

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 01.02.03 Прикладная математика и информатика
 Отделение школы (НОЦ) Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

| Тема работы |
|---|
| Исследование методов синтеза входной информации и реализация генеративно-состязательных сетей для расширения обучающей выборки |

УДК 004.65:004.7.032.26

Студент

| Группа | ФИО | Подпись | Дата |
|--------|---------------------------|---------|------|
| 8Б71 | Селезнёв Аркадий Олегович | | |

Руководитель ВКР

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---------------------|-------------------------|------------------------|---------|------|
| Профессор ОИТ ИШИТР | Гергет Ольга Михайловна | д.т.н. | | |

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|-------------------------------|------------------------|---------|------|
| Доцент ОСГН ШБИП | Трубченко Татьяна Григорьевна | к.э.н. | | |

По разделу «Социальная ответственность»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---------------|-----------------------------|------------------------|---------|------|
| Ассистент ООД | Черемискина Мария Сергеевна | - | | |

ДОПУСТИТЬ К ЗАЩИТЕ:

| Руководитель ООП | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------|---------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Шевелев Геннадий Ефимович | к.ф. – м.н. | | |

Планируемые результаты освоения ООП по направлению 01.03.02

«Прикладная математика и информатика»

| Код компетенции | Наименование компетенции |
|---|--|
| Универсальные компетенции | |
| УК(У)-1 | Способность осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач |
| УК(У)-2 | Способность определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений |
| УК(У)-3 | Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде |
| УК(У)-4 | Способность осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах) |
| УК(У)-5 | Способность воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах |
| УК(У)-6 | Способность управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни |
| УК(У)-7 | Способность поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности |
| УК(У)-8 | Способность создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций |
| УК(У)-9 | Способность проявлять предприимчивость в профессиональной деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи |
| Общепрофессиональные компетенции | |
| ОПК(У)-1 | Способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой |
| ОПК(У)-2 | Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию образовательного контента, прикладных баз данных |
| ОПК(У)-3 | Способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии |
| ОПК(У)-4 | Способность понимать сущность и значение информации в развитии современного общества, осознавать опасность и угрозу, возникающие в этом процессе, соблюдать основные требования информационной безопасности |
| ОПК(У)-5 | Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией |
| ОПК(У)-6 | Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической |

| | |
|-------------------------------------|--|
| | культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности |
| ОПК(У)-7 | Способность использовать в своей профессиональной деятельности знание иностранного языка |
| ОПК(У)-8 | Способность критически переосмысливать накопленный опыт, изменять при необходимости направление своей деятельности |
| ОПК(У)-9 | Способность получить организационно-управленческие навыки при работе в научных группах и других малых коллективах исполнителей |
| Профессиональные компетенции | |
| ПК(У)-1 | Способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности |
| ПК(У)-2 | Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения |
| ПК(У)-3 | Способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно-телекоммуникационной сети "Интернет" и в других источниках |
| ПК(У)-4 | Способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям |
| ПК(У)-5 | Способность понимать, совершенствовать и применять современный математический аппарат |
| ПК(У)-6 | Способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности |
| ПК(У)-7 | Способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы |
| ПК(У)-8 | Способность к реализации решений, направленных на поддержку социально-значимых проектов, на повышение информационной грамотности населения, обеспечения общедоступности информационных услуг |
| ПК(У)-9 | Способность к организации педагогической деятельности в конкретной предметной области (прикладная математика и информатика) |

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 01.02.03 Прикладная математика и информатика
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

| |
|---------------------|
| Бакалаврской работы |
|---------------------|

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

Студенту:

| Группа | ФИО |
|--------|-----------------------------|
| 8Б71 | Селезнёву Аркадию Олеговичу |

Тема работы:

| | |
|---|--|
| Исследование методов синтеза входной информации и реализация генеративно-состязательных сетей для расширения обучающей выборки | |
| Утверждена приказом директора (дата, номер) | |

| | |
|--|--|
| Срок сдачи студентом выполненной работы: | |
|--|--|

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

| | |
|---|--|
| <p>Исходные данные к работе <i>(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.).</i></p> | <p>Предметом исследования является задача генерации данных с помощью моделей нейронных сетей. Разрабатываемые алгоритмы должны уметь синтезировать реалистичные данные на основе переданных примеров.</p> |
| <p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(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;</i></p> | <p>Обзор литературы, обзор аналогичных проектов, исследование возможных способов генерации данных, выбор оптимальных алгоритмов нейронных сетей, выбор инструментов для работы (языки программирования, библиотеки).</p> |

| | |
|---|--|
| наименование дополнительных разделов, подлежащих разработке; заключение по работе). | |
|---|--|

Консультанты по разделам выпускной квалификационной работы

| Раздел | Консультант |
|---|---|
| Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | Трубченко Татьяна Григорьевна, доцент ОСГН ШБИП |
| Социальная ответственность | Черемискина Мария Сергеевна, ассистент ООД |

| | |
|---|--|
| Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику | |
|---|--|

Задание выдал руководитель:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---------------------|-------------------------|-------------------------------|----------------|-------------|
| Профессор ОИТ ИШИТР | Гергет Ольга Михайловна | д.т.н. | | |

Задание принял к исполнению студент:

| Группа | ФИО | Подпись | Дата |
|---------------|---------------------------|----------------|-------------|
| 8Б71 | Селезнёв Аркадий Олегович | | |

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

Студенту:

| | |
|---------------|-----------------------------|
| Группа | ФИО |
| 8Б71 | Селезнёву Аркадию Олеговичу |

| | | | |
|---------------------|--------------|---------------------------|--|
| Школа | ИШИТР | Отделение (НОЦ) | ОИТ |
| Уровень образования | Бакалавриат | Направление/специальность | 01.03.02 Прикладная математика и информатика |

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

| | |
|--|--|
| 1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих | Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г. Томска; В реализации задействованы два человека. |
| 2. Нормы и нормативы расходования ресурсов | Накладные расходы 16%; премии 30%; районный коэффициент 30%; минимальный размер оплаты труда (на 01.01.2021) – 12792 р. |
| 3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования | Коэффициент отчислений во внебюджетные фонды – 30%. |

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

| | |
|--|---|
| 1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения | Анализ конкурентных технических решений; SWOT-анализ. |
| 2. Планирование и формирование бюджета научных исследований | Формирование плана и графика разработки: определение структуры и трудоемкости работ, построение диаграммы Ганта. Формирование бюджета затрат. |
| 3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования | Расчет интегрального показателя ресурсоэффективности |

Перечень графического материала (с точным указанием обязательных чертежей):

| |
|--|
| 1. Оценка конкурентоспособности технических решений |
| 2. Матрица SWOT |
| 3. График проведения НИ |
| 4. Расчёт бюджета НИ |
| 5. Оценка ресурсной, финансовой и экономической эффективности НИ |

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

Задание выдал консультант:

| | | | | |
|------------------|-------------------------------|------------------------|---------|------|
| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
| Доцент ОСГН ШБИП | Трубченко Татьяна Григорьевна | К. Э. Н | | |

Задание принял к исполнению студент:

| | | | |
|--------|---------------------------|---------|------|
| Группа | ФИО | Подпись | Дата |
| 8Б71 | Селезнёв Аркадий Олегович | | |

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

| | |
|----------------|------------------------------------|
| Группа 8Б71 | ФИО Селезнёву Аркадию Олеговичу |
|----------------|------------------------------------|

| | | | |
|---------------------|-------------|---------------------------|--|
| Школа | ИШИТР | Отделение (НОЦ) | ОИТ |
| Уровень образования | Бакалавриат | Направление/специальность | 01.03.02 Прикладная математика и информатика |

Тема ВКР:

| | |
|---|---|
| «Исследование методов синтеза входной информации и реализация генеративно-состязательных сетей для расширения обучающей выборки» | |
| Исходные данные к разделу «Социальная ответственность»: | |
| Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения | Объект исследования: алгоритмы нейронных сетей для генерации данных: GAN, DCGAN, WGAN, VAE. Область применения: медицина. |
| Перечень вопросов, подлежащих исследованию, проектированию и разработке: | |
| 1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. | Трудовой кодекс Российской Федерации от 30.12.2001 N 197; ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования; СанПиН 2.2.2.542-96. Гигиенические требования к видеодисплейным терминалам, персональным ЭВМ и организации работы. |
| 2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия | Опасные и вредные факторы: 1. Электромагнитные поля. 2. Шум и вибрации. 3. Микроклимат в помещении. 4. Освещенность рабочей зоны. 5. Психофизиологические факторы. 6. Электробезопасность. |
| 3. Экологическая безопасность: | Атмосфера: выделение тепла при работе ПЭВМ. Гидросфера: загрязнение водных объектов при неправильной утилизации ПЭВМ. Литосфера: загрязнение почвы при неправильной утилизации ПЭВМ. |
| 4. Безопасность в чрезвычайных ситуациях: | Возможные ЧС: короткое замыкание в электропроводке, возгорание устройств ПЭВМ, возгорание устройств искусственного освещения. Наиболее типичная ЧС: пожар техногенного характера. |

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

Задание выдал консультант:

| | | | | |
|---------------|-----------------------------|------------------------|---------|------|
| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
| Ассистент ООД | Черемискина Мария Сергеевна | - | | |

Задание принял к исполнению студент:

| | | | |
|--------|---------------------------|---------|------|
| Группа | ФИО | Подпись | Дата |
| 8Б71 | Селезнёв Аркадий Олегович | | |

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 01.02.03 Прикладная математика и информатика
 Уровень образования бакалавр
 Отделение школы (НОЦ) Отделение информационных технологий
 Период выполнения _____ (осенний / весенний семестр 2018/2019 учебного года)

Форма представления работы:

Бакалаврская работа

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы: _____

| Дата контроля | Название раздела (модуля) / вид работы (исследования) | Максимальный балл раздела (модуля) |
|---------------|---|------------------------------------|
| | Написание основной части | 75 |
| | Написание части финансового менеджмента, ресурсоэффективности и ресурсосбережения | 15 |
| | Написание части социальной ответственности | 10 |

СОСТАВИЛ:

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|------------------------|----------------------------|------------------------|---------|------|
| Профессор ОИТ ИШИТР | Гергет Ольга Михайловна | д.т.н. | | |

СОГЛАСОВАНО:

Руководитель ООП

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---------------------|------------------------------|------------------------|---------|------|
| Доцент ОИТ ИШИТР | Шевелев Геннадий Ефимович | к.ф. – м.н. | | |

Реферат

Выпускная квалификационная работа содержит 81 страницу, 36 рисунков, 17 таблиц, 17 литературных источников и 2 приложения.

Ключевые слова: машинное обучение, искусственный интеллект, нейронные сети, генеративно-состязательные сети, вариационные автоэнкодеры, Python.

Объектом исследования являются модели нейронных сетей для генерации данных.

Цель работы – выбор эффективного метода синтеза данных в зависимости от типа исходной информации, расширение обучающей выборки посредством реализации алгоритмов нейронных сетей (вариационный автоэнкодер и генеративно-состязательная сеть).

В процессе исследования проводилось изучение существующих алгоритмов генерации и способов их программной реализации. Результатом исследования являются обученные нейронные сети для расширения выборки.

В процессе выполнения работы использовалась интегрированная среда разработки PyCharm для создания программной системы.

В первой главе представлено определение и математическое описание модели автоэнкодера и вариационного автоэнкодера.

Во второй главе представлено определение и математическое описание модели генеративно-состязательных сетей.

В третьей главе представлена реализация модели вариационного автоэнкодера.

В четвёртой главе представлена реализация модели генеративно-состязательной сети.

Результатом работы являются две готовые нейросети для генерации текстовых и графических данных.

Содержание

| | |
|---|----|
| Введение | 12 |
| 1. ВАРИАЦИОННЫЕ АВТОЭНКОДЕРЫ | 13 |
| 1.1. Понятие автоэнкодера..... | 13 |
| 1.2. Понятие вариационного автоэнкодера..... | 15 |
| 2. ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫЕ СЕТИ | 22 |
| 2.1. Понятие генеративно-состязательных сетей | 22 |
| 2.2. Математическое описание генеративно-состязательных сетей | 23 |
| 2.3. Бинарная кросс-энтропия..... | 26 |
| 2.4. Расстояние Вассерштейна..... | 27 |
| 2.5. Генеративно-состязательная сеть с расстоянием Вассерштейна | 27 |
| 3. РАЗРАБОТКА ВАРИАЦИОННОГО АВТОЭНКОДЕРА | 30 |
| 3.1. Анализ входных данных | 30 |
| 3.2. Создание обучающего датасета | 37 |
| 3.3. Программная реализации вариационного автоэнкодера..... | 38 |
| 3.4. Обучение VAE и анализ результатов | 40 |
| 4. РАЗРАБОТКА ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ СЕТИ..... | 43 |
| 4.1. Обзор входных данных | 43 |
| 4.2. Программная реализация генеративно-состязательной сети..... | 43 |
| 4.3. Обучение WGAN и анализ результатов | 44 |
| 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ | 48 |
| 5.1. Оценка коммерческого потенциала и перспективности научного исследования с позиции ресурсоэффективности и ресурсосбережения..... | 48 |
| 5.1.1. Анализ конкурентных технических решений | 48 |
| 5.2. SWOT-анализ | 50 |
| 5.3. Планирование и формирование бюджета научных исследований..... | 51 |
| 5.3.1. Структура выполнения работ..... | 51 |
| 5.3.2. Трудоёмкость выполнения и график проведения работ | 52 |
| 5.3.3. Формирование бюджета затрат..... | 56 |
| 5.3.3.1. Материальные затраты | 56 |
| 5.3.3.2. Затраты на оборудование | 57 |
| 5.3.3.3. Амортизационные отчисления..... | 57 |
| 5.3.3.4. Основная заработная плата | 58 |
| 5.3.3.5. Дополнительная заработная плата | 59 |

| | |
|---|----|
| 5.3.3.6. Отчисления во внебюджетные фонды (страховые отчисления) ... | 59 |
| 5.3.3.7. Накладные расходы..... | 60 |
| 5.3.3.8. Итоговый бюджет затрат | 61 |
| 5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.. | 61 |
| 5.4.1. Интегральный показатель ресурсоэффективности | 61 |
| 5.5. Выводы по разделу | 62 |
| 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ | 63 |
| 6.1. Правовые и организационные вопросы обеспечения безопасности... 63 | |
| 6.1.1. Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства..... | 63 |
| 6.1.2. Организационные мероприятия при компоновке рабочей зоны..... | 64 |
| 6.2. Производственная безопасность | 65 |
| 6.2.1. Анализ выявленных вредных и опасных факторов | 66 |
| 6.2.1.1. Электромагнитные поля | 66 |
| 6.2.1.2. Производственный шум | 67 |
| 6.2.1.3. Микроклимат в помещении | 67 |
| 6.2.1.4. Освещённость рабочей зоны..... | 68 |
| 6.2.1.5. Психофизиологические факторы..... | 69 |
| 6.2.1.6. Электробезопасность | 69 |
| 6.2.2. Обоснование мероприятий по снижению воздействия | 70 |
| 6.3. Экологическая безопасность | 71 |
| 6.4. Безопасность в чрезвычайных ситуациях | 72 |
| 6.5. Выводы и рекомендации..... | 74 |
| Заключение | 75 |
| Список использованных источников..... | 76 |
| Приложение А | 78 |
| Приложение Б..... | 80 |

Введение

В современном мире поток новой создаваемой и обрабатываемой информации огромен. Однако многие сферы человеческой деятельности страдают от нехватки достаточного объёма данных для последующего анализа. Одной из таких сфер является медицина. Зачастую сбор и разметка медицинских данных требуют большого количества времени и сил. Из-за этого в данной сфере очень актуальны алгоритмы и модели генерации новой информации на основе имеющейся. С этой задачей отлично справляются нейронные сети, а именно некоторые их подвиды: генеративно-состязательные сети, автоэнкодеры, вариационные автоэнкодеры.

Основная задача выпускной квалификационной работы – расширение выборки данных двух видов: текстовая информация об аллергической реакции пациентов и графические изображения сосудов.

Целью выпускной квалификационной работы является разработка вариационного автоэнкодера и генеративно-состязательной сети для расширения выборки имеющихся данных.

1. ВАРИАЦИОННЫЕ АВТОЭНКODЕРЫ

1.1. Понятие автоэнкодера

Автоэнкодер (АЕ) – это тип нейронной сети, которая состоит из двух частей: кодера и декодера. Задача АЕ – закодировать набор данных x в представление, называемое кодом z , а затем использовать это представление для восстановления данных и получить \tilde{x} . Следовательно, автоэнкодер будет отображать $x \rightarrow \tilde{x}$ через кодер $f(x) \rightarrow z$ и декодер $g(z) \rightarrow \tilde{x}$.

Обычно, АЕ используются для таких задач, как уменьшение размерности и изучение свойств данных. Хотя копирование входа в выход и является основной задачей АЕ, чаще всего наибольшее внимание отдаётся коду z , созданному кодировщиком; идея состоит в том, чтобы АЕ улавливал характерные особенности данных в представлении z . Это может произойти как побочный результат способности хорошо выполнять задачу преобразования $x \rightarrow \tilde{x}$. АЕ не является задачей обучения с учителем, так как не указаны ключи y . Процесс обучения АЕ можно описать как минимизацию функции потерь

$$\mathcal{L}(x, g \circ f(x)) = \mathcal{L}(x, \tilde{x})$$

Поскольку цель обучения состоит в том, чтобы АЕ научилась копировать вход x на выход \tilde{x} , этой функцией потерь может быть, например, MSE (среднеквадратичная ошибка) между этими сущностями

$$\mathcal{L}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^n [x^{(i)} - f_{\theta}(g_{\phi}(x^{(i)}))]^2,$$

где $x^{(i)}$ – оригинальные данные, $f_{\theta}(g_{\phi}(x^{(i)}))$ – восстановленные данные.

Учитывая, что АЕ в основном является FNN (нейронной сетью с прямой связью), она может быть обучена с помощью методов, используемых для обучения стандартных FNN, таких как GD (градиентный спуск) и BP (обратное распространение ошибки). Кроме того, существуют связи между АЕ и классическими методами обучения представления, а именно анализом PCA

(метод главных компонент). Если сеть АЕ обучается по критерию MSE и состоит из одного линейного скрытого слоя с p единицами измерения, то кодер проецирует входные данные на промежуток p первых главных компонент данных. Однако нелинейность в скрытом слое даст совершенно другие характеристики, в том числе способность передавать другие аспекты входного распределения. Таким образом, мы можем достичь «более мощного нелинейного обобщения PCA». Кроме того, более абстрактные и глубокие представления могут быть визуализированы путем группировки АЕ, то есть подключением нескольких автоэнкодеров в сетевую структуру.

Размерность сущностей, входящих в АЕ, будет отражать совершенно иные характеристики сети. Если код z имеет меньшую размерность, чем данные x , то АЕ называется неполным. Неполный АЕ может преуспеть в захвате ключевых особенностей данных в своем коде, так как сеть вынуждена реконструировать данные, используя представление в более низком измерении. Слишком большая ёмкость АЕ может ограничить полезность изучаемых признаков в том смысле, что АЕ может хорошо выполнять задачу преобразования $x \rightarrow \tilde{x}$, но не может захватить характеризующие признаки из данных. Кроме того, если размерность кода больше, чем размерность данных, мы имеем сверхполный АЕ. В сверхполном случае мы также можем столкнуться с проблемой кода, неспособного отразить важные особенности данных, даже если АЕ может преуспеть в задаче копирования. Эти следствия, связанные с измерением, показывают, что нужно быть внимательным при проектировании архитектуры любого АЕ, чтобы получить сеть, которая хорошо функционирует в соответствии со своей предполагаемой задачей. Следовательно, ёмкость и размеры должны быть модулированы в соответствии с характеристиками имеющихся данных.

Существует определенный тип АЕ, называемый регуляризованным автоэнкодером, который, помимо того, что способен хорошо выполнять задачу копирования, может находить компромисс в отношении ёмкости и размера. Это достигается изменением функции потерь L , и тогда АЕ

приобретает такие характеристики, как устойчивость к шуму и разреженность представления. Для воспроизведения разреженного автоэнкодера функция потерь должна иметь дополнительный штрафной член $\Omega(z)$, который поощряет разреженность кода, следовательно, функция потерь для регуляризованного АЕ задается как

$$\mathcal{L}(x, g(z)) + \Omega(z)$$

Существуют и другие типы АЕ, такие как шумоподавляющий автоэнкодер, сжимающий автоэнкодер и вариационный автоэнкодер (VAE).

1.2. Понятие вариационного автоэнкодера

VAE по существу является стохастическим обобщением АЕ. Здесь кодер и декодер состоят из стохастических отображений. Аналогом функции кодирования $f(x)$ является распределение кодирования $q_\phi(z|x)$, а аналогом функции декодирования $g(z)$ – распределение декодирования $p_\theta(x|z)$, где ϕ – параметры модели распознавания, θ – параметры генеративной модели.

Целью VAE является поиск такого распределения $q_\phi(z|x)$ некоторых скрытых переменных z , благодаря которому мы можем получить $z \sim q_\phi(z|x)$ для создания новых данных $\tilde{x} \sim p_\theta(x|z)$. Именно благодаря получению значения из распределения мы можем избежать статичности в генерации и восстановления идентичных входных примеров.

Для вариационного автоэнкодера функция потерь выглядит следующим образом:

$$\mathcal{L}(\theta, \phi) = -E_{z \sim q_\phi(z|x)}[p_\theta(x|z)] + D_{KL}(q_\phi(z|x) || p_\theta(z))$$

где E – математическое ожидание, $p_\theta(x|z)$ и $q_\phi(z|x)$ – условные вероятности, D_{KL} – дивергенция Кульбака-Лейблера.

Рассмотрим более подробно дивергенцию Кульбака-Лейблера. Она является мерой отличия одного распределения вероятностей от другого. Чем меньше её значение, тем более распределения схожи между собой. Для

дискретных распределений вероятностей P и Q дивергенция Кульбака-Лейблера определяется как

$$D_{KL}(P \parallel Q) = \sum_x P(X = x) \log \frac{P(X = x)}{Q(X = x)} = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

Свойства KL:

1. $KL(P \parallel Q)$ и $KL(Q \parallel P) \geq 0$
2. $KL(P \parallel Q) \neq KL(Q \parallel P)$

Предположим, мы имеем многомерные нормальные распределения, определённые как

$$p(x) = \mathcal{N}(x; \mu_1; \Sigma_1)$$

$$q(x) = \mathcal{N}(x; \mu_2; \Sigma_2)$$

где μ_1 и μ_2 – средние, Σ_1 и Σ_2 – матрицы ковариации.

И многомерная плотность вероятности имеет вид

$$\mathcal{N}(x; \mu; \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right)$$

При условии, что оба распределения имеют одну размерность k (и опуская доказательство)

$$D_{KL}(p(x) \parallel q(x)) = \frac{1}{2} \left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - k + \text{tr}(\Sigma_2^{-1} \Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) \right]$$

tr – след матрицы.

Однако, вернёмся к основной задаче VAE и рассмотрим проблему приближённого вывода. Пусть x – набор наблюдаемых переменных и z – набор скрытых переменных с совместным распределением $p(z, x)$. Тогда проблемой вывода является вычисление условного распределения $p(z|x)$. По теореме Байеса

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

где $p(x)$ не может быть найдено, потому что

$$p(x) = \int_z p(x|z)p(z)dz = \int_z p(x, z)dz$$

Интеграл считается трудноразрешимым, так как подразумевает интегрирование по всему вектору скрытых переменных z , и, соответственно, является многомерным.

Альтернативным подходом является аппроксимации $P(z|x)$ другим распределением, определённым $Q(z|x)$ таким образом, что оно имеет обрабатываемое решение. Это достигается путём вариационного вывода, основная идея которого состоит в представлении пробелы вывода в виде проблемы оптимизации. Моделируя $P(z|x)$ с использованием $Q(z|x)$ (при условии, что $Q(z|x)$ имеет простое распределение, например, как у гауссиана), рассчитаем дивергенцию Кульбака-Лейблера

$$\begin{aligned}
 D_{KL} \left(Q_\phi(z|x) \parallel P_\theta(z|x) \right) &= \sum_z Q_\phi(z|x) \log \frac{Q_\phi(z|x)}{P_\theta(z|x)} = \\
 &= E_{z \sim Q_\phi(z|x)} \left(\log \frac{Q_\phi(z|x)}{P_\theta(z|x)} \right) = \\
 &= E_{z \sim Q_\phi(z|x)} (\log Q_\phi(z|x) - \log P_\theta(z|x)) = \\
 &= E_{z \sim Q_\phi(z|x)} \left(\log Q_\phi(z|x) - \log \frac{P_\theta(x|z)P_\theta(z)}{P_\theta(x)} \right) = \\
 &= E_{z \sim Q_\phi(z|x)} (\log Q_\phi(z|x) - \log P_\theta(x|z) - \log P_\theta(z) + \log P_\theta(x))
 \end{aligned}$$

$\log P_\theta(x)$ можно вынести за математическое ожидание, так как $P_\theta(x)$ не содержит z .

$$\begin{aligned}
 D_{KL} \left(Q_\phi(z|x) \parallel P_\theta(z|x) \right) - \log P_\theta(x) &= \\
 &= E_{z \sim Q_\phi(z|x)} (\log Q_\phi(z|x) - \log P_\theta(x|z) - \log P_\theta(z))
 \end{aligned}$$

Сменим знак выражения и запишем

$$\begin{aligned}
 \log P_\theta(x) - D_{KL} \left(Q_\phi(z|x) \parallel P_\theta(z|x) \right) &= \\
 &= E_{z \sim Q_\phi(z|x)} (\log P_\theta(x|z)) - E_{z \sim Q_\phi(z|x)} (\log Q_\phi(z|x) - \log P_\theta(z)) = \\
 &= E_{z \sim Q_\phi(z|x)} (\log P_\theta(x|z)) - D_{KL} (Q_\phi(z|x) \parallel P_\theta(z))
 \end{aligned}$$

Мы получили целевую функцию VAE, в которой первый член представляет вероятность реконструкции, а второй член обеспечивает похожесть изученного распределения Q на распределение P .

Таким образом, функция потерь является обратным от целевой функции

$$\mathcal{L}(\theta, \phi) = -E_{z \sim Q_\phi(z|x)}[\log P_\theta(x|z)] + D_{KL}(Q_\phi(z|x) || P_\theta(z))$$

в то же время

$$\log P_\theta(x) - D_{KL}(Q_\phi(z|x) || P_\theta(z|x)) = -\mathcal{L}(\theta, \phi)$$

Обратим внимание на части функции потерь. Левая часть также называется логарифмической вероятностью, потому что если мы возьмём $P_\theta(x|z) = N(\mu_\theta(z), \Sigma_\theta(z))$ (гауссова функция), то, найдя логарифм от этой функции, мы получим квадратичную ошибку между образцом данных x и средним распределения Гаусса.

$$P_\theta(x|z) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_\theta(z)|}} \exp\left(-\frac{1}{2}(x - \mu_\theta(z))^T \Sigma_\theta(z)^{-1}(x - \mu_\theta(z))\right)$$

$$\log P_\theta(x|z) \propto (x - \mu_\theta(z))^T \Sigma_\theta(z)^{-1}(x - \mu_\theta(z))$$

что является квадратичной ошибкой реконструкции, также обозначаемая как достоверность данных в обратной задаче.

Правая часть функции потерь называется регуляризатором, так как дивергенция Кульбака-Лейблера штрафует функцию, если плотности распределения скрытых переменных $Q_\phi(z|x)$ отклоняются от $P_\theta(z) \sim N(0,1)$.

$N(0,1)$ является самым простым выбором для $P_\theta(z)$. Допустим, мы хотим, чтобы $Q_\phi(z|x)$ было гауссианом с параметрами $\mu_\phi(x)$ и $\Sigma_\phi(x)$, дивергенция Кульбака-Лейблера примет закрытую форму, описанную выше. Таким образом

$$\begin{aligned} D_{KL}(N(\mu_\phi(x), \Sigma_\phi(x)) || N(0,1)) &= \\ &= \frac{1}{2} \left[\text{tr}(\Sigma_\phi(x)) + \mu_\phi(x)^T \mu_\phi(x) - k - \log|\Sigma_\phi(x)| \right] \end{aligned}$$

Упростим выражение

$$\begin{aligned}
D_{KL} \left(N \left(\mu_\phi(x), \Sigma_\phi(x) \right) \parallel N(0,1) \right) &= \\
&= \frac{1}{2} \left[\sum_k \Sigma_\phi(x) + \sum_k \mu_\phi(x)^2 - \sum_k 1 - \log \prod_k \Sigma_\phi(x) \right] = \\
&= \frac{1}{2} \left[\sum_k \Sigma_\phi(x) + \sum_k \mu_\phi(x)^2 - \sum_k 1 - \sum_k \log \Sigma_\phi(x) \right] = \\
&= \frac{1}{2} \sum_k \left[\Sigma_\phi(x) + \mu_\phi(x)^2 - 1 - \log \Sigma_\phi(x) \right]
\end{aligned}$$

На практике принято заменять $\Sigma_\phi(x)$ на $\exp(\Sigma_\phi(x))$ для вычислительной устойчивости. В итоге имеем

$$\begin{aligned}
D_{KL} \left(N \left(\mu_\phi(x), \Sigma_\phi(x) \right) \parallel N(0,1) \right) &= \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x) \right] \\
\mathcal{L}(\theta, \phi) &= -E_{z \sim Q_\phi(z|x)} [\log P_\theta(x|z)] \\
&\quad + \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x) \right]
\end{aligned}$$

Теперь рассмотрим задачу оптимизации функции потерь. Наша цель найти оптимальные θ, ϕ , такие, что

$$\theta^*, \phi^* = \arg \min \mathcal{L}(\theta, \phi)$$

В вариационных байесовских методах наша функция потерь также известна как нижняя граница доказательств или ELBO (evidence lower bound). Определение «нижняя граница» исходит из того факта, что дивергенция Кульбака-Лейблера всегда положительна и тогда $\mathcal{L}(\theta, \phi)$ есть нижняя граница $\log P_\theta(x)$. Таким образом, минимизируя потери, мы максимизируем нижнюю границу вероятности генерации реалистичных данных.

Воспроизведём оптимизацию по переменным θ, ϕ для одновременного обучения $Q_\phi(z|x)$ и $P_\theta(x|z)$

$$\min_{\theta, \phi} \mathcal{L}(\theta, \phi) = \min_{\theta, \phi} \left\{ -E_{z \sim Q_\phi(z|x)} [\log P_\theta(x|z)] \right. \\ \left. + \frac{1}{2} \sum_k [\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x)] \right\}$$

$$i = 1, 2, \dots, n$$

$$\hat{\theta}_i = \nabla_\theta \mathcal{L}(\theta, \phi)$$

$$\hat{\theta}_i = \nabla_\theta \left\{ -E_{z \sim Q_\phi(z|x)} [\log P_\theta(x|z)] \right. \\ \left. + \frac{1}{2} \sum_k [\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x)] \right\}$$

Второй член не зависит от θ , поэтому он равен нулю. В первом члене занесём вычисление производной под математическое ожидание и получим (с помощью оценки Монте-Карло)

$$\hat{\theta}_i = \nabla_\theta \mathcal{L}(\theta, \phi) = -\frac{1}{L} \sum_{l=1}^L \nabla_\theta \log P_\theta(x|z^{(l)}),$$

где $z^{(l)} \sim Q_\phi(z|x)$.

Для ϕ аналогично с учётом $\hat{\theta}$

$$\hat{\phi}_i = \nabla_\phi \mathcal{L}(\theta = \hat{\theta}, \phi)$$

$$\hat{\phi}_i = \nabla_\phi \left\{ -E_{z \sim Q_\phi(z|x)} [\log P_\theta(x|z)] \right. \\ \left. + \frac{1}{2} \sum_k [\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x)] \right\}$$

Производная ∇_ϕ сложно вычислима, так как ϕ появляется в распределении, относительно которого берётся математическое ожидание, другими словами

$$\nabla_\phi E_{Q_\phi(z|x)} [f(z)] \neq E_{Q_\phi(z|x)} [\nabla_\phi f(z)]$$

Таким образом, мы должны попытаться переписать математическое ожидание таким образом, чтобы ϕ появилось внутри него. Только тогда мы сможем пронести градиент внутрь математического ожидания. Допустим

$$E_{Q_\phi(z|x)}[f(z)] = E_{p(\epsilon)}[f(g_\phi(\epsilon, x))]$$

где $z = g_\phi(\epsilon, x)$ с $\epsilon \sim N(0,1)$ – любая линейная трансформация. В нашем случае

$$g_\phi(\epsilon, x) = \mu_\phi(x) + \epsilon \circ \Sigma_\phi(x)^{1/2} = z \sim N(\mu(x), \Sigma(x)),$$

$N(\mu(x), \Sigma(x))$ получается из $N(0,1)$ путём линейной трансформации.

Вместо того, чтобы брать значения $z \sim Q_\phi(z|x)$, мы берём их из $N(0,1)$, то есть $\epsilon \sim N(0,1) \Rightarrow \epsilon \sim p(\epsilon)$ и затем применяем линейную трансформацию используя $z = \mu_\phi(x) + \epsilon \circ \Sigma_\phi(x)^{1/2}$, получая $z \sim N(\mu(x), \Sigma(x))$.

Теперь ничего не мешает нам продолжить вычисление градиентов (с применением оценки Монте-Карло)

$$\begin{aligned} \hat{\phi}_i &= \nabla_\phi \left\{ -E_{z^{(l)} \sim p(\epsilon)} [\log P_\theta(x|z^{(l)})] \right. \\ &\quad \left. + \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x) \right] \right\} \\ \hat{\phi}_i &= -E_{z^{(l)} \sim p(\epsilon)} [\nabla_\phi \log P_\theta(x|z^{(l)})] \\ &\quad + \nabla_\phi \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x) \right] \\ \hat{\phi}_i &= -\frac{1}{S} \sum_{s=1}^S \nabla_\phi \log P_\theta(x|z^{(l)}) + \nabla_\phi \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi(x)^2 - 1 - \Sigma_\phi(x) \right] \end{aligned}$$

где $z^{(l)} = \mu_\phi(x) + \epsilon \circ \sigma_\phi(x)$ и $\epsilon^{(l)} \sim N(0,1)$.

Применённая техника, позволившая рассчитать градиент через функцию ошибки, называется «reparameterization trick» или трюк с перепараметризацией.

2. ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫЕ СЕТИ

2.1. Понятие генеративно-сопязательных сетей

Генеративно-сопязательная сеть (англ. Generative adversarial network, сокращённо GAN) – алгоритм машинного обучения без учителя, построенный на комбинации из двух нейронных сетей, одна из которых (сеть G, от Generator) генерирует образцы, а другая (сеть D, от Discriminator) старается отличить правильные («подлинные») образцы от неправильных. Так как сети G и D имеют противоположные цели – создать образцы и отбраковать образцы – между ними возникает антагонистическая игра (некооперативная игра, в которой участвуют два или более игроков, выигрыши которых противоположны). Впервые GAN была описана в 2014 году сотрудником компании Google[4].

Используя набор переменных латентного пространства (латентные переменные – переменные, которые не могут быть измерены в явном виде, а могут быть только выведены через математические модели с использованием наблюдаемых переменных), генеративная сеть пытается создать новый образец, смешав несколько исходных образцов. Дискриминативная сеть обучается различать подлинные и поддельные образцы, а результаты различения подаются на вход генеративной сети так, чтобы она смогла подобрать лучший набор латентных параметров, и дискриминативная сеть уже не смогла бы отличить подлинные образцы от поддельных. Таким образом целью сети G является повысить процент ошибок сети D, а целью сети D является, наоборот, улучшение точности распознавания.

Дискриминационная сеть D, анализируя образцы из оригинальных данных и из подделанных генератором, достигает некоторой точности различения. Генератор при этом начинает со случайных комбинаций параметров латентного пространства, а после оценки полученных образцов сетью D применяется метод обратного распространения ошибки (backpropagation), который позволяет улучшить качество генерации,

подправив входной набор латентных параметров. Постепенно искусственные изображения на выходе генеративной сети становятся всё более качественными. Сеть D реализуется как свёрточная нейронная сеть, в то время как сеть G наоборот разворачивает изображение на базе скрытых параметров.

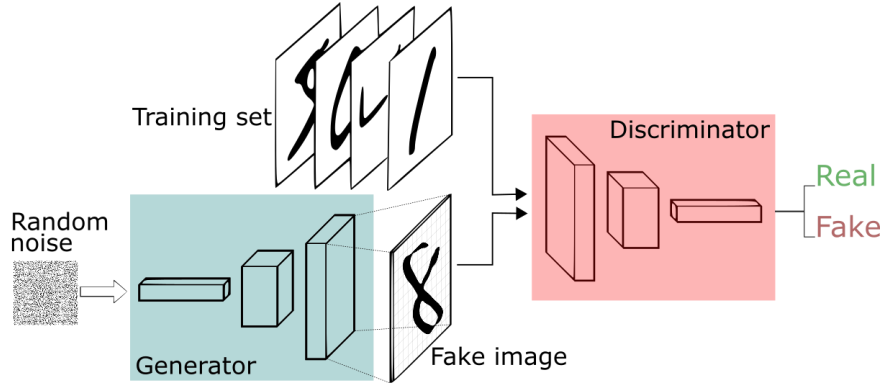


Рисунок 1. Схематичный принцип работы GAN

2.2. Математическое описание генеративно-сопоставительных сетей

Пусть X – некоторое пространство объектов. На некотором вероятностном пространстве Ω задана векторная случайная величина $x : \Omega \rightarrow X$ с распределением вероятностей, имеющим плотность $p(x)$ такую, что подмножество пространства X , на котором $p(x)$ принимает ненулевые значения – это, например, фотографии человеческих лиц. Нам дана случайная выборка независимых одинаково распределённых случайных величин фотографий лиц $\{x_i, i \in [1, N], x_i \sim p(x)\}$. Дополнительно определим вспомогательное пространство $Z = R^n$ и случайную величину $z : \Phi \rightarrow Z$ с распределением вероятностей, имеющим плотность $q(z)$. $D : X \rightarrow (0, 1)$ – функция-дискриминатор. Эта функция принимает на вход объект $x \in X$ (в нашем примере – картинку соответствующего размера) и возвращает вероятность того, что входная картинка является фотографией человеческого лица. $G : Z \rightarrow X$ – функция-генератор. Она принимает значение $z \in Z$ и выдаёт объект пространства X , то есть, в нашем случае, картинку.

Предположим, что у нас есть идеальный дискриминатор D . Для любого примера x он выдаёт истинную вероятность принадлежности этого примера

заданному подмножеству X , из которого получена выборка $\{x_i\}$. Переформулируя задачу обмана дискриминатора на вероятностном языке мы получаем, что необходимо максимизировать вероятность, выдаваемую идеальным дискриминатором на сгенерированных примерах. Таким образом оптимальный генератор находится как $G^* = \arg \max_G E_{z \sim q(x)} D_k(G(z))$. Так как $\log(x)$ – монотонно возрастающая функция и не имеет положения экстремумов аргумента, эту формулу можно переписать в виде $G^* = \arg \max_G E_{z \sim q(x)} \log D_k(G(z))$, что будет удобно в дальнейшем. В реальности обычно идеального дискриминатора нет и его надо найти. Так как задача дискриминатора — предоставлять сигнал для обучения генератора, вместо идеального дискриминатора достаточно взять дискриминатор, идеально отделяющий настоящие примеры от сгенерированных текущим генератором, т.е. идеальный только на подмножестве X из которого генерируются примеры текущим генератором. Эту задачу можно переформулировать, как поиск такой функции D , которая максимизирует вероятность правильной классификации примеров как настоящих или сгенерированных. Это называется задачей бинарной классификации и в данном случае мы имеем бесконечную обучающую выборку: конечное число настоящих примеров и потенциально бесконечное число сгенерированных примеров. У каждого примера есть метка: настоящий он или сгенерированный.

Распишем метод максимального правдоподобия для нашей задачи классификации. Итак, наша выборка $S = \{(x, 1), x \sim p(x)\} \cup \{(G(z), 0), z \sim q(z)\}$. Определим плотность распределения $f(\xi | \eta = 1) = D(\xi)$, $f(\xi | \eta = 0) = 1 - D(\xi)$, тогда $f(\xi | \eta)$ – это переформулировка дискриминатора D , выдающего вероятность класса 1 (настоящий пример) в виде распределения на классах $\{0, 1\}$. Так как $D(\xi) \in (0, 1)$, это определение задает корректную плотность вероятности. Тогда оптимальный дискриминатор можно найти как:

$$D^* = f^*(\xi|\eta) = \arg \max_f f(\xi_1, \dots | \eta_1, \dots) = \arg \max_f \prod_i f(\xi_i | \eta_i)$$

Сгруппируем множители для $\eta_i = 0$ и $\eta_i = 1$:

$$\begin{aligned} D^* &= \arg \max_f \prod_{i,\eta=1} f(\xi_i | \eta_i = 1) \prod_{i,\eta=0} f(\xi_i | \eta_i = 0) = \\ &= \arg \max_D \prod_{x_i \sim p(x)} D(x_i) \prod_{z_i \sim q(z)} (1 - D(G(z_i))) = \\ &= \arg \max_D \prod_{x_i \sim p(x)} \log D(x_i) \prod_{z_i \sim q(z)} \log(1 - D(G(z_i))) \end{aligned}$$

И при стремлении размера выборки в бесконечность, получаем:

$$D^* = \arg \max_D E_{x_i \sim p(x)} \log D(x_i) + E_{z_i \sim q(z)} \log(1 - D(G(z_i)))$$

Итого, получаем следующий итерационный процесс:

1. Устанавливаем произвольный начальный $G_0(z)$.
2. Начинается k -я итерация, $k = 1, \dots, K$.
3. Ищем оптимальный для текущего генератора дискриминатор:

$$D_k = \arg \max_D E_{x_i \sim p(x)} \log D(x_i) + E_{z_i \sim q(z)} \log(1 - D(G_{k-1}(z_i)))$$

4. Улучшаем генератор, используя оптимальный дискриминатор:

$$G_k = \arg \max_G E_{z \sim q(z)} \log D_k(G(z))$$

Важно находиться в окрестности текущего генератора. Если отойти далеко от текущего генератора, то дискриминатор перестанет быть оптимальным и алгоритм перестанет быть верным.

5. Задача обучения генератора считается решенной, когда $D_k(x) = 1/2$ для любого x . Если процесс не сошелся, то переходим на следующую итерацию в пункт (2).

Этот алгоритм сводится в одну формулу, задающую в некотором смысле минимакс-игру между дискриминатором и генератором:

$$\min_G \max_D L(D, G) = E_{x \sim p(x)} \log D(x) + E_{z \sim q(z)} \log(1 - D(G(z)))$$

Обе функции D, G могут быть представлены в виде нейросетей: $D(x) = D(x, \theta_1), G(z) = G(z, \theta_2)$, после чего задача поиска оптимальных функций

сводится к задаче оптимизации по параметрам и ее можно решать с помощью традиционных методов: `backpropagation` и `SGD`. Так как нейросеть – это универсальный аппроксиматор функций, $G(z, \theta_2)$ может приблизить произвольное распределение вероятностей, что снимает вопрос выбора распределения $q(z)$. Это может быть любое непрерывное распределение в некоторых разумных рамках. Например, `Uniform(-1, 1)` (непрерывное равномерное распределение) или `N(0, 1)` (нормальное распределение).

2.3. Бинарная кросс-энтропия

Бинарная кросс-энтропия решает задачу бинарной классификации. Кросс-энтропия измеряет расхождение между двумя вероятностными распределениями. Если кросс-энтропия велика, это означает, что разница между двумя распределениями велика, а если кросс-энтропия мала, то распределения похожи друг на друга. Формула бинарной кросс-энтропии выглядит следующим образом:

$$BCE = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p))$$

где y – индикатор проверяемого класса, p – прогнозируемая вероятность модели.

В случае генеративно-сопоставительных сетей бинарная кросс-энтропия будет вычисляться для генератора (между 1 и результатом дискриминатора от сгенерированного изображения) и дискриминатора (реальное: между 1 и результатом дискриминатора для реального изображения, нереальное: между 0 и результатом дискриминатора от сгенерированного изображения). Оптимизация весов будет осуществляться от значения функции для генератора и для дискриминатора (суммы значений реального и нереального).

2.4. Расстояние Вассерштейна

Расстояние Вассерштейна (Earth mover's (EM) distance) можно интерпретировать как минимальное количество работы, необходимое для трансформации одного распределения в другое.

$$EM(P, Q) = \inf_{\gamma \in \Pi(P, Q)} (E_{(u, v) \in \gamma} (\|u - v\|))$$

Инфимум (наибольшая нижняя грань) в EM берётся по $\Pi(P, Q)$, являющимся коллекцией всех совместных распределений, предельные значения которых есть P и Q . Тогда $\gamma(u, v)$ – это план передачи, указывающий на то, каким образом мы перераспределяем «землю» из u в v , подчиняющийся некоторым ограничениям для поддержания действительных распределений после таких передач. Вычисление EM-расстояния само по себе является задачей оптимизации, которая заключается в поиске оптимального плана передачи $\gamma(u, v)$.

Применение EM-расстояния обладает значительным преимуществом. Допустим, есть два распределения P и Q , являющиеся двумя параллельными линиями. Одна линия фиксирована в $x = 0$, а другая может перемещаться вдоль X , но изначально находится в $x = \theta$, где $\theta > 0$. Можно показать, что KL (расстояние Кульбака-Лейблера), TL (полная вариация) и JS (расстояние Дженсена-Шэннона) равняются $KL(P||Q) = +\infty$, $TV(P, Q) = 1$, $JS(P, Q) = \frac{1}{2}$. Ни одна из этих мер различия не зависит от θ , и, соответственно, не может быть дифференцирована по θ , делая распределения P и Q похожими друг на друга. С другой стороны, EM-расстояние есть $EM(P, Q) = |\theta|$, чей градиент может привести Q к P .

2.5. Генеративно-состязательная сеть с расстоянием Вассерштейна

Теперь рассмотрим то, как применить EM-расстояние в обучении GAN [6]. Положим P_r как распределение реальных примеров, а P_g как распределение

нереальных (сгенерированных) примеров. P_r и P_g заменят P и Q в уравнении EM-расстояния. Так как вычисление EM-расстояния является проблемой оптимизации, то эта задача является достаточно сложной, особенно с учётом того, что вычисление должно проводиться на каждой итерации обучения GAN. К счастью, нахождение EM-расстояния можно упростить с помощью теоремы двойственности Канторовича-Рубинштейна:

$$W(P_r, P_g) = \sup_{\|f\|_L \leq 1} E_{u \in P_r}[f(u)] - E_{v \in P_g}[f(v)]$$

Здесь супремум (точная верхняя грань) берётся по всем 1-Липшицевым непрерывным функциям, обозначаемым $\|f\|_L \leq 1$. Основываясь на 1-Липшицевом отображении, функция f должна удовлетворять следующему:

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|$$

Как же найти 1-Липшицево отображение для вычисления расстояния Вассерштейна для GAN между распределениями реальных (P_r) и нереальных (P_g) выходов? Поскольку нейронные сети являются универсальными аппроксиматорами функций, мы можем обучить нейронную сеть приближать функцию расстояния Вассерштейна, используя дискриминатор в качестве классификатора. Для WGAN (GAN с применением расстояния Вассерштейна) дискриминатор может выступать неким критиком, который будет возвращать скалярную оценку вместо вероятностного значения. Можно интерпретировать эту оценку как показатель, насколько реалистичным было входное изображение. Таким образом, можно определить методику подсчёта потерь:

- реальная часть ошибки дискриминатора: $L_{real}^D = -\frac{1}{N} \sum_i D(x_i)$
- нереальная часть ошибки дискриминатора: $L_{fake}^D = -\frac{1}{N} \sum_i D(G(z_i))$
- ошибка генератора: $L^G = -\frac{1}{N} \sum_i D(G(z_i))$

Кроме того, нужно гарантировать, чтобы 1-Липшицево свойство выполняется по ходу тренировки. Для этого рекомендуется зажимать значения весов [7] в небольшой области, например $[-0.01, 0.01]$.

Экспериментально было выяснено [8], что зажатие весов для выполнения 1-Липшецева свойства может приводить к взрыву или затуханию градиентов. Кроме того, ограничение весов может привести к неэффективному использованию ресурсов, что означает, что сеть-критик (дискриминатор) ограничивается обучением только на некоторых простых функциях, исключая более сложные.

Поэтому вместо зажатия весов был предложен штраф градиента. Сеть, использующая штраф градиента для нахождения расстояния Вассерштейна, называется WGAN-GP (gradient penalty). Процедура штрафа градиента выполняется на каждой итерации и может быть описан следующими шагами:

1. Для каждой пары реального и нереального примера $(x^{[i]}, \tilde{x}^{[i]})$ в полученном батче выбирается случайное число $\alpha^{[i]}$ из равномерного распределения ($\alpha^{[i]} \in U(0,1)$).
2. Вычисляется интерполяция между реальным и нереальным примерами: $\check{x}^{[i]} = \alpha x^{[i]} + (1 - \alpha)\tilde{x}^{[i]}$, результирующее в батч интерполируемых примеров.
3. Находятся значения дискриминатора (критика) от всех интерполируемых примеров, $D(\check{x}^{[i]})$.
4. Вычисляются градиенты от полученных выходов дискриминатора, $\nabla_{\check{x}^{[i]}} D(\check{x}^{[i]})$.
5. Найдём штраф градиента: $L_{gp}^D = \frac{1}{N} \sum_i (\|\nabla_{\check{x}^{[i]}} D(\check{x}^{[i]})\|_2 - 1)^2$.

Общий loss дискриминатора будет:

$$L_{total}^D = L_{real}^D + L_{fake}^D + \lambda L_{gp}^D$$

где λ – изменяемый гиперпараметр.

3. РАЗРАБОТКА ВАРИАЦИОННОГО АВТОЭНКОДЕРА

3.1. Анализ входных данных

Исходные данные представлены в виде таблица MS Excel, где собрана информация об аллергических реакциях 1160 пациентов. Для каждого пациента указан ID, пол, возраст, дата рождения, дата сдачи анализа и город. Кроме того, каждому пациенту поставлен в соответствие перечень аллергенов (125 шт.), где на пересечении записана реакция на данный аллерген (какое-либо число или отсутствие числа, если реакции нет). Аллергены разделены по видам и источникам.

Данные представлены на рисунке 2:

| | 1 | 2 | 3 | 4 | 5 | 7 | 8 | 9 | 10 | 11 | 12 | |
|----|------|---------|---------|---------------|------------|--------------|------------------|--|----------|---------------------|----------|--|
| 1 | | | | | | | Пищевые продукты | Пищевые аллергены еще будут появляться | | | | |
| 2 | | | | | | | Белок яйца | | | Желток/куриное мясо | Коровье | |
| 3 | ШИФР | Пол | Возраст | Дата рождения | Дата сдачи | Город | nGal d 1 | nGal d 2 | nGal d 3 | nGal d 5 | nBos d 4 | |
| 4 | 1 | женский | 29 | 03.01.1984 | 03.10.2013 | Кириши | | | | | | |
| 5 | 2 | мужской | 9 | 08.01.2004 | 15.10.2013 | СПб | | | | | | |
| 6 | 3 | женский | 18 | 18.12.1995 | 06.05.2014 | СПб | | | | | | |
| 7 | 4 | мужской | 38 | 15.10.1975 | 10.12.2013 | Екатеринбург | | | | | | |
| 8 | 5 | женский | 6 | 04.07.2007 | 19.02.2014 | СПб | | | | | | |
| 9 | 6 | мужской | 5 | 01.11.2008 | 15.03.2014 | СПб | | | | | | |
| 10 | 7 | женский | 39 | 14.07.1974 | 27.03.2014 | Кириши | | | | | | |
| 11 | 8 | мужской | 21 | 06.02.1993 | 15.03.2014 | СПб | | | | | | |
| 12 | 9 | мужской | 3 | 26.09.2010 | 02.02.2014 | Екатеринбург | | | | | | |
| 13 | 10 | мужской | 46 | 06.10.1967 | 14.03.2014 | СПб | | | | | | |
| 14 | 11 | женский | 43 | | | СПб | | | | | | |
| 15 | 12 | мужской | 31 | 22.01.1983 | 25.01.2014 | СПб | | | | | | |
| 16 | 13 | мужской | 13 | 17.08.2000 | 24.03.2014 | СПб | | | | | | |
| 17 | 14 | мужской | 29 | 17.06.1984 | 22.11.2013 | СПб | | | | | | |
| 18 | 15 | мужской | 24 | 04.06.1989 | 10.12.2013 | СПб | | | | | | |
| 19 | 16 | женский | 43 | 26.12.1969 | 04.12.2013 | СПб | | | | | | |
| 20 | 17 | женский | 6 | 15.04.2007 | 20.10.2013 | СПб | | | | | | |
| 21 | 18 | мужской | 5 | 27.01.2009 | 04.05.2014 | Кириши | | | | | | |

Рисунок 2. Данные об аллергических реакциях

Исходные данные имеют сложную структуру, что затрудняет процесс исследования. Поэтому было принято решение разделить главную таблицу на несколько специализированных: люди, аллергены и реакции.

В первую таблицу запишем все данные о людях (ID, gender, age, birth_date, test_date, city). Таблица представлена на рисунке 3.

| | 1 | 2 | 3 | 4 | 5 | 6 |
|----|----|---------|-----|------------|------------|--------------|
| 1 | id | gender | age | birth_date | test_date | city |
| 2 | 1 | женский | 29 | 03.01.1984 | 03.10.2013 | Кириши |
| 3 | 2 | мужской | 9 | 08.01.2004 | 15.10.2013 | СПб |
| 4 | 3 | женский | 18 | 18.12.1995 | 06.05.2014 | СПб |
| 5 | 4 | мужской | 38 | 15.10.1975 | 10.12.2013 | Екатеринбург |
| 6 | 5 | женский | 6 | 04.07.2007 | 19.02.2014 | СПб |
| 7 | 6 | мужской | 5 | 01.11.2008 | 15.03.2014 | СПб |
| 8 | 7 | женский | 39 | 14.07.1974 | 27.03.2014 | Кириши |
| 9 | 8 | мужской | 21 | 06.02.1993 | 15.03.2014 | СПб |
| 10 | 9 | мужской | 3 | 26.09.2010 | 02.02.2014 | Екатеринбург |
| 11 | 10 | мужской | 46 | 06.10.1967 | 14.03.2014 | СПб |
| 12 | 11 | женский | 43 | | | СПб |
| 13 | 12 | мужской | 31 | 22.01.1983 | 25.01.2014 | СПб |
| 14 | 13 | мужской | 13 | 17.08.2000 | 24.03.2014 | СПб |
| 15 | 14 | мужской | 29 | 17.06.1984 | 22.11.2013 | СПб |
| 16 | 15 | мужской | 24 | 04.06.1989 | 10.12.2013 | СПб |
| 17 | 16 | женский | 43 | 26.12.1969 | 04.12.2013 | СПб |
| 18 | 17 | женский | 6 | 15.04.2007 | 20.10.2013 | СПб |
| 19 | 18 | мужской | 5 | 27.01.2009 | 01.05.2014 | Пушкин |
| 20 | 19 | женский | 6 | 14.11.2007 | 14.04.2014 | Конаково |
| 21 | 20 | женский | 8 | 28.11.2005 | 22.01.2014 | Москва |
| 22 | 21 | женский | 42 | 24.04.1971 | 14.12.2013 | Колпино |
| 23 | 22 | мужской | 10 | 11.04.2003 | 11.12.2013 | СПб |

Рисунок 3. Таблица people.csv

Во вторую таблицу запишем информацию об аллергенах (component, description, class, group). Таблица представлена на рисунке 4.

| | 1 | 2 | 3 | 4 |
|----|--------------------|-------------------------|-------------------------|-----------------|
| 1 | component | description | class | group |
| 2 | nGal d 1 | Белок яйца | Пищевые продукты | |
| 3 | nGal d 2 | Белок яйца | Пищевые продукты | |
| 4 | nGal d 3 | Белок яйца | Пищевые продукты | |
| 5 | nGal d 5 | Желток/куриное мясо | Пищевые продукты | |
| 6 | nBos d 4 | Коровье молоко | Пищевые продукты | |
| 7 | nBos d 5 | Коровье молоко | Пищевые продукты | |
| 8 | nBos d 8 | Коровье молоко | Пищевые продукты | |
| 9 | nBos d lactoferrin | Коровье молоко | Пищевые продукты | |
| 10 | rGad c 1 | Треска | Пищевые продукты | |
| 11 | nPen m 2 | Креветка | Пищевые продукты | |
| 12 | nPen m 4 | Креветка | Пищевые продукты | |
| 13 | rAna o 2 | Кешью | Пищевые продукты | |
| 14 | rBer e 1 | Бразильский орех | Пищевые продукты | |
| 15 | nCor a 9 | Фундук | Пищевые продукты | |
| 16 | rJug r 1 | Грецкий орех | Пищевые продукты | |
| 17 | nJug r 2 | Грецкий орех | Пищевые продукты | |
| 18 | nSes i 1 | Кунжут | Пищевые продукты | |
| 19 | rAra h 1 | Арахис | Пищевые продукты | |
| 20 | rAra h 2 | Арахис | Пищевые продукты | |
| 21 | rAra h 3 | Арахис | Пищевые продукты | |
| 22 | nAra h 6 | Арахис | Пищевые продукты | |
| 23 | nGly m 5 | Соевые бобы | Пищевые продукты | |
| 24 | nGly m 6 | Соевые бобы | Пищевые продукты | |
| 25 | nFag e 2 | Гречиха, гречневая мука | Пищевые продукты | |
| 26 | rTri a 19_0101 | Пшеница | Пищевые продукты | |
| 27 | nTri a aA_T1 | Пшеница | Пищевые продукты | |
| 28 | nAct d 1 | Киви | Пищевые продукты | |
| 29 | nAct d 5 | Киви | Пищевые продукты | |
| 30 | nCyn d 1 | Свиной | Ингаляционные аллергены | Пыльца растений |
| 31 | rPhi p 1 | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| 32 | rPhi p 2 | | Ингаляционные аллергены | Пыльца растений |

Рисунок 4. Таблица allergens.csv

В последней таблице запишем все реакции пациентов на аллергены (ID, component, rc_amount). Таблица представлена на рисунке 5.

| | 1 | 2 | 3 |
|----|----|--------------------|-----------|
| 1 | id | component | rc_amount |
| 2 | 1 | nGal d 1 | 0 |
| 3 | 1 | nGal d 2 | 0 |
| 4 | 1 | nGal d 3 | 0 |
| 5 | 1 | nGal d 5 | 0 |
| 6 | 1 | nBos d 4 | 0 |
| 7 | 1 | nBos d 5 | 0 |
| 8 | 1 | nBos d 8 | 0 |
| 9 | 1 | nBos d lactoferrin | 0 |
| 10 | 1 | rGad c 1 | 0 |
| 11 | 1 | nPen m 2 | 0 |
| 12 | 1 | nPen m 4 | 0 |
| 13 | 1 | rAna o 2 | 0 |
| 14 | 1 | rBer e 1 | 0 |
| 15 | 1 | nCor a 9 | 0 |
| 16 | 1 | rJug r 1 | 0 |
| 17 | 1 | nJug r 2 | 0 |
| 18 | 1 | nSes i 1 | 0 |
| 19 | 1 | rAra h 1 | 0 |
| 20 | 1 | rAra h 2 | 0 |
| 21 | 1 | rAra h 3 | 0 |

Рисунок 5. Таблица reactions.csv

Проведём анализ данных на языке программирования Python с помощью библиотеки pandas.

Загрузим все данные и объединим в одну таблицу (рисунок 6):

| | id | component | rc_amount | gender | age | birth_date | test_date | city | description | class | group |
|-------|------|-----------|-----------|---------|------|------------|------------|--------|-------------|-------------------------------------|----------------------------------|
| 0 | 1 | rPhi p 1 | 1,4 | женский | 29.0 | 03.01.1984 | 03.10.2013 | Кириши | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| 1 | 2 | rPhi p 1 | 1,3 | мужской | 9.0 | 08.01.2004 | 15.10.2013 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| 2 | 8 | rPhi p 1 | 8,2 | мужской | 21.0 | 06.02.1993 | 15.03.2014 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| 3 | 10 | rPhi p 1 | 0,9 | мужской | 46.0 | 06.10.1967 | 14.03.2014 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| 4 | 23 | rPhi p 1 | 10 | мужской | 24.0 | 01.08.1989 | 06.12.2013 | Кириши | Тимофеевка | Ингаляционные аллергены | Пыльца растений |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 11939 | 552 | rAni s 1 | 0,6 | женский | 7.0 | 30.10.2009 | 03.11.2016 | Москва | Анисакис | Яды насекомых (инсектные аллергены) | Другие главные аллергокомпоненты |
| 11940 | 1157 | rAni s 1 | 0,4 | женский | 39.0 | 17.05.1978 | 15.04.2018 | СПб | Анисакис | Яды насекомых (инсектные) | Другие главные |

Рисунок 6. Общая таблица

- 1) Для определения зависимости наличия аллергических реакций от места проживания сгруппируем реакции по городам (рисунок 7), при условии, что число реакций больше 10, и найдём число пациентов в каждом городе (рисунок 8).

| | city | component | reactions_count |
|-----|--------------|-----------|-----------------|
| 0 | Екатеринбург | nArt v 1 | 12 |
| 1 | Екатеринбург | rAln g 1 | 26 |
| 2 | Екатеринбург | rAra h 8 | 20 |
| 3 | Екатеринбург | rBet v 1 | 88 |
| 4 | Екатеринбург | rCan f 1 | 20 |
| ... | ... | ... | ... |
| 193 | Тюмень | rPhl p 1 | 15 |
| 194 | Тюмень | rPru p 1 | 22 |
| 195 | Уфа | rBet v 1 | 16 |
| 196 | Челябинск | rBet v 1 | 26 |
| 197 | Челябинск | rFel d 1 | 13 |

Рисунок 7. Группировка реакций по городам

| | city | patients_count |
|-----|--------------|----------------|
| 0 | Москва | 4924 |
| 1 | СПб | 2564 |
| 2 | Тюмень | 588 |
| 3 | Екатеринбург | 552 |
| 4 | Ставрополь | 523 |
| ... | ... | ... |
| 80 | Одинцово | 1 |
| 81 | Тихвин | 1 |
| 82 | Анапа | 1 |
| 83 | Вологда | 1 |
| 84 | Донское | 1 |

Рисунок 8. Число пациентов в городе

Теперь найдём процент реакций от общего числа пациентов и отсортируем по убыванию (рисунок 9).

| | city | component | percent |
|-----|----------------|---------------|-----------|
| 0 | Кириши | rBet v 1 | 19.745223 |
| 1 | Уфа | rBet v 1 | 17.582418 |
| 2 | Красногорск | rBet v 1 | 17.241379 |
| 3 | Екатеринбург | rBet v 1 | 15.942029 |
| 4 | Пушкин | rBet v 1 | 15.625000 |
| 5 | Сестрорецк | rBet v 1 | 15.533981 |
| 6 | Тюмень | rBet v 1 | 14.285714 |
| 7 | Зеленоград | rBet v 1 | 13.793103 |
| 8 | Коломна | rBet v 1 | 13.333333 |
| 9 | СПб | rBet v 1 | 13.260530 |
| 10 | Челябинск | rBet v 1 | 12.560386 |
| 11 | Москва | rBet v 1 | 12.103981 |
| 12 | Ростов-на-Дону | nAmb a 1 | 10.655738 |
| 13 | Тула | rBet v 1 | 10.526316 |
| 14 | Калининград | rBet v 1 | 10.084034 |
| 15 | Сочи | rBet v 1 | 9.929078 |
| 16 | Ставрополь | nAmb a 1 | 8.413002 |
| 17 | Тула | rFel d 1 | 6.666667 |
| 18 | Челябинск | rFel d 1 | 6.280193 |
| 19 | СПб | rFel d 1 | 6.279251 |
| 20 | Екатеринбург | rMal d 1 | 5.797101 |
| 21 | Екатеринбург | rCor a 1.0401 | 5.615942 |
| 22 | Екатеринбург | rFel d 1 | 5.434783 |
| 23 | СПб | rCor a 1.0401 | 5.109204 |
| 24 | Тюмень | rCor a 1.0401 | 5.102041 |
| ... | ... | ... | ... |

Рисунок 9. Процент реакции от числа пациентов (город)

По результатам вычислений можно сделать вывод, что реакция на аллерген «rBet v 1» (берёза) наблюдается у большого числа пациентов из разных городов. Кроме того, реакция на аллерген «nAmb a1» (амброзия) часто встречается в городах Ростов-на-Дону и Ставрополь, которые находятся в соседних регионах.

Найдём топ-5 самых часто встречаемых аллергенов по городам при условии, что количество реакций больше 10 (рисунок 10)

| | city | component | reactions_count | | | | |
|----|--------------|---------------|-----------------|----|----------------|---------------|-----|
| 0 | Екатеринбург | rBet v 1 | 88 | | | | |
| 1 | Екатеринбург | rMal d 1 | 32 | | | | |
| 2 | Екатеринбург | rCor a 1.0401 | 31 | | | | |
| 3 | Екатеринбург | rFel d 1 | 30 | | | | |
| 4 | Екатеринбург | rAln g 1 | 26 | | | | |
| 5 | Зеленоград | rBet v 1 | 12 | 14 | Москва | rAln g 1 | 205 |
| 6 | Калининград | rBet v 1 | 12 | 15 | Пушкин | rBet v 1 | 10 |
| 7 | Кириши | rBet v 1 | 31 | 16 | Ростов-на-Дону | nAmb a 1 | 13 |
| 8 | Коломна | rBet v 1 | 10 | 17 | СПб | rBet v 1 | 340 |
| 9 | Красногорск | rBet v 1 | 10 | 18 | СПб | rFel d 1 | 161 |
| 10 | Москва | rBet v 1 | 596 | 19 | СПб | rCor a 1.0401 | 131 |
| 11 | Москва | rCor a 1.0401 | 244 | 20 | СПб | rMal d 1 | 130 |
| 12 | Москва | rMal d 1 | 229 | 21 | СПб | rAln g 1 | 105 |
| 13 | Москва | rFel d 1 | 220 | 22 | Сестрорецк | rBet v 1 | 16 |

Рисунок 10. Наиболее частые реакции по городам

Данная таблица позволяет определить аллергены, встречающиеся в каком-то конкретном городе. Так, например, в крупных городах (Москва, Екатеринбург, Санкт-Петербург) наблюдается один и тот же перечень аллергенов.

- Для определения зависимости наличия аллергических реакций от возраста выделим и соотнесём с пациентами следующие возрастные группы: [0, 18), [18, 44), [44, 69), [69, 74), [74, 90). Результат представлен на рисунке 11.

| id | component | rc_amount | gender | age | birth_date | test_date | city | description | class | group | age_group |
|----|-----------|-----------|---------|------|------------|------------|--------|-------------|-------------------------|-----------------|-----------|
| 1 | rPhl p 1 | 1,4 | женский | 29.0 | 03.01.1984 | 03.10.2013 | Кириши | Тимофеевка | Ингаляционные аллергены | Пыльца растений | [18, 44) |
| 2 | rPhl p 1 | 1,3 | мужской | 9.0 | 08.01.2004 | 15.10.2013 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений | [0, 18) |
| 8 | rPhl p 1 | 8,2 | мужской | 21.0 | 06.02.1993 | 15.03.2014 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений | [18, 44) |
| 10 | rPhl p 1 | 0,9 | мужской | 46.0 | 06.10.1967 | 14.03.2014 | СПб | Тимофеевка | Ингаляционные аллергены | Пыльца растений | [44, 69) |
| 23 | rPhl p 1 | 10 | мужской | 24.0 | 01.08.1989 | 06.12.2013 | Кириши | Тимофеевка | Ингаляционные аллергены | Пыльца растений | [18, 44) |

Рисунок 11. Главная таблица с добавлением возрастных групп

Аналогично с первым пунктом, сгруппируем реакции по возрастной группе (рисунок 12) и рассчитаем число пациентов в каждой группе (рисунок 13).

| age_group | component | reactions_count |
|-----------|------------------|-----------------|
| 0 | [0, 18) nAct d 1 | 33 |
| 1 | [0, 18) nAct d 2 | 30 |
| 2 | [0, 18) nAmb a 1 | 85 |
| 3 | [0, 18) nAra h 6 | 38 |
| 4 | [0, 18) nArt v 1 | 145 |

Рисунок 12. Группировка реакций по возрастной группе

| age_group | patients_count |
|-----------|----------------|
| 0 | [0, 18) 8267 |
| 1 | [18, 44) 3015 |
| 2 | [44, 69) 616 |
| 3 | [74, 90) 22 |
| 4 | [69, 74) 7 |

Рисунок 13. Число пациентов в возрастной группе

Найдём процент реакций от общего числа пациентов и отсортируем по убыванию (рисунок 14).

| | age_group | component | percent |
|---|-----------|---------------|-----------|
| 0 | [44, 69) | rBet v 1 | 12.987013 |
| 1 | [18, 44) | rBet v 1 | 12.669983 |
| 2 | [0, 18) | rBet v 1 | 12.072094 |
| 3 | [44, 69) | rFel d 1 | 5.357143 |
| 4 | [0, 18) | rFel d 1 | 5.346559 |
| 5 | [44, 69) | rPhl p 1 | 5.032468 |
| 6 | [0, 18) | rCor a 1.0401 | 4.971574 |
| 7 | [18, 44) | rFel d 1 | 4.875622 |
| 8 | [44, 69) | rMal d 1 | 4.870130 |

Рисунок 14. Процент реакций от числа пациентов (возраст)

По таблице видно, что реакций на аллерген «rBet v 1» (берёза) является самой частой и в одинаковой степени наблюдается в каждой из основных возрастных групп.

Теперь определим топ-5 самых частых реакций для каждой возрастной группы (рисунок 15).

| | age_group | component | reactions_count |
|----|-----------|---------------|-----------------|
| 0 | [0, 18) | rBet v 1 | 998 |
| 1 | [0, 18) | rFel d 1 | 442 |
| 2 | [0, 18) | rCor a 1.0401 | 411 |
| 3 | [0, 18) | rMal d 1 | 380 |
| 4 | [0, 18) | rAln g 1 | 331 |
| 5 | [18, 44) | rBet v 1 | 382 |
| 6 | [18, 44) | rFel d 1 | 147 |
| 7 | [18, 44) | rMal d 1 | 144 |
| 8 | [18, 44) | rPhl p 1 | 138 |
| 9 | [18, 44) | rCor a 1.0401 | 137 |
| 10 | [44, 69) | rBet v 1 | 80 |
| 11 | [44, 69) | rFel d 1 | 33 |
| 12 | [44, 69) | rPhl p 1 | 31 |
| 13 | [44, 69) | rMal d 1 | 30 |

Рисунок 15. Наиболее частые реакции по возрастным группам

Можно увидеть, что для возрастной группы [0, 18) аллерген «rPhl p 1» (Тимофеевка) не входит в пятёрку наиболее частых, в отличие от групп [18, 44) и [44, 69).

3.2. Создание обучающего датасета

Для того, чтобы использовать имеющиеся данные в обучении вариационного автоэнкодера, необходимо привести их к определённому виду. В нашем случае избавимся от текстовых данных, заменив их на числовую интерпретацию.

Во-первых, каждому городу поставим в соответствие ID города (рисунок 16).

```
cities = pd.read_csv('cities.csv', sep=',')
cities['city'].unique()
unique_city = pd.DataFrame(cities['city'].unique()).reset_index().rename(columns={'index':'city_id', 0:'city'})
display(unique_city)
unique_city.to_csv('unique_city.csv', sep=";", encoding='utf-8-sig', index=False)
```

| | city_id | city |
|-----|---------|--------------|
| 0 | 0 | Сестрорецк |
| 1 | 1 | Екатеринбург |
| 2 | 2 | Москва |
| 3 | 3 | Саратов |
| 4 | 4 | СПб |
| ... | ... | ... |
| 80 | 80 | Воронеж |
| 81 | 81 | Новомосковск |
| 82 | 82 | Красноярск |
| 83 | 83 | Симферополь |
| 84 | 84 | Шахты |

85 rows x 2 columns

Рисунок 16. Создание ID городов

Во-вторых, обозначим пол как 0 (пациент – мужчина) и 1 (пациент – женщина). Объединив эти изменения, получим итоговый датасет (рисунок 17).

```
row_data = pd.read_csv('row_data.csv', sep=',')
row_data = row_data.replace(',', '.', regex=True)
row_data = row_data.fillna(0)
row_data = row_data.merge(unique_city, how='inner')
cityID = row_data.pop('city_id')
row_data.insert(2, 'city_id', cityID)
row_data = row_data.drop('city', 1)
row_data['gender'] = row_data['gender'].apply(lambda x: 0 if x == 'мужской' else 1)
row_data = row_data.astype(np.float64)
display(row_data)
row_data.to_csv('train_data.csv', sep=',', encoding='utf-8-sig', index=False)
```

| | gender | age | city_id | nGal d1 | nGal d2 | nGal d3 | nGal d5 | nBos d4 | nBos d5 | nBos d8 | ... | rCyp c1 | rPen a1 | nPen l1 | rAni s 3.1 | rDat c1 | rMer a 1.1 | nSal k 1.1 | rEur m2 | rChe a 1.1 | rPla l 1.1 |
|------|--------|------|---------|---------|---------|---------|---------|---------|---------|---------|-----|---------|---------|---------|------------|---------|------------|------------|---------|------------|------------|
| 0 | 1.0 | 29.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 1.0 | 39.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 24.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 28.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 1.0 | 38.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1155 | 0.0 | 5.0 | 74.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1156 | 1.0 | 1.0 | 74.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 0.7 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1157 | 1.0 | 7.0 | 66.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1158 | 0.0 | 5.0 | 75.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1159 | 0.0 | 5.0 | 76.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

1160 rows x 127 columns

Рисунок 17. Обучающий датасет

3.3. Программная реализации вариационного автоэнкодера

Программа написана на языке программирования Python с использованием специализированных библиотек *tensorflow*, *numpy* и *pandas*.

Создание вариационного автоэнкодера подразумевает под собой описание структуры класса VAE, включающего функционал, необходимый для инициализации параметров нейронной сети, её обучения и тестирования.

Для начала построим модели кодера и декодера. Их программный вид и структура представлены на рисунках 18 и 19 соответственно.

```
def build_encoder(self):
    encoder_inputs = tf.keras.layers.Input(shape=(self.data_dim,))
    x = tf.keras.layers.Dense(units=self.hidden_layer_dim, activation="relu")(encoder_inputs)
    z_mean = tf.keras.layers.Dense(units=self.latent_dim)(x)
    z_log_var = tf.keras.layers.Dense(units=self.latent_dim)(x)

    return tf.keras.Model(encoder_inputs, [z_mean, z_log_var], name="Encoder")

def build_decoder(self):
    decoder_inputs = tf.keras.layers.Input(shape=(self.latent_dim,))
    x = tf.keras.layers.Dense(units=self.hidden_layer_dim, activation="relu")(decoder_inputs)
    decoder_outputs = tf.keras.layers.Dense(units=self.data_dim, activation="tanh")(x)

    return tf.keras.Model(decoder_inputs, decoder_outputs, name="Decoder")
```

Рисунок 18. Создание кодера и декодера

```
Model: "Encoder"
-----
Layer (type)                Output Shape         Param #   Connected to
-----
input_1 (InputLayer)        [(None, 127)]        0
-----
dense (Dense)                (None, 64)           8192      input_1[0][0]
-----
dense_1 (Dense)              (None, 2)            130       dense[0][0]
-----
dense_2 (Dense)              (None, 2)            130       dense[0][0]
-----
Total params: 8,452
Trainable params: 8,452
Non-trainable params: 0
-----
Model: "Decoder"
-----
Layer (type)                Output Shape         Param #
-----
input_2 (InputLayer)        [(None, 2)]          0
-----
dense_3 (Dense)              (None, 64)           192
-----
dense_4 (Dense)              (None, 127)          8255
-----
Total params: 8,447
Trainable params: 8,447
Non-trainable params: 0
-----
```

Рисунок 19. Структура кодера и декодера

Функция, отвечающая за «reparametrization trick», изображена на рисунке 20.

```
def sampling(self, inputs):
    z_mean, z_log_var = inputs
    batch = tf.shape(z_mean)[0]
    dim = tf.shape(z_mean)[1]
    epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
    z = z_mean + tf.exp(0.5 * z_log_var) * epsilon
    return z_mean, z_log_var, z
```

Рисунок 20. Reparametrization trick

Шаг обучения, обозначенный на рисунке 21, включает в себя подсчёт ошибок и оптимизацию весов.

```
@tf.function
def train_step(self, data):
    with tf.GradientTape() as tape:
        z_mean, z_log_var, z = self.sampling(self.encoder(data))
        reconstruction = self.decoder(z)
        reconstruction_loss = tf.reduce_mean(
            tf.reduce_sum(
                tf.keras.losses.binary_crossentropy(data, reconstruction)
            )
        )
        kl_loss = -0.5 * (1 + z_log_var - tf.square(z_mean) - tf.exp(z_log_var))
        kl_loss = tf.reduce_mean(tf.reduce_sum(kl_loss, axis=1))
        total_loss = reconstruction_loss + kl_loss
    grads = tape.gradient(total_loss, self.trainable_weights)
    self.optimizer.apply_gradients(zip(grads, self.trainable_weights))
    self.total_loss_tracker.update_state(total_loss)
    self.reconstruction_loss_tracker.update_state(reconstruction_loss)
    self.kl_loss_tracker.update_state(kl_loss)
```

Рисунок 21. Функция шага обучения

Кроме того, в программе имеются функции, предназначенные для нормализации входных данных (нормализация средним) и обработки сгенерированных данных для приведения их к нужному виду. Обе эти функции представлены на рисунке 22.

```

def preprocess(samples):
    for column in samples.columns:
        mean = samples[column].mean()
        std = samples[column].std()

        samples[column] = ((samples[column] - mean) / (std + 1e-10))

        normalization_data.append([mean, std])

    return samples.astype('float32')

def recover_data(data_list):
    recovered_data_list = []
    for data in data_list:
        data = data.numpy()
        result = []

        for item, prop in zip(data[0], normalization_data):
            item = item * prop[1] + prop[0]
            if item < 0:
                item = 0.
            item = round(item, 1)
            result.append(item)
        result[0] = 0 if result[0] < 0.5 else 1
        result[1] = int(result[1] + 0.5)
        result[2] = int(result[2] + 0.5)
        recovered_data_list.append(result)
    return np.array(recovered_data_list)

```

Рисунок 22. Функция нормализации входных данных и функция обработки сгенерированных данных

3.4. Обучение VAE и анализ результатов

Время обучения составляет 50 эпох. Процесс обучения представлен на рисунке 23.

```

Epoch: 1, loss: 799.8038330078125, time elapse for current epoch: 6.597399473190308
Epoch: 2, loss: 669.7175903320312, time elapse for current epoch: 0.21570038795471191
Epoch: 3, loss: 556.9549560546875, time elapse for current epoch: 0.22620534896850586
Epoch: 4, loss: 471.73162841796875, time elapse for current epoch: 0.22920799255371094
Epoch: 5, loss: 409.1576843261719, time elapse for current epoch: 0.22620511054992676
Epoch: 6, loss: 361.3291931152344, time elapse for current epoch: 0.21419453620910645
Epoch: 7, loss: 322.9765930175781, time elapse for current epoch: 0.2402193546295166
Epoch: 8, loss: 292.0148620605469, time elapse for current epoch: 0.21770215034484863
Epoch: 9, loss: 266.5415344238281, time elapse for current epoch: 0.21619701385498047
Epoch: 10, loss: 245.1831512451172, time elapse for current epoch: 0.22920823097229004

```

Рисунок 23. Процесс обучения VAE

График изменения ошибки изображён на рисунке 24.

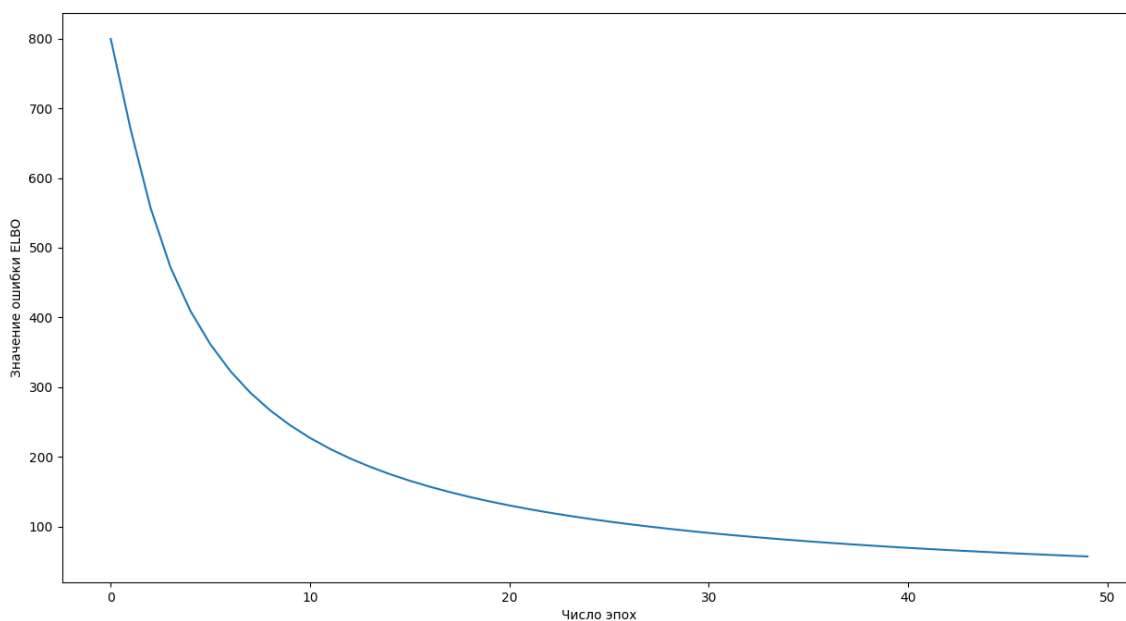


Рисунок 24. График зависимости ошибки ELBO от времени обучения

Теперь рассмотрим сгенерированные данные в количестве 10 штук (рисунок 25).

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|----|--------|-----|---------|----------|----------|----------|----------|----------|----------|----------|------------|----------|----------|----------|------|
| 1 | gender | age | city_id | nGal d 1 | nGal d 2 | nGal d 3 | nGal d 5 | nBos d 4 | nBos d 5 | nBos d 8 | nBos d lac | rGad c 1 | nPen m 2 | nPen m 4 | rAna |
| 2 | 1 | 16 | 8 | 0 | 0,1 | 0,2 | 0 | 0,1 | 0 | 0,3 | 0 | 1,3 | 0 | 0 | |
| 3 | 1 | 13 | 6 | 0 | 0 | 0,5 | 0 | 0 | 0 | 0,2 | 0 | 1,3 | 0 | 0 | |
| 4 | 0 | 16 | 13 | 0 | 0 | 0,1 | 0 | 0,2 | 0 | 0,1 | 0 | 0,2 | 0 | 0 | |
| 5 | 0 | 16 | 12 | 0,2 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,7 | 0 | 0 | |
| 6 | 1 | 15 | 6 | 0 | 0,1 | 0,2 | 0 | 0 | 0 | 0,3 | 0 | 1,4 | 0 | 0 | |
| 7 | 1 | 17 | 10 | 0,2 | 0,1 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,6 | 0 | 0 | |
| 8 | 1 | 17 | 11 | 0,2 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,4 | 0 | 0 | |
| 9 | 1 | 15 | 9 | 0,3 | 0 | 0,1 | 0 | 0 | 0,1 | 0,1 | 0 | 1,1 | 0 | 0 | |
| 10 | 1 | 15 | 7 | 0 | 0,1 | 0,2 | 0 | 0 | 0 | 0,3 | 0 | 1,4 | 0 | 0 | |
| 11 | 1 | 16 | 10 | 0,2 | 0,1 | 0,2 | 0,1 | 0,2 | 0,1 | 0,2 | 0 | 1 | 0 | 0 | |

Рисунок 25. Сгенерированные данные

Видно, что все сгенерированные пациенты принадлежат к возрастной группе $[0, 18)$, что неудивительно, так как эта возрастная группа в тренировочных данных значительно превосходит все остальные. Посмотрим на значения реакций самых часто встречаемых аллергенов для данной возрастной группы. Как было выяснено раньше, это следующие аллергены: rBet v 1, rFel d 1, rCor 1.0401, rMal d 1, rAln g 1. Выпишем сгенерированные данные по этим аллергенам (рисунок 26).

| AMT | BD | CA | CT | CV |
|------------|------------|---------------|----------|---------|
| rBet v 1 r | rFel d 1 r | rCor a 1.0401 | rMal d 1 | rAlng 1 |
| 10,6 | 5,3 | 1 | 3,3 | 2,3 |
| 12 | 3,6 | 0 | 3,5 | 2,3 |
| 8,9 | 4,9 | 4,2 | 1,5 | 4,8 |
| 10,4 | 5,1 | 4 | 3,1 | 3,4 |
| 10,3 | 5,3 | 0 | 2,9 | 2,3 |
| 9,9 | 5,8 | 3 | 4,1 | 2,6 |
| 9,6 | 5,8 | 3,1 | 3,9 | 3,1 |
| 11,8 | 4,6 | 2,5 | 3,9 | 2,4 |
| 10,4 | 5,3 | 0 | 3 | 2,3 |
| 11,3 | 5,1 | 2,7 | 4,1 | 2,5 |

Рисунок 26. Значения ключевых реакций

Реакции присущи сгенерированным пациентам и их значения достаточно велики.

4. РАЗРАБОТКА ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ СЕТИ

4.1. Обзор входных данных

В качестве исходных данных имеются чёрно-белые снимки сосудов 6 пациентов размером 1000x1000 пикселей в количестве 286 штук. Некоторые из них представлены на рисунке 27.



Рисунок 27. Снимки пациентов

4.2. Программная реализация генеративно-сопоставительной сети

Программа написана на языке программирования Python с использованием специализированных библиотек *tensorflow* и *numpy*.

Создание генеративно-сопоставительной сети подразумевает под собой описание структуры класса WGAN-GP, включающего функционал, необходимый для инициализации параметров нейронной сети, её обучения и тестирования.

Помимо самой сети, в программе присутствует функционал для предварительной обработки изображений с целью создания датасета, пригодного для обучения.

Модели генератора и дискриминатора представлены в виде совокупности слоёв нейросетей (методов библиотеки *tensorflow*). Структуры генератора и дискриминатора представлены на рисунке 28.

| Model: "Generator" | | | Model: "Discriminator" | | |
|---|----------------------|---------|-----------------------------|-----------------------|---------|
| Layer (type) | Output Shape | Param # | Layer (type) | Output Shape | Param # |
| input_1 (InputLayer) | [(None, 128)] | 0 | input_2 (InputLayer) | [(None, 512, 512, 3)] | 0 |
| flatten (Flatten) | (None, 128) | 0 | conv2d (Conv2D) | (None, 512, 512, 8) | 600 |
| dense (Dense) | (None, 32768) | 4227072 | leaky_re_lu_7 (LeakyReLU) | (None, 512, 512, 8) | 0 |
| batch_normalization (Batch Normalization) | (None, 32768) | 131072 | dropout (Dropout) | (None, 512, 512, 8) | 0 |
| leaky_re_lu (LeakyReLU) | (None, 32768) | 0 | conv2d_1 (Conv2D) | (None, 256, 256, 16) | 3200 |
| tf.reshape (TFOPLambda) | (None, 8, 8, 512) | 0 | leaky_re_lu_8 (LeakyReLU) | (None, 256, 256, 16) | 0 |
| conv2d_transpose (Conv2DTranspose) | (None, 16, 16, 256) | 3276800 | dropout_1 (Dropout) | (None, 256, 256, 16) | 0 |
| batch_normalization_1 (Batch Normalization) | (None, 16, 16, 256) | 1024 | conv2d_2 (Conv2D) | (None, 128, 128, 32) | 12800 |
| leaky_re_lu_1 (LeakyReLU) | (None, 16, 16, 256) | 0 | leaky_re_lu_9 (LeakyReLU) | (None, 128, 128, 32) | 0 |
| conv2d_transpose_1 (Conv2DTranspose) | (None, 32, 32, 128) | 819200 | dropout_2 (Dropout) | (None, 128, 128, 32) | 0 |
| batch_normalization_2 (Batch Normalization) | (None, 32, 32, 128) | 512 | conv2d_3 (Conv2D) | (None, 64, 64, 64) | 51200 |
| leaky_re_lu_2 (LeakyReLU) | (None, 32, 32, 128) | 0 | leaky_re_lu_10 (LeakyReLU) | (None, 64, 64, 64) | 0 |
| conv2d_transpose_2 (Conv2DTranspose) | (None, 64, 64, 64) | 204800 | dropout_3 (Dropout) | (None, 64, 64, 64) | 0 |
| batch_normalization_3 (Batch Normalization) | (None, 64, 64, 64) | 256 | conv2d_4 (Conv2D) | (None, 32, 32, 128) | 204800 |
| leaky_re_lu_3 (LeakyReLU) | (None, 64, 64, 64) | 0 | leaky_re_lu_11 (LeakyReLU) | (None, 32, 32, 128) | 0 |
| conv2d_transpose_3 (Conv2DTranspose) | (None, 128, 128, 32) | 51200 | dropout_4 (Dropout) | (None, 32, 32, 128) | 0 |
| batch_normalization_4 (Batch Normalization) | (None, 128, 128, 32) | 128 | conv2d_5 (Conv2D) | (None, 16, 16, 256) | 819200 |
| leaky_re_lu_4 (LeakyReLU) | (None, 128, 128, 32) | 0 | leaky_re_lu_12 (LeakyReLU) | (None, 16, 16, 256) | 0 |
| conv2d_transpose_4 (Conv2DTranspose) | (None, 256, 256, 16) | 12800 | dropout_5 (Dropout) | (None, 16, 16, 256) | 0 |
| batch_normalization_5 (Batch Normalization) | (None, 256, 256, 16) | 64 | conv2d_6 (Conv2D) | (None, 8, 8, 512) | 3276800 |
| leaky_re_lu_5 (LeakyReLU) | (None, 256, 256, 16) | 0 | leaky_re_lu_13 (LeakyReLU) | (None, 8, 8, 512) | 0 |
| conv2d_transpose_5 (Conv2DTranspose) | (None, 512, 512, 8) | 3200 | dropout_6 (Dropout) | (None, 8, 8, 512) | 0 |
| batch_normalization_6 (Batch Normalization) | (None, 512, 512, 8) | 32 | flatten_1 (Flatten) | (None, 32768) | 0 |
| leaky_re_lu_6 (LeakyReLU) | (None, 512, 512, 8) | 0 | dense_1 (Dense) | (None, 1) | 32769 |
| conv2d_transpose_6 (Conv2DTranspose) | (None, 512, 512, 3) | 600 | | | |
| tf.math.tanh (TFOPLambda) | (None, 512, 512, 3) | 0 | | | |
| Total params: 8,728,760 | | | Total params: 4,401,369 | | |
| Trainable params: 8,662,216 | | | Trainable params: 4,401,369 | | |
| Non-trainable params: 66,544 | | | Non-trainable params: 0 | | |

Рисунок 28. Вид модели генератора и дискриминатора

Функции шага обучения для генератора и дискриминатора прописаны отдельно с учётом вычисления ошибки и применением расстояния Вассерштейна и градиентного штрафа. Далее они объединяются в одну функцию, через вызов которой и реализуется последовательное обучение сети.

Помимо основных методов, в программе имеются вспомогательные инструменты для сохранения значений параметров (добавляет возможность продолжить обучение модели в любой момент), сохранения изображений по ходу обучения, построения графика ошибок.

4.3. Обучение WGAN и анализ результатов

Было проведено 850 эпох обучения. Изображения, полученные по ходу обучения, представлены на рисунках 29-32.

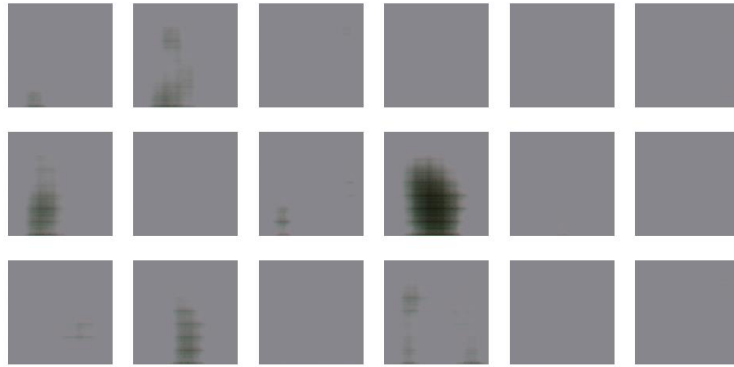


Рисунок 29. Эпоха №10

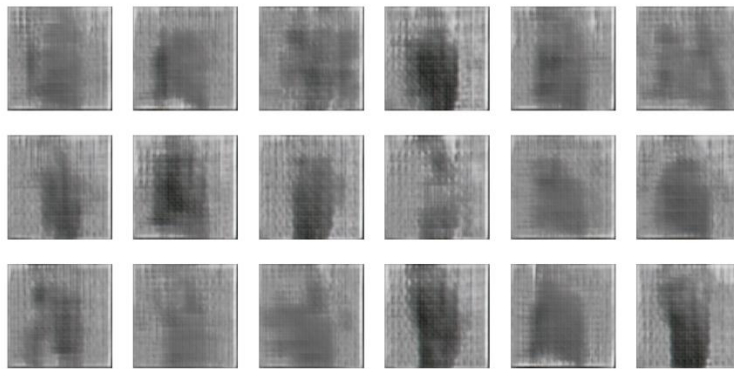


Рисунок 30. Эпоха №50

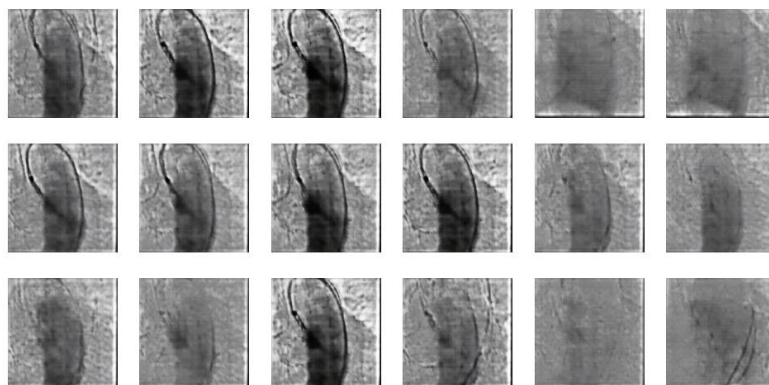


Рисунок 31. Эпоха №200

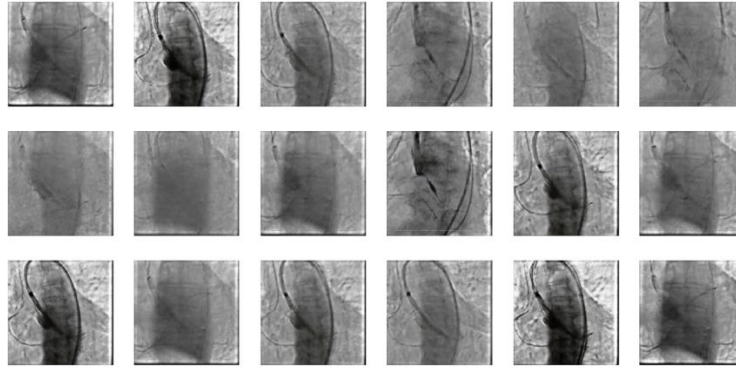


Рисунок 32. Эпоха №850

Изображения в высоком разрешении, полученные в результате тестовой генерации (рисунок 33):

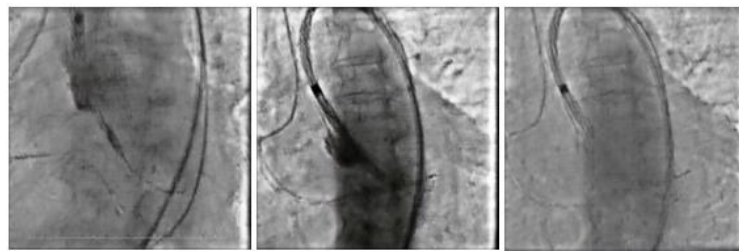


Рисунок 33. Тестовые изображения

Сохранённые логи указаны на рисунке 34.

```
Файл  Правка  Формат  Вид  Справка
Epoch 834 | ET 51.87 min | Gen Loss: 136.521179, Disc Loss: -31.264023 Adv Loss: -46.204819
Epoch 835 | ET 52.49 min | Gen Loss: -42.352100, Disc Loss: -25.890335 Adv Loss: -34.768166
Epoch 836 | ET 53.10 min | Gen Loss: -11.066015, Disc Loss: -21.740564 Adv Loss: -30.113920
Epoch 837 | ET 53.72 min | Gen Loss: 38.461430, Disc Loss: -18.728235 Adv Loss: -25.727467
Epoch 838 | ET 54.33 min | Gen Loss: -38.048695, Disc Loss: -23.240896 Adv Loss: -33.564316
Epoch 839 | ET 54.94 min | Gen Loss: -42.745045, Disc Loss: -28.266018 Adv Loss: -41.570370
Epoch 840 | ET 55.56 min | Gen Loss: 83.939323, Disc Loss: -21.676744 Adv Loss: -31.433281
Epoch 841 | ET 56.20 min | Gen Loss: -35.846188, Disc Loss: -23.004381 Adv Loss: -32.620644
Epoch 842 | ET 56.81 min | Gen Loss: -22.385500, Disc Loss: -24.315603 Adv Loss: -33.166584
Epoch 843 | ET 57.43 min | Gen Loss: 72.876366, Disc Loss: -22.194214 Adv Loss: -31.664621
Epoch 844 | ET 58.04 min | Gen Loss: -41.634529, Disc Loss: -28.495020 Adv Loss: -39.751842
Epoch 845 | ET 58.66 min | Gen Loss: 20.677338, Disc Loss: -26.163309 Adv Loss: -37.525093
Epoch 846 | ET 59.27 min | Gen Loss: 11.303022, Disc Loss: -21.758667 Adv Loss: -30.101496
Epoch 847 | ET 59.89 min | Gen Loss: -34.915672, Disc Loss: -26.373028 Adv Loss: -35.660339
Epoch 848 | ET 60.51 min | Gen Loss: 48.222569, Disc Loss: -19.861261 Adv Loss: -27.889763
Epoch 849 | ET 61.12 min | Gen Loss: 5.621695, Disc Loss: -23.426270 Adv Loss: -32.009369
Epoch 850 | ET 61.74 min | Gen Loss: -17.700726, Disc Loss: -21.771738 Adv Loss: -29.623301
```

Рисунок 34. Логи обучения

График потерь на протяжении всего процесса обучения изображён на рисунке 35.

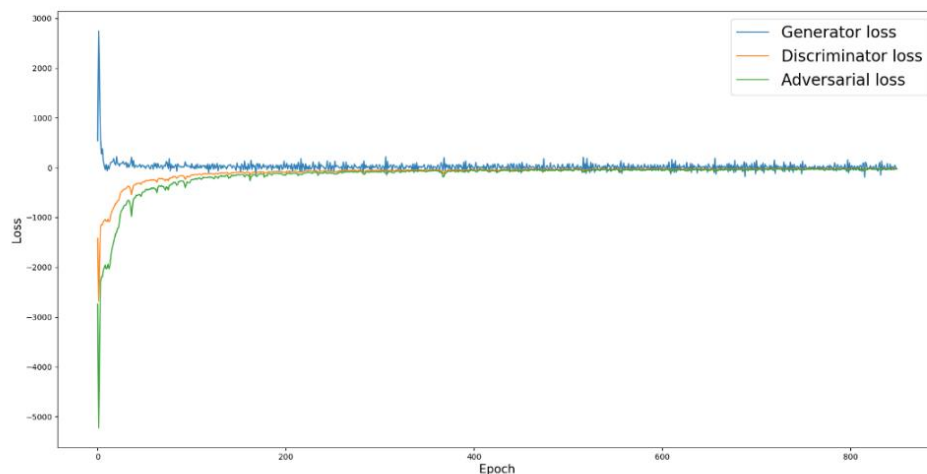


Рисунок 35. Ошибки всего обучения

График потерь с 200 эпохи по 850 эпоху представлен на рисунке 36.

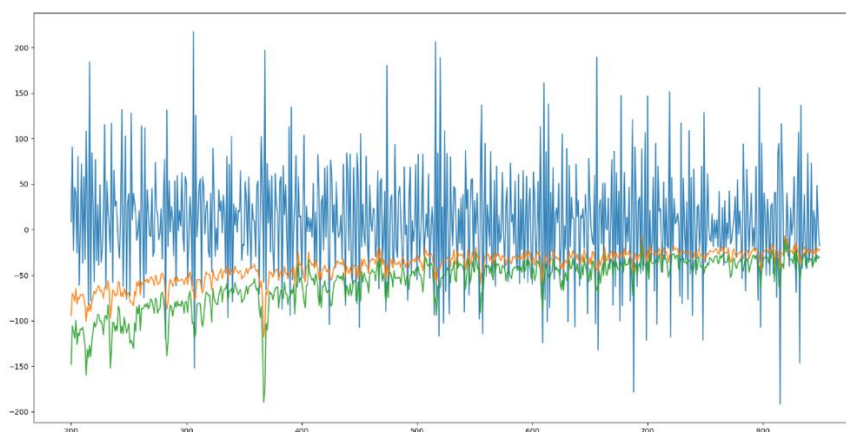


Рисунок 36. Ошибки с 200 по 850 эпохи

Как видно из графиков, если в самом начале обучения можно сказать о хорошей сходимости ошибок, то на поздних этапах генератор демонстрирует сильный разброс значений ошибки. Это говорит о том, что модель генератора является не оптимальной для данной задачи, хотя полученные результаты достаточно близки к реальным данным.

Любая научно-исследовательская работа требует своего экономического обоснования с целью продемонстрировать количественное и качественное доказательство целесообразности разработки, а также определения организационно-экономических условий ее эффективного функционирования.

В рамках данной выпускной квалификационной работы выполняется разработка программного обеспечения для генерации новых медицинских данных с использованием алгоритмов нейронных сетей.

Целью данного раздела является расчёт технико-экономических показателей для последующей оценки с точки зрения ресурсоэффективности и ресурсосбережения проводимой работы.

Для достижения поставленной цели будут выполнены следующие задачи:

- Оценка конкурентоспособности технических решений;
- Проведение SWOT-анализа для выявления сильных и слабых сторон проекта;
- Планирование проведения работ с построением диаграммы Ганта;
- Расчёт бюджета проекта;
- Оценка ресурсной, финансовой и экономической эффективности.

5.1. Оценка коммерческого потенциала и перспективности научного исследования с позиции ресурсоэффективности и ресурсосбережения

5.1.1. Анализ конкурентных технических решений

Анализ конкурентных технических решений позволяет провести сравнение между существующими решениями с целью определения наиболее эффективного варианта. Для этой цели следует определить набор критериев,

по которым будет проходить оценка. Для каждого критерия экспертным путём определяется вес; в сумме веса должны составлять 1 (100%). Положения по каждому критерию оцениваются по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная.

Анализ конкурентных технических решений определяется по формуле

$$K = \sum V_i \cdot B_i$$

где K – конкурентоспособность научной разработки или конкурента; V_i – вес показателя (в долях единицы); B_i – балл i-го показателя.

Конкурентные решения возникли при выборе подходящего алгоритма нейронной сети для генерации изображений. Имеется три конкурентных алгоритма: K – классический GAN; C – свёрточный глубокий GAN (DCGAN); B – GAN Вассерштейна с градиентным штрафом (WGAN-GP).

В качестве критериев оценки были выбраны следующие показатели: простота алгоритма, простота реализации, гибкость параметров обучения, эффективность, удобство в эксплуатации, цена, необходимая мощность оборудования, предполагаемый срок эксплуатации.

Оценочная карта для сравнения конкурентных технических решений представлена в таблице 1.

Таблица 1 – Оценочная карта для сравнения конкурентных технических решений

| Критерии оценки | Вес | Баллы | | | Конкурентоспособность | | |
|---|------|-----------|-----------|-----------|-----------------------|----------------|----------------|
| | | K | C | B | K _K | K _C | K _B |
| Технические критерии оценки ресурсоэффективности | | | | | | | |
| 1. Простота алгоритма | 0,2 | 4 | 3 | 2 | 0,8 | 0,6 | 0,4 |
| 2. Простота реализации | 0,2 | 4 | 4 | 3 | 0,8 | 0,8 | 0,6 |
| 3. Гибкость параметров обучения | 0,05 | 1 | 3 | 4 | 0,05 | 0,15 | 0,2 |
| 4. Эффективность | 0,25 | 1 | 3 | 5 | 0,25 | 0,75 | 1,25 |
| 5. Удобство в эксплуатации | 0,05 | 3 | 3 | 3 | 0,15 | 0,15 | 0,15 |
| Экономические критерии оценки эффективности | | | | | | | |
| 1. Цена | 0,1 | 4 | 3 | 3 | 0,4 | 0,3 | 0,3 |
| 2. Необходимая мощность оборудования | 0,1 | 4 | 3 | 3 | 0,4 | 0,3 | 0,3 |
| 3. Предполагаемый срок эксплуатации | 0,05 | 3 | 4 | 4 | 0,15 | 0,2 | 0,2 |
| Итого | 1 | 24 | 26 | 27 | 3 | 3,25 | 3,4 |

Согласно оценочной карте, наиболее целесообразным и конкурентоспособным является применение алгоритма WGAN-GP с показателем конкурентоспособности 3,4 условных единиц. Хотя данный алгоритм сложнее конкурентов с точки зрения теории и реализации, он даёт высокую эффективность, что является самым значимым критерием оценки.

5.2. SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. С его помощью можно описать сильные и слабые стороны НИИ, выявить возможности и угрозы реализации работы, которые проявляются или могут появиться во внешней среде.

SWOT-анализ представлен в таблице 2.

Таблица 2 – SWOT-анализ проекта

| | | |
|--|--|---|
| | <p>Сильные стороны проекта:</p> <ol style="list-style-type: none"> 1. Работа реализована с использованием самой популярной библиотеки для нейронных сетей TensorFlow. 2. Решает задачу нехватки данных. 3. Простота применения для конечного пользователя. | <p>Слабые стороны проекта:</p> <ol style="list-style-type: none"> 1. Требуется много времени на обучение. 2. Пользователь не может менять параметры сети. 3. Узкая направленность генерации данных. |
| <p>Возможности проекта:</p> <ol style="list-style-type: none"> 1. Внедрение улучшений при необходимости. 2. Переобучение для генерации других данных. 3. Применение в других задачах машинного обучения. | <p>Использование результатов работы позволит предоставить решение задачи расширения выборки данных с использованием современных технологий.</p> | <p>Настройка алгоритма и целевой задачи генерации потенциально позволит устранить недостатки и применять разработку во многих сферах деятельности.</p> |
| <p>Угрозы проекта:</p> <ol style="list-style-type: none"> 1. Появление более эффективных алгоритмов. 2. Отсутствие спроса. | <p>Актуальность применяемых инструментов позволит доработать программу в соответствии с трендами, а качество результатов делает её одним из лучших решений на рынке.</p> | <p>Для минимизации угроз следует расширить область применения и следить за развитием используемых технологий.</p> |

В ходе анализа было установлено, что слабые стороны и угрозы проекта в большей степени нивелируются его сильными сторонами и возможностями, что делает проект весьма перспективным.

Наиболее значимой сильной стороной является то, что разработка решает проблему нехватки данных – одну из самых важных в современном машинном обучении. Главной возможностью выступает дальнейшее применение сгенерированной информации в других задачах (например, сегментация).

Основная слабость перекликается с основной угрозой – узкая направленность и отсутствие спроса. Однако эти проблемы решаются путём расширения области применения программы.

5.3. Планирование и формирование бюджета научных исследований

5.3.1. Структура выполнения работ

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может варьироваться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей.

В данном случае была сформирована рабочая группа, в состав которой входят 1 инженер и один научный руководитель.

Вся деятельность в рамках научного исследования представляет собой комплекс этапов и работ, перечень которых, с указанием исполнителей, представлен в таблице 3.

Таблица 3 – Перечень этапов, работ и распределение исполнителей

| Основные этапы | № работы | Содержание работы | Исполнитель |
|--------------------------------|----------|---|------------------------|
| Разработка темы и задания | 1 | Выбор темы ВКР | инженер; науч. рук. |
| | 2 | Получение ТЗ | инженер; науч. рук. |
| Выбор направления исследования | 3 | Подбор и анализ материала | инженер; науч. рук. |
| | 4 | Календарное планирование работ по теме | инженер; науч. рук. |
| Теоретические исследования | 5 | Изучение существующих разновидностей методов решения задачи | инженер |
| | 6 | Определение и изучение используемых средств разработки | инженер |
| Процесс разработки | 7 | Разработка, отладка проекта на ЭВМ | инженер |
| | 8 | Тестирование | инженер |
| | 9 | Доработка, устранение выявленных ошибок | инженер |
| Заключительный этап | 10 | Составление отчета о проделанной работе и оценка эффективности полученных результатов | инженер; науч. рук. |
| | 11 | Защита дипломного проекта | инженер |

5.3.2. Трудоёмкость выполнения и график проведения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников проекта.

Трудоёмкость выполнения оценивается экспертным путем в человеко-днях и носит вероятностный характер, так как зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ож_i}$ используется следующая формула:

$$t_{ож_i} = \frac{3t_{min_i} + 2t_{max_i}}{5}$$

где $t_{ож_i}$ – ожидаемая трудоёмкость выполнения i -ой работы, чел.-дн.; t_{min_i} – минимально возможная трудоёмкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного

стечения обстоятельств), чел.-дн.; t_{max_i} – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоёмкости определяется продолжительность каждой работы в рабочих днях, учитывая параллельность выполнения работ несколькими исполнителями, по следующей формуле:

$$T_{p_i} = \frac{t_{ож_i}}{Ч_i}$$

где T_{p_i} – продолжительность одной работы, раб. дн.; $t_{ож_i}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.; $Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Чтобы наглядно описать трудовые затраты, следует построить график проведения научных работ в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{кi} = T_{p_i} \cdot k_{кал}$$

где $T_{кi}$ – продолжительность выполнения i -ой работы в календарных днях; T_{p_i} – продолжительность выполнения i -ой работы в рабочих днях; $k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}}$$

где $T_{кал}$ – количество календарных дней в году; $T_{вых}$ – количество выходных дней в году; $T_{пр}$ – количество праздничных дней в году.

Рассчитанные значения в календарных днях по каждой работе T_{ki} необходимо округлить до целого числа.

Коэффициент календарности на 2021 год составляет:

$$k_{\text{кал}} = \frac{365}{365 - 122} = \frac{365}{243} = 1,5$$

Рассчитанные значения трудоёмкости и длительности работ представлены в Таблица 4 – Временные показатели проведения научного исследования 4.

Таблица 4 – Временные показатели проведения научного исследования

| Номер работы | Трудоёмкость работ, чел.-дн. | | | Длительность работ, дн. | | | |
|---------------------|------------------------------|-----------|----------|-------------------------|---------|------------|---------|
| | t_{min} | t_{max} | $t_{ож}$ | T_p | | T_k | |
| | | | | науч. рук. | инженер | науч. рук. | инженер |
| 1 | 3 | 6 | 4,2 | 2,1 | 2,1 | 3 | 3 |
| 2 | 4 | 6 | 4,8 | 2,4 | 2,4 | 4 | 4 |
| 3 | 5 | 10 | 7 | 3,5 | 3,5 | 5 | 5 |
| 4 | 4 | 8 | 5,6 | 2,8 | 2,8 | 4 | 4 |
| 5 | 12 | 16 | 13,6 | 0 | 13,6 | 0 | 20 |
| 6 | 10 | 14 | 11,6 | 0 | 11,6 | 0 | 17 |
| 7 | 15 | 20 | 17 | 0 | 17 | 0 | 26 |
| 8 | 8 | 11 | 9,2 | 0 | 9,2 | 0 | 14 |
| 9 | 4 | 6 | 4,8 | 0 | 4,8 | 0 | 7 |
| 10 | 10 | 13 | 11,2 | 5,6 | 5,6 | 8 | 8 |
| 11 | 1 | 1 | 1 | 0 | 1 | 0 | 2 |
| <u>Итого</u> | 76 | 111 | 90 | 16,4 | 73,6 | 24 | 110 |

На основе полученной таблицы построим календарный план-график (таблице 5).

Таблица 5 – Календарный план-график проведения работ

| № | Вид работы | Исполнитель | T _к | Продолжительность выполнения работ | | | | | | | | | | | | |
|----|---|------------------------|----------------|------------------------------------|---|------|---|---|--------|---|---|-----|---|---|---|---|
| | | | | февраль | | март | | | апрель | | | май | | | | |
| | | | | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | | |
| 1 | Выбор темы ВКР | инженер; науч. рук. | 3 | ■ | ■ | | | | | | | | | | | |
| 2 | Получение ТЗ | инженер; науч. рук. | 4 | ■ | ■ | | | | | | | | | | | |
| 3 | Подбор и анализ материала | инженер; науч. рук. | 5 | ■ | ■ | | | | | | | | | | | |
| 4 | Календарное планирование работ по теме | инженер; науч. рук. | 4 | | ■ | | | | | | | | | | | |
| 5 | Изучение существующих разновидностей методов решения задачи | инженер | 20 | | | ■ | ■ | | | | | | | | | |
| 6 | Определение и изучение используемых средств разработки | инженер | 17 | | | | | ■ | ■ | | | | | | | |
| 7 | Разработка, отладка проекта на ЭВМ | инженер | 26 | | | | | | ■ | ■ | ■ | | | | | |
| 8 | Тестирование | инженер | 14 | | | | | | | | | ■ | ■ | | | |
| 9 | Доработка, устранение выявленных ошибок | инженер | 7 | | | | | | | | | | ■ | ■ | | |
| 10 | Составление отчета о проделанной работе и оценка эффективности полученных результатов | инженер; науч. рук. | 8 | | | | | | | | | | | | ■ | ■ |
| 11 | Защита дипломного проекта | инженер | 2 | | | | | | | | | | | | ■ | ■ |

■ – инженер, ■■■ – науч. рук.

Таким образом, согласно составленному календарному плану, продолжительность проекта составляет 11 полных декад (с 10 февраля по 30 мая).

5.3.3. Формирование бюджета затрат

При планировании бюджета должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета используется следующая группировка затрат по статьям:

- материальные затраты;
- затраты на оборудование;
- амортизационные отчисления;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы;

5.3.3.1. Материальные затраты

Данная статья включает стоимость всех материалов, используемых при разработке проекта (например, сырьё и материалы для создания продукции). В текущей работе материалами можно считать выборки данных, используемых для обучения нейронных сетей. Так как работа является исследовательской, данные были предоставлены научным руководителем без затрат. Прочие материальные затраты представлены в таблице 6.

Таблица 6 – Материальные затраты

| Наименование | Цена за ед., руб. | Кол-во, шт. | Сумма, руб. |
|----------------------------------|-------------------|-------------|-------------|
| Офисная бумага, упак. 500 листов | 310 | 1 | 310 |
| Шариковая ручка | 30 | 2 | 60 |
| <u>Итого</u> | | | 370 |

5.3.3.2. Затраты на оборудование

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ. Определение стоимости спецоборудования производится по действующим прейскурантам.

При выполнении научно-исследовательской работы использовался ПК с видеокартой высокой производительности. Его стоимость представлена в таблице 7.

Таблица 7 – Расчёт бюджета затрат на приобретение спецоборудования

| Наименование оборудования | Количество, шт. | Цена за 1 ед., тыс. руб. | З _{обор} , тыс. руб. |
|---------------------------|-----------------|--------------------------|-------------------------------|
| Персональный компьютер | 1 | 104450 | 104450 |

5.3.3.3. Амортизационные отчисления

Амортизационные отчисления – отчисления на полное восстановление основных производственных фондов, вычисленная исходя из их балансовой стоимости и утвержденных норм амортизации.

Срок полезного использования для офисных машин (код 330.28.23.23) составляет 3 года.

Затраты на амортизацию рассчитывается по формуле:

$$Z_{\text{ам}} = \frac{T_{\text{исп}}}{12} \cdot Z_{\text{обор}} \cdot \frac{N_a}{100\%}$$

где $T_{\text{исп}}$ – время использования, мес.; $Z_{\text{об}}$ – стоимость оборудования, руб.; N_a – норма амортизации.

Норма амортизации:

$$N_a = \frac{100\%}{T_{\text{п.и.}}}$$

где $T_{\text{п.и.}}$ – срок полезного использования оборудования.

Выполним расчёт амортизационных отчислений за 4 месяца использования персонального компьютера:

$$Z_{ам} = \frac{4}{12} \cdot 104450 \cdot 0,33 = 11375 \text{ руб.}$$

5.3.3.4. Основная заработная плата

Статья включает расчёт основной заработной платы работников, непосредственно занятых выполнением проекта, то есть инженера и научного руководителя.

Основная заработная плата рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p$$

где $Z_{осн}$ – основная заработная плата одного работника, руб.; $Z_{дн}$ – среднедневная заработная плата работника, руб.; T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_d}$$

где Z_m – месячный должностной оклад работника, руб.; M – количество месяцев работы без отпуска в течение года (10,4 для 6-дневной недели); F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 8 – Баланс рабочего времени (для 6-дневной недели)

| Показатели рабочего времени | Дни |
|--|-----|
| Календарные дни | 365 |
| Нерабочие дни (праздники/выходные) | 122 |
| Потери рабочего времени (отпуск/невыходы по болезни) | 55 |
| Действительный годовой фонд рабочего времени | 188 |

Месячный должностной оклад работника:

$$Z_m = Z_{тс} \cdot (1 + k_{пр} + k_d) \cdot k_p$$

где $Z_{тс}$ – заработная плата по тарифной ставке, руб.; $k_{пр}$ – премиальный коэффициент (0,3); k_d – коэффициент доплат и надбавок (0,2-0,5); k_p – районный коэффициент (для Томска – 1,3).

Таким образом, расчёт основной заработной платы представлен в таблице 9.

Таблица 9 – Расчёт основной заработной платы

| Исполнители | $Z_{тс},$ руб. | $k_{пр}$ | k_d | k_p | $Z_m,$ руб. | M | F_d | $Z_{дн},$ руб. | T_p | $Z_{осн},$ руб. |
|---------------------|-------------------|----------|-------|-------|----------------|------|-------|-------------------|-------|--------------------|
| Инженер | 20000 | 0,3 | 0,2 | 1,3 | 39000 | 10,4 | 188 | 2157 | 73,6 | 158788 |
| Руководитель | 47317 | 0,3 | 0,2 | 1,3 | 92268 | 10,4 | 188 | 5104 | 16,4 | 83706 |
| <u>Итого</u> | | | | | | | | | | 242494 |

5.3.3.5. Дополнительная заработная плата

Затраты по дополнительной заработной платы исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн}$$

где $k_{доп}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12–0,15). Расчёт дополнительной заработной платы приведён в таблице 10.

Таблица 10 – Расчёт дополнительной заработной платы

| Исполнители | $Z_{осн},$ руб. | $k_{доп}$ | $Z_{доп},$ руб. |
|---------------------|-----------------|-----------|-----------------|
| Инженер | 158788 | 0,12 | 19055 |
| Руководитель | 83706 | 0,12 | 10044 |
| <u>Итого</u> | | | 29099 |

5.3.3.6. Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам

государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}})$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды.

Представим отчисления во внебюджетные фонды в табличной форме (таблице 11):

Таблица 11 – Отчисления во внебюджетные фонды

| Исполнители | $Z_{\text{осн}}$, руб. | $Z_{\text{доп}}$, руб. | $k_{\text{внеб}}$ | $Z_{\text{внеб}}$, руб. |
|---------------------|-------------------------|-------------------------|-------------------|--------------------------|
| Инженер | 158788 | 19055 | 0,3 | 53353 |
| Руководитель | 83706 | 10044 | 0,3 | 28125 |
| <u>Итого</u> | | | | 81478 |

5.3.3.7. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не включенные в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы и т.д. Величина коэффициента накладных расходов принимается в размере 16%. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = 0,16 \cdot (Z_{\text{мат}} + Z_{\text{обор}} + Z_{\text{ам}} + Z_{\text{осн}} + Z_{\text{доп}} + Z_{\text{внеб}})$$

Запишем результат в таблице 12.

Таблица 12 – Накладные расходы

| $Z_{\text{мат}}$, руб. | $Z_{\text{обор}}$, руб. | $Z_{\text{ам}}$, руб. | $Z_{\text{осн}}$, руб. | $Z_{\text{доп}}$, руб. | $Z_{\text{внеб}}$, руб. | $Z_{\text{накл}}$, руб. |
|-------------------------|--------------------------|------------------------|-------------------------|-------------------------|--------------------------|--------------------------|
| 370 | 104450 | 11375 | 242494 | 29099 | 81478 | 75083 |
| <u>Итого</u> | | | | | | 75083 |

5.3.3.8. Итоговый бюджет затрат

Рассчитанная величина затрат проектировочной работы является основной для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается проектной организацией в качестве нижнего предела затрат на разработку. Данные бюджета затрат приведены в таблице 13.

Таблица 13 – Бюджет затрат

| Наименование | Сумма, руб. | Удельный вес, % |
|--|---------------|-----------------|
| Материальные затраты | 370 | 0,07 |
| Затраты на оборудование | 104450 | 19,19 |
| Амортизационные отчисления | 11375 | 2,09 |
| Затраты на основную заработную плату | 242494 | 44,55 |
| Затраты на дополнительную заработную плату | 29099 | 5,35 |
| Отчисления во внебюджетные фонды | 81478 | 14,97 |
| Накладные расходы | 75023 | 13,79 |
| <u>Общий бюджет</u> | 544349 | 100 |

Исходя из полученного бюджета затрат, можно сделать вывод, что основная его часть уходит на выплату заработной платы и на закупку оборудования.

5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

5.4.1. Интегральный показатель ресурсоэффективности

Оценка сравнительной эффективности исследования основывается на определении интегрального показателя ресурсоэффективности, который имеет следующий вид:

$$I_p = \sum a_i \cdot b_i$$

где I_p – интегральный показатель ресурсоэффективности; a_i – весовой коэффициент i -го параметра; b_i – бальная оценка i -го параметра, которая устанавливается экспертным путем по выбранной шкале оценивания.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 14.

Таблица 14 – Оценка характеристик исполнения проекта

| Критерии | Весовой коэффициент | Бальная оценка разработки |
|--|----------------------------|----------------------------------|
| Программная реализация, сложность сопровождения | 0,3 | 4 |
| Степень соответствия современным требованиям в области использования | 0,25 | 5 |
| Простота ввода в эксплуатацию | 0,2 | 5 |
| Сопроводительная документация | 0,1 | 3 |
| Предполагаемый срок эксплуатации | 0,15 | 4 |
| <u>Итог</u> | 1 | 4,35 |

5.5. Выводы по разделу

В рамках представленного раздела была доказана целесообразность разработки проекта на основе следующих пунктов:

- согласно оценке конкурентоспособности, используемый в работе алгоритм оказался наиболее предпочтительным в сравнении с другими аналогами;
- с помощью SWOT-анализа были выявлены сильные и слабые стороны проекта, возможности и угрозы. Проанализировано их взаимное влияние на проект;
- оценка трудоёмкости и длительности работ по проекту показала, что продолжительность выполнения составит 74 рабочих дня или 110 календарных дней. Был составлен календарный план-график проведения работ в виде диаграммы Ганта, отражающий деятельность инженера и научного руководителя;
- затраты на выполнение проекта составляют 544349 рублей;
- интегральный показатель ресурсоэффективности составил 4,35 баллов из 5 возможных, что свидетельствует об эффективности реализации проекта.

6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Работа посвящена разработке программного обеспечения для генерации новых данных на основе примеров с помощью нейронных сетей. Потенциальными потребителями результатов исследования являются отрасли медицины. Для синтеза информации используются наборы различных медицинских данных (снимки, выборка пациентов).

В качестве рабочего оборудования выступает ПЭВМ, поскольку обучение и дальнейшее применение нейронных сетей происходит именно на нём. С каждым годом электронно-вычислительные машины (ЭВМ) находят всё большее применение как на производстве, так и для научно-исследовательских и конструкторских работ, а также в сфере управления и образования. Уже нельзя представить работу на предприятиях, в организациях или в домашних условиях без использования компьютеров. Однако компьютер является источником вредного воздействия на организм человека, а, следовательно, и источником профессиональных заболеваний. Это предъявляет к каждому пользователю персонального компьютера определённые требования – знать о вредном воздействии ПЭВМ на организм человека и соблюдать необходимые меры защиты от этих воздействий.

6.1. Правовые и организационные вопросы обеспечения безопасности

6.1.1. Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства

Основные положения по охране труда изложены в Трудовом кодексе Российской Федерации. В этом документе указано, что охрана здоровья трудящихся, обеспечение безопасных условий труда, ликвидация профессиональных заболеваний и производственного травматизма являются одной из главных забот государства.

Трудовой кодекс, в частности, устанавливает права и обязанности работника и работодателя, регулирует вопросы охраны труда, профподготовки, переподготовки и повышения квалификации, трудоустройства, социального партнерства. Закрепляются правила оплаты и нормирования труда, порядок разрешения трудовых споров.

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. Продолжительность рабочего дня работников не должна превышать 40 часов в неделю.

6.1.2. Организационные мероприятия при компоновке рабочей зоны

Рабочее место при работе с ПК должно составлять не менее чем 6 кв. м. Пространство для ног должно соответствовать следующим параметрам: высота не менее 600 мм, расстояние сиденья до нижнего края рабочей поверхности не менее 150 мм, а высота сиденья 420 мм. Высота стола должна зависеть от роста оператора.

Также предусмотрены следующие требования к организации рабочего места пользователя ПЭВМ:

- Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ и позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.
- Тип рабочего стула (кресла) следует выбирать с учётом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а

также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Разработка программного обеспечения для генерации данных производилась в комнате общей площадью 20 кв. м. Размеры стола соответствовали всем необходимым нормам, рабочий стул представлял собой современной компьютерный стул с необходимым функционалом.

6.2. Производственная безопасность

Производственная безопасность – это система организационных мероприятий и технических средств, уменьшающих вероятность воздействия на рабочих опасных производственных факторов до приемлемого уровня. Для определения опасных факторов воспользуемся классификацией опасных и вредных производственных факторов по ГОСТ 12.0.003-2015. Результаты анализа представлены в таблице 15.

Таблица 15 – Опасные и вредные факторы

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | Нормативные документы |
|--|-------------|--------------|---|
| | Разработка | Эксплуатация | |
| 1. Электромагнитные поля | + | + | Гигиенические требования к персональным электронно-вычислительным машинам и организации СанПиН 2.2.2.542-96[11] |
| 2. Превышение уровня шума | + | + | Уровень шума на рабочих местах СН 2.2.4/2.1.8.562–96[12] |
| 3. Отклонение показателей микроклимата | + | + | Гигиенические требования к микроклимату производственных помещений СанПиН 2.2.4.548–96[13] |
| 4. Недостаточная освещенность рабочей зоны | + | + | Естественное и искусственное освещение. СП 52.13330.2016[14] |
| 5. Психофизиологические факторы | + | + | Трудовой кодекс Российской Федерации от 30.12.2001 N 197[15] |
| 6. Электробезопасность | + | + | Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов ГОСТ 12.1.038-82[16] |

6.2.1. Анализ выявленных вредных и опасных факторов

Вредным называется производственный фактор, воздействие которого на работающего в определенных условиях приводит к заболеванию или снижению работоспособности. При изменении уровня и времени воздействия вредные производственные факторы могут стать опасными. Опасными считаются производственные факторы, воздействие которых на работающего в конкретных условиях может привести к травмам, а также другим внезапным резким ухудшениям здоровья.

6.2.1.1. Электромагнитные поля

ПЭВМ генерирует в окружающее пространство широкий спектр ЭМП различной интенсивности, в том числе:

- электростатическое поле;
- переменные низкочастотные ЭМП;
- электромагнитное излучение радиочастотного диапазона;
- электромагнитное излучение оптического (видимого) диапазона;
- ультрафиолетовое (УФ) и рентгеновское излучения ЭЛТ.

Источником электростатического поля является экран дисплея, несущий высокий электростатический потенциал. Рентгеновское и ультрафиолетовое излучения практически полностью поглощаются внутри корпуса дисплея, а интенсивность излучений радиочастотного диапазона пренебрежимо мала.

По частотному спектру ЭМП, согласно СанПиН 2.2.2.542-96, разделяются на две группы:

- низкочастотные поля в частотном диапазоне до 2 кГц, создаваемые блоком сетевого питания и блоком кадровой развертки дисплея;
- высокочастотные поля в частотном диапазоне 2–400 кГц, создаваемые блоком строчной развертки и блоком сетевого питания (в случае, если он импульсный).

Следствием систематического воздействия переменных ЭМП с параметрами, превышающими допустимые нормы, являются функциональные нарушения нервной, эндокринной и сердечно-сосудистой систем.

6.2.1.2. Производственный шум

При выполнении работ, описанных выше, специалист может оказаться под шумовым воздействием со стороны оборудования, находящегося в рабочем помещении: ПК, печатающие устройства, оборудование поддержки микроклимата (кондиционеры, вентиляция) и пр.

Работы, выполняемые специалистом, оцениваются как научная деятельность, конструирование и проектирование, программирование, следовательно, согласно ГОСТ 12.1.003-83 «ССБТ. Общие требования безопасности» эквивалентный уровень шума в рабочем помещении не должен превышать 50 дБА. Допустимый уровень шума ограничен ГОСТ 12.1.003-83 и СН 2.2.4/2.1.8.562–96. Уровень шума на рабочем месте математиков-программистов и операторов видеоматериалов не должен превышать 50 дБА, а в залах обработки информации на вычислительных машинах – 65 дБА.

Вибрация оборудования на рабочих местах не должна превышать допустимых величин, установленных ГОСТ 12.1.012-96.

6.2.1.3. Микроклимат в помещении

Микроклимат производственных помещений – это климат внутренней среды помещений, который определяется действующими на организм человека сочетаниями температур воздуха и поверхностей, относительной влажности воздуха, скорости движения воздуха и интенсивности теплового излучения.

Нормы оптимальных и допустимых показателей микроклимата при работе с ЭВМ устанавливает СанПиН 2.2.4.548–96. Все категории работ разграничиваются на основе интенсивности энергозатрат организма в ккал/ч

(Вт). Работа, производимая сидя и сопровождаемая незначительным физическим напряжением, относится к категории Ia – работа с интенсивностью энергозатрат до 120 ккал/ч (до 139 Вт). Для данной категории допустимые нормы микроклимата представлены в таблице 16.

Таблица 16 – Допустимые нормы микроклимата в рабочей зоне производственных помещений

| Период года | Категория работы | Температура воздуха, °С | | Относительная влажность, % | | Скорость движения воздуха, м/с | |
|-------------|------------------|-------------------------|-------|----------------------------|------|--------------------------------|------|
| | | Фактич. | Доп. | Фактич. | Доп. | Фактич. | Доп. |
| Холодный | Легкая Ia | 22-24 | 19-24 | 55 | <75 | 0,1 | ≤0,1 |
| Теплый | | 23-25 | 20-28 | 55 | <60 | 0,1 | ≤0,2 |

6.2.1.4. Освещённость рабочей зоны

Источниками света могут быть как естественные, так и искусственные объекты. Естественным источником в помещении служит солнце, искусственными являются электрические лампочки. При длительной работе в условиях недостаточной освещенности и при нарушении других параметров световой среды зрительное восприятие снижается, развивается близорукость, болезнь глаз, появляются головные боли.

По нормативу ГОСТ Р 55710-2013 освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Яркость светильников общего освещения в зоне углов излучения от 50 до 90° с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м, защитный угол светильников должен быть не менее 40°.

Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1,4. Коэффициент пульсации не должен превышать 5%.

6.2.1.5. Психофизиологические факторы

Значительное умственное напряжение и другие нагрузки приводят к переутомлению функционального состояния центральной нервной системы, нервно-мышечного аппарата рук. Нерациональное расположение элементов рабочего места вызывает необходимость поддержания вынужденной рабочей позы. Длительный дискомфорт вызывает повышенное позвоночное напряжение мышц и обуславливает развитие общего утомления и снижение работоспособности.

При длительной работе за экраном дисплея появляется выраженное напряжение зрительного аппарата с появлением жалоб на неудовлетворительность работы, головные боли, усталость и болезненное ощущение в глазах, в пояснице, в области шеи, руках.

6.2.1.6. Электробезопасность

Токи статического электричества, наведённые в процессе работы компьютера на корпусах монитора, системного блока, клавиатуры, а также при работе за паяльной станцией, могут приводить к разрядам при прикосновении к этим элементам.

Категории помещения по степени опасности поражения электрическим током подразделены на три класса. Комната, в которой проводилась основная работа, относится к первой категории: сухая, с нормальной температурой воздуха, с токонепроводящими полами.

На рабочем месте пользователя размещены дисплей, клавиатура и системный блок. Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.

Согласно ГОСТ 12.1.038-82 при работе за ПК допускаются уровни напряжений прикосновения и токов, представленные в таблице 17.

Таблица 17 – Уровни напряжений прикосновения и токов

| Род тока | Напряжение прикосновения, В | Ток, мА |
|-------------------|-----------------------------|---------|
| | не более | |
| Переменный, 50 Гц | 2,0 | 0,3 |
| Постоянный | 8,0 | 1,0 |

6.2.2. Обоснование мероприятий по снижению воздействия

Для минимизации вредных воздействий на организм в процессе использования ПЭВМ необходимо принять следующие меры:

1. Использование современных дисплеев, оборудованных эффективной системой защиты от электростатического поля. Кроме того, СанПиН 2.2.2.542-96 допускает применение приэкранных фильтров, специальных экранов и других средств индивидуальной защиты.
2. Обеспечение чистоты воздуха в производственном помещении достигается удалением загрязненного или нагретого воздуха из помещения и подачей в него свежего воздуха, т.е. вентиляцией. Для благоприятного микроклимата рабочие участки необходимо оснастить кондиционерным оборудованием.
3. Для снижения уровня шума, производимого персональными компьютерами, рекомендуется регулярно проводить их техническое обслуживание: чистка от пыли, замена смазывающих веществ; также применяются звукопоглощающие материалы. При необходимости можно использовать индивидуальные средства защиты (например, противошумные наушники).
4. Естественное и искусственное освещение должно соответствовать требованиям СП 52.13330.2016. Для местного освещения используют светильники, отрегулированные так, чтобы освещённость была не ниже значений, установленных санитарными нормами. Так как

освещенность, создаваемая естественным светом, изменяется в зависимости от времени дня, года, метеорологических факторов, то для поддержания постоянного уровня освещенности необходимо применять комбинированное освещение – естественное и искусственное. Искусственное общее освещение – лампы накаливания необходимо располагать в верхней зоне помещения и непосредственной близости рабочей зоны.

5. С целью снижения или устранения нервно-психологического, зрительного и мышечного напряжения, предупреждение переутомления необходимо проводить комплекс физических упражнений и сеансы психофизической разгрузки и снятия усталости во время регламентируемых перерывов, и после окончания рабочего дня.
6. В качестве защиты от опасности поражения электрическим током применяются следующие меры: электрическая изоляция токоведущих частей (сопротивление изоляции должно быть не менее 0,5 МОм); использование малых напряжений, например, не более 50 В; электрическое разделение сетей на отдельные короткие участки; защитное заземление и зануление. Для защиты от воздействия статического электричества нужно проводить влажную уборку помещения, использовать увлажнители воздуха и средства индивидуальной защиты, такие как антистатические спреи и браслеты.

6.3. Экологическая безопасность

Охрана окружающей среды – это комплексная проблема. Наиболее активная форма её решения – это сокращение вредных выбросов промышленных предприятий через полный переход к безотходным или малоотходным технологиям производства.

Разработанный программный продукт не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде.

Тепловыделение от оборудования ПЭВМ вносит существенный вклад в тепловую нагрузку помещения. В современных задачах (например, обучение нейронных сетей) возникает необходимость использовать мощные комплектующие. Чаще всего это графические процессоры (видеокарты), которые подвержены сильному нагреву вплоть до 80 и более градусов Цельсия при продолжительной работе и полной загрузке. Система из нескольких подобных ЭВМ выделяет огромное количество тепла, создавая целые тепловые кластеры, и вносит свой вклад в тепловое загрязнение атмосферы.

Для обеспечения минимального вреда окружающей среде с точки зрения тепловыделения необходимо использовать специализированные системы охлаждения, например водяное охлаждение. Это позволит значительно снизить температуру рабочих устройств.

Негативное воздействие на экологию оказывают факторы, связанные с производством и эксплуатацией компьютерной техники и комплектующих. Так, например, аккумуляторные батареи компьютеров содержат в себе различные щелочи и кислоты, которые наносят вред окружающей среде, если они были неправильно утилизированы.

Кроме того, необходимо отправлять на утилизацию все вышедшие из строя комплектующие ПЭВМ во избежание загрязнения окружающей среды.

6.4. Безопасность в чрезвычайных ситуациях

К возможным чрезвычайным ситуациям на данном рабочем месте выделяют внезапное обрушение здания, аварии на коммунальных системах жизнеобеспечения населения, пожар, угроза пандемии.

С учетом специфики работы и наличием вычислительной техники в помещении наиболее вероятно возникновение пожара, под которым

понимается вышедший из-под контроля процесс горения, обусловленный возгоранием вычислительной техники и угрожающий жизни и здоровью работников.

Причинами возникновения данного вида ЧС могут являться:

- возникновение короткого замыкания в электропроводке;
- возгорание устройств ПЭВМ из-за неисправности аппаратуры;
- возгорание устройств искусственного освещения;
- возгорание мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок.

При повреждении изоляции проводов оголяются жилы, по которым идет электрический ток. Если оголенные жилы под напряжением касаются друг друга, то происходит короткое замыкание. При коротком замыкании проводники сильно нагреваются, что может вызвать пожар. Загореться может как сам провод (изоляция), так и проводка в помещении.

Согласно ГОСТ 12.1.004-91[17] «Пожарная безопасность. Общие требования», при работе с компьютером необходимо соблюдать следующие нормы пожарной безопасности:

- для предохранения сети от перегрузок запрещается одновременно подключать к сети количество потребителей, превышающих допустимую нагрузку;
- работы за компьютером проводить только при исправном состоянии оборудования, электропроводки;
- иметь средства для тушения пожара (огнетушитель);
- установить количество, размеры и соответствующее конструктивное исполнение эвакуационных путей и выходов;
- обеспечить возможность беспрепятственного движения людей по эвакуационным путям

При возникновении пожара должна сработать система пожаротушения, передав на пункт пожарной станции сигнал о ЧС. В случае если система не сработала, то необходимо самостоятельно произвести вызов пожарной службы по телефону 101, сообщить точный адрес места возникновения ЧС и ожидать приезда специалистов. В это время нужно принять меры по эвакуации людей, отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

6.5. Выводы и рекомендации

Проанализировав условия труда на рабочем месте, где была разработана бакалаврская работа, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья сотрудника.

Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму. Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность не представляет опасности окружающей среде. Важно добавить, что монитор компьютера служит источником ЭМП – вредного фактора, который отрицательно влияет на здоровье работника при продолжительной непрерывной работе и приводит к снижению работоспособности. Поэтому во избежание негативного влияния на здоровье необходимо делать перерывы при работе с ЭВМ и проводить специализированные комплексы упражнений для глаз.

Заключение

В ходе выполнения выпускной квалификационной работы была достигнута основная цель работы – разработаны модели нейронных сетей (вариационный автоэнкодер и генеративно-сопоставительная сеть) для расширения обучающих выборок. Основой для создания является язык программирования Python, а также следующие библиотеки: TensorFlow, NumPy, Pandas.

Данная разработка позволит значительно упростить процесс получения обучающих выборок в области медицины.

В процессе разработки были выполнены следующие задачи:

- обзор литературы в выбранной предметной области;
- исследование особенностей генерации различных типов данных и изучение принципов работы генеративных алгоритмов;
- разработка модели вариационного автоэнкодера и генеративно-сопоставительной сети;
- обучение и оценка работоспособности нейронных сетей.

В качестве мер по дальнейшему развитию проекта рассматривается изменение модели генератора в генеративно-сопоставительной сети с целью получения более качественной генерации изображений.

Список использованных источников

[1] Hanna Gruselius. Generative Models and Feature Extraction on Patient Images and Structure Data in Radiation Therapy / Hanna Gruselius. KTH Royal Institute of Technology, 2018. – 90 с.

[2] Convolutional Variational Autoencoder [Электронный ресурс]: TensorFlow Core. URL: <https://www.tensorflow.org/tutorials/generative/cvae> (дата обращения: 08.04.2021)

[3] Вариационные автоэнкодеры [Электронный ресурс]: Habr. URL: <https://habr.com/ru/post/331552/> (дата обращения: 08.04.2021)

[4] arXiv e-print archive [Электронный ресурс] // Generative Adversarial Nets.– 2014г.– Режим доступа: <https://arxiv.org/pdf/1406.2661.pdf> (дата обращения: 25.12.2019)

[5] arXiv e-print archive [Электронный ресурс] // UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS.– 2016г.– Режим доступа: <https://arxiv.org/pdf/1511.06434.pdf> (дата обращения: 25.12.2019)

[6] Sebastian Raschka. Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 / Sebastian Raschka, Vahid Mirjalili. Published by Packt Publishing Ltd., 2019. – 771 с.

[7] Proceedings of Machine Learning Research [Электронный ресурс] // Wasserstein Generative Adversarial Networks. – 2017г.– Режим доступа: <http://proceedings.mlr.press/v70/arjovsky17a/arjovsky17a.pdf> (дата обращения: 23.04.2020)

[8] arXiv e-print archive [Электронный ресурс] // Improved Training of Wasserstein GANs.– 2017г.– Режим доступа: <https://arxiv.org/pdf/1704.00028.pdf> (дата обращения: 24.04.2020)

[9] Видяев, И.Г. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / И.Г. Видяев, Г.Н.

Серикова, Н.А. Гаврикова; Томский политехнический университет – Томск: Изд-во ТПУ, 2014. – 36 с.

[10] Пашков, Е.Н. Методические указания по разработке раздела «Социальная ответственность» ВКР бакалавра всех направлений (специальностей) и форм обучения ТПУ / Сост. Е.Н. Пашков, А.И. Сечин, И.Л. Мезенцева – Томск: Изд-во ТПУ, 2019. – 24 с.

[11] СанПиН 2.2.2.542-96. Гигиенические требования к персональным электронно-вычислительным машинам и организации. – URL: <https://docs.cntd.ru/document/5200235> (дата обращения: 06.05.2021).

[12] СН 2.2.4/2.1.8.562-96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. – URL: <https://docs.cntd.ru/document/901703278> (дата обращения: 06.05.2021).

[13] СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. – URL: <https://docs.cntd.ru/document/901704046> (дата обращения: 06.05.2021).

[14] СП 52.13330.2016. Естественное и искусственное освещение. – URL: <https://docs.cntd.ru/document/456054197> (дата обращения: 06.05.2021).

[15] "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.04.2021) (с изм. и доп., вступ. в силу с 01.05.2021). – URL: http://www.consultant.ru/document/cons_doc_LAW_34683/ (дата обращения: 06.05.2021).

[16] ГОСТ 12.1.038-82. Предельно допустимые значения напряжений прикосновения и токов. – URL: <https://docs.cntd.ru/document/5200313> (дата обращения: 06.05.2021).

[17] ГОСТ 12.1.004-91. Система стандартов безопасности труда. Пожарная безопасность. Общие требования. – URL: <https://docs.cntd.ru/document/9051953> (дата обращения: 06.05.2021).

Приложение А

Описание функций для обучения генеративно-сопоставительной сети

```
def train(self, train_set):
    start_time = time.time()
    self.setup_losses_from_file()

    for epoch in range(self.current_epoch, self.current_epoch +
self.epochs):
        for batch in train_set:
            for _ in range(5):
                self.d_train_step(batch)
                self.g_train_step()

            epoch_log = 'Epoch {:03d} | ET {:.2f} min | Gen Loss: {:.6f}, ' \
                'Disc Loss: {:.6f} Adv Loss:
{:6.6f}'.format(self.current_epoch,
(time.time() - start_time) / 60,
self.g_loss.result(),
self.d_loss.result(),
self.adv_loss.result())

            print(epoch_log)

            self.all_losses.append([self.g_loss.result().numpy(),
                self.d_loss.result().numpy(),
                self.adv_loss.result().numpy()])

            if self.current_epoch % 10 == 0:
                self.save_demo_images(self.current_epoch)
                print('Saving demo images for epoch {} at
result\images_at_epoch_{}.png'.format(self.current_epoch,
self.current_epoch))

                ckpt_save_path = self.ckpt_manager.save()
                print('Saving checkpoint for epoch {} at
{}'.format(self.current_epoch, ckpt_save_path))

                self.plot_losses()
                self.save_losses_to_file()

            with open('train_logs.txt', 'a') as train_logs:
                train_logs.write(epoch_log + '\n')

            self.d_loss.reset_states()
            self.adv_loss.reset_states()
            self.g_loss.reset_states()

            self.current_epoch += 1

@tf.function
def g_train_step(self):
    with tf.GradientTape() as tape:
        z = tf.random.uniform(shape=(self.batch_size, self.z_size), minval=-
1.0, maxval=1.0)
        fake_sample = self.g_model(z, training=True)
```

```

        fake_score = self.d_model(fake_sample, training=True)
        loss = - tf.reduce_mean(fake_score, axis=1)[0]

        gradients = tape.gradient(loss, self.g_model.trainable_variables)
        self.g_optimizer.apply_gradients(grads_and_vars=zip(gradients,
self.g_model.trainable_variables))

        self.g_loss(loss)

@tf.function
def d_train_step(self, real_sample):
    with tf.GradientTape() as tape:
        z = tf.random.uniform(shape=(self.batch_size, self.z_size), minval=-
1.0, maxval=1.0)
        fake_sample = self.g_model(z)
        real_score = self.d_model(real_sample, training=True)
        fake_score = self.d_model(fake_sample, training=True)

        alpha = tf.random.uniform([self.batch_size, 1, 1, 1], 0.0, 1.0)
        inter_sample = fake_sample * alpha + real_sample * (1 - alpha)

        with tf.GradientTape() as tape_gp:
            tape_gp.watch(inter_sample)
            inter_score = self.d_model(inter_sample, training=True)
            gp_gradients = tape_gp.gradient(inter_score, inter_sample)
            gp_gradients_norm = tf.sqrt(tf.reduce_sum(tf.square(gp_gradients),
axis=[1, 2, 3]))
            gp = tf.reduce_mean((gp_gradients_norm - 1.0) ** 2)

        loss = tf.reduce_mean(fake_score) - tf.reduce_mean(real_score) + gp
        * self.lambda_gp

        gradients = tape.gradient(loss, self.d_model.trainable_variables)
        self.d_optimizer.apply_gradients(grads_and_vars=zip(gradients,
self.d_model.trainable_variables))

        self.d_loss(loss)
        self.adv_loss(loss - gp * self.lambda_gp)

```

Приложение Б

Описание модели вариационного автоэнкодера

```
class VAE(tf.keras.Model):
    def __init__(self):
        super(VAE, self).__init__()
        self.data_dim = 127
        self.latent_dim = 2
        self.hidden_layer_dim = 64
        self.optimizer = tf.keras.optimizers.Adam(1e-4)
        self.encoder = self.build_encoder()
        self.decoder = self.build_decoder()
        self.encoder.summary()
        self.decoder.summary()
        self.total_loss_tracker = tf.keras.metrics.Mean(name="total_loss")
        self.reconstruction_loss_tracker =
tf.keras.metrics.Mean(name="reconstruction_loss")
        self.kl_loss_tracker = tf.keras.metrics.Mean(name="kl_loss")
        self.total_loss_list = []

    def build_encoder(self):
        encoder_inputs = tf.keras.layers.Input(shape=(self.data_dim,))
        x = tf.keras.layers.Dense(units=self.hidden_layer_dim,
activation="relu")(encoder_inputs)
        z_mean = tf.keras.layers.Dense(units=self.latent_dim)(x)
        z_log_var = tf.keras.layers.Dense(units=self.latent_dim)(x)
        return tf.keras.Model(encoder_inputs, [z_mean, z_log_var],
name="Encoder")

    def build_decoder(self):
        decoder_inputs = tf.keras.layers.Input(shape=(self.latent_dim,))
        x = tf.keras.layers.Dense(units=self.hidden_layer_dim,
activation="relu")(decoder_inputs)
        decoder_outputs = tf.keras.layers.Dense(units=self.data_dim,
activation="tanh")(x)
        return tf.keras.Model(decoder_inputs, decoder_outputs,
name="Decoder")

    def sampling(self, inputs):
        z_mean, z_log_var = inputs
        batch = tf.shape(z_mean)[0]
        dim = tf.shape(z_mean)[1]
        epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
        z = z_mean + tf.exp(0.5 * z_log_var) * epsilon
        return z_mean, z_log_var, z

    def generate_data(self, num, test_data=None):
        new_data_list = []
        for _ in range(num):
            if test_data is None:
                z = tf.keras.backend.random_normal(shape=(1,
self.latent_dim))
            else:
                z = self.sampling(self.encoder(test_data))
            new_data = self.decoder(z)
            new_data_list.append(new_data)
        return new_data_list

    @tf.function
    def train_step(self, data):
        with tf.GradientTape() as tape:
```



```

        z_mean, z_log_var, z = self.sampling((self.encoder(data)))
        reconstruction = self.decoder(z)
        reconstruction_loss = tf.reduce_mean(
            tf.reduce_sum(
                tf.keras.losses.binary_crossentropy(data,
reconstruction)
            )
        )
        kl_loss = -0.5 * (1 + z_log_var - tf.square(z_mean) -
tf.exp(z_log_var))
        kl_loss = tf.reduce_mean(tf.reduce_sum(kl_loss, axis=1))
        total_loss = reconstruction_loss + kl_loss
        grads = tape.gradient(total_loss, self.trainable_weights)
        self.optimizer.apply_gradients(zip(grads, self.trainable_weights))
        self.total_loss_tracker.update_state(total_loss)
        self.reconstruction_loss_tracker.update_state(reconstruction_loss)
        self.kl_loss_tracker.update_state(kl_loss)

    def train(self, train_dataset, epochs):
        for epoch in range(1, epochs + 1):
            start_time = time.time()
            for train_x in train_dataset:
                self.train_step(train_x)
            end_time = time.time()

self.total_loss_list.append(self.total_loss_tracker.result().numpy())
            print('Epoch: {}, loss: {}, time elapse for current epoch: {}'.format(epoch, self.total_loss_tracker.result(), end_time
- start_time))

    def plot_loss(self):
        fig = plt.figure(figsize=(15, 8))
        ax = fig.add_subplot(1, 1, 1)
        plt.plot(self.total_loss_list, label='Total loss')
        plt.xlabel("Число эпох")
        plt.ylabel("Значение ошибки ELBO")
        plt.savefig('loss.png')
        plt.close(fig)

```