

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

А.С. Кузнецов

подпись

инициалы, фамилия

« _____ »

_____ 2019г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 Программная инженерия

код и наименование специальности

Разработка системы поддержки принятия решения

при диагностировании ишемического инсульта

тема

Научный руководитель

старший преподаватель

А.С. Михалев

подпись, дата

должность, ученая степень

инициалы, фамилия

Консультант ВКР

доцент, канд. техн. наук

М.И. Никитина

подпись, дата

должность, ученая степень

инициалы, фамилия

Выпускник

А.В. Тарасевич

подпись, дата

инициалы, фамилия

Нормоконтролер

доцент, канд. техн. наук

О.А. Антамошкин

подпись, дата

должность, ученая степень

инициалы, фамилия

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

А.С. Кузнецов

подпись

инициалы, фамилия

« _____ » _____ 2019г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Тарасевич Анастасии Владимировне.

Группа КИ15-16Б, направление 09.03.04 «Программная инженерия».

Тема выпускной квалификационной работы «Разработка системы поддержки принятия решения при диагностировании ишемического инсульта».

Утверждена приказом по университету №6264/с от 2019-05-13.

Руководитель Михалев А.С. старший преподаватель кафедры информатики ИКИТ СФУ.

Исходные данные для ВКР: учебники, научные журналы и статьи, справочные данные сети Internet-сайтов.

Перечень разделов ВКР:

- введение;
- анализ предметной области;
- выводы по главе 1;
- разработка нейронной сети;
- выводы по главе 2;
- экспериментальные исследования;
- выводы по 3 главе;
- список использованных источников.

Перечень графического материала: презентация «Разработка система поддержки принятия решения при диагностировании ишемического инсульта».

Руководитель ВКР

подпись

А.С. Михалев

Задание принял к исполнению

подпись

А.В. Тарасевич

« ___ » _____ 2019 г.

График

Выполнения выпускной квалификационной работы студентом направления 09.03.04 «Программная инженерия».

График выполнения выпускной квалификационной работы приведен в таблице 1.

Таблица 1 – График выполнения этапов ВКР

Наименование этапа	Срок выполнения этапа	Результат выполнения этапов	Примечание руководителя (отметка о выполнении этапа)
Ознакомление с целью и задачами работы	04.02-10.02	Краткая информация о теме ВКР	Выполнено
Сбор литературных источников	11.02-17.02	Список использованных источников	Выполнено
Анализ собранных источников литературы	18.02-24.02	Подготовлена 1 глава ВКР о предметной области	Выполнено
Уточнение и обоснование актуальности цели и задач ВКР	25.02-28.02	Окончательная формулировка цели и задач ВКР	Выполнено
Сбор снимков для обучения нейронной сети	18.03-31.03	Готовый набор данных, получена консультация врача-радиолога	Выполнено
Предобработка снимков	01.04-14.04	Готовый для обучения нейросети набор данных	Выполнено
Подбор подходящей архитектуры нейронной сети	15.04-01.05	Протестированы несколько архитектур, выбрана наиболее подходящая	Выполнено
Подбор параметров нейронной сети	02.05-14.05	Подобраны удовлетворяющие всем условиям параметры для нейронной сети	Выполнено
Написание 2 главы ВКР	15.05-22.05	Подготовлена 2 глава ВКР о разработке нейронной сети	Выполнено
Экспериментальное исследование	23.05-28.05	Составлен отчет по экспериментальному исследованию	Выполнено

Окончание таблицы 1

Наименование этапа	Срок выполнения этапа	Результат выполнения этапов	Примечание руководителя (отметка о выполнении этапа)
Написание 3 главы ВКР	29.05-31.05	Подготовлена 3 глава ВКР об экспериментальном исследовании	Выполнено
Подготовка доклада и презентации по теме ВКР	01.06-05.06	Доклад с презентацией по теме ВКР	Выполнено
Компоновка отчета по преддипломной практике	06.06-13.06	Отчет по результатам решения задач ВКР	Выполнено
Итоговый нормоконтроль (Н/К)	05.06	Презентация ВКР	Выполнено
Защита ВКР	09.07	Презентация по результатам бакалаврской работы	

Руководитель ВКР

подпись

А.С. Михалев

Студент группы КИ15-16Б

подпись

А.В. Тарасевич

« ____ » _____ 2019 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 Анализ предметной области.....	6
1.1 Особенности диагностики ишемического инсульта на МРТ снимках .	6
1.2 Основы теории нейронных сетей.....	9
1.2.1 Искусственный нейрон.....	10
1.2.2 Архитектуры нейронных сетей	12
1.2.3 Обучение нейронных сетей.....	15
1.2.3.1 Методы обучения нейронных сетей.....	15
1.2.3.2 Правила обучения нейронных сетей	16
1.2.4 Свёрточные нейронные сети.....	17
Вывод по 1 главе	24
Глава 2 Разработка нейронной сети	25
2.1 Предобработка снимков	25
2.1.1 Масштабирование	25
2.1.2 Средний фильтр.....	28
2.2 Выбор архитектуры.....	30
2.2.1 Использование готовых слоев	30
2.2.2 ResNet50	32
2.2.3 GoogLeNet	33
2.2.4 VGG16	35
2.2.5 Итоги выбора необходимой архитектуры	37
2.3 Обучение нейронной сети	38
2.3.1 Выбор необходимых параметров и подготовка изображений	39
2.4 Результат обучения нейронной сети	47
2.4.1 Карта признаков нейронной сети	47
2.4.2 Результаты обучения	48
Вывод по 2 главе	51
Глава 3 Экспериментальные исследования.....	52

3.1 Набор данных	52
3.2 Результаты экспериментального исследования.....	55
3.3 Перспективы развития.....	59
Вывод по 3 главе	61
ЗАКЛЮЧЕНИЕ	62
СПИСОК СОКРАЩЕНИЙ.....	64
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	65

ВВЕДЕНИЕ

В данной работе в качестве средства для обнаружения поражений мозга ишемическим инсультом предложены нейронные сети. В последнее время технологии нейронных сетей набирают популярность и используются во многих сферах, таких, как медицина, банковская сфера, сфера обслуживания и других. Данная технология набирает популярность, так как может обрабатывать большое количество данных, находить взаимосвязи, обучаться на примерах и исправлять ошибки.

В качестве предмета исследования выбраны изображения МРТ, так как они являются золотым стандартом исследования инсульта по данным ВОЗ. Анализ изображений МРТ, чаще всего, производится людьми вручную, который включает поиск и распознавание объектов, выявление границ. Данный процесс является трудоемкой и непростой задачей, которая требует определенного профессионализма. Исходя из вышесказанного следует, что методы распознавания объектов на снимках МРТ являются актуальной задачей в настоящее время. Детальный анализ МРТ снимков позволит быстро и точно ставить диагноз, для незамедлительного начала лечения.

Цель работы – разработка алгоритма распознавания ишемического инсульта по МРТ снимкам головного мозга с использованием нейронных сетей.

Программный продукт сможет выполнять функции распознавания определенного вида инсульта и в дальнейшем внесет огромный вклад в автоматизацию процессов чтения снимков в области медицины.

Предположительные результаты функционирования системы состоят в следующем:

- ускорение процесса выявления у пациента ишемического инсульта с помощью МРТ в несколько раз;
- увеличение точности поставленного диагноза за счет обучения нейронной сети на большой выборке данных.

Основной задачей является полная работоспособность системы, а также дальнейшее обучение разрабатываемой нейронной сети на большей выборке исходных данных, поскольку в тестовом варианте мы имеем нейронную сеть, обученную лишь на доступных для нас снимках, однако это будет нести в себе достаточно большую вероятность ошибки в дальнейшей работе.

Задачи:

- определить архитектуру создаваемой НС для поставленной задачи;
- сформировать исходную выборку данных;
- произвести предварительную обработку и нормализацию исходной выборки;
- настроить параметры НС и алгоритма ее обучения, протестировать полученную НС;
- подготовить готовый программный продукт для его дальнейшего представления и использования.

Глава 1 Анализ предметной области

В данной главе представлены статистические данные в области исследования ишемического инсульта, согласно которым было выбрано данное заболевание за основу выпускной квалификационной работы, а также основные теоретические аспекты в области искусственных нейронных сетей.

1.1 Особенности диагностики ишемического инсульта на МРТ снимках

На сегодняшний день такое заболевание, как инсульт, всё чаще встречается не только среди людей пожилого возраста, но и среди молодого населения. По данным Всемирной Организации Здравоохранения инсульт занимает 3 место в мире по числу гибели людей, инсульт и ишемическая болезнь сердца уносит большинство жизней в год, в общей сложности 15 миллионов. Последние 15 лет эти заболевания являются ведущими причинами смерти [1].

В Российской Федерации по данным СЗГМУ им И.И. Мечникова 450 тыс. человек переносят инсульт в год из них 80% составляют инсульты ишемического характера [2].

Ишемический инфаркт (инсульт) – органическое поражение ЦНС, вызванное острым нарушением мозгового кровообращения с развитием ишемии нервной ткани и появлением инфаркта, сопровождающееся характерными морфологическими проявлениями на визуализации (МРТ и КТ).

Одним из способов диагностики инсульта является МРТ головного мозга. Это безопасный метод обследования, в основе которого лежит эффект магнитного резонанса и электромагнитное излучение, по-разному отражающееся от более или менее плотных тканей. МРТ показывает

результаты сканирования пациента в трех плоскостях, а также позволяет очень четко увидеть проблемы с мягкими тканями.

Существует несколько режимов МРТ, например, T1 ВИ, T2 ВИ, T2 FLAIR, DWI и тд. Режимы предназначены для рассмотрения тканей, которые обладают разными физическими свойствами, чтобы отличить нормальные ткани от патологических.

На рисунке 1 представлены виды инсульта и интенсивность сигнала в зависимости от периода и режима МРТ.

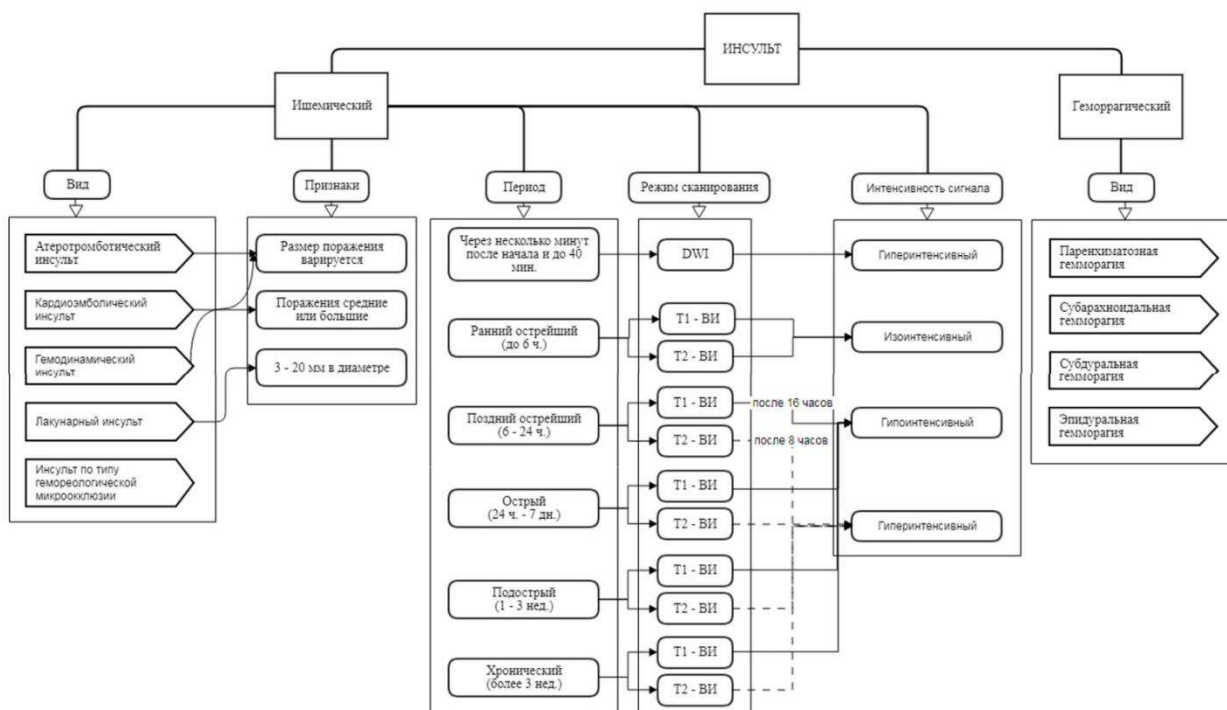


Рисунок 1 – Виды инсульта и режимы сканирования МРТ

На рисунке 1 видно, что при сканировании в первые минуты после инсульта лишь в режиме DWI (ДВИ) прослеживается гиперинтенсивный сигнал, который характеризуется светлыми участками на изображении. В режиме T1 ВИ и T2 ВИ первые признаки инсульта проявляются примерно через 6 часов после его начала, на T1 ВИ сигнал становится гипоинтенсивным, т.е. участок поражения становится тёмным, на T2 ВИ сигнал становится гиперинтенсивным. На рисунке 2 наглядно представлена сравнительная характеристика компьютерной томографии (КТ) и режимов МРТ в разные фазы развития ишемического инсульта.

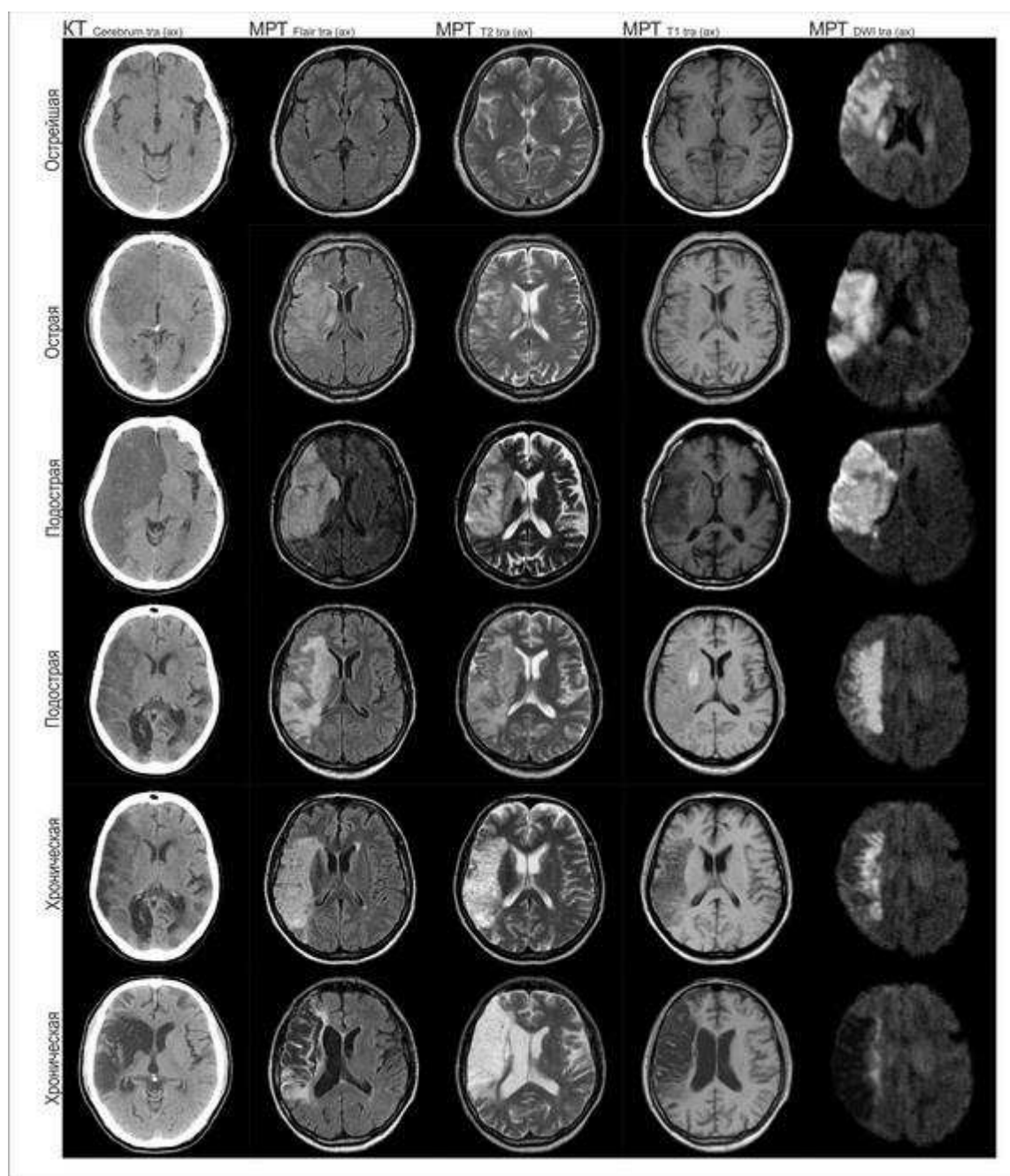


Рисунок 2 – Сравнительная характеристика КТ и режимов МРТ

Проанализировав рисунок 2, можно сделать вывод, что наиболее информативный метод диагностики ишемического инсульта является МРТ в режиме DWI. На сегодняшний день ДВИ – это один из наиболее быстрых и высокоспецифичных методов диагностики инфаркта мозга на ранних стадиях его развития (до 6 часов), когда существует «терапевтическое окно» для восстановления пораженных тканей мозга. В острой фазе мозгового инсульта на ДВИ зона поражения мозга имеет типично высокий МР-сигнал, тогда как

нормальные ткани мозга выглядят темными. На рисунке 3 представлены изображения МРТ в режиме DWI, слева на МРТ нарушение мозгового кровообращения в веществе правого большого полушария, патологическое нарушение имеет гиперинтенсивный сигнал, справа изображение здорового мозга с изоинтенсивным сигналом.

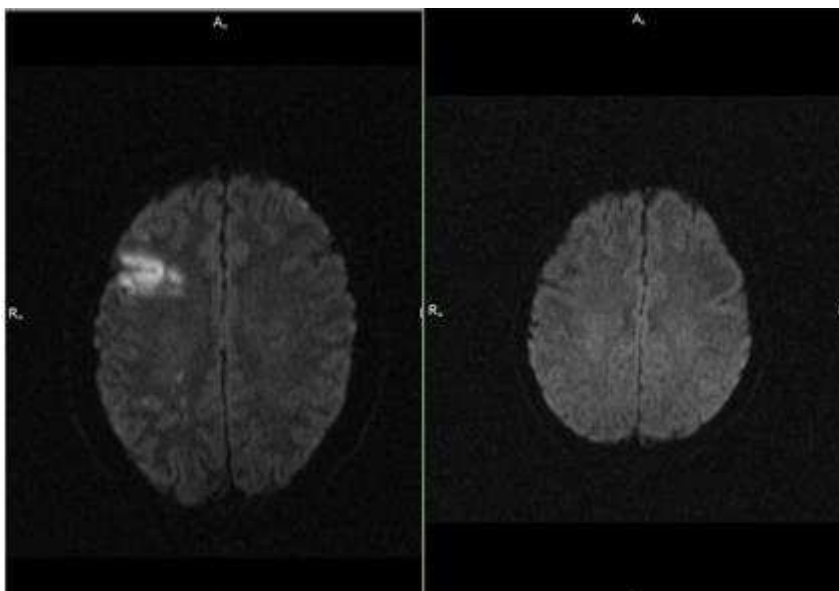


Рисунок 3 – МРТ изображение ишемического инсульта и здорового мозга

Режим ДВИ предназначен не только для выявления инсульта, но и для других патологий головного мозга, например, таких, как абсцесс мозга, солитарный метастаз, лимфома, демиелинизирующие заболевания и другие, которые могут похоже выглядеть на МРТ снимке и при неправильной диагностике могут быть распознаны как другое заболевание. Использование нейронной сети может быть методом решения данной проблемы.

1.2 Основы теории нейронных сетей

Нейронные сети вошли в практику, где необходимо решать задачи прогнозирования, классификации, управления – иными словами, в области человеческой деятельности, где есть плохо программируемые задачи, для решения которых либо нужна постоянная работа нескольких

квалифицированных экспертов, либо адаптивные системы автоматизации, каковыми и являются нейронные сети [3].

1.2.1 Искусственный нейрон

Несмотря на то, что существуют разные архитектуры ИНС, есть одинаковые элементы, например, нейрон, который имитирует функции биологического. Искусственный нейрон – ячейка НС. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены [4]. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рисунке 4 [5].

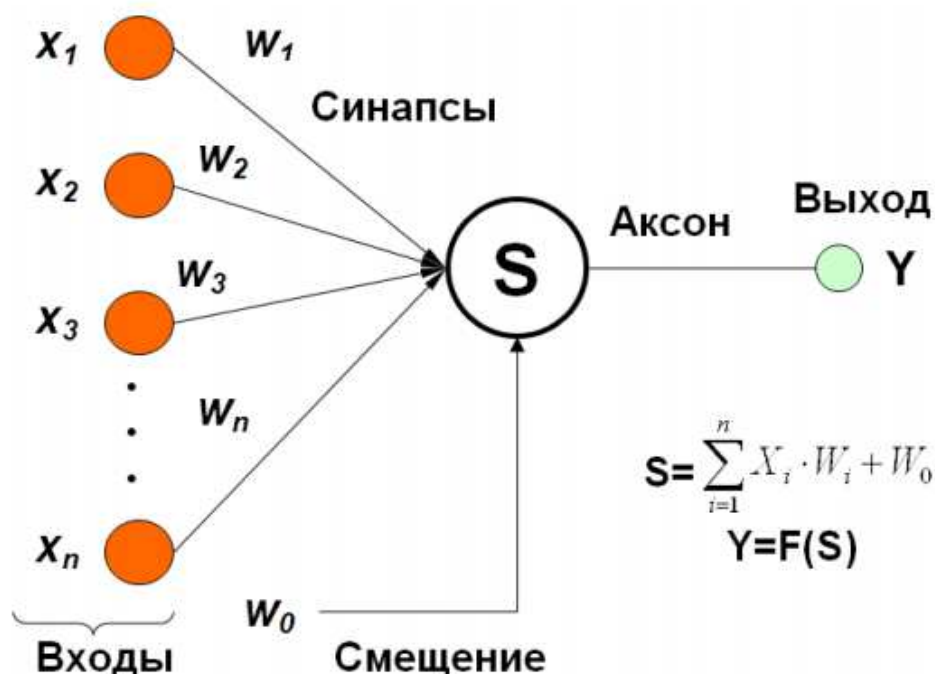


Рисунок 4 – Модель искусственного нейрона

Текущее состояние нейрона определяется, как взвешенная сумма его входов:

$$s = \sum_{i=1}^n x_i \cdot w_i \quad (1)$$

Выход нейрона есть функция его состояния:

$$y = f(s) \quad (2)$$

Нелинейная функция f называется активационной и может иметь различный вид, как показано на рисунке 5. Одной из наиболее распространенных, является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид:

$$f(x) = \frac{1}{1+e^{-ax}} \quad (3)$$

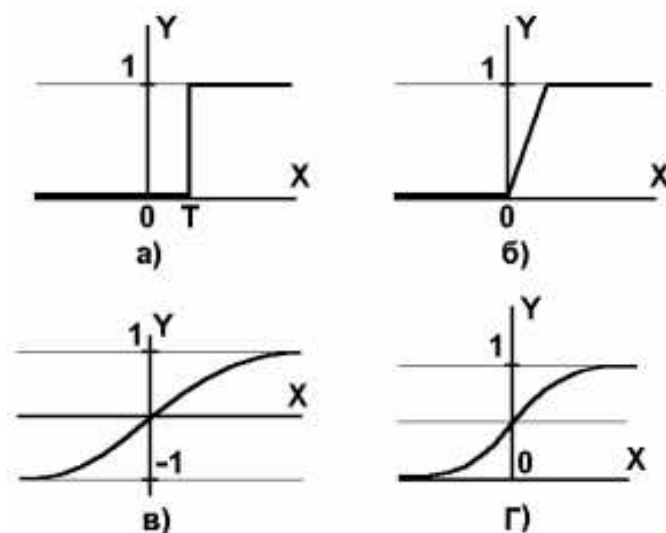


Рисунок 5 – Типы функций активации: а) функция единичного скачка; б) линейный порог (гистерезис); в) сигмоид – гиперболический тангенс; г) логистическая функция

При уменьшении α сигмоид становится более пологим, в пределе вырождаясь в горизонтальную линию на уровне 0,5. При увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с

порогом T в точке $x = 0$. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне $[0,1]$. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)) \quad (4)$$

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

1.2.2 Архитектуры нейронных сетей

Нейронная сеть – это структура связанных между собой нейронов, со взвешенными связями. Нейросети бывают прямого распространения и сети с обратными связями. На рисунке 6 представлены архитектуры сетей.

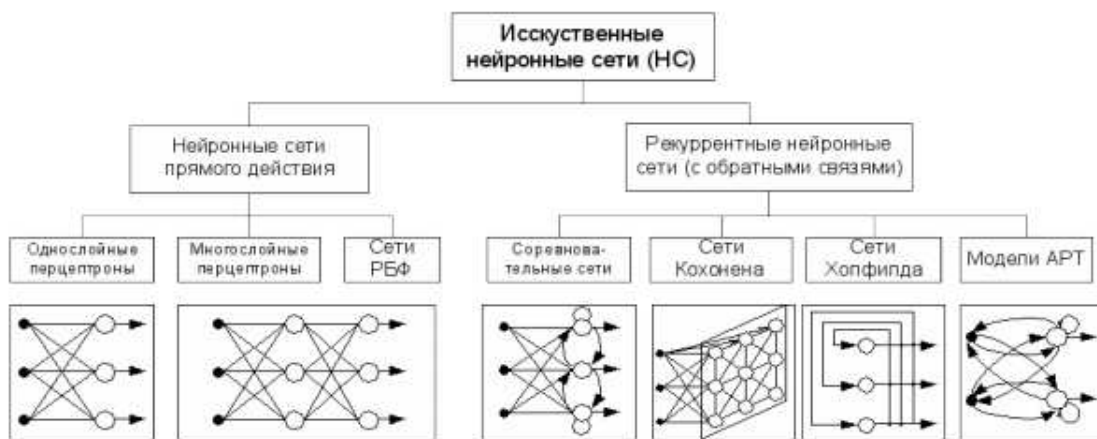


Рисунок 6 – Виды и архитектуры сетей

Существуют следующие типы сетей:

– Однослойные.

Информация со входа идет сразу на выходные нейроны, которые в свою очередь производят обработку. В такой сети только один слой нейронов. Возможности такой сети ограничены.

– Многослойные.

Более сложные и большие нейронные сети. Часто они имеют сигмоидную активационную функцию. Их главное отличие – наличие скрытых слоев, которых не видно. Однозначного решения, какое количество промежуточных слоев и нейронов выбирать, не существует, подбор параметров конфигурации сети происходит на интуитивном уровне.

– С прямыми связями.

Могут иметь один или несколько слоев. Связь между нейронами передается только от одного слоя к следующему. Данные сети не имеют память, их выход обусловлен текущими значениями входа и весов.

– С обратными связями.

Разрешаются связи между нейронами в пределах одного слоя. Сигналы передаются от выхода ко входу. Здесь выходное значение определяется текущим входом и предыдущими выходами.

В настоящее время существуют множество различных моделей нейронных сетей, некоторые из которых приведены ниже:

1. Персептрон. Это известный и популярный тип нейронных сетей с прямолинейной архитектурой. Как правило, состоят из входного слоя, одного или нескольких промежуточных слоев и выходного слоя нейронов. Информация передается от входа к выходу. Каждый нейрон обладает нелинейной передаточной функцией (чаще всего сигмоидальную). Обычно обучение такой сети производится методом обратного распространения ошибки, при котором на вход предоставляется множество 17 входных и верных выходных данных. При ее работе сравнивается выходной вектор с правильным, разница между которыми и есть величина ошибки, которая распространяется обратно по сети. Чтобы уменьшить ошибку, идёт корректирование значений весов, и сеть снова обучается.

2. Радиально базисные функции. Структура данного типа нейронных сетей складывается из трех слоев с разными функциями. Будучи

единственным, второй скрытый слой состоит из радиальных элементов. В выходном слое содержатся линейные элементы. Особенность радиально базисной сети в том, что в ней один промежуточный слой, в большинстве случаев, он имеет размерность намного большую, чем входной. Нейроны второго слоя имеют нелинейную функцию активации, преобразуя входные сигналы в скрытые. Значения весов сети равняются единицам. Нейронные сети на основе радиальных базисных функций, по сравнению с многослойным персептроном, менее точны и требуют больше входных данных. Такие сети просты по структуре и обучение происходит достаточно быстро.

3. Нейронная сеть Хопфилда. Сеть Хопфилда называют рекуррентной сетью из-за наличия обратной связи. Она состоит из одного слоя нейронов и является полносвязной, так как выход каждого связан с входами всех остальных нейронов, таким образом создается симметричная матрица связей. Во время обучения каждый входной узел становится скрытым и далее выходным. Поданный на вход произвольный сигнал проверяется на соответствие с уже известным образцом. Нейронную сеть Хопфилда можно определить, как сеть ассоциативной памяти, она может восстанавливать данные с наложенным на них шумом, или которые содержат лишь часть предъявленного исходного образа.

4. Сети Кохонена или карты самоорганизации. Для обучения этих нейросетей не требуется учитель. Они способны обучаться самостоятельно, при этом выходной вектор не играет главную роль, так как сети построены на конкурентном обучении. Нейроны выходного слоя конкурируют между собой за право стать активным. Тот нейрон, который выиграл, называют нейроном-победителем. Для организации карт можно использовать отрицательную обратную связь. Нейроны находятся в узлах двумерного пространства. Во время конкурентного процесса нейроны в узлах адаптируются под входные сигналы и их положения выстраиваются по порядку относительно друг друга так, что образуется система координат.

Местоположение нейронов в пространстве является указателем статистических признаков входных векторов. Из этого следует название «карты самоорганизации».

Также сети Конохена могут сжать большие данные, уменьшая их размерность. Часто находят применение в задачах распознавания изображений и прогнозирования.

1.2.3 Обучение нейронных сетей

Обучение НС – это процесс, в котором настраиваются параметры с помощью моделирования среды. Тип обучения определяется способом подстройки параметров. Обучение производится до тех пор, пока ошибка не достигнет нулевого значения, либо близкого к нему.

1.2.3.1 Методы обучения нейронных сетей

Различают четыре парадигмы обучения [6]:

- с учителем;
- без учителя;
- смешанная форма;
- с подкреплением.

Обучение с учителем предполагает заранее известный правильный ответ к каждому входному сигналу, веса подстраиваются так, чтобы свести ошибку к минимуму. Используется в основном для задач классификации и регрессии.

Обучение без учителя основывается на самостоятельном выделении признаков и зависимостей, и анализирует их. Данный метод в основном используется для задач кластеризации, обнаружений аномалий, ассоциаций, также автоэнкодеры используют данный тип обучения.

Смешанная форма использует небольшое количество данных с известными ответами и большой объем данных с неизвестными заранее ответами. Этот метод используется, если из данных сложно выделить важные признаки и не представляется возможным разметить все объекты.

Обучение с подкреплением предполагает тренировку алгоритма по системе поощрений. За правильные действия агент получает обратную связь с вознаграждением. Такой подход особенно полезен для обучения роботов, которые управляют автономными транспортными средствами или инвентарем на складе.

1.2.3.2 Правила обучения нейронных сетей

Существует четыре основных правила обучения нейронных сетей [7]:

1. Коррекция ошибки;

Для каждого входного примера задан требуемый выход d , который может не совпадать с реальным u . Правило обучения при коррекции по ошибке состоит в использовании разницы $(d-u)$ для изменения весов, с целью уменьшения ошибки рассогласования. Обучение производится только в случае ошибочного результата. Известны многочисленные модификации этого правила обучения.

2. Правило Больцмана;

Правило Больцмана является стохастическим правилом обучения, обусловленным аналогией с термодинамическими принципами. В результате его выполнения осуществляется настройка весовых коэффициентов нейронов в соответствии требуемым распределением вероятностей. Обучение правилу Больцмана может рассматриваться как отдельный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

3. Правило Хебба;

Правило Хебба является самым известным алгоритмом обучения нейронных сетей, суть которого заключается в следующем: если нейроны с обеих сторон синапса возбуждаются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью является то, что изменение синаптического веса зависит только от активности связанных этим синапсом нейронов.

Общая формула, описывающая правило Хебба, выглядит так:

$$\Delta W_{ij}(k) = r * x_j(k) * y_i(k), \quad (5)$$

где r – коэффициент скорости обучения.

4. Метод соревнования.

В отличие от правила Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, здесь выходные нейроны соревнуются между собой и выходной нейрон с максимальным значением взвешенной суммы является "победителем" ("победитель забирает все"). Выходы же остальных выходных нейронов устанавливаются в неактивное состояние. При обучении модифицируются только веса нейрона-«победителя» в сторону увеличения близости к данному входному примеру.

Доказано, что применение различных правил и методов обучения для разных слоев нейронной сети способствует процессу обучения.

1.2.4 Свёрточные нейронные сети

Свёрточная нейронная сеть (CNN) – специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание изображений, входит в состав технологий глубокого обучения. Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв и слоёв подвыборки. Структура сети – однонаправленная (без обратных связей), принципиально

многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов (передаточная функция) – любая, по выбору исследователя.

В отличие от обычных НС, в свёрточных сетях существует несколько видов слоёв. Краткое описание каждого из видов:

1) Свёрточный слой;

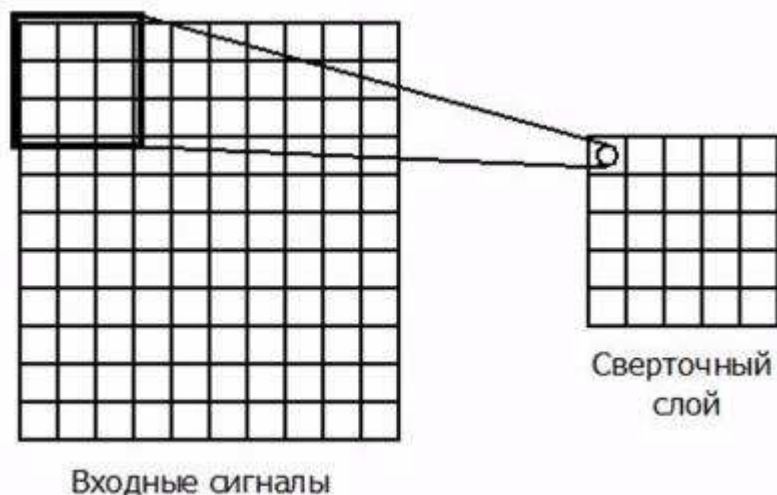


Рисунок 7 – Процесс преобразования входного изображения

В CNN нейрон свёрточного слоя получает на вход только ограниченное количество пикселей (3*3, 5*5, и тд) от некоторой области предыдущего слоя – это свойство носит название локальности зоны восприимчивости нейрона, т. е. входное изображение предыдущего слоя сканируется небольшим окном и пропускается сквозь набор синаптических коэффициентов, а результат отображается на соответствующий нейрон текущего слоя.

Для свертки рассматривается некоторая область изображения, например, 3*3, для операции необходимо ядро свёртки (матрица) такого же размера, как и участок изображения, который мы рассматриваем.

Пусть есть сигнал $x(t)$ и функция взвешивания $w(t)$, которую еще называют ядро или фильтр. Тогда операция свертки запишется как

$$s(t) = (x * w)(t) = \int x(a)w(t - a)da. \quad (6)$$

В большинстве библиотек машинного обучения реализуется несколько другая операция – кросс-корреляция:

$$S(i) = \sum_m X(i + m)K(m), \quad (7)$$

где X – вход слоя, K – фильтр

или в двухмерном случае:

$$S(i,j) = \sum_m \sum_n X(i + m, j + n)K(m, n) \quad (8)$$

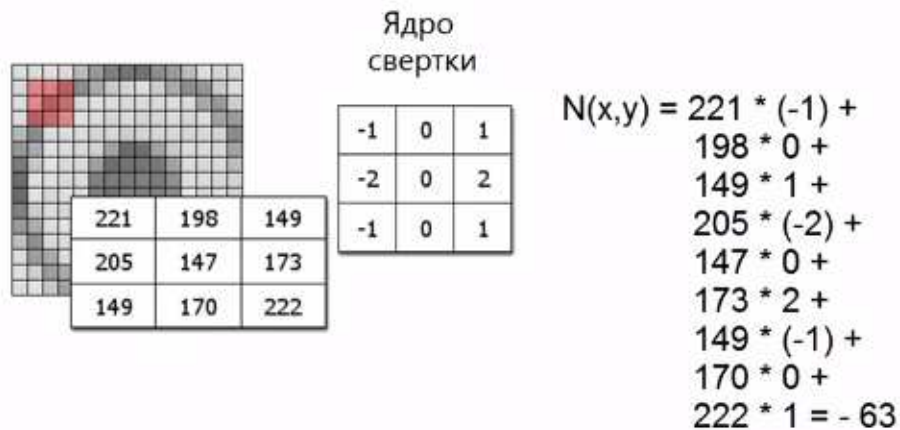


Рисунок 8 – Процесс свертки

Свёртка широко используется для обработки изображения независимо от нейронных сетей, было придумано множество ядер свертки, которые позволяют выполнять различные операции с изображениями.

Размытие

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Выделение границ

0	-1	0
-1	4	-1
0	-1	0

Повышение четкости

0	-1	0
-1	5	-1
0	-1	0

Рисунок 9 – Примеры ядер свертки для различных задач

Особенностью НС является то, что ядра свертки определяются автоматически в процессе обучения, в зависимости от того, какие данные она обрабатывает, так как подобрать ядро для нужной задачи не всегда возможно.

Следующий принцип используемый в CNN – это разделяемые веса. Все нейроны сети используют одинаковые ядра свертки для обработки разных регионов изображения, таким образом находим какой-то элемент, который важен для нас в разных участках изображения и за счёт совместного использования одних и тех же весов в ядрах свёртки, существенно уменьшается количество весов, которые нужно определить в процессе обучения.

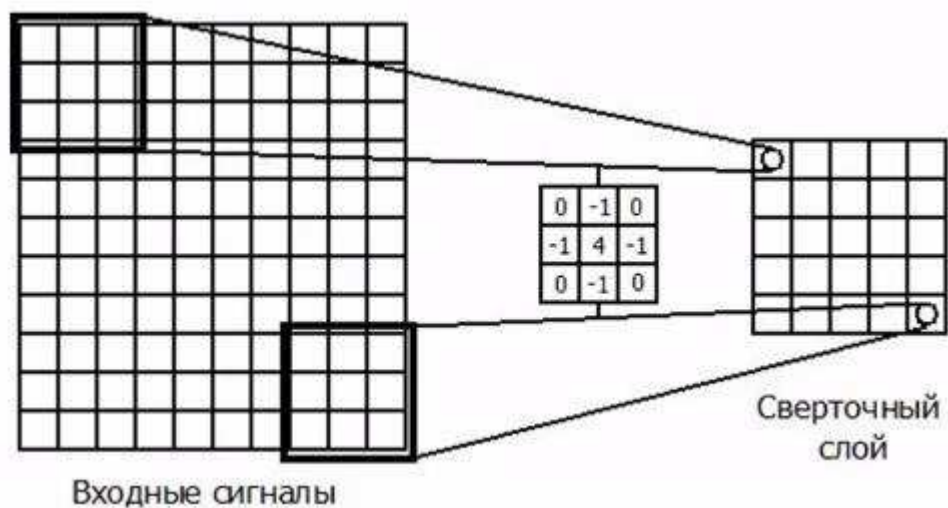


Рисунок 10 – Демонстрация принципа разделяемых весов

Главная задача одного свёрточного слоя выделение простых шаблонов с помощью фильтров. При расположении свёрточных слоёв друг за другом в сочетании с другими слоями, т.е с увеличением глубины сети, растёт сложность выделяемых признаков.

2) Активационные слои;

Выход свертчного слоя есть линейное преобразование над его входом. Композиция линейных преобразований дает линейное преобразование. Наша задача – аппроксимировать функцию высокой сложности с помощью НС, для

этого добавляется нелинейность – это и есть задача активационного слоя. В нейронных сетях обычно в качестве нелинейной функции в промежуточных слоях используют «выпрямитель» (ReLU) $f(x) = \max(0, x)$, либо её аналоги. Если использовать традиционные функции активации, такие как сигмоида или гиперболический тангенс, то это может привести на этапе обучения к проблеме размывающихся или взрывающихся градиентов.

3) Слои подвыборки (субдискретизации);

Подвыборкой называется замена элементов входа на некоторую статистику его близлежащих элементов. Основной функцией данного слоя является снижение размера данных после предыдущего слоя, для уменьшения числа параметров, количества вычислений в сети и для избегания переобучения. Для уменьшения размерности используют функции подвыборки:

- среднее по области;
- взвешенное среднее с весами, пропорциональными расстоянию до центральной точки;
- выбор максимального значения;
- и др.

Выбор функции зависит от поставленной задачи. Самая используемая функция – максимум, оператор возвращает максимум из области, которая была подана на вход. На рисунке 11 изображен пример такой подвыборки.



Рисунок 11 – Подвыборка по максимальному значению

Уменьшение размерности полезно, так как имеется возможность распознавания объекта, вне зависимости от его размера, а также если важен факт наличия признака, чем знания места его точного положения на изображении.

4) Полносвязные слои;

Скомбинированные в некоторую структуру, предыдущие слои, по сути являются генератором нового представления входных данных. Классификация производится, по полученным признакам, используя обычную НС из нескольких полносвязных слоёв. Наиболее часто используемая архитектура СНС, выглядит так:

- несколько сверточных слоёв, следующих друг за другом;
- нелинейная функция активации;
- слой подвыборки.

Такая последовательность может повторяться несколько раз.

На рисунке 12 представлен результат обучения СНС и выделенные признаки на различных уровнях.

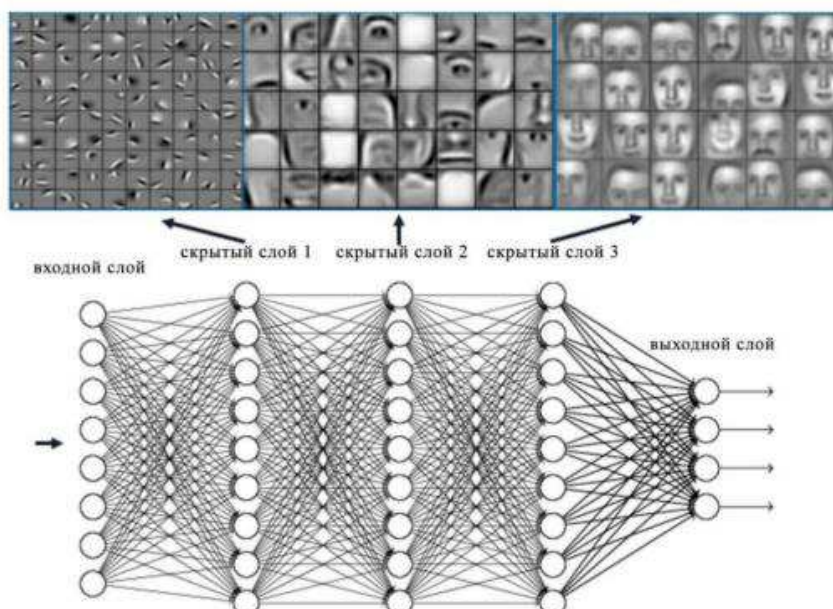


Рисунок 12 – Визуализация признаков на различных уровнях нейронной сети

Преимущества данной архитектуры:

- учитываются локальные особенности изображения;
- необходимо небольшое количество параметров, благодаря разделению веса в ядрах свёртки;
- является одним из лучших алгоритмов для распознавания и классификации изображений;
- относительная устойчивость распознавания при сдвиге и повороте распознаваемого объекта.

Недостатки:

– много различных варьируемых параметров сети, только с помощью тестов удаётся узнать какие настройки подходят для решаемой задачи. К варьируемым параметрам можно отнести: количество слоёв, размерность ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, шаг сдвига ядра при обработке слоя, необходимость слоёв субдискретизации, степень уменьшения ими размерности, функция по уменьшению размерности (выбор максимума, среднего и т. п.), передаточная функция нейронов, наличие и параметры выходной полносвязной нейросети на выходе свёрточной.

Вывод по 1 главе

В первой главе описаны особенности изображения инсульта на МРТ снимках, а также представлена основная теория по искусственным нейронным сетям. Рассмотрены такие понятия, как искусственный нейрон, искусственная нейронная сеть. Кроме того, рассмотрены архитектуры нейронных сетей, обучение нейронных сетей, а также основные принципы работы свёрточной нейронной сети.

Для задачи классификации изображений была выбрана свёрточная нейронная сеть, являющаяся лучшим алгоритмом для классификации изображений и распознавания образов.

Глава 2 Разработка нейронной сети

В данной главе будут рассмотрены основные алгоритмы и методы обработки МРТ снимков. Для каждого метода будут приведены достоинства и недостатки, а также математическое и алгоритмическое описание. В конце главы будут приведены результаты экспериментов для оценки производительности и качества работы описанных методов.

2.1 Предобработка снимков

Операции распознавания на изображениях определенных объектов, как правило, предваряются обработкой изображений для создания условий, повышающих эффективность и качество выделения и распознавания искомым или изучаемым объектам [8]. Методы предварительной обработки зависят от задач исследований; они довольно разнообразны и могут включать, например, выделение наиболее информативных фрагментов, их увеличение, получение 3-мерных изображений, цветокартирование, реализация высокого пространственного разрешения, повышение контрастного разрешения, улучшение качества изображений и т.п. Несмотря на то, что МРТ позволяет врачу увидеть те области органов, которые часто недоступны при использовании других методов таких, как КТ и рентген, однако случайные артефакты и шум могут ухудшить читаемость и качество снимка. После удаления шума и ненужных деталей, необходимо произвести выравнивание гистограммы или контрастности.

2.1.1 Масштабирование

Предобработка снимков происходила в несколько этапов. Поскольку снимки предоставлены из различных источников, их необходимо привести к единому размеру. Для этого на первом этапе было произведено масштабирование снимков до размера 160x160 пикселей с помощью библиотеки PIL (Python Imaging Library).

Масштабирование заключается в изменении вертикального и горизонтального размеров изображения. Масштабирование может быть пропорциональным – в этом случае соотношение между высотой и шириной рисунка не изменяется, а меняется общий размер, и непропорциональным – в этом случае оба измерения изменяются по-разному [9].

Масштабирование векторных рисунков выполняется просто и без потери качества. Масштабирование растровых рисунков является намного более сложным процессом, чем для векторной графики, и часто сопровождается потерей качества. При изменении размеров растрового изображения выполняется одно из следующих действий:

- одновременное изменение размеров всех пикселей (в большую или меньшую сторону);

- добавление или убавление пикселей из рисунка для отражения производимых в нем изменений, называемое выборкой пикселей в изображении.

Простейший способ изменения масштаба растрового рисунка состоит в изменении размера всех его пикселей. Так как внутри самого рисунка пиксели не имеют размера и приобретают его уже при выводе на внешнее устройство, то изменение размера пикселей раstra в сильной степени похоже на масштабирование векторных объектов – необходимо сменить только описание пикселя, а остальное выполнит устройство вывода.

Устройство вывода для создания пикселя определенного физического размера использует столько своих минимальных элементов, сколько сможет. При масштабировании изображения количество входящих в него пикселей не меняется, а изменяется количество создаваемых устройством вывода элементов, идущих на построение отдельного пикселя изображения.

Выборка растрового рисунка может быть сделана двумя различными способами.

По первому способу просто дублируется или удаляется необходимое количество пикселей. При этом в результате масштабирования, как правило, ухудшается качество изображения. Например, при увеличении размера рисунка возрастают его зернистость и дискретность. При уменьшении размера рисунка потери в качестве не столь заметны, однако при последующем восстановлении уменьшенного рисунка до прежнего размера опять возрастают зернистость и дискретность. Это связано с тем, что при уменьшении размера рисунка часть пикселей была удалена из исходного изображения и потеряна безвозвратно, а при последующем восстановлении размеров рисунка недостающие пиксели дублировались из соседних.

По второму способу с помощью определенных вычислений можно создать пиксели другого цвета, определяемого цветами первоначального пикселя и его окружения. Этот метод называется интерполяцией и является более сложным, чем простое дублирование. При интерполяции кроме дублируемых пикселей, отбираются и соседние с ними, с помощью которых вновь создаваемые пиксели получают от существующих усредненный цвет или оттенок серого. В результате переходы между пикселями становятся более плавными, что позволяет убрать или уменьшить эффект «пилообразного» изображения.

Для наших снимков был взят такой метод, как интерполяция фильтром Ланцоша [10]. Этот способ даёт большую плавность результата, но при этом время обработки значительно увеличивается.

Идея фильтра основана на применении нормированной функция $\text{sinc}(x) = \frac{\sin \pi x}{\pi x}$ с растянутым по оси x главным лепестком и равной нулю вне заданного параметром ширины α интервала.

Метод связан с оконной функцией Ланцоша, $L_\omega(x)$ представляющей собой главный лепесток функции $\text{sinc}(x)$, вне этого лепестка оконная функция равна нулю:

$$L_\omega(x) = \text{sinc}(x/\alpha) \quad (9)$$

При обработке изображений, так как они являются двумерными функциями для свертки применяется двумерное ядро Ланцоша:

$$L(x, y) = L(x)L(y) \quad (10)$$

Применение этого фильтра позволяет добиться высокой чёткости изображения, но при обработке возможно появление нежелательных артефактов. Это искажение заключается в появлении вокруг контрастных переходов яркости узких контрастных ореолов, что позволяет сохранить резкость контрастных линий при сохранении достаточной гладкости тональных переходов.

Возникновение ореолов обусловлено тем, что при значении параметра $\alpha > 1$ ядро Ланцоша принимает отрицательные значения при некоторых значениях аргумента. Поэтому обработанный сигнал может принимать даже отрицательные значения при положительных значениях выборок.

2.1.2 Средний фильтр

На втором этапе предобработки необходимо производить удаление шума, так как случайные артефакты и шум могут ухудшить читаемость и качество снимка. Поскольку подавление белого и гауссового шума более эффективно при использовании линейных фильтров для нашей системы был взят средний фильтр из библиотеки OpenCV.

Средний фильтр является примером линейного фильтра. Действие этого фильтра сводится к замене значения в точке изображения на медиану значений яркости в окрестности этой точки. Это приводит к сглаживанию изображения (уменьшению количества изменений интенсивности между пикселем и следующим), удалению шума из изображения и увеличению яркости изображения. Таким образом, при средней фильтрации каждый пиксель изображения будет заменен на среднее значение его соседей, включая сам пиксель.

Далее хотелось бы рассмотреть некоторые формулы, составляющие средний фильтр [11]. Пусть S_{xy} – некоторая окрестность размерами $g(s,t)$ и с центром в точке (x,y) . Необходимо вычислить среднее арифметическое по окрестности S_{xy} .

$$\hat{f}(x, y) = med_{st \in S_{xy}} \{g(s, t)\} \quad (11)$$

При вычислении медианы значение в самой точке также учитывается. Широкая популярность этих фильтров обоснована тем, что они прекрасно приспособлены к подавлению случайных шумов и при этом приводят к наименьшему размыванию по сравнению с другими фильтрами. Медианные фильтры особенно успешно работают в случаях импульсного шума.

На рисунке 13 наглядно продемонстрирована работа данного фильтра.



Рисунок 13 – Пример работы медианного фильтра длиной 5 единиц

В самой же программе для среднего фильтра используется ядро 3×3 , а также функция `filter2D ()` для его применения [12]. Первым параметром этой функции является наше входное изображение, второе – желаемая глубина выходного изображения `ddepth`, а третьим параметром является наше ядро. Далее присваивается -1 для параметра `ddepth`, что означает, что выходное изображение будет иметь ту же глубину, что и входное изображение.

В результате на выходе мы получаем более плавное изображение, которое имеет четкие границы патологий головного мозга.

2.2 Выбор архитектуры

Факторы, влияющие на распознавание ишемического инсульта на МРТ снимках, носят сложный характер. Поэтому построение адекватной математической модели процесса распознавания данной патологии, позволяющей с достаточной точностью диагностировать ишемический инсульт, затруднено. Это обусловлено схожестью рассматриваемой болезни с другими заболеваниями на МРТ снимках, недостаточной интенсивностью сигнала на DWI снимках и рядом других факторов.

Именно в таких случаях оправдано использование различных информационных систем, в том числе нейронных сетей, позволяющих с наибольшей достоверностью обнаружить патологию головного мозга. Однако эффективность распознавания достигается при надлежащем выборе архитектуры ИНС и процесса ее обучения. Далее пойдет речь о некоторых архитектурах, которые были опробованы для решения поставленной задачи.

2.2.1 Использование готовых слоев

На первом этапе в качестве структуры нейронной сети была использована сеть с чередующимися между собой тремя сверточными слоями и субдискретизирующими тремя слоями, которые формируют входной вектор признаков для двух полносвязных слоев. Для обучения использовался метод «обучение с учителем». Для активации использовалась функция ReLU после каждого сверточного и полносвязного слоя в качестве нелинейной функции «выпрямителя». Если использовать традиционные функции активации, такие как сигмоида или гиперболический тангенс, то это может привести на этапе обучения к проблеме размывающихся или взрывающихся градиентов.

После последнего слоя использовалась нелинейная функция с насыщением, так называемая логистическая функция или сигмоид. На этапе

обучения сеть сходится после 30 эпох. Достигнутая точность на снимках здорового головного мозга и пораженного ишемическим инсультом – 95%.

На рисунке 14 представлена архитектура используемой нейронной сети. На рисунке 15 предоставлены графики зависимости точности (слева) и ошибки (справа) от количества итераций.

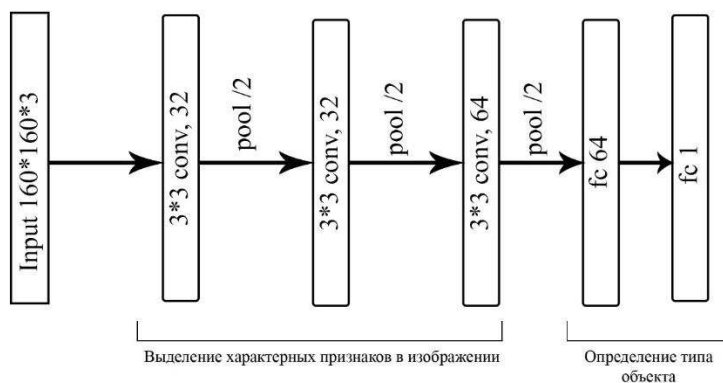


Рисунок 14 – Архитектура нейронной сети



Рисунок 15 – Графики зависимости точности и ошибки от количества итераций

Данная нейронная сеть показала себя хорошо лишь на выборке данных, состоящей из снимков здорового головного мозга и головного мозга с патологией в виде ишемического инсульта. Когда в противовес были взяты также снимки других заболеваний, точность упала до 56%, в связи с чем мы стали рассматривать другие нейронные сети.

2.2.2 ResNet50

ResNet [13] – это сокращенное название для Residual Network (дословно – «остаточная сеть»).

Когда более глубокая сеть начинает сворачиваться, возникает проблема: с увеличением глубины сети точность сначала увеличивается, а затем быстро уменьшается. Снижение точности обучения показывает, что не все сети легко оптимизировать.

Чтобы преодолеть эту проблему, Microsoft ввела глубокую «остаточную» структуру обучения. Вместо того, чтобы надеяться на то, что каждые несколько stacked layers непосредственно соответствуют желаемому основному представлению, они явно позволяют этим слоям соответствовать «остаточному». Формулировка $F(x) + x$ может быть реализована с помощью нейронных сетей с соединениями для быстрого доступа.

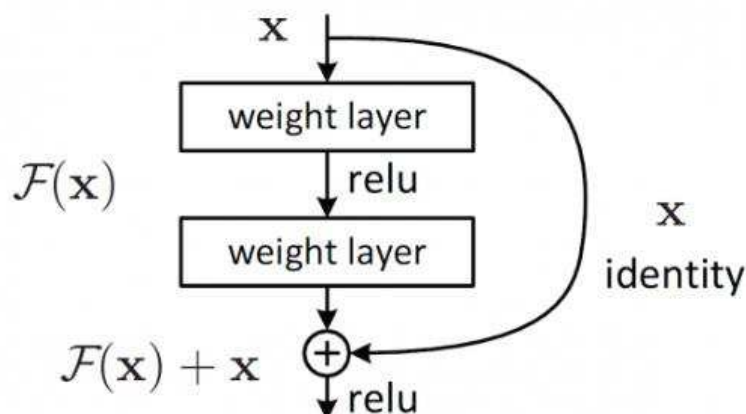


Рисунок 16 – Соединения быстрого доступа

Соединения быстрого доступа (shortcut connections) пропускают один или несколько слоев и выполняют сопоставление идентификаторов. Их выходы добавляются к выходам stacked layers. Используя ResNet, можно решить множество проблем, таких как:

– ResNet относительно легко оптимизировать: «простые» сети (которые просто складывают слои) показывают большую ошибку обучения, когда глубина увеличивается.

– ResNet позволяет относительно легко увеличить точность благодаря увеличению глубины, чего с другими сетями добиться сложнее.

Архитектура данной нейронной сети представлена на рисунке 17.

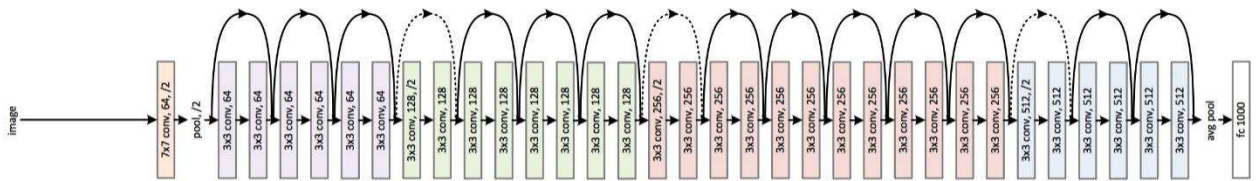


Рисунок 17 – Архитектура сети ResNet50

Однако, обучение на данной нейронной сети занимало слишком много времени, за счет того, что для ее обучения необходимо 30 и более эпох, но все же результат она выдавала достаточно низкий (таблица 1). Потому было принято решение не брать данную архитектуру для решения нашей задачи.

2.2.3 GoogLeNet

Архитектура Inception нейронной сети GoogLeNet была создана с целью снижения объема вычислений в глубоких нейросетях, а также повышения производительности [14].

На рисунке 18 представлен модуль Inception:

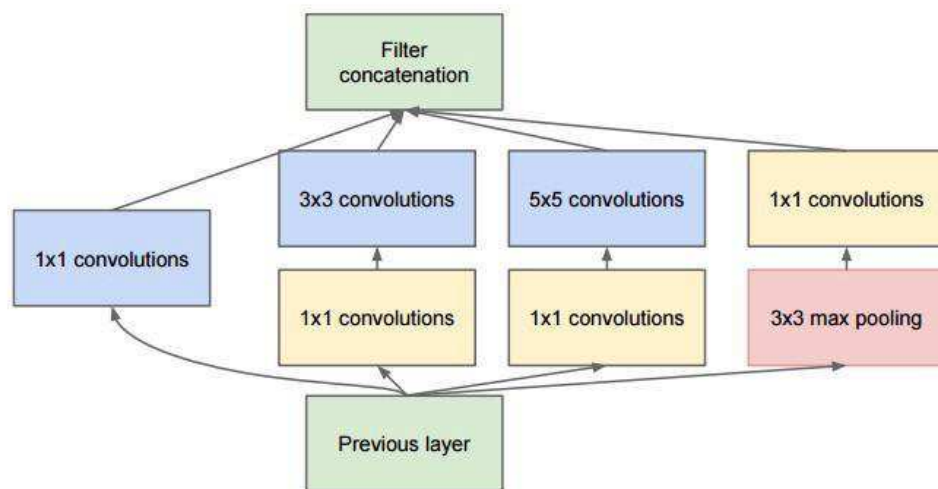


Рисунок 18 – Модуль Inception

На первый взгляд, это параллельная комбинация свёрточных фильтров 1x1, 3x3 и 5x5. Но изюминка заключалась в использовании свёрточных блоков 1x1 (NiN) для уменьшения количества свойств перед подачей в «дорогие» параллельные блоки. В GoogLeNet в качестве начального слоя применяется stem без Inception-модулей, а также используется average pooling и softmax классификатор, аналогичный NiN.

Сама же нейронная сеть состоит из 9 таких блоков. Ее архитектура представлена на рисунке 19.

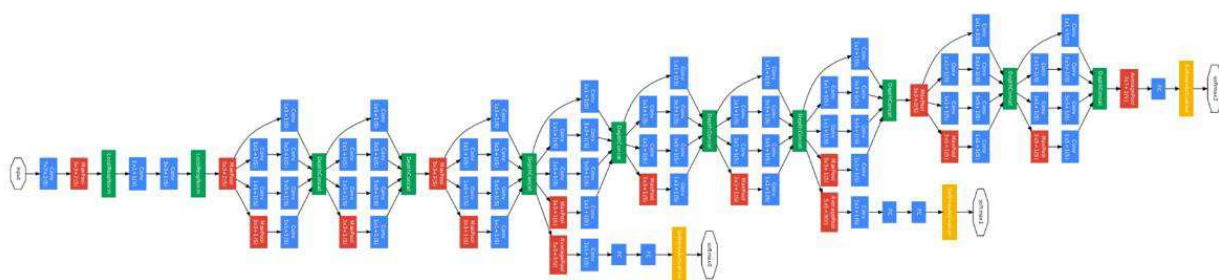


Рисунок 19 – Архитектура нейронной сети GoogLeNet

Она выиграла ImageNet recognition challenge в 2014-м году с результатом 6.67% top 5 error. Напомню, top 5 error – метрика, в которой алгоритм может выдать 5 вариантов класса картинки, и ошибка засчитывается, если среди всех этих вариантов нет правильного. Всего в тестовом датасете 150К картинок и 1000 категорий, то есть задача крайне нетривиальна.

Однако, при нашем количестве данных данная нейронная сеть вовсе не сходилась при большинстве параметров, при сходимости выдавала достаточно небольшой результат (таблица 1), потому было принято решение опробовать другие архитектуры.

2.2.4 VGG16

VGG16 [15] – модель сверточной нейронной сети, предложенная К. Simonyan и А. Zisserman из Оксфордского университета в статье “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Модель достигает точности 92.7% – топ-5, при тестировании на ImageNet в задаче распознавания объектов на изображении. Этот датасет состоит из более чем 14 миллионов изображений, принадлежащих к 1000 классам.

VGG16 – одна из самых знаменитых моделей, отправленных на соревнование ILSVRC-2014. Она является улучшенной версией AlexNet, в которой заменены большие фильтры (размера 11 и 5 в первом и втором сверточном слое, соответственно) на несколько фильтров размера 3x3, следующих один за другим. Сеть VGG16 обучалась на протяжении нескольких недель при использовании видеокарт NVIDIA TITAN BLACK.

Архитектура VGG16 представлена на рисунке ниже.

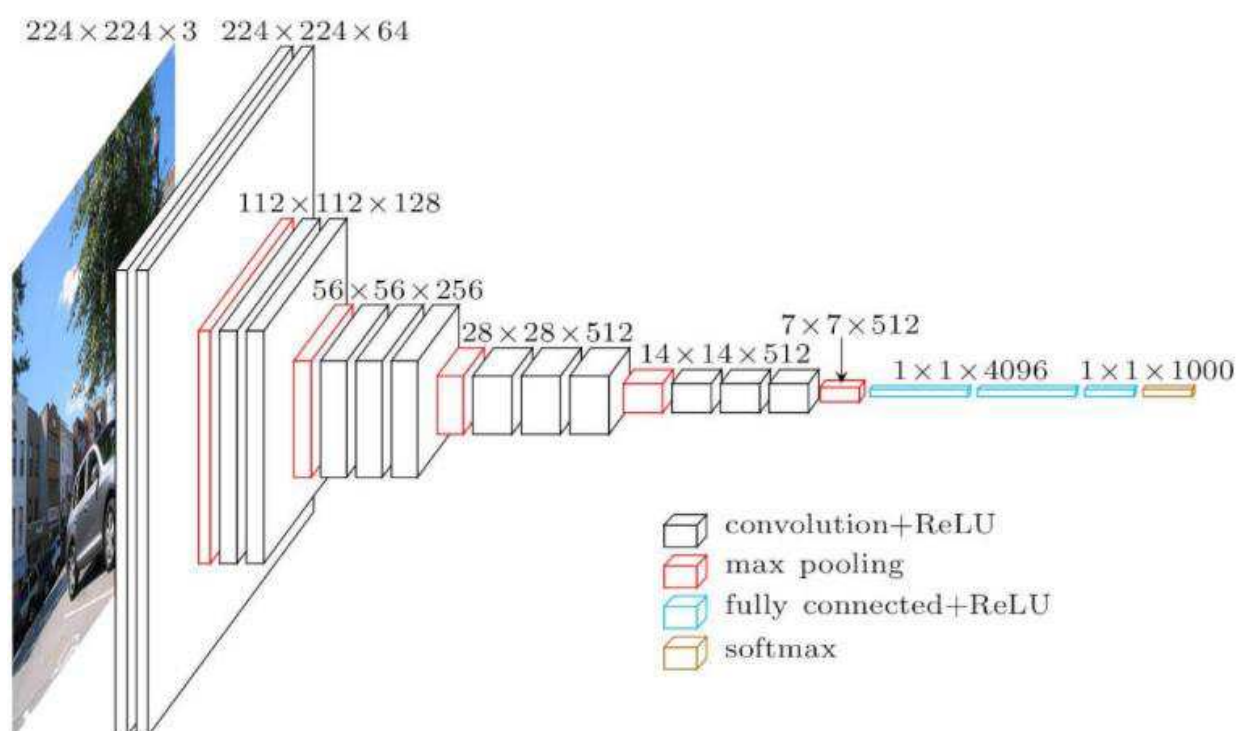


Рисунок 20 – Архитектура нейросети VGG16

К сожалению, сеть VGG имеет два серьезных недостатка:

- Очень медленная скорость обучения.
- Сама архитектура сети весит слишком много (появляются проблемы с памятью жесткого диска, видеопамятью, и пропускной способностью).

VGG16 используется для решения многих проблем классификации. Несмотря на недостатки, данная архитектура является отличным строительным блоком для обучения, так как её легко реализовать.

Был проведен ряд тестирований рассмотренных выше нейронных сетей с уже готовыми весами с различными параметрами на обработанном наборе данных. Результат тестирования приведен в таблице 1.

Таблица 1 – Сравнительная характеристика сверточных нейронных сетей

Параметры	Использование готовых слоев	ResNet50	GoogleNet	VGG16
Скорость обучения	lr=1e-5	lr=1e-5	lr=1e-5	lr=1e-5
Размер мини выборки	20	20	20	20
Количество эпох	30	5	7	7
Полученная точность обучения	61%	56,5%	73,2%	79%

При этом сошлась нейронная сеть лишь в 50% тестирований данных нейронных сетей. По результатам был сделан вывод, что, используя архитектуру VGG16 мы сможем добиться лучших результатов при большей выборке данных. Однако, точность, которую показывает нейронная сеть, составляет 79%. Разница процента ошибки и точности на обучающем и проверочном наборе данных все еще является достаточно большим – 20%. Поскольку VGG16 используется для изображений размера 224x224, при работе с изображениями размера 160x160 пикселей, на последнем 16 слое каждое в свою очередь сворачивается до размера 2x2. При данных условиях

процесс распознавания становится затруднительным. По этой причине было принято решение взять архитектуру, основанную на архитектуре VGG16, также взять те же параметры, и обучать без готовых весов.

2.2.5 Итоги выбора необходимой архитектуры

Таким образом, в качестве окончательной структуры нейронной сети была использована сеть с шестью сверточными слоями и тремя субдискретизирующими слоями, которые формируют входной вектор признаков для двух полносвязных слоев. Для активации использовалась также функция ReLU после каждого сверточного и одного полносвязного слоя. После последнего слоя использовалась нелинейная функция сигмоид. На этапе обучения сеть сходится после 6 эпох. Достигнутая точность – 84,82%.

Архитектура нейронной сети представлена на рисунке 21.

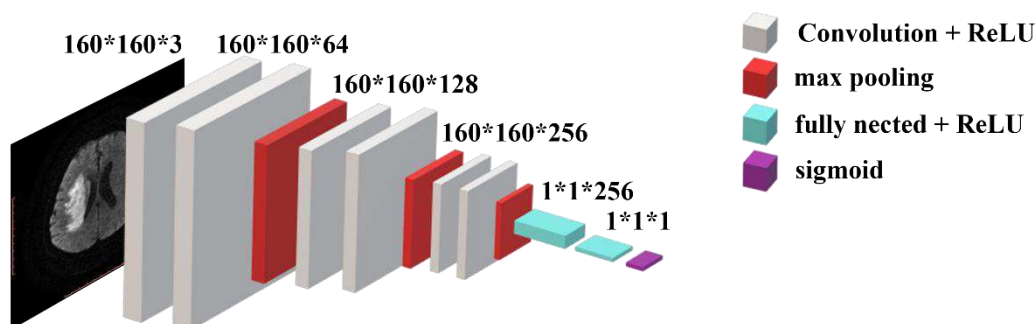


Рисунок 21 – Архитектура нейронной сети

Программная реализация нейронной сети представлена на рисунке 22.

In [26]:

```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

In [27]:

Рисунок 22 – Программная реализация нейронной сети

Далее будет представлен процесс обучения выбранной нейронной сети.

2.3 Обучение нейронной сети

Самым важным свойством нейронных сетей является их способность обучаться на основе предложенных данных и в результате обучения повышать свою производительность. Обучение нейронной сети происходит посредством интерактивного процесса корректировки синаптических весов и порогов. В идеальном случае нейронная сеть получает знания о предмете обучения на каждой итерации.

С позиций нейронной сети, вероятно, можно использовать следующее определение:

Обучение – это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки этих параметров.

Это определение процесса обучения нейронной сети предполагает следующую последовательность событий:

- В нейронную сеть поступают стимулы из внешней среды.
- В результате первого пункта изменяются свободные параметры нейронной сети.
- После изменения внутренней структуры нейронная сеть отвечает на возбуждения уже иным образом.

Наиболее простым и популярным способом обучения является метод обучения с учителем (на маркированных данных) – метод обратного распространения ошибки и его модификации. Для улучшения работы сети, повышения её устойчивости и предотвращения переобучения применяется также исключение – метод тренировки подсети с выбрасыванием случайных одиночных нейронов.

Далее будет подробно рассмотрен процесс обучения сверточной нейронной сети.

2.3.1 Выбор необходимых параметров и подготовка изображений

Особенностью свёрточного слоя является сравнительно небольшое количество параметров, устанавливаемое при обучении. Для решения задачи перебирались лишь три вида параметра:

- оптимизатор;
- размер мини-выборки;
- количество эпох.

Далее будет представлен процесс поиска необходимых величин.

2.3.1.1 Выбор оптимизатора

Процесс обучения нейронной сети заключается в подстройке весов её нейронов. Целью обучения нейронных сетей является поиск такого

состояния их весов, которое минимизирует некоторую целевую функцию на обучающей и тестирующей выборках. Задача обучения нейронных сетей достаточно сложна и это связано со следующими причинами:

- нейронные сети, которые содержат более чем один нейрон в скрытом слое, имеют не одно, а несколько оптимальных состояний, обусловленных наличием у функции ошибки нескольких минимумов, однако истинно оптимальным состоянием является только одно, где обеспечивается глобальный минимум;

- наличие множества локальных минимумов существенно затрудняет поиск глобального оптимума.

В настоящее время для решения задачи оптимизации параметров нейронных сетей разработано большое количество различных методов [16]. Обновление весов нейронной сети происходит посредством применения итерационного метода обратного распространения ошибки. Далее будут рассмотрены популярные методы оптимизации.

Градиентный спуск

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом. Тогда выражение изменения весов δw на i -ой итерации имеет следующий вид:

$$\delta w_n = -\lambda_n \nabla_{w_{n-1}} L(w_{n-1}), \quad (12)$$

где: λ_n – коэффициент скорости обучения на n -ой итерации;

$\nabla_{w_{n-1}} L(w_{n-1})$ – градиент целевой функции L относительно весов w , обновлённых на итерации $n-1$.

Метод градиентного спуска может застревать в локальных минимумах целевой функции L . Для преодоления этого используется так называемый метод моментов или инерции, который состоит в добавлении в формулу (1) термина «инерции»:

$$\delta w_n = -\lambda_n \nabla_{w_{n-1}} L(w_{n-1}) + \alpha \cdot \delta w_{n-1}, \quad (13)$$

где: α – коэффициент инерции, может принимать как фиксированное значение, так и изменяться в зависимости от λ_n , например $\alpha_n = 1 - \lambda_n$;

δw_{n-1} – изменение весов на предыдущей итерации.

AdaGrad

Суть AdaGrad состоит в масштабировании коэффициента скорости обучения для каждого параметра в соответствии с историей градиентов этого параметра. Такое масштабирование делается путём деления текущего градиента на сумму квадратов предыдущих градиентов. Формула изменения весов методом AdaGrad следующая:

$$\begin{aligned} G_n &= G_{n-1} + \left(\nabla_{w_{n-1}} L(w_{n-1}) \right)^2, \\ \delta w_n &= \frac{-\lambda_n}{\sqrt{G_n + \epsilon}} \cdot \nabla_{w_{n-1}} L(w_{n-1}), \end{aligned} \quad (14)$$

где: G_n – сумма квадратов градиентов;

ϵ – коэффициент сглаживания, $0 < \epsilon \ll 1$.

Adam

Метод Adam сочетает в себе и идею накопления движения, и идею более слабого обновления весов для типичных признаков. $m_n = \beta_1 m_{n-1} + (1 - \beta_1) \nabla_{w_{n-1}} L(w_{n-1})$,

$$v_n = \beta_2 v_{n-1} + (1 - \beta_2) \nabla_{w_{n-1}} L(w_{n-1}),$$

$$\widehat{m}_n = \frac{m_n}{1 - \beta_1^n},$$

$$\widehat{v}_n = \frac{v_n}{1 - \beta_2^n},$$

$$\delta w_n = -\lambda_n \frac{\widehat{m}_n}{\sqrt{\widehat{v}_n + \epsilon}}, \quad (15)$$

где: m_n – оценка математического ожидания;

v_n – оценка средней нецентрированной дисперсии;

β_1 и β_2 – коэффициенты накопления, $0 < \beta_1 < 1$ и $0 < \beta_2 < 1$;

\widehat{m}_n и \widehat{v}_n – скорректированные значения оценок математического ожидания и средней нецентрированной дисперсии, влияют на первых шагах обучения;

ϵ – коэффициент сглаживания, $0 < \epsilon \ll 1$.

Выбор эффективного метода проводился на основе результатов тестирования нескольких методов оптимизации по критерию скорости сходимости.

В качестве тестовой задачи была выбрана задача классификации рукописных цифр и для обучения использовалась база MNIST, состоящая из бинарных изображений. Количество изображений в обучающей и в тестирующей выборках составило 60000 и 10000 соответственно.

В эксперименте использовались четыре LeNet-подобные свёрточные нейронные сети, описание структур которых приведено в таблице 2.

Таблица 2 – Структуры сетей, используемых в проводимом эксперименте

Уровень глубины	CNN-3	CNN-4
	9226 параметров	79296 параметров
1	Conv8 5x5, ReLU, Max-Pool 2x2	
2	Conv16 5x5, ReLU, Max-Pool 2x2	
3	MatMulFC-98, ReLU	
Классификатор	MatMulFC-10, SoftMax	

Шаг обучения λ задавался для всех методов равным 0,001. Коэффициент инерции для градиентного спуска с инерцией вычислялся по следующей формуле: $1-\lambda$.

Для метода Adam параметры β_1 и β_2 устанавливались равными 0,1 и 0,001 соответственно. Значения весов смещения инициализировались константным значением равным 1. Последний слой сетей полностью инициализировался константным значением 0. Всего было проведено 50 эпох обучения для каждой сети.

На рисунках 23 и 24 приведены графики изменения значения целевой функции в процессе обучения сетей с шифрами CNN-3 и CNN-4. В таблицу 3 сведены значения целевой функции на обучающей и тестирующей выборках, обученных разными методами оптимизации свёрточных нейронных сетей.

Таблица 3 – Значения целевой функции обученных сетей

Структура сети	Градиентный спуск		Градиентный спуск с инерцией		AdaGrad		Adam	
	обуч.	тест.	обуч.	тест.	обуч.	обуч.	тест.	обуч.
CNN-3	1,136	1,108	0,140	0,130	0,370	0,343	0,050	0,049
CNN-4	2,240	2,234	0,128	0,126	0,324	0,301	0,025	0,044

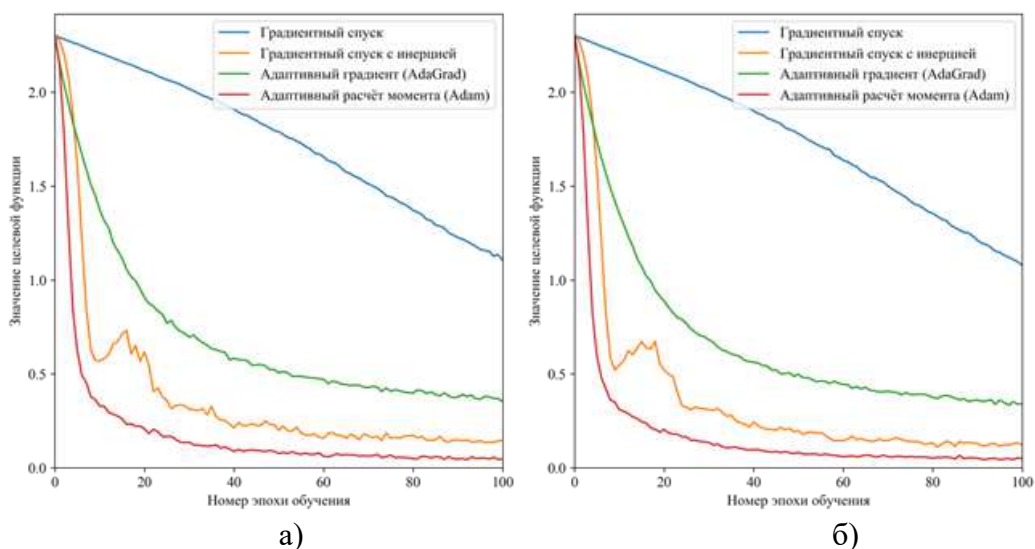


Рисунок 23 – Графики изменения целевой функции сети с шифром CNN-3

(а) – на обучающей выборке, (б) – на тестирующей выборке

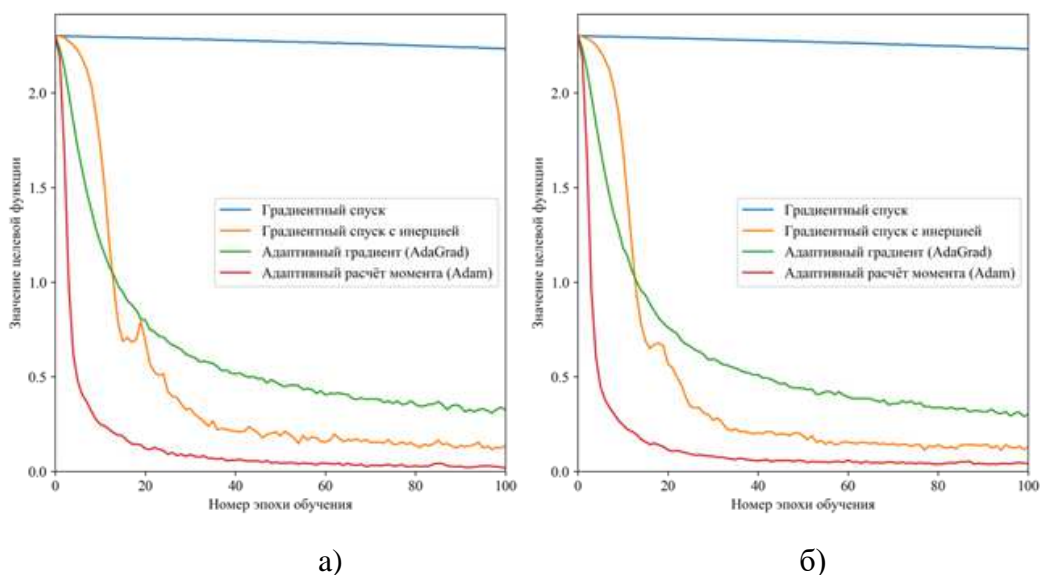


Рисунок 24 – Графики изменения целевой функции сети с шифром CNN-4

(а) – на обучающей выборке, (б) – на тестирующей выборке

Основываясь на полученных результатах, очевидно, что наиболее эффективным методом оптимизации является метод Adam. Он показал наибольшую скорость сходимости для обоих вариантов сетей по сравнению с другими методами.

На рисунке 25 представлена реализация данного метода.

```
model.compile(loss='binary_crossentropy',  
              optimizer=Adam(lr=1e-4),  
              metrics=['accuracy'])
```

Рисунок 25 – Оптимизация при помощи метода Adam

2.3.1.2 Выбор размера выборки

На следующем этапе необходимо было определиться с размером мини-выборки. Мини-выборка представляет собой размер мини порций, на которых проводится обучение. Размер мини-выборки определяет количество элементов, которые будут распространяться по сети. Обычно больший размер мини-выборки приводит к более быстрому обучению, но не всегда – к быстрой сходимости. Меньший размер мини-выборки обучает медленнее, но может быстрее сходиться. Выбор того или иного варианта определенно зависит от типа решаемой задачи. В таблице 4 представлены результаты, полученные нейронной сетью с использованием различных размеров мини-выборки.

Таблица 4 – Сравнительная характеристика различных размеров мини-выборки.

Размер мини-выборки	Полученная точность
20	81,29%
24	77,54%
32	75%
64	70%

Очевидно, что с увеличением количества данных в мини-выборке уменьшается полученная точность, поскольку общий объем выборки является достаточно небольшим. Потому был сделан вывод, что необходимо выбрать мини-выборку размером 20 элементов.

2.3.1.3 Выбор необходимого количества эпох и слоев

Эпоха [17] – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества, иными словами – это один полный проход по выборке.

Как было отмечено ранее, было принято решение взять архитектуру, основанную на архитектуре VGG16, однако, необходимым являлось подобрать меньшее количество слоев. В архитектуре VGG16 – 13 сверточных слоев. В ходе исследования количество слоев постепенно уменьшалось и тестировалось на различном количестве эпох для достижения результата, удовлетворяющего поставленным задачам: разница ошибки и точности на обучающей и проверочной выборках должна быть минимальной, а точность, в свою очередь, должна быть максимальной в данных условиях. В таблице 5 представлены результаты исследования.

Таблица 5 – Результаты тестирования нейронной сети с различным количеством эпох и слоев

Число эпох \ Число слоев	4	5	6	7	8
4	64%	71,2%	75,16%	83,04%	85%
5	67,4%	72,5%	81,56%	83,84%	84,2%
6	71,14%	79%	84,82%	82,14%	83,35%
7	73,56%	77,96%	81,15%	78,11%	82,86%
8	74%	74,13%	83,82%	81,6%	78,84%
9	78,6%	71,8%	75,38%	80,9%	79%
10	71,2%	76,15%	82,06%	84%	84,15%
11	79%	82,35%	79,19%	80,35%	86,4%

Таким образом, наилучшие результаты были получены при 11 слоях и 8 эпохах, 4 слоях и 8 эпохах, а также при 6 слоях и 6 эпохах. Однако, учитывая такой параметр, как разница ошибки и точности на обучающей и проверочной выборках, следует отметить, что в первых двух рассмотренных случаях данный параметр находится в пределах 15-20%, когда как при 6 эпохах и 6 сверточных слоях данный параметр находится в пределах 10%. Исходя из этого, можно сделать вывод, что данное соотношение является наиболее оптимальным.

На рисунке 26 определение количества эпох и размера мини-выборки.

```
: # Каталог с данными для обучения
train_dir = 'train'
# Каталог с данными для проверки
val_dir = 'val'
# Каталог с данными для тестирования
test_dir = 'test'
# Размеры изображения
img_width, img_height = 160, 160
# Размерность тензора на основе изображения для входных данных в нейронную сеть
# backend Tensorflow, channels_last
input_shape = (img_width, img_height, 3)
# Количество эпох
epochs = 6
# Размер мини-выборки
batch_size = 20
# Количество изображений для обучения
nb_train_samples = 2440
# Количество изображений для проверки
nb_validation_samples = 112
# Количество изображений для тестирования
nb_test_samples = 112
```

Рисунок 26 – Определение количества эпох и размера мини-выборки

Кроме того, перед обучением необходимо подготовить набор данных. Keras содержит средства для автоматической загрузки изображений и преобразование их в вид, пригодный для обучения нейронной сети с помощью генераторов. Для генераторов используется класс Keras `ImageDataGenerator()`, который может преобразовывать изображения. Для поставленной задачи используется один тип преобразований: каждый элемент изображения делится на 255. Таким образом все элементы изображения будут находиться в диапазоне от 0 до 1, что удобно для обучения нейронной сети. Затем используется метод `flow_from_directory()`:

поток данных из каталога, в котором находятся файлы. Необходимо указать каталог, размер изображений, размер мини-выборки. Также параметру `class_mode` присваивается бинарный тип, поскольку нейронная сеть делит изображения на два класса. Реализация данного процесса представлена на рисунке 27.

```
datagen = ImageDataGenerator(rescale=1. / 255)

In [29]:

train_generator = datagen.flow_from_directory(
    train_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

Found 2440 images belonging to 2 classes.
```

Рисунок 27– Реализация подготовки изображений

Далее будут предоставлены результаты обучения с подобранными параметрами.

2.4 Результат обучения нейронной сети

Результатом обучения рассматриваемой нейронной сети выступает точность поставленного нейронной сетью «диагноза» и ее возможность отличать ишемический инсульт от других заболеваний.

2.4.1 Карта признаков нейронной сети

Работа свёрточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к ещё более абстрактным деталям вплоть до выделения понятий высокого уровня [18]. При этом сеть самонастраивается и вырабатывает сама необходимую иерархию абстрактных признаков

(последовательности карт признаков), фильтруя маловажные детали и выделяя существенное.

Подобная интерпретация носит скорее метафорический или иллюстративный характер. Фактически «признаки», вырабатываемые сложной сетью, малопонятны и трудны для интерпретации настолько, что в практических системах не особенно рекомендуется пытаться понять содержания этих признаков или пытаться их «подправить». Вместо этого рекомендуется усовершенствовать саму структуру и архитектуру сети, чтобы получить лучшие результаты. Так, игнорирование системой каких-то существенных явлений может говорить о том, что-либо не хватает данных для обучения, либо структура сети обладает недостатками, и система не может выработать эффективных признаков для данных явлений.

На рисунке 28 продемонстрирован результат выделения нейронной сетью признаков на предложенных нами снимках.

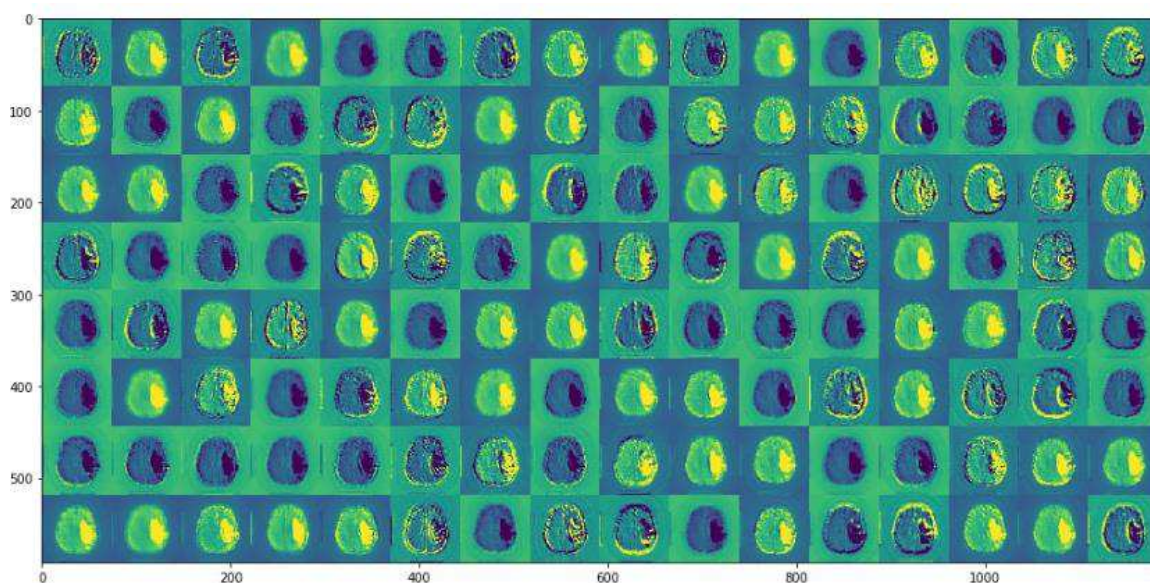


Рисунок 28 – Карта признаков на 7 слое

2.4.2 Результаты обучения

На рисунке 29 представлен процесс обучения нейронной сети с подобранными параметрами, 6 слоями и 6 эпохами.

```
history=model.fit_generator(  
    train_generator,  
    steps_per_epoch=nb_train_samples // batch_size,  
    epochs=epochs,  
    validation_data=val_generator,  
    validation_steps=nb_validation_samples // batch_size)
```

```
Epoch 1/6  
4/4 [=====] - 0s 98ms/step - loss: 0.8928 - acc: 0.  
6518  
77/77 [=====] - 16s 211ms/step - loss: 0.5193 - ac  
c: 0.7783 - val_loss: 0.8928 - val_acc: 0.6518  
Epoch 2/6  
4/4 [=====] - 0s 71ms/step - loss: 0.5861 - acc: 0.  
6786  
77/77 [=====] - 16s 205ms/step - loss: 0.4456 - ac  
c: 0.8225 - val_loss: 0.5861 - val_acc: 0.6786  
Epoch 3/6  
4/4 [=====] - 0s 71ms/step - loss: 0.4769 - acc: 0.  
7500  
77/77 [=====] - 16s 205ms/step - loss: 0.3803 - ac  
c: 0.8467 - val_loss: 0.4769 - val_acc: 0.7500  
Epoch 4/6  
4/4 [=====] - 0s 72ms/step - loss: 0.3724 - acc: 0.  
8393  
77/77 [=====] - 16s 205ms/step - loss: 0.3221 - ac  
c: 0.8672 - val_loss: 0.3724 - val_acc: 0.8393  
Epoch 5/6  
4/4 [=====] - 0s 73ms/step - loss: 0.3457 - acc: 0.  
8661  
77/77 [=====] - 16s 205ms/step - loss: 0.2875 - ac  
c: 0.8869 - val_loss: 0.3457 - val_acc: 0.8661  
Epoch 6/6  
4/4 [=====] - 0s 71ms/step - loss: 0.2967 - acc: 0.  
8750  
77/77 [=====] - 16s 205ms/step - loss: 0.2532 - ac  
c: 0.8963 - val_loss: 0.2967 - val_acc: 0.8750
```

Рисунок 29– Обучение нейронной сети

На рисунке 30 представлены результаты обучения.

In [33]:

```
scores = model.evaluate_generator(test_generator, nb_test_samples // batch_size)
```

In [34]:

```
print("Аккуратность на тестовых данных: %.2f%%" % (scores[1]*100))
```

```
Аккуратность на тестовых данных: 84.82%
```

Рисунок 30 – Результаты обучения

Достигнутая точность на этапе обучения – 84,82%. На рисунке 31 представлены графики зависимости точности (слева) и ошибки (справа) от количества итераций.

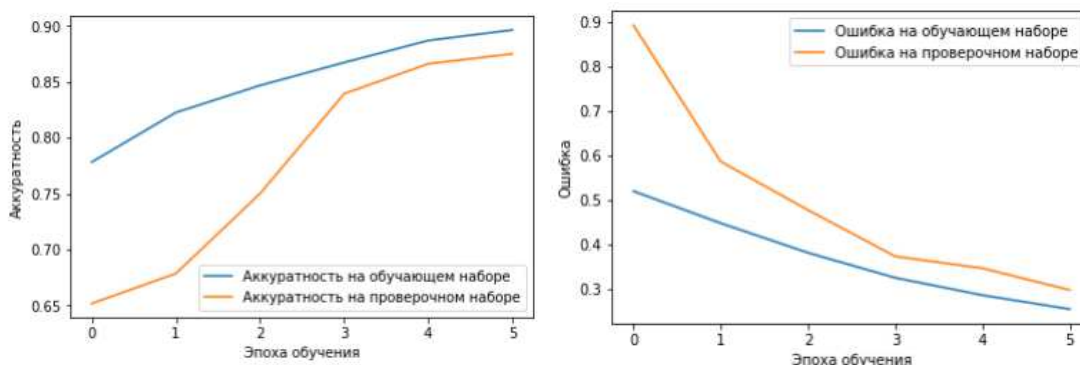


Рисунок 31 – Графики зависимости точности и ошибки от количества итераций

Анализ полученных результатов показывает, что использование построенной нейронной сети совместно с разработанными сценариями предобработки снимков позволяет говорить об успешной работе классификатора при распознавании ишемического инсульта.

Таким образом, сверточная нейронная сеть, основанная на архитектуре VGG16, сходится при следующих параметрах:

- Оптимизатор Adam с параметрами $lr=1e-5$ (скорость обучения).
- Размер мини-выборки в 20 снимков.
- 5 эпох для обучения при 16 слоях нейронной сети.

По мере увеличения количества эпох точность определения возрастает, вероятность же ошибки в свою очередь уменьшается. При этом точность, которую показывает нейронная сеть, составляет 84,82%.

Вывод по 2 главе

Таким образом, во второй главе рассмотрен процесс выбора нейронной сети, подбор необходимых параметров, а также обучение сверточной нейронной сети, основанной на архитектуре VGG16, и его результаты.

Анализ полученных результатов показывает, что использование построенной нейронной сети совместно с разработанными сценариями предобработки снимков позволяет говорить об успешной работе классификатора при распознавании ишемического инсульта.

Глава 3 Экспериментальные исследования

Довольно часто для принятия медицинских решений характерны ограниченность временных ресурсов, отсутствие возможности привлечения компетентных экспертов, неполнота информации о состоянии больного, а также в целом недостаточность знаний. Данные факторы являются причинами врачебных ошибок, которые могут привести к плачевным последствиям. Учитывая то, что у радиологов МРТ центров имеется лишь минимальная информация о пациенте, а также многие патологии действительно схожи на МРТ снимках, в частности, ишемический инсульт имеет схожие признаки с рядом болезней, процесс диагностирования является затруднительным. Поэтому достаточно важным является задача создания медицинских систем поддержки принятия решений (СППР), которые являются информационными системами, способными облегчить процесс постановки диагноза.

В рамках данной работы для данной цели была выбрана сверточная нейронная сеть, обученная на наборе данных, состоящем из МРТ снимков болезней, схожих между собой, а также для нее были подобраны параметры, при которых оптимальным является соотношение сложности вычислений и желаемой точности.

Далее будет рассмотрен процесс тестирования полученной нейронной сети.

3.1 Набор данных

Для исследования использовались снимки, предоставленные центром МРТ-диагностики на базе «ЛДЦ МИБС» им. С.М. Березина г. Красноярска, а также медицинским центром «БиКей Медика». Исходными данными для обучения нейронной сети являлись МРТ снимки в DWI формате (диффузионно-взвешенные изображения), поскольку на такого рода снимках

в острой фазе ишемического инсульта по данным исследований у 84,6% пациентов область поражения имела гиперинтенсивный МР-сигнал. Иными словами, использование диффузионно-взвешенных изображений раньше и более точно определяет участки острой ишемии головного мозга [19].

При обучении на снимках головного мозга здорового и пораженного ишемическим инсультом нейронная сеть показывала результат в 95%, поэтому было принято решение рассмотреть и другие заболевания, которые имеют схожий сигнал с рассматриваемой патологией.

Нейронная сеть обучалась на снимках таких заболеваний, как:

- ишемический инсульт;
- энцефалит;
- глиобластома;
- геморрагический инсульт;
- глиальные опухоли;
- понтинный миелинолиз;
- абсцесс;
- рассеянный склероз.

По итогу, для обучения в рамках работы мы имеем 2130 снимков иных болезней, поражающих головной мозг, и 570 снимков ишемического инсульта.

На рисунках 32-34 представлены примеры использованных изображений.

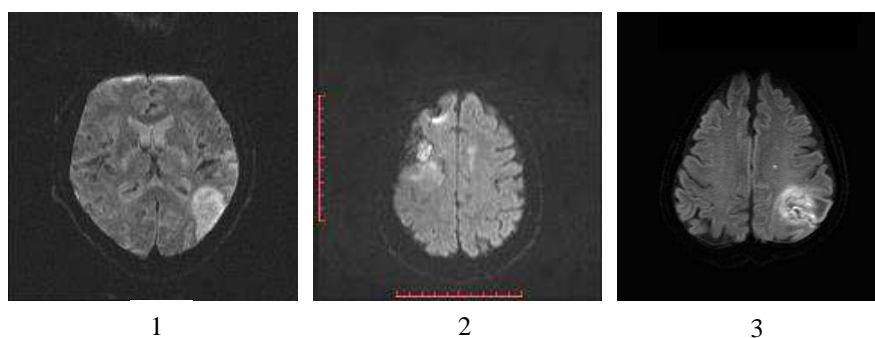


Рисунок 32. (а) – ишемический инсульт, (б) – энцефалит, (в) – глиобластома

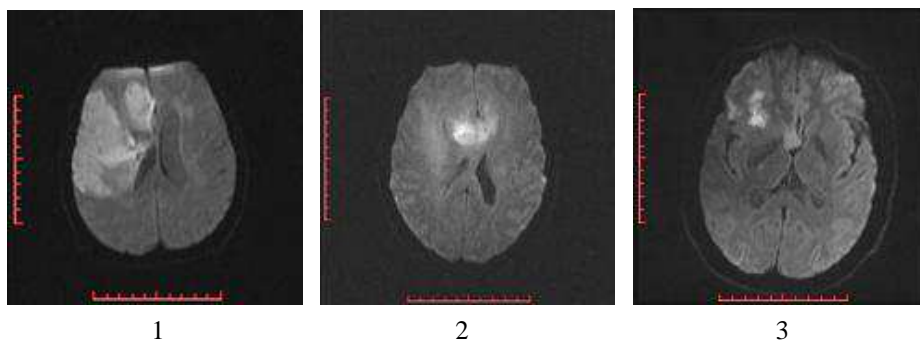


Рисунок 33. (а) – геморрагический инсульт, (б) – глиальные опухоли, (в) – ПОНТИННЫЙ МИЕЛИНОЛИЗ

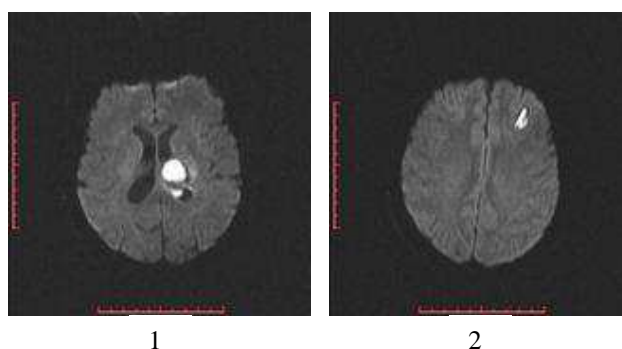


Рисунок 34. (а) – абсцесс, (б) – рассеянный склероз

На рисунке 35 представлена гистограмма, детально показывающая, какое количество снимков каждого из заболеваний головного мозга было взято для исследования.

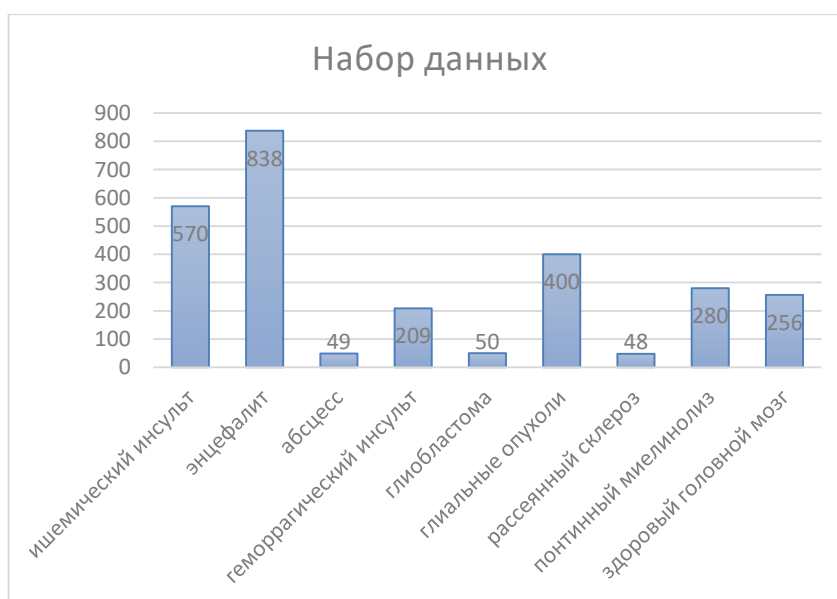


Рисунок 35 – Набор данных

Таким образом, в данном наборе данных количество изображений исследуемой патологии в 3 раза меньше, чем количество снимков иных заболеваний. Данный набор данных был подвергнут выбранному сценарию предобработки, с целью подготовить данные для обучения. Далее, на данном этапе, нейронная сеть была обучена на полном наборе данных, после чего появилась возможность приступить к самому эксперименту.

3.2 Результаты экспериментального исследования

Разработка нейронной сети проводилась на языке Python, с помощью таких инструментов, как Jupyter-Notebook и JetBrains PyCharm Community Edition, ML-framework TensorFlow (рисунок 36).

```
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Conv2D, MaxPooling2D
from tensorflow.python.keras.layers import Activation, Dropout, Flatten, Dense
from tensorflow.python.keras.optimizers import Adam
from tensorflow.python.keras import utils
from tensorflow.python.keras.callbacks import TensorBoard
import numpy as np
import matplotlib.pyplot as plt
from scipy.misc import toimage
%matplotlib inline
```

Рисунок 36 – Используемые библиотеки

Для оценки качества работы нейронной сети после завершения обучения, необходимым является провести эксперимент, в ходе которого удастся оценить качество обучения. В рамках эксперимента было взято с Интернет-ресурсов 50 снимков для одного класса – инсульт, и 50 снимков для второго – не инсульт. Также, были подобраны снимки энцефалита, поскольку данное заболевание является наиболее сложным в диагностировании, и, порой, даже для врачей достаточно затруднительно определить, что именно выявлено на снимках.

На первом этапе эксперимента выбранный снимок подвергся предобработке по выбранному сценарию: масштабирование с помощью

интерполяции Лацоша, реализованного функцией ANTIALIAS из библиотеки Pillow, а также были удалены со снимка случайные артефакты и шум с помощью среднего фильтра. Реализация представлена на рисунке 37.

```
In [24]: new_width = 160
         new_height = 160

         for infile in glob.glob("abs1/IMG-0001-00050.jpg"):
             file, ext = os.path.splitext(infile)
             im = Image.open(infile)
             im = im.resize((new_width, new_height), Image.ANTIALIAS)
             f = misc.face(gray=True)
             f = f[230:290, 220:320]
             noisy = f + 0.4 * f.std() * np.random.random(f.shape)
             med_denoised = ndimage.median_filter(noisy, 1)
             im.save(file + ".jpg", "JPEG")
```

Рисунок 37 – Программная реализация предобработки снимка

Далее загружались обученные веса, а также само изображение, после чего оно преобразовалось в массив для распознавания и происходила нормализация изображения; следующим шагом была добавлена еще одна размерность, т.к. Keras рассчитан на работу со списком данных, а не с отдельными элементами. Реализация представлена на рисунке 38.

```
|: x = image.img_to_array(img)
   x /= 255
   x = np.expand_dims(x, axis=0)
```

Рисунок 38 – Реализация преобразования массива

На рисунке 39 представлено загруженное изображение для дальнейшего распознавания.

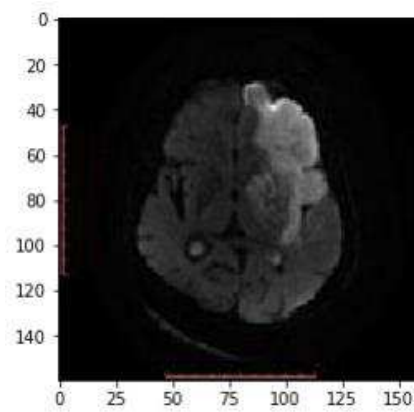


Рисунок 39 – Изображение, подготовленное для распознавания

На следующем этапе запускалось и само распознавание с помощью вызова метода `predict()` у модели. В результате нейронная сеть выдает значение, которое показывает вероятность принадлежности данного изображения определенному классу. По окончании мы имеем заключение: указана граница, которая эквивалентна значению 0,5: если значение, определенное нейронной сетью меньше 0,5, то на снимке изображен не инсульт, иначе – инсульт. Результат выполнения представлен на рисунке 40.

```
prediction = loaded_model.predict(x)
print(prediction)
print(classes[np.argmax(prediction)])

[[0.1538747]]
не инсульт

In [58]:
if prediction[[0]] < 0.5:
    print('Not a stroke')
else:
    print('Stroke')

Not a stroke
```

Рисунок 40 – Результат распознавания построенной НС

По такому алгоритму в рамках проведенного эксперимента были подобраны в общей сложности 100 снимков, относящихся к двум классам. Полученные при распознавании значения представлены в таблице 6.

Таблица 6 – Результаты экспериментального эксперимента

Вероятность принадлежности к классу «инсульт», %	Инсульт					Не инсульт				
	94	77	93	50	56	1	9	3	30	11
99	32	58	32	88	9	8	1	14	6	
98	68	75	60	66	4	6	4	1	28	
99	7	10	33	52	7	16	23	23	8	
99	21	72	65	68	7	5	4	8	18	
99	8	74	25	70	1	1	6	3	17	
92	81	77	80	5	18	31	45	13	34	
69	82	61	20	85	7	3	41	13	5	
18	93	89	29	87	12	8	19	44	6	
34	85	63	78	81	27	85	6	6	11	

Среднее значение при распознавании ишемического инсульта на снимках с данной патологией: 63,14.

Среднее значение при распознавании ишемического инсульта на снимках иных заболеваний: 14,32.

По результатам данного исследования можно сделать вывод, что в 63% случаев на МРТ снимках головного мозга, пораженного ишемическим инсультом, обученная нейронная сеть работает корректно. Однако, бывают такие ситуации, при которых нейронная сеть показывает результат в менее чем 10-20%. Это связано с тем, что на стадии сбора данных было найдено минимальное количество снимков со стволовым ишемическим инсультом, вызываемым разрывом сосудов в области ствола мозга (15 снимков), а также снимков лакунарного ишемического инсульта, вследствие которого образуется полость или размягчение в тканях серого вещества головного мозга (13 снимков). Именно такого рода снимки подавались на обученную нейронную сеть в ситуациях с плохими показателями. Наиболее частыми в имеющемся наборе данных являются снимки ишемического инсульта в бассейне средней мозговой артерии. В случае с использованием для проверки НС данных снимков мы видим результат в 97-99%.

Анализируя полученные результаты по распознаванию ишемического инсульта на МРТ снимках иных заболеваний, можно сделать вывод, что классификатор работает успешно. Из 50 проведенных экспериментов нейронная сеть ошиблась лишь один раз. На снимке, который ошибочно был отнесен к классу «ишемический инсульт», присутствовал абсцесс. Как было сказано ранее, в наборе данных для обучения присутствует лишь 49 изображений данного заболевания головного мозга. Учитывая схожесть данных заболеваний, следует отметить, что такого количества снимков недостаточно для безошибочной работы классификатора.

Таким образом, из 100 экспериментов построенная нейронная сеть корректно классифицировала изображения в 86 случаях. В 14 случаях нейронная сеть ошиблась лишь по причине недостатка данных для обучения. Анализ полученных результатов показывает, что использование построенной нейронной сети совместно с разработанными сценариями предобработки снимков позволяет говорить об успешной работе классификатора при распознавании ишемического инсульта.

3.3 Перспективы развития

На следующем этапе необходимым является рассмотреть возможность создания графического интерфейса, работающего на основе построенного алгоритма классификации.

Было проведено исследование, в рамках которого была собрана фокус-группа врачей-радиологов, и каждый дал свои оценки и высказал мнение о разрабатываемой системе, и было выявлено, что для поставленной задачи это является лишним, поскольку наиболее оптимальным вариантом на данный момент является внедрение в уже существующую систему по ряду причин.

Наиболее весомой причиной выступает наличие в существующем программном обеспечении значений ИКД (измеряемых коэффициентов диффузии), которые рассчитываются автоматически с помощью

программного обеспечения, а затем отображаются в виде параметрической карты, которая отражает степень диффузии молекул воды в различных тканях. Потому, внедряя разработанную систему в уже существующую, мы добьемся наиболее высоких результатов, поскольку появится возможность обучать нейронную сеть, учитывая данные значения по причине того, что высокий МР-сигнал на изображениях ДВИ подтверждается патологическим ограничением диффузии на ИКД картах/изображениях.

Также, разрабатывая дополнительное приложение, появляется риск лишь увеличить время постановки диагноза, поскольку врачу необходимо будет работать с двумя различными приложениями по следующему сценарию: выгрузка архива с изображениями по конкретному пациенту из приложения eFILM, поиск из другого приложения каталога с экспортированными данными, а также импорт конкретного изображения из всей серии в разработанную систему. По результатам исследования 7 из 7 врачей-радиологов отметили, что в виду недостатка времени данное приложение не будет пользоваться спросом. Однако, если появится возможность добавить расширение в имеющееся приложение, это упростит задачу постановки диагноза, а также оно будет выполнять свою главную задачу: поддержка принятия решения при постановке диагноза.

Вывод по 3 главе

В 3 главе были рассмотрены основные моменты экспериментального исследования на основе построенного алгоритма классификации изображений, а также представлены результаты экспериментов.

У построенной нейронной сети есть некоторые недостатки. Основным недостатком является небольшая точность по причине недостатка данных. Для решения этой проблемы необходимо увеличить количество данных в 10 раз. Также необходимо набрать минимум 500 снимков на каждый отдел мозга, поскольку при распознавании нейронной сетью на изображениях с такими видами ишемического инсульта, как лакунарный и стволовой, нейронная сеть показывает результат в менее чем 10-20%. Однако, основываясь на среднем значении точности определения ишемического инсульта на МРТ снимках, можно говорить об успешной работе классификатора.

ЗАКЛЮЧЕНИЕ

На основании проведенного исследования можно сделать следующие основные выводы по теме.

Для решения задачи классификации объектов на изображении был выбран такой алгоритм, как сверточная нейронная сеть. Сверточная нейронная сеть – специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание изображений, входит в состав технологий глубокого обучения. Сверточная нейронная сеть, а также ее модификации, являются лучшими по точности и скорости алгоритмами распознавания объектов на изображениях на данный момент.

В качестве архитектуры нейронной сети была использована сеть с шестью сверточными слоями и тремя субдискретизирующими слоями, которые формируют входной вектор признаков для двух полносвязных слоев. Для активации использовалась функция ReLU после каждого сверточного и одного полносвязного слоя. После последнего слоя использовалась нелинейная функция сигмоид. На основе результатов исследования можно сделать вывод о том, что построенная нейронная сеть сходится при следующих параметрах:

- оптимизатор Adam с параметрами $lr=1e-5$ (скорость обучения);
- размер мини-выборки в 20 элементов;
- 6 эпох;
- 6 сверточных слоев.

Практическая значимость нашего исследования состоит в том, что выборка для обучения состояла из МРТ снимков в DWI формате с различными заболеваниями головного мозга, что повышает актуальность данной работы. Были взяты снимки тех заболеваний, которые даже врач-радиолог способен не отличить от рассматриваемой патологии:

- энцефалит;
- глиобластома;

- геморрагический инсульт;
- глиальные опухоли;
- понтинный миелинолиз;
- абсцесс;
- рассеянный склероз.

Для тестирования нейронной сети был проведен эксперимент, в рамках которого для распознавания было взято 50 снимков головного мозга с ишемическим инсультом, а также 50 снимков с иными болезнями, при этом было выявлено, что построенная нейронная сеть способна распознать ишемический инсульт, либо же его отсутствие, в 86% случаев. На основании полученных результатов можно прийти к выводу, что разработанный алгоритм классификации изображений, совместно с разработанными сценариями предобработки снимков позволяет говорить об успешной работе классификатора при распознавании ишемического инсульта.

СПИСОК СОКРАЩЕНИЙ

CNN (СНС) –	Convolutional Neural Network (сверточная нейронная сеть)
DWI (ДВИ) –	Diffusion Weighted Imaging (диффузионно-взвешенные изображения)
ILSVRC-2014 –	ImageNet Large Scale Visual Recognition Competition-2014
ML-framework –	Machine Learning-framework
MNIST –	Modified National Institute of Standards and Technology
PIL –	Python Imaging Library
T2 FLAIR –	T2 weighted Fluid-attenuated inversion recovery
VGG –	CNN named after the Visual Geometry Group from Oxford
ВОЗ –	Всемирная Организация Здравоохранения
ИКД –	Измеряемые коэффициенты диффузии
ИНС –	Искусственная нейронная сеть
КТ –	Компьютерная томография
МР-сигнал –	Магнитно-резонансный сигнал
МРТ –	Магнитно-резонансная томография
НС –	Нейронная сеть
СППР –	Система поддержки принятия решений
T1 ВИ –	T1 взвешенные изображения
T2 ВИ –	T2 взвешенные изображения
ЦНС –	Центральная нервная система

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Всемирная Организация Здравоохранения [Электронный ресурс] Режим доступа: <https://www.who.int/ru/news-room/fact-sheets/detail/the-top-10-causes-of-death>
2. СЗГМУ им. И.И. Мечникова [Электронный ресурс] Режим доступа: <http://szgmu.ru/rus/m/457/>
3. Ф. Уоссермен. Нейрокомпьютерная техника: теория и практика. //М. - Мир - 1992.
4. Нейрохирурги с Ордынки [Электронный ресурс]. Режим доступа: <https://www.weekit.ru/themes/detail.php?ID=78195>
5. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – Москва: Горячая линия –Телеком, 2004. – 452 с.
6. Pratt W.K. Digital Image Processing: PIKS Scientific inside (4th ed.) // Wiley-Interscience, John Wiley & Sons, Inc., Los Altos, California, 2007, 782 p.
7. Рашид Т. Создаем нейронную сеть. // СПб.: Альфа-книга, 2017. – 274 с.
8. Обработка изображений [Электронный ресурс]. Электронная энциклопедия Wikipedia. Режим доступа: https://ru.wikipedia.org/wiki/Обработка_изображений
9. Масштабирование изображений [Электронный ресурс]. Инфопедия для углубления знаний. Режим доступа: https://studopedia.ru/3_67751_masshtabirovanie-izobrazheniy.html
10. Фильтр Ланцоша [Электронный ресурс]. Электронная энциклопедия Wikipedia. Режим доступа: https://ru.wikipedia.org/wiki/Фильтр_Ланцоша
11. Частотные и пространственные методы цифровой фильтрации изображений // Черный С.А. – Молодежный научно-технический Вестник ФГБОУ ВПО «МГТУ им. Н.Э. Баумана», 2012. – 6с.
12. Пространственная фильтрация изображений // Чичварин Н. В. – Национальная библиотека им. Н. Э. Баумана, 2016г.

13. ResNet (34, 50, 101): «остаточные» CNN для классификации изображений [Электронный ресурс]. <https://neurohive.io/ru/vidy-nejrosetej/resnet-34-50-101/>
14. Архитектуры нейросетей [Электронный ресурс]. Режим доступа: http://ainews.ru/2018/11/arhitektury_nejrosetej.html
15. VGG16 – сверточная сеть для выделения признаков изображений [Электронный ресурс]. Режим доступа: <https://neurohive.io/ru/vidy-nejrosetej/vgg16-model/>
16. Анализ методов оптимизации параметров нейронных сетей на примере задачи классификации изображений // Суша А.В. – Студенческий: электрон. научн. журн. 2017. №8. Режим доступа: <https://sibac.info/journal/student/8/79708> (дата обращения: 12.05.2019).
17. Методы классификации и прогнозирования. Нейронные сети [Электронный ресурс]. Режим доступа: <https://www.intuit.ru/studies/courses/6/6/lecture/178?page=4>
18. Сверточная нейронная сеть [Электронный ресурс]. Электронная энциклопедия Wikipedia. Режим доступа: https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть
19. Магнитно-резонансная томография с основными и дополнительными программами в диагностике острой фазы ишемического инсульта // Чехонацкий В.А., Чехонацкий И.А.– Журнал «Бюллетень медицинских интернет-конференций», 2017г.
20. Anthony, M. Neural Network Learning: Theoretical Foundations : науч. изд. / М. Anthony, P. L. Bartlett. - Кембридж : Cambridge University Press, 1999. – 404 с.
21. Саймон Хайкин. Нейронные сети: полный курс, 2 издание.// Пер. с англ.. – М.: Издательский дом «Виллямс», 2006. – 1104 с.
22. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. СФУ, 2014. – 60 с.40.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт
Кафедра «Информатика»
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

А.С. Кузнецов

подпись

инициалы, фамилия

« 05 »

07 2019г.

БАКАЛАВРСКАЯ РАБОТА


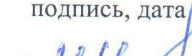
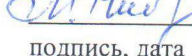

09.03.04 Программная инженерия

код и наименование специальности

Разработка системы поддержки принятия решения

при диагностировании ишемического инсульта

тема

Научный руководитель	 подпись, дата	старший преподаватель должность, ученая степень	А.С. Михалев инициалы, фамилия
Консультант ВКР	 подпись, дата	доцент, канд. техн. наук должность, ученая степень	М.И. Никитина инициалы, фамилия
Выпускник	 подпись, дата		А.В. Тарасевич инициалы, фамилия
Нормоконтролер	 подпись, дата	доцент, канд. техн. наук должность, ученая степень	О.А. Антамошкин инициалы, фамилия

Красноярск 2019