

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

БЕККЕРМАН АННА АЛЕКСАНДРОВНА

Выпускная квалификационная работа

**Распознавание опухолей на МРТ-изображениях головного
мозга**

Уровень образования магистратура

Направление 01.04.02 «Прикладная математика и информатика»

Основная образовательная программа: ВМ.5691 «Прикладная математика и информатика
в задачах медицинской диагностики»

Научный руководитель:

кандидат технических наук,

доцент кафедры КММС

Гришкин Валерий Михайлович

Рецензент:

старший научный сотрудник,

ООО «Нокиа солюшнз энд нетворкс»

Епифанов Николай Анатольевич

Санкт-Петербург

2021 год

Оглавление

Введение.....	3
Постановка задачи.....	5
Обзор литературы.....	6
Искусственные нейронные сети	8
Нейронные сети в медицине	11
Проблемы и ограничения.....	12
Применение сверточных нейронных сетей	14
Оценка точности	15
Набор данных	16
VGG16	17
ResNet	20
Преобразование VGG16	26
Выводы	32
Заключение	33
Список литературы	34

Введение

С самого появления компьютеров люди стараются переложить на них решение как можно большего количества задач. В медицинской диагностике одной из самых главных и сложных задач является правильная постановка диагноза. От неё зависит всё дальнейшее лечение. Конечно, эту задачу также пытаются автоматизировать для минимизации влияния человеческого фактора. Ведь он может быть как положительным (например, большой опыт врача, полученный за долгие годы практики), так и отрицательным (например, плохое самочувствие).

Хотя на данный момент создано немало медицинских систем принятия решения, при их разработке возникают некоторые методологические трудности. Ведь сложно схематизировать данные, когда несколько специалистов может по-разному определять одну и ту же болезнь, что нередко происходит. При помощи четкого алгоритма не всегда можно описать некоторые сложные клинические картины. Самой большой же проблемой является то, что при создании системы нужных знаний может и не быть вовсе. В связи с чем производятся попытки разработки системы, знающей больше ее разработчиков.

В идеале разработанный метод должен обладать стопроцентной чувствительностью (вероятность того, что все люди с положительным результатом будут отнесены к нужному классу) и, одновременно, - стопроцентной специфичностью (вероятность того, что все люди с отрицательным результатом будут определены правильно).

Очень часто высокая чувствительность влечет за собой низкую специфичность. Это может происходить из-за того, что не для каждого человека выход определенного параметра за принятую норму будет считаться заболеванием. Здесь играют роль индивидуальные особенности организма.

Увеличить для метода чувствительность так, чтобы при этом не страдала специфичность, помогают нейронные сети. Это нелинейные системы, которые способны классифицировать данные намного лучше часто применяемых линейных методов. Нейронные сети учатся делать выводы, анализируя обнаруженные ими скрытые связи в данных. При этом они не используют какой-то определенный алгоритм вывода для принятия решения, а учатся этому на примерах. Также нейронные сети могут проводить классификацию, подытоживая предыдущий опыт и используя его в последующих задачах. [1]

Постановка задачи

Целью настоящей работы является проведение, методами машинного обучения, распознавания опухолей головного мозга на МРТ-снимках. На основе работы разработанной нейронной сети производится классификация снимков по наличию на них опухолей.

Для реализации заявленной цели требуется осуществить следующие задачи:

- Провести анализ данных МРТ-снимков и произвести их предварительную обработку;
- Провести анализ существующих нейронных сетей, применимых для достижения поставленной цели;
- Разработать многослойную сверточную нейронную сеть, распознающую опухоль головного мозга;
- Продемонстрировать полученные результаты разработанной нейронной сети.

Обзор литературы

Книга [2], написанная Сэмюэлем Бернсом, представляет собой введение в методы глубокого обучения. Автор не уделяет слишком много внимания математике, так как это руководство предназначено для разработчиков, которые являются новичками в области глубокого обучения. Книга была сгруппирована по главам, в каждой из которых рассматриваются различные функции библиотек глубокого обучения (Tensorflow, Keras и PyTorch), которые могут быть использованы в языке программирования Python.

В [3] описываются экспериментальные результаты на основе двух основных параметров, таких как время и точность. Это исследование дает сравнение производительности различных алгоритмов, таких как DNN, ANN (Artificial Neural Networks) и KNN. Экспериментальные результаты и статистический анализ показывают процент точности: 93,18, 90,90 и 81,81 соответственно. По результатам ясно, что DNN показал более высокий уровень по сравнению с другими оставшимися методологиями KNN и ANN. В качестве набора данных в этом исследовании используются изображения МРТ.

В этом исследовании [4] представляется метод сегментации, который полезен для облегчения пользователям быстрого и эффективного распознавания опухолей с помощью МРТ головного мозга. Эта методология основана на анализе симметрии и применяется к многочисленным наборам данных по опухолям разного размера, локализации и интенсивности для автоматического обнаружения и сегментации различных категорий опухоли головного мозга. Эта методика позволяет врачам находить опухоль в головном мозге пациента и вычислять её площадь, чтобы можно было запланировать эффективную терапию и лечение. Обработка изображений осуществляется при помощи MATLAB.

Автор [5] применил методику SVM-классификации на МРТ-изображениях головного мозга для классификации на нормальный и

патологический. Для извлечения особенностей использовался Matlab 7.9. Извлеченные результаты используются в качестве входных данных для процесса классификации, чтобы сделать вывод. С точностью 65% нормальные изображения классифицируются как успешные, а патологические – нет. Согласно этому исследованию SVM не может дать достоверных результатов с большими данными.

В другом исследовании [6] МРТ-изображения сегментированы с использованием техники пороговой сегментации. Перед процессом сегментации изображения преобразуются в изображения с оттенками серого, а затем фильтруются для удаления шума и повышения яркости или резкости изображений для получения лучшего результата. Метод классификации SVM используется в качестве классификатора, который показывает, является ли опухоль злокачественной, доброкачественной или нормальной.

В [7] и [8] использовался метод нейронной сети обратного распространения для классификации. Вейвлет-преобразование используется для извлечения признаков, а РСА (метод главных компонент) используется для выбора признаков. Сокращенные данные используются для получения лучших результатов.

Искусственные нейронные сети

Искусственные нейронные сети (ИНС) представляют собой сложные структуры, содержащие взаимосвязанные адаптивные элементы, известные как искусственные нейроны, которые могут выполнять большие вычисления для представления знаний. Они обладают всеми фундаментальными качествами биологической нейронной системы, включая способность к обучению, устойчивость, нелинейность, высокий параллелизм, отказоустойчивость, способность обрабатывать неточную и нечеткую информацию и способность к обобщению.

Основные характеристики:

- нелинейность (придает лучшее соответствие данным);
- высокий параллелизм (способствует быстрой обработке и отказоустойчивости оборудования);
- обобщение (позволяет применять модель к необученным данным);
- нечувствительность к шуму (позволяет хорошо предсказывать даже неопределенные данные и ошибки измерений);
- обучение и адаптивность (позволяют модели обновлять свою внутреннюю архитектуру в соответствии с меняющейся средой). [9]

Нейронные сети особенно хорошо подходят для выполнения распознавания образов для идентификации и классификации объектов или сигналов в речи, зрении и системах управления. Они также могут быть использованы для выполнения прогнозирования и моделирования временных рядов.

Нейронная сеть объединяет несколько уровней обработки, используя простые элементы, работающие параллельно и вдохновленные биологическими нервными системами. Он состоит из входного слоя, одного или нескольких скрытых слоев и выходного слоя. В каждом слое есть несколько узлов, или нейронов, причем каждый слой использует выход

предыдущего слоя в качестве своего входа, поэтому нейроны соединяют различные слои. Входной слой подается с входными, то есть необработанными данными. Он передает информацию из внешнего мира в сеть. В этом слое вычисления не выполняются – узлы просто передают информацию скрытому слою. Все вычисления по объектам, введенным через входной слой, происходят в скрытом слое/слоях, а затем он передает результат на выходной слой. Выходной же слой отображает результаты вычислений, выполненных сетью.

Каждый нейрон сети имеет свой вес, который корректируется в процессе обучения, и по мере того, как вес уменьшается или увеличивается, он изменяет силу сигнала этого нейрона. Эти веса автоматически корректируются в соответствии с заданным правилом обучения до тех пор, пока искусственная нейронная сеть не выполнит требуемую задачу правильно.

ИНС используют функции активации для выполнения сложных вычислений в скрытых слоях, а затем передают результат на выходной слой. Основная цель этих функций – введение нелинейных свойств в нейронную сеть.

Они преобразуют линейные входные сигналы узла в нелинейные выходные сигналы, чтобы облегчить изучение полиномов высокого порядка, выходящих за пределы одной степени для глубоких сетей. Уникальным аспектом функций активации является то, что они дифференцируемы. Это помогает им функционировать во время обратного распространения нейронных сетей.

Если функции активации не применяются, выходной сигнал будет линейной функцией, которая является полиномом одной степени. Хотя линейные уравнения легко решать, они имеют ограниченный коэффициент сложности и, следовательно, имеют меньше возможностей для изучения сложных функциональных отображений из данных. Таким образом, без

функции активации нейронная сеть была бы моделью линейной регрессии с ограниченными возможностями.

К самым популярным функциям активации можно отнести следующие:

➤ Сигмоидная $\sigma(x) = \frac{1}{1+e^{-x}}$

Сигмоидная функция берет вещественное число и преобразует его в число в диапазоне от 0 до 1. В частности, большие отрицательные числа становятся 0, а большие положительные – 1.

➤ Tanh

Функция $\tanh x = \frac{2}{1+e^{-2x}} - 1$ сжимает вещественное число в диапазон $[-1, 1]$. Ее выход имеет нулевой центр. Можно заметить, что это скорректированная сигмоидная функция $\tanh x = 2\sigma(2x) - 1$.

➤ Softmax

Эта функция генерирует выход, который колеблется между значениями 0 и 1 и сумма вероятностей равна 1.

➤ ReLU (Rectified linear unit)

Линейный выпрямитель стал очень популярным в последние несколько лет. Он вычисляет функцию $f(x) = \max(0, x)$. Другими словами, активация просто имеет нулевой порог.

➤ ELUs (Exponential Linear Units) $f(x) = \begin{cases} \alpha(e^x - 1), & x \leq 0 \\ x, & x > 0 \end{cases}$

Здесь отрицательная составляющая моделируется с помощью экспоненциального представления. Однако у нас все равно будет обучаемый параметр α .

Нейронные сети, состоящие из двух или трех связанных нейронных слоев, известны как неглубокие нейронные сети. Сети глубокого обучения могут иметь много уровней, даже сотни. Оба метода являются методами машинного обучения, которые учатся непосредственно из входных данных.

Глубокое обучение особенно хорошо подходит для сложных приложений идентификации, таких как распознавание лиц, перевод текста и распознавание голоса. [10][11]

Нейронные сети в медицине

В последнее время глубокое обучение всё чаще применяется для решения различных медицинских задач. Этому способствует увеличение вычислительной мощности и значительное количество маркированных снимков, а также возможность использования облачных хранилищ.

Нейронные сети используются в медицине на следующих уровнях:

- быстрое и четкое распознавание изображений;
- снижение вероятности врачебных ошибок;
- контроль и анализ своего состояния самими пациентами с помощью датчиков.

Из-за существующих ограничений применение нейронных сетей в медицине довольно ограничено и развивается несколько медленнее, чем могло бы.

На данный момент они помогают врачам в постановке диагнозов, удаляют различные помехи с биологических сигналов, из множества имеющихся данных определяют наиболее важные по некоторым условиям. И это лишь малая часть возможностей, открывающихся благодаря использованию искусственных нейронных сетей.

Глубокие нейронные сети решают такие задачи как интерпретация различных патологий на медицинских сканах и расшифровка электрокардиограмм.

С их помощью также возможно определить некоторые виды рака, кровоизлияния, кожные заболевания, различные переломы и еще многие другие болезни. [12]

Проблемы и ограничения

Главной причиной, тормозящей развитие искусственного интеллекта в области медицинской диагностики, является вопрос защиты и конфиденциальности данных. Данные пациента могут быть получены из его истории болезни. Также нельзя исключать вероятность преднамеренного взлома системы с целью причинения вреда человеку.

Также проблемы могут возникнуть из-за неточностей в работе алгоритма. К примеру, алгоритм «Watson for Oncology» для лечения больных раком был основан на маленьком количестве как синтетических случаев, так и реальных данных. Он предлагал неправильные и даже опасные рекомендации к лечению вроде использования несовместимого лекарства, хотя это определено и было противопоказано.

Следующей проблемой является предвзятость и непропорциональное использование возможностей ИИ. Полученные результаты могут быть не совсем объективны из-за отсутствия в наборе данных исследований различных слоев общества. При этом преждевременная смертность часто связана именно с низким социально-экономическим статусом. [12]

Для разных поставленных задач и используемых при этом обучающих данных в задачах медицинской диагностики применяются разные архитектуры нейронных сетей. Для их написания используется языки различной сложности – от простых вроде BASIC до сложных языков ООП, как например Python.

На данный момент известно достаточно большое количество моделей применяемых архитектур искусственных нейронных сетей, имеющих различные вычислительные сложности и меру схожести с нейронами человеческого мозга. Данные модели уникальны, что не дает применить к ним стандарты существующих классификаций, в отличии от традиционных статистических методов.

Одна из главных проблем использования искусственных нейронных сетей – невозможность заранее определить как архитектуру проектируемой сети, так и ее сложность, необходимые для требуемого уровня точности результатов. Из-за слишком высоких требований может возникнуть необходимость в еще большем усложнении архитектуры. Нейронные сети, включающие только один скрытый слой, могут справиться только с самыми простыми задачами. Увеличение же количества скрытых слоев, необходимых для решения более сложных задач, требует также и увеличения вычислительных мощностей. Решению данной проблемы через возрастание мощностей процессоров компьютера и улучшение систем кластеризации посвящено большое количество исследований. [13]

Применение сверточных нейронных сетей

В задачах по обработке изображений наиболее часто используются сверточные нейронные сети (CNN). Основными задачами по распознаванию изображений считают:

- классификацию (определение объекта к классу);
- обнаружение (выделение объекта рамкой);
- сегментацию (попиксельная классификация объекта).

Сверточная нейронная сеть состоит из слоев:

- сверточных;
- объединения или подвыборки;
- полносвязных слоев.

Сверточные слои – это основной блок нейронной сети. Именно в этих слоях производится большинство вычислений. Свертка производится с помощью детектора признаков, который также называют ядром или фильтром. Он перемещается по всему изображению, проверяя присутствие признаков в каждой его области (рис. 1). Вычисленные в каждой области точечные произведения между входными пикселями и фильтром подаются в выходной массив. Конечный вывод из серии точечных продуктов из входного сигнала и фильтра известен как карта объектов или карта активации.

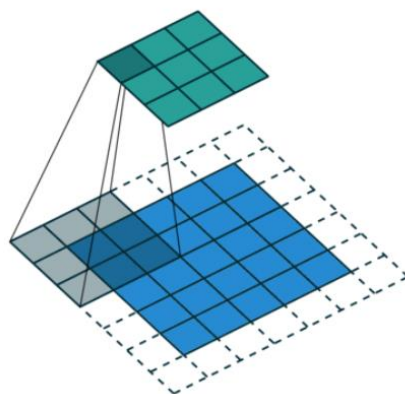


Рисунок 1. Свертка изображения

Затем к нейрону применяется нелинейная функция активации. После сверточного слоя идет слой подвыборки (рис. 2), уменьшающий размерность, за счет чего уменьшается также количество параметров во входных данных. Остаются лишь наиболее информативные области.

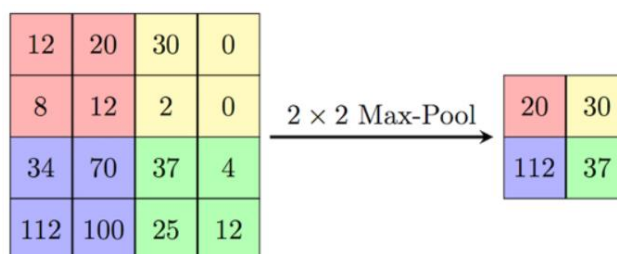


Рисунок 2. Подвыборка изображения

Затем следует часть, отвечающая за регуляризацию и часть, устанавливающая класс изображения по найденным признакам. [14]

Оценка точности

Для оценки точности предсказанных значений будем использовать метрику F-мера.

Для начала необходимо ввести понятия True Positive (TP), True Negative (TN), False Positive (FP) и False Negative (FN).

Пусть y — реальное значение объекта, а \hat{y} — предсказанное.

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Далее введем понятия precision и recall.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Precision определяет, какая часть объектов, что были отнесены классификатором к положительным, действительно является положительными, а recall — какая часть действительно положительных объектов была отнесена классификатором к положительным.

В бинарной классификации recall для положительного класса (в нашей задаче это изображения с опухолью) также известен как «чувствительность», а отзыв отрицательного класса (изображения без опухоли) – «специфичность».

С помощью данных метрик мы не даем классификатору отнести все объекты к одному и тому же классу, т.е. сказать, что на всех снимках головного мозга нет опухолей или же наоборот, что они есть на всех снимках.

Обычно оптимизацию параметров производят с использованием только одной метрики, изменение значения которой и отслеживают на тестовой выборке. Совместить precision и recall в один критерий можно разными способами. Воспользуемся их средним гармоническим - F-мерой.

$$F = 2 * \frac{precision * recall}{precision + recall}$$

F-мера имеет максимальное значение при равных единице precision и recall, и стремится к нулю, если один из аргументов стремится к нулю. [15]

Набор данных

В данной работе используется набор данных из 4253 снимков МРТ, собранных из нескольких открытых источников. Данные содержат два класса изображений: с опухолью и без (рис. 3).

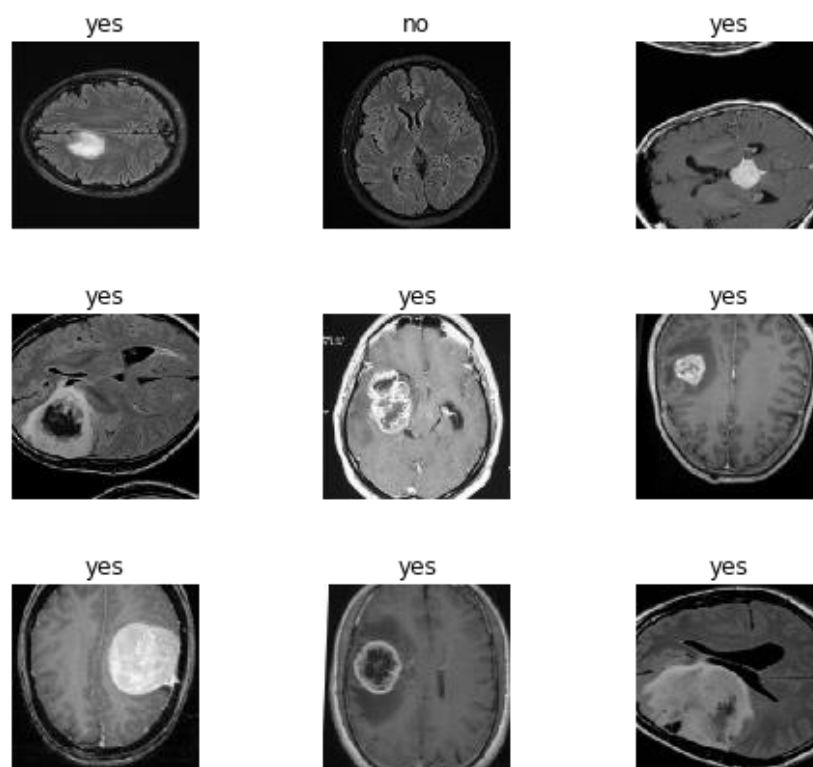


Рисунок 3. Обучающие данные

Весь набор данных разделен на две части: обучающий и тестовый. Каждый набор состоит как из изображений с опухолью, так и без опухоли. Все 100% данных делятся следующим образом: 80% от общего набора данных выделяется для обучения модели, а оставшиеся 20% - для тестирования.

Рассмотрим подробнее работу нескольких нейронных сетей (моделирование производится на языке Python с использованием библиотеки TensorFlow).

VGG16

Самая трудная часть работы с CNN – это их создание. Они отлично смотрятся, когда уже готовы, а также когда они реализуются, но попытка реализовать CNN, это всегда лотерея. Есть так много вещей, с которыми нужно справиться, когда мы определяемся с количеством CNN-подобных слоев, размером фильтра, типом заполнения и многим другим. Решение всего этого состоит в том, чтобы использовать предварительно обученную модель для нашей классификации изображений.

Одной из самых простых для понимания и построения моделей является VGG16. Это также одна из наиболее часто используемых моделей в наши дни.

Эта модель была предложена группой визуальной геометрии из Оксфорда и также называется моделью OxfordNet в их честь.

VGG16 (рис. 4) это улучшенная модель AlexNet, где вместо больших фильтров (11x11 для первого слоя свертки и 5x5 для второго) используется несколько фильтров размера 3x3, которые следуют друг за другом. Главный принцип VGG-архитектур – увеличение числа слоев с применением фильтров небольшого размера. Разработчики доказали, что слой с фильтром 7x7 равнозначен трем слоям 3x3, также как слой 5x5 – двум слоям 3x3, что дает сэкономить 55% и 22% параметров соответственно. Фильтры маленького размера получают меньше параметров, однако это позволяет повысить эффективность их обработки.

Основные характеристики VGG16:

- число 16 означает, что модель имеет в общей сложности 16 слоев с некоторым весом;
- в ней есть только сверточные и объединяющие слои;
- всегда используется ядро 3x3 для свертки;
- размер максимального бассейна 2x2;
- имеет в общей сложности около 138 миллионов параметров.

Входным сигналом модели является изображение 224x224x3 пикселей. Далее идут два слоя свертки каждым размером 224x224x64, после чего применяется пулинговый или же объединяющий слой, который уменьшает высоту и ширину изображения до 112x112x64.

Затем есть два слоя свертки conv128 размера 112x112x128, после чего идет объединяющий слой, который снова уменьшает высоту и ширину изображения до 56x56x128.

Затем еще три слоя свертки conv256 размера $56 \times 56 \times 256$ и объединяющий слой, уменьшающий размер изображения до $28 \times 28 \times 256$.

Далее три слоя conv512 $28 \times 28 \times 512$, после чего объединяющий слой уменьшает размер изображения до $14 \times 14 \times 512$.

Затем снова идут три слоя conv512 размера $14 \times 14 \times 512$, после этого объединяющий слой с $7 \times 7 \times 512$ и два полносвязных слоя каждый из 4090 узлов. И, наконец, мы имеем выходной слой с 1000 узлами (по 1 для каждого класса). Последним идет softmax слой для нормализации вектора классификации. В качестве функции активации все скрытые слои используют ReLU. [16]

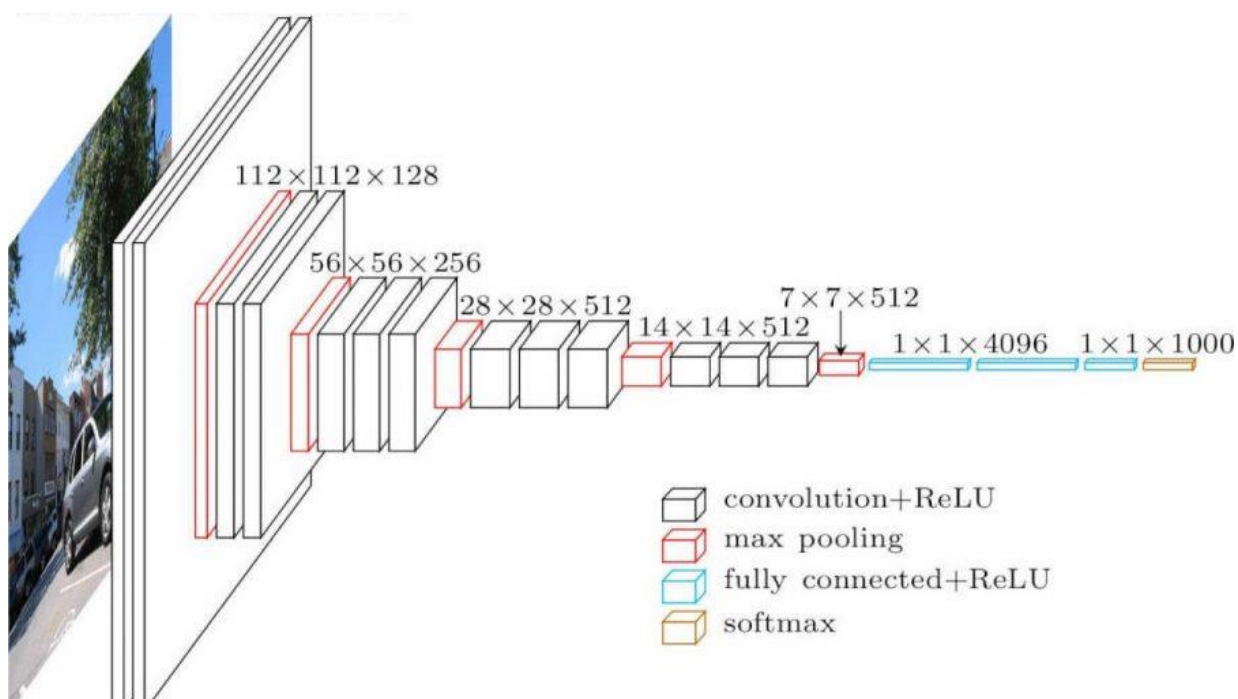


Рисунок 4. Архитектура VGG16

Рассмотрим работу данной сети для задачи обнаружения опухоли на снимках головного мозга.

Вычислим precision и recall для нашей модели, а также значение F-меры.

В sklearn есть удобная функция sklearn.metrics.classification_report, возвращающая recall, precision и F-меру для каждого из классов. Результаты приведены на рисунке 5.

	precision	recall	f1-score
0	0.83	0.87	0.85
1	0.87	0.83	0.85

Рис. 5. Результаты обучения VGG16

ResNet

В последние годы нейронные сети стали более глубокими, с современными сетями, простирающимися от нескольких слоев (например, AlexNet) до более чем ста слоев.

Одним из главных преимуществ очень глубокой сети является то, что она может представлять очень сложные функции.

Однако огромным препятствием для их обучения является исчезновение градиентов: очень глубокие сети часто имеют сигнал градиента, который быстро стремится к нулю, что делает градиентный спуск непомерно медленным.

Более конкретно, во время градиентного спуска, когда мы возвращаемся от конечного слоя обратно к первому слою, мы умножаем на матрицу веса на каждом шаге. Если градиенты малы из-за большого числа умножений, градиент может быстро уменьшаться экспоненциально до нуля (или, в редких случаях, быстро расти экспоненциально и «взрываться», принимая очень большие значения).

Обычные сети, такие как VGG16, называются «простыми» сетями.

В простых сетях, когда количество слоев увеличивается с 20 до 56 (рис.6), даже после тысяч итераций ошибка обучения была хуже для 56-го слоя по сравнению с 20-слойной сетью.

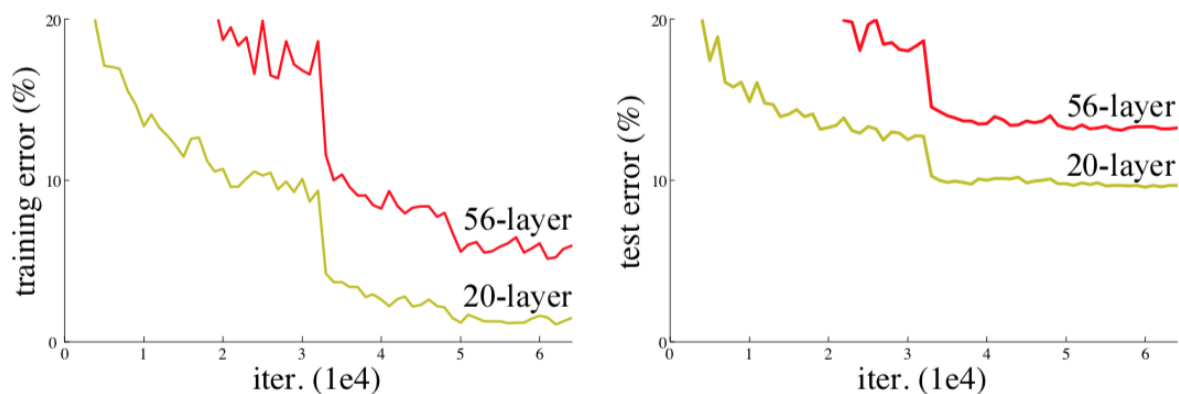


Рисунок 6. Сравнение ошибок для 20-слойной и 56-слойной сетей

Теоретически мы ожидаем, что наличие более глубокой сети должно только помочь, но на самом деле более глубокая сеть имеет более высокую ошибку обучения и, следовательно, ошибку тестирования.

Когда более глубокие сети могут начать сходиться, обнаруживается проблема деградации: с увеличением глубины сети точность насыщается (что неудивительно), а затем быстро ухудшается.

Использование более глубоких сетей снижает производительность модели. Исследовательская работа Microsoft предлагает в качестве решения проблемы остаточный блок/блок идентификации.

Идея заключается в том, что вместо того, чтобы позволять слоям обучаться на базовом отображении, позволить сети обучиться на остаточном отображении. Итак, вместо того, чтобы говорить $H(x)$ (начальное отображение), пусть сеть соответствует $F(x) := H(x) - x$, что дает $H(x) := F(x) + x$.

Подход состоит в том, чтобы добавить ярлык или пропустить соединение, что позволит информации течь легче от одного слоя к следующему, т. е. вы обходите данные вместе с обычным потоком CNN от одного слоя к следующему слою после непосредственного следующего.

Остаточный Блок:

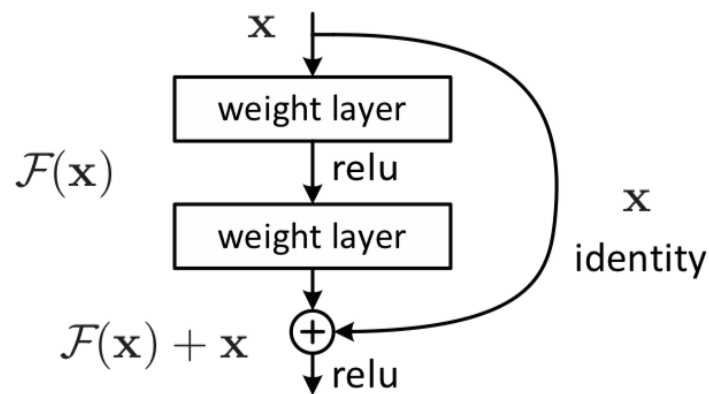


Рисунок 7. Остаточный блок

Два свойства остаточного блока:

1. Добавление дополнительных/новых слоев не повредит производительности модели, так как регуляризация пропустит их, если эти слои не будут полезны.
2. Если дополнительные/новые слои были полезны, даже при наличии регуляризации, веса или ядра слоев будут ненулевыми, и производительность модели может немного увеличиться.

Поэтому, добавляя новые слои из-за «пропущенного» или «остаточного соединения», гарантируется, что производительность модели не снизится, но может немного увеличиться.

Укладывая эти блоки ResNet друг на друга, можно сформировать очень глубокую сеть. Наличие блоков ResNet с ярлыком также очень упрощает для одного из блоков обучение функции идентификации. Это означает, что вы можете накапливать дополнительные блоки ResNet с небольшим риском снижения производительности обучающего набора.

В сети ResNet используются два основных типа блоков, в основном в зависимости от того, одинаковы или различны размеры ввода/вывода.

1. Блок идентификации — такой же, как тот, который мы видели выше. Блок идентификации является стандартным блоком, используемым в сетях

ResNet, и соответствует случаю, когда входная активация имеет тот же размер, что и выходная активация.

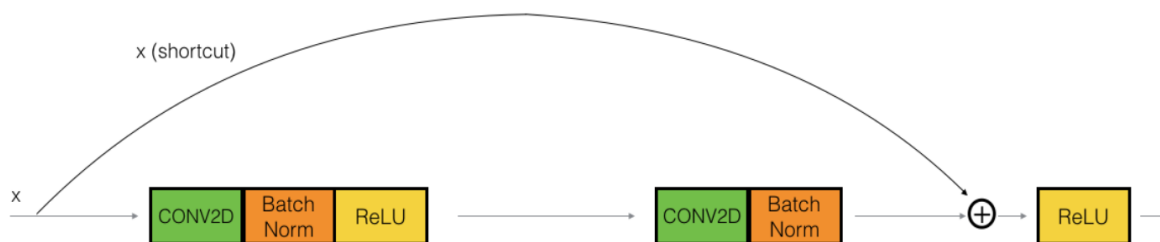


Рисунок 8. Блок идентификации

2. Сверточный блок — мы можем использовать этот тип блока, когда входные и выходные размеры не совпадают. Разница с блоком идентификации заключается в том, что в пути быстрого доступа есть Conv2DLayer.

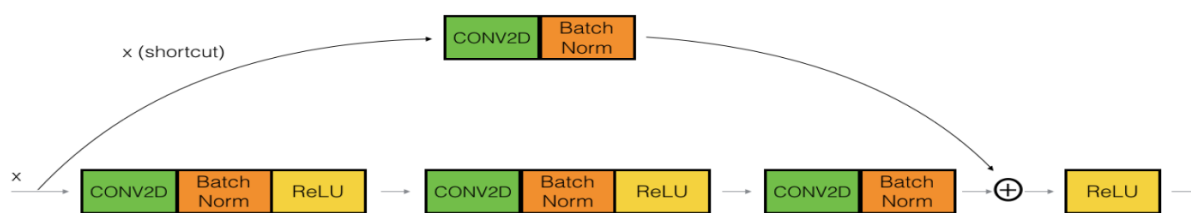


Рисунок 9. Сверточный блок

Теперь у нас есть необходимые блоки для создания очень глубокой сети. На следующем рисунке подробно описана архитектура этой нейронной сети на примере ResNet50. «ID BLOCK» здесь означает «блок идентификаторов», а «ID BLOCK x3» означает, что необходимо сложить 3 блока идентификаторов вместе.

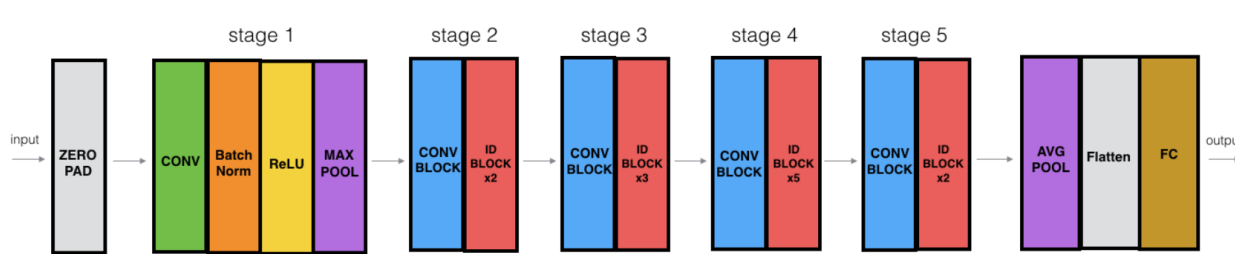


Рисунок 10. Архитектура ResNet

Слой Zero-padding заполняет вход прокладкой (3,3).

2D-свертка (CONV) имеет 64 фильтра размера (7,7) и использует шаг (2,2). BatchNorm применяется к оси каналов входного сигнала. Максимальное объединение (MAX POLL) использует окно (3,3) и шаг (2,2).

Сверточный блок второго этапа, как и два блока идентификации, использует три набора фильтров размером 64x64x256.

ResNet включает в себя большой стек одинаковых остаточных блоков. В каждом из этих блоков по два слоя свертки 3x3. Фильтры периодически удваиваются, размерность же их при этом уменьшается с шагом 2.

На пятом этапе используются фильтры уже с размерами 512x512x2048 для сверточного блока и 256x256x2048 для блока идентификации.

В 2D-среднем пулинге (AVG POOL) используется окно (2,2).

Слой Dense уменьшает свой вход до количества классов с помощью активации softmax.

Если количество слоев возрастает, используются дополнительные слои для снижения размерности у изображения. [17]

Работа данной сети с 34 и 152 слоями была проверена на том же наборе данных, что и VGG16. Полученные значения precision, recall и F-меры представлены на рисунке 11.1 и 11.2.

	precision	recall	f1-score
0	0.82	0.83	0.83
1	0.84	0.83	0.83

Рисунок 11.1. ResNet34

	precision	recall	f1-score
0	0.91	0.65	0.76
1	0.74	0.94	0.83

Рисунок 11.2. ResNet152

Сравним средние значения результатов работы у трех рассмотренных сетей.

	VGG16	ResNet34	ResNet152
Precision	0.85	0.83	0.82
Recall	0.85	0.83	0.80
F-мера	0.85	0.83	0.80

В таблице видно, что полученные значения отличаются не сильно, однако VGG16 показывает более высокие результаты. Попробуем улучшить её работу.

Преобразование VGG16

Результаты нейронной сети в значительной степени зависят от выбора параметров. Какого-то одного общего алгоритма для их определения не существует, поэтому всё осуществляется через метод проб и ошибок. Для того, чтобы определить наиболее оптимальные значения, приходится реализовать несколько разнообразных вариантов с различными значениями.

Загрузим VGG16 без слоев классификации и добавим к ней свои слои:

➤ Flatten

Данный слой конвертирует входные данные в меньшую размерность (к примеру, $(batch_size, 3, 2) \rightarrow (batch_size, 6)$).

➤ Dropout

Используется для избежания проблемы переобучения путем присваивания случайно выбранным долям единиц значения нуля при каждом обновлении.

➤ Dense

Принимает данные со всех узлов предыдущего слоя. [18]

➤ BatchNormalization

Данный слой применяется для ускорения глубокого обучения и помогает решить следующую проблему: по мере распространения сигнала по сети, даже если мы нормализовали его на входе, пройдя через внутренние слои, он может сильно исказиться как по математическому ожиданию, так и по дисперсии, что чревато серьезными несоответствиями между градиентами на различных уровнях. Поэтому возникает необходимость применять более сильные регуляризаторы, замедляя тем самым темп обучения.

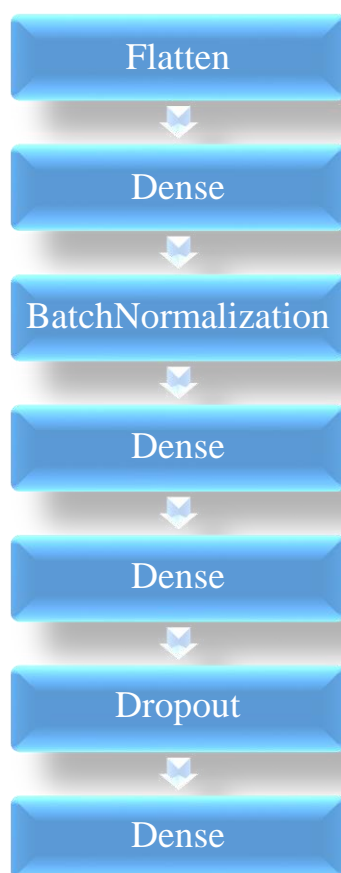
BatchNormalization предлагает весьма простое решение данной проблемы: нормализовать входные данные таким образом, чтобы получить нулевое математическое ожидание и единичную дисперсию. [19]

Обычно слой Dense добавляется последним после всех сверток для классификации изображения. Однако входные данные слоя Dense имеют размерность 2D, а выход Convolution слоя представляет собой массив 4D. Поэтому нужно изменить размерность выходного сигнала, полученного от слоя свертки, на двумерный массив.

Осуществим это вставив слой Flatten после слоя Convolution. Выровниваем слой, преобразовав три измерения изображения до одного измерения. Теперь у нас есть 2D массив, который является приемлемым для Dense слоев. [20]

Рассмотрим результаты работы четырех вариантов модели нейронной сети.

Для первой модели выберем следующую структуру слоев классификации:

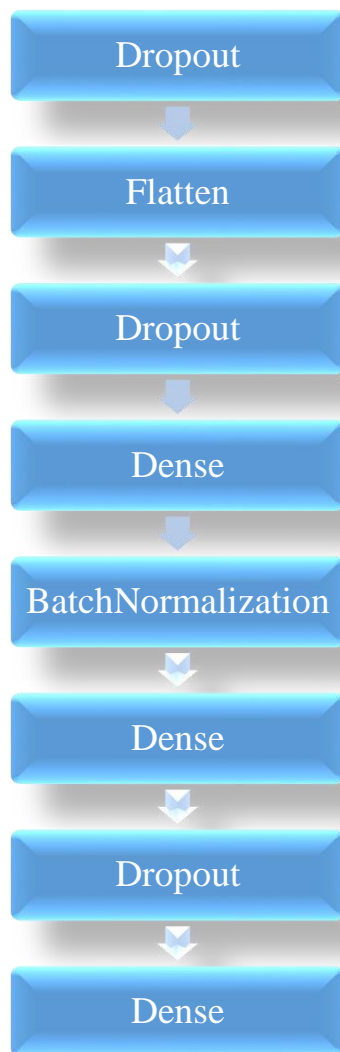


В результате обучения получаем следующий результат (рис. 12)

	precision	recall	f1-score
0	0.94	0.69	0.79
1	0.76	0.96	0.85

Рисунок 12. Результат обучения модели №1

У второй модели для слоев классификации зададим структуру:

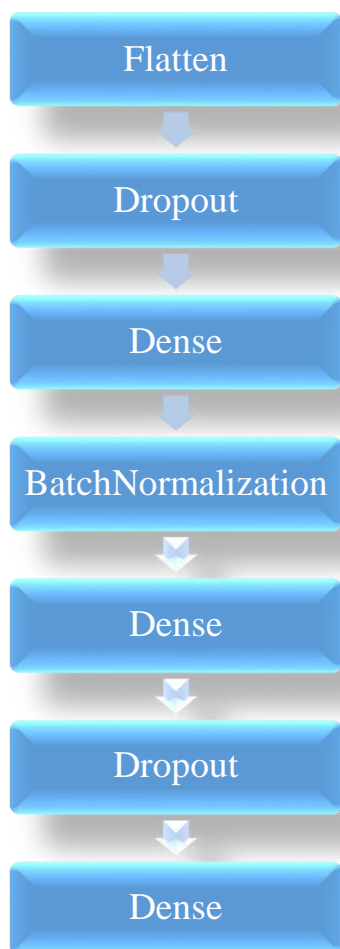


Результат обучения этой модели представлен на рисунке 13.

	precision	recall	f1-score
0	0.97	0.75	0.85
1	0.80	0.98	0.88

Рисунок 13. Результат обучения модели №2

Слой классификации для третьей модели:

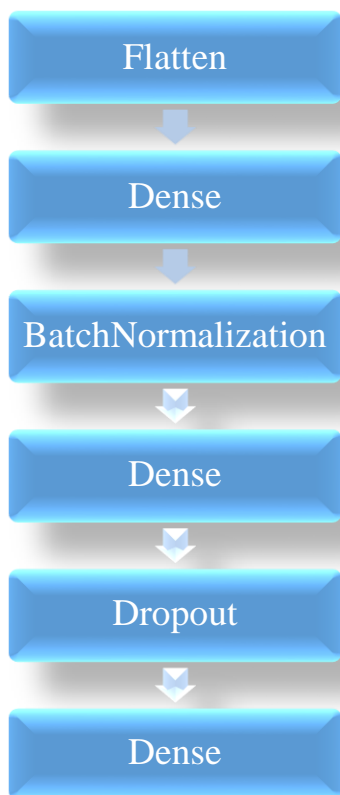


Результат обучения этой модели представлен на рисунке 14.

	precision	recall	f1-score
0	0.99	0.77	0.87
1	0.82	0.99	0.90

Рисунок 14. Результат обучения модели №3

И для четвертой модели слои классификации имеют следующую структуру:



Результат обучения видим на рисунке 15.

	precision	recall	f1-score
0	0.93	0.88	0.90
1	0.89	0.93	0.91

Рисунок 15. Результат обучения модели №4

Сравним средние значения для наших моделей.

	№1	№2	№3	№4
Precision	0.85	0.89	0.91	0.91
Recall	0.82	0.87	0.88	0.91
F-меры	0.82	0.87	0.89	0.91

Как видно, наилучшие результаты показывает четвертая модель. Хотя значения precision у последних двух моделей совпадают, остальные

показатели выше у четвертой модели. Основным критерием для принятия решения является F-мера, полученная наибольшей для четвертой модели. Данную модель и выберем, как наиболее успешную.

Выводы

Целью данной работы являлось проведение, методами машинного обучения, распознавания опухолей головного мозга на МРТ-снимках.

Были выполнены следующие задачи:

- в результате сравнения трех нейронных сетей (VGG16, ResNet34, ResNet152) для улучшения была выбрана VGG16;
- разработаны несколько модификаций базовой сети, вместо стандартных слоев классификации добавлены новые;
- результаты работы новых моделей нейронных сетей были оценены при помощи F-меры, совмещающей в один критерий precision и recall;
- выбрана наиболее успешная модель.

Заключение

Подводя итог, можно сказать, что все поставленные задачи были выполнены. Была проведена классификация снимков по наличию на них опухолей на основе работы четырех разработанных моделей нейронной сети и выбрана наилучшая, показавшая значение F-меры равное 0.91.

В дальнейшем применение нейронных сетей показывает свою целесообразность в использовании для программного обеспечения, которое будет быстро и точно обрабатывать огромные массивы данных, и машин, способных видеть и делать то, с чем не справляется человек. Проводимые исследования с применением нейронной сети говорят о перспективности данного направления и еще многих ее неизученных возможностях.

Список литературы

1. Чурюмова И. Г. Медицинская система принятия решений с использованием нейронной сети // Вестник Нац. техн. ун-та "ХПИ" : сб. науч. тр. Темат. вып. : Информатика и моделирование. – Харьков : НТУ "ХПИ". – 2004. – № 34. – С. 199-202.
2. Samuel Burns. Python Deep learning // Kindle Edition, 2019, С. 178
3. Ramaswamy Reddy A., Prasad E. V., Reddy L. S. S. Comparative analysis of brain tumor detection using different segmentation techniques //International Journal of Computer Applications. – 2013. – Т. 82. – №. 14. – С. 0975-8887.
4. Das S., Siddiqui N. N., Kriti N., & Tamang, S. P. Detection and area calculation of brain tumour from MRI images using MATLAB //International Journal. – 2017. – Т. 4. – №. 1. – С. 35
5. Othman M. F. B., Abdullah N. B., Kamal N. F. B. MRI brain classification using support vector machine //2011 Fourth International Conference on Modeling, Simulation and Applied Optimization. – IEEE, 2011. – С. 1-4.
6. Kumar D. D., Vandhana S., Priya, K. S., & Subashini S. J. Brain tumour image segmentation using MATLAB //vol. – 2015. – Т. 1. – С. 447-451.
7. Zhang Y., Dong Z., Wu L., & Wang S. A hybrid method for MRI brain image classification //Expert Systems with Applications. – 2011. – Т. 38. – №. 8. – С. 10049-10053.
8. Othman M. F., Basri M. A. M. Probabilistic neural network for brain tumor classification //2011 Second International Conference on Intelligent Systems, Modelling and Simulation. – IEEE, 2011. – С. 136-138.
9. 6 Types of Activation Function in Neural Networks You Need to Know [Электронный ресурс], // URL: <https://www.upgrad.com/blog/types-of-activation-function-in-neural-networks/> (дата обращения: 13.02.2020)
10. Neural Network | Machine Learning Tutorial [Электронный ресурс], // URL: https://sci2lab.github.io/ml_tutorial/neural_network/#Common-Activation-Functions (дата обращения: 13.01.2020)

11. What Is a Neural Network? [Электронный ресурс], 1994-2021 // The MathWorks, Inc, URL: <https://www.mathworks.com/discovery/neural-network.html> (дата обращения: 05.03.2020)
12. Neurohive. Рутинные задачи с минимальным риском [Электронный ресурс], // URL: <https://neurohive.io/ru/novosti/nejronnye-seti-v-medicine/> (20.05.2020)
13. Волчек Ю.А., Шишко О.Н., Спиридонова О.С., Мохорт Т.В.. Положение модели искусственной нейронной сети в медицинских экспертных системах// Juvenis scientia. – 2017. – №. 9.
14. Фролов В.В., Шляпкин А.В., Яценко Н.В., Алгоритмы анализа и классификации патологий на рентгенограммах грудной клетки, с использованием нейронных сетей // ТГУ, 2018, с.35-36
15. Метрики в задачах машинного обучения [Электронный ресурс], // URL: <https://habr.com/ru/company/ods/blog/328372/> (дата обращения: 15.09.2020)
16. Vgg16 Architecture, Implementation and Practical Use [Электронный ресурс], // URL: <https://medium.com/pythoneers/vgg-16-architecture-implementation-and-practical-use-e0fef1d14557> (дата обращения: 05.11.2020)
17. Understanding and implementation of Residual Networks(ResNets) [Электронный ресурс], // URL: <https://medium.com/analytics-vidhya/understanding-and-implementation-of-residual-networks-resnets-b80f9a507b9c> (дата обращения: 17.11.2020)
18. Как работают слои в Keras, их типы и свойства ~ PythonRu [Электронный ресурс], // URL: <https://pythonru.com/biblioteki/sloi-keras-parametry-i-svoystva-keras-5> (дата обращения: 17.02.2021)
19. Глубокое обучение для новичков: тонкая настройка нейронной сети [Электронный ресурс], // URL: <https://habr.com/en/company/wunderfund/blog/315476/> (дата обращения: 19.02.2021)

20. Понимание размерности ввода-вывода в свёрточной нейронной сети
[Электронный ресурс], // URL: [https://id-lab.ru/posts/developers/
ponimanie-razmernosti-vvoda-vyvoda-v-svyortochnoj-nejronnoj-seti-keras/](https://id-lab.ru/posts/developers/ponimanie-razmernosti-vvoda-vyvoda-v-svyortochnoj-nejronnoj-seti-keras/)
(дата обращения: 10.04.2021)