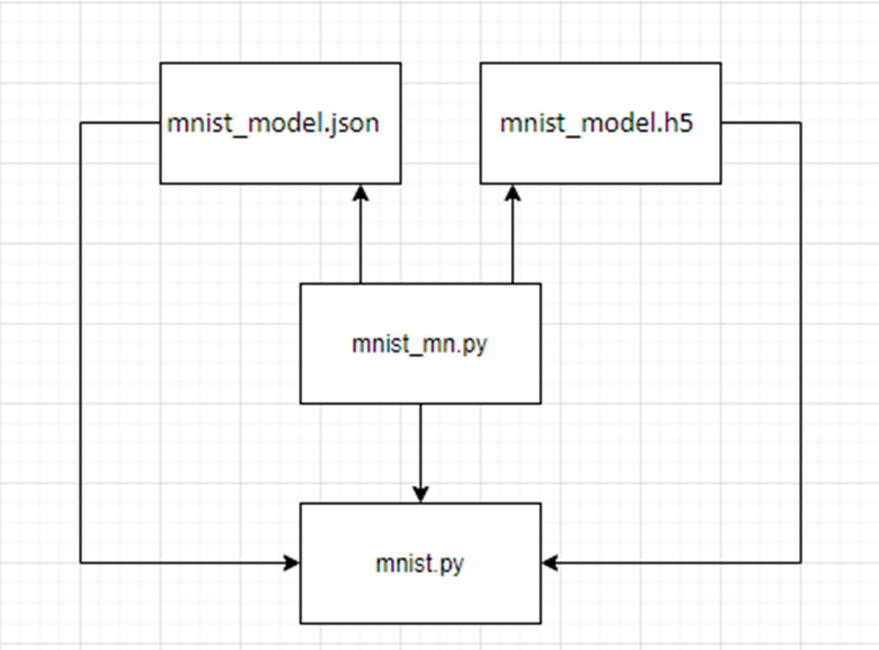


Figure 4.6 – Interaction of modules in the program

Description of methods used in files:

numpy.random.seed. The number of repeatable results.

mnist.load_data(). Loading mnist data. This function will contain a set of training that will be needed for training.

Reshape is a method for transforming the resulting images, which allows us to change the shape of the tensor. The method converts data into a three-dimensional array, the values of which are numbers in the interval [0,255], which will then be converted to another data type in the interval [0,1].

To_categorical. Direct encoding for category formatting.

Conv2D. Determines the size of templates extracted from the input data and the depth of the feature map output.

MaxPooling2D is specifying a subsample layer. Selects the maximum value from neighboring.

Dropout is a layer of regularization, which allows neural networks to exclude overtraining

Flatten() is a data conversion layer that integrates all tensors.

Dense() is a sequence of two layers, which are closely related neural layers. The first layer uses the relu activation function; the second (and also the last) layer is

a 10 variable loss layer (softmax) that returns an array with 10 probability estimates. Each estimate determines the probability of belonging to one of the 10 number classes. The argument passed to each layer is the number of hidden neurons in the layer.

Batch size. Package or mini-package is a small set of samples processed by the model simultaneously. During training, one mini-package is used in the gradient descent to calculate one change in the model's weights.

Fit. A method that tries to adapt the model to the training data and starts to look through the training data in mini packs of 200 samples.

Categorical crossentropy is a loss function that is used as a feedback signal for weight tensor training, and which the training phase tries to minimize. The exact rules governing a specific gradient descent application are determined by the adam optimizer, which is passed in the second argument and performs 10 iterations.

For each mini packet, the network calculates the gradients of weights taking into account losses in the packet and changes the weights in the appropriate direction.

Prediction. The result of the model operation.

4.5 Results

The purpose of the experimental part of the work is to ensure maximum accuracy of recognition by the neural network of handwritten numeric characters in the range from 0 to 9. During the operation of the neural network algorithm, the load on the hardware of the computer should be adequate and uniform. The hardware of the machine, in this case, should include the CPU and GPU.

In the course of the experiment, the main tasks were to study the influence on the accuracy of the output data of such parameters as:

- The number of layers of the CNN;
- Number of training cycles (Epoch);
- sets the epoch’s number (Batch size);

– The ratio of training and validation objects of the training set (Validation split).

Overfitting occurs when a too long training, insufficient numbers of training examples or an overcomplicated structure of the neural network are found. The first stage of the experiment is to determine the dependence of the learning error on the "Validation split" parameter. One of the options to deal with network overfitting is to vary this parameter. It is also responsible for dividing the training set into two sets - training and validation.

The number of epochs and network layers should be chosen as minimal to exclude the influence of this parameter on the error function. The number of sets of the epoch should be selected based on the capabilities of the hardware of the computer. For example, when 2 parameters "Batch size": 100 and 200 are selected. The range of variation of the "Validation split" parameter is accepted in the range from 0.1 to 0.9.

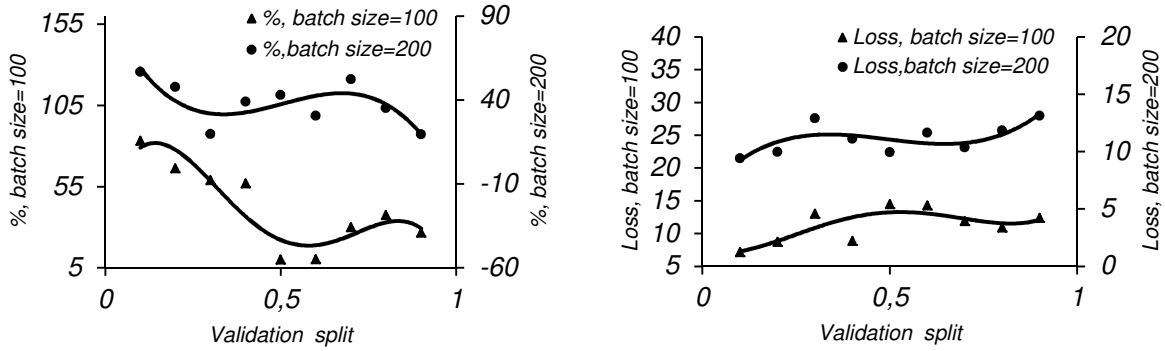
Table 4.1 – Validation split results for single-layer network

Epoch	**batch_size**	**val_sp**	**time**	**loss**	**acc**	**val_loss**	**val_acc**	**Acc_CNN, %**
1	100	0,1	207	7,1487	0,5522	2,615	0,8355	83,26
	100	0,2	160	8,7339	0,4549	5,4367	0,6607	66,26
	100	0,3	174	13,0207	0,1919	12,9488	0,5645	58,98
	100	0,4	152	8,8789	0,4462	6,9959	0,5645	56,98
	100	0,5	151	14,4622	0,1025	14,4751	0,1019	10,1
	100	0,6	118	14,3056	0,1119	14,5014	0,1003	10,32
	100	0,7	109	11,8767	0,2621	11,384	0,2933	29,94
	100	0,8	80	10,8949	0,3209	10,0635	0,3746	37,39
	100	0,9	61	12,397	0,2284	11,8742	0,2622	26,71

Table 4.2 - Validation split results for single-layer network

Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	Accuracy, %
1	100	0,1	207	7,1487	0,5522	2,615	0,8355	83,26
	100	0,2	160	8,7339	0,4549	5,4367	0,6607	66,26
	100	0,3	174	13,0207	0,1919	12,9488	0,5645	58,98
	100	0,4	152	8,8789	0,4462	6,9959	0,5645	56,98
	100	0,5	151	14,4622	0,1025	14,4751	0,1019	10,1
	100	0,6	118	14,3056	0,1119	14,5014	0,1003	10,32
	100	0,7	109	11,8767	0,2621	11,384	0,2933	29,94
	100	0,8	80	10,8949	0,3209	10,0635	0,3746	37,39
	100	0,9	61	12,397	0,2284	11,8742	0,2622	26,71

The results of the tables clearly demonstrate graphical dependencies.



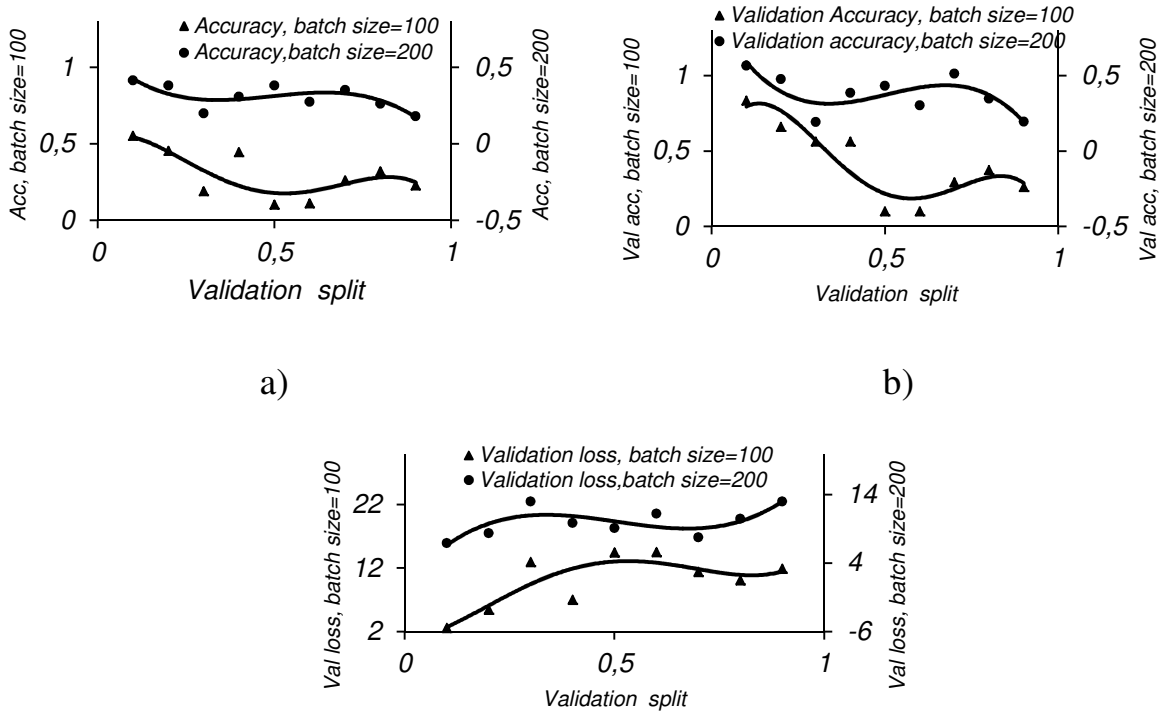
a) b)

Figure 4.7 – a) - Accuracy CNN; b) - Training loss

With different numbers of sets in the epoch, we can notice a decrease in the accuracy of the output data when Validation split tends to one. It shows that the more training objects of the total set allocated to network training, the higher the accuracy of training is. The error function behaves somewhat differently.

For small values of the Batch size, there is no clear correlation and trend, while with an increase in Batch size, an increase in the accuracy of the output data is noticeable when the ratio of training and validation sets is 50/50.

With other ratios, the accuracy of training tends to decrease.



a) b)

c)

Figure 4.8 – a) - Training acc; b) - Validation acc; c) - Validation loss

The constructed neural network based on one convolutional layer shows insufficient accuracy of the output data with the optimal number of sets of the same epoch in the entire range of Validation split. It is possible to increase the accuracy of the algorithm by increasing the layers of the neural network.

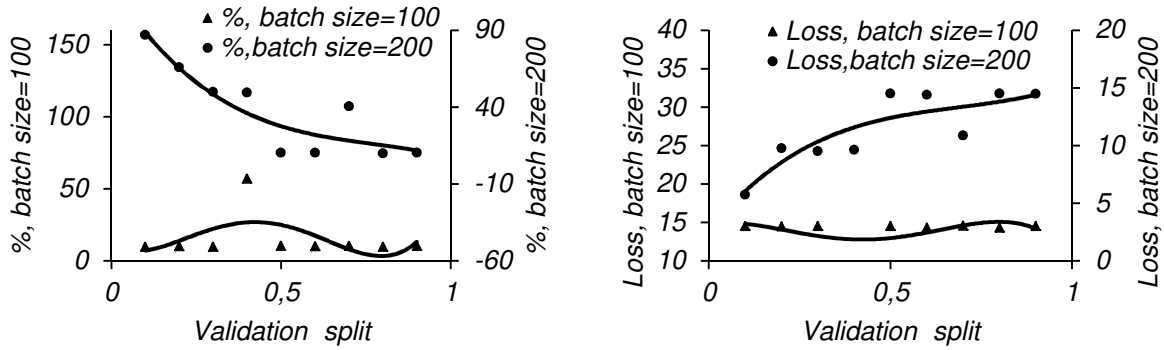
It is enough for comparison to implement one more layer into the network and select the optimal network parameters.

Table 4.3 - Validation split results of a two-layer network

Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	Accuracy, %
1	100	0,1	535	14,5482	0,0972	14,5197	0,0992	9,82
	100	0,2	418	14,4952	0,1005	14,4499	0,1035	10,1
	100	0,3	388	14,5323	0,092	14,5573	0,0968	9,82
	100	0,4	352	9,6922	0,395	6,9544	0,5672	56,96
	100	0,5	314	14,5329	0,0982	14,5009	0,1003	10,32
	100	0,6	272	14,3135	0,1108	14,534	0,0983	10,09
	100	0,7	236	14,5619	0,0964	14,4994	0,1004	10,32
	100	0,8	197	14,3308	0,1093	14,5318	0,0984	9,74
	100	0,9	160	14,5277	0,0975	14,5132	0,0996	10,32

Table 4.4 - Validation split results of a two-layer network

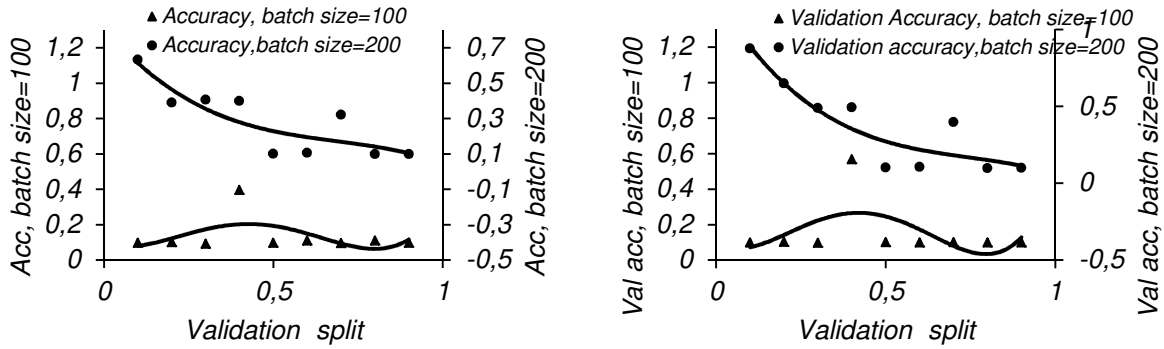
Epoch	batch_size	val_sp	time	loss	acc	val_loss	val_acc	Accuracy, %
1	200	0,1	445	5,7497	0,6329	1,8936	0,8757	86,83
	200	0,2	412	9,7557	0,3898	5,5786	0,6488	65,88
	200	0,3	358	9,5275	0,4057	8,2191	0,4893	49,89
	200	0,4	338	9,6307	0,3979	8,1579	0,4925	49,44
	200	0,5	298	14,5042	0,0995	14,5009	0,1003	10,32
	200	0,6	269	14,4049	0,1052	14,4275	0,1049	10,28
	200	0,7	231	10,8672	0,3217	9,7025	0,3973	40,54
	200	0,8	193	14,5131	0,0986	14,554	0,097	9,82
	200	0,9	163	14,483	0,0988	14,5132	0,0996	10,32



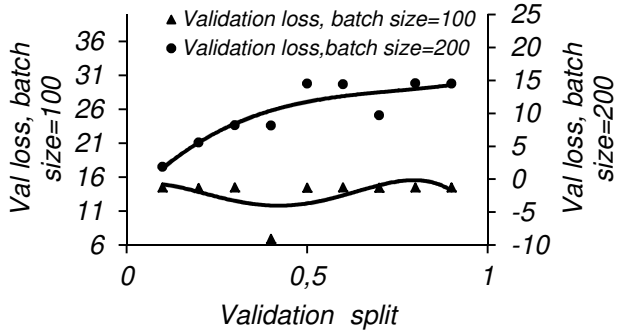
a) b)

Figure 4.9 – a) - Training acc; b) - Validation acc

By analogy with the previous single-layer network, a neural network with a large number of Batch size has a more pronounced decrease in accuracy.



a) b)



c)

Figure 4.10 – a) - Training acc; b) - Validation acc c) - Validation loss

In comparison with a single-layer neural network, the most pronounced trends of curves are observed in a two-layer neural network. At the same time, the same quality indicators of both networks correlate with each other.

For example, the index precision network comprising a convolutional layer tends to decrease qualitative characteristics substantially independently of the number of sets in a single epoch. The same accuracy indicator has a more pronounced tendency to decline in a neural network with two layers. The greatest steepness of the curve has a network configuration with a large number of sets.

The value of Batch size greatly affects network performance and local machine load. The configuration with Batch size = 200 is quite acceptable for this machine, however, such a neural network does not have sufficient accuracy of the output data. Therefore, it is necessary to increase the number of epochs up to ten, fixing at the same time the Batch size value of two hundred objects.

Table 4.5 - Two-layer network summary table

Epoch	batch_size	Valid_split	time	loss	acc	val_loss	val_acc	Accuracy, %
10	200	0,1	446,4	2,8768	0,81942	1,92426	0,87955	87,38
	200	0,2	332,1	0,05536	0,98267	0,03438	0,98985	99,17
	200	0,3	342,3	3,50257	0,78082	2,7259	0,8328	95,66
	200	0,4	333,7	0,7712	0,93217	0,13884	0,97466	99,08
	200	0,5	218,8	14,52574	0,09878	14,51122	0,09967	10,09
	200	0,6	210,8	5,06448	0,68337	4,37316	0,72703	75,84
	200	0,7	174,3	8,05972	0,49921	7,33944	0,54388	64,03
	200	0,8	154,2	14,52728	0,09863	14,554	0,097	9,82
	200	0,9	126	12,45534	0,22568	12,03594	0,25205	27,88

As it can be noticed from the previous tables, the network with the maximum number convolutional layers has the greatest accuracy.

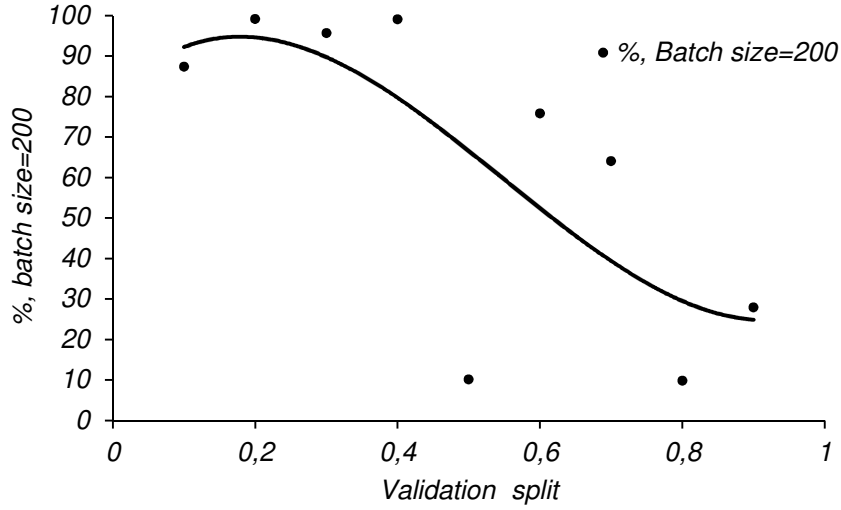


Figure 4.11 – Final graph with neural network accuracy

Depending on the ratio of the training and test sets, the accuracy of the output data tends to fall predominantly. However, maximum accuracy is achieved with a ratio of 20% - validation data and 80% - training.

As a result, the structure of the final neural network contains:

- Input layer of 784 neurons;
- Hidden layer of 100 neurons;
- Second hidden layer consisting of 75 neurons;
- And an output layer, consisting of 10 neurons. The numbers are within the range of 0 to 9.

Input and output data have not changed due to the fact that a 28x28 image is standardly fed to the input, which will be equal to 784 neurons. The output will be standard - these are numbers from zero to nine.

This neural network with its structure will be used in further experiments with optimization of hardware resources.

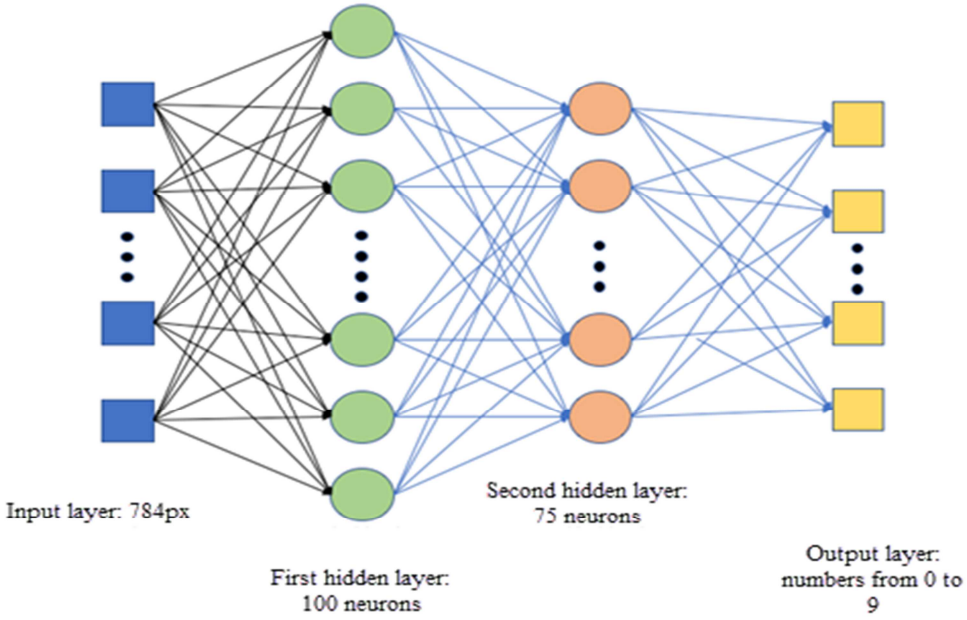


Figure 4.12 – Neural network structure

4.6 Conclusion

As a result of the work done, a study was carried out on the principle of the convolutional neural network and its application for pattern recognition, in particular handwritten numbers using modern machine learning methods. A console application was developed in the Python programming language for working with data from the open MNIST database.

The network architecture for the neural network will be as follows:

- Alternating convolution and subsampling layers;
- Fully connected layers for classification;
- Technology with the fight against overfitting (Dropout).

A technology to deal with an overfitting the developed application can later serve as a starting point for developing the core of a more powerful handwriting recognition program, namely a bunch of numbers, symbols (car numbers, postal codes, etc.).

Приложение Б

(обязательное)

Итоговая программа

mn.py

```
import numpy as np
from keras.preprocessing import image
from keras.models import model_from_json
from PIL import Image
img_path = '4.png'
img = image.load_img(img_path, target_size=(28, 28), color_mode="grayscale")
x = image.img_to_array(img)
x = 255 - x
x /= 255
x = np.expand_dims(x, axis=0)
json_file = open("mnist_model.json", "r")  #для мнист
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
loaded_model.load_weights("mnist_model.h5")
loaded_model.compile(loss="categorical_crossentropy", optimizer="adam",
metrics=["accuracy"])
prediction = loaded_model.predict(x)
print(np.argmax(prediction))
```

Mnist.py

```
import numpy
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.utils import np_utils
numpy.random.seed(42)
img_rows, img_cols = 28, 28
(X_train, y_train), (X_test, y_test) = mnist.load_data()
X_train = X_train.reshape(X_train.shape[0], img_rows, img_cols, 1)
X_test = X_test.reshape(X_test.shape[0], img_rows, img_cols, 1)
input_shape = (img_rows, img_cols, 1)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
Y_train = np_utils.to_categorical(y_train, 10)
Y_test = np_utils.to_categorical(y_test, 10)
model = Sequential()
model.add(Conv2D(75, kernel_size=(5, 5), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(100, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(500, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
print(model.summary())
model.fit(X_train, Y_train, batch_size=200, epochs=10, validation_split=0.2, verbose=2)
scores = model.evaluate(X_test, Y_test, verbose=0)
print("Точность работы на тестовых данных: %.2f%%" % (scores[1]*100))
model_json = model.to_json()
json_file = open("mnist_model.json", "w")
json_file.write(model_json)
json_file.close()
model.save_weights("mnist_model.h5")
```

Data preprocessing

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(train_dir,target_size=(150,
150),batch_size=20,class_mode='binary')
validation_generator = test_datagen.flow_from_directory(validation_dir,target_size=(150,
150), batch_size=20, class_mode='binary')
for data_batch, labels_batch in train_generator: print('data batch shape:',
data_batch.shape)
    print('labels batch shape:', labels_batch.shape)
    break
history = model.fit_generator(train_generator, steps_per_epoch=100, epochs=30,
validation_data=validation_generator, validation_steps=50)
model.save('cats_and_dogs_small_1.h5')
import matplotlib.pyplot as plt
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(acc))
```