

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра Прикладная математика и информатика

(наименование)

02.03.03 Математическое обеспечение и администрирование
информационных систем

(код и наименование направления подготовки, специальности)

Технология программирования

(направленность (профиль)/ специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Реализация сверточной нейронной сети для дифференциации
затемнений на снимках рентгенограмм»

Студент

Е.В. Прокопов

(И.О. Фамилия)

(личная подпись)

Руководитель

М.Г. Лисовская

(ученая степень, звание, И.О. Фамилия)

Консультант

О.А. Головач

(ученая степень, звание, И.О. Фамилия)

АННОТАЦИЯ

Темой данной выпускной квалификационной работы является «Реализация сверточной нейронной сети для дифференциации затемнений на снимках рентгенограмм».

Работа выполнена студентом Тольяттинского Государственного Университета, института математики, физики и информационных технологий, группы МОп – 1601б, Прокоповым Егором Витальевичем.

Целью ВКР повышение степени автоматизации в анализе рентгенограммы с помощью сверточной нейронной сети.

Объектом ВКР процесс классификации патологий на рентгенограмме.

Предмет исследования – нейронные сети в задаче классификации патологий на рентгенограмме.

В данной выпускной квалификационной работе исследуются патологии грудной клетки и методы их диагностики при помощи искусственных нейронных сетей. Также в работе представлены результаты тестирования нескольких нейронных сетей.

Структура ВКР состоит из введения, трех глав, заключения, списка литературы.

Во введении описывается актуальность проводимого исследования.

В первой главе приводится анализ дифференциации патологий грудной клетки вручную, сопутствующие этому проблемы и их решение.

Во второй главе рассматриваются технические средства для реализации приложения и используемые в работе архитектуры нейронных сетей.

В третьей главе рассматривается процесс обучения и тестирования нейронных сетей с выбором наилучшей архитектуры для классификации патологий, а также описан метод выделения области интереса.

Объём дипломной работы 41 страниц, на которых размещены 20 рисунков и 3 таблицы. При написании данной работы использовалось 19 источников.

ABSTRACT

The title of the graduation work is “Differentiation of darkened on the X-ray”.

This graduation project is about categorizing of pathologies on chest X-rays by means of neural networks. The project should reduce the amount of missing pathologies in the analysis of X-ray images.

The author dwells on analysis of X-ray images by radiologist, the existing problems and their solutions. The key issue of the graduation project is the development of solutions for the categorizing of pathologies with different architectures of neural networks and distinguish of pathology on chest radiographs for radiologist. Thus, if the neural network detects pathology and distinguishes between the images, the doctor will be able to focus on that area and make a final decision for a patient.

The graduation project may be divided into several logically connected parts which are: description of the subject area and formalization of requirements for the software product, selection tools for development, description of a few architectures of neural networks and their differences, testing neural networks and selection best architecture for the project, description of a method for distinguishing pathology on chest X-rays images.

The result of this work is a model of convolution neural network for categorizing of pathology on image into 14 classes with an accuracy of an average of about 81.5%.

The developed network can be used by experts to reduce the likelihood of skipping pathology concentrating their attention on a particular area with the assumption of pathology situated in there.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. Анализ способов дифференциации затемнений рентгенограммы	6
1.1 Анализ снимка рентгенологом	6
1.2 Методы исследования дыхательных путей.....	7
1.3 Автоматизированная классификация изображений.....	8
1.4 Анализ нейросетевого подхода	10
2. Разработка алгоритма дифференциации	22
2.1 Выбор технических средств	22
2.2 Сравнение архитектур для классификации.....	26
2.3 Анализ датасета	32
3. Обучение нейронных сетей и тестирование моделей.....	34
3.1 Обучение нейронной сети для классификации	34
3.2 Выделение области интереса.....	36
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	40

ВВЕДЕНИЕ

Заболевания легких являются одной из самых больших угроз для здоровья человека. По данным Европейской Белой Книги по заболеваниям легких [1], из-за них происходит каждая шестая смерть во всем мире, а в России на каждые 100000 человек приходится около 90 человек с патологиями легких. Эти патологии приводят к инвалидности и даже могут стать причиной смерти, что влечет за собой огромные затраты на первую помощь, госпитализацию и лекарства. Кроме этого, огромный ущерб наносит потеря людьми трудоспособности и преждевременные смерти в результате болезни.

Цифровая рентгенография является одним из первых и основных вариантов диагностики различных заболеваний. Рентгенограммы позволяют увидеть даже самые незначительные изменения и не только выявить патологию на ранних стадиях, но и поставить точный диагноз заболевания. Именно поэтому рентгенографическое исследование широко распространено по всему миру и остается приоритетным в медицинской практике даже в хорошо оснащенных клиниках и медицинских центрах. Однако работа с рентгеновскими снимками – это постоянная работа с однородными изображениями, поэтому на стадии постановки диагноза есть вероятность, что болезнь может быть не замечена врачом. В связи с тем, что планово рентген делают лишь раз в год, незамеченная болезнь на ранней стадии может сильно прогрессировать до следующего обследования.

Чтобы снизить влияние человеческого фактора при анализе снимков и обнаружить как можно большее количество патологических процессов необходимо создать вспомогательный инструмент «Второй глаз», выступающий в роле интерна просматривающего снимки для выявления патологии и выделения области требующей внимания. Таким инструментом может стать ПО со сверточной нейронной сетью, способной выполнять

поставленные задачи без вмешательства со стороны. Особенно востребован такой инструмент будет при проведении массового обследования.

Глава 1 Анализ способов дифференциации затемнений рентгенограммы

1.1 Анализ снимка рентгенологом

Оценка одного снимка рентгенологом происходит в несколько этапов [2]:

1. общий просмотр снимка на экране монитора, подключенного к компьютеру. Специализированное программное обеспечение позволяет проводить преобразования изображения, такие как: изменение параметров отображения, измерения расстояний, объемов, углов;

2. использование специальных приложений для дополнительных расчетов физиологических или анатомических показателей, например определить объем камер сердца или степень сужения сосуда, оценить скорость накопления контрастного вещества опухолью. Эти сложные расчеты помогают сделать диагноз более достоверным;

3. составление квалифицированного заключения по исследованию — протокола или отчета. Он должен описывать все ключевые моменты исследования, с упоминанием как технических особенностей сканирования, так и выявленных признаков заболевания. В конце протокола делается вывод о том или ином патологическом состоянии.

В соответствии с Письмом Минздрава РФ от 31.08.2000г. № 2510/9736-32 «О нормировании труда специалистов рентгеновских кабинетов» общее количество исследований за смену должно составлять 25 исследований. Каждое исследование это тщательная работа над однородным изображением, в силу человеческого фактора даже опытный специалист со временем может перестать видеть незначительные отличия на снимке, что неизбежно влечет за собою снижение качества анализа информации, содержащейся в рентгенограмме. Также сложность в работу специалиста вносит качество снимков. Возможные артефакты и слабый контраст являются основными причинами плохого качества изображения, затрудняющие интерпретацию

медицинских изображений. Плохое качество изображения приводит к ненадежным результатам анализа и распознавания патологий.

С помощью автоматического распознавания патологии на снимке и выделении области с этой патологией, появляется возможность сконцентрировать внимание врача на ней и не допустить ее дальнейшего развития.

1.2 Методы исследования дыхательных путей

Для постановки диагноза, врачу нужно провести несколько исследований грудной клетки, ведь одного анализа жалоб мало, а разные заболевания могут иметь схожие симптомы. Врачу необходимо верно подобрать метод исследования, чтобы быстро определить патологию. Для исследования грудной клетки и обнаружения патологий может применяться несколько методик [3].

Одним из способов обнаружить изменение структуры легких или найти уплотнения является рентгеноскопия. В рентгеноскопии изображение создается с помощью рентгеновского аппарата и отображается на экране. Этот метод дает возможность в живую наблюдать за внутренними процессами. Главным недостатком метода является невозможность увидеть небольшие участки патологических процессов.

Этого недостатка лишена рентгенография позволяющая увидеть даже небольшие изменения в тканях. Благодаря рентгенографии обнаруживают аномалии в дыхательных путях, например опухоль, пневмонию или туберкулез, последствия травм. Изображение также как и в рентгеноскопии создается благодаря рентгеновскому аппарату, но не имеет такой динамики, т.к. снимок фиксируется на пленке, а не выводится на монитор. Рентгеновские снимки яркостью пикселей определяют различные структуры. Кости поглощают больше радиации, чем другие ткани и выглядят светлыми на плёнке, а через менее плотную ткань проходит больше излучения, оставляя на пленке более темные участки. Исследование грудной клетки с

помощью рентгенографии крайне эффективно для уточнения или постановки диагноза [4].

Еще одним рентгенологическим методом диагностики легких является компьютерная томография, позволяющая получить хорошее изображение поперечных «срезов» грудной клетки. Благодаря этому методу появляется возможность обнаружить изменения в легочной ткани, в лимфатических узлах, в бронхах и трахее.

Также для диагностики состояния бронхов иногда проводится бронхография, являющейся одним из вариантов рентгенографии. В бронхи вводят вещество не позволяющее пройти рентгеновским лучам. После чего делают рентгеновский снимок получая бронхиальное дерево. Все происходит под анестезией. Благодаря этому методу появляется возможность увидеть расширение бронхов, инородные тела и новообразования.

Самым частым методом применяемом в профилактических обследованиях является флюорография. Флюорография похожа на рентгенографию, но снимок выводится на монитор или на малоформатную пленку. У цифровой флюорографии более низкое радиационное воздействие на человека по сравнению с рентгенографией, но и качество снимка хуже не позволяя увидеть детали.

1.3 Автоматизированная классификация изображений

Наибольшее значение компьютерные технологии в медицине имеют в обработке цифровых изображений, например в улучшении качества снимка или распознавании отдельных элементов.

Одной из самых значимых задач анализа медицинских изображений является распознавание патологических процессов. Для этого чаще всего применяются компьютерные системы диагностики [5]. Эти системы могут производить сегментацию снимка, выделение объекта интереса, анализ и классификацию [6]. Классификация обычно проводится по методу опорных векторов и дискриминантного анализа [5, 7].

Классификация представляет собой задачу определения положения рентгеновского снимка в n -мерной пространстве признаков, состоящем из областей по количеству искомым патологий, где каждая область содержит в себе значения признаков для конкретной патологии.

Чтобы автоматически определять к какому классу относится снимок, необходимо сформировать области в пространстве признаков под каждую патологию. Для этого на размеченных данных (обучающей выборке) проводится обучение классификатора, который будет выделять из изображений признаки и соотносить метку патологии из выборки с пространством признаков формируя и помечая области под каждую патологию, с каждой итерацией обучения корректируя границы областей.

Чаще всего компьютерные системы диагностики применяется для обнаружения опухолей. К примеру, больницы используют его для профилактических осмотров и исследования внутренних органов. Однако у них есть существенный недостаток, специалисту по-прежнему приходится самому просматривать и обрабатывать снимки, представляющие из себя однородные изображения, для выделения областей интереса и распознавания патологии, что неминуемо влечет за собой ошибки.

Задача автоматизированной диагностики патологических процессов по данным медицинских изображений далека от своего разрешения. Одной из проблем является определение оптимальных методов параметрического описания объектов интереса, что может оказывать влияние на качество их классификации [6,8].

Одним из ключевых элементов в создании медицинских инструментов для эффективного получения результата является разработка автоматизированных методов диагностики и анализа медицинских изображений с использованием современных технологий, позволяя улучшить производительность облегчая диагностику и минимизировать медицинские ошибки, позволяя тем самым улучшить обслуживания пациентов.

1.4 Анализ нейросетевого подхода

Искусственная нейронная сеть — математическая модель, основанная на нейронной структуре мозга, способная обучаться выполнению предсказания, принятия решений, задач классификации, визуализации за счет обучения на имеющихся примерах.

Само понятие искусственной нейронной сети и модели искусственного нейрона было предложено в 1943 году У. Маккалоком и У. Питтсом в статье [9].

Чтобы отразить суть биологических нейронных систем, искусственный нейрон получает входные сигналы, это могут быть исходные данные или выходные сигналы других нейронов сети, через несколько входных каналов. Каждый входной сигнал проходит через соединение, имеющее определенный вес, называемый синапсом. С каждым нейроном связано определенное пороговое значение. Вычисляется взвешенная сумма входов и из нее вычитается пороговое значение, в результате получается величина активации нейрона. Сигнал активации преобразуется с помощью функции активации, которая нормализует сигнал на определенном диапазоне значений, и в результате получается выходной сигнал нейрона. Наглядный пример нейрона приведен на рисунке 1.1, где x_i — входной сигнал, w_i — вес входного сигнала, а f — функция активации.

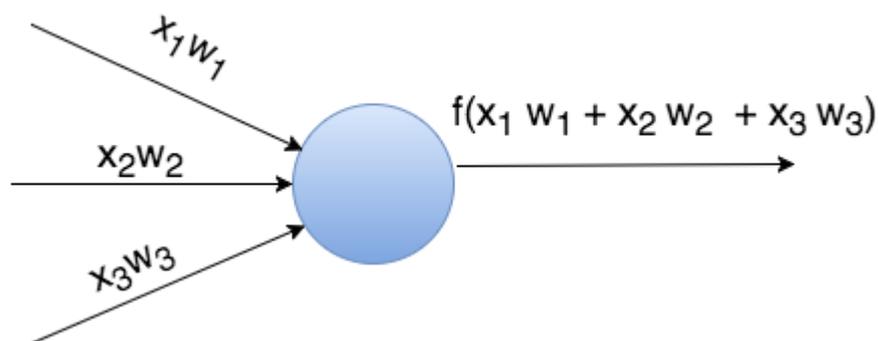


Рисунок 1.1 – Искусственный нейрон

Функция активации — это способ нормализации входных данных. Если на входе у нейрона будет большое число, пропустив его через функцию активации, на выходе оно выйдет в нужном диапазоне, в зависимости от

выбранной функции активации. Без функций активации глубокие сети теряют большую часть способности к обучению. Нелинейность этих функций позволяет повышать степень свободы, что дает обобщать проблемы высокой размерности в более низких измерениях.

Линейная функция активации выглядит следующим образом:

$$f(x) = x \quad \#(1.1)$$

Линейная функция активации почти никогда не используется, за исключением случаев, когда нужно протестировать нейронную сеть или передать значение без преобразований.

Сигмоида это самая распространенная функция активации, ее диапазон значений от 0 до 1:

$$f(x) = \frac{1}{1 + e^{-x}} \quad \#(1.2)$$

Также ее иногда называют логистической функцией. Если на выходе нейрона отрицательное значение, то нужна функция которая охватывает и отрицательные значения, такой может стать гиперболический тангенс.

Гиперболический тангенс имеет смысл использовать когда желаемые значения могут быть и отрицательными, и положительными, так как диапазон функции от -1 до 1:

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad \#(1.3)$$

Использовать эту функцию только с положительными значениями нецелесообразно так как это значительно ухудшает результаты нейросети.

ReLU (Rectified Linear Unit) позволяет побороть проблему затухания градиентов, свойственную вышеописанным функциям активации:

$$y(x) = \max(0, x) \quad \#(1.4)$$

Использование функции активации ReLU вместо более традиционных функций сигмоиды и гиперболического тангенса позволяет снизить количество эпох обучения в несколько раз.

В задачах классификации используется функция активации Softmax:

$$\sigma(n)_i = \frac{e^{n_i}}{\sum_{k=1}^K e^{n_k}} \#(1.5)$$

Softmax — это обобщение логистической функции для многомерного случая. Функция преобразует вектор n -размерности K в вектор σ той же размерности, где каждое значение σ_i полученного вектора представлено числом в интервале от 0 до 1, а сумма чисел равна 1. Значение σ_i полученного вектора трактуются как вероятности того, что объект принадлежит к классу i . Для задачи классификации сеть строят таким образом, что на последнем слое количество нейронов оказывается равным количеству искомых классов.

Несколько нейронов сгруппированных вместе образуют слой нейронной сети, а несколько слоев образуют многослойный персептрон, как показано на рисунке 1.2. У любой сети есть входной слой, которому на вход подается информация для анализа, n скрытых слоев, которые ее обрабатывают и выходной слой, который выводит результат работы сети. У каждого из нейронов есть 2 основных параметра: входные данные и выходные данные. Для входного нейрона входная информация — это информация, которую подают в нейронную сеть. В остальных нейронах, на вход попадает суммарная информация всех нейронов с предыдущего слоя, после чего, она нормализуется, с помощью функции активации и попадает на выход.

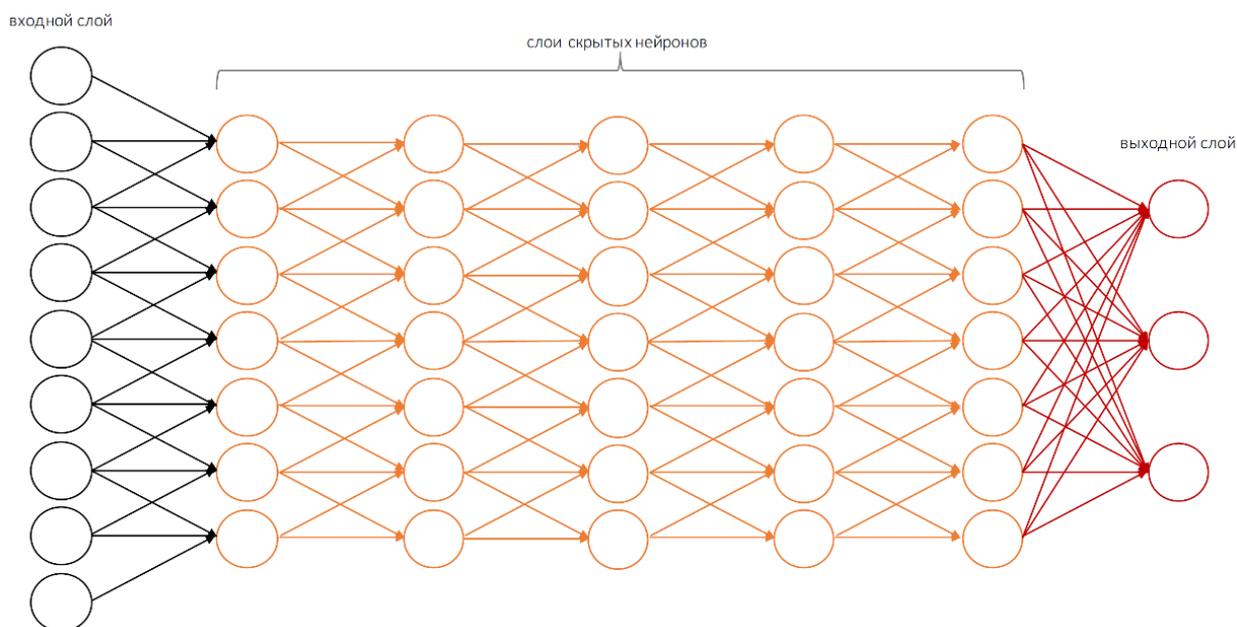


Рисунок 1.2 – Слои объединённые в многослойный перцептрон

В сети каждый нейрон соединен с последующим с помощью синапса. У синапсов есть 1 параметр — вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. К примеру для 3 нейронов, которые передают информацию следующему. Тогда есть 3 веса, соответствующие каждому из этих нейронов. Нейрон с большим весом будет приносить больший вклад в следующем нейроне чем другие, пример продемонстрирован на рисунке 1.3. Совокупность весов нейронной сети или матрица весов — это своеобразный мозг всей системы. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат.

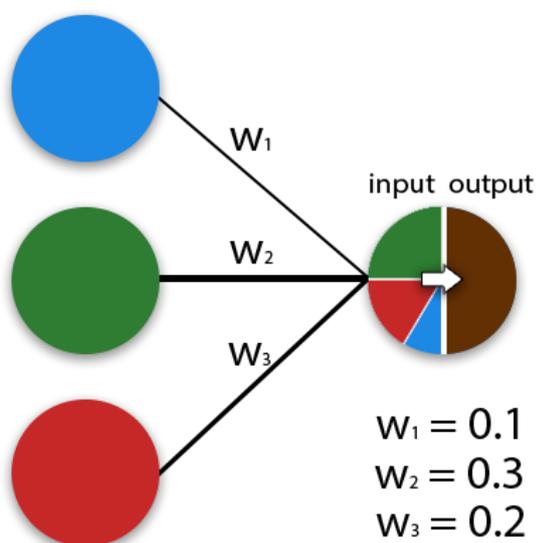


Рисунок 1.3 – Влияние синапсов на нейрон

После того как сеть выдает результат анализа, переданной ей информации, нужно узнать насколько хорошо сеть проанализировала информацию, т.е. посчитать ошибку сети по одной из функции потерь.

Ошибка — это процентная величина, отражающая расхождение между ожидаемым и полученным результатами. Цель обучения сети - минимизировать эту ошибку. Таким образом, функция потерь эффективно приближает обучение нейронной сети к этой цели. Функция потерь измеряет то насколько хороша нейронная сеть в отношении данной обучающей выборки и ожидаемых ответов. Она также может зависеть от таких переменных, как веса и смещения.

Самые распространенные функции потерь: MSE(среднеквадратичная ошибка), RMSE и Arctan:

$$MSE = \frac{(t_1 - p_1)^2 + (t_2 - p_2)^2 + \dots + (t_n - p_n)^2}{n} \quad \#(1.6)$$

$$RMSE = \sqrt{\frac{(t_1 - p_1)^2 + (t_2 - p_2)^2 + \dots + (t_n - p_n)^2}{n}} \quad \#(1.7)$$

$$Arctan = \frac{\arctan^2(t_1 - p_1) + \dots + \arctan^2(t_n - p_n)}{n} \quad \#(1.8)$$

где t - ожидаемый результат, а p – выход нейронной сети.

Каждый метод считает ошибки по-разному. У Arctan, ошибка, почти всегда, будет больше, так как он работает по принципу: чем больше разница, тем больше ошибка. У RMSE будет наименьшая ошибка, поэтому, чаще всего, используют MSE, которая сохраняет баланс в вычислении ошибки.

Для оценки качества классификаторов применяют категориальную крос-энтропийную функцию потерь:

$$CrossEntropy = - \sum_i^n t_i \log(p_i) \quad \#(1.9)$$

где p – предсказание нейронной сети, а t – ожидаемый ответ.

Принцип подсчета ошибки во всех случаях одинаков. Каждый раз считается ошибка и отнимется от ожидаемого ответа. Далее либо возводится в квадрат, либо вычисляется квадратный тангенс из этой разности, после чего полученное число делится на количество проходов.

Рассмотрим несколько типов нейронных сетей и какие задачи они решают.

Многослойный перцептрон может состоять из 3 и более слоев. Он использует функцию активации, позволяющую классифицировать линейно неразделимые данные. Каждый нейрон в слое соединен с каждым последующим нейроном, такая сеть называется полносвязной. Многослойный перцептрон находит применение во многих задачах, например в машинном переводе, распознавание речи, классификации.

Рекурсивные нейронные сети обрабатывают данные переменной длины. Модели рекурсивных сетей используют иерархические структуры образцов при обучении. Такие модели применяются при обучении последовательных структур и деревьев в задачах обработки естественного языка, где фразы и предложения моделируются через векторное представление слов.

Рекуррентная нейронная сеть представляет из себя вариант рекурсивной нейронной сети, где связи между элементами образуют направленную последовательность. Из-за чего появляется возможность обрабатывать серии событий во времени или последовательные цепочки. Выходная информация в такой сети зависит не только от текущего входа, но также от состояний нейрона на предыдущем шаге. Такая память позволяет решать задачи распознавания речи или рукописного текста.

Сеть долгой краткосрочной памяти (LSTM) представляет из себя разновидность архитектуры рекуррентной нейросети. LSTM создавалась для более точного моделирования временных последовательностей и их долгосрочных зависимостей, чем рекуррентная сеть. LSTM не использует функцию активации в рекуррентных компонентах, сохраненные значения не

модифицируются, а градиент не стремится исчезнуть во время тренировки. В статье [10] авторы показывают архитектуру глубокой LSTM, которая достигает хороших результатов для крупномасштабного акустического моделирования.

Лучше всего при работе с изображениями, такими как рентгеновские снимки, показали себя сверточные нейронные сети. Ян Лекун представил их в 1998 году, для распознавания чисел присутствующих во входном изображении [11]. Также сверточные нейронные сети применяются для распознавания речи, сегментации изображения, классификации и обработки текста. До появления сверточных нейронных сетей многослойные перцептроны использовались для классификации изображений. Однако многослойные перцептроны используют огромное количество ресурсов для поиска информации в изображениях, т.к. каждый вход связан с каждым нейроном в следующем слое. Сверточные нейронные сети превзошли их, используя концепцию называемую локальной связностью, каждый нейрон подключается только к локальной области входов. Такое решение уменьшает количество параметров, позволяя разным частям сети специализироваться на высокоуровневых признаках, таких как повторяющийся узор или текстура. Таким образом сверточные нейронные сети используют в работе некие фильтры, помогающие им распознавать определенные признаки на изображении.

Фильтр представляет из себя матрицу чисел. Каждый фильтр по мере обучения корректирует значения в матрицах для поиска на изображениях определенных признаков. Фильтр передвигается по изображению и определяет наличие некоторого признака в конкретной части изображения. Для нахождения признака используется операция свертки, представляющая из себя сумму поэлементного произведения входной матрицы и матрицы фильтра, что продемонстрировано на рисунке 1.4.

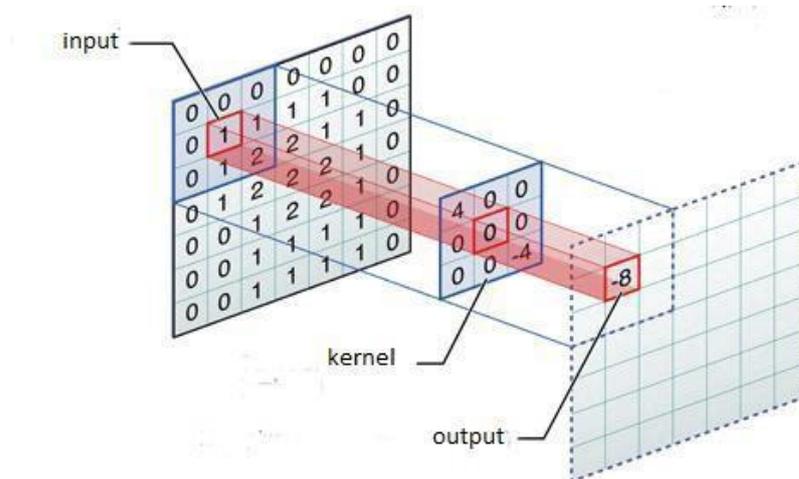


Рисунок 1.4 – Операция свертки

В многослойном перцептроне каждый узел отвечает за понимание всей картины, а в сверточной нейронной сети изображение разбивается на области. Выходной слой объединяет полученные от фильтров данные чтобы найти закономерности. Если некоторый признак обнаруживается во фрагменте изображения, то результат свертки на выходе будет иметь число с большим значением, чем если бы признака не было. Результатом движения фильтра по изображению является матрица, состоящая из результатов сверток.

Ядро свертки двигаясь по входной матрице обрезает ее по краям из-за того что они никогда не попадут в центр ядра. Чтобы этого избежать и получить выходную матрицу такого же размера что и входную используют паддинг. Паддинг добавляет вокруг изображения пиксели со значением нуль, как показано на рисунке 1.5. Благодаря чему все пиксели входной матрицы оказываются в центре ядра свертки и выходная матрица остается того же размера, что и входная.

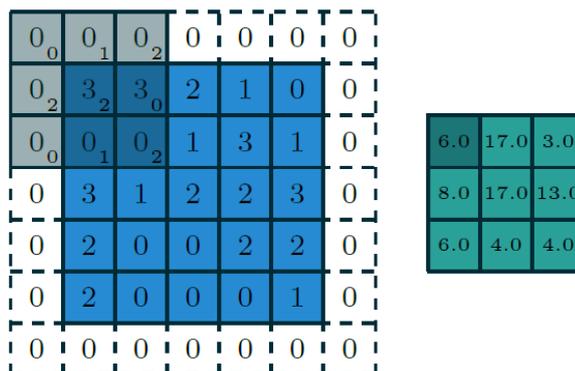


Рисунок 1.5 – Добавление паддинга

Не всегда фильтр перемещается вдоль матрицы входных сигналов с шагом равным единице. Шаг перемещения фильтра называется страйдом. Страйд определяет, на какое количество пикселей должен сдвинуться фильтр за один шаг. Иногда при работе со сверточным слоем, нужно получить выходную матрицу меньшего размера. Это используется в сверточных нейронных сетях, где размер матриц уменьшается при увеличении количества каналов, этого можно добиться с помощью страйда. Смысл страйда заключается в том, чтобы пропустить некоторые области, над которыми двигается ядро свертки. Шаг 1 означает, что фильтр проходит через каждый пиксель. Шаг 2 означает, что фильтр перемещается на каждый второй пиксель пропуская промежуточные в процессе и уменьшая их количество в 2 раза, как показано на рисунке 1.6.

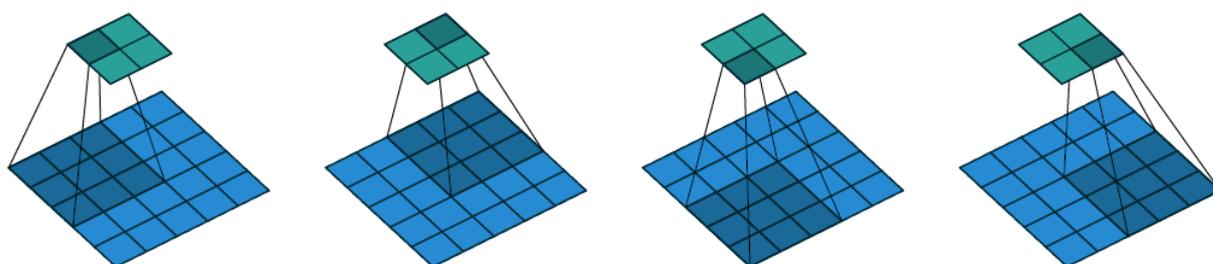


Рисунок 1.6 – Применение свертки с шагом 2

Размер выходной матрицы после свертки может быть рассчитано по формуле и из нее же могут быть выражены другие параметры свертки:

$$out = \frac{in - f + 2 * p}{s} + 1 \#(1.10)$$

где out – размер матрицы на выходе, in – размер матрицы на входе, f – размер матрицы ядра, p – размер паддинга, s – страйд свертки.

После нескольких сверточных слоев зачастую производится редуцирование промежуточного результата. Представляющее из себя понижение количества данных с сохранением общего смысла. В машинном обучении это называется даунсемплингом. Даунсемплинг делают с целью ускорения процесса обучения и уменьшения потребляемых вычислительных ресурсов промежуточных или исходных данных. Самым распространенным

способом сделать это является максимальное объединение (max pooling). Макспулинг заключается в перемещении вдоль данных так называемого окна просеивания. Из этого окна берется пиксель имеющий самое большое значение и перемещается в выходную матрицу, рисунок 1.7. Страйд не всегда имеет является единичным, он может меняться в зависимости от того какого размера выходную матрицу нужно получить и как сильно необходимо сжать данные. Благодаря макспулингу нейронная сеть способна концентрироваться на значимых признаках в изображении, не беря во внимание несущественные детали.

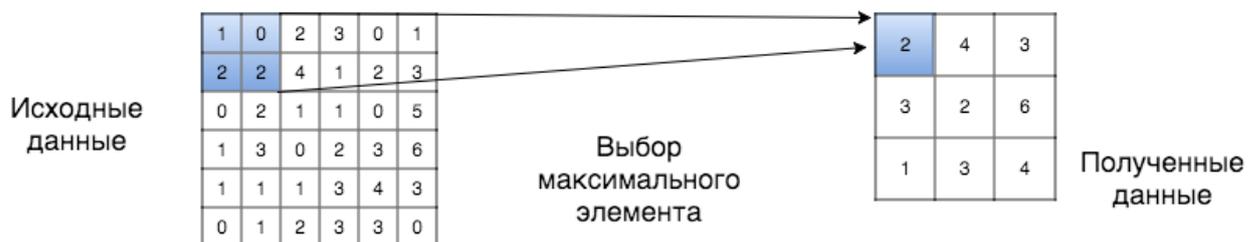


Рисунок 1.7 – Применение максимального объединения

Обратной операцией к даунсеплингу является апсеплинг (unpooling, upsampling). После выполнения даунсеплинга уменьшается пространственная размерность изображения. Например, если ядро даунсеплинга имеет размер 2×2 , то размеры изображения уменьшатся в два раза вдоль каждого измерения. Выполняя повышающую дискретизацию, увеличивают размеры изображения, повторяя значение пикселя в некоторой окрестности, как на рисунке 1.8.

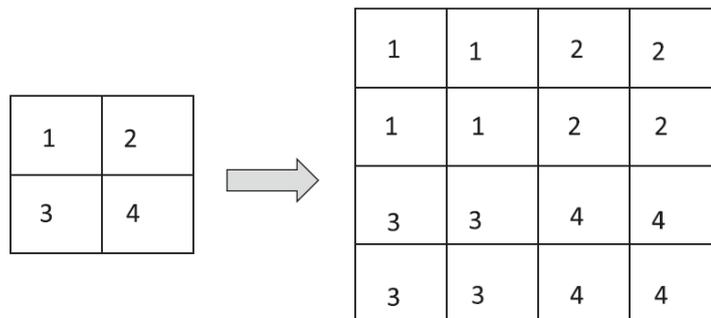


Рисунок 1.8 – Применение повышающей дискретизации

Сверточные слои часто обладают симметрией, поскольку операция даунсеплинг, выполняемая в первой части сети, дополняется операцией

повышения дискретизации во второй для восстановления размера изображения. На выходе невозможно понять точное расположение пикселя с максимальным значением, а значит в этом процессе теряется некоторая информации о входе. Одним из способов восстановления утерянной информации на стадии реконструирования изображения посредством повышающей дискретизации является помещение значения входного пикселя в выходное расположение, соответствующее тому расположению, из которого выход даунсеплинга получил свой вход. Этот метод повышения дискретизации называют повышением дискретизации по максимуму (max unpooling).

Апсеплинг выполняемый повышающей дискретизации или повышающей дискретизации по максимуму, является неизменными преобразованиям, т.е. эти преобразования не включают никаких параметров для обучения нейронной сети. Способ дающий возможность обучения сети заключается в выполнении интерполяции посредством транспонированной свертки, подобной операциям свертки. Транспонированная свертка содержит параметры, которым сеть должна обучаться, нужно научить сеть выполнять интерполяцию таким способом, чтобы уменьшить функцию потерь.

В отличие от обычной свертки, когда вычисляется точечное произведение фильтра и части входа, для транспонированной свертки значения фильтра при каждом расположении взвешиваются входным значением из той позиции, в которой находится фильтр, и эти взвешенные значения фильтра заполняют соответствующие расположения в выходе. Выходы для последовательных входных значений вдоль того же пространственного измерения размещаются с зазором, определяемым величиной шага транспонированной свертки. Эти действия выполняются по отношению ко всем входным значениям. В конце выходы, соответствующие каждому из входных значений, складываются для получения конечного выхода как на рисунке 1.9.

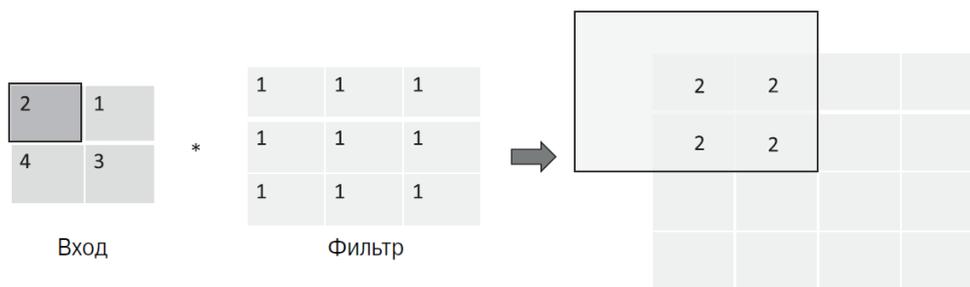


Рисунок 1.9 – Применение транспонированной свертки

После всех сверточных слоев и блоков даунсемплинга, карта признаков изображения будет развернута в вектор для последующей передачи в полносвязную нейронную сеть, как на рисунке 1.10.

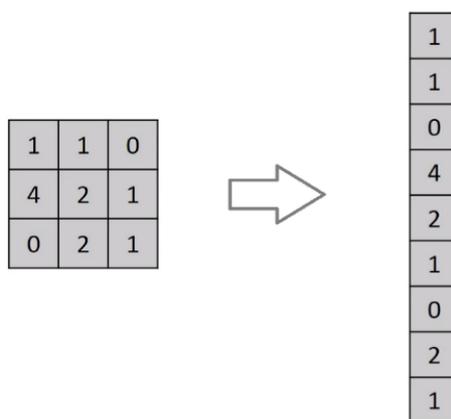


Рисунок 1.10 – Развертывание двумерной матрицы в вектор

Полностью архитектура сверточной нейронной сети для классификации показана на рисунке 1.11. Все сверточные нейронные сети похожи по принципу работы и развивались от этой архитектуры дополняя новые приемы по работе, меняя слои или делая сеть глубже увеличивая количество слоев.

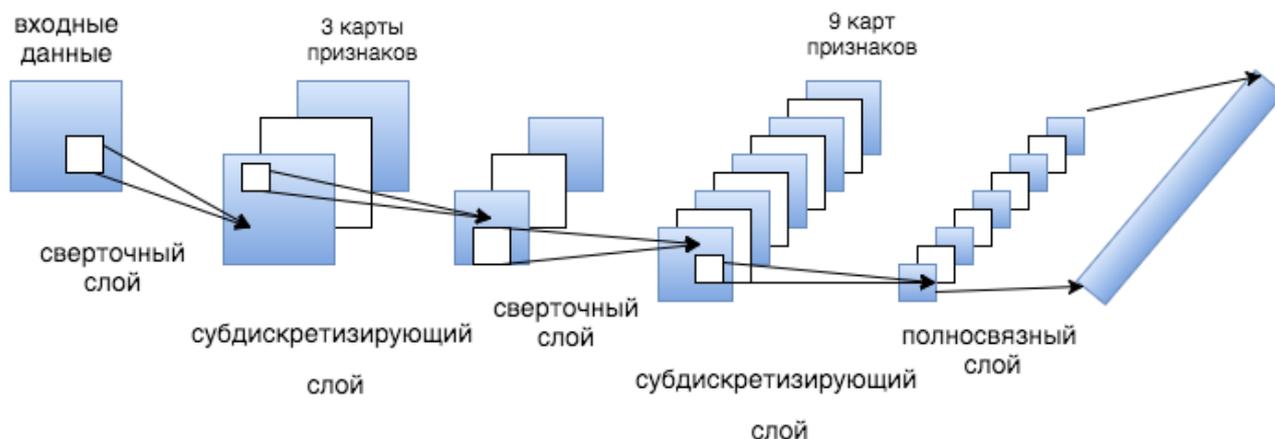


Рисунок 1.11 – Архитектура свстрочной нейронной сети

Глава 2 Разработка алгоритма дифференциации

2.1 Выбор технических средств

Для решения задачи классификации патологий рассматриваются три языка программирования, а именно: Python, MATLAB и R.

MATLAB — это коммерческая вычислительная среда и язык программирования. Концепция MATLAB относится ко всему пакету, включая IDE. Стандартная библиотека не содержит общих функций программирования, но включает матричную алгебру и обширную библиотеку для обработки и построения графиков. MATLAB применяется, главным образом в инженерных расчетах.

Преимущества MATLAB:

1. простой и лаконичный язык для всего, что связано с матричными операциями;
2. широкий спектр библиотек;
3. удобный графический интерфейс и простота в работе.

Недостатки MATLAB:

1. дороговизна лицензий;
2. неполная поддержка статистических функций.

R — объектно-ориентированный open source язык программирования для статистического анализа в финансовом секторе.

Преимущества R:

1. язык был создан специально для анализа данных;
2. множество полезных для анализа данных функций сразу встроены в язык;
3. множество готовых тестов под большинство методов Data Science;
4. эффективная работа с векторами и матрицами;
5. несколько хороших пакетов для визуализации.

Недостатки R:

1. невысокая производительность;
2. узкоспециализированный язык в следствии чего весьма специфичен;

3. синтаксис для решения некоторых задач не очевиден;
4. из-за большого количества библиотек, документация наименее популярных из них неполная.

Python, высокоуровневый язык программирования, ориентированный на читаемость кода и повышение скорости разработки.

Преимущества Python:

1. многоцелевой язык позволяющий производить обработку данных, их поиск и использовать результат обработки в приложении;
2. Python является языком с динамической типизацией, что ускоряет разработку приложений;
3. автоматическое управление памятью;
4. кроссплатформенная поддержка;
5. бесплатный с открытым исходным кодом;
6. повышение производительности за счет использования библиотек написанных на более низкоуровневых языках, таких как C;
7. большое количество библиотек для Data Science, таких как SciPy, NumPy, Matplotlib.

Недостатки Python:

1. нет альтернатив для множества библиотек имеющихся в языке R;
2. динамическая типизация не только ускоряет разработку, но еще и усложняет поиск ошибок связанных с присваиванием различных данных одним и тем же переменным.

Благодаря множеству библиотек, простоте освоения и разработки, в качестве языка реализации дипломного проекта был выбран язык Python.

Для работы с изображениями была выбрана библиотека OpenCV.

OpenCV (англ. Open Source Computer Vision Library) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также имеет интерфейсы на различных языках, среди которых есть Python,

Java, C++ и Matlab. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.

OpenCV — библиотека с лёгкими алгоритмами, которые могут использоваться в 3D-рендере, профессиональном редактировании изображений и видео, отслеживании и идентификации объектов и людей на видео, поиске идентичных изображений из набора, также эта библиотека очень важна для тех, кто разрабатывает проекты, связанные с машинным обучением в области изображений. В неё входят более 2500 алгоритмов, в которых есть как классические, так и современные алгоритмы для компьютерного зрения и машинного обучения. Фактически, OpenCV — это набор типов данных, функций и классов для обработки изображений алгоритмами компьютерного зрения.

В качестве инструмента разработки и обучения нейронных сетей был выбран PyTorch.

PyTorch является библиотекой машинного обучения для языка Python с открытым исходным кодом, разрабатываемая командой по искусственному интеллекту от Facebook. PyTorch используется для решения множества задач, например в компьютерном зрении или обработке естественного языка. Вокруг PyTorch создана некая экосистема включающая в себя различные библиотеки, разрабатываемые сторонними командами, такими как: Fast.ai упрощающая процесс обучения моделей, Pyro модуль для вероятностного программирования от Uber, Flair для обработки естественного языка и Catalyst для обучения DL и RL моделей.

PyTorch основывается на двух высокоуровневых моделях:

1. тензорные вычисления с поддержкой ускорения на GPU. Тензоры библиотеки схожи с массивами библиотеки numpy, но дополнительно могут обрабатываться на видеокартах;

2. глубокие нейронные сети с использованием метода автоматической дифференциации. Производится запись вычислений, произведенных в прямом направлении, затем производится воспроизведение в обратном

порядке для вычисления градиентов. Этот метод полезен при построении нейронных сетей, так как позволяет рассчитывать дифференциальные поправки параметров одновременно с прямым проходом.

PyTorch отличается от других фреймворков машинного обучения тем, что в нем не используются статические расчетные графы определяемые заранее как в TensorFlow, Caffe2 или MXNet. Расчетные графы в PyTorch динамические и определяются на лету. Таким образом, при каждом вызове слоев в модели PyTorch динамически определяется новый расчетный граф. Этот граф создается имплицитно – то есть, библиотека сама записывает поток данных, идущих через программу, и связывает вызовы функций вместе в расчетный граф.

PyTorch также предоставляет TorchScript, который может использоваться для запуска моделей независимо от среды выполнения Python. Это можно рассматривать как виртуальную машину с инструкциями, в основном специфичными для тензоров. Также есть возможность конвертировать модели, обученные с помощью PyTorch в форматы, такие как ONNX, которые позволяют использовать эти модели в других фреймворках машинного обучения, таких как MXNet, CNTK, Caffe2 или Tensorflow.

Для обучения нейронных сетей предполагается использование специализированной платформы для этого, такой стала Google Colab.

Google Colaboratory — это облачный сервис, направленный на упрощение исследований в области машинного и глубокого обучения. Используя Colaboratory, можно получить удаленный доступ к машине с подключенной видеокартой совершенно бесплатно, что сильно упрощает глубокое обучение нейросетей. Можно сказать, что Colab является некоторым аналогом гугл-документов для Jupyter Notebook, командной оболочки для интерактивных вычислений на Python.

Наличие GPU ускорителя является критическим фактором для скорости обучения deep learning моделей. Без GPU обучение нейросети займет многие часы/дни и не позволит полноценно экспериментировать со

структурой сети. Объем видеопамати так же важен. Чем больше памяти, тем больший размер батча можно установить и использовать более сложные модели.

В Colaboratory предустановлена PyTorch, OpenCV и практически все необходимые для работы Python-библиотеки, а код выполняется на сервере в Docker контейнере.

Для обучения нейронных сетей необходимы специально отобранные и размеченные данные, относящиеся к той или иной области. Такие данные может предоставить специализированный ресурс Kaggle.

Kaggle — система организации конкурсов по исследованию данных, а также социальная сеть специалистов по обработке данных и машинному обучению. Принадлежит корпорации Google.

Kaggle организована как публичная веб-платформа, на которой пользователи и организации могут публиковать наборы данных, исследовать и создавать модели, взаимодействовать с другими специалистами по данным и инженерами по машинному обучению, организовывать конкурсы по исследованию данных и участвовать в них. В системе размещены наборы открытых данных, предоставляются облачные инструменты для обработки данных и машинного обучения. Также реализованы обучающие ресурсы, имеется раздел для размещения вакансий работодателями, где тоже возможна организация конкурсов для отбора наилучших кандидатов.

2.2 Сравнение архитектур для классификации

Для решения задачи классификации были взяты на рассмотрение три архитектуры сверточных нейронных сетей, а именно VGG, ResNet, DenseNet, архитектура показавшая наилучший результат будет использоваться для классификации патологий на рентгеновском снимке.

VGG — наиболее часто используемое семейство архитектур, из-за его простоты, предложенная К. Simonyan и А. Zisserman из Оксфордского университета в статье [12]. Она является улучшенной версией AlexNet, в которой заменены большие фильтры на несколько фильтров размера 3x3,

следующих один за другим. AlexNet — сверточная нейронная сеть, которая оказала большое влияние на развитие машинного обучения, в особенности — на алгоритмы компьютерного зрения. Архитектура AlexNet схожа с созданной Яном Лекуном сетью LeNet. Однако у AlexNet больше фильтров на слое и вложенных сверточных слоев.

Архитектура самой старшей модели семейства VGG, VGG19 показана на рисунке 2.1.

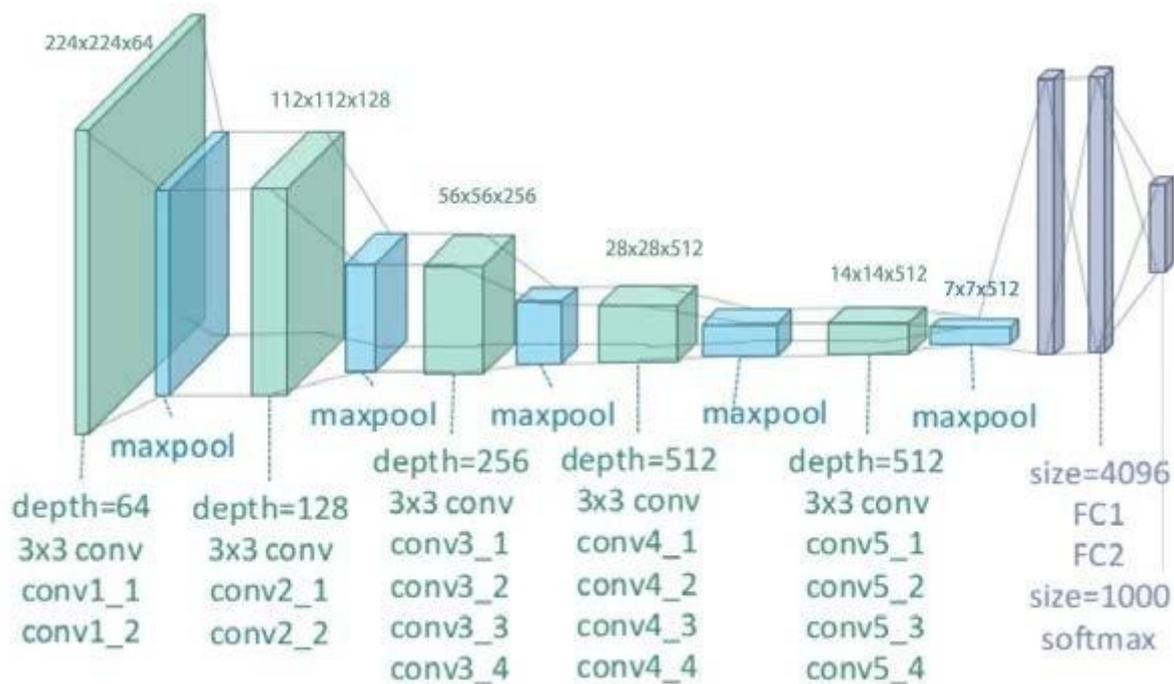


Рисунок 2.1 – Архитектура VGG19

На вход слоя свертки conv1 подается изображение размером 224x224. Далее изображения проходят через несколько сверточных слоев, в которых используются фильтры размером 3x3 и функция активации ReLU. Шаг свертки равен 1. Паддинг входа сверточного слоя берется такой, что пространственное разрешение сохраняется после свертки, т.е. паддинг равен единице для матриц 3x3 сверточных слоев. Даунсеплинг осуществляется при помощи пяти max-pooling слоев, которые следуют за каждым сверточным слоем. Операция max-pooling выполняется на окне размера 2x2 пикселей с шагом 2. После нескольких сверточных слоев, которые имеет разную глубину, идут три полносвязных слоя: первые два имеют по 4096 каналов, третий — 1000 каналов под каждый класс, так как в соревновании ILSVRC

требуется классифицировать объекты по 1000 классам. Последним идет softmax слой.

К недостаткам VGG семейства относятся маленькая скорость обучения и большой вес модели из-за большой глубины архитектур. При этом увеличение количества слоев может повлечь за собой ухудшение обучаемости модели из-за затухания градиента по мере прохождения в глубь сети, как показано на рисунке 2.2.

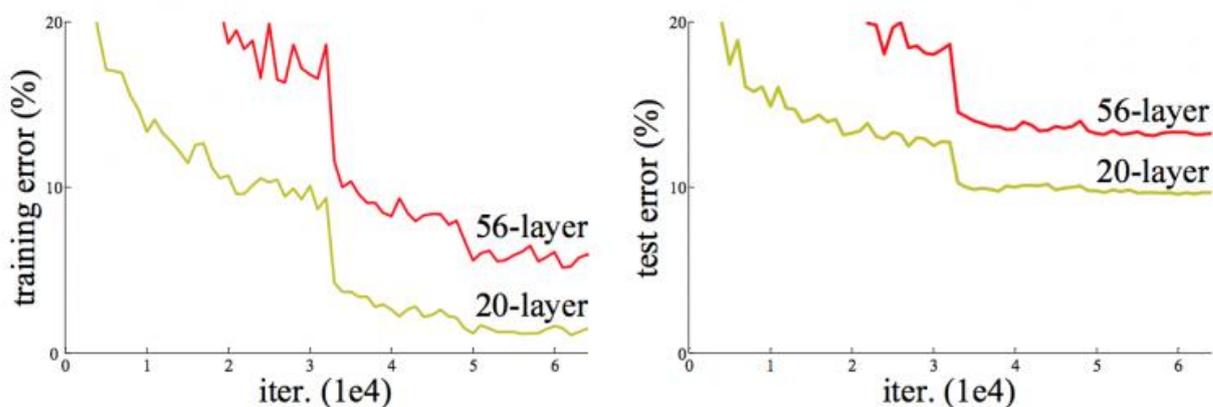


Рисунок 2.2 – Графики изменения обучаемости

Когда очень глубокая сверточная сеть начинает сворачиваться, возникает проблема, с увеличением глубины сети точность сначала увеличивается, а затем быстро ухудшается.

Чтобы решить эту проблему, Microsoft представила глубокую остаточную структуру обучения. Вместо того, чтобы надеяться на то, что каждые несколько уровней непосредственно соответствуют желаемому основному представлению, они явно позволяют этим слоям соответствовать остаточному [13].

Соединения быстрого доступа (shortcut connections) пропускают один или несколько слоев и выполняют сопоставление идентификаторов. Их выходы добавляются к выходам сверточных слоев создавая остаточный блок (Residual block), если матрица на входе сверточного блока такого же размера что и свертки в блоке. Если матрица на входе больше выходной матрицы, то перед сложением на входной матрице проводят свертку с матрицей

единичного размера и страйдом равным 2 для уменьшения размера входной матрицы в два раза, как показано на рисунке 2.3.

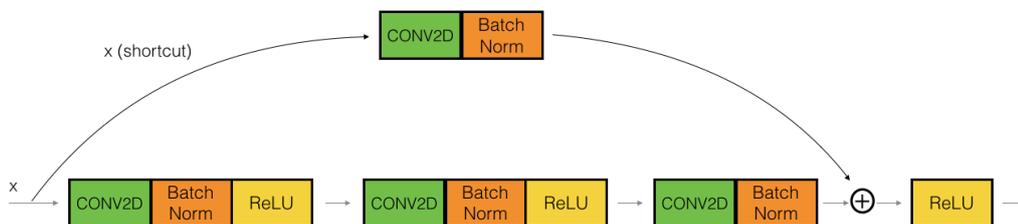


Рисунок 2.3 – Сопоставление входной и выходной матрицы

Нейросеть состоит из большого числа одинаковых остаточных блоков, каждый из которых имеет два свёрточных слоя 3×3 . Периодически число фильтров удваивается, а их размерность уменьшается с шагом 2, чтобы сохранить временную сложность каждого слоя. В самом начале архитектуры присутствует дополнительный свёрточный слой. Также у ResNet нет полносвязных слоёв в конце — используется только один слой с выходными классами. С увеличением числа слоёв для уменьшения размерности изображения применяются точно такие же дополнительные слои, как и в GoogleNet.

Общая архитектура ResNet с остаточными блоками продемонстрирована на рисунке 2.4.

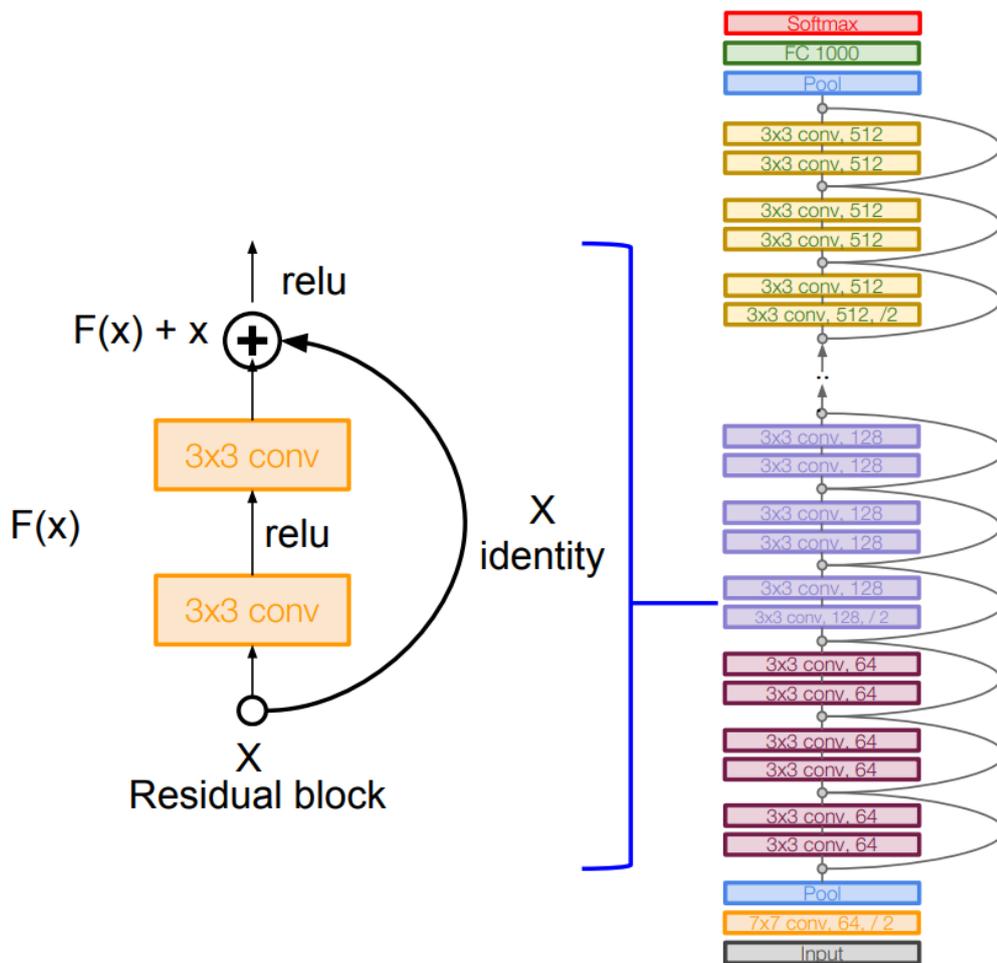


Рисунок 2.4 – Архитектура ResNet

Используя ResNet, можно решить множество проблем, таких как:

1. ResNet относительно легко оптимизировать: «простые» сети (которые просто складывают слои) показывают большую ошибку обучения, когда глубина увеличивается;
2. ResNet позволяет относительно легко увеличить точность благодаря увеличению глубины, чего с другими сетями добиться сложнее.

DenseNet (Densely Connected Convolutional Network) была предложена совместно Университетом Корнуэлл, Университетом Цинхуа и Facebook AI Research (FAIR) в 2017 году [14], архитектура сети показана на рисунке 2.5.

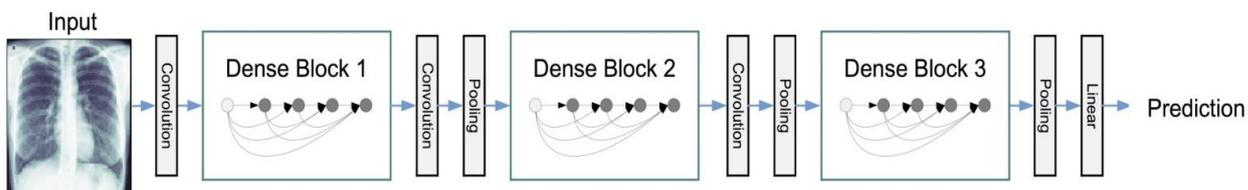


Рисунок 2.5 – Архитектура DenseNet

Успех ResNet позволил предположить, что укороченное соединение в сверточных нейронных сетях позволяет обучать более глубокие и точные модели. Авторы проанализировали это наблюдение и представили компактно соединенный блок (Dense Block), который соединяет каждый слой с каждым последующим слоем, как показано на рисунке 2.6, где k (Growth Rate) это скорость роста, заключающаяся в дополнительном количестве каналов для каждого слоя.

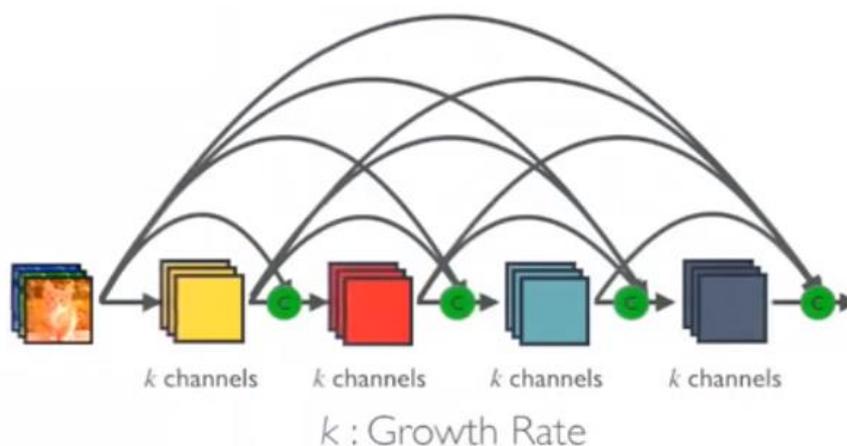


Рисунок 2.6 – Плотный блок сверток

Свертки в блоке такие же как в архитектурах описанных ранее, с ядром размера 3×3 и активацией ReLu.

В отличие от ResNet признаки, прежде чем они будут переданы в следующий слой, не суммируются, а конкатенируются в единый тензор. При этом количество параметров сети DenseNet намного меньше, чем у сетей с такой же точностью работы. Авторы утверждают, что DenseNet работает особенно хорошо на малых наборах данных.

В DenseNet каждый слой получает дополнительные входные данные от всех предыдущих слоев и передает свои собственные карты признаков всем последующим слоям, таким образом каждый слой получает знания от всех предыдущих слоев.

Так как каждый слой получает карты характеристик от всех предыдущих уровней, сеть может быть более компактной, то есть число каналов может быть меньше чем в других архитектурах при той же точности.

Таким образом, блок имеет более высокую вычислительную эффективность и использовать меньше памяти.

В конце последнего плотного блока выполняется глобальное среднее объединение (avgpool) после которого следует softmax.

Сравнение характеристик 3 архитектур по размеру, количеству параметров и точности на датасете ImageNet [15], приведены в таблице 2.1.

Таблица 2.1 – Сравнение архитектур сверточных нейронных сетей

Архитектура	Размер	Топ-1 точности	Топ-5 точности	Количество параметров
VGG16	528 MB	0.713	0.901	138357544
VGG19	549 MB	0.713	0.900	143667240
ResNet50	98 MB	0.749	0.921	25636712
ResNet152	232 MB	0.766	0.931	60419944
DenseNet121	33 MB	0.750	0.923	8062504
DenseNet201	80 MB	0.773	0.936	20242984

Как видно из таблицы семейство архитектур VGG значительно тяжелее остальных и уступает им в точности. При этом отчетливо видна проблема затухания градиента в ней, более глубокая сеть с большим количеством параметров показывает результаты хуже, чем младшая сеть семейства.

Этих недостатков лишены архитектуры ResNet и DenseNet, при этом DenseNet имеет намного меньший вес и количество параметров чем ResNet и как следствие большую скорость работы.

Таким образом для решения задачи классификации патологии будут рассматриваться две архитектуры: ResNet и DenseNet.

2.3 Анализ датасета

Для обучения модели классификации использовался датасет NIH Chest X-rays, с портала Kaggle, который был подготовлен Клиническим Центром в составе Национальных Институтов Здравоохранения США (NIH Clinical Center) [16]. Данный датасет содержит 112120 снимков от 30805 пациентов и

имеет 2 вариации на 8 и 14 патологий, разница между ними по патологиям и количеству снимков описана в таблице 3.1.

Таблица 3.1 – Сравнение Chest X-rays8 и Chest X-rays14

Патология	X-ray8	X-ray14
Atelectasis	5789	11535
Cardiomegaly	1010	2772
Effusion	6331	13307
Infiltration	10317	19871
Mass	6046	5746
Nodule	1971	6323
Pneumonia	1062	1353
Pneumothorax	2793	5298
Consolidation		4667
Edema		2303
Emphysema		2516
Fibrosis		1686
Pleural Thickening		3385
Hernia		227
No findings	84312	60412
Всего	108948	112120

Изображения в данном датасете одноканальные, имеют размер 1024×1024, в формате PNG.

Снимки в этом датасете могут содержать не только одну патологию, на некоторых из них по 2 и 3 патологии, поэтому задача классификации на нем не совсем обычная. Чтобы решить ее Stanford ML Group предложил вместо классической softmax после полносвязного слоя использовать сигмоиду для каждого выхода слоя таким образом модель выдает вероятности нахождения каждой патологии на снимке, т.е. для датасета с 14 патологиями на выходе модели должен быть вектор из 14 значений от 0 до 1, где каждое значение соответствует вероятности нахождения патологии на снимке. Такое решение для этого датасета было названо CheXNet [17].

Глава 3 Обучение нейронных сетей и тестирование моделей

3.1 Обучение нейронной сети для классификации

Для обучения классификатора будет использовано решение Stanford ML Group, показавшее хорошие результаты на датасете превзошедшие рентгенологов по некоторым патологиям.

Обучение сети происходило на 106772 снимков и тестировалась на 5348 в течении 20 эпох. За один проход обучения и тестирования на вход сети поступал батч размером в 16 снимков. В качестве оптимизации использовался метод Adam. Adam — adaptive moment estimation, сочетающий в себе идею накопления движения и идею более слабого обновления весов для типичных признаков [18].

Ошибка выхода сети оценивалась по бинарной кросс-энтропии (BCE):

$$BCE = -\frac{1}{N} \sum_{i=1}^N t_i \cdot \log(p(t_i)) + (1 - t_i) \cdot \log(1 - p(t_i)) \quad \#(3.1)$$

где t – ожидаемый результат 1 или 0, в зависимости от того есть патология на снимке или нет, p – предсказание сети для патологии.

В качестве аугментации для снимков использовался случайный поворот снимка. Под аугментацией понимается увеличение выборки для обучения через модификацию существующих данных.

Начальная скорость обучения равна 0.001 и уменьшалась в 10 раз если в течении двух эпох не происходило уменьшение ошибки сети на тестовой выборке.

Для сравнения были взяты самая старшая модель архитектуры ResNet (ResNet152) и самая старшая модель архитектуры DenseNet (DenseNet201).

Оби модели обучались 20 эпох на одном разбиении и одинаковыми гиперпараметрами. На выходе каждой была сигмоида и 14 патологий для классификации.

Так как для каждой патологии решается задача бинарной классификации, качество классификации будет оцениваться по ROC и AUC.

ROC-кривая — график, который позволяет оценить качество бинарной классификации. График показывает зависимость верно определенных патологий на снимке от количества ошибочных ответов при варьировании порога. Идеальным случаем для классификатора является проход графика через точку (0,1). Графики по 5 лучшим результатам представлены на рисунке 3.1. Красной линией показано положение графика, при котором сеть просто угадывает ответ и ничему не научилась.

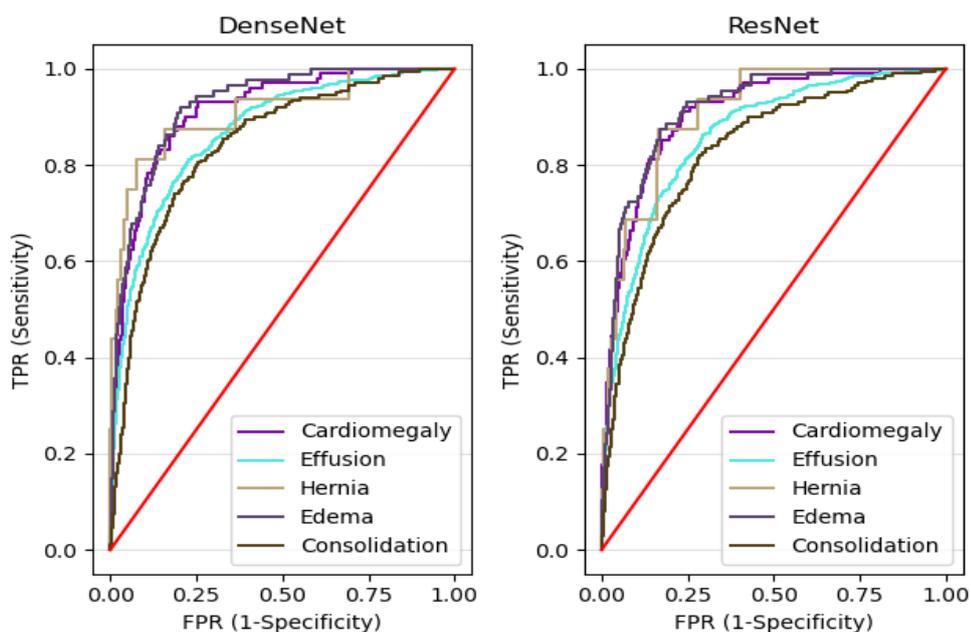


Рисунок 3.1 – ROC-кривые по 5 патологиям

Для получения численного значения эффективности работы сети используется показатель AUC (Area Under Curve), являющейся площадью под ROC-кривой.

AUC можно рассматривается как вероятность того, что сеть верно распознает патологию на снимке.

Результаты обучения представлены в таблице 3.2.

Таблица 3.2 – Сравнение архитектур для задачи классификации

Патология	AUC для DenseNet201	AUC для ResNet152
Cardiomegaly	0.912	0.907
Emphysema	0.829	0.792

Продолжение таблицы 3.2.

Патология	AUC для DenseNet201	AUC для ResNet152
Effusion	0.868	0.861
Hernia	0.909	0.909
Infiltration	0.708	0.694
Mass	0.852	0.83
Nodule	0.703	0.66
Atelectasis	0.8	0.78
Pneumothorax	0.864	0.849
Pleural Thickening	0.805	0.803
Pneumonia	0.75	0.703
Fibrosis	0.81	0.784
Edema	0.924	0.917
Consolidation	0.841	0.83
Среднее	0.827	0.809

Как видно из таблицы обе сети примерно равны с небольшим превосходством DenseNet201. Но ResNet152 при этом имеет намного больше параметров и соответственно вес, а также скорость работы уступающую DenseNet201 больше чем в 2 раза. Исходя из этого для классификации была выбрана архитектура DenseNet201.

3.2 Выделение области интереса

Чтобы помочь рентгенологу в обнаружении патологии в работу классификатора было добавлена возможность локализовать патологию на снимке с помощью Grad-CAM (Gradient-weighted Class Activation Mapping) [19], рисунок 3.2.

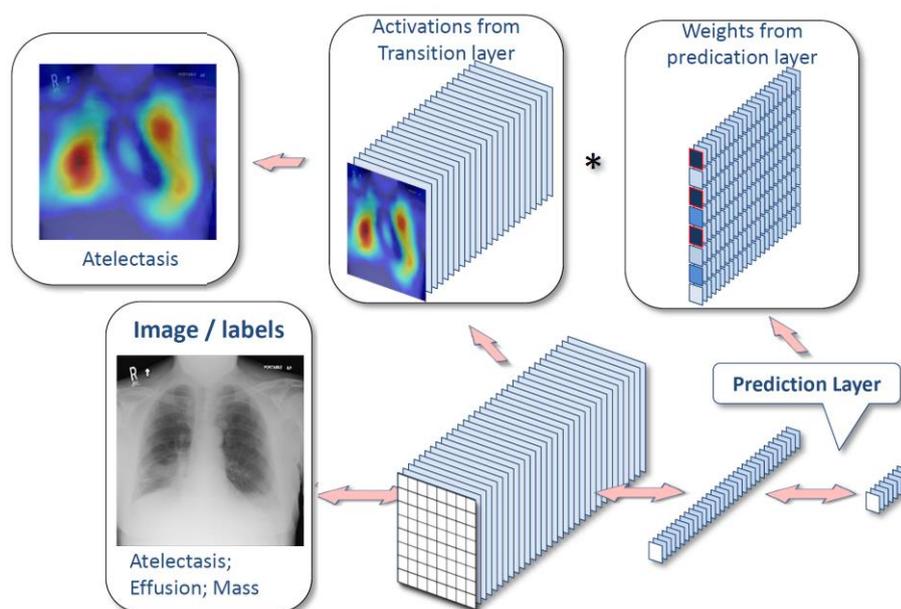


Рисунок 3.2 – Grad-CAM

Grad-CAM – помогает понять из-за чего сеть дала то или иное решение. Идея метода заключается в том, что на последнем слое свертки перед обобщающим пулингом и полносвязным слоем получается множество фильтров определяющих ту или иную патологию (1024 для DenseNet). Просуммировав свертки с этими фильтрами и нормализовав на определенном диапазоне можно получить матрицу со значениями давшими ответ сети. Получившуюся матрицу расширяют до размера исходного изображения и накладывают на него. Из-за такого расширения точность получившейся области не высокая, но она позволяет примерно определить место на снимке давшее ответ модели.

Пример работы Grad-CAM показан на рисунке 3.3, с легкими здорового человека, больного пневмонией и туберкулезом соответственно.

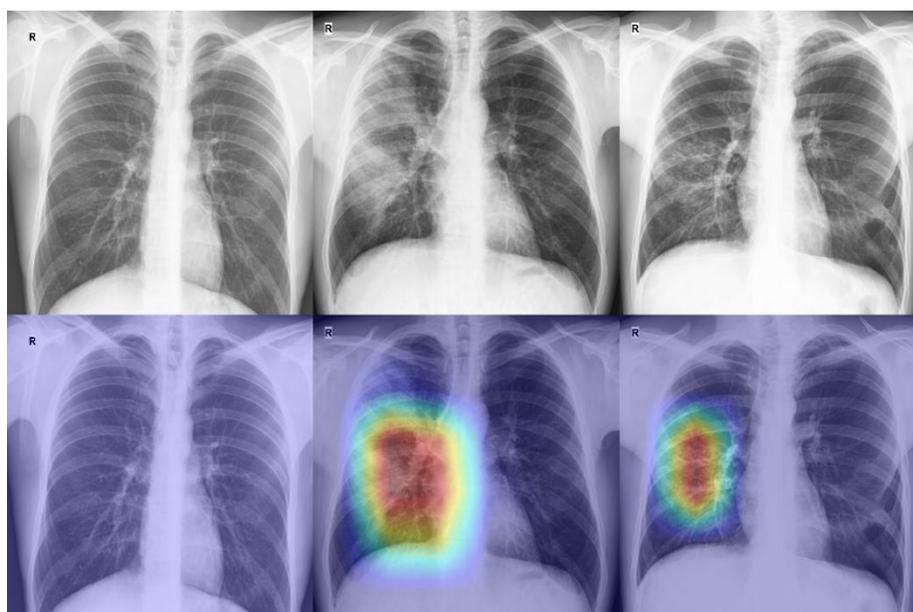


Рисунок 3.3 – Результат работы Grad-CAM

Для здоровых легких выход модели в идеальном случае будет состоять из вектора с 14 нулями. Значения матриц при свертках со всеми фильтрами в таком случае будут близки к нулевым матрицам и давать при накладывании монотонное изображение без выделяющихся областей. Если же модель предсказала нахождение на снимке патологии определенные фильтры будут давать отклик на областях с этой патологией.

ЗАКЛЮЧЕНИЕ

Целью выпускной квалификационной работы являлась повышение степени автоматизации в анализе рентгенограммы и разработка приложения для этого. Создание данного приложения позволяет классифицировать рентгеновские снимки груди по патологиям на 14 классов. На основе анализа установлено, что задачу распознавания можно успешно решить с помощью сверточных нейронных сетей, которые обладают рядом преимуществ для работы с изображениями.

В рамках данной выпускной квалификационной работы были достигнуты следующие результаты:

1. произведен анализ работы рентгенолога и методов, которые он использует в работе, выявлены проблемы и предложено их решение, а именно использование нейросетевого подхода;

2. были исследованы основные методы систем распознавания и архитектуры нейронных сетей;

3. выбран и подготовлен подходящий набор данных для обучения;

4. для обучения был разработан соответствующий алгоритм, выбраны наиболее подходящие архитектуры нейронных сетей для задачи;

5. проведены эксперименты посредством обучения нейросетей на данных, по результатам которых была обучена модель сверточной нейронной сети построенная по архитектуре DenseNet201 и показавшая наилучшие результаты классификации при анализе рентгенограммы, а именно точность порядка 0.827 по AUC при классификации по 14 патологиям грудной клетки;

6. был реализован метод Grad-CAM позволяющий локализовать патологии если таковые имеются.

Разработанная сеть может быть использована специалистами для уменьшения вероятности пропуска патологии за счет концентрирования их внимания на определенной области с предположением о находящейся там патологии.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. European Lung white book [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: www.erswhitebook.org
2. Рентгенология [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://teleradiologia.ru/что-такое-рентгенология-радиология/>
3. Методы исследования легких и дыхательных путей [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://delhimodi.com/other/funktsionalnoe-issledovanie-legkih.html>
4. Хофер М., Абанадор Н., Кампер Л., Центаи К. Рентгенологическое исследование грудной клетки / Перевод с Английского: В.В. Ипатов, А. П. Кутько // под. Ред.: Г. Е. Труфанова. —М, 2008. — 220 с.
5. Никитин О.Р., Пасечник А.С. Оконтуривание и сегментация в задачах автоматизированной диагностики патологий // Методы и устройства передачи и обработки информации. 2009. № 11. 300–309 с.
6. Kosykh N.E., Gostuyshkin V.V., Savin S.Z., Vorozhtov I.V. Designing the systems of computer diagnostics of medical images // Proc. of The First Russia and Pacific Conference on Computer Technology and Applications (RPC 2010). Vladivostok, Russia. 6–9 September, 2010. 4 p.
7. Гостюшкин В.В., Коваленко В.Л., Косых Н.Э., Савин С.З. Методы повышения эффективности компьютерных автоматизированных технологий в задачах радионуклидной диагностики // Врач и информационные технологии. 2013. № 6. 42–48 с.
8. Косых Н.Э., Смагин С.И., Гостюшкин В.В., Савин С.З., Литвинов К.А. Система автоматизированного компьютерного анализа медицинских изображений // Информационные технологии и вычислительные системы. 2011. № 3. С.52–60.
9. McCulloch, W. S. A logical calculus of the ideas immanent in nervous activity /Warren S. McCulloch, Walter Pitts // Springer New York. — 1943.

10. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/ListenTerm1201415/sak2.pdf>
11. Yann LeCun Leon Bottou, Y. B. Gradient-based learning applied to document recognition / Yoshua Bengio Yann LeCun, Leon Bottou, Patrick Haffner // IEEE. - 1998
12. Karen Simonyan, Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://arxiv.org/abs/1409.1556>
13. Deep Residual Learning for Image Recognition [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://arxiv.org/abs/1512.03385>
14. Densely Connected Convolutional Networks [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://arxiv.org/abs/1608.06993>
15. Keras Applications [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://keras.io/api/applications>
16. NIH Chest X-rays [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://www.kaggle.com/nih-chest-xrays/data>
17. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://stanfordmlgroup.github.io/projects/chexnet/>
18. Adam: A Method for Stochastic Optimization [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://arxiv.org/abs/1412.6980>
19. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization [Электронный ресурс]. – Электрон. дан. – [2020]. – Режим доступа: <https://arxiv.org/abs/1610.02391>